



# Piano di Qualifica

Sons Of SWE - Progetto Marvin

sonsofswe.swe@gmail.com

## Informazioni sul documento

<b>Versione</b>	5.0.0
<b>Redazione</b>	Caldart Federico Favero Andrea Menegon Lorenzo Thiella Eleonora Cavallin Giovanni Stefano Panozzo Dalla Riva Giovanni
<b>Verifica</b>	Dalla Riva Giovanni Thiella Eleonora Cavallin Giovanni
<b>Approvazione</b>	Caldart Federico
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Vardanega Tullio Cardin Riccardo Gruppo Sons Of SWE

## Descrizione

Documento contenente le strategie adottate dal gruppo Sons Of SWE per raggiungere gli obiettivi qualitativi richiesti per il prodotto Marvin.



## Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
5.0.0	2018-07-12	Approvazione	Federico Caldart	<i>Responsabile di Progetto</i>
4.1.0	2018-07-11	Verificazione	Giovanni Cavallin	<i>Verificatore</i>
4.0.3	2018-07-09	Incremento della sezione <i>Appendice B</i>	Eleonora Thiella	<i>Verificatore</i>
4.0.2	2018-07-08	Incremento della sezione <i>Appendice C</i>	Lorenzo Menegon	<i>Verificatore</i>
4.0.1	2018-06-16	Revisione correttiva dei contenuti in seguito alla segnalazioni del committente	Giovanni Dalla Riva	<i>Verificatore</i>
4.0.0	2018-06-04	Approvazione	Lorenzo Menegon	<i>Responsabile di Progetto</i>
3.2.0	2018-06-01	Verificazione	Giovanni Cavallin	<i>Verificatore</i>
3.1.2	2018-05-28	Incremento della sezione <i>Appendice A e Appendice B</i>	Federico Caldart	<i>Verificatore</i>
3.1.1	2018-05-26	Incremento della sezione <i>Appendice C</i>	Stefano Panozzo	<i>Verificatore</i>
3.1.0	2018-05-20	Verificazione	Eleonora Thiella	<i>Verificatore</i>
3.0.1	2018-05-19	Revisione correttiva dei contenuti in seguito alla segnalazioni del committente	Giovanni Cavallin	<i>Verificatore</i>
3.0.0	2018-05-07	Approvazione	Federico Caldart	<i>Responsabile di Progetto</i>
2.1.0	2018-05-06	Verificazione	Giovanni Dalla Riva	<i>Verificatore</i>
2.0.1	2018-05-04	Incremento sezione resoconto	Giovanni Cavallin	<i>Verificatore</i>
2.0.0	2018-04-29	Approvazione	Giovanni Dalla Riva	<i>Responsabile di Progetto</i>
1.1.0	2018-04-28	Verificazione	Giovanni Cavallin	<i>Verificatore</i>
1.0.2	2018-04-27	Aggiunta sezione specifica dei test	Giovanni Cavallin	<i>Verificatore</i>
1.0.1	2018-04-26	Modifica obiettivi di qualità	Eleonora Thiella	<i>Verificatore</i>
1.0.0	2018-04-11	Approvazione	Stefano Panozzo	<i>Responsabile di Progetto</i>
0.2.0	2018-04-10	Verificazione	Eleonora Thiella	<i>Verificatore</i>
0.1.0	2018-04-09	Verificazione	Giovanni Dalla Riva	<i>Verificatore</i>
0.0.10	2018-04-09	Fine stesura <i>Appendice C</i>	Federico Caldart	<i>Verificatore</i>
0.0.9	2018-04-01	Fine stesura <i>Appendice B</i>	Federico Caldart	<i>Verificatore</i>
0.0.8	2018-04-01	Fine stesura <i>Appendice A</i>	Andrea Favero	<i>Verificatore</i>
0.0.7	2018-03-30	Conclusione stesura riguardante la parte relativa allo standard ISO\IEC 15504 ed inizio stesura delle parti relative allo standard ISO/IEC 9126 ed al ciclo di Deming dell' <i>Appendice A</i>	Andrea Favero	<i>Verificatore</i>
0.0.6	2018-03-28	Inizio stesura <i>Appendice C</i>	Lorenzo Menegon	<i>Verificatore</i>
0.0.5	2018-03-28	Inizio stesura <i>Appendice B</i>	Federico Caldart	<i>Verificatore</i>



Versione	Data	Descrizione	Autore	Ruolo
0.0.4	2018-03-28	Inizio stesura riguardante la parte relativa allo standard ISO\IEC 15504 <i>Appendice A</i>	Andrea Favero	<i>Verificatore</i>
0.0.3	2018-03-27	Fine stesura degli <i>obiettivi di qualità</i>	Lorenzo Menegon	<i>Verificatore</i>
0.0.2	2018-03-23	Inizio stesura degli <i>obiettivi di qualità</i>	Lorenzo Menegon	<i>Verificatore</i>
0.0.1	2018-03-17	Creato lo scheletro del documento e stesura <i>Introduzione</i>	Lorenzo Menegon	<i>Verificatore</i>



# Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Glossario	7
1.4	Riferimenti	7
1.4.1	Normativi	7
1.4.2	Informativi	8
1.5	Premessa	8
<b>2</b>	<b>Obiettivi di qualità</b>	<b>8</b>
2.1	Qualità di processo	8
2.1.1	Processo di pianificazione di progetto	9
2.1.1.1	Compiti assegnati	9
2.1.2	Processo di controllo e accertamento	9
2.1.2.1	Schedule Variance	9
2.1.2.2	Cost Variance	9
2.1.3	Processo di misurazione e miglioramento	10
2.1.3.1	SPICE	10
2.1.4	Processo di gestione dei rischi	10
2.1.4.1	Rischi non preventivati	10
2.1.5	Processo di progettazione dell'architettura software/di sistema	10
2.1.5.1	SFIN	11
2.1.5.2	SFOUT	11
2.1.6	Processi di progettazione di dettaglio e codifica del software	11
2.1.6.1	Complessità ciclomatica	11
2.1.6.2	CxSLOC - commenti per linee di codice	11
2.1.6.3	Parametri per metodo	11
2.1.6.4	Linee di codice per metodo	11
2.1.7	Processi di verifica e validazione	12
2.1.7.1	Copertura del codice	12
2.1.7.2	Copertura dei branch	12
2.2	Qualità di prodotto	12
2.2.1	Qualità di documento	12
2.2.1.1	Ortografia	12
2.2.1.2	Comprensibilità e leggibilità	12
2.2.1.3	Correttezza dei contenuti	13
2.2.1.4	Adesione alle norme interne	13
2.2.2	Qualità del prodotto software	13
2.2.2.1	Funzionalità	13
2.2.2.2	Affidabilità	13
2.2.2.3	Efficienza	14
2.2.2.4	Usabilità	14
2.2.2.5	Manutenibilità	14
2.2.2.6	Portabilità	14
<b>A</b>	<b>Qualità secondo gli standard</b>	<b>16</b>
A.1	Standard di processo: ISO/IEC 15504 - Software Process Improvement and Capability Determination	16
A.2	Standard di prodotto: ISO/IEC 9126	17
A.3	Ciclo di Deming	20
<b>B</b>	<b>Specifica dei test</b>	<b>22</b>
B.1	Test di validazione	22
B.1.1	Tracciamento Test di validazione - Requisiti	26



B.2	Test di sistema . . . . .	29
B.2.1	Tracciamento Test di sistema - Requisiti . . . . .	31
B.3	Test di integrazione . . . . .	32
B.4	Test di unità . . . . .	34
B.4.1	Tracciamento Test di unità - Metodi . . . . .	39
<b>C</b>	<b>Resoconto delle attività di verifica . . . . .</b>	<b>42</b>
C.1	Attività di analisi dei requisiti . . . . .	42
C.2	Attività di analisi dei requisiti in dettaglio . . . . .	42
C.3	Attività di prototipazione . . . . .	42
C.4	Attività di prototipazione in dettaglio . . . . .	42
C.5	Attività di progettazione finale e codifica . . . . .	42
C.6	Verifica dei documenti . . . . .	42
C.6.1	Errori ortografici . . . . .	42
C.6.2	Errori concettuali . . . . .	43
C.6.3	Errori di forma . . . . .	44
C.6.4	Indice Gulpease . . . . .	44
C.7	Verifica dei processi . . . . .	44
C.7.1	Schedule Variance . . . . .	44
C.7.2	Cost Variance . . . . .	45
C.7.3	Schedule Variance - processo di verifica . . . . .	46
C.7.4	Cost Variance - verifica . . . . .	46
C.7.5	Rischi non preventivati . . . . .	48
C.7.6	Complessità ciclomatica . . . . .	48
C.7.7	SFIN - Structural fan in . . . . .	49
C.7.8	CxSLOC - Commenti per linee di codice . . . . .	49
C.7.9	Parametri per metodo . . . . .	50
C.7.10	Linee di codice per metodo . . . . .	50
C.7.11	ROF - Requisiti obbligatori soddisfatti . . . . .	51
C.7.12	ROFS - Requisiti obbligatori e facoltativi soddisfatti . . . . .	51
C.7.13	Successo dei test . . . . .	52
C.7.14	Copertura del codice . . . . .	52
C.7.15	Copertura dei branch . . . . .	52



## Elenco delle figure

1	Ultimi 5 livelli di capacità di SPICE . . . . .	17
2	Caratteristiche associate al modello di qualità interna ed esterna . . . . .	18
3	Caratteristiche associate al modello di qualità in uso . . . . .	18
4	Ciclo di Deming . . . . .	20
5	Media degli errori ortografici. . . . .	43
6	Media degli errori concettuali . . . . .	43
7	Media degli errori di forma. . . . .	44
8	Media dell'indice di Gulpease; più il valore è alto, più ci si avvicina all'ottimalità. . . . .	44
9	Schedule Variance sui processi di stesura dei documenti, progettazione e codifica. . . . .	45
10	Cost Variance per il processo di documentazione e codifica. . . . .	45
11	Schedule Variance per il processo di verifica. . . . .	46
12	Cost Variance per il processo di verifica. . . . .	46
13	Rischi non preventivati negli ultimi due periodi. . . . .	48
14	Complessità ciclomatica. . . . .	48
15	Structural fan in, più il valore è alto, meglio è. . . . .	49
16	Commenti per linee di codice, un valore alto è positivo. . . . .	49
17	Numero di parametri per metodo, un numero basso indica meno complessità. . . . .	50
18	Linee di codice per metodo, un numero basso indica meno complessità. . . . .	50
19	Requisiti obbligatori soddisfatti. . . . .	51
20	Requisiti obbligatori e facoltativi soddisfatti. . . . .	51
21	Percentuale di successo dei test. . . . .	52



## Elenco delle tabelle

2	Test di validazione . . . . .	26
3	Tracciamento test di validazione - requisiti . . . . .	28
4	Test di sistema . . . . .	30
5	Tracciamento test di sistema - requisiti . . . . .	31
6	Test di integrazione . . . . .	33
7	Test di sistema . . . . .	39
8	Tracciamento test di unità - metodi . . . . .	41



# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di descrivere gli obiettivi di qualità, di processo e di prodotto da raggiungere nella realizzazione del progetto e le strategie di verifica e validazione adottate per il raggiungimento di tali obiettivi.

## 1.2 Scopo del prodotto

Lo scopo del prodotto è quello di realizzare un *prototipo<sub>G</sub>* di Uniweb come una *DApp<sub>G</sub>* in esecuzione su *Ethereum<sub>G</sub>*. I cinque attori principali che si rapportano con Marvin sono:

- Utente non autenticato;
- Università;
- Amministratore;
- Professore;
- Studente.

Il portale deve quindi permettere agli studenti di accedere alle informazioni riguardanti le loro carriere universitarie, di iscriversi agli esami, di accettare o rifiutare voti e di poter vedere il loro libretto universitario. Ai professori deve invece essere permesso di registrare i voti degli studenti. L'università ogni anno crea una serie di corsi di laurea rivolti a studenti, dove ognuno di essi comprende un elenco di esami disponibili per anno accademico. Ogni esame ha un argomento, un numero di crediti e un professore associato. Gli studenti si iscrivono ad un corso di laurea e tramite il libretto elettronico mantengono traccia ufficiale del progresso.

## 1.3 Glossario

Nel documento *Glossario.v3.0.0* i termini tecnici, gli acronimi e le abbreviazioni sono definiti in modo chiaro e conciso, in modo tale da evitare ambiguità e massimizzare la comprensione dei documenti.

I vocaboli presenti in esso saranno posti in corsivo e presenteranno una "G" maiuscola a pedice.

## 1.4 Riferimenti

### 1.4.1 Normativi

- **Capitolato d'appalto C6 - Marvin: dimostratore di Uniweb su Ethereum** <http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C6.pdf>;
- *NormeDiProgetto\_v4.0.0*;
- **ISO/IEC 15504 - SPiCE, Plays-In-Business** <http://www.plays-in-business.com/isoiec-15504-spice/>;
- **Software Process Improvement and Capability Determination (SPICE) - ISO/IEC 15504** <https://shahanali.wordpress.com/2011/04/25/software-process-improvement-and-capability-determination/>;
- **La qualità del software secondo il modello ISO/IEC 9126**, Ercole F. Colonese [http://www.colonese.it/00-Manuali\\_Pubblicatii/07-ISO-IEC9126\\_v2.pdf](http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf);
- **Lo Standard ISO/IEC 9126 – Software engineering – Product Quality**, Anna Rita Fasolino <http://www.federica.unina.it/ingegneria/ingegneria-software-ii/isoiec-9126-software-engineering-c>



### 1.4.2 Informativi

- *PianoDiProgetto\_v4.0.0*;
- Slides del corso di Ingegneria del software - Qualità del prodotto software <http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L13.pdf>;
- Slides del corso di Ingegneria del software - Qualità di processo <http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L15.pdf> <http://www.math.unipd.it/~tullio/IS-1/2017/>;
- Ingegneria del software, decima edizione - Ian Sommerville, capitolo 21;
- SLOC [https://it.wikipedia.org/wiki/Source\\_lines\\_of\\_code](https://it.wikipedia.org/wiki/Source_lines_of_code);
- Ciclo di Deming [https://it.wikipedia.org/wiki/Ciclo\\_di\\_Deming](https://it.wikipedia.org/wiki/Ciclo_di_Deming) (aggiornato al 2018/03/31);
- Deming Cycle: The Wheel of Continuous Improvement <https://totalqualitymanagement.wordpress.com/2009/02/25/deming-cycle-the-wheel-of-continuous-improvement/>;
- Indice Gulpease [https://it.wikipedia.org/wiki/Indice\\_Gulpease](https://it.wikipedia.org/wiki/Indice_Gulpease) (aggiornato al 2018/03/31);
- Complessità ciclomatica [https://it.wikipedia.org/wiki/Complessit%C3%A0\\_ciclomatica](https://it.wikipedia.org/wiki/Complessit%C3%A0_ciclomatica) (aggiornato al 2018/03/31);
- Metriche di progetto <http://www.aivosto.com/project/help/pm-index.html>

### 1.5 Premessa

Questo documento sarà soggetto a continue aggiunte e modifiche per tutta la durata del progetto ed è da considerarsi *incrementale*<sub>G</sub> data la natura del suo contenuto. In particolare le sezioni riguardanti la specifica dei test ed il resoconto delle sottoattività di verifica saranno aggiornate in base ai risultati ottenuti nel proseguire del tempo.

