

DIGITAL COOKIES

Piano di Qualifica

Gruppo DigitalCookies – Progetto SWEDesigner

digitalcookies.group@gmail.com

Informazioni sul documento

Versione	3.0.0
Redazione	Carlo Sindico, Alessia Bragagnolo, Christian Cabrera, Alberto Giudice, Davide Albertini, Saverio Follador
Verifica	Alessia Bragagnolo, Alberto Rossetti
Approvazione	Christian Cabrera
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo DigitalCookies

Descrizione

Questo documento descrive le operazioni di verifica e validazione seguite dal gruppo DigitalCookies relativi al progetto SWEDesigner.

Registro delle modifiche

Versione	Data	Collaboratori	Ruolo	Descrizione
3.0.0	20-06-2017	Christian Cabrera	Responsabile	Approvazione del documento
2.4.0	20-06-2017	Alessia Bragagnolo	Verificatore	Verifica del documento
2.3.1	19-06-2017	Alberto Rossetti	Verificatore	Incremento appendice E
2.3.0	13-06-2017	Alessia Bragagnolo	Verificatore	Verifica del documento
2.2.1	12-06-2017	Christian Cabrera	Verificatore	Incremento appendice E
2.2.0	06-06-2017	Alessia Bragagnolo	Verificatore	Verifica del documento
2.1.1	05-06-2017	Alberto Rossetti	Verificatore	Incremento appendice E
2.1.0	26-05-2017	Alberto Rossetti	Verificatore	Verifica del documento
2.0.3	25-05-2017	Alberto Giudice	Verificatore	Incremento della sezione 3.1
2.0.2	25-05-2017	Alessia Bragagnolo	Verificatore	Modifica appendice E con l'uso di serie storiche
2.0.1	23-05-2017	Christian Cabrera	Progettista	Stesura test di unità
2.0.0	08-05-2017	Alessia Bragagnolo	Responsabile	Approvazione
1.2.0	08-05-2017	Christian Cabrera	Verificatore	Verifica del documento
1.1.6	05-05-2017	Davide Albertini	Verificatore	Modifica della sezione D.3 per via di cambiamenti alla <i>Specifica Tecnica</i>
1.1.5	04-05-2017	Saverio Follador	Verificatore	Incremento appendice E

1.1.4	03-05-2017	Davide Albertini	Progettista	Stesura test di integrazione
1.1.3	03-05-2017	Alberto Giudice	Progettista	Stesura test di sistema
1.1.2	02-05-2017	Alberto Giudice	Progettista	Stesura test di validazione
1.1.1	02-05-2017	Alberto Giudice	Progettista	Creazione appendice relativa alla specifica dei test
1.1.0	25-04-2017	Carlo Sindico	Verificatore	Verifica del documento
1.0.7	24-04-2017	Alberto Giudice	Verificatore	Riorganizzazione e incremento appendice E
1.0.6	24-04-2017	Alberto Giudice	Verificatore	Stesura sezione Esiti delle revisioni
1.0.5	20-04-2017	Christian Cabrera	Verificatore	Ampliamento appendice A per maggiore completezza
1.0.4	20-04-2017	Christian Cabrera	Verificatore	Ampliamento appendice sugli standard adottati, ora divisa in tre appendici distinte
1.0.3	19-04-2017	Alberto Giudice	Verificatore	Stesura obiettivi di qualità
1.0.2	18-04-2017	Alberto Giudice	Verificatore	Rimozione sezione relativa alle metriche, spostate nelle <i>Norme di Progetto</i>

1.0.1	18-04-2017	Alberto Giudice	Verificatore	Ristrutturazione sezione 2 con annessione di contenuti della sezione Gestione amministrativa della revisione (rimossa)
1.0.0	23-03-2017	Alberto Rossetti	Responsabile	Approvazione
0.1.0	22-03-2017	Christian Cabrera	Verificatore	Verifica del documento
0.0.7	21-03-2017	Alberto Giudice	Analista	Stesura sezione Resoconto delle attività di verifica
0.0.6	20-03-2017	Alessia Bragagnolo	Analista	Stesura sezione Pianificazione ed esecuzione del collaudo e appendice B
0.0.5	16-03-2017	Alberto Giudice	Verificatore	Stesura sezione Gestione amministrativa della revisione
0.0.4	14-03-2017	Carlo Sindico	Verificatore	Stesura sezione Misure e Metriche
0.0.3	09-03-2017	Alberto Giudice	Verificatore	Stesura sezione Visione generale della strategia di verifica
0.0.2	08-03-2017	Carlo Sindico	Verificatore	Stesura sezione Introduzione
0.0.1	08-03-2017	Alberto Rossetti	Responsabile	Creazione del template

Indice

1 Introduzione	7
1.1 Scopo del documento	7
1.2 Scopo del prodotto	7
1.3 Ambiguità	8
1.4 Riferimenti	8
1.4.1 Normativi	8
1.4.2 Informativi	8
2 Visione generale	10
2.1 Standard adottati	10
2.2 Risorse	10
2.3 Gestione delle attività di verifica	11
2.3.1 Responsabilità	11
2.3.2 Classificazione delle anomalie	11
2.4 Organizzazione	11
3 Obiettivi di qualità	13
3.1 Processo	13
3.1.1 Sviluppo	13
3.1.2 Documentazione	14
3.1.3 Verifica	14
3.1.4 Validazione	14
3.1.5 Gestione	15
3.2 Prodotto	15
3.2.1 Funzionalità	15
3.2.2 Affidabilità	15
3.2.3 Efficienza	16
3.2.4 Usabilità	16
3.2.5 Manutenibilità	17
3.2.6 Portabilità	17
A Standard ISO/IEC 15504	18
B Standard ISO/IEC 9126	22
C PDCA	23
D Specifica dei Test	25
D.1 Test di Validazione	25
D.1.1 Tracciamento test di validazione-requisiti	40
D.2 Test di Sistema	43
D.2.1 Tracciamento test di sistema-requisiti	50

D.3	Test di Integrazione	53
D.3.1	Tracciamento test di integrazione-componenti	57
D.4	Test di Unità	57
D.4.1	Tracciamento test di unità-metodi	64
E	Resoconto delle attività di verifica	73
E.1	Processi	73
E.1.1	Sviluppo	73
E.1.1.1	MPC1 - Requirement Stability Index	73
E.1.1.2	MPC2 - Instability	74
E.1.1.3	MPC3 - Violazioni dello stile di codifica	75
E.1.1.4	Livello SPICE	75
E.1.2	Documentazione	76
E.1.2.1	MPC4 - Errori frequenti nella documentazione	76
E.1.2.2	Livello SPICE	76
E.1.3	Verifica	77
E.1.3.1	MPC5 - Test di unità eseguiti	77
E.1.3.2	MPC6 - Test di integrazione eseguiti	78
E.1.3.3	Livello SPICE	78
E.1.4	Gestione	79
E.1.4.1	MPC9 - Schedule Variance	79
E.1.4.2	MPC10 - Budget Variance	80
E.1.4.3	Livello SPICE	80
E.2	Prodotti	81
E.2.1	Funzionalità	81
E.2.1.1	MPS1 - Copertura requisiti obbligatori	81
E.2.1.2	MPS2 - Copertura requisiti desiderabili	82
E.2.2	Affidabilità	83
E.2.2.1	MPS10 - Copertura del codice	83
E.2.3	Efficienza	84
E.2.3.1	MPS9 - Nested Block Depth	84
E.2.4	Usabilità	85
E.2.4.1	MPD1 - Indice Gulpease	85
E.2.5	Manutenibilità	86
E.2.5.1	MPS3 - Average Cyclomatic Complexity	86
E.2.5.2	MPS4 - Coupling Between Object Classes	87
E.2.5.3	MPS5 - Weighted Method Complexity	88
E.2.5.4	MPS6 - Class Size Attributes	89
E.2.5.5	MPS7 - Number of Parameters	90
E.2.5.6	MPS8 - Non-comment Lines of Code	91
F	Esiti delle revisioni	92
F.1	Revisione dei Requisiti	92
F.2	Revisione di Progettazione	93

Elenco delle tabelle

2	Range delle metriche per il processo di sviluppo	13
3	Range delle metriche per il processo di documentazione	14
4	Range delle metriche per il processo di verifica	14
5	Range delle metriche per il processo di validazione	14
6	Range delle metriche per il processo di gestione	15
7	Range delle metriche per la funzionalità dei prodotti	15
8	Range delle metriche per l'affidabilità dei prodotti	16
9	Range delle metriche per l'efficienza dei prodotti	16
10	Range delle metriche per l'usabilità dei prodotti	16
11	Range delle metriche per la manutenibilità dei prodotti	17
12	Test di Validazione	39
14	Test di Sistema	49
16	Test di Integrazione	56
18	Test di Unità	63

Elenco delle figure

1	Continuous quality improvement with PDCA	23
2	Test di validazione - stato corrente	40
3	Test di sistema - stato corrente	50
4	Test di integrazione - stato corrente	56
5	Test di unità - stato corrente	63
6	MPC1 - Serie storica	73
7	MPC2 - Serie storica	74
8	MPC3 - Serie storica	75
9	MPC4 - Serie storica	76
10	MPC5 - Serie storica	77
11	MPC6 - Serie storica	78
12	MPC9 - Serie storica	79
13	MPC10 - Serie storica	80
14	MPS1 - Serie storica	81
15	MPS2 - Serie storica	82
16	MPS10 - Serie storica	83
17	MPS9 - Serie storica	84
18	MPD1 - Serie storica	85
19	MPS3 - Serie storica	86
20	MPS4 - Serie storica	87
21	MPS5 - Serie storica	88
22	MPS6 - Serie storica	89
23	MPS7 - Serie storica	90
24	MPS8 - Serie storica	91

1 Introduzione

L'obiettivo primario da perseguire è la *qualità_G* del *prodotto_G* e dei suoi processi, ottenibile tramite una serie di controlli stabiliti inizialmente. L'assenza di queste verifiche, abbinata ad un *team di sviluppo_G* con più componenti senza particolari accortezze e competenze, portano al progressivo deterioramento del materiale prodotto, sia esso sorgente o documentazione.

Bisogna pertanto prevenire l'inserimento di materiale non conforme alle *Norme di Progetto v3.0.0* in quanto si avvierebbe un graduale degrado della sua qualità.

1.1 Scopo del documento

Il Piano di Qualifica illustra la strategia di *verifica_G* e *validazione_G* che il gruppo DigitalCookies ha deciso di adottare. È necessario dare una dimensione alla qualità dei prodotti e dei processi, operazioni che non rientrano nei normali ruoli di progetto, bensì rappresentano una *funzione aziendale_G*. Secondo le strategie riportate in questo documento il *committente_G* sarà in grado di valutare oggettivamente quanto è stato prodotto e disporrà di una solida base di verifica.

1.2 Scopo del prodotto

Lo scopo del prodotto è creare un software di costruzione di diagrammi *UML_G* con relativa generazione di codice *Java_G*. Il codice potrà essere generato dall'utente a partire dai diagrammi UML delle *classi_G* e da una versione modificata del diagramma delle *attività_G*.

L'utente, interagendo con il sistema, sarà in grado di:

- delineare la struttura delle classi utilizzando lo standard UML;
- definire il corpo dei metodi delle classi sfruttando una versione modificata del diagramma delle attività;
- generare un applicativo scritto in codice Java a partire dai diagrammi sopracitati.

L'utente potrà inoltre sfruttare la *libreria_G* fornita con il prodotto per generare con facilità diagrammi relativi al dominio dei giochi di carte.

L'*editor_G* sarà fruibile dall'utente attraverso un *browser_G* desktop idoneo all'utilizzo delle tecnologie *HTML5_G*, *CSS3_G* e *JavaScript_G*.

1.3 Ambiguità

Al fine di evitare ogni ambiguità relativa al linguaggio impiegato nei documenti viene fornito il *Glossario v3.0.0*, contenente la definizione dei termini in corsivo marcati con una G pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v3.0.0*;
- **Verbale di incontro esterno** con il *proponente_G* Zucchetti S.p.A. del 23-02-2017;
- **Verbale di incontro interno** con i componenti del gruppo del 21-04-2017;
- **Capitolato_G d'appalto C6:** *SWEDesigner*: editor di diagrammi UML con generazione di codice
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf> (ultima consultazione effettuata in data 22-03-2017);
- **Standard ISO/IEC 15504:** http://en.wikipedia.org/wiki/ISO/IEC_15504 (ultima consultazione effettuata in data 23-04-2017);
- **Standard ISO/IEC 9126:** http://en.wikipedia.org/wiki/ISO/IEC_9126 (ultima consultazione effettuata in data 24-04-2017);
- **Standard IEEE 610.12-90:** https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2677 (ultima consultazione effettuata in data 24-04-2017);
- **Verbale di incontro interno** con i componenti del gruppo del 25-05-2017.

1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v3.0.0*;
- **Slide del corso di Ingegneria del Software:** Qualità del software <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L10.pdf> (ultima consultazione effettuata in data 20-04-2017);
- **SWEBOK 2004 Version - capitolo 11:** <http://www.computer.org/portal/web/swebok/htmlformat>;
- **Slide del corso di Ingegneria del Software:** Qualità di *processo_G* <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L11.pdf> (ultima consultazione effettuata in data 20-04-2017);

-
- **Ciclo di Deming:** https://it.wikipedia.org/wiki/Ciclo_di_Deming (ultima consultazione effettuata in data 20-04-2017);
 - **Indice $Gulpease_G$:** https://it.wikipedia.org/wiki/Indice_Gulpease (ultima consultazione effettuata in data 22-03-2017).

2 Visione generale

La strategia generale adottata è quella di automatizzare il più possibile il lavoro di verifica; questo richiede scelta e uso di $tools_G$ adeguatamente configurati. L'obiettivo è avere un riscontro affidabile e numericamente trattabile che permetta di assicurare il grado di qualità predeterminato. L'aspettativa è la riduzione del lavoro manuale permettendo così un'attività di verifica più semplice.

2.1 Standard adottati

Al fine di fornire linee guide per la valutazione e il miglioramento dei livelli qualitativi dei prodotti realizzati e dei processi in uso, il gruppo ha adottato degli standard come riferimento:

- **qualità di processo:**
 - standard ISO/IEC 15504, detto $SPICE_G$, che permette di classificare la maturità dei processi ed è descritto in dettaglio in appendice A;
 - ciclo di Deming, che permette il miglioramento continuo dei processi ed è descritto in dettaglio in appendice C.
- **qualità di prodotto:**
 - standard ISO/IEC 9126, che individua sei attributi cardine per valutare la qualità dei prodotti ed è descritto in appendice B.

2.2 Risorse

La qualifica dei processi, essendo essa stessa un processo, consuma due tipologie di risorse:

- **Umane:** le figure coinvolte sono il *Responsabile* di Progetto e il *Verificatore_G*, i cui processi effettuati consumano ore di produttività contabilizzate e schedulate secondo il *Piano di Progetto v3.0.0*, che ne definisce anche l'aspetto economico;
- **Tecnologiche:** riguardano i mezzi adottati per utilizzare degli automatismi per la qualità. Trattandosi esclusivamente di mezzi informatici il consumo si limita a $unità_G$ di calcolo considerate a costo nullo, dato che tutte le elaborazioni sono svolte su mezzi per i quali non è richiesto né un contributo economico, né un quantitativo temporale degno di nota. Le modalità in cui tali risorse vengono impiegate sono descritte all'interno del documento *Norme di Progetto v3.0.0*.

2.3 Gestione delle attività di verifica

2.3.1 Responsabilità

La responsabilità della verifica viene attribuita al *Responsabile* di Progetto e ai *Verificatori*. I compiti e le modalità di attuazione sono definiti nel *Piano di Progetto v3.0.0*.

2.3.2 Classificazione delle anomalie

Il processo di *Software Quality Management_G* è finalizzato alla ricerca dei difetti. L'identificazione delle anomalie ne permette la correzione e informa il *Responsabile* di Progetto sullo stato del prodotto. Distinguere e catalogare le anomalie è utile per discutere, durante le revisioni e le riunioni, su quali correzioni attuare e con quale priorità. Di seguito vengono elencate le definizioni di anomalie (IEEE 610.12-90) adottate dal gruppo:

- **Error:** differenza riscontrata tra risultato di una computazione e valore teorico atteso (e.g. uscita dal range di accettazione degli indici di misurazione);
- **Fault:** un passo, un processo o un dato definito in modo erroneo (e.g. violazioni di norme tipografiche da parte di un documento), che corrisponde a quanto viene definito come *bug_G*;
- **Failure:** il risultato di un *fault* (e.g. incongruenza del prodotto con funzionalità indicate nell'*Analisi dei Requisiti v3.0.0*, incongruenza del codice con la definizione del prodotto);
- **Mistake:** azione umana che produce un risultato errato (e.g. anomalie nel *repository_G*).

La distinzione tra i tipi di anomalie consente di impostare le metriche per valutarne l'andamento e in alcuni casi predirlo.

2.4 Organizzazione

Viene verificata la qualità dei singoli processi e dei loro output. La verifica degli output dei periodi descritti nel *Piano di Progetto v3.0.0* prevede attività di verifica mirate, grazie anche alla possibilità di esaminare i soli cambiamenti incrementali esaminando i diari delle modifiche.

- **Analisi dei Requisiti di Massima**

Durante tale attività verrà verificata la conformità dei processi e della documentazione prodotta rispetto le *Norme di Progetto v3.0.0*. Sarà inoltre verificata la corrispondenza tra requisiti e casi d'uso.

- **Analisi dei Requisiti di Dettaglio**

Durante tale attività verranno ampliati i requisiti richiesti dal sistema e migliorati i documenti attuando delle correzioni in base all'esito della Revisione dei Requisiti. Tali modifiche saranno verificate per consistenza e conformità.