## 2 Obiettivi di qualità

Questa sezione ha l'obiettivo di definire le caratteristiche riguardanti la qualità di prodotto e di processo che dovranno essere perseguite durante lo sviluppo del progetto. Ogni caratteristica viene valutata da una metrica, una soglia di accettabilità e una possibile soglia di miglioramento che il *team*<sub>G</sub> si prefigge di raggiungere e possibilmente superare; descrizione e formula per il calcolo di tali metriche sono disponibile nel documento *NormeDiProgetto\_v4.0.0*.

### 2.1 Qualità di processo

La qualità di processo influenza direttamente il prodotto finale realizzato. È necessario quindi sviluppare un processo in grado di produrre ciclicamente un prodotto di alta qualità. Per questo motivo si è deciso di stabilire le seguenti caratteristiche da rispettare per tutto lo sviluppo del progetto, contemporaneamente a:

- L'applicazione del *Ciclo di Deming*<sub>G</sub>, o *PDCA*, al fine di perseguire il miglioramento continuo delle attività di processo;
- L'adesione allo standard ISO/IEC 15504, denominato *SPICE*<sub>G</sub>, al fine di applicare una valutazione oggettiva sulla maturità dei processi.

Inoltre, facendo riferimento allo standard ISO/IEC 12207:2008, il team ha identificato i processi più significativi della sezione *Project Processes* (§6.3) e *Software Implementation Processes* (§7.1) per i quali verranno definiti gli obiettivi ed in seguito le metriche utilizzate per verificarne il raggiungimento.

### 2.1.1 Processo di pianificazione di progetto

Scopo di tale processo è produrre e comunicare un piano di progetto fattibile ed efficace, stabilendo le attività che dovranno essere svolte, con i relativi costi e scadenze, in ogni fase del progetto; tali responsabilità saranno in mano al *Responsabile*, il quale dovrà assicurare che:

- Costi e scadenze pianificati relativi ad ogni attività siano definiti.
- Le attività siano divise in compiti, ognuno dei quali dovrà essere assegnato ad un membro del team.
- Gli obiettivi di progetto siano definiti in maniera ponderata, tenendo conto delle risorse a disposizione del team.

#### 2.1.1.1 Compiti assegnati

Tale metrica verrà utilizzata per controllare l'assegnazione dei compiti all'interno del team.

- **Soglia di accettabilità e ottimalità:** si ritiene doveroso avere ogni compito in ogni fase assegnato ad un membro del team, dunque accettabilità ed ottimalità coincideranno con un risultato = 100%.

### 2.1.2 Processo di controllo e accertamento

Scopo di tale processo è determinare lo stato del progetto ed assicurare che esso sia in linea con la pianificazione prevista nel *PianoDiProgetto\_v4.0.0*, per quanto riguarda il rispetto di tempi e costi preventivati. In particolare, per il team è fondamentale rispettare le scadenze ed i budget previsti: andranno effettuati quindi dovuti controlli attraverso un continuo monitoraggio della situazione corrente di ogni attività di progetto. Nel caso in cui venissero rilevati valori non accettabili a seguito del calcolo della  $Schedule\ Variance_G$  o  $Cost\ Variance_G$ , sarà compito del team sanarli il prima possibile, compensando con un risparmio (economico o temporale rispettivamente) entro la scadenza finale di consegna: si dovrà prestare particolare attenzione ai costi, per i quali non è ammesso eccedere rispetto alla pianificazione. In generale si concretizzano dunque due obiettivi:

- Ogni attività dovrà essere portata a termine nella sua completezza entro le scadenze prefissate.
- Ogni attività dovrà essere portata a termine nella sua completezza senza eccedere i costi per essa preventivati; si cercherà se possibile di ottenere inoltre un risparmio orario (e quindi economico) in ogni fase, poiché anche un risparmio molto piccolo potrà essere utile a bilanciare eventuali eccedenze nei costi nelle fasi successive.

#### 2.1.2.1 Schedule Variance

Tale metrica verrà utilizzata per quanto riguarda il controllo delle scadenze temporali.

- **Soglia di accettabilità:** si è deciso di ritenere accettabile un ritardo massimo del 10% rispetto a quanto specificato nel *PianoDiProgetto\_v4.0.0*;
- **Soglia di ottimalità:** si ritiene un miglioramento rispetto all'obiettivo prefissato il caso in cui un lavoro venga portato a termine in anticipo rispetto a quanto specificato nel *PianoDiProgetto\_v4.0.0*.

#### 2.1.2.2 Cost Variance

Tale metrica verrà utilizzata per quanto riguarda il controllo del rispetto dei costi stabiliti.

- **Soglia di accettabilità:** sarà accettabile un risultato  $\leq 0$ .
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta nel caso in cui i costi saranno inferiori a quanto stabilito nel *PianoDiProgetto\_v4.0.0*, quindi un risultato  $< 0$ .

### 2.1.3 Processo di misurazione e miglioramento

Tale processo serve a valutare e migliorare la qualità del lavoro svolto.

#### 2.1.3.1 SPICE

Verrà utilizzata come metrica la struttura a 6 livelli che rappresenta la scala di maturità secondo SPICE; la misura di ogni livello sarà effettuata con i 4 livelli N, P, L e F definiti dallo standard.

- **Soglia di accettabilità:** il livello minimo accettabile di maturità della scala in riferimento ai processi è il 2 (Managed); il processo deve cioè fornire i risultati conformi agli standard e ai requisiti iniziali in maniera pianificata e tracciabile;
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta con il livello 4 (Predictable); il processo dovrà cioè essere eseguito in conformità ai principi dell'ingegneria del software e attuato all'interno di limiti ben definiti.

Per informazioni più approfondite riguardo allo standard ISO/IEC 15504 o SPICE, si rimanda alla sezione [Standard di processo: ISO/IEC 15504 - Software Process Improvement and Capability Determination](#) dell'appendice A.

### 2.1.4 Processo di gestione dei rischi

Tale processo verrà utilizzato per individuare, analizzare e monitorare e trattare i rischi in cui si potrebbe incorrere durante l'intera durata del progetto, in maniera continua, fin dalla prima fase del progetto, al fine di riconoscere e saper affrontare potenziali situazioni dannose. Si individuano i seguenti obiettivi:

- Sarà necessario, all'inizio di ogni fase di progetto, analizzare i possibili rischi specifici di tale fase, considerando i possibili nuovi pericoli introdotti dalle precedenti fasi;
- Alla luce dell'esperienza maturata nelle precedenti fasi di progetto, sarà necessario valutare l'entità delle ripercussioni dei problemi riscontrati, al fine di anticiparli e affrontarli efficacemente nel caso in cui si ripresentassero.
- Nel caso in cui i rischi possibili siano molti, sarà necessario che il *Responsabile* assegni loro diverse priorità, a seconda dell'entità delle possibili ripercussioni.

#### 2.1.4.1 Rischi non preventivati

Tale metrica verrà utilizzata per valutare l'efficacia della previsione dei rischi da parte del team.

- **Soglia di accettabilità:** verrà considerato come accettabile un valore compreso tra 0 e 3 in una singola fase di progetto.
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta con il valore 0.

### 2.1.5 Processo di progettazione dell'architettura software/di sistema

Scopo di tale processo è associare i requisiti del sistema alle singole parti del sistema, implementandole correttamente di conseguenza. L'architettura del sistema dovrà dunque essere sviluppata attraverso una progettazione che consenta la tracciabilità delle relazioni tra componenti e requisiti, nonché tra le componenti stesse; i seguenti obiettivi concretizzano quanto detto:

- Ogni componente progettata e sviluppata dovrà essere tracciabile in relazione all'intero sistema e al requisito che soddisfa.
- Ogni parte del sistema dovrà avere il più basso grado possibile di accoppiamento ed alta coesione.

#### 2.1.5.1 SFIN

Tale metrica sarà utile per verificare il riuso del codice.

- **Soglia di accettabilità:** verrà considerato come accettabile un valore compreso  $\geq 0$ .
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta con il valore  $\geq 2$ .

#### 2.1.5.2 SFOUT

Tale metrica sarà utile a calcolare il grado di accoppiamento delle varie componenti software.

- **Soglia di accettabilità:** verrà considerato come accettabile un valore  $\leq 4$ .
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta con il valore  $= 0$ .

### 2.1.6 Processi di progettazione di dettaglio e codifica del software

Scopo di tali processi è progettare e creare unità software che implementino e possano essere verificate rispetto ai requisiti, quindi sufficientemente dettagliate da permetterne esecuzione e test. Il team si impegna a raggiungere i seguenti obiettivi:

- Fornire una progettazione con un grado di dettaglio tale da poter produrre codice testabile.
- Produrre tutte le unità software previste dalla progettazione, le quali devono correttamente interagire tra loro e nel sistema, se previsto.
- Il codice prodotto dovrà avere bassa complessità, in modo da essere facilmente capibile, correggibile e mantenibile.

#### 2.1.6.1 Complessità ciclomatica

- **Accettabilità:** un valore di complessità compreso tra 1 e 15, purchè per valori tra 10 e 15 sia specificato il motivo di tale complessità;
- **Ottimalità:** un valore di complessità compreso tra 1 e 10.

#### 2.1.6.2 CxSLOC - commenti per linee di codice

- **Accettabilità:** sarà accettato un valore CxSLOC compreso tra 20 e 25;
- **Ottimalità:** sarà dichiarato ottimale un valore CxSLOC compreso tra 25 e 35.

#### 2.1.6.3 Parametri per metodo

- **Accettabilità:** saranno accettati metodi con un numero di parametri minore o uguale a 10;
- **Ottimalità:** saranno considerati ottimi metodi con un numero di parametri minore o uguale a 5.

#### 2.1.6.4 Linee di codice per metodo

- **Accettabilità:** saranno accettati metodi con una lunghezza pari o inferiore alle 50 righe;
- **Ottimalità:** saranno considerati ottimi metodi con una lunghezza pari o inferiore alle 30 righe.

### 2.1.7 Processi di verifica e validazione

Scopo di tali processi è accertare rispettivamente che l'esecuzione delle attività di processo attuate nel periodo in esame non abbiano introdotto errori e che il prodotto realizzato soddisfi i requisiti. Il team si impegna ad assicurare che:

- Siano correttamente applicate le tecniche di Walkthrough prima e di Inspection poi, come specificato nelle *NormeDiProgetto\_v4.0.0*, durante le attività di analisi statica;
- I test dinamici effettuati su codice e documenti saranno automatizzati il più possibile;
- I test dinamici effettuati sul codice copriranno il maggior numero possibile di casi possibili e di statement.

#### 2.1.7.1 Copertura del codice

- **Accettabilità:** sarà accettata un numero di statement testati pari al 70%;
- **Ottimalità:** sarà considerata ottima la capacità di testare almeno il 90% degli statement.

#### 2.1.7.2 Copertura dei branch

- **Accettabilità:** sarà accettata un numero di rami testati pari al 75%;
- **Ottimalità:** sarà considerata ottima la capacità di testare almeno il 95% dei rami per funzionalità non ancora testate, mentre per codice già testato l'ottimalità sarà data dalla capacità di testarne l'80%.

## 2.2 Qualità di prodotto

### 2.2.1 Qualità di documento

Il team si impegna a redigere dei documenti di alta qualità, rispettando le caratteristiche di forma e contenuto descritte di seguito.

#### 2.2.1.1 Ortografia

Un documento deve essere prima di tutto privo di errori dal punto di vista grammaticale e ortografico. Il primo controllo avverrà proprio durante la stesura del documento stesso, tramite il sistema di autocontrollo dell'ambiente *TexStudio*, per poi essere controllato una seconda volta dal *Verificatore*.

- **Metrica:** la quantità di errori riscontrata durante la verifica definitiva del documento sarà l'unità di misura presa in considerazione;
- **Soglia di accettabilità:** si è accettata come tollerabile la presenza di massimo 3 errori nella seconda e definitiva verifica da parte del *Verificatore*;
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta nel caso in cui dopo la prima revisione del documento non vengano più riscontrati errori dal *Verificatore* e dal *Responsabile di Progetto*.

#### 2.2.1.2 Comprensibilità e leggibilità

Poiché un documento venga considerato leggibile e scorrevole si è deciso di adottare l'*Indice Gulpease<sub>G</sub>* al fine di avere un parametro oggettivo e facilmente misurabile.

- **Metrica:** l'unità di misura utilizzata è l'*Indice Gulpease*;
- **Soglia di accettabilità:** verrà considerato come accettabile un valore di 45 sulla scala dell'*Indice Gulpease*;
- **Soglia di ottimalità:** la soglia di ottimalità verrà raggiunta nel caso in cui l'*Indice Gulpease* sia maggiore di 60.

### 2.2.1.3 Correttezza dei contenuti

Oltre ad essere corretto nella forma, un documento necessita di un contenuto adeguato dal punto di vista argomentativo. Gli *Analisti* saranno direttamente responsabili della qualità del contenuto, che poi verrà controllato e corretto dal *Verificatore*. Per verificare la correttezza concettuale dei documenti prenderemo in esame i seguenti parametri:

- **Metrica:** la quantità di errori di contenuto riscontrata durante la verifica definitiva del documento sarà l'unità di misura presa in considerazione;
- **Soglia di accettabilità:** si è accettata come tollerabile la presenza di massimo 3 errori nella seconda e definitiva verifica da parte del *Verificatore*;
- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta nel caso in cui non si riscontrino errori durante la verifica definitiva del documento.

### 2.2.1.4 Adesione alle norme interne

Al fine di ottenere un prodotto coerente ogni documento dovrà essere redatto rispettando strettamente quanto dichiarato nelle *NormeDiProgetto\_v4.0.0*. Qualunque riferimento non attinente o in contrasto a quanto dichiarato verrà considerato un errore.

- **Metrica:** la quantità di errori di adesione alle norme interne riscontrata durante la verifica definitiva del documento sarà l'unità di misura presa in considerazione;
- **Soglia di accettabilità:** si è accettata come tollerabile la presenza di massimo 3 errori nella seconda e definitiva verifica da parte del *Verificatore*;
- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta nel caso in cui non si riscontrino errori dopo la prima verifica del documento.

## 2.2.2 Qualità del prodotto software

Come detto in precedenza, è impossibile distinguere in maniera netta la qualità di processo dalla qualità del software, in quanto la prima influenza direttamente la seconda; è dunque fondamentale avere alla base una qualità di processo sufficientemente buona per garantire la qualità del prodotto. Nonostante ciò, è necessario stabilire degli obiettivi quantitativi di qualità del software oggettivi e misurabili. A tal fine verrà seguito lo standard ISO/IEC 9126, il quale si sostanzia nei seguenti sei punti:

### 2.2.2.1 Funzionalità

È un requisito funzionale che indica la capacità del software di soddisfare le esigenze espresse dal capitolato ed individuate durante l' *AnalisiDeiRequisiti\_v4.0.0*. Per valutare la funzionalità del software prenderemo in considerazione i seguenti parametri:

- **Metrica:** la valutazione si baserà sul numero di requisiti soddisfatti;
- **Soglia di accettabilità:** il prodotto verrà valutato come accettabile se tutti i requisiti obbligatori saranno soddisfatti;
- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta nel caso in cui siano soddisfatti sia i requisiti obbligatori che tutti i requisiti opzionali.