- **Progettazione Architetturale**

Durante tale attività avverrà la verifica dei processi incrementali relativi all'analisi e ai nuovi documenti di progettazione. Viene inoltre verificato che i test siano adeguatamente pianificati, secondo le tempistiche descritte nel *Piano di Progetto v3.0.0*, ed eseguiti secondo quanto descritto nelle *Norme di Progetto v3.0.0*.

- **Progettazione di Dettaglio**

Durante tale attività è prevista la stesura, in modo dettagliato, dell'intero sistema, specificando in modo approfondito il comportamento e l'interazione tra i vari componenti. Ogni requisito dovrà essere tracciabile nelle componenti individuate in questa attività.

- **Codifica**

Durante tale attività sarà compiuta la verifica delle attività di codifica tramite tecniche di analisi statica e dinamica, come descritto nelle *Norme di Progetto v3.0.0*.

- **Validazione**

Durante tale attività vengono effettuati tutti i test necessari per garantire che il prodotto soddisfi tutti i requisiti e funzioni correttamente.

Il *Registro delle Modifiche* viene incluso in ogni documento, in modo da mantenere uno storico delle modifiche apportate e delle relative responsabilità.

3 Obiettivi di qualità

In questa sezione vengono descritti gli obiettivi di qualità per i processi e i prodotti adottati dal gruppo e ove possibile espressi in termini quantitativi al fine di semplificare e rendere oggettiva la loro verifica. In particolare sono state usate delle metriche ritenute significative per i vari ambiti, ognuna caratterizzata da un codice identificativo e riportate nel documento *Norme di Progetto v3.0.0*. Per ogni metrica definiremo dunque in questa sezione gli obiettivi qualitativi fissati, denotati da:

- **Range di accettazione:** intervallo in cui il valore misurato viene considerato sufficiente, seppur migliorabile. Qualsiasi processo o prodotto che non soddisfi tale range per una delle sue metriche, andrà obbligatoriamente sottoposto a revisione;
- **Range ottimale:** intervallo in cui il valore misurato viene ritenuto ottimo.

3.1 Processo

Lo standard utilizzato per valutare la qualità dei processi è ISO/IEC 15504, descritto approfonditamente in appendice A, che permette di valutare la maturità dei processi. L'obiettivo perseguito dal gruppo sarà quello di migliorare nel tempo i vari processi secondo gli attributi e i livelli previsti dallo standard. Per ogni processo attuato dal gruppo sono stati inoltre individuati dei vincoli di qualità, ove possibile sotto forma di metriche.

3.1.1 Sviluppo

Nome	Codice	Range
Requirement Stability Index	MPC1	<ul style="list-style-type: none">• Range di accettazione: $\geq 80\%$;• Range ottimale: $\geq 90\%$.
Instability	MPC2	<ul style="list-style-type: none">• Range di accettazione: $\leq 90\%$;• Range ottimale: $\leq 50\%$.
Violazioni dello stile di codifica	MPC3	<ul style="list-style-type: none">• Range di accettazione: 10;• Range ottimale: 0.

Tabella 2: Range delle metriche per il processo di sviluppo

3.1.2 Documentazione

Nome	Codice	Range
Errori frequenti nella documentazione	MPC4	<ul style="list-style-type: none"> • Range di accettazione: 5; • Range ottimale: 0.

Tabella 3: Range delle metriche per il processo di documentazione

3.1.3 Verifica

Nome	Codice	Range
Test di unità eseguiti	MPC5	<ul style="list-style-type: none"> • Range di accettazione: $\geq 90\%$; • Range ottimale: 100%.
Test di integrazione eseguiti	MPC6	<ul style="list-style-type: none"> • Range di accettazione: $\geq 90\%$; • Range ottimale: 100%.
Test di sistema eseguiti	MPC7	<ul style="list-style-type: none"> • Range di accettazione: $\geq 90\%$; • Range ottimale: 100%.

Tabella 4: Range delle metriche per il processo di verifica

3.1.4 Validazione

Nome	Codice	Range
Test di validazione implementati	MPC8	<ul style="list-style-type: none"> • Range di accettazione: $\geq 90\%$; • Range ottimale: 100%.

Tabella 5: Range delle metriche per il processo di validazione

3.1.5 Gestione

Nome	Codice	Range
Schedule Variance	MPC9	<ul style="list-style-type: none"> • Range di accettazione: $\geq -5\%$; • Range ottimale: ≥ 0.
Budget Variance	MPC10	<ul style="list-style-type: none"> • Range di accettazione: $\geq -10\%$; • Range ottimale: ≥ 0.

Tabella 6: Range delle metriche per il processo di gestione

3.2 Prodotto

Lo standard utilizzato per valutare la qualità dei prodotti è ISO/IEC 9126, descritto approfonditamente in appendice B. Per ognuna delle categorie individuate dallo standard abbiamo identificato dei vincoli di qualità, ove possibile sotto forma di metriche.

3.2.1 Funzionalità

Tale caratteristica rappresenta la capacità del prodotto software di provvedere le funzioni necessarie ad adempiere nel modo più completo possibile ai requisiti individuati nel documento *Analisi dei Requisiti v3.0.0*.

Nome	Codice	Range
Copertura requisiti obbligatori	MPS1	<ul style="list-style-type: none"> • Range di accettazione: 100%; • Range ottimale: 100%.
Copertura requisiti desiderabili	MPS2	<ul style="list-style-type: none"> • Range di accettazione: $\geq 50\%$; • Range ottimale: $\geq 65\%$.

Tabella 7: Range delle metriche per la funzionalità dei prodotti

3.2.2 Affidabilità

Tale caratteristica rappresenta la capacità del prodotto software di portare a termine un compito o una funzionalità nel modo previsto, anche al verificarsi di condizioni non previste o non pianificate, le quali possono portare alla visualizzazione di un messaggio di errore ma mai a comportamenti inaspettati.

Nome	Codice	Range
Copertura del codice	MPS10	<ul style="list-style-type: none"> • Range di accettazione: $\geq 75\%$; • Range ottimale: $\geq 90\%$.

Tabella 8: Range delle metriche per l'affidabilità dei prodotti

3.2.3 Efficienza

Tale caratteristica rappresenta la capacità del prodotto software di portare a termine le funzioni richieste nel minor tempo possibile, evitando al contempo sprechi di risorse. In particolare risulta necessario garantire una buona progettazione della base di dati e dei processi per la manipolazione dei dati per quanto riguarda il lato $server_G$, per il quale risulta difficoltoso stabilire metriche adeguate poiché dipendente dalla qualità del servizio esterno AWS_G . Nella codifica delle componenti verrà comunque adottata una metrica per misurare il grado di annidamento delle strutture di controllo.

Nome	Codice	Range
Nested Block Depth	MPS9	<ul style="list-style-type: none"> • Range di accettazione: [0-6]; • Range ottimale: [0-3].

Tabella 9: Range delle metriche per l'efficienza dei prodotti

3.2.4 Usabilità

Tale caratteristica rappresenta la capacità del prodotto software di essere comprensibile ed utilizzabile da parte dell'utente finale. In particolare, oltre a porre attenzione nella realizzazione di un' $interfaccia_G$ intuitiva e $user-friendly_G$, il software sarà corredata di un esaustivo $manuale_G$ utente. Per tale manuale, come per tutti gli altri documenti prodotti dal gruppo, l'usabilità si esplicita in una buona leggibilità, per la quale abbiamo adottato la seguente metrica:

Nome	Codice	Range
Indice Gulpease	MPD1	<ul style="list-style-type: none"> • Range di accettazione: [40-100]; • Range ottimale: [60-100].

Tabella 10: Range delle metriche per l'usabilità dei prodotti

3.2.5 Manutenibilità

Tale caratteristica rappresenta la capacità del prodotto software di essere modificato o espanso, in alcune o tutte le sue componenti, in modo semplice e privo di effetti indesiderati.

Nome	Codice	Range
Average Cyclomatic Complexity	MPS3	<ul style="list-style-type: none"> Range di accettazione: [0-20]; Range ottimale: [0-10].
Coupling Between Object Classes	MPS4	<ul style="list-style-type: none"> Range di accettazione: [0-15]; Range ottimale: [0-12].
Weighted Method Complexity	MPS5	<ul style="list-style-type: none"> Range di accettazione: [0-30]; Range ottimale: [0-20].
Class Size Attributes	MPS6	<ul style="list-style-type: none"> Range di accettazione: [2-16]; Range ottimale: [1-5].
Number of Parameters	MPS7	<ul style="list-style-type: none"> Range di accettazione: [0-9]; Range ottimale: [0-4].
Non-comment Lines of Code	MPS8	<ul style="list-style-type: none"> Range di accettazione: [1-70]; Range ottimale: [1-30].

Tabella 11: Range delle metriche per la manutenibilità dei prodotti

3.2.6 Portabilità

Tale caratteristica rappresenta la capacità del prodotto software di adattarsi ad una molteplicità di ambienti hardware/software. Questa caratteristica verrà soddisfatta implicitamente poiché l'applicativo sarà fruibile attraverso browser web e quindi indipendente dal dispositivo utilizzato dall'utente.

A Standard ISO/IEC 15504

Lo standard ISO/IEC 15504, altrimenti noto come SPICE, è un modello che prevede la classificazione della maturità dei processi in sei livelli e nove attributi distribuiti in questo modo:

- **Livello 0:** Incompleto, non include nessun tipo di indicatore, indica che un processo non è implementato oppure è fallito, cioè non ha prodotto nessun risultato.
- **Livello 1:** Eseguito, il processo è stato implementato e adempie all'obiettivo prefissato. Il relativo attributo di processo è:
 - *Prestazioni di processo:* è la misura in cui lo scopo del progetto è stato raggiunto, come conseguenza della piena realizzazione di questo attributo:
 - * il processo raggiunge i risultati previsti.
- **Livello 2:** Gestito, il processo precedentemente eseguito, i suoi prodotti controllati e mantenuti e le attività pianificare e controllate, documentandone il loro svolgimento. I relativi attributi di processo sono:
 - *Gestione delle prestazioni:* è la misura in cui il processo produce risultati coerenti con gli obiettivi prefissati. Come conseguenza della piena realizzazione di questo attributo:
 - * gli obiettivi delle prestazioni volute sono identificati;
 - * le prestazioni dei processi sono pianificate e monitorate;
 - * le prestazioni dei processi sono adattate in accorda pianificazione;
 - * i ruoli per la realizzazione dei processi sono definiti e comunicati;
 - * risorse e informazioni per realizzare il processo identificate e usate;
 - * gli strumenti tramite i quali avvengono l'assegnazione dei compiti e le comunicazioni formali sono definiti.
 - *Gestione dei prodotti:* è la misura in cui i processi per la produzione dei prodotti sono adeguatamente gestiti. Come conseguenza della piena realizzazione di questo attributo:
 - * i requisiti per i processi di sviluppo del prodotto sono stati identificati;
 - * i requisiti per la documentazione ed il controllo del prodotto sono stati identificati;
 - * i prodotti sono identificati, documentati e controllati adeguatamente;
 - * i prodotti sono revisionati secondo gli accordi presi con il proponente.

- **Livello 3:** Stabilito, il processo definito precedentemente è ora implementato tramite processi capaci di realizzare prodotti adatti. I relativi attributi di processo sono:
 - *Definizione di processo:* è la misura in cui il processo raggiunge i risultati prefissati, aderendo ad uno standard di processo. Come conseguenza della piena realizzazione di questo attributo:
 - * viene definito un processo standard, che includa adeguate linee guida per la sua realizzazione e che descriva gli elementi fondamentali che vanno inclusi in un processo definito;
 - * la $sequenza_G$ e l'interazione dei processi standard con gli altri processi è determinata;
 - * l'infrastruttura e gli ambienti di lavoro per i processi sono identificate come parte dei processi standard;
 - * sono determinati metodi adatti al monitoraggio dell'efficacia e della idoneità dei processi.
 - *Utilizzo di processo:* è la misura in cui il processo sfrutta le risorse allocate. Come conseguenza della piena realizzazione di questo attributo:
 - * è definito un processo basato su un'appropriata e ordinata coda di processi standard;
 - * la formazione personale dei componenti del gruppo è ritenuta accettabile;
 - * sono rese disponibili le risorse per eseguire il processo, gestite e mantenute;
 - * viene dimostrata la fattibilità del processo e valutata l'attività di miglioramento continuo, basata sull'analisi dei dati resi disponibili.
- **Livello 4:** Predicibile, il precedente processo definito come Stabilito, ora viene eseguito con limiti e obiettivi di produzione definiti. I relativi attributi di processo sono:
 - *Misurazione del processo:* è la misura in cui sono misurati i risultati ottenuti per assicurare una buona prestazione dei processi di processo al fine di garantire il raggiungimento degli obiettivi prefissati. Come conseguenza della piena realizzazione di questo attributo:
 - * i processi di supporto agli obiettivi sono identificati;
 - * gli obiettivi di misura del processo sono derivati dalle necessità identificate per il processo;
 - * gli obiettivi quantitativi per le prestazioni del processo sono fissati;

- * le misure e la frequenza delle misurazione sono identificate e pianificate relativamente agli obiettivi fissati;
 - * i risultati delle misurazioni sono raccolti e analizzati;
 - * i risultati delle misurazioni sono usate per tracciare il miglioramento dei processi.
- *Controllo del processo*: è la misura in cui il processo viene controllato tramite la raccolta, l’analisi e l’uso di misurazioni del prodotto e di processo con il fine di correggere, solo se necessario, la sua esecuzione per il raggiungimento degli obiettivi fissati. Come conseguenza della piena realizzazione di questo attributo:
- * le tecniche di controllo e di analisi sono determinate e applicate;
 - * i limiti delle variazioni delle prestazioni del processo vengono accertate;
 - * i dati rilevati con le misurazioni sono analizzate per rilevare cause particolari di variazione;
 - * sono attivate azioni correttive per le cause particolari di variazione rilevate;
 - * i controlli dei limiti sono assodati nuovamente a seguito di azioni correttive.
- **Livello 5:** Ottimizzato, il processo è continuamente migliorato per soddisfare gli obiettivi di business attuali e previsti.
- I relativi attributi di processo sono:
- *Innovazione di processo*: è la misura in cui i cambiamenti relativi alla gestione, all’esecuzione e alla definizione di processo sono controllati per raggiungere gli obiettivi di business. Come conseguenza della piena realizzazione di questo attributo:
- * sono definiti i processi di supporto agli obiettivi per il raggiungimento di rilevanti obiettivi di business;
 - * vengono analizzati e identificati le cause comuni di variazioni alle prestazioni dei processi;
 - * vengono analizzati le opportunità di attuazione di nuove *best practice_G* al fine di innovazione;
 - * vengono identificate nuove opportunità di miglioramento derivate da nuove tecnologie.

– *Ottimizzazione di processo:* è la misura in cui vengono identificati e implementati i cambiamenti relativi all'esecuzione del processo. Come conseguenza della piena realizzazione di questo attributo:

- * l'impatto di tutte le modifiche in base agli obiettivi viene valutato;
- * l'implementazione di qualsiasi cambiamento concordato è gestita in modo da assicurare che qualsiasi calo delle prestazioni del processo sia compreso e accettato;
- * l'efficacia del processo di cambiamento, relativamente al processo attuale, viene valutata tenendo conto dei requisiti e degli obiettivi di processo al fine di determinare quale dia risultati migliori.

Nella realizzazione del progetto il gruppo punterà a migliorare il livello dei vari processi tramite un metodo di miglioramento continuo descritto in appendice C.

I risultati riscontrati sono riportati nell'appendice E.

B Standard ISO/IEC 9126

Per la qualità del prodotto software sono state considerate le normative e le linee guida presenti nello Standard ISO/IEC 9126, che è suddiviso in quattro parti:

- **9126-1**: modello della qualità del software;
- **9126-2**: metriche per la qualità esterna;
- **9126-3**: metriche per la qualità interna;
- **9126-4**: metriche per la qualità d'uso.

Il modello della qualità del software è in particolare classificato da sei caratteristiche generali:

- **funzionalità**: la capacità del prodotto software di fornire funzioni che soddisfino le esigenze stabilite;
- **affidabilità**: la capacità del prodotto software di mantenere un certo livello di prestazioni;
- **efficienza**: la capacità del prodotto software di fornire adeguate prestazioni in relazione alle risorse utilizzate;
- **usabilità**: la capacità del prodotto software di essere compreso e apprezzato dall'utente;
- **manutenibilità_G**: la capacità del prodotto software di essere modificato, migliorato o corretto facilmente nel tempo;
- **portabilità**: la capacità del prodotto software di essere trasportato da un ambiente di lavoro ad un altro.

Tali caratteristiche sono misurabili tramite varie metriche, da noi adottate come obiettivi di qualità nella sezione 3. Le varie metriche, così come i test pianificati in questo documento, si attengono alle tre categorie di qualità previste da questo standard:

- **Qualità esterna**: rappresenta il comportamento del software durante la sua esecuzione, ed è rilevata dai test su obiettivi stabiliti in un contesto tecnico rilevante;
- **Qualità interna**: rappresenta la qualità del software sorgente non ancora eseguibile e della documentazione correlata al prodotto. Tali aspetti permettono di stimare la qualità esterna e la qualità in uso del prodotto finale, poiché gli attributi interni influiscono su quelli esterni e quelli in uso;
- **Qualità in uso**: rappresenta il punto di vista dell'utente sul prodotto software, che deve garantire efficacia, produttività, sicurezza e soddisfazione. Un livello adeguato di qualità in uso implica il previo raggiungimento di un buon livello di qualità interna ed esterna.