### 2.2.2.2 Affidabilità

È un requisito non funzionale che indica la capacità del software di svolgere correttamente il suo compito mantenendo delle buone prestazioni anche al variare dell'ambiente nel tempo. Per valutare l'affidabilità del software prenderemo in considerazione i seguenti parametri:

- **Metrica:** la valutazione si baserà sul numero di fallimenti durante la fase di test;

- **Soglia di accettabilità:** il prodotto verrà valutato come accettabile se i test falliti saranno inferiori o uguali al 5%;
- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta nel caso in cui il 100% dei test avrà dato l'esito desiderato.

#### 2.2.2.3 Efficienza

È un requisito non funzionale che valuta la capacità di un prodotto software di realizzare le funzioni richieste nel minor tempo possibile e con l'uso minimo di risorse necessarie.

- **Metrica:** la valutazione si baserà sui secondi impiegati dal prodotto per eseguire le richieste dell'utente;
- **Soglia di accettabilità:** la soglia di accettabilità è il periodo tra 0 e 10 secondi;
- **Soglia di ottimalità:** la soglia di ottimalità è 1 secondo.

#### 2.2.2.4 Usabilità

L'usabilità è un requisito non funzionale che indica la capacità del software di essere capito e usato correttamente da parte dell'utente finale. Dato che il prodotto finale sarà per l'utente un portale web, è impossibile trovare una metrica quantificabile per valutarne l'usabilità: essa dipende da molteplici fattori che coinvolgono anche le capacità dell'utente stesso e gli strumenti a sua disposizione. Verrà dunque valutata in modo oggettivo basandosi sugli standard del web dichiarati dal  $W3C_G$  e sugli strumenti che tale organizzazione mette a disposizione, al fine di creare un'interfaccia web il più accessibile possibile. Prenderemo in considerazione i seguenti parametri:

- **Metrica:** la valutazione si baserà sul numero di errori trovati dagli strumenti del  $W3C$ ;
- **Soglia di accettabilità:** la soglia di accettabilità è di 2 errori rilevati;
- **Soglia di ottimalità:** il prodotto sarà dichiarato ottimo se saranno rilevati 0 errori.

Quanto detto non assicura però una valutazione completa dell'usabilità, la quale è soggettiva; sarà necessario dunque predisporre test specifici per la misurazione, coinvolgendo ad esempio persone esterne al gruppo al fine di stabilire quanto mediamente il software sia capibile. Al momento, tuttavia, il team non è in grado di stabilire con precisione una metrica adatta a misurare questo risultato.

#### 2.2.2.5 Manutenibilità

La manutenibilità è un requisito non funzionale che indica la capacità di un prodotto di essere evolvibile nel tempo attraverso correzioni, miglioramenti ed aggiunte.

- **Metrica:** saranno usate le metriche riguardanti il codice, dato che esso influenza direttamente la manutenibilità del software;
- **Soglia di accettabilità:** la soglia di accettabilità sarà raggiunta se il prodotto raggiungerà tale soglia in tutte le metriche utilizzate per il codice;
- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta se il prodotto raggiungerà tale soglia in tutte le metriche utilizzate per il codice.

#### 2.2.2.6 Portabilità

La portabilità è un requisito non funzionale che indica la capacità del prodotto di operare in  $ambienti_G$  diversi, limitando le necessità di apportare cambiamenti.

- **Metrica:** la valutazione si baserà sul numero di versioni di  $browser_G$  e numero di browser stessi su cui il prodotto riesce a venire utilizzato e visualizzato correttamente;
- **Soglia di accettabilità:** la soglia di accettabilità sarà raggiunta se il prodotto sarà supportato correttamente, offrendo la totalità delle sue funzionalità, dalla versione 60.3.3112 o successive di *Google*



*Chrome<sub>G</sub>* (*metamask<sub>G</sub>* versione 4.6 o superiore) e dalla versione 50 o superiore di *Mozilla Firefox<sub>G</sub>* (*metamask* versione 4.5 o superiore);

- **Soglia di ottimalità:** la soglia di ottimalità sarà raggiunta se il prodotto sarà supportato correttamente, offrendo la totalità delle sue funzionalità in aggiunta ai sopra citati, da *Opera<sub>G</sub>* nella versione 52 o superiore (*metamask* versione 3.13.4 o superiore).



## A Qualità secondo gli standard

Al fine di perseguire la qualità secondo quanto descritto in questo documento, si è deciso di basarsi su degli standard (descritti qui di seguito) per poter bilanciare la poca esperienza del team con la conoscenza presente in tali documenti, ricavata da anni di pratica nell'ambito dell'ingegneria del software.

### A.1 Standard di processo: ISO/IEC 15504 - Software Process Improvement and Capability Determination

La qualità di un prodotto software dipende dalla qualità dei suoi processi. L'*ISO/IEC 15504 - Software Process Improvement and Capability Determination* o *SPICE* è uno standard che permette di valutare i processi software di un prodotto con lo scopo di migliorarli (in modo continuativo). La valutazione dei processi permette di identificare in modo indipendente la *capacità<sub>G</sub>* (capability) di ciascuno di essi attraverso i loro attributi (ovvero gli esiti della valutazione). Basandoci su tali risultati di valutazione, che devono essere comparabili, ripetibili ed oggettivi, ci si può aspettare un miglioramento *continuativo* dei processi e si possono identificare i loro punti di forza, di debolezza, i rischi ed i modi per prevenirli.

Ad ogni processo viene assegnato un livello di capacità a seconda della classificazione dei suoi attributi:

**0 - incomplete:** il processo presenta una incapacità generale nel raggiungere il proprio obbiettivo. A questo livello di capacità non viene associato alcun attributo;

**1 - performed:** il processo è riuscito a raggiungere il proprio obbiettivo. Il raggiungimento di tale obbiettivo potrebbe non essere stato pianificato e tracciato in modo rigoroso. A questo livello è associato l'attributo **process performance**;

**2 - managed:** il processo (che appartiene anche al livello 1) rilascia i propri prodotti secondo procedure specifiche ed è pianificato e tracciato. I prodotti sono conformi agli standard specificati ed ai requisiti. A questo livello sono associati due attributi: **performance management** e **work product management**;

**3 - established:** il processo (che appartiene anche al livello 2) viene implementato utilizzando dei buoni principi di ingegneria del software ed è in grado di raggiungere i medesimi risultati ogni volta che viene eseguito. A questo livello sono associati due attributi: **process definition** e **process deployment**;

**4 - predictable:** il processo (che appartiene anche al livello 3) viene eseguito nella pratica in modo coerente rimanendo dentro ai limiti di controllo che sono stati definiti per raggiungere il suo obbiettivo. Il livello ha associati gli attributi **process controll** e **process measurement**;

**5 - optimizing:** le performance del processo (che appartiene anche al livello 4) sono ottimizzate in modo continuo per andare incontro agli obbiettivi ed alle necessità (bisogni) di progetto o di business aziendali presenti e futuri. Anche a questo livello sono associati due attributi: **process innovation** e **process optimization**.

I 9 attributi che servono per misurare la capacità di un processo sono definiti nel seguente modo:

**Process performance:** è una misura che indica il raggiungimento degli obbiettivi del processo;

**Performance management:** è una misura che indica come sono gestite le performance del processo;

**Work product management:** è una misura che indica quanto i prodotti del processo siano gestiti in modo appropriato;

**Process definition:** è una misura che indica quanto il processo sia effettivamente impegnato a rispettare gli standard quando produce i propri esiti;

**Process deployment:** è una misura di quanto il processo standard venga diffuso efficacemente per raggiungere i propri risultati;

**Process measurement:** è una misura che indica quanto vengano usate le misurazioni dei risultati del processo per assicurarsi che le sue performance supportino il raggiungimento degli obiettivi aziendali fissati;

**Process control:** è una misura che dà un'indicazione di quanto il processo sia gestito in modo quantitativo, in modo tale da produrre un processo che sia stabile, capace<sup>1</sup> e prevedibile entro i limiti definiti;

**Process innovation:** è una misura di quanto i cambiamenti apportati al processo siano identificati grazie ad analisi di cause comuni delle variazioni delle performance e da indagini di approcci innovativi per le definizioni e lo sviluppo dei processi;

**Process optimization:** è una misura che indica quanto i cambiamenti alla definizione, gestione e performance del processo abbiano un impatto effettivo che permetta di raggiungere gli obiettivi rilevanti di miglioramento del processo.

Ogni attributo di processo viene valutato attraverso una scala di valutazione di quattro livelli<sup>2</sup>. Il punteggio è basato sulle prove raccolte tramite degli indicatori che permettono di sapere in quale livello della classifica sia posizionato l'attributo.

**N - Not achieved:** (0% - 15%);

**P - Partially achieved:** (15% - 50%);

**L - Largely achieved:** (50% - 85%);

**F - Fully achieved:** (85% - 100%).

Per raggiungere un certo livello di capacità, tutti gli attributi di processo del livello in questione devono essere realizzati almeno come "L" e tutti gli attributi di tutti i livelli di capacità sottostanti devono essere "F".

In Figura 1 sono rappresentati gli ultimi cinque livelli di capacità dei processi di SPICE ed i relativi attributi ad essi associati.

	Level	Attributes
5	"Optimising"	<ul style="list-style-type: none"> <li>Process innovation</li> <li>Continuous optimization</li> </ul>
4	"Predictable"	<ul style="list-style-type: none"> <li>Process measurement</li> <li>Process control</li> </ul>
3	"Established"	<ul style="list-style-type: none"> <li>Process definition</li> <li>Process deployment</li> </ul>
2	"Managed"	<ul style="list-style-type: none"> <li>Performance management</li> <li>Work product management</li> </ul>
1	"Performed"	<ul style="list-style-type: none"> <li>Process performance</li> </ul>

Figura 1: Ultimi 5 livelli di capacità di SPICE

## A.2 Standard di prodotto: ISO/IEC 9126

Per valutare la qualità del prodotto software si è deciso di utilizzare lo standard ISO/IEC 9126, che è suddiviso in 4 parti:

<sup>1</sup>vedere definizione di *capacità<sub>G</sub>*

<sup>2</sup>Descritti nella parte 3 dello standard

**Quality Model:** caratteristiche di qualità che possono essere usate per descrivere i fattori di qualità di un prodotto software;

**External Metrics:** metriche non misurabili direttamente che possono essere usate per valutare se il prodotto software è conforme al modello di qualità;

**Internal Metrics:** metriche direttamente misurabili utilizzabili per valutare le external metrics;

**Quality in Use Metrics:** metriche rivolte alla valutazione del sottoinsieme di caratteristiche di qualità legate all'utente.

Secondo lo standard sono necessari tre punti di vista per valutare la qualità del prodotto software:

- **Percepita/in uso:** è correlata a ciò che percepisce l'utente e per questo motivo definisce delle metriche che possono essere applicate solamente quando il prodotto è finito. Tali metriche esprimono l'efficacia e l'efficienza con cui il software serve le esigenze del suo utilizzatore;
- **Esterna:** rappresenta le prestazioni del prodotto e le funzionalità che esso offre. Definisce le metriche che esprimono il comportamento dinamico del software e per questo motivo è rilevata attraverso l'analisi dinamica e determina la qualità in uso. È una misura dell'interazione tra il cliente ed il prodotto in un contesto d'uso specifico e permette di osservare il comportamento del software mentre questo viene utilizzato;
- **Interna/intrinseca:** rappresenta le qualità intrinseche del prodotto, ovvero quelle misurabili direttamente dal codice sorgente attraverso un'analisi di tipo statico. Si realizza partendo dalle specifiche di qualità fornite dall'utente e da quelle tecniche tradotte dallo sviluppatore nell'architettura del software.

Gli attributi di qualità interni influenzano alcuni degli attributi di qualità esterni e quest'ultimi influenzano quelli della qualità in uso.

Per descrivere i tre punti di vista di cui sopra, lo standard definisce due modelli, uno che riguarda la qualità interna ed esterna ed un altro che riguarda la qualità in uso. Tali modelli presentano due livelli di caratteristiche (primo e secondo) che definiscono la qualità. Le caratteristiche del secondo livello sono sotto-caratteristiche del primo livello e vengono valutate rispetto alle metriche interne ed esterne. In Figura 2 e in Figura 3 si possono vedere le caratteristiche associate ai due modelli.

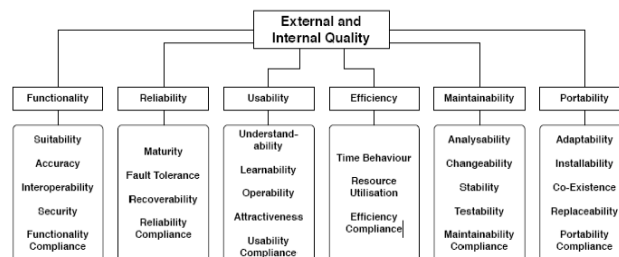


Figura 2: Caratteristiche associate al modello di qualità interna ed esterna

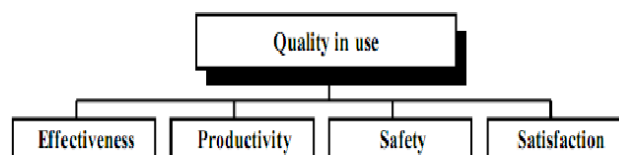


Figura 3: Caratteristiche associate al modello di qualità in uso

Di seguito sono elencate le caratteristiche di primo e secondo livello del modello per la qualità interna ed esterna.

**Functionality (funzionalità):** Il prodotto deve essere in grado di fornire delle funzioni che soddisfino esigenze stabilite (ovvero emerse dall'*Analisi Dei Requisiti\_v4.0.0*);

- **Suitability (appropriatezza):** capacità del prodotto di fornire all'utente delle funzioni in grado di soddisfare le esigenze stabilite ed implicite;
- **Accuracy (accuratezza):** capacità del prodotto di fornire risultati corretti con la precisione richiesta;
- **Interoperability (interoperabilità):** capacità del prodotto di interagire con uno o più sistemi specificati;
- **Security (sicurezza):** capacità del prodotto di proteggere i dati e le informazioni in modo che persone/sistemi non autorizzate/i riescano ad accedervi in lettura o scrittura;
- **Compliance (conformità):** capacità del prodotto di aderire agli standard relativi alle funzionalità che offre.

**Reliability (affidabilità):** Il prodotto software deve mantenere un livello di prestazioni specificato quando viene eseguito sotto certe condizioni specificate;

- **Maturity (maturità):** capacità del prodotto di evitare anomalie;
- **Fault tolerance (tolleranza all'errore):** capacità del prodotto di mantenere un livello di prestazioni specificato nel caso in cui occorranو anomalie;
- **Recoverability (recuperabilità):** capacità del prodotto di recuperare un livello di prestazioni specificato ed i dati colpiti da malfunzionamenti;
- **Compliance (conformità):** capacità del prodotto di aderire a standard relativi all'affidabilità.

**Usability (usabilità):** Il prodotto software deve essere compreso ed utilizzato con gradimento dall'utente;

- **Understandability (comprensibilità):** il prodotto software permette di far capire all'utente se può essergli utile per dei compiti particolari;
- **Learnability (apprendibilità):** il prodotto è in grado di far apprendere all'utente come utilizzare le proprie applicazioni;
- **Operability (operabilità):** il prodotto permette all'utente di utilizzarlo e di esercitarne il controllo;
- **Attractiveness (attrattività):** la capacità del prodotto di attrarre l'utente suscitandone un certo livello di gradimento;
- **Compliance (conformità):** la capacità del prodotto di aderire agli standard di usabilità.

**Efficiency (efficienza):** Il prodotto sfrutta al massimo e al meglio le risorse di cui necessita per espletare le proprie funzioni;

- **Time behaviour (comportamento nel corso del tempo):** capacità del prodotto di fornire un tempo di risposta appropriato quando esegue le proprie funzioni;
- **Resource utilisation (utilizzo delle risorse):** la capacità del prodotto di usare la giusta quantità ed il giusto tipo di risorse quando esegue le proprie funzioni;
- **Compliance (conformità):** capacità del prodotto di soddisfare gli standard relativi all'efficienza.

**Maintainability (manutenibilità):** capacità del prodotto di evolversi nel tempo grazie a delle modifiche o correzioni;

- **Analysability (analizzabilità):** la capacità del prodotto di essere analizzato per scovare le cause dei malfunzionamenti;
- **Changeability (modificabilità):** la capacità del prodotto di essere modificato;
- **Stability (stabilità):** capacità del software di evitare malfunzionamenti dopo essere stato modificato;

- **Testability (testabilità):** capacità del software modificato di essere verificato e validato;
- **Compliance (conformità):** capacità del prodotto di soddisfare gli standard relativi alla manutenibilità.

**Portability (portabilità):** Il software deve poter essere trasferito da un ambiente ad un altro con l'avanzare delle nuove tecnologie.

- **Adaptability (adattabilità):** la capacità del prodotto di adattarsi ad ambienti diversi senza che ci sia il bisogno di applicare alcuna azione o di utilizzare mezzi diversi da quelli che sono stati già forniti;
- **Installability (installabilità):** la capacità del prodotto di essere installato in un ambiente specifico;
- **Co-existence (coesistenza):** la capacità del prodotto di coesistere con altri prodotti software indipendenti in uno stesso ambiente condividendone le risorse;
- **Replaceability (sostituibilità):** la capacità del prodotto di sostituire un altro prodotto software con gli stessi scopi nello stesso ambiente;
- **Compliance (conformità):** capacità del prodotto di soddisfare gli standard relativi alla portabilità.

Di seguito sono elencate le caratteristiche del modello per la qualità in uso che rappresentano il punto di vista dell'utente sulla qualità del prodotto software:

- **Effectiveness (efficacia):** permette agli utenti di raggiungere il proprio obiettivo portandolo a termine con accuratezza e completezza;
- **Productivity (produttività):** la capacità del prodotto di utilizzare una adeguata quantità di risorse garantendo efficienza;
- **Satisfaction (soddisfazione):** la capacità del prodotto software di soddisfare gli utenti;
- **Safety (sicurezza):** la capacità del prodotto di raggiungere livelli accettabili di rischio di danni a persone, software e ambiente operativo su cui è installato.

### A.3 Ciclo di Deming

Il ciclo di Deming, chiamato anche PDCA (Plan, Do, Check, Act.) è uno strumento che permette di realizzare il miglioramento continuo della qualità dei processi e quindi anche dei loro prodotti.

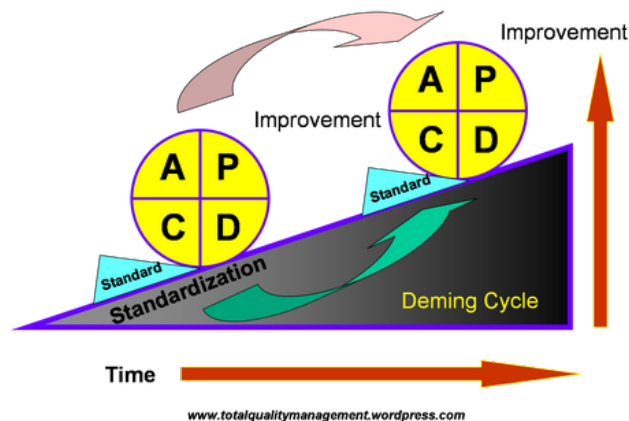


Figura 4: Ciclo di Deming

Come si può vedere in Figura 4 bisogna ripetere in modo iterativo i quattro passi *Plan*, *Do*, *Check* e *Act* per ottenere il miglioramento di un processo.



- **Plan:** vengono pianificati gli obiettivi per il miglioramento del processo. In questa fase viene analizzata la situazione attuale, vengono raccolti i dati e sviluppate delle metodologie per ottenere dei miglioramenti. Vengono definite le attività che bisogna svolgere, le risorse di cui esse necessitano e si fissano le scadenze.

Per fare ciò è necessario porsi tre domande durante la fase di planning:

- Cosa si sta cercando di realizzare?
  - Quali cambiamenti si possono fare per ottenere un miglioramento?
  - Come si è in grado di capire che un dato cambiamento rappresenta un miglioramento?
- **Do:** viene attuato il piano definito nella fase di Plan, in questo modo il processo viene eseguito e così viene creato il prodotto;
  - **Check:** viene controllato che il processo funzioni come pianificato, in particolare si confrontano i risultati misurati nella fase di Do con gli obiettivi stabiliti nella fase di Plan (ovvero i risultati attesi);
  - **Act:** il processo viene migliorato attuando se necessario delle azioni correttive che devono agire sulle differenze riscontrate tra i risultati attesi e quelli misurati.

Quando tutte queste quattro fasi vengono portate a termine con il massimo della soddisfazione, il miglioramento viene standardizzato. Il prodotto standardizzato è il risultato dell'iniziativa di miglioramento. È possibile che con il cambiamento di alcune circostanze, il processo sia soggetto ad un nuovo miglioramento, in questo modo il ciclo di deming viene ripetuto di volta in volta.

## B Specifica dei test

Questa sezione definirà i test che verranno implementati ed eseguiti dal team al fine di garantire, in seguito al loro superamento, la creazione di software di alta qualità che soddisfi le richieste ed aspettative del proponente. Ogni test viene riconosciuto tramite un identificativo univoco, la cui sintassi è descritta nelle *NormeDiProgetto\_v4.0.0*(§3.2.2) e può assumere uno tra questi stati: *Superato*, *Implementato*, *Superato*. Dato che il progetto è ancora alla fase iniziale, si presume che questa sezione sarà soggetta a modifiche basate sulle esigenze ed i problemi che si incontreranno durante lo sviluppo.

### B.1 Test di validazione

Questi test verranno utilizzati durante la fase di collaudo finale, in presenza di committente e proponente, per valutare se il prodotto è conforme a quanto specificato nel contratto.

Codice test	Descrizione	Stato
TV0F1A	Viene verificato che un utente non autenticato visualizzi la pagina di registrazione.	<i>Superato</i>
TV0F1B	Viene verificato l'inserimento nel campo reattivo al nome.	<i>Superato</i>
TV0F1C	Viene verificato l'inserimento nel campo reattivo al cognome.	<i>Superato</i>
TV0F1D	Viene verificato l'inserimento nel campo reattivo all'e-mail.	<i>Superato</i>
TV0F1E	Viene verificato l'inserimento nel campo reattivo al codice fiscale.	<i>Superato</i>
TV0F1F	Viene verificato l'inserimento nel campo reattivo al codice univoco.	<i>Superato</i>
TV0F1G	Viene verificato l'inserimento nel campo reattivo al matricola.	<i>Superato</i>
TV0F1H	Viene verificato che <i>metamask<sub>G</sub></i> sia attivato correttamente nel caso in cui un utente non autenticato voglia registrarsi.	<i>Superato</i>
TV0F1I	Viene verificato che venga confermata la registrazione.	<i>Superato</i>
TV0F1L	Viene verificato che la creazione sia andata a buon fine effettuando un login.	<i>Superato</i>
TV2F1.8	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente non autenticato riempi i campi dato relativi alla registrazione con informazioni già presenti, non valide o nulle.	<i>Superato</i>
TV0F2A	Viene verificato che <i>metamask<sub>G</sub></i> sia attivato correttamente nel caso in cui un utente non autenticato voglia autenticarsi dopo essersi registrato.	<i>Superato</i>
TV0F2B	Viene verificato che, nel caso in cui l'utente non autenticato clicchi il pulsante di login, il sistema segnali la corretta autenticazione.	<i>Superato</i>
TV0F3A	Viene verificato che l'utente amministratore o l'utente università siano correttamente autenticati rispettivamente come amministratore o come università.	<i>Superato</i>





Codice test	Descrizione	Stato
TV0F3B	Viene verificata la visualizzazione della pagina per l'inserimento di un anno accademico.	<i>Superato</i>
TV0F3C	Viene verificato l'inserimento dell'inizio dell'anno accademico.	<i>Superato</i>
TV0F3D	Viene verificato l'inserimento della fine dell'anno accademico.	<i>Superato</i>
TV0F3E	Viene verificato l'inserimento della nome dell'anno accademico.	<i>Superato</i>
TV0F3F	Viene verificata la conferma dell'aggiunta dell'anno accademico.	<i>Superato</i>
TV0F3G	Viene verificato che l'aggiunta dell'anno accademico sia andata a buon fine.	<i>Superato</i>
TV2F3.5	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente amministratore o università riempiano i campi dato relativi all'inserimento di un anno accademico con informazioni errate o nulle.	<i>Superato</i>
TV0F4A	Viene verificata la visualizzazione della pagina per l'inserimento di un corso di laurea.	<i>Superato</i>
TV0F4B	Viene verificato l'inserimento del codice del corso di laurea.	<i>Superato</i>
TV0F4C	Viene verificato l'inserimento del nome del corso di laurea.	<i>Superato</i>
TV0F4D	Viene verificato l'inserimento della descrizione del corso di laurea.	<i>Superato</i>
TV0F4E	Viene verificato l'inserimento della tipologia del corso di laurea.	<i>Superato</i>
TV0F4G	Viene verificata la conferma dell'aggiunta di un corso di laurea.	<i>Superato</i>
TV0F4H	Viene verificato che l'aggiunta del corso di laurea sia andata a buon fine.	<i>Superato</i>
TV2F4.7	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente amministratore o università riempiano i campi dato relativi all'inserimento di un corso di laurea con informazioni errate o nulle.	<i>Superato</i>
TV0F5A	Viene verificata la visualizzazione della pagina per l'inserimento di un'attività didattica.	<i>Superato</i>
TV0F5B	Viene verificato l'inserimento del codice dell'attività didattica.	<i>Superato</i>
TV0F5C	Viene verificato l'inserimento del nome dell'attività didattica.	<i>Superato</i>
TV0F5D	Viene verificato l'inserimento della descrizione dell'attività didattica.	<i>Superato</i>
TV0F5E	Viene verificato l'inserimento del professore associato all'attività didattica.	<i>Superato</i>
TV0F5F	Viene verificato l'inserimento dei crediti dell'attività didattica.	<i>Superato</i>
TV0F5G	Viene verificato l'inserimento del periodo dell'attività didattica.	<i>Superato</i>





Codice test	Descrizione	Stato
TV0F5H	Viene verificata la conferma dell'aggiunta dell'attività didattica.	<i>Superato</i>
TV0F5I	Viene verificato che l'aggiunta dell'attività didattica sia andata a buon fine.	<i>Superato</i>
TV2F5.9	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente amministratore o università riempiano i campi dato relativi all'inserimento di un'attività didattica con informazioni errate o nulle.	<i>Superato</i>
TV0F6A	Viene verificata la visualizzazione della pagina per l'inserimento di un esame.	<i>Superato</i>
TV0F6B	Viene verificato l'inserimento del codice dell'esame.	<i>Superato</i>
TV0F6C	Viene verificato l'inserimento della descrizione dell'esame.	<i>Superato</i>
TV0F6D	Viene verificato l'inserimento dell'intervallo di prenotazione dell'esame.	<i>Superato</i>
TV0F6E	Viene verificato l'inserimento della data dell'esame.	<i>Superato</i>
TV0F6F	Viene verificato l'inserimento della tipologia dell'esame.	<i>Superato</i>
TV0F6G	Viene verificato l'inserimento del luogo dell'esame.	<i>Superato</i>
TV0F6H	Viene verificato l'inserimento del professore associato all'esame.	<i>Superato</i>
TV0F6I	Viene verificata la conferma dell'aggiunta dell'esame.	<i>Superato</i>
TV0F6L	Viene verificato che l'aggiunta dell'esame sia andata a buon fine.	<i>Superato</i>
TV2F6.9	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente amministratore o università riempiano i campi dato relativi all'inserimento di un'esame con informazioni errate o nulle.	<i>Superato</i>
TV2F11A	Viene verificata la visualizzazione della pagina relativa alla cancellazione di un anno accademico.	<i>Superato</i>
TV2F11B	Viene verificata l'effettiva eliminazione dell'anno accademico qualora l'utente amministratore o l'utente università abbia cliccato sul pulsante di eliminazione.	<i>Superato</i>
TV2F12A	Viene verificata la visualizzazione della pagina relativa alla cancellazione di un corso di laurea.	<i>Superato</i>
TV2F12B	Viene verificata l'effettiva eliminazione del corso di laurea qualora l'utente amministratore o l'utente università abbia cliccato sul pulsante di eliminazione.	<i>Superato</i>
TV2F13A	Viene verificata la visualizzazione della pagina relativa alla cancellazione di un'attività didattica.	<i>Superato</i>