C PDCA

Il $PDCA_G$, altrimenti noto come "Ciclo di Deming", è un metodo iterativo in quattro fasi G ideato per il miglioramento continuo della qualità dei processi in un'ottica a lungo raggio. Tale metodo si compone di quattro stadi che vengono ripetuti iterativamente. Ciò permette di migliorare gradualmente la qualità dei processi per quanto riguarda la loro efficienza, ossia l'ottimizzazione delle risorse che essi utilizzano, e la loro efficacia, ossia la loro conformità rispetto alle aspettative. Le quattro attività previste sono le seguenti:

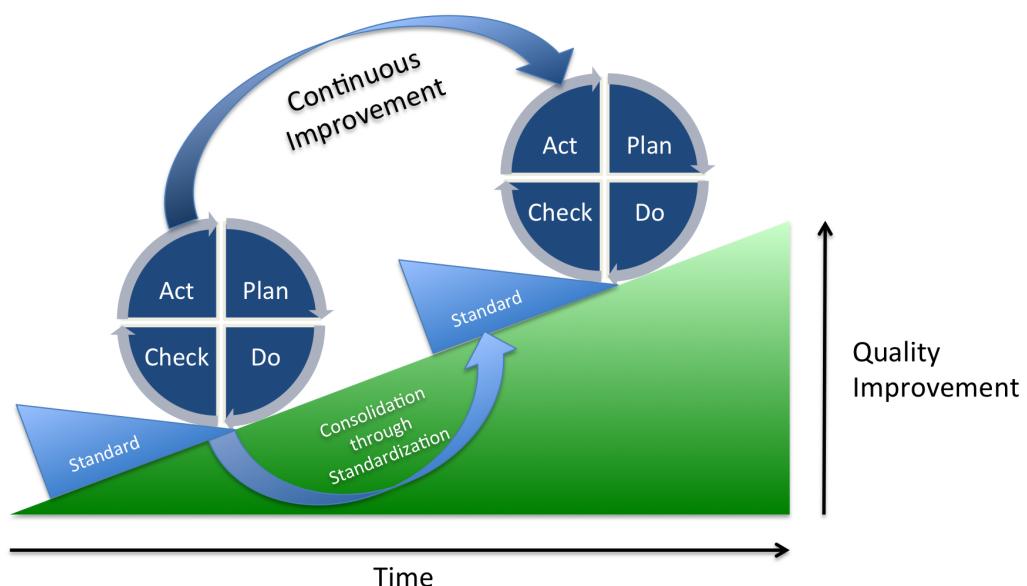


Figura 1: Continuous quality improvement with PDCA

- **Plan:** definizione degli obiettivi di miglioramento e delle relative strategie da adottare per raggiungerli, se possibile inizialmente su scala ridotta per poterne valutare gli effetti;
- **Do:** attuazione di quanto pianificato al punto precedente, con raccolta di dati significativi da poter analizzare negli stadi successivi;
- **Check:** verifica dell'esito del processo in seguito all'attuazione delle strategie di miglioramento, con studio dei risultati raccolti durante il *Do* e confronto di questi ultimi con i risultati attesi nel *Plan* per stimare l'impatto del miglioramento apportato;
- **Act:** attuazione delle strategie che hanno portato miglioramenti, eventualmente estendendole anche all'infuori dei singoli processi per cui erano state pianificate inizialmente. Nel caso di differenze significative tra i risultati previsti e quelli

effettivi possono essere richieste delle azioni correttive a seguito di un'analisi delle cause.

Una volta raggiunto un livello tale per cui non è ritenuto necessario migliorare la portata a cui è attualmente applicato, il PDCA può essere raffinato con l'obiettivo di migliorare in maggior dettaglio la successiva iterazione oppure l'attenzione può essere spostata su una differente fase del processo.

D Specifica dei Test

Vengono riportati e descritti in questa sezione i test che nelle successive attività verranno implementati in modo tale che, al loro superamento, sia possibile garantire livelli di qualità ottimali, il corretto funzionamento e la conformità alle aspettative del committente dell'applicativo software prodotto. La definizione dei test potrà comunque subire incrementi e variazioni fino all'approvazione definitiva da parte di un responsabile del documento *Definizione di Prodotto*, che verrà redatto nella successiva revisione e comporterà l'aggiunta di test di unità. Ogni test è identificato da un codice univoco la cui sintassi viene descritta nel documento *Norme di Progetto v3.0.0*.

D.1 Test di Validazione

Tale tipologia di test viene utilizzata durante l'attività di collaudo del prodotto finale, per accertare che il prodotto sia conforme alle attese del committente. Per ogni test viene specificato il proprio codice univoco, la descrizione, contenente i passi che l'utente deve seguire, e lo stato di implementazione attuale.

Codice	Descrizione	Stato
TV1O2	<p>L'utente intende creare un nuovo progetto. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere sul pulsante di creazione di un nuovo progetto; • inserire il nome del nuovo progetto; • confermare la creazione di un nuovo progetto. 	Non implementato
TV1O2.1	<p>L'utente intende inserire il nome di un nuovo progetto. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionarsi sulla form per l'inserimento del nome del nuovo progetto; • digitare il nome del nuovo progetto. 	Non implementato
TV1O3.2	<p>L'utente intende caricare un progetto da un <i>file_G</i> locale. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere sul pulsante di caricamento di un progetto salvato; • scegliere il file da cui caricare il progetto; • confermare la scelta del file. 	Non implementato
TV1O3.2.1	<p>L'utente intende visualizzare un messaggio d'errore nel caso in cui il file scelto non abbia estensione adeguata. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • scegliere file da locale che non sia di estensione adeguata. 	Non implementato

TV1O4	<p>L'utente intende visualizzare la schermata di creazione del diagramma delle classi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • aprire un progetto se si trova nella schermata iniziale; • selezionare la scheda relativa alla visualizzazione del diagramma delle classi se si trova all'interno di un progetto. 	Non implementato
TV1O4.1	<p>L'utente intende inserire una <i>classe_G</i> di <i>libreria_G</i> fornita dall'editor. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • espandere la lista delle classi fornite dalla libreria; • cliccare sulla classe di interesse. 	Non implementato
TV1D4.2	<p>L'utente intende inserire un <i>design pattern_G</i> di libreria fornito dall'editor. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • espandere la lista dei design pattern forniti da libreria; • cliccare sul design pattern di interesse. 	Non implementato
TV1O4.3	<p>L'utente intende inserire una nuova classe nel diagramma delle classi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova classe nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O4.4	<p>L'utente intende modificare una classe esistente nel diagramma delle classi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare doppio click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno della classe per uscire dall'interfaccia di modifica. 	Non implementato
TV1O4.4.1	<p>L'utente intende modificare la priorità di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore sul campo della priorità attuale e fare click con il tasto destro; • selezionare la nuova priorità della classe dal menù a tendina. 	Non implementato
TV1O4.4.2	<p>L'utente intende modificare il nome di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul nome attuale della classe; • digitare il nuovo nome della classe nell'apposita form. 	Non implementato

TV1D4.4.3	<p>L'utente intende modificare il colore di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore in qualsiasi punto vuoto interno alla classe e fare click con il tasto destro; • selezionare il nuovo colore della classe dal menù a tendina. 	Non implementato
TV1O4.4.4	<p>L'utente intende inserire un attributo in una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • espandere la lista degli attributi se ridotta; • premere il pulsante di aggiunta di un attributo. 	Non implementato
TV1O4.4.5	<p>L'utente intende modificare un attributo di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attributo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno della classe per uscire dall'interfaccia di modifica. 	Non implementato
TV1O4.4.5.1	<p>L'utente intende modificare la visibilità di un attributo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore sopra l'attuale visibilità dell'attributo e fare click con il tasto destro; • selezionare la nuova visibilità dell'attributo dal menù a tendina. 	Non implementato
TV1O4.4.5.2	<p>L'utente intende modificare il tipo di un attributo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale tipo dell'attributo a cui vuole apportare modifiche; • digitare nell'apposita form il nuovo tipo dell'attributo. 	Non implementato
TV1O4.4.5.3	<p>L'utente intende modificare il nome di un attributo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale nome dell'attributo a cui vuole apportare modifiche; • digitare nell'apposita form il nuovo nome dell'attributo. 	Non implementato

TV1O4.4.5.4	<p>L'utente intende modificare il valore di default di un attributo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale valore di default dell'attributo a cui vuole apportare modifiche; • digitare nell'apposita form il nuovo valore di default dell'attributo. 	Non implementato
TV1O4.4.6	<p>L'utente intende rimuovere un attributo della classe che sta modificando. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere il pulsante "x" relativo all'attributo da rimuovere. 	Non implementato
TV1O4.4.7	<p>L'utente intende inserire un nuovo metodo in una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • espandere la lista dei metodi se ridotta; • premere il pulsante di inserimento di un metodo. 	Non implementato
TV1O4.4.8	<p>L'utente intende modificare un metodo di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare doppio click sul metodo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno della classe per uscire dall'interfaccia di modifica. 	Non implementato
TV1O4.4.8.1	<p>L'utente intende modificare la visibilità di un metodo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore sopra l'attuale visibilità del metodo e fare click con il tasto destro; • selezionare la nuova visibilità del metodo dal menù a tendina. 	Non implementato
TV1O4.4.8.2	<p>L'utente intende modificare il nome di un metodo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale nome del metodo a cui vuole apportare modifiche; • digitare nell'apposita form il nuovo nome del metodo. 	Non implementato

TV1O4.4.8.3	<p>L'utente intende modificare il tipo di ritorno di un metodo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale tipo di ritorno del metodo a cui vuole apportare modifiche; • digitare il nuovo tipo di ritorno del metodo. 	Non implementato
TV1O4.4.8.4	<p>L'utente intende inserire un nuovo parametro in un metodo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sull'attuale lista di parametri del metodo a cui vuole apportare modifiche; • posizionare il cursore al termine della segnatura di uno dei parametri già presenti; • digitare una virgola; • digitare il nome del nuovo parametro, seguito da due punti e dal tipo del nuovo parametro. 	Non implementato
TV1O4.4.8.5	<p>L'utente intende modificare un parametro di un metodo a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sulla lista di parametri del metodo a cui vuole apportare modifiche; • posizionare il cursore all'altezza del parametro da modificare; • apportare le modifiche desiderate; • cliccare all'esterno della classe per uscire dall'interfaccia di modifica. 	Non implementato
TV1O4.4.8.5.1	<p>L'utente intende modificare il tipo di un parametro a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore all'altezza del tipo del parametro da modificare; • digitare il nuovo tipo del parametro. 	Non implementato
TV1O4.4.8.5.2	<p>L'utente intende modificare il nome di un parametro a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore all'altezza del nome del parametro da modificare; • digitare il nuovo nome del parametro. 	Non implementato

TV1O4.4.8.5.3	<p>L'utente intende modificare il valore di default di un parametro a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore all'altezza de valore di default del parametro da modificare; • digitare il nuovo valore di default del parametro. 	Non implementato
TV1O4.4.8.6	<p>L'utente intende rimuovere un parametro di un metodo che sta modificando. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • posizionare il cursore all'altezza del parametro da rimuovere; • eliminare il testo relativo al parametro da rimuovere. 	Non implementato
TV1O4.4.9	<p>L'utente intende rimuovere un metodo di una classe che sta modificando. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere il pulsante "x" relativo al metodo da rimuovere. 	Non implementato
TV1O4.4.10	<p>L'utente intende modificare lo stereotipo di una classe a cui sta apportando modifiche. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sopra lo stereotipo attuale della classe; • selezionare il nuovo stereotipo della classe dal menù a tendina. 	Non implementato
TV1O4.5	<p>L'utente intende rimuovere una classe dal diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare la classe da rimuovere; • premere il tasto "canc". 	Non implementato
TV1O4.5.1	<p>L'utente intende visualizzare un messaggio d'errore nel caso in cui l'eliminazione della classe scelta non vada a buon fine. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • scegliere una classe alla quale sono collegate delle relazioni. 	Non implementato
TV1O4.6.1	<p>L'utente intende inserire una nuova relazione di associazione nel diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova relazione di associazione nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O4.6.2	<p>L'utente intende inserire una nuova relazione di dipendenza nel diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova relazione di dipendenza nella posizione in cui desidera il suo inserimento. 	Non implementato

TV1O4.6.3	<p>L'utente intende inserire una nuova relazione di composizione nel diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova relazione di composizione nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O4.6.4	<p>L'utente intende inserire una nuova relazione di aggregazione nel diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova relazione di aggregazione nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O4.6.5	<p>L'utente intende inserire una nuova relazione di generalizzazione nel diagramma. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuova relazione di generalizzazione nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O4.7.1	<p>L'utente intende modificare la cardinalità di una relazione tra due classi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare la form relativa alla cardinalità della relazione che vuole modificare; • digitare la nuova cardinalità della relazione. 	Non implementato
TV1O4.7.2	<p>L'utente intende scegliere una nuova la classe di partenza per una relazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare l'inizio della relazione sulla classe di interesse. 	Non implementato
TV1O4.7.3	<p>L'utente intende scegliere una nuova la classe d'arrivo per una relazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il termine della relazione sulla classe di interesse. 	Non implementato
TV1O4.8	<p>L'utente intende rimuovere una relazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare la relazione da rimuovere; • premere il tasto "canc". 	Non implementato
TV1D4.9	<p>L'utente intende inserire un nuovo commento. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante una nuovo commento nella posizione in cui desidera. 	Non implementato
TV1D4.10	<p>L'utente intende modificare il testo di un commento. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • digitare il nuovo testo del commento. 	Non implementato

TV1D4.11	<p>L'utente intende rimuovere un commento. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare il commento da rimuovere; • premere il tasto "canc". 	Non implementato
TV1D4.12	<p>L'utente intende espandere una classe ridotta. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare una classe ridotta; • premere i triangolini rovesciati collocati a destra della lista degli attributi e dei metodi. 	Non implementato
TV1D4.13	<p>L'utente intende ridurre una classe espansa. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare una classe espansa; • premere i triangolini collocati a destra della lista degli attributi e dei metodi. 	Non implementato
TV1D4.14	<p>L'utente intende espandere un commento ridotto. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un commento ridotto; • premere il triangolino rovesciato collocato a destra del commento. 	Non implementato
TV1D4.15	<p>L'utente intende ridurre un commento espanso. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un commento espanso; • premere il triangolino collocato a destra del commento. 	Non implementato
TV1D4.16	<p>L'utente intende applicare filtri per la visualizzazione grafica delle classi in base alla loro priorità. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare dalla barra di sinistra i checkbox delle priorità da visualizzare. 	Non implementato
TV1D4.17	<p>L'utente intende spostare un blocco esistente all'interno del diagramma delle classi. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare il blocco che si vuole spostare; • trascinare il blocco nella nuova posizione. 	Non implementato
TV1O5	<p>L'utente intende visualizzare la schermata di creazione del diagramma delle attività. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare la scheda relativa al diagramma delle attività. 	Non implementato

TV1O5.1	L'utente intende visualizzare il diagramma delle attività di uno dei metodi definiti. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare il metodo per il quale vuole visualizzare il diagramma delle attività. 	Non implementato
TV1O5.2.1.1	L'utente intende inserire un nuovo blocco variabile nel metodo selezionato. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco variabile nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O5.2.1.1.1	L'utente intende collegare una variabile ad un nuovo blocco variabile nel metodo selezionato. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare il blocco variabile di interesse; • digitare il nome di una variabile già esistente nel campo relativo al nome della variabile' 	Non implementato
TV1O5.2.1.2	L'utente intende inserire un blocco chiamata di un metodo. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco chiamata di un metodo nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O5.2.1.3	L'utente intende inserire un nuovo blocco ciclo all'interno di un metodo. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco ciclo nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O5.2.1.4	L'utente intende inserire un nuovo blocco if/else all'interno di un metodo. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco if/else nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O5.2.1.5	L'utente intende inserire un nuovo blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco operatore nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1O5.2.1.6	L'utente intende inserire un nuovo blocco avanzamento. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare il blocco di partenza; • trascinare dal bordo del blocco selezionato il cursore del mouse fino al raggiungimento del blocco d'arrivo. 	Non implementato

TV1O5.2.1.7	<p>L'utente intende inserire un nuovo blocco Jolly. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • trascinare il blocco rappresentante il blocco Jolly nella posizione in cui desidera il suo inserimento. 	Non implementato
TV1D5.2.2.1	<p>L'utente intende modificare un blocco variabile nel metodo selezionato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno del blocco per uscire dall'interfaccia di modifica. 	Non implementato
TV1D5.2.2.1.1	<p>L'utente intende modificare il tipo di un blocco variabile nel metodo selezionato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • digitare il nuovo tipo. 	Non implementato
TV1D5.2.2.1.2	<p>L'utente intende modificare l'inizializzazione di un blocco variabile nel metodo selezionato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • digitare il nuovo valore d'inizializzazione del blocco variabile. 	Non implementato
TV1O5.2.2.2	<p>L'utente intende modificare un blocco chiamata di un metodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno del blocco per uscire dall'interfaccia di modifica. 	Non implementato
TV1O5.2.2.2.1	<p>L'utente intende modificare l'invocazione del metodo di un blocco. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • digitare il nuovo metodo da invocare. 	Non implementato
TV1O5.2.2.2.2	<p>L'utente intende modificare i parametri di invocazione ad un metodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • digitare per ogni parametro un valore o il nome di una variabile compatibile negli appositi spazi. 	Non implementato
TV1O5.2.2.3	<p>L'utente intende modificare un blocco ciclo di un metodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno del blocco per uscire dall'interfaccia di modifica. 	Non implementato

TV1O5.2.2.3.1	<p>L'utente intende modificare la condizione di un blocco ciclo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante la condizione i blocchi necessari a comporre la condizione del blocco. 	Non implementato
TV1O5.2.2.3.2	<p>L'utente intende modificare il corpo di un blocco ciclo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante il corpo i blocchi necessari a comporre il corpo del blocco. 	Non implementato
TV1O5.2.2.4	<p>L'utente intende modificare un blocco if/else all'interno di un metodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno del blocco per uscire dall'interfaccia di modifica. 	Non implementato
TV1O5.2.2.4.1	<p>L'utente intende modificare la condizione di un blocco if/else. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante la condizione i blocchi necessari a comporre la condizione del blocco. 	Non implementato
TV1O5.2.2.4.2	<p>L'utente intende modificare il corpo del ramo if di un blocco if/else. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante il corpo del ramo if i blocchi necessari a comporre il corpo del ramo. 	Non implementato
TV1O5.2.2.4.3	<p>L'utente intende modificare il corpo del ramo else di un blocco if/else. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante il corpo del ramo else i blocchi necessari a comporre il corpo del ramo. 	Non implementato
TV1O5.2.2.5	<p>L'utente intende modificare un blocco operatore all'interno di un metodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • fare click sul campo a cui vuole apportare modifiche; • apportare le modifiche desiderate; • cliccare all'esterno del blocco per uscire dall'interfaccia di modifica. 	Non implementato

TV1O5.2.2.5.1	L'utente intende modificare l'operatore di un blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • posizionare il cursore sul blocco operatore e fare click con il tasto destro; • selezionare il nuovo operatore del blocco operatore dal menù tendina. 	Non implementato
TV1O5.2.2.5.2	L'utente intende modificare l'elemento a destra di un blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante l'elemento a destra i blocchi necessari. 	Non implementato
TV1O5.2.2.5.3	L'utente intende modificare l'elemento a sinistra di un blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • inserire nel sotto-diagramma delle attività rappresentante l'elemento a sinistra i blocchi necessari. 	Non implementato
TV1O5.2.2.5.4	L'utente intende modificare l'elemento risultato di un blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • inserire il nuovo blocco variabile a cui assegnare il risultato dell'operazione. 	Non implementato
TV1O5.2.2.6.1	L'utente intende scegliere un nuovo blocco di partenza per un blocco avanzamento. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare l'inizio dell'associazione sul blocco desiderato. 	Non implementato
TV1O5.2.2.6.2	L'utente intende scegliere un nuovo blocco d'arrivo per un blocco avanzamento. All'utente è richiesto di: <ul style="list-style-type: none"> • trascinare il termine dell'associazione sul blocco desiderato. 	Non implementato
TV1O5.2.2.7	L'utente intende modificare il corpo di un blocco Jolly. All'utente è richiesto di: <ul style="list-style-type: none"> • digitare il nuovo testo del blocco Jolly. 	Non implementato
TV1D5.2.3.1	L'utente intende rimuovere un blocco variabile. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco variabile; • premere il tasto "canc". 	Non implementato
TV1O5.2.3.2	L'utente intende rimuovere un blocco chiamata di un metodo. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco chiamata di un metodo; • premere il tasto "canc". 	Non implementato

TV1O5.2.3.3	L'utente intende rimuovere un blocco ciclo. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco ciclo; • premere il tasto "canc". 	Non implementato
TV1O5.2.3.4	L'utente intende rimuovere un blocco if/else. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco if/else; • premere il tasto "canc". 	Non implementato
TV1O5.2.3.5	L'utente intende rimuovere un blocco operatore. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco operatore; • premere il tasto "canc". 	Non implementato
TV1O5.2.3.6	L'utente intende rimuovere un blocco avanzamento. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco avanzamento; • premere il tasto "canc". 	Non implementato
TV1O5.2.3.7	L'utente intende rimuovere un blocco Jolly. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco Jolly; • premere il tasto "canc". 	Non implementato
TV1D5.2.4.1	L'utente intende espandere un blocco variabile ridotto. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco variabile ridotto; • premere il triangolino rovesciato collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.4.2	L'utente intende espandere un blocco chiamata ridotto. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco chiamata ridotto; • premere il triangolino rovesciato collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.4.3	L'utente intende espandere un blocco ciclo ridotto. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco ciclo ridotto; • premere il bottone '+' collocato a sinistra del nome del blocco. 	Non implementato
TV1D5.2.4.4	L'utente intende espandere un blocco if/else ridotto. All'utente è richiesto di: <ul style="list-style-type: none"> • selezionare un blocco if/else ridotto; • premere il bottone '+' collocato a sinistra del nome del blocco. 	Non implementato

TV1D5.2.4.5	<p>L'utente intende espandere un blocco Jolly ridotto.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco Jolly ridotto; • premere il triangolino rovesciato collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.5.1	<p>L'utente intende ridurre un blocco variabile espanso.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco variabile espanso; • premere il triangolino collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.5.2	<p>L'utente intende ridurre un blocco chiamata espanso.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco chiamata espanso; • premere il triangolino collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.5.3	<p>L'utente intende ridurre un blocco ciclo espanso.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco ciclo espanso; • premere il bottone '-' collocato a sinistra del nome del blocco. 	Non implementato
TV1D5.2.5.4	<p>L'utente intende ridurre un blocco if/else espanso.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco if/else espanso; • premere il bottone '-' collocato a sinistra del nome del blocco. 	Non implementato
TV1D5.2.5.5	<p>L'utente intende ridurre un blocco Jolly espanso.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare un blocco Jolly espanso; • premere il triangolino collocato a destra del nome del blocco. 	Non implementato
TV1D5.2.6	<p>L'utente intende spostare un blocco esistente all'interno del diagramma delle attività.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare il blocco che si vuole spostare; • trascinare il blocco nella nuova posizione. 	Non implementato
TV1O5.3	<p>L'utente intende rimuovere il diagramma delle attività attualmente visualizzato.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere il pulsante 'x' per la rimozione del diagramma delle attività. 	Non implementato

TV1O5.4	<p>L'utente intende inserire un metodo da libreria fornito dall'editor. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • premere il pulsante 'passa a libreria' nella barra di destra; • premere sul pulsante di inserimento relativo al metodo di libreria d'interesse. 	Non implementato
TV1O6	<p>L'utente intende visualizzare il codice sorgente generato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • selezionare la scheda relativa alla generazione del codice; • premere sul pulsante di generazione del codice. 	Non implementato
TV1O6.1	<p>L'utente intende visualizzare un messaggio d'errore nel caso in cui la creazione del codice non vada a buon fine. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • non implementare uno dei metodi del diagramma delle attività; • premere sul pulsante di generazione del codice. 	Non implementato
TV1D6.2	<p>L'utente intende modificare il codice sorgente generato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • collocarsi nell'editor; • apportare le modifiche desiderate; • cliccare sul pulsante di salvataggio. 	Non implementato
TV1O6.3	<p>L'utente intende esportare il codice sorgente generato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • cliccare sul pulsante di esportazione codice. 	Non implementato
TV1O7	<p>L'utente intende salvare un progetto. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • cliccare sul pulsante di salvataggio per il progetto. 	Non implementato
TV1O7.1	<p>L'utente intende esportare un progetto come file. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • cliccare sul pulsante per scaricare il progetto come file. 	Non implementato

Tabella 12: Test di Validazione

Test di validazione

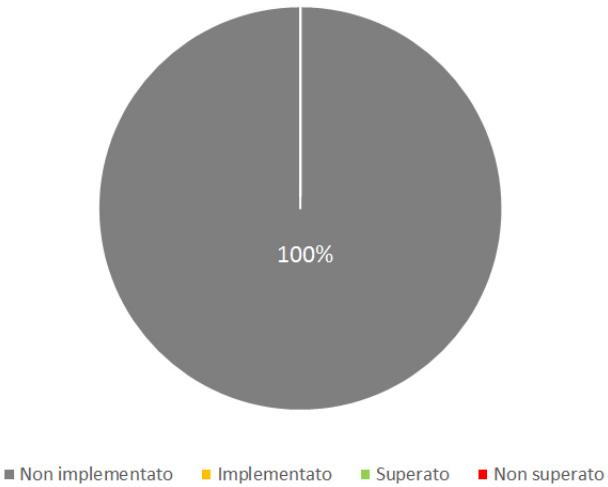


Figura 2: Test di validazione - stato corrente

D.1.1 Tracciamento test di validazione-requisiti

Test	Requisito
TV1O2	R1O2
TV1O2.1	R1O2.1
TV1O3.2	R1O3.2
TV1O3.2.1	R1O3.2.1
TV1O4	R1O4
TV1O4.1	R1O4.1
TV1D4.2	R1D4.2
TV1O4.3	R1O4.3
TV1O4.4	R1O4.4
TV1O4.4.1	R1O4.4.1
TV1O4.4.2	R1O4.4.2
TV1D4.4.3	R1D4.4.3
TV1O4.4.4	R1O4.4.4
TV1O4.4.5	R1O4.4.5
TV1O4.4.5.1	R1O4.4.5.1
TV1O4.4.5.2	R1O4.4.5.2
TV1O4.4.5.3	R1O4.4.5.3
TV1O4.4.5.4	R1O4.4.5.4
TV1O4.4.6	R1O4.4.6

TV1O4.4.7	R1O4.4.7
TV1O4.4.8	R1O4.4.8
TV1O4.4.8.1	R1O4.4.8.1
TV1O4.4.8.2	R1O4.4.8.2
TV1O4.4.8.3	R1O4.4.8.3
TV1O4.4.8.4	R1O4.4.8.4
TV1O4.4.8.5	R1O4.4.8.5
TV1O4.4.8.5.1	R1O4.4.8.5.1
TV1O4.4.8.5.2	R1O4.4.8.5.2
TV1O4.4.8.5.3	R1O4.4.8.5.3
TV1O4.4.8.6	R1O4.4.8.6
TV1O4.4.9	R1O4.4.9
TV1O4.4.10	R1O4.4.10
TV1O4.5	R1O4.5
TV1O4.5.1	R1O4.5.1
TV1O4.6.1	R1O4.6.1
TV1O4.6.2	R1O4.6.2
TV1O4.6.2	R1O4.6.2
TV1O4.6.2	R1O4.6.3
TV1O4.6.2	R1O4.6.4
TV1O4.6.2	R1O4.6.5
TV1O4.7.1	R1O4.7.1
TV1O4.7.2	R1O4.7.2
TV1O4.7.3	R1O4.7.3
TV1O4.8	R1O4.8
TV1D4.9	R1D4.9
TV1D4.10	R1D4.10
TV1D4.11	R1D4.11
TV1D4.12	R1D4.12
TV1D4.13	R1D4.13
TV1D4.14	R1D4.14
TV1D4.15	R1D4.15
TV1D4.16	R1D4.16
TV1D4.17	R1D4.17
TV1O5	R1O5
TV1O5.1	R1O5.1
TV1O5.2.1.1	R1O5.2.1.1
TV1O5.2.1.1.1	R1O5.2.1.1.1
TV1O5.2.1.2	R1O5.2.1.2
TV1O5.2.1.3	R1O5.2.1.3
TV1O5.2.1.4	R1O5.2.1.4

TV1O5.2.1.5	R1O5.2.1.5
TV1O5.2.1.6	R1O5.2.1.6
TV1O5.2.1.7	R1O5.2.1.7
TV1D5.2.2.1	R1D5.2.2.1
TV1D5.2.2.1.1	R1D5.2.2.1.1
TV1D5.2.2.1.2	R1D5.2.2.1.2
TV1O5.2.2.2	R1O5.2.2.2
TV1O5.2.2.2.1	R1O5.2.2.2.1
TV1O5.2.2.2.2	R1O5.2.2.2.2
TV1O5.2.2.3	R1O5.2.2.3
TV1O5.2.2.3.1	R1O5.2.2.3.1
TV1O5.2.2.3.2	R1O5.2.2.3.2
TV1O5.2.2.4	R1O5.2.2.4
TV1O5.2.2.4.1	R1O5.2.2.4.1
TV1O5.2.2.4.2	R1O5.2.2.4.2
TV1O5.2.2.4.3	R1O5.2.2.4.3
TV1O5.2.2.5	R1O5.2.2.5
TV1O5.2.2.5.1	R1O5.2.2.5.1
TV1O5.2.2.5.2	R1O5.2.2.5.2
TV1O5.2.2.5.3	R1O5.2.2.5.3
TV1O5.2.3.5.4	R1O5.2.3.5.4
TV1O5.2.2.6.1	R1O5.2.2.6.1
TV1O5.2.2.6.2	R1O5.2.2.6.2
TV1O5.2.2.7	R1O5.2.2.7
TV1D5.2.3.1	R1D5.2.3.1
TV1O5.2.3.2	R1O5.2.3.2
TV1O5.2.3.3	R1O5.2.3.3
TV1O5.2.3.4	R1O5.2.3.4
TV1O5.2.3.5	R1O5.2.3.5
TV1O5.2.3.6	R1O5.2.3.6
TV1O5.2.3.7	R1O5.2.3.7
TV1D5.2.4.1	R1D5.2.4.1
TV1D5.2.4.2	R1D5.2.4.2
TV1D5.2.4.3	R1D5.2.4.3
TV1D5.2.4.4	R1D5.2.4.4
TV1D5.2.4.5	R1D5.2.4.5
TV1D5.2.5.1	R1D5.2.5.1
TV1D5.2.5.2	R1D5.2.5.2
TV1D5.2.5.3	R1D5.2.5.3
TV1D5.2.5.4	R1D5.2.5.4
TV1D5.2.5.5	R1D5.2.5.5

TV1D5.2.6	R1D5.2.6
TV1O5.3	R1O5.3
TV1O5.4	R1O5.4
TV1O6	R1O6
TV1O6.1	R1O6.1
TV1D6.2	R1D6.2
TV1O6.3	R1O6.3
TV1O7	R1O7
TV1O7.1	R1O7.1

D.2 Test di Sistema

Tale tipologia di test serve per verificare che il comportamento dinamico complessivo dell'intero sistema sia conforme ai requisiti definiti nel documento *Analisi dei Requisiti v3.0.0*. Per ogni test viene specificato il proprio codice univoco, la descrizione e lo stato di implementazione attuale.

Codice	Descrizione	Stato
TS1O2	Viene verificato che il sistema permetta all'utente di creare un nuovo progetto.	Implementato
TS1O2.1	Viene verificato che il sistema permetta all'utente di inserire il nome del nuovo progetto durante la sua creazione.	Implementato
TS1O3.2	Viene verificato che il sistema permetta all'utente di caricare un progetto da un file locale.	Non implementato
TS1O3.2.1	Viene verificato che il sistema visualizzi un errore nel caso in cui il file scelto non sia di estensione adeguata.	Non implementato
TS1O4	Viene verificato che il sistema permetta all'utente di visualizzare la schermata di creazione del diagramma delle classi.	Implementato
TS1O4.1	Viene verificato che il sistema permetta all'utente di inserire una classe di libreria fornita dall'editor.	Non implementato
TS1D4.2	Viene verificato che il sistema permetta all'utente di inserire un design pattern di libreria fornito dall'editor.	Non implementato
TS1O4.3	Viene verificato che il sistema permetta all'utente di inserire una nuova classe nel diagramma delle classi.	Implementato
TS1O4.4	Viene verificato che il sistema permetta all'utente di modificare una classe esistente nel diagramma delle classi.	Implementato

TS1O4.4.1	Viene verificato che il sistema permetta all'utente di modificare la priorità di una classe esistente.	Implementato
TS1O4.4.2	Viene verificato che il sistema permetta all'utente di modificare il nome di una classe esistente.	Implementato
TS1D4.4.3	Viene verificato che il sistema permetta all'utente di modificare il colore di una classe esistente.	Implementato
TS1O4.4.4	Viene verificato che il sistema permetta all'utente di inserire un attributo in una classe esistente.	Implementato
TS1O4.4.5	Viene verificato che il sistema permetta all'utente di modificare un attributo di una classe esistente.	Implementato
TS1O4.4.5.1	Viene verificato che il sistema permetta all'utente di modificare la visibilità di un attributo esistente.	Implementato
TS1O4.4.5.2	Viene verificato che il sistema permetta all'utente di modificare il tipo di un attributo esistente.	Implementato
TS1O4.4.5.3	Viene verificato che il sistema permetta all'utente di modificare il nome di un attributo esistente.	Implementato
TS1O4.4.5.4	Viene verificato che il sistema permetta all'utente di modificare il valore di default di un attributo esistente.	Implementato
TS1O4.4.6	Viene verificato che il sistema permetta all'utente di rimuovere un attributo esistente.	Implementato
TS1O4.4.7	Viene verificato che il sistema permetta all'utente di inserire un nuovo metodo in una classe esistente.	Implementato
TS1O4.4.8	Viene verificato che il sistema permetta all'utente di modificare un metodo di una classe esistente.	Implementato
TS1O4.4.8.1	Viene verificato che il sistema permetta all'utente di modificare la visibilità di un metodo esistente.	Implementato
TS1O4.4.8.2	Viene verificato che il sistema permetta all'utente di modificare il nome di un metodo esistente.	Implementato
TS1O4.4.8.3	Viene verificato che il sistema permetta all'utente di modificare il tipo di ritorno di un metodo esistente.	Implementato
TS1O4.4.8.4	Viene verificato che il sistema permetta all'utente di inserire un nuovo parametro in un metodo esistente.	Implementato
TS1O4.4.8.5	Viene verificato che il sistema permetta all'utente di modificare un parametro di un metodo esistente.	Implementato
TS1O4.4.8.5.1	Viene verificato che il sistema permetta all'utente di modificare il tipo di un parametro esistente.	Implementato
TS1O4.4.8.5.2	Viene verificato che il sistema permetta all'utente di modificare il nome di un parametro esistente.	Implementato
TS1O4.4.8.5.3	Viene verificato che il sistema permetta all'utente di modificare il valore di default di un parametro esistente.	Implementato