Codice test	Descrizione	Stato
TV2F13B	Viene verificata l'effettiva eliminazione dell'attività didattica qualora l'utente amministratore o l'utente università abbia cliccato sul pulsante di eliminazione.	<i>Superato</i>
TV2F14A	Viene verificata la visualizzazione della pagina relativa alla cancellazione di un'esame.	<i>Superato</i>
TV2F14B	Viene verificata l'effettiva eliminazione dell'esame qualora l'utente amministratore o l'utente università abbia cliccato sul pulsante di eliminazione.	<i>Superato</i>
TV2F16A	Viene verificato che l'utente studente sia correttamente autenticato come studente.	<i>Superato</i>
TV2F16B	Viene verificata la visualizzazione della pagina della lista di esami associati allo studente.	<i>Superato</i>
TV2F16C	Viene verificata la conferma dell'iscrizione all'esame da parte dello studente.	<i>Superato</i>
TV2F16D	Viene verificato che l'iscrizione all'esame da parte dello studente sia andata a buon fine.	<i>Superato</i>
TV0F17A	Viene verificato che l'utente professore sia correttamente autenticato come professore.	<i>Superato</i>
TV0F17B	Viene verificata la visualizzazione della pagina della lista di esami associati al professore.	<i>Superato</i>
TV0F18	Viene verificata la visualizzazione della pagina della lista di studenti iscritti ad un esame associato al professore.	<i>Superato</i>
TV0F23A	Viene verificato l'inserimento di un voto ad uno studente da parte di un professore.	<i>Superato</i>
TV0F23B	Viene verificato che l'inserimento di un voto ad uno studente da parte di un professore sia andato a buon fine.	<i>Superato</i>
TV2F23.3	Viene verificata la visualizzazione di un messaggio d'errore qualora un professore inserisca un voto in un formato errato.	<i>Superato</i>
TV0F24A	Viene verificata la visualizzazione della pagina relativa all'aggiunta di un nuovo utente da parte di un utente amministratore o di un utente università.	<i>Superato</i>
TV0F24B	Viene verificato l'inserimento del codice fiscale dell'utente che l'utente amministratore o l'utente università vuole inserire.	<i>Superato</i>
TV0F24C	Viene verificato l'inserimento del codice univoco dell'utente che l'utente amministratore o l'utente università vuole inserire.	<i>Superato</i>
TV0F24D	Viene verificata la selezione della tipologia dell'utente che l'utente amministratore o l'utente università vuole inserire.	<i>Superato</i>
TV0F24E	Viene verificata la selezione dell'anno accademico relativo all'utente che l'utente amministratore o l'utente università vuole inserire.	<i>Superato</i>

Codice test	Descrizione	Stato
TV0F24F	Viene verificata la selezione del corso di laurea relativa all'utente che l'utente amministratore o l'utente università vuole inserire.	<i>Superato</i>
TV0F24G	Viene verificata la conferma dell'aggiunta di un nuovo utente da parte di un utente amministratore o di un utente università.	<i>Superato</i>
TV0F24H	Viene verificato che l'aggiunta di un utente da parte di un utente amministratore o di un utente università sia andata a buon fine.	<i>Superato</i>
TV2F24.6	Viene verificata la visualizzazione di un messaggio d'errore qualora l'utente amministratore o università riempiano i campi dato relativi all'aggiunta di un nuovo utente con informazioni errate o nulle.	<i>Superato</i>
TV0F25A	Viene verificata l'eliminazione di un utente studente da parte dell'utente amministratore o dell'utente università.	<i>Superato</i>
TV0F25B	Viene verificata l'eliminazione di un utente professore da parte dell'utente amministratore o dell'utente università.	<i>Superato</i>
TV0F25C	Viene verificata l'eliminazione di un utente amministratore da parte dell'utente università.	<i>Superato</i>
TV0F26A	Viene verificata la visualizzazione della pagina con la lista di esami a cui lo studente è registrato.	<i>Superato</i>
TV0F26B	Viene verificata la visualizzazione della pagina del libretto universitario da parte dello studente.	<i>Superato</i>

Tabella 2: Test di validazione

### B.1.1 Tracciamento Test di validazione - Requisiti

Codice test	Codice requisito
TV0F1A	R0F1
TV0F1B	R0F1
TV0F1C	R0F1
TV0F1D	R0F1
TV0F1E	R0F1
TV0F1F	R0F1
TV0F1G	R0F1
TV0F1H	R0F1
TV0F1I	R0F1
TV0F1L	R0F1
TV2F1.8	R2F1.8
TV0F2A	R0F2
TV0F2B	R0F2
TV0F3A	R0F3
TV0F3B	R0F3
TV0F3C	R0F3



Codice test	Codice requisito
TV0F3D	R0F3
TV0F3E	R0F3
TV0F3F	R0F3
TV0F3G	R0F3
TV2F3.5	R2F3.5
TV0F4A	R0F4
TV0F4B	R0F4
TV0F4C	R0F4
TV0F4D	R0F4
TV0F4E	R0F4
TV0F4F	R0F4
TV0F4G	R0F4
TV0F4H	R0F4
TV2F4.7	R2F4.7
TV0F5A	R0F5
TV0F5B	R0F5
TV0F5C	R0F5
TV0F5D	R0F5
TV0F5E	R0F5
TV0F5F	R0F5
TV0F5G	R0F5
TV0F5H	R0F5
TV0F5I	R0F5
TV2F5.9	R2F5.9
TV0F6A	R0F6
TV0F6B	R0F6
TV0F6C	R0F6
TV0F6D	R0F6
TV0F6E	R0F6
TV0F6F	R0F6
TV0F6G	R0F6
TV0F6H	R0F6
TV0F6I	R0F6
TV0F6L	R0F6
TV2F6.9	R2F6.9
TV2F11A	R2F11
TV2F11B	R2F11
TV2F12A	R2F12
TV2F12B	R2F12
TV2F13A	R2F13
TV2F13B	R2F13
TV2F14A	R2F14
TV2F14B	R2F14
TV2F16A	R2F16
TV2F16B	R2F16
TV2F16C	R2F16
TV2F16D	R2F16
TV0F17A	R0F17
TV0F17B	R0F17
TV0F18	R0F18
TV0F23A	R0F23
TV0F23B	R0F23



Codice test	Codice requisito
TV2F23.3	R2F23.3
TV0F24A	R0F24
TV0F24B	R0F24
TV0F24C	R0F24
TV0F24D	R0F24
TV0F24E	R0F24
TV0F24F	R0F24
TV0F24G	R0F24
TV0F24H	R0F24
TV2F24.6	R2F24.6
TV0F25A	R0F25
TV0F25B	R0F25
TV0F25C	R0F25
TV0F26A	R0F26
TV0F26B	R0F26

Tabella 3: Tracciamento test di validazione - requisiti



## B.2 Test di sistema

Questi test servono per verificare il comportamento dinamico complessivo dell'intero sistema in riferimento ai requisiti dichiarati nel documento *AnalisiDeiRequisiti\_v4.0.0*, come attività di controllo interna svolta dal fornitore.

Codice test	Descrizione	Stato
TS0F1	Viene verificato che il sistema permetta la creazione di un account permettendo di usufruire delle funzionalità del sistema, limitate dal tipo di account creato.	<i>Superato</i>
TS0F2	Viene verificato che il sistema permetta ad ogni utente correttamente registrato di autenticarsi automaticamente.	<i>Superato</i>
TS0F3	Viene verificato che il sistema permetta ad ogni utente amministratore o università di aggiungere un nuovo anno accademico al sistema.	<i>Superato</i>
TS0F4	Viene verificato che il sistema permetta ad ogni utente amministratore o università di aggiungere un nuovo corso di laurea al sistema.	<i>Superato</i>
TS0F5	Viene verificato che il sistema permetta ad ogni utente amministratore o università di aggiungere una nuova attività didattica al sistema.	<i>Superato</i>
TS0F6	Viene verificato che il sistema permetta ad ogni utente amministratore o università di aggiungere un nuovo esame al sistema.	<i>Superato</i>
TS2F11	Viene verificato che il sistema permetta ad ogni utente amministratore o università di eliminare un anno accademico già presente nel sistema.	<i>Superato</i>
TS2F12	Viene verificato che il sistema permetta ad ogni utente amministratore o università di eliminare un corso di laurea già presente nel sistema.	<i>Superato</i>
TS2F13	Viene verificato che il sistema permetta ad ogni utente amministratore o università di eliminare un'attività didattica già presente nel sistema.	<i>Superato</i>
TS2F14	Viene verificato che il sistema permetta ad ogni utente amministratore o università di eliminare un esame già presente nel sistema.	<i>Superato</i>
TS2F15	Viene verificato che il sistema permetta ad ogni utente amministratore o università di inserire un nuovo professore associato ad un'attività didattica.	<i>Superato</i>
TS2F16	Viene verificato che il sistema permetta ad ogni utente studente di iscriversi ad un esame.	<i>Superato</i>
TS0F17	Viene verificato che il sistema permetta ad ogni utente professore di visualizzare la lista degli esami a lui associati.	<i>Superato</i>



TS0F18	Viene verificato che il sistema permetta ad ogni utente professore di visualizzare la lista degli studenti iscritti ad uno degli esami a lui associati.	<i>Superato</i>
TS0F19	Viene verificato che il sistema visualizzi per ogni operazione il costo associato.	<i>Superato</i>
TS0F23	Viene verificato che il sistema permetta ad ogni utente professore di registrare il voto relativo ad un esame di uno studente.	<i>Superato</i>
TS0F24	Viene verificato che il sistema permetta ad ogni utente amministratore o università di creare un nuovo utente nel sistema.	<i>Superato</i>
TS0F25	Viene verificato che il sistema permetta ad ogni utente amministratore o università di rimuovere un utente dal sistema.	<i>Superato</i>
TS0F26	Viene verificato che il sistema permetta ad ogni utente studente di visualizzare il proprio libretto.	<i>Superato</i>
TS	Viene verificato che il sistema offra correttamente tutte le sue funzionalità con la versione 6.0.286 o superiore di Google Chrome, utilizzando metamask versione 4.6 o superiore.	<i>Superato</i>
TS	Viene verificato che il sistema offra correttamente tutte le sue funzionalità con la versione 50 o superiore di Mozilla Firefox, utilizzando metamask versione 4.5 o superiore.	<i>Superato</i>
TS	Viene verificato che il sistema offra correttamente tutte le sue funzionalità con la versione 52 o superiore di Opera, utilizzando metamask versione 3.13.4 o superiore.	<i>Superato</i>

Tabella 4: Test di sistema



### B.2.1 Tracciamento Test di sistema - Requisiti

Codice test	Codice requisito
TS0F1	R0F1
TS0F2	R0F2
TS0F3	R0F3
TS0F4	R0F4
TS0F5	R0F5
TS0F6	R0F6
TS2F11	R2F11
TS2F12	R2F12
TS2F13	R2F13
TS2F14	R2F14
TS2F15	R2F15
TS2F16	R2F16
TS0F17	R0F17
TS0F18	R0F18
TS0F19	R0F19
TS0F23	R0F23
TS0F24	R0F24
TS0F25	R0F25
TS0F26	R0F26
TS0F27	R0F27
TS0F28	R0F28

Tabella 5: Tracciamento test di sistema - requisiti





### B.3 Test di integrazione

Questi test verificano il corretto comportamento di ogni singola componente e le relazioni con il resto del sistema.

Codice test	Descrizione	Stato
TI1	Viene verificato che l'applicazione Web gestisca correttamente il front end del prodotto e le sue interazioni con il back end.	<i>Superato</i>
TI2	Viene verificato che i contratti Solidity si integrino correttamente con la componente web3.	<i>Superato</i>
TI3	Viene verificato che le componenti <i>action</i> , <i>store</i> e <i>reducers</i> di Redux comunichino correttamente tra loro.	<i>Superato</i>
TI4	Viene verificato che le componenti visive di React si integrino correttamente con la gestione degli <i>state</i> di Redux.	<i>Superato</i>
TI5	Viene verificato che l'immissione dei dati nel front end corrisponda ad una loro effettiva scrittura nel database del back end.	<i>Superato</i>
TI6	Viene verificato che l'applicativo web mostri correttamente i dati prelevati dal back end.	<i>Superato</i>
TI7	Viene verificato che venga ritornata la lista dei corsi di laurea corretta	<i>Superato</i>
TI8	Viene verificato che venga ritornata la lista delle attività didattiche corretta	<i>Superato</i>
TI9	Viene verificato che venga ritornata la lista degli anni accademici corretta	<i>Superato</i>
TI10	Viene verificato che uno studente possa loggarsi premendo sul bottone di login solo dopo che questo si sia correttamente registrato	<i>Superato</i>
TI11	Viene verificato che se alcuni parametri non sono corretti, allora l'azione di fallimento sia controllata	<i>Superato</i>
TI12	Viene verificato che venga ritornata la lista di tutti gli esami presenti nell'anno accademico indicato	<i>Superato</i>
TI13	Viene verificato che venga ritornata la lista di tutti gli esami di un corso di laurea	<i>Superato</i>
TI14	Viene verificato che venga ritornata la lista di tutti gli studenti registrati	<i>Superato</i>
TI15	Viene verificato che venga ritornata la lista di tutti i professori inseriti nel sistema	<i>Superato</i>
TI16	Viene verificato che possa venir assegnato un esame presente nella lista degli esami ad un professore	<i>Superato</i>
TI17	Viene verificato che venga aggiunto un anno accademico	<i>Superato</i>
TI18	Viene verificato che venga aggiunta un'attività didattica	<i>Superato</i>
TI19	Viene verificato che venga effettuato correttamente il login secondo il ruolo dell'utente che accede al sistema	<i>Superato</i>

TI20	Viene verificato che venga lanciato un errore nel caso dell'inserimento di un voto irregolare di un esame da parte di un professore	<i>Superato</i>
TI21	Viene verificato che venga aggiunto un corso di laurea	<i>Superato</i>
TI22	Viene verificato che venga visualizzata la lista degli esami associati ad un professore	<i>Superato</i>
TI23	Viene verificato che venga visualizzata la lista degli esami a cui è registrato uno studente	<i>Superato</i>
TI24	Viene verificato che venga visualizzata la lista degli esami a cui può iscriversi uno studente	<i>Superato</i>
TI25	Viene verificato che venga visualizzata la lista degli esami associati ad uno studente	<i>Superato</i>
TI26	Viene verificato che uno studente possa iscriversi a un esame	<i>Superato</i>
TI27	Viene verificato che possa essere correttamente eliminato un anno accademico senza corsi di laurea al suo interno	<i>Superato</i>
TI28	Viene verificato che possa essere correttamente eliminato uno studente che non ha ancora eseguito il signup	<i>Superato</i>
TI29	Viene verificato che possa essere correttamente eliminato un corso di laurea senza che abbia attività didattiche al suo interno	<i>Superato</i>
TI30	Viene verificato che possa essere correttamente eliminata un'attività didattica che non abbia al suo interno alcun esame	<i>Superato</i>
TI31	Viene verificato che possa essere correttamente eliminato un amministratore	<i>Superato</i>
TI32	Viene verificato che possa essere correttamente eliminato un professore	<i>Superato</i>
TI33	Viene verificato che venga accettato un voto da parte di uno studente	<i>Superato</i>

Tabella 6: Test di integrazione

## B.4 Test di unità

Questi test servono per verificare il corretto funzionamento delle singole parti del sistema.