TS1O4.4.8.6	Viene verificato che il sistema permetta all'utente di rimuovere un parametro esistente.	Implementato
TS1O4.4.9	Viene verificato che il sistema permetta all'utente di rimuovere un metodo esistente.	Implementato
TS1O4.4.10	Viene verificato che il sistema permetta all'utente di modificare lo stereotipo di una classe esistente.	Implementato
TS1O4.5	Viene verificato che il sistema permetta all'utente di rimuovere una classe esistente.	Implementato
TS1O4.5.1	Viene verificato che il sistema visualizzi un messaggio di errore nel caso in cui l'eliminazione di una classe non fosse possibile.	Non implementato
TS1O4.6.1	Viene verificato che il sistema permetta all'utente di inserire una nuova relazione di associazione nel diagramma delle classi.	Implementato
TS1O4.6.2	Viene verificato che il sistema permetta all'utente di inserire una nuova relazione di dipendenza nel diagramma delle classi.	Implementato
TS1O4.6.3	Viene verificato che il sistema permetta all'utente di inserire una nuova relazione di composizione nel diagramma delle classi.	Implementato
TS1O4.6.4	Viene verificato che il sistema permetta all'utente di inserire una nuova relazione di aggregazione nel diagramma delle classi.	Implementato
TS1O4.6.5	Viene verificato che il sistema permetta all'utente di inserire una nuova relazione di generalizzazione nel diagramma delle classi.	Implementato
TS1O4.7.1	Viene verificato che il sistema permetta all'utente di modificare la cardinalità di una relazione esistente.	Implementato
TS1O4.7.2	Viene verificato che il sistema permetta all'utente di modificare la classe di partenza di una relazione esistente.	Implementato
TS1O4.7.3	Viene verificato che il sistema permetta all'utente di modificare la classe di arrivo di una relazione esistente.	Implementato
TS1O4.8	Viene verificato che il sistema permetta all'utente di rimuovere una relazione esistente.	Implementato
TS1D4.9	Viene verificato che il sistema permetta all'utente di inserire un nuovo commento nel diagramma delle classi.	Implementato
TS1D4.10	Viene verificato che il sistema permetta all'utente di modificare il testo di un commento esistente.	Implementato

TS1D4.11	Viene verificato che il sistema permetta all'utente di rimuovere un commento esistente.	Implementato
TS1D4.12	Viene verificato che il sistema permetta all'utente di espandere una classe ridotta.	Implementato
TS1D4.13	Viene verificato che il sistema permetta all'utente di ridurre una classe espansa.	Implementato
TS1D4.14	Viene verificato che il sistema permetta all'utente di espandere un commento ridotto.	Implementato
TS1D4.15	Viene verificato che il sistema permetta all'utente di ridurre un commento espanso.	Implementato
TS1D4.16	Viene verificato che il sistema permetta all'utente di applicare un filtro alla visualizzazione delle classi basato sulla loro priorità.	Implementato
TS1D4.17	Viene verificato che il sistema permetta all'utente di spostare un blocco esistente all'interno del diagramma delle classi.	Implementato
TS1O5	Viene verificato che il sistema permetta all'utente di visualizzare la schermata di creazione del diagramma delle attività.	Implementato
TS1O5.1	Viene verificato che il sistema permetta all'utente di visualizzare il diagramma delle attività relativo ad un metodo.	Implementato
TS1O5.2.1.1	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco variabile.	Implementato
TS1O5.2.1.1.1	Viene verificato che il sistema permetta all'utente di collegare una variabile esistente ad un blocco variabile esistente.	Implementato
TS1O5.2.1.2	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco di chiamata di un metodo.	Implementato
TS1O5.2.1.3	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco ciclo.	Implementato
TS1O5.2.1.4	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco if/else.	Implementato
TS1O5.2.1.5	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco operatore.	Implementato
TS1O5.2.1.6	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco avanzamento.	Implementato
TS1O5.2.1.7	Viene verificato che il sistema permetta all'utente di inserire un nuovo blocco jolly.	Implementato
TS1D5.2.2.1	Viene verificato che il sistema permetta all'utente di modificare un blocco variabile esistente.	Implementato

TS1D5.2.2.1.1	Viene verificato che il sistema permetta all'utente di modificare il tipo di un blocco variabile esistente.	Implementato
TS1D5.2.2.1.2	Viene verificato che il sistema permetta all'utente di modificare l'inizializzazione di un blocco variabile esistente.	Implementato
TS1O5.2.2.2	Viene verificato che il sistema permetta all'utente di modificare un blocco di chiamata di un metodo esistente.	Implementato
TS1O5.2.2.2.1	Viene verificato che il sistema permetta all'utente di modificare l'invocazione del metodo di un blocco di chiamata di un metodo esistente.	Implementato
TS1O5.2.2.2.2	Viene verificato che il sistema permetta all'utente di modificare i parametri di invocazione del metodo di un blocco di chiamata di un metodo esistente.	Implementato
TS1O5.2.2.3	Viene verificato che il sistema permetta all'utente di modificare un blocco ciclo esistente.	Implementato
TS1O5.2.2.3.1	Viene verificato che il sistema permetta all'utente di modificare la condizione di un blocco ciclo esistente.	Implementato
TS1O5.2.2.3.2	Viene verificato che il sistema permetta all'utente di modificare il corpo di un blocco ciclo esistente.	Implementato
TS1O5.2.2.4	Viene verificato che il sistema permetta all'utente di modificare un blocco if/else esistente.	Implementato
TS1O5.2.2.4.1	Viene verificato che il sistema permetta all'utente di modificare la condizione di un blocco if/else esistente.	Implementato
TS1O5.2.2.4.2	Viene verificato che il sistema permetta all'utente di modificare il corpo del ramo if di un blocco if/else esistente.	Implementato
TS1O5.2.2.4.3	Viene verificato che il sistema permetta all'utente di modificare il corpo del ramo else di un blocco if/else esistente.	Implementato
TS1O5.2.2.5	Viene verificato che il sistema permetta all'utente di modificare un blocco operatore esistente.	Implementato
TS1O5.2.2.5.1	Viene verificato che il sistema permetta all'utente di modificare l'operatore di un blocco operatore esistente.	Implementato
TS1O5.2.2.5.2	Viene verificato che il sistema permetta all'utente di modificare l'elemento a sinistra di un blocco operatore esistente.	Implementato
TS1O5.2.2.5.3	Viene verificato che il sistema permetta all'utente di modificare l'elemento a destra di un blocco operatore esistente.	Implementato

TS1O5.2.2.5.4	Viene verificato che il sistema permetta all'utente di modificare l'elemento a cui viene assegnato il risultato di un blocco operatore esistente.	Implementato
TS1O5.2.2.6	Viene verificato che il sistema permetta all'utente di modificare un blocco avanzamento esistente.	Implementato
TS1O5.2.2.6.1	Viene verificato che il sistema permetta all'utente di modificare il blocco di partenza di un blocco avanzamento esistente.	Implementato
TS1O5.2.2.6.2	Viene verificato che il sistema permetta all'utente di modificare il blocco di arrivo di un blocco avanzamento esistente.	Implementato
TS1O5.2.2.7	Viene verificato che il sistema permetta all'utente di modificare il corpo di un blocco jolly esistente.	Implementato
TS1D5.2.3.1	Viene verificato che il sistema permetta all'utente di rimuovere un blocco variabile esistente.	Implementato
TS1O5.2.3.2	Viene verificato che il sistema permetta all'utente di rimuovere un blocco di chiamata di un metodo esistente.	Implementato
TS1O5.2.3.3	Viene verificato che il sistema permetta all'utente di rimuovere un blocco ciclo esistente.	Implementato
TS1O5.2.3.4	Viene verificato che il sistema permetta all'utente di rimuovere un blocco if/else esistente.	Implementato
TS1O5.2.3.5	Viene verificato che il sistema permetta all'utente di rimuovere un blocco operatore esistente.	Implementato
TS1O5.2.3.6	Viene verificato che il sistema permetta all'utente di rimuovere un blocco avanzamento esistente.	Implementato
TS1O5.2.3.7	Viene verificato che il sistema permetta all'utente di rimuovere un blocco jolly esistente.	Implementato
TS1D5.2.4.1	Viene verificato che il sistema permetta all'utente di espandere un blocco variabile ridotto.	Implementato
TS1D5.2.4.2	Viene verificato che il sistema permetta all'utente di espandere un blocco chiamata di un metodo ridotto.	Implementato
TS1D5.2.4.3	Viene verificato che il sistema permetta all'utente di espandere un blocco ciclo ridotto.	Implementato
TS1D5.2.4.4	Viene verificato che il sistema permetta all'utente di espandere un blocco if/else ridotto.	Implementato
TS1D5.2.4.5	Viene verificato che il sistema permetta all'utente di espandere un blocco jolly ridotto.	Implementato
TS1D5.2.5.1	Viene verificato che il sistema permetta all'utente di ridurre un blocco variabile espanso.	Implementato
TS1D5.2.5.2	Viene verificato che il sistema permetta all'utente di ridurre un blocco chiamata di un metodo espanso.	Implementato

TS1D5.2.5.3	Viene verificato che il sistema permetta all'utente di ridurre un blocco ciclo espanso.	Implementato
TS1D5.2.5.4	Viene verificato che il sistema permetta all'utente di ridurre un blocco if/else espanso.	Implementato
TS1D5.2.5.5	Viene verificato che il sistema permetta all'utente di ridurre un blocco jolly espanso.	Implementato
TS1O5.2.6	Viene verificato che il sistema permetta all'utente di spostare un blocco esistente all'interno del diagramma delle attività.	Implementato
TS1O5.3	Viene verificato che il sistema permetta all'utente di rimuovere un diagramma delle attività esistente.	Implementato
TS1O5.4	Viene verificato che il sistema permetta all'utente di caricare un metodo dalla libreria offerta dal sistema.	Non implementato
TS1O6	Viene verificato che il sistema permetta all'utente di generare codice a partire dai diagrammi.	Implementato
TS1O6.1	Viene verificato che il sistema visualizzi un messaggio di errore nel caso in cui la generazione del codice non fosse andata a buon fine.	Non implementato
TS1D6.2	Viene verificato che il sistema permetta all'utente di modificare tramite editor online il codice generato.	Implementato
TS1O6.3	Viene verificato che il sistema permetta all'utente di esportare il codice generato.	Non implementato
TS1O7	Viene verificato che il sistema permetta all'utente di salvare il progetto aperto.	Non implementato
TS1O7.1	Viene verificato che il sistema permetta all'utente di esportare il progetto come file.	Non implementato
TS1O9	Viene verificato che il sistema permetta all'utente di produrre il corpo delle classi e dei metodi nel linguaggio Java a partire dai diagrammi delle classi e delle attività.	Implementato
TS1D15	Viene verificato che il sistema permetta all'utente di realizzare un programma di una certa consistenza utilizzando il prodotto stesso.	Non implementato
TS4O5	Viene verificato che il sistema funzioni correttamente con versioni 50.x o superiori di <i>Chrome_G</i> .	Implementato
TS4O6	Viene verificato che il sistema funzioni correttamente con versioni 48.x o superiori di <i>Firefox_G</i> .	Implementato
TS4O7	Viene verificato che il sistema funzioni correttamente con versioni 9.1 o superiori di <i>Safari_G</i> .	Implementato

Tabella 14: Test di Sistema

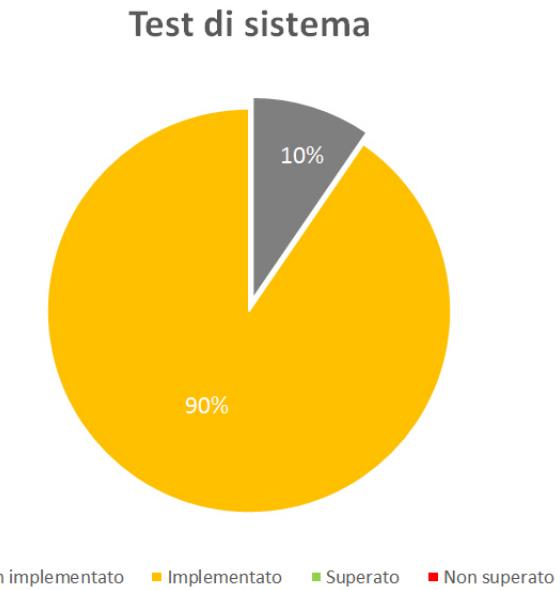


Figura 3: Test di sistema - stato corrente

D.2.1 Tracciamento test di sistema-requisiti

Test	Requisito
TS1O2	R1O2
TS1O2.1	R1O2.1
TS1O3.2	R1O3.2
TS1O3.2.1	R1O3.2.1
TS1O4	R1O4
TS1O4.1	R1O4.1
TS1D4.2	R1D4.2
TS1O4.3	R1O4.3
TS1O4.4	R1O4.4
TS1O4.4.1	R1O4.4.1
TS1O4.4.2	R1O4.4.2
TS1O4.4.3	R1O4.4.3
TS1O4.4.4	R1O4.4.4
TS1O4.4.5	R1O4.4.5
TS1O4.4.5.1	R1O4.4.5.1
TS1O4.4.5.2	R1O4.4.5.2
TS1O4.4.5.3	R1O4.4.5.3
TS1O4.4.5.4	R1O4.4.5.4
TS1O4.4.6	R1O4.4.6

TS1O4.4.7	R1O4.4.7
TS1O4.4.8	R1O4.4.8
TS1O4.4.8.1	R1O4.4.8.1
TS1O4.4.8.2	R1O4.4.8.2
TS1O4.4.8.3	R1O4.4.8.3
TS1O4.4.8.4	R1O4.4.8.4
TS1O4.4.8.5	R1O4.4.8.5
TS1O4.4.8.5.1	R1O4.4.8.5.1
TS1O4.4.8.5.2	R1O4.4.8.5.2
TS1O4.4.8.5.3	R1O4.4.8.5.3
TS1O4.4.8.6	R1O4.4.8.6
TS1O4.4.9	R1O4.4.9
TS1O4.4.10	R1O4.4.10
TS1O4.5	R1O4.5
TS1O4.5.1	R1O4.5.1
TS1O4.6.1	R1O4.6.1
TS1O4.6.2	R1O4.6.2
TS1O4.6.3	R1O4.6.3
TS1O4.6.4	R1O4.6.4
TS1O4.6.1	R1O4.6.5
TS1O4.7.1	R1O4.7.1
TS1O4.7.2	R1O4.7.2
TS1O4.7.3	R1O4.7.3
TS1O4.8	R1O4.8
TS1D4.9	R1D4.9
TS1D4.10	R1D4.10
TS1D4.11	R1D4.11
TS1D4.12	R1D4.12
TS1D4.13	R1D4.13
TS1D4.14	R1D4.14
TS1D4.15	R1D4.15
TS1D4.16	R1D4.16
TS1D4.17	R1D4.17
TS1O5	R1O5
TS1O5.1	R1O5.1
TS1O5.2.1.1	R1O5.2.1.1
TS1O5.2.1.1.1	R1O5.2.1.1.1
TS1O5.2.1.2	R1O5.2.1.2
TS1O5.2.1.3	R1O5.2.1.3
TS1O5.2.1.4	R1O5.2.1.4
TS1O5.2.1.5	R1O5.2.1.5

TS1O5.2.1.6	R1O5.2.1.6
TS1O5.2.1.7	R1O5.2.1.7
TS1D5.2.2.1	R1D5.2.2.1
TS1D5.2.2.1.1	R1D5.2.2.1.1
TS1D5.2.2.1.2	R1D5.2.2.1.2
TS1O5.2.2.2	R1O5.2.2.2
TS1O5.2.2.2.1	R1O5.2.2.2.1
TS1O5.2.2.2.2	R1O5.2.2.2.2
TS1O5.2.2.2.3	R1O5.2.2.2.3
TS1O5.2.2.3.1	R1O5.2.2.3.1
TS1O5.2.2.3.2	R1O5.2.2.3.2
TS1O5.2.2.4	R1O5.2.2.4
TS1O5.2.2.4.1	R1O5.2.2.4.1
TS1O5.2.2.4.2	R1O5.2.2.4.2
TS1O5.2.2.4.3	R1O5.2.2.4.3
TS1O5.2.2.5	R1O5.2.2.5
TS1O5.2.2.5.1	R1O5.2.2.5.1
TS1O5.2.2.5.2	R1O5.2.2.5.2
TS1O5.2.2.5.3	R1O5.2.2.5.3
TS1O5.2.2.5.4	R1O5.2.2.5.4
TS1O5.2.2.6	R1O5.2.2.6
TS1O5.2.2.6.1	R1O5.2.2.6.1
TS1O5.2.2.6.2	R1O5.2.2.6.2
TS1O5.2.2.7	R1O5.2.2.7
TS1D5.2.3.1	R1D5.2.3.1
TS1D5.2.3.2	R1D5.2.3.2
TS1D5.2.3.3	R1D5.2.3.3
TS1D5.2.3.4	R1D5.2.3.4
TS1D5.2.3.5	R1D5.2.3.5
TS1D5.2.3.6	R1D5.2.3.6
TS1D5.2.3.7	R1D5.2.3.7
TS1D5.2.4.1	R1D5.2.4.1
TS1D5.2.4.2	R1D5.2.4.2
TS1D5.2.4.3	R1D5.2.4.3
TS1D5.2.4.4	R1D5.2.4.4
TS1D5.2.4.5	R1D5.2.4.5
TS1D5.2.5.1	R1D5.2.5.1
TS1D5.2.5.2	R1D5.2.5.2
TS1D5.2.5.3	R1D5.2.5.3
TS1D5.2.5.4	R1D5.2.5.4
TS1D5.2.5.5	R1D5.2.5.5

TS1O5.2.6	R1O5.2.6
TS1O5.3	R1O5.3
TS1O5.4	R1O5.4
TS1O6	R1O6
TS1O6.1	R1O6.1
TS1O6.2	R1O6.2
TS1O6.3	R1O6.3
TS1O7	R1O7
TS1O7.1	R1O7.1
TS1O9	R1O9
TS1D15	R1D15
TS4O5	R4O5
TS4O6	R4O6
TS4O7	R4O7

D.3 Test di Integrazione

Tale tipologia di test serve per verificare che le varie componenti del sistema software interagiscano tra loro nel modo atteso. La strategia seguita per definire questi test è stata di tipo bottom-up, in modo da poter realizzare il prodotto partendo dalle singole componenti e realizzare in questo modo le varie funzionalità in ordine di importanza. Per ogni test viene specificato il proprio codice univoco, la descrizione e lo stato di implementazione attuale.

Codice	Descrizione	Stato
TI1	Test d'integrazione finale tra l'applicazione Front-end e l'applicazione Back-end.	Implementato
TI2	Test d'integrazione fra le componenti Back-end::DataTier, Back-end::ApplicationTier e Back-end::PresentationTier.	Implementato
TI3	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Back-end::DataTier, in particolare che venga gestita correttamente l'interazione con il controller di Back-end::ApplicationTier e MongoDB _G .	Implementato
TI4	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Back-end::ApplicationTier, in particolare che venga gestita correttamente l'interazione con il template _G del Back-end::DataTier e il Back-end::PresentationTier::Middleware.	Implementato

TI5	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::PresentationTier</code> , in particolare che venga gestita correttamente l'interazione con il controller di <code>Back-end::ApplicationTier</code> , <code>Back-end::ApplicationTier::Error</code> e con Express.	Superato
TI6	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::ApplicationTier::Generator</code> , in particolare che venga gestita correttamente l'interazione con il controller di <code>Back-end::ApplicationTier</code> e <code>Back-end::ApplicationTier::Error</code> .	Implementato
TI7	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::ApplicationTier::Error</code> , in particolare che venga gestita correttamente l'interazione con <code>Back-end::ApplicationTier::Generator</code> e <code>Back-end::PresentationTier::Middleware</code> .	Superato
TI8	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::ApplicationTier::JavaGenerator</code> , in particolare che venga gestita correttamente l'interazione con l'interfaccia <code>BaseGenerator</code> di <code>Back-end::ApplicationTier::Generator</code> .	Implementato
TI9	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::ApplicationTier::JavaGenerator</code> <code>::ClassDiaJavaGenerator</code> , in particolare che venga gestita correttamente l'interazione con <code>Back-end::ApplicationTier::Generator</code> <code>::JavaGenerator</code> e con <code>Back-end::ApplicationTier::JavaGenerator</code> <code>::ActivityDiaJavaGenerator</code> .	Implementato
TI10	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Back-end::ApplicationTier::JavaGenerator</code> <code>::ActivityDiaJavaGenerator</code> , in particolare che venga gestita correttamente l'interazione con <code>Back-end::ApplicationTier::JavaGenerator</code> <code>::ClassDiaJavaGenerator</code> .	Superato

TI11	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Back-end::PresentationTier::Middleware , in particolare che venga gestita correttamente l'interazione con il controller di Back-end::ApplicationTier , Back-end::PresentationTier::Controller , Back-end::ApplicationTier::Error ed Express.	Superato
TI12	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Back-end::PresentationTier::Controller , in particolare che venga gestita correttamente l'interazione con Back-end::PresentationTier::Middleware ed Express.	Superato
TI13	Test d'integrazione fra le componenti Front-end::Model , Front-end::View e Front-end::ModelView .	Implementato
TI14	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Front-end::View , in particolare che venga gestita correttamente l'interazione con i controller del Front-end::ViewModel , con <i>GoJS_G</i> e con <i>Angular2_G</i> .	Implementato
TI15	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Front-end::ViewModel , in particolare che venga gestita correttamente l'interazione con Front-end::View , con Front-End::Model e Angular2.	Implementato
TI16	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Front-end::Model , in particolare che venga gestita correttamente l'interazione con i controller del Front-end::ViewModel , con GoJS e con Angular2.	Implementato
TI17	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Front-end::Model::Services , in particolare che venga gestita correttamente l'interazione con i controller del Front-end::Model::Commands .	Implementato
TI18	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a Front-end::Model::Objects , in particolare che venga gestita correttamente l'interazione con i controller del Front-end::ViewModel e con GoJS.	Implementato

TI19	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Front-end::Model::Commands</code> , in particolare che venga gestita correttamente l'interazione con i controller del <code>Front-end::ViewModel</code> e con Angular2.	Implementato
TI20	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Front-end::Model::Objects::ClassObjects</code> , in particolare che venga gestita correttamente l'interazione con <code>Front-end::Model::Objects</code> .	Superato
TI21	Viene verificato che il sistema gestisca correttamente l'integrazione tra le componenti relative a <code>Front-end::Model::Objects::ActivityObjects</code> , in particolare che venga gestita correttamente l'interazione con <code>Front-end::Model::Objects</code> .	Superato

Tabella 16: Test di Integrazione

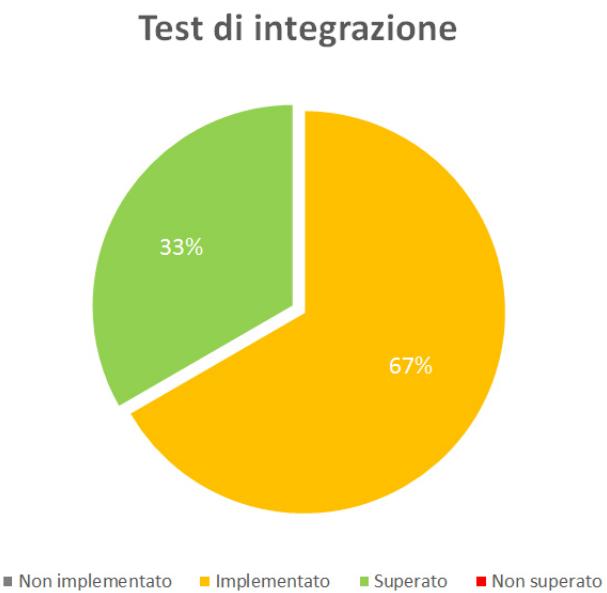


Figura 4: Test di integrazione - stato corrente

D.3.1 Tracciamento test di integrazione-componenti

Test	Componente
TI1	SWEDesigner
TI2	Back-end
TI3	Back-end::DataTier
TI4	Back-end::ApplicationTier
TI5	Back-end::PresentationTier
TI6	Back-end::ApplicationTier::Generator
TI7	Back-end::ApplicationTier::Error
TI8	Back-end::ApplicationTier::JavaGenerator
TI9	Back-end::ApplicationTier::JavaGenerator ::ClassDiaJavaGenerator
TI10	Back-end::ApplicationTier::JavaGenerator ::ActivityDiaJavaGenerator
TI11	Back-end::PresentationIter::Middleware
TI12	Back-end::PresentationIter::Controller
TI13	Front-end
TI14	Front-end::View
TI15	Front-end::ViewModel
TI16	Front-end::Model
TI17	Front-end::Model::Services
TI18	Front-end::Model::Objects
TI19	Front-end::Model::Commands
TI20	Front-end::Model::Objects::ClassObjects
TI21	Front-end::Model::Objects::ActivityObjects

D.4 Test di Unità

Tale tipologia di test serve per testare il corretto funzionamento delle singole unità, ossia delle più piccole componenti software singolarmente verificabili. Solitamente l'unità trova corrispondenza in un metodo di una classe tra quelli descritti nel documento *Definizione di Prodotto*.

Per ogni test viene specificato il proprio codice univoco, la descrizione e lo stato di implementazione attuale.

Codice	Descrizione	Stato
TU1	Viene verificato che la richiesta di creazione di un nuovo progetto vada a buon fine.	Superato
TU2	Viene verificato che la richiesta di caricamento di un progetto vada a buon fine.	Implementato

TU3	Viene verificato che la richiesta di salvataggio del progetto attualmente aperto termini con il download di un file $JSON_G$.	Implementato
TU4	Viene verificato che avvenga il corretto caricamento di un diagramma delle classi a partire da un JSON.	Superato
TU5	Viene verificato che venga restituito il JSON relativo diagramma delle classi d'interesse.	Superato
TU6	Viene verificato che avvenga il corretto caricamento di un diagramma delle attività a partire da un JSON.	Superato
TU7	Viene verificato che venga restituito il JSON relativo diagramma delle attività d'interesse.	Superato
TU8	Viene verificato che gli oggetti DiagramPalettes vengano istanziati correttamente nel diagramma di riferimento.	Superato
TU9	Viene verificata la correttezza della lista delle classi ritornata dalla libreria.	Superato
TU10	Viene verificata la correttezza della lista dei design pattern ritornata dalla libreria.	Superato
TU11	Viene verificato che il passaggio tra libreria e blocchi all'interno del diagramma delle attività avvenga correttamente.	Superato
TU12	Viene verificata la correttezza della lista dei metodi ritornata dalla libreria.	Superato
TU13	Viene verificato che la richiesta di generazione del codice vada a buon fine.	Superato
TU14	Viene verificato che la richiesta di esportazione di codice generato termini con il download di un file.	Implementato
TU15	Viene verificato che all'apertura del diagramma delle classi vengano visualizzati il diagramma delle classi e le palette ad esso relativa.	Superato
TU16	Viene verificato che all'apertura del diagramma delle attività vengano visualizzati il diagramma delle attività e le palette ad esso relativa.	Superato
TU17	Viene verificato che alla richiesta di uno specifico $oggetto_G$ di un diagramma venga restituito il JSON ad esso corrispondente.	Superato
TU18	Viene verificato che vengano aggiornate le informazioni relative ad uno specifico oggetto nel diagramma a partire da un JSON.	Superato
TU19	Viene verificato che alla richiesta di uno specifico oggetto di una palette venga restituito il JSON ad esso corrispondente.	Superato

TU20	Viene verificato che vengano aggiornate le informazioni relative ad uno specifico oggetto nella paletta a partire da un JSON.	Superato
TU21	Viene verificato che il codice generato venga correttamente restituito all'editor di codice.	Superato
TU22	Viene verificato che il template richiesto venga correttamente restituito al diagramma.	Implementato
TU23	Viene verificato che all'inserimento di un oggetto Class all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU24	Viene verificato che l'aggiunta di un metodo ad una classe avvenga correttamente.	Superato
TU25	Viene verificato che la rimozione di un metodo da una classe avvenga correttamente.	Superato
TU26	Viene verificato che l'aggiunta di un attributo ad una classe avvenga correttamente.	Superato
TU27	Viene verificato che la rimozione di un attributo da una classe avvenga correttamente.	Superato
TU28	Viene verificato che il cambio di priorità di una classe avvenga correttamente.	Superato
TU29	Viene verificato che il cambio di colore di una classe avvenga correttamente.	Superato
TU30	Viene verificato che il cambio di visibilità di un attributo di una classe avvenga correttamente.	Superato
TU31	Viene verificato che il cambio di visibilità di un metodo di una classe avvenga correttamente.	Superato
TU32	Viene verificato che il cambio di stereotipo di una classe avvenga correttamente.	Superato
TU33	Viene verificato che al cambio di stereotipo di una classe l'opacità dello stereotipo venga aggiornata.	Superato
TU34	Viene verificato che al cambio di priorità di una classe la sua opacità venga aggiornata.	Superato
TU35	Viene verificato che all'inserimento di un oggetto Comment all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU36	Viene verificato che all'inserimento di un oggetto Composition all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU37	Viene verificato che all'inserimento di un oggetto Dependency all'interno del diagramma delle classi venga costruito correttamente.	Superato

TU38	Viene verificato che all'inserimento di un oggetto Association all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU39	Viene verificato che all'inserimento di un oggetto Aggregation all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU40	Viene verificato che all'inserimento di un oggetto Generalization all'interno del diagramma delle classi venga costruito correttamente.	Superato
TU41	Viene verificato che il nodo di partenza di una relazione sia correttamente associato ad essa.	Superato
TU42	Viene verificato che il nodo di arrivo di una relazione sia correttamente associato ad essa.	Superato
TU43	Viene verificato che l'inserimento della cardinalità di partenza di una relazione vada a buon fine.	Superato
TU44	Viene verificato che l'inserimento della cardinalità di arrivo di una relazione vada a buon fine.	Superato
TU45	Viene verificato che durante l'inserimento di un blocco venga illuminata l'area di inserimento prevista per il blocco.	Superato
TU46	Viene verificato che un blocco venga correttamente inserito all'interno di un altro blocco.	Superato
TU47	Viene verificato che un blocco ridotto venga correttamente espanso.	Superato
TU48	Viene verificato che un blocco espanso venga correttamente ridotto.	Superato
TU49	Viene verificato che all'inserimento di un oggetto VariableObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU50	Viene verificato che all'inserimento di un oggetto MethodCallObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU51	Viene verificato che all'inserimento di un oggetto CycleObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU52	Viene verificato che all'inserimento di un oggetto IfElseObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU53	Viene verificato che all'inserimento di un oggetto OperatorObj all'interno del diagramma delle attività venga costruito correttamente.	Superato

TU54	Viene verificato che all'inserimento di un oggetto StepObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU55	Viene verificato che all'inserimento di un oggetto JollyObj all'interno del diagramma delle attività venga costruito correttamente.	Superato
TU56	Viene verificato che l'aggiunta di un parametro ad un blocco chiamata metodo avvenga correttamente.	Superato
TU57	Viene verificato che la rimozione di un parametro da un blocco chiamata metodo avvenga correttamente.	Superato
TU58	Viene verificato che il cambio del tipo di operatore di un blocco operatore avvenga correttamente	Superato
TU59	Viene verificato che il codice generato sia correttamente restituito in un array	Superato
TU60	Viene verificato che sia correttamente ritornato il JSON di un template di libreria richiesto	Implementato
TU61	Viene verificato che i vari componenti della classe Middleware siano correttamente inizializzati.	Superato
TU62	Viene verificato che gli errori previsti siano correttamente gestiti.	Superato
TU63	Viene verificato che le richieste di routing siano correttamente gestite.	Superato
TU64	Viene verificato che le situazioni in cui viene richiesto un oggetto dal Server, ed esso non viene trovato, vengano gestite correttamente.	Superato
TU65	Viene verificato che all'apertura di SWEDesigner venga ritornata la pagina index.html .	Superato
TU66	Viene verificato che alla richiesta di un template esso venga ritornato correttamente dal <i>database_G</i> .	Implementato
TU67	Viene verificato che il database restituisca correttamente la lista dei template di una determinata categoria.	Implementato
TU68	Viene verificato che il codice generato venga restituito correttamente partendo dal JSON contenente i diagrammi.	Superato
TU69	Viene verificato che la generazione del codice in linguaggio Java avvenga correttamente.	Superato
TU70	Viene verificato che la generazione del codice in linguaggio Java del diagramma delle classi avvenga correttamente.	Superato

TU71	Viene verificato che la generazione del codice in linguaggio Java di una classe con i relativi attributi e metodi avvenga correttamente.	Superato
TU72	Viene verificato che il codice in linguaggio Java degli attributi di una classe venga generato correttamente.	Superato
TU73	Viene verificato che il codice in linguaggio Java dei metodi di una classe venga generato correttamente.	Superato
TU74	Viene verificato che il codice in linguaggio Java delle relazioni tra classi venga generato correttamente.	Superato
TU75	Viene verificato che il JSON relativo al padre di una classe sottotipo venga ritornato correttamente.	Superato
TU76	Viene verificato che un commento del diagramma delle classi sia inserito correttamente all'interno del codice generato.	Implementato
TU77	Viene verificato che la generazione del codice in linguaggio Java del diagramma delle attività di un metodo avvenga correttamente.	Superato
TU78	Viene verificato che la generazione del codice di un blocco del diagramma delle attività, in linguaggio Java, di un metodo avvenga correttamente.	Superato
TU79	Viene verificato che per ogni livello di annodamento di blocchi venga ritornato un JSON che li contiene.	Superato
TU80	Viene verificato che i blocchi di un JSON relativo ad un diagramma delle attività vengano riordinati secondo l'ordine delle relazioni che li collegano.	Superato
TU81	Viene verificato che la ricerca di una relazione d'arrivo nelle classi avvenga correttamente.	Superato
TU82	Viene verificato che la ricerca di una relazione di partenza dalle classi avvenga correttamente.	Superato
TU83	Viene verificato che la ricerca di blocco nel diagramma delle attività avvenga correttamente.	Superato
TU84	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco variabile.	Superato
TU85	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco chiamata metodo.	Superato
TU86	Viene verificato che venga correttamente inserito il contenuto relativo ad un blocco jolly.	Superato
TU87	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco avanzamento.	Superato

TU88	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco ciclo.	Superato
TU89	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco operatore.	Superato
TU90	Viene verificato che venga correttamente generato il codice, in linguaggio Java, relativo ad un blocco if/else.	Superato
TU91	Viene verificato che l'errore venga ritornato in forma di JSON.	Superato
TU92	Viene verificato che la concatenazione dei campi dati dell'errore in formato String sia restituita correttamente.	Superato
TU93	Viene verificato che la conversione di un errore al tipo Error avvenga correttamente.	Superato