Codice test	Descrizione	Stato
TU1	Viene verificato che il metodo <i>LogoutUser()</i> faccia uscire l'utente dal sistema e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU2	Viene verificato che il metodo <i>OnInsertUserFormSubmit(string,string,int)</i> inserisca un utente nel sistema e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU3	Viene verificato che dopo aver aggiunto un utente al sistema, esso sia effettivamente salvato nella blockchain assieme al suo codice univoco e codice fiscale.	<i>Superato</i>
TU4	Viene verificato che, dopo che un utente ha eseguito la registrazione tramite il codice univoco fornito dall'università ed il suo codice fiscale, immettendo i propri dati personali, esso risulti effettivamente registrato con i propri dati salvati.	<i>Superato</i>
TU5	Viene verificato che l'inserimento di un nuovo anno accademico sia effettivamente salvato con i dati relativi specificati.	<i>Superato</i>
TU6	Viene verificato che l'inserimento di un nuovo corso di laurea sia effettivamente salvato con i dati relativi specificati.	<i>Superato</i>
TU7	Viene verificato che l'inserimento di un nuovo esame sia effettivamente salvato con i dati relativi specificati.	<i>Superato</i>
TU8	Viene verificato che l'associazione di un esame con un professore comporti effettivamente un collegamento tra le due parti.	<i>Superato</i>
TU9	Viene verificato che il metodo <i>RemoveAdmic(admin)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU10	Viene verificato che il metodo <i>RemoveStudent(student)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU11	Viene verificato che il metodo <i>ViewStudentList(exam)</i> visualizzi correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU12	Viene verificato che il metodo <i>LoginUser()</i> autentichi correttamente l'utente nel sistema e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU13	Viene verificato che il metodo <i>render()</i> del component <i>Help</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>

TU14	Viene verificato che il metodo <i>render()</i> del component <i>LoginButoon</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU15	Viene verificato che il metodo <i>render()</i> del component <i>LogoutButton</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU16	Viene verificato che il metodo <i>render()</i> del component <i>NavButton</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU17	Viene verificato che il metodo <i>render()</i> del component <i>Home</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU18	Viene verificato che il metodo <i>render()</i> del component <i>App</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU19	Viene verificato che il metodo <i>render()</i> del component <i>InsertUserForm</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU20	Viene verificato che il metodo <i>render()</i> del component <i>InsertUser</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU21	Viene verificato che il metodo <i>render()</i> del component <i>EmptyData</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU22	Viene verificato che il metodo <i>render()</i> del component <i>DeleteAdministrator</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU23	Viene verificato che il metodo <i>render()</i> del component <i>Administrators</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU24	Viene verificato che il metodo <i>render()</i> del component <i>LoadingData</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU25	Viene verificato che il metodo <i>render()</i> del component <i>LoadingUser</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU26	Viene verificato che il metodo <i>render()</i> del component <i>NotFound</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU27	Viene verificato che il metodo <i>render()</i> del component <i>ModifyAcademicYear</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU28	Viene verificato che il metodo <i>render()</i> del component <i>InsertAcademicYear</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU29	Viene verificato che il metodo <i>render()</i> del component <i>DeleteAcademicYear</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU30	Viene verificato che il metodo <i>render()</i> del component <i>AcademicYears</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU31	Viene verificato che il metodo <i>render()</i> del component <i>ModifyDegreeCourse</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>

TU32	Viene verificato che il metodo <i>render()</i> del component <i>InsertDegreeCourse</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU33	Viene verificato che il metodo <i>render()</i> del component <i>DeleteDegreeCourse</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU34	Viene verificato che il metodo <i>render()</i> del component <i>DegreeCourses</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU35	Viene verificato che il metodo <i>render()</i> del component <i>ModifyDidacticActivity</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU36	Viene verificato che il metodo <i>render()</i> del component <i>InsertDidacticActivity</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU37	Viene verificato che il metodo <i>render()</i> del component <i>DeleteDidacticActivity</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU38	Viene verificato che il metodo <i>render()</i> del component <i>DidacticActivities</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU39	Viene verificato che il metodo <i>render()</i> del component <i>InsertExam</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU40	Viene verificato che il metodo <i>render()</i> del component <i>DeleteProfessor</i> comporti la corretta renderizzazione del componente.	<i>Implementato</i>
TU41	Viene verificato che il metodo <i>render()</i> del component <i>Professors</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU42	Viene verificato che il metodo <i>render()</i> del component <i>DeleteStudent</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU43	Viene verificato che il metodo <i>render()</i> del component <i>Students</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU44	Viene verificato che il metodo <i>render()</i> del component <i>ExamList</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU45	Viene verificato che il metodo <i>render()</i> del component <i>ExamPage</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU46	Viene verificato che il metodo <i>render()</i> del component <i>ExamsProfessorList</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU47	Viene verificato che il metodo <i>render()</i> del component <i>RegisteredStudentsList</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU48	Viene verificato che il metodo <i>render()</i> del component <i>ExamsStudentList</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU49	Viene verificato che il metodo <i>render()</i> del component <i>SchoolRecords</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>

TU50	Viene verificato che il metodo <i>render()</i> del component <i>Profile</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU51	Viene verificato che il metodo <i>render()</i> del component <i>SubNavbar</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU52	Viene verificato che il metodo <i>render()</i> del component <i>SubNavBarButton</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU53	Viene verificato che il metodo <i>render()</i> del component <i>SignUp</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU54	Viene verificato che il metodo <i>render()</i> del component <i>SignUpForm</i> comporti la corretta renderizzazione del componente.	<i>Superato</i>
TU55	Viene verificato che il metodo <i>ReadAcademicData()</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU56	Viene verificato che il metodo <i>ReadDidacticActivities(int,int)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU57	Viene verificato che il metodo <i>ReadDegreeCourses(int)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU58	Viene verificato che il metodo <i>ReadExams(int,int,int)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU59	Viene verificato che il metodo <i>ReadProfessorList()</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU60	Viene verificato che il metodo <i>ReadAdminList()</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU61	Viene verificato che il metodo <i>ReadStudentsList()</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU62	Viene verificato che il metodo <i>ReadExam(int,int,int)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU63	Viene verificato che il metodo <i>ReadExamList(professor)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>

TU64	Viene verificato che il metodo <i>ReadExamList(student)</i> legga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU65	Viene verificato che il metodo <i>ModifyAcademicData()</i> modifichi correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU66	Viene verificato che il metodo <i>ModifyDidacticActivity(int,int,int)</i> modifichi correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU67	Viene verificato che il metodo <i>ModifyDegreeCourse(int,int)</i> modifichi correttamente le informazioni e faccia il dispatch dell'operazione.	<i>Superato</i>
TU68	Viene verificato che il metodo <i>ModifyExam(int,int,int,int)</i> modifichi correttamente le informazioni e faccia il dispatch dell'operazione.	<i>Superato</i>
TU69	Viene verificato che il metodo <i>OnSignUp-Form.Submit(string,string,string,string,int)</i> inserisca un utente nel sistema registrandone i dati nel sistema e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU70	Viene verificato che il metodo <i>AddAcademicYear(int)</i> aggiunga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU71	Viene verificato che il metodo <i>AddDidacticActivity(int,int,int)</i> aggiunga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU72	Viene verificato che il metodo <i>AddDegreeCourse(int,int)</i> aggiunga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU73	Viene verificato che il metodo <i>AddExam(int,int,int,int)</i> aggiunga correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU74	Viene verificato che il metodo <i>RemoveAcademicYear(int)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU75	Viene verificato che il metodo <i>RemoveDidacticActivity(int,int,int)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU76	Viene verificato che il metodo <i>RemoveDegreeCourse(int,int)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>

TU77	Viene verificato che il metodo <i>RemoveExam(int,int,int,int)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>
TU78	Viene verificato che il metodo <i>RemoveProfessor(admin)</i> rimuova correttamente le informazioni e faccia il dispatch dello stato dell'operazione.	<i>Superato</i>

Tabella 7: Test di sistema

#### B.4.1 Tracciamento Test di unità - Metodi

Codice test	Classe:metodo
TU1	<i>InsertUserFormAction : OnInsertUserFormSubmit(string, string, int)</i>
TU2	<i>LogoutButtonAction : LogoutUser()</i>
TU3	<i>Admin : addUser(bytes32_fiscalCode, bytes10_uniCode, uint8_userType)</i> <i>UserData : getUsersUniCode(bytes32_fiscalCode)</i> <i>UserData : getUsersUserType(bytes32_fiscalCode)</i>
TU4	<i>UserLogic : signUp(bytes32_fiscalCode, bytes10_uniCode, bytes32_hashData)</i> <i>UserData : getRegUsersUniCode(address_address)</i> <i>UserData : getRegUsersFiscalCode(address_address)</i> <i>UserData : getRegUsersUserType(address_address)</i> <i>UserData : getRegUsersBadgeNumber(address_address)</i> <i>UserData : getRegUsersHashData(address_address)</i>
TU5	<i>Admin : addNewDegree(bytes10_degreeUniCode, bytes4_year, bytes32_hashData)</i> <i>DegreeData : isDegree(bytes10_degreeUniCode)</i> <i>DegreeData : getDegreeCourses(bytes10_degreeUniCode)</i>
TU6	<i>Admin : addNewExam(bytes10_courseUniCode, bytes10_examUniCode, uint32_examTeacher, bytes32_examHashData)</i> <i>ExamData : examExist(bytes10_examUniCode)</i> <i>ExamData : getExamTeacher(bytes10_examUniCode)</i> <i>ExamData : getAllIdentifiers()</i>
TU7	<i>Teacher : myExams()</i>
TU8	<i>Teacher : registerResult(bytes10_examUniCode, uint32_studentBadgeNumber, bytes2_result)</i>
TU9	<i>DeleteAdministrator : RemoveAdmin(admin)</i>
TU10	<i>DeleteStudent : RemoveStudent(student)</i>
TU11	<i>RegisteredStudentListAction : ViewStudentListAction(exam)</i>
TU12	<i>LoginButtonAction : LoginUser()</i>
TU13	<i>Help : render</i>
TU14	<i>LoginButton : render</i>
TU15	<i>LogoutButton : render</i>
TU16	<i>NavButton : render</i>
TU17	<i>Home : render</i>
TU18	<i>App : render</i>
TU19	<i>InsertUserForm : render</i>
TU20	<i>InsertUser : render</i>
TU21	<i>EmptyData : render</i>
TU22	<i>DeleteAdministrator : render</i>
TU23	<i>Administrators : render</i>
TU24	<i>LoadingData : render</i>
TU25	<i>LoadingUser : render</i>
TU26	<i>NotFound : render</i>





Codice test	Classe:metodo
TU27	<i>ModifyAcademicYear : render</i>
TU28	<i>InsertAcademicYear : render</i>
TU29	<i>DeleteAcademicYear : render</i>
TU30	<i>AcademicYears : render</i>
TU31	<i>ModifyDegreeCourse : render</i>
TU32	<i>InsertDegreeCourse : render</i>
TU33	<i>DeleteDegreeCourse : render</i>
TU34	<i>DegreeCourses : render</i>
TU35	<i>ModifyDidacticActivity : render</i>
TU36	<i>InsertDidacticActivity : render</i>
TU37	<i>DeleteDidacticActivity : render</i>
TU38	<i>DidacticActivities : render</i>
TU39	<i>InsertExam : render</i>
TU40	<i>DeleteProfessor : render</i>
TU41	<i>Professors : render</i>
TU42	<i>DeleteStudent : render</i>
TU43	<i>Students : render</i>
TU44	<i>ExamList : render</i>
TU45	<i>ExamPage : render</i>
TU46	<i>ExamProfessorList : render</i>
TU47	<i>RegisteredStudentsList : render</i>
TU48	<i>ExamsStudentList : render</i>
TU49	<i>SchoolRecords : render</i>
TU50	<i>Profile : render</i>
TU51	<i>SubNavbar : render</i>
TU52	<i>SubNavButton : render</i>
TU53	<i>SignUp : render</i>
TU54	<i>SignUpForm : render</i>
TU55	<i>ViewAcademicYearAction : ReadAcademicData()</i>
TU56	<i>ViewDidacticActivitiesAction : ReadDidacticActivities(int, int)</i>
TU57	<i>ViewDegreeCoursesAction : ReadDegreeCourses(int)</i>
TU58	<i>ViewExamsAction : ReadExams(int, int, int)</i>
TU59	<i>ViewProfessor : ReadProfessorList()</i>
TU60	<i>ViewAdministrators : ReadAdminList()</i>
TU61	<i>ViewStudents : ReadStudentList()</i>
TU62	<i>ExamPageAction : ReadExam(int, int, int)</i>
TU63	<i>ExamProfessorListActions : ReadExamList(professor)</i>
TU64	<i>ViewExamsAction : ReadExamList(student)</i>
TU65	<i>ModifyAcademicYearsActions : ModifyAcademicData()</i>
TU66	<i>ModifyDidacticActivityAction : ModifyDidacticActivity(int, int, int)</i>
TU67	<i>ModifyDegreeCourseAction : ModifyDegreeCourse(int, int)</i>
TU68	<i>ModifyExamAction : ModifyExam(int, int, int, int)</i>
TU69	<i>InsertUserFormAction : OnSignUpFormSubmit(string, string, string, string, int)</i>
TU70	<i>InsertAcademicYearAction : AddAcademicYear(int)</i>
TU71	<i>InsertDidacticActivitiesAction : AddDidacticActivity(int, int, int)</i>
TU72	<i>InsertDegreeCourseAction : AddDegreeCourse(int, int)</i>
TU73	<i>InsertExamAction : AddExam(int, int, int, int)</i>
TU74	<i>DeleteAcademicYearAction : RemoveAcademicYear(int)</i>
TU75	<i>RemoveDidacticActivities : RemoveDidacticActivities(int, int, int)</i>
TU76	<i>DeleteDegreeCoursesAction : removeDegreeCourse(int, int)</i>
TU77	<i>DeleteExamAction : removeExam(int, int, int)</i>
TU78	<i>DeleteProfessor : RemoveProfessor(admin)</i>



Codice test	Classe:metodo
-------------	---------------

Tabella 8: Tracciamento test di unità - metodi

## C Resoconto delle attività di verifica

### C.1 Attività di analisi dei requisiti

Dopo aver redatto tutti i documenti presenti nella Revisione dei Requisiti, il team ha svolto le attività di verifica su di essi e sui processi analizzati. I documenti sono stati sottoposti al processo di analisi statica definito nel documento *NormeDiProgetto\_v4.0.0*. Prima è stata utilizzata la tecnica del Walkthrough, segnalando gli errori incontrati tramite una lettura approfondita in un'apposita lista presa in carico dal *Verificatore* per attuare la correzione del documento. In seguito la stessa lista è stata utilizzata per la tecnica dell'Inspection, che è servita ad individuare la presenza di nuovi errori utilizzando il confronto della lista di quelli commessi in precedenza. In seguito i documenti sono stati interamente verificati secondo le metriche descritte nell'Appendice ?? e sono stati riportati i risultati ottenuti.

### C.2 Attività di analisi dei requisiti in dettaglio

In questo periodo di attività, il team si è impegnato a colmare le proprie lacune tecnologiche necessarie allo svolgimento del progetto. Parallelamente, si mira a consolidare ed ampliare i requisiti richiesti dal sistema e a migliorare il documento di AnalisiDeiRequisiti v1.0.0 attuando le correzioni in base all'esito della Revisione dei Requisiti; vengono inoltre corretti e verificati anche gli altri documenti, secondo le modalità descritte in precedenza.

### C.3 Attività di prototipazione

Questo periodo è caratterizzato dalla realizzazione di un prototipo utilizzando le tecnologie necessarie, con lo scopo di comprendere pienamente il dominio tecnologico del progetto attraverso la realizzazione dei casi d'uso essenziali e ritenuti significativi per la buona riuscita del prodotto finale. Sono inoltre incrementati quasi tutti i documenti per i quali vengono dunque ricalcolate le metriche di qualità.

### C.4 Attività di prototipazione in dettaglio

In tale periodo viene incrementata e migliorata la  $PoC_G$  e vengono corretti i documenti, sui quali sono ancora calcolate le metriche di qualità.

### C.5 Attività di progettazione finale e codifica

Scopo del periodo è progettare dettagliatamente l'architettura del sistema, anche attraverso la creazione di diagrammi dei design pattern, delle classi e di sequenza; segue una lunga fase di codifica e l'incremento di tutti i documenti eccetto l'analisi dei requisiti, a cui si aggiungono i manuali utente e sviluppatore. Sono dunque calcolate tutte le metriche delle fasi precedenti, con l'aggiunta di quelle relative al codice prodotto.

### C.6 Verifica dei documenti

#### C.6.1 Errori ortografici

Durante l'ultima verifica, sono stati rilevati all'interno dei vari documenti alcuni errori ortografici, il cui numero è specificato nel seguente grafico:

Il grafico evidenzia che la il numero medio di errori ortografici, è rimasto sempre entro i limiti di accettabilità.

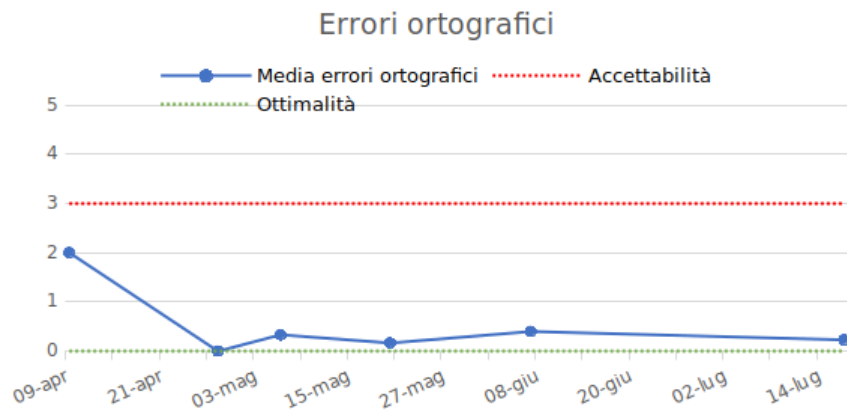


Figura 5: Media degli errori ortografici.

### C.6.2 Errori concettuali

Durante l'ultima verifica, sono stati rilevati all'interno dei vari documenti alcuni errori concettuali, il cui numero è specificato nel seguente grafico:

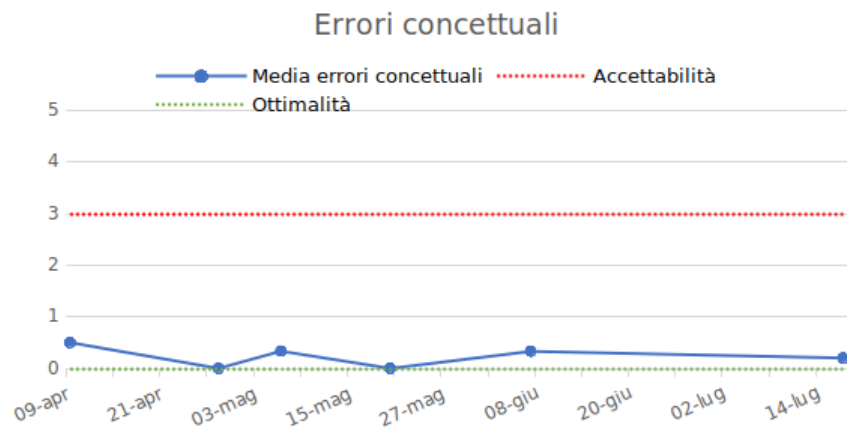


Figura 6: Media degli errori concettuali

Il grafico evidenzia che la il numero medio di errori concettuali è rimasto sempre entro i limiti di accettabilità.

### C.6.3 Errori di forma

Durante l'ultima verifica, sono stati rilevati all'interno dei vari documenti alcuni errori forma, il cui numero è specificato nel seguente grafico:

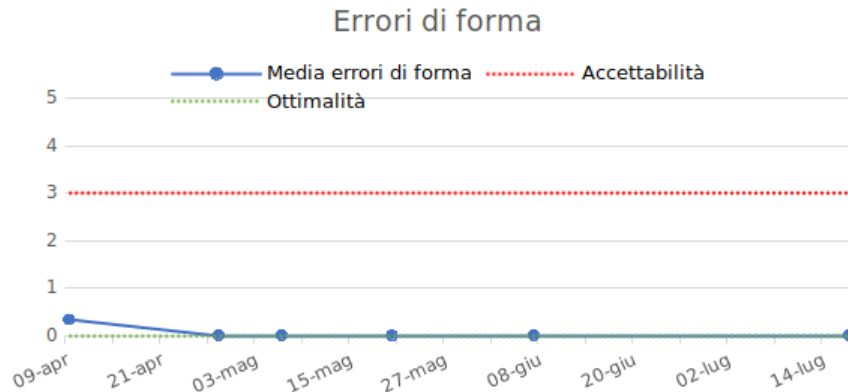


Figura 7: Media degli errori di forma.

Sempre entro i limiti di accettabilità e dopo un certo periodo si sono stabilizzati sulla soglia dell'ottimalità.

### C.6.4 Indice Gulpease

Tutti i documenti consegnati sono stati sottoposti al calcolo dell'Indice Gulpease per valutarne il grado di leggibilità, il quale è riportato nel seguente grafico:

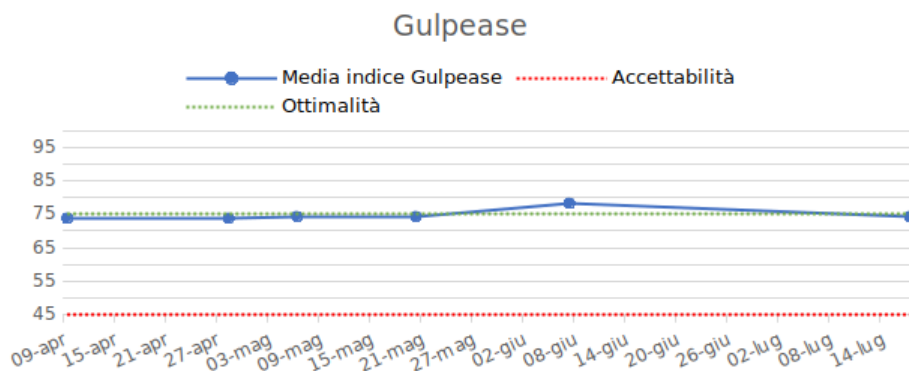


Figura 8: Media dell'indice di Gulpease; più il valore è alto, più ci si avvicina all'ottimalità.

Il grafico evidenzia che la il valore medio del calcolo dell'indice di gulpease è rimasto sempre entro i limiti di accettabilità, quasi raggiungendo l'ottimalità.

## C.7 Verifica dei processi

### C.7.1 Schedule Variance

Nel seguente grafico vengono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di stesura dei documenti, progettazione e codifica del prodotto software rispetto ai tempi prefissati nel *PianoDiProget-to\_v4.0.0*:

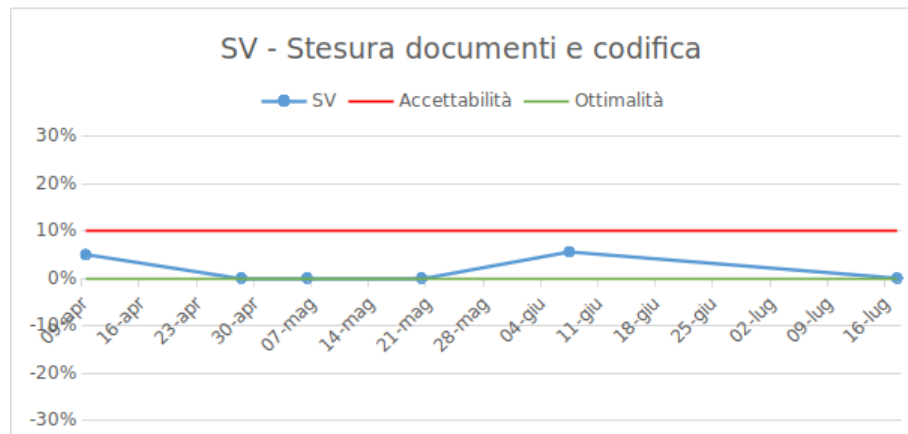


Figura 9: Schedule Variance sui processi di stesura dei documenti, progettazione e codifica.

Il grafico evidenzia come il team sia riuscito a portare a termine i propri compiti sempre in tempo, oscillando tra accettabilità e ottimalità.

### C.7.2 Cost Variance

Il calcolo della Cost Variance sui processi di documentazione e codifica ha portato il seguente risultato:

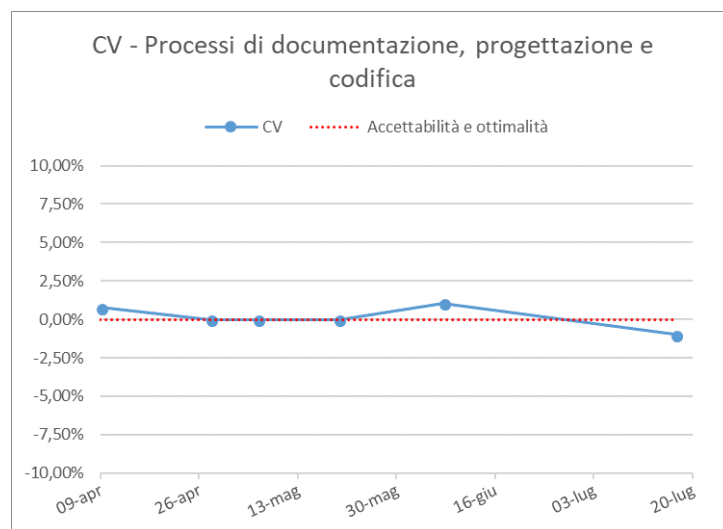


Figura 10: Cost Variance per il processo di documentazione e codifica.

Ha oscillato tra accettabilità e non accettabilità. Il non raggiungimento dell'accettabilità è stato comunque sempre compensato da un risparmio sul processo di verifica.

Di seguito vengono mostrati i risultati calcolati negli ultimi due periodi trascorsi, in cui si è iniziato ad utilizzare le metriche riguardanti la qualità del codice prodotto, al fine di evidenziare le differenze tra il codice della PoC e quello attuale.

### C.7.3 Schedule Variance - processo di verifica

Nel seguente grafico vengono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di verifica dei documenti e del prodotto software rispetto ai tempi prefissati nel *PianoDiProgetto\_v4.0.0*:

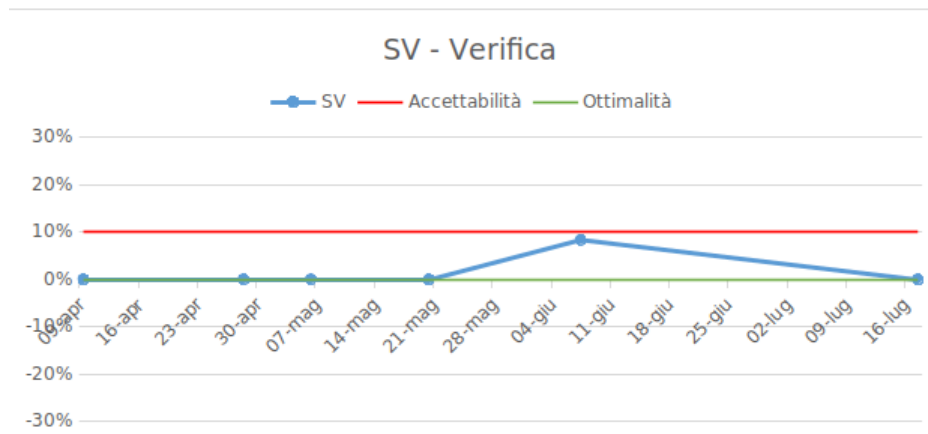


Figura 11: Schedule Variance per il processo di verifica.

Il grafico evidenzia che la schedule variance calcolata sul processo di verifica della documentazione è rimasta sempre entro i limiti di accettabilità.

### C.7.4 Cost Variance - verifica

Il calcolo della Cost Variance sul processo di verifica ha portato il seguente risultato:

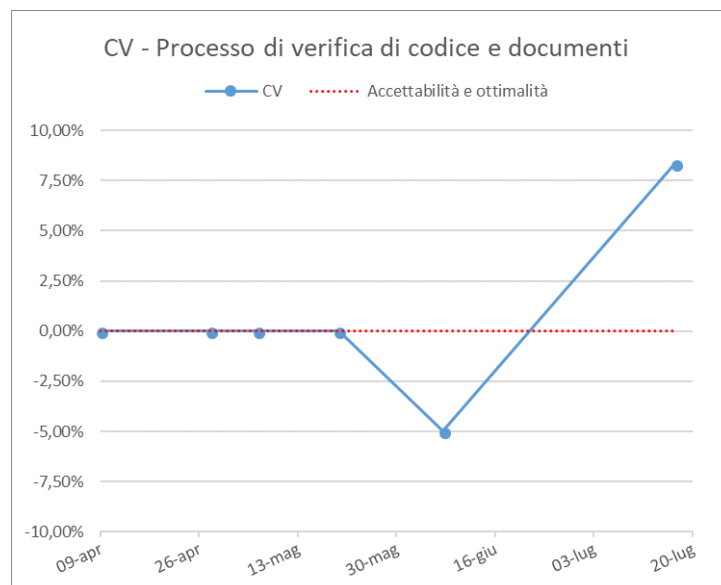


Figura 12: Cost Variance per il processo di verifica.



Nel periodo di RQ è stato presente un valore negativo causato dall'inesperienza in ambito test, che però si è riflesso in un notevole risparmio in RA.



### C.7.5 Rischi non preventivati

Tale metrica è stata aggiunta dopo le prime fasi del progetto, dato che il team si è reso conto di aver bisogno di una maggior attenzione ai rischi, per essere sicuro di gestirli correttamente nel momento della loro manifestazione.