Tabella 18: Test di Unità

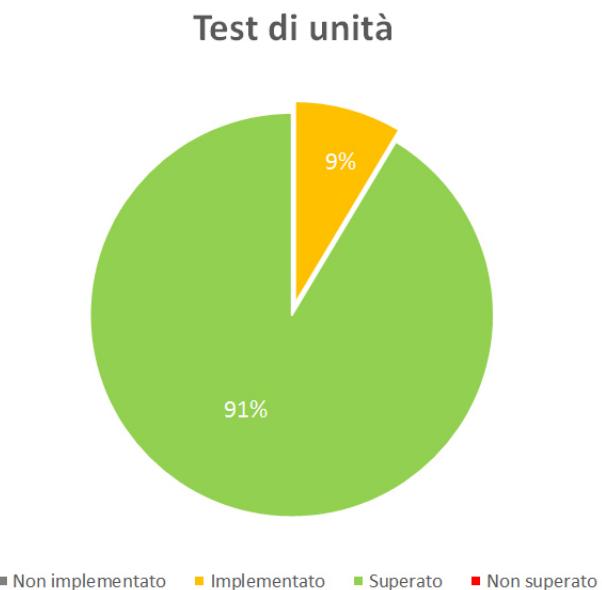


Figura 5: Test di unità - stato corrente

D.4.1 Tracciamento test di unità-metodi

Test	Metodo
TU1	<code>Front-end::View::SinglePageApp::newProject()</code>
TU2	<code>Front-end::View::SinglePageApp::uploadProject()</code>
TU3	<code>Front-end::View::SinglePageApp::saveProject()</code>
TU4	<code>Front-end::View::DiagramEditor::ClassDiagramEditor::setDiagram()</code>
TU5	<code>Front-end::View::DiagramEditor::ClassDiagramEditor::getDiagram()</code>
TU6	<code>Front-end::View::DiagramEditor::ActivityDiagramEditor::setDiagram()</code>
TU7	<code>Front-end::View::DiagramEditor::ActivityDiagramEditor::getDiagram()</code>
TU8	<code>Front-end::View::DiagramPalette::connectEditor()</code>
TU9	<code>Front-end::View::DiagramPalette::ClassDiagramPalette::retrieveClassesList()</code> <code>Front-end::Model::Commands::Command::GetTemplate::getClassesList()</code>
TU10	<code>Front-end::View::DiagramPalette::ClassDiagramPalette::retrievePatternsList()</code> <code>Front-end::Model::Commands::Command::GetTemplate::getPatternsList()</code>
TU11	<code>Front-end::View::DiagramPalette::ActivityDiagramPalette::open()</code>
TU12	<code>Front-end::View::DiagramPalette::ActivityDiagramPalette::retrieveMethodsList()</code>
TU13	<code>Front-end::View::CodeEditor::generateCode()</code>
TU14	<code>Front-end::View::CodeEditor::exportCode()</code>

TU15	<code>Front-end::View::DiagramFactory::ClassFactory ::createDiagramEditor() Front-end::View::DiagramFactory::ClassFactory ::createDiagramPalette()</code>
TU16	<code>Front-end::View::DiagramFactory:: ActivityFactory::createDiagramEditor() Front-end::View::DiagramFactory:: ActivityFactory::createDiagramPalette()</code>
TU17	<code>Front-end::ViewModel::EditorObjController:: getEditorObj()</code>
TU18	<code>Front-end::ViewModel::EditorObjController:: setEditorObj()</code>
TU19	<code>Front-end::ViewModel::PaletteObjController:: getPaletteObj()</code>
TU20	<code>Front-end::ViewModel::PaletteObjController:: setPaletteObj()</code>
TU21	<code>Front-end::ViewModel::CodeCommandController:: requestCode()</code>
TU22	<code>Front-end::ViewModel:: PaletteCommandController::getTemplate()</code>
TU23	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::Class() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class:: printPriorityAdornment()</code>
TU24	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::addMethod() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::updateMethodCount()</code>
TU25	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::deleteMethod() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::updateMethodCount()</code>
TU26	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::addAttribute() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::updateAttributeCount()</code>

TU27	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::deleteAttribute() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::updateAttributeCount()</code>
TU28	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::changePriority()</code>
TU29	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::changeColorClass()</code>
TU30	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class:: changeVisibilityAttribute()</code>
TU31	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class:: changeVisibilityMethod()</code>
TU32	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class:: changeStereotypeClass()</code>
TU33	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::adaptStereotypeClass()</code>
TU34	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::refreshClassOpacity() Front-end::Model::Objects::ClassObjects:: classDiaObj::Class::globalChangeOpacity()</code>
TU35	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Comment::Comment()</code>
TU36	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Composition:: Composition()</code>
TU37	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Dependency:: Dependency()</code>
TU38	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Association:: Association()</code>
TU39	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Aggregation:: Aggregation()</code>

TU40	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Generalization:: Generalization()</code>
TU41	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::setStart()</code>
TU42	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::setEnd()</code>
TU43	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Composition:: setStartCardinality() Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Association:: setStartCardinality() Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Aggregation:: setStartCardinality()</code>
TU44	<code>Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Composition:: setEndCardinality() Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Association:: setEndCardinality() Front-end::Model::Objects::ClassObjects:: classDiaObj::Relation::Aggregation:: setEndCardinality()</code>
TU45	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::highlightGroup()</code>
TU46	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::finishDrop()</code>
TU47	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::expandObj()</code>
TU48	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::reduceObj()</code>
TU49	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::VariableObj::VariableObj()</code>

TU50	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::MethodCallObj:: MethodCallObj()</code>
TU51	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::CycleObj::CycleObj()</code>
TU52	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::IfElseObj::IfElseObj()</code>
TU53	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::OperatorObj::OperatorObj()</code>
TU54	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::StepObj::StepObj()</code>
TU55	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::JollyObj::JollyObj()</code>
TU56	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::MethodCallObj::addParam() Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::MethodCallObj:: updateParametersCount()</code>
TU57	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::MethodCallObj:: removeParam() Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::MethodCallObj:: updateParametersCount()</code>
TU58	<code>Front-end::Model::Objects::ActivityObjects:: ActivityDiaObj::OperatorObj:: changeOperatore()</code>
TU59	<code>Front-end::Model::Commands::Command:: RequestCode::getgeneratedCode() Front-end::Model::Commands::Command:: RequestCode::elaborate() Front-end::Model::Services::ServerConnector:: getgeneratedCode()</code>

TU60	<pre> Front-end::Model::Commands::Command:: GetTemplate::getTemplate() Front-end::Model::Commands::Command:: GetTemplate::elaborate() Front-end::Model::Services::ServerConnector:: getTemplate() </pre>
TU61	<pre> Back-end::PresentationTier::Middleware:: MiddlewareLoader::app() </pre>
TU62	<pre> Back-end::PresentationTier::Middleware:: Middleware::ErrorHandler::handler() </pre>
TU63	<pre> Back-end::PresentationTier::Middleware:: Router::handler() </pre>
TU64	<pre> Back-end::PresentationTier::Middleware:: NotFoundHandler::handler() </pre>
TU65	<pre> Back-end::PresentationTier::Controller:: getIndex() </pre>
TU66	<pre> Back-end::ApplicationTier:: ApplicationController::getTemplate() Back-end::DataTier::Template::getTemplate() </pre>
TU67	<pre> Back-end::ApplicationTier:: ApplicationController::getCategoria() Back-end::DataTier::Template::getCategoria() </pre>
TU68	<pre> Back-end::ApplicationTier:: ApplicationController::getGeneratedCode() Back-end::ApplicationTier::Generator:: BaseGenerator::getGeneratedCode() </pre>
TU69	<pre> Back-end::ApplicationTier::Generator:: JavaGenerator::JavaGenerator:: getGeneratedCode() </pre>
TU70	<pre> Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: ClassDiaJavaGenerator::getGeneratedCode() </pre>
TU71	<pre> Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: ClassJavaGenerator:: getGeneratedClassCode() </pre>

TU72	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: AttributeJavaGenerator:: getGeneratedAttributeCode()</code>
TU73	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: MethodJavaGenerator:: getGeneratedMethodCode()</code>
TU74	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: RelationJavaGenerator:: getGeneratedRelationCode()</code>
TU75	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: RelationJavaGenerator::getPadre() Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: RelationJavaGenerator::controllo()</code>
TU76	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ClassDiaJavaGenerator:: CommentJavaGenerator:: getGeneratedCommentCode()</code>
TU77	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: ActivityDiaJavaGenerator:: getGeneratedActivityCode()</code>
TU78	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator::scrittura()</code>
TU79	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator:: cernitaNodeData()</code>
TU80	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator::riordina()</code>

TU81	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator::cercaInRelTo()</code>
TU82	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator::cercaInRelFrom()</code>
TU83	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: InstructionJavaGenerator::cerca()</code>
TU84	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: VariableJavaGenerator:: getGeneratedVariableCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: VariableConnectJavaGenerator:: getGeneratedVariableCode()</code>
TU85	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: MethodCallJavaGenerator:: getGeneratedMethodClassCode()</code>
TU86	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: JollyJavaGenerator:: getGeneratedJollyCode()</code>
TU87	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: StepJavaGenerator::getGeneratedStepCode()</code>
TU88	<code>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: CycleJavaGenerator:: getGeneratedConditionCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: CycleJavaGenerator:: getGeneratedCycleCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: CycleJavaGenerator::getGeneratedBodyCode()</code>

TU89	<pre>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: OperatorJavaGenerator:: getGeneratedCycleCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: OperatorJavaGenerator:: getGeneratedSxCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: OperatorJavaGenerator:: getGeneratedDxCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: OperatorJavaGenerator:: getGeneratedOperationCode()</pre>
TU90	<pre>Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: IfElseJavaGenerator:: getGeneratedWrapCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: IfElseJavaGenerator::getGeneratedIfCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: IfElseJavaGenerator:: getGeneratedConditionCode() Back-end::ApplicationTier::Generator:: JavaGenerator::ActivityDiaJavaGenerator:: IfElseJavaGenerator:: getGeneratedElseCode()</pre>
TU91	<pre>Back-end::ApplicationTier::Error:: ErrorApplication::toJSON()</pre>
TU92	<pre>Back-end::ApplicationTier::Error:: ErrorApplication::toString()</pre>
TU93	<pre>Back-end::ApplicationTier::Error:: ErrorApplication::toError()</pre>

E Resoconto delle attività di verifica

In questa sezione vengono riportate le misurazioni relative agli obiettivi di qualità perseguiti dal gruppo, presentati nella sezione 3. Tali misurazioni vengono utilizzate come indice per evidenziare l'andamento dei processi e la qualità dei prodotti e le loro variazioni nel tempo, in modo da poter agire in un'ottica di miglioramento.

E.1 Processi

E.1.1 Sviluppo

E.1.1.1 MPC1 - Requirement Stability Index

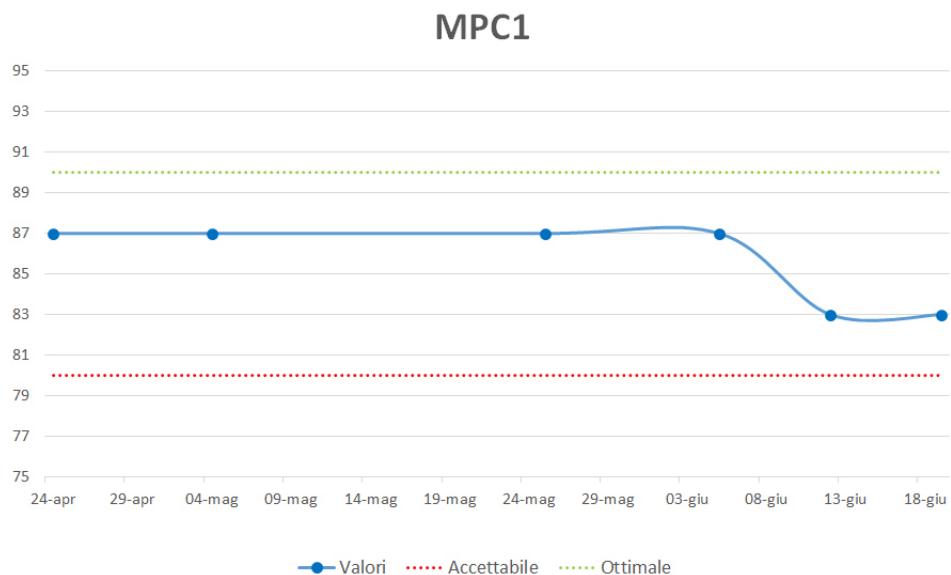


Figura 6: MPC1 - Serie storica

Le modifiche apportate ai requisiti nel tempo, ovviamente in accordo con il proponente, benché presenti non hanno raggiunto un livello eccessivamente elevato. Al momento sono ancora entro il range accettabile, lasciando quindi margine per piccole correzioni senza il rischio di superare tale soglia.

E.1.1.2 MPC2 - Instability

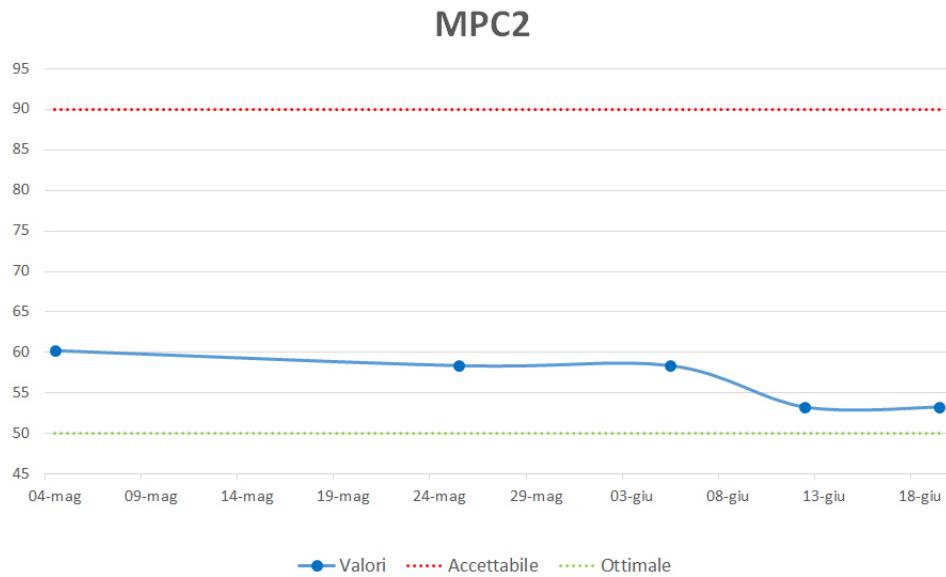


Figura 7: MPC2 - Serie storica

Nella creazione dell’architettura di sistema i *Progettisti* hanno perseguito un livello di instabilità largamente accettabile fin dall’inizio. Nel tempo grazie ad alcuni interventi migliorativi è sceso di valore. Il livello raggiunto allo stato attuale si avvicina all’ottimalità.

E.1.1.3 MPC3 - Violazioni dello stile di codifica

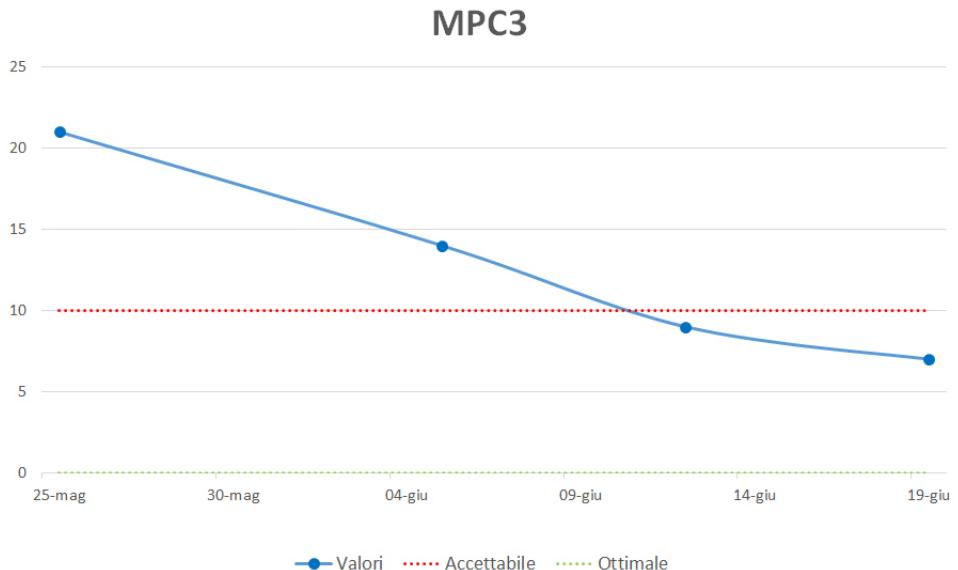


Figura 8: MPC3 - Serie storica

Inizialmente i *Programmatori* non avevano posto particolare attenzione alle norme riguardanti lo stile di codifica, prediligendo la velocità nella creazione della struttura cardine dell'applicazione per poter progredire con le attività successive. Nel tempo sono state tuttavia corrette tali inesattezze e attualmente il livello raggiunto è accettabile, con ancora margini di miglioramento per alcuni membri del gruppo.

E.1.1.4 Livello SPICE

Il livello complessivo a cui si attesta il processo di sviluppo è *Predicibile*. Questo perché il gruppo ha correttamente individuato tutti i processi di supporto e definito le misure del processo con obiettivi quantitativi fissati e una pianificazione dettagliata. Tali misure sono inoltre sfruttate per determinare variazioni nelle prestazioni del processo, individuarne le cause e attivare azioni correttive.