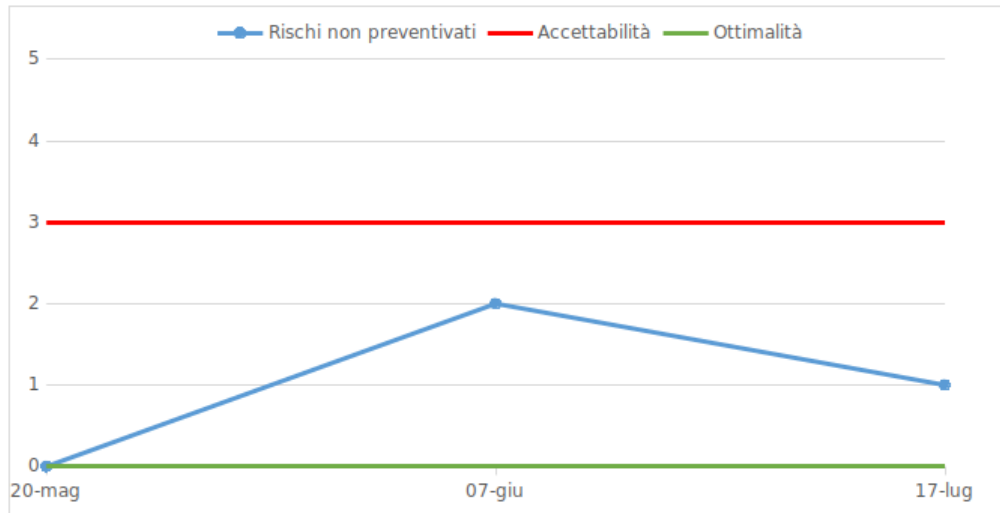


Figura 13: Rischi non preventivati negli ultimi due periodi.

Il grafico evidenzia che nello scorso periodo sono stati riscontrati alcuni rischi non preventivati, la cui causa è la scarsa conoscenza pregressa da parte del team delle tecnologie utilizzate, dato che erano state utilizzate più nel dettaglio. I rischi riscontrati in questo periodo si sono invece affievoliti grazie all'esperienza fatta precedentemente.

### C.7.6 Complessità ciclomatica

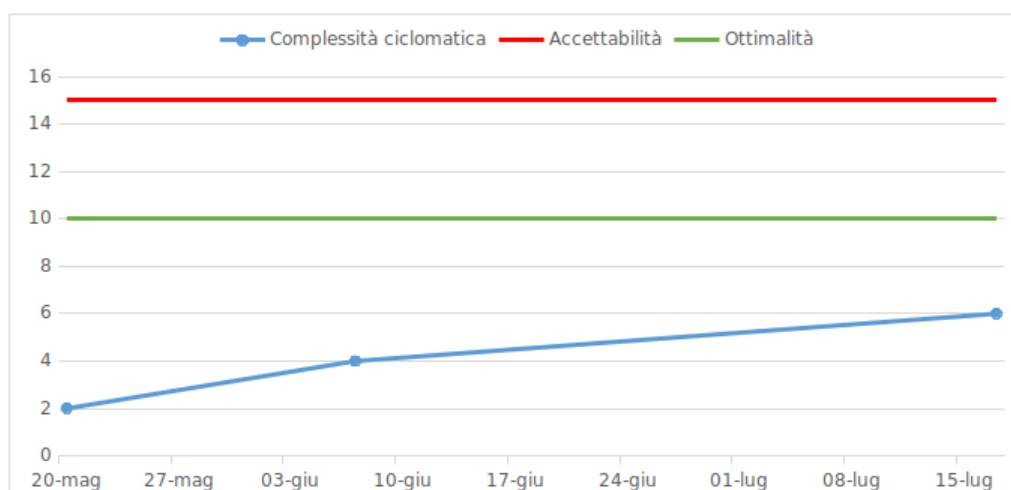


Figura 14: Complessità ciclomatica.

Il grafico evidenzia come la maggior complessità delle funzionalità offerte dal prodotto durante lo sviluppo si riflette in una maggior complessità del codice stesso.

### C.7.7 SFIN - Structural fan in

Essendo questo risultato una media complessiva calcolata sulle classi presenti nel codice dell'intero progetto, si è ritenuto riportarla singolarmente, senza calcolare anche la SFOUT - Structural fan out, per evitare la ripetizione di un risultato identico.

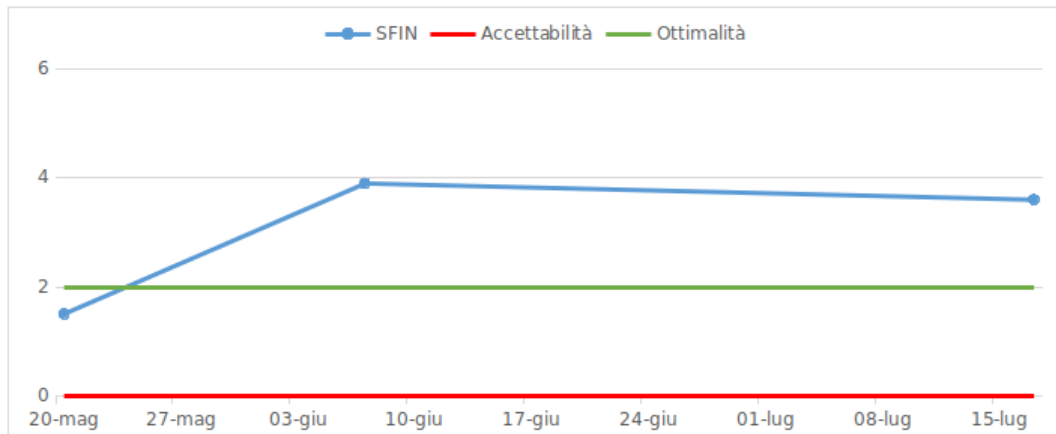


Figura 15: Structural fan in, più il valore è alto, meglio è.

Nonostante inizialmente si è rimasti sotto l'ottimalità, con il crescere del prodotto si è arrivati a superare di gran lunga la soglia di ottimalità, il che indica un alto riuso del codice.

### C.7.8 CxSLOC - Commenti per linee di codice

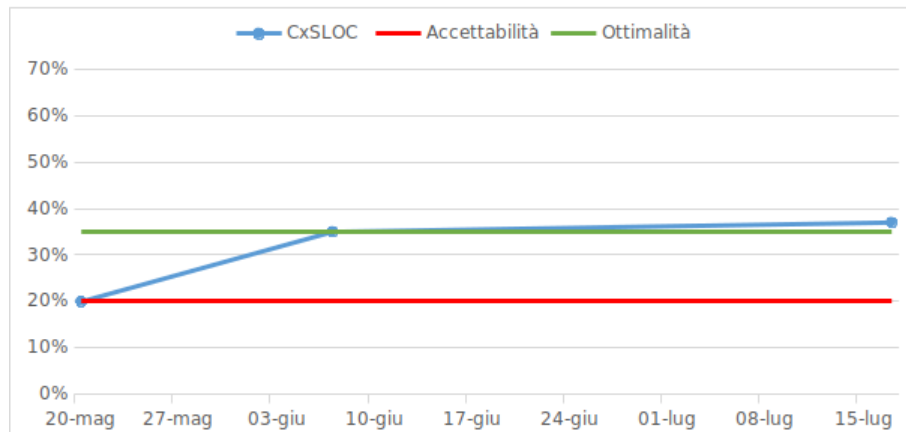


Figura 16: Commenti per linee di codice, un valore alto è positivo.

Il grafico evidenzia come il team si sia impegnato a fornire un codice quanto più capibile possibile, aumentando di molto la percentuale di commenti rispetto alla versione iniziale del prodotto.

### C.7.9 Parametri per metodo

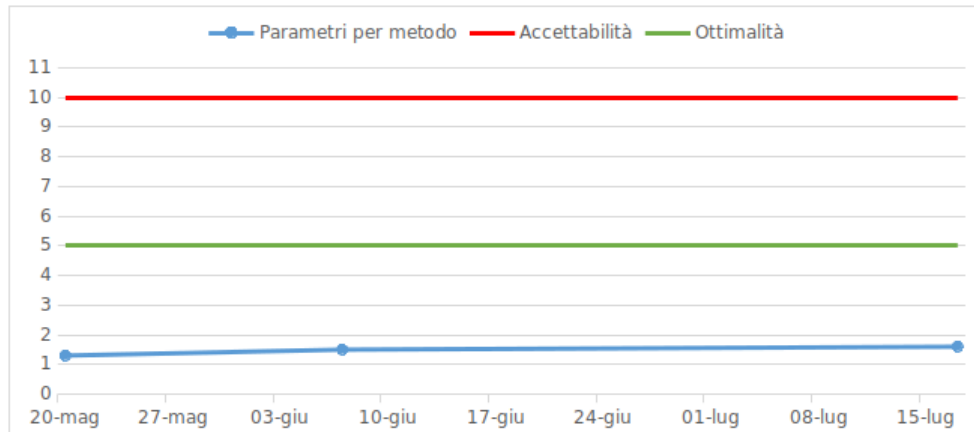


Figura 17: Numero di parametri per metodo, un numero basso indica meno complessità.

Mediamente, i metodi scritti dal team sono risultati poco complessi sotto questo punto di vista, con un numero di parametri ben oltre la soglia di ottimalità.

### C.7.10 Linee di codice per metodo

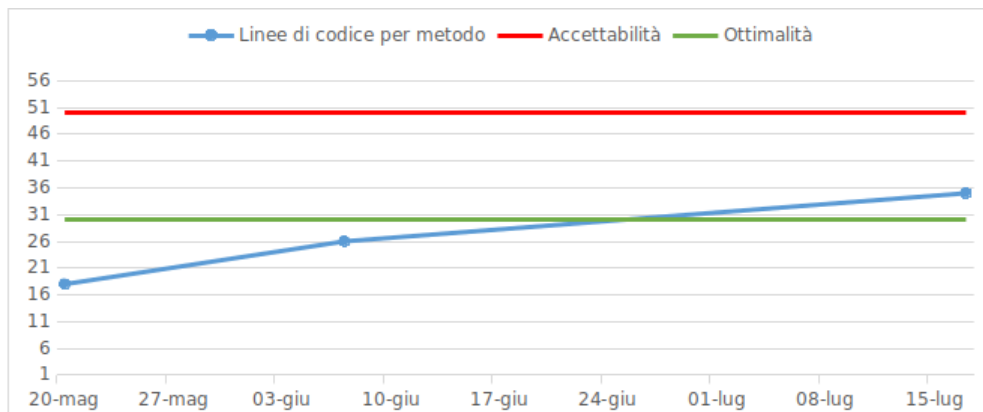


Figura 18: Linee di codice per metodo, un numero basso indica meno complessità.

Mediamente, i metodi scritti dal team sono risultati non troppo complessi sotto questo punto di vista, con un numero di righe accettabili.

### C.7.11 ROF - Requisiti obbligatori soddisfatti

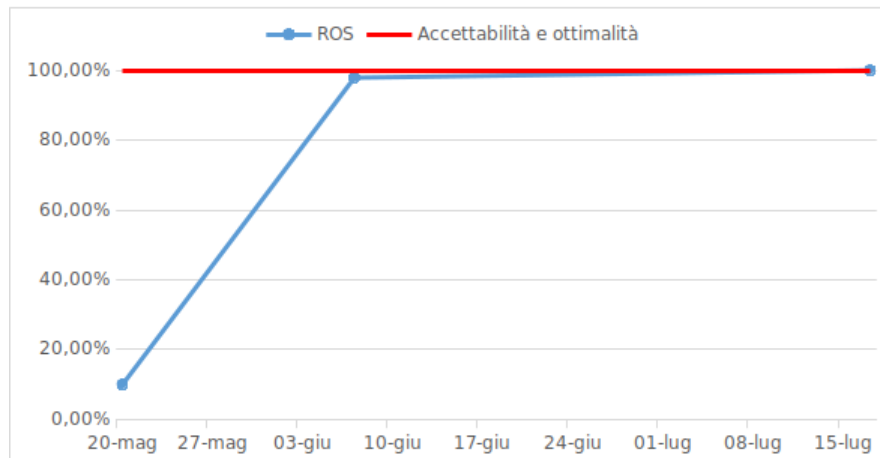


Figura 19: Requisiti obbligatori soddisfatti.

Il grafico mostra come nonostante la prima bozza del prodotto coprisse una percentuale esigua di requisiti, il team è riuscito a soddisfarli tutti con la versione attuale del prodotto.

### C.7.12 ROFS - Requisiti obbligatori e facoltativi soddisfatti

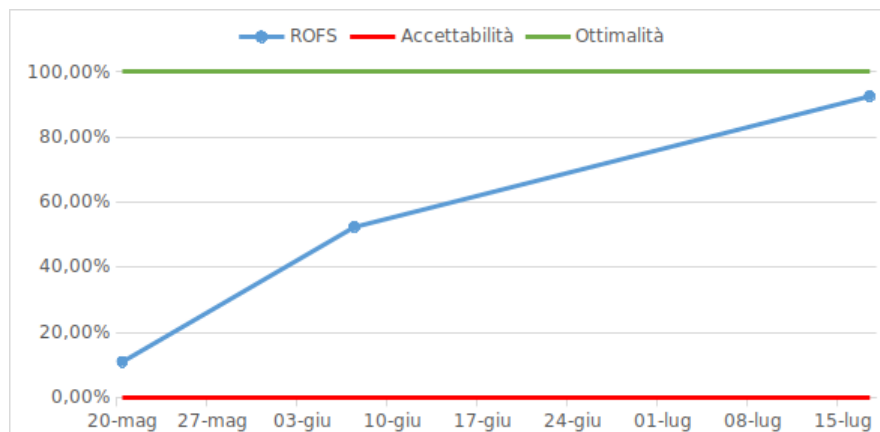


Figura 20: Requisiti obbligatori e facoltativi soddisfatti.

Il grafico mostra come anche una buona parte dei requisiti facoltativi sono stati soddisfatti.

### C.7.13 Successo dei test

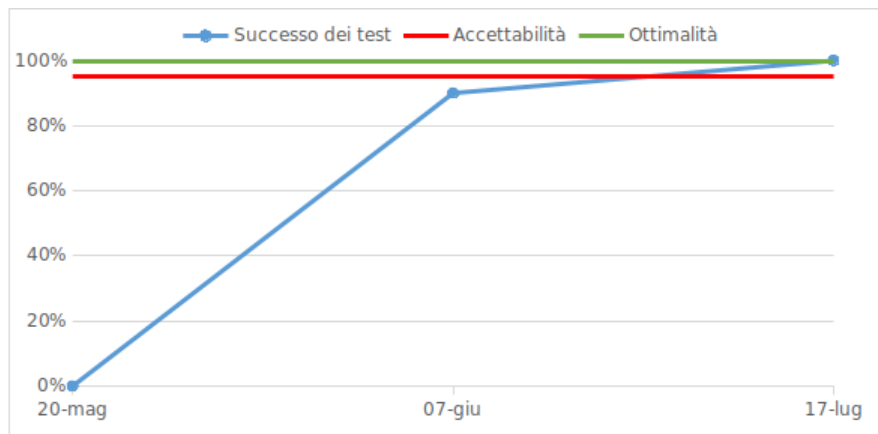


Figura 21: Percentuale di successo dei test.

Il grafico mostra che la totalità dei test hanno esito positivo, è stata raggiunta l'ottimalità.

### C.7.14 Copertura del codice

Valore	Accettabilità	Ottimalità
81	70%	90%

Come si può vedere è stata raggiunta l'accettabilità.

### C.7.15 Copertura dei branch

Valore	Accettabilità	Ottimalità
92	70%	90%

Come si può vedere è stata raggiunta l'ottimalità.