Non è ancora stato raggiunto il livello successivo poiché il livello di analisi in ottica di miglioramento non è ancora ottimale.

E.1.2 Documentazione

E.1.2.1 MPC4 - Errori frequenti nella documentazione

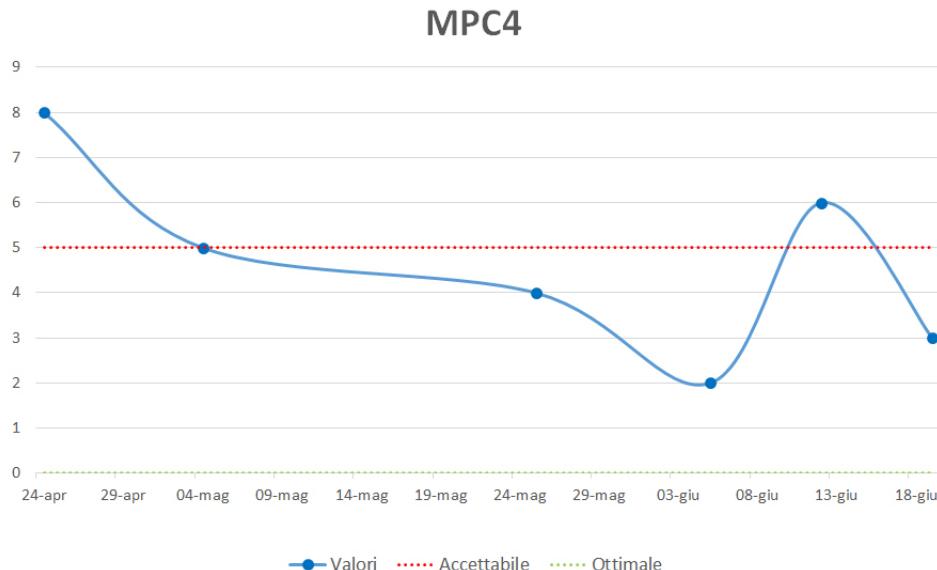


Figura 9: MPC4 - Serie storica

Nella realizzazione della documentazione gli errori frequenti, stilati in una lista di controllo nel documento *Norme di Progetto v3.0.0*, hanno avuto un andamento non costante. In particolare c'è stato un calo notevole il 5 giugno per via delle poche modifiche apportate ai documenti nella settimana antecedente. A ciò è seguito subito dopo un picco oltre il valore accettabile il 12 giugno, dovuto alla stesura di un nuovo documento la cui stesura non era ancora stata completata in data del riscontro. La tendenza generale tende comunque a scendere, con la possibilità di migliorare ulteriormente fino ad evitare del tutto gli errori frequenti.

E.1.2.2 Livello SPICE

Il livello complessivo a cui si attesta il processo di documentazione è *Predicibile*. Questo perché il gruppo ha correttamente individuato tutti i processi di supporto e definito le misure del processo con obiettivi quantitativi fissati e una pianificazione dettagliata. Tali misure sono inoltre sfruttate per determinare variazioni nelle prestazioni del processo, individuarne le cause e attivare azioni correttive.

Non è ancora stato raggiunto il livello successivo poiché il livello di analisi in ottica di miglioramento non è ancora ottimale.

E.1.3 Verifica

E.1.3.1 MPC5 - Test di unità eseguiti

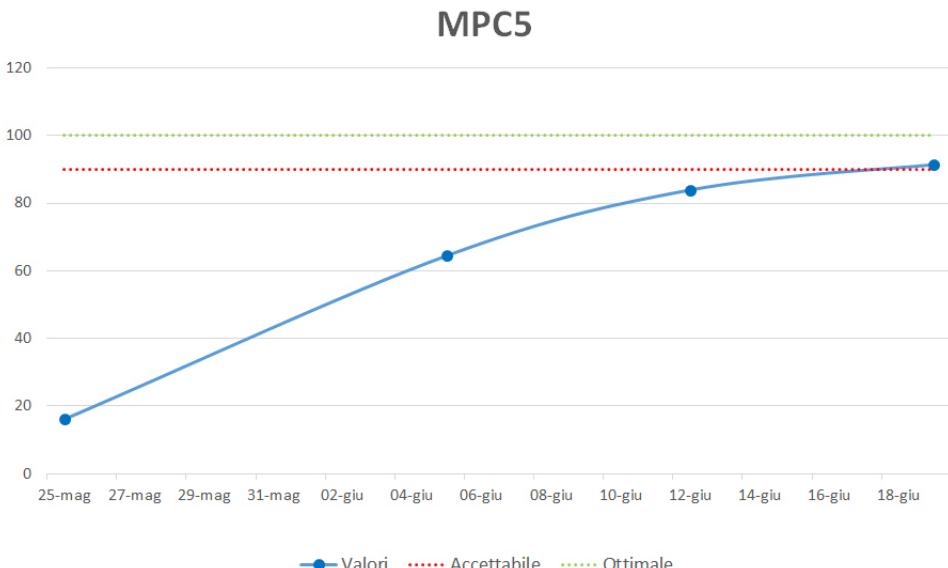


Figura 10: MPC5 - Serie storica

La percentuale di test di unità eseguiti è cresciuta di pari passo con la codifica del prodotto, che ne ha reso possibile l'esecuzione. Attualmente è stata superata la soglia di accettabilità, con l'obiettivo però di raggiungere un valore ottimale entro la consegna del prodotto.

E.1.3.2 MPC6 - Test di integrazione eseguiti

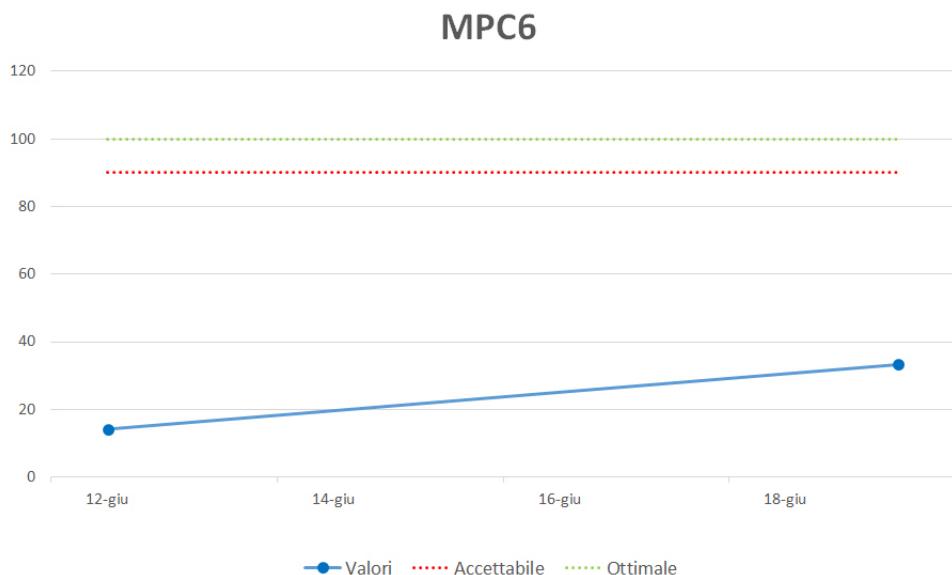


Figura 11: MPC6 - Serie storica

La percentuale di test di integrazione eseguiti sta attualmente crescendo, pur non avendo ancora raggiunto la soglia di accettabilità poiché sono attualmente mancanti delle parti di codice necessarie ad eseguire correttamente la maggior parte dei test implementati. L'obiettivo è però di raggiungere un valore ottimale entro la consegna del prodotto.

E.1.3.3 Livello SPICE

Il livello complessivo a cui si attesta il processo di verifica è *Stabilito*. Questo perché tutte le linee guida e la sequenza di processi standard su cui si basa la verifica sono stati individuati e definiti.

Non è stato attualmente raggiunto il livello successivo perché le misurazioni non sono state del tutto positive e alcune tecniche di controllo non sono ancora state del tutto automatizzate.

E.1.4 Gestione

E.1.4.1 MPC9 - Schedule Variance

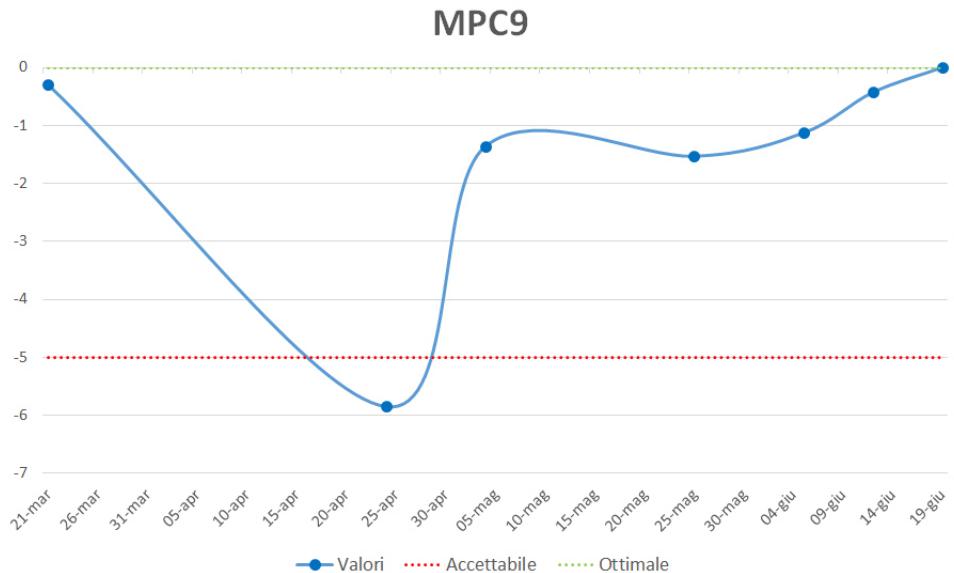


Figura 12: MPC9 - Serie storica

La Schedule Variance dopo un quasi ottimo riscontro iniziale in corrispondenza della consegna della *Revisione dei Requisiti* era scesa sotto il valore accettabile per via di una pianificazione poco attenta ai fattori esterni al gruppo. Grazie al continuo raffinamento della pianificazione per renderla più flessibile, correggendo le ingenuità commesse durante i primi periodi, i riscontri sono migliorati nel corso del tempo. Attualmente il livello raggiunto è ottimale.

E.1.4.2 MPC10 - Budget Variance

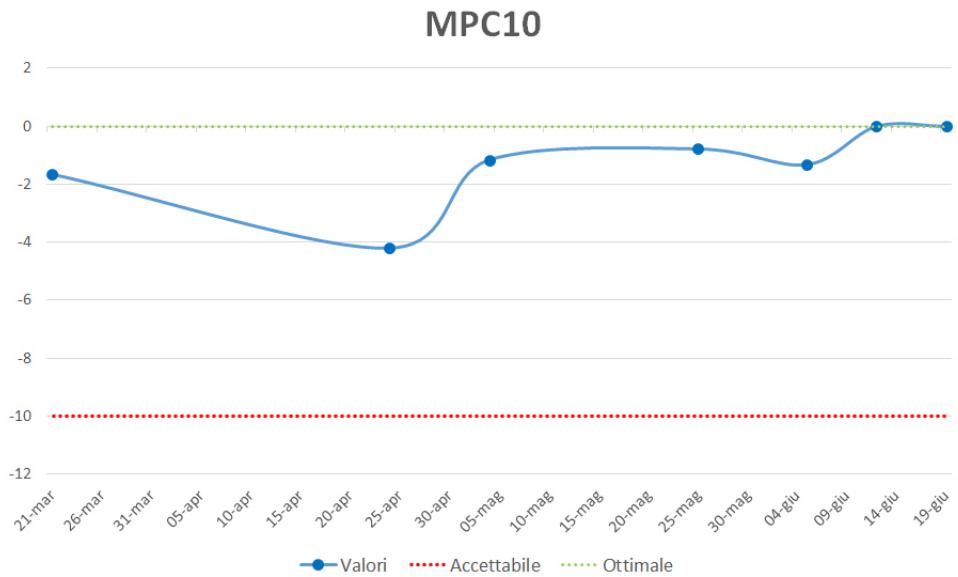


Figura 13: MPC10 - Serie storica

La Budget Variance si è sempre mantenuta largamente entro valori accettabili, pur con un vistoso peggioramento inizialmente dovuto ad una gestione ottimistica delle risorse. A seguito dei vari cambiamenti apportati alla pianificazione e alla gestione delle risorse umane le misurazioni sono gradualmente migliorate. Attualmente il livello raggiunto è ottimale.

E.1.4.3 Livello SPICE

Il livello complessivo a cui si attesta il processo di gestione è *Predicibile*. Questo perché il gruppo ha correttamente individuato tutti i processi di supporto e definito le misure del processo con obiettivi quantitativi fissati e una pianificazione dettagliata. Tali misure sono inoltre sfruttate per determinare variazioni nelle prestazioni del processo, individuarne le cause e attivare azioni correttive.

Non è ancora stato raggiunto il livello successivo poiché il livello di analisi in ottica di miglioramento non è ancora ottimale.

E.2 Prodotti

E.2.1 Funzionalità

E.2.1.1 MPS1 - Copertura requisiti obbligatori

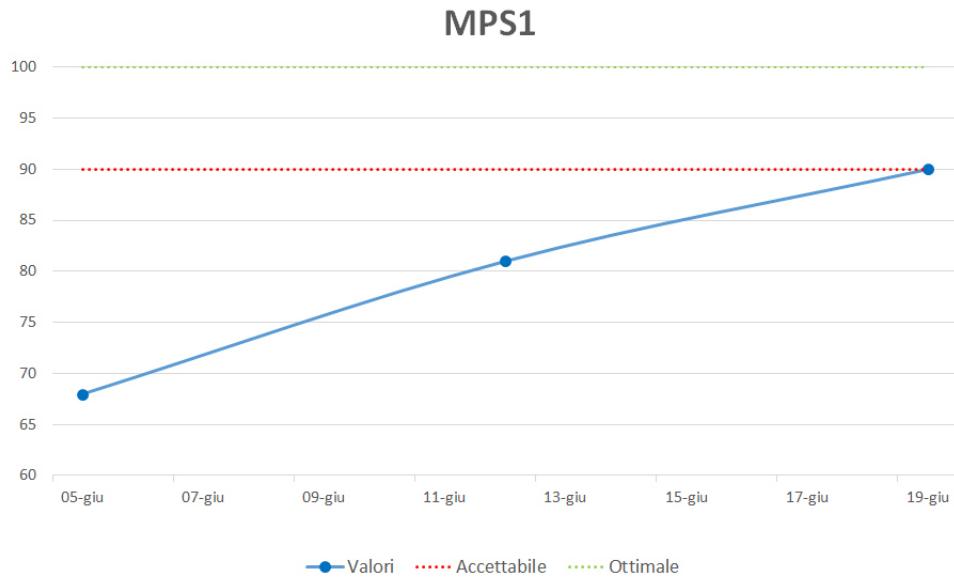


Figura 14: MPS1 - Serie storica

La percentuale di requisiti obbligatori soddisfatti è cresciuta di pari passo con la codifica del prodotto, che ne ha implementato le funzionalità correlate. Attualmente è stata superata la soglia di accettabilità, con l'obiettivo però di raggiungere un valore ottimale entro la consegna del prodotto.

E.2.1.2 MPS2 - Copertura requisiti desiderabili

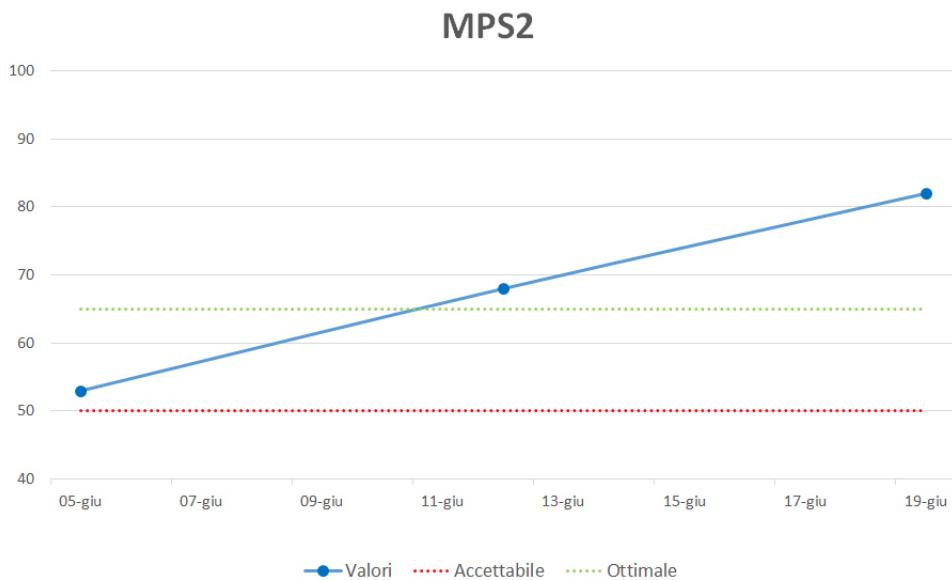


Figura 15: MPS2 - Serie storica

La percentuale di requisiti desiderabili soddisfatti è stata fin dal ciclo uno di codifica tenuta in considerazione, partendo già da una base accettabile. La copertura è inoltre cresciuta ulteriormente nel tempo. Attualmente supera ampiamente il valore ritenuto ottimale, ma entro la consegna del prodotto il gruppo intende migliorare ulteriormente tale risultato.

E.2.2 Affidabilità

E.2.2.1 MPS10 - Copertura del codice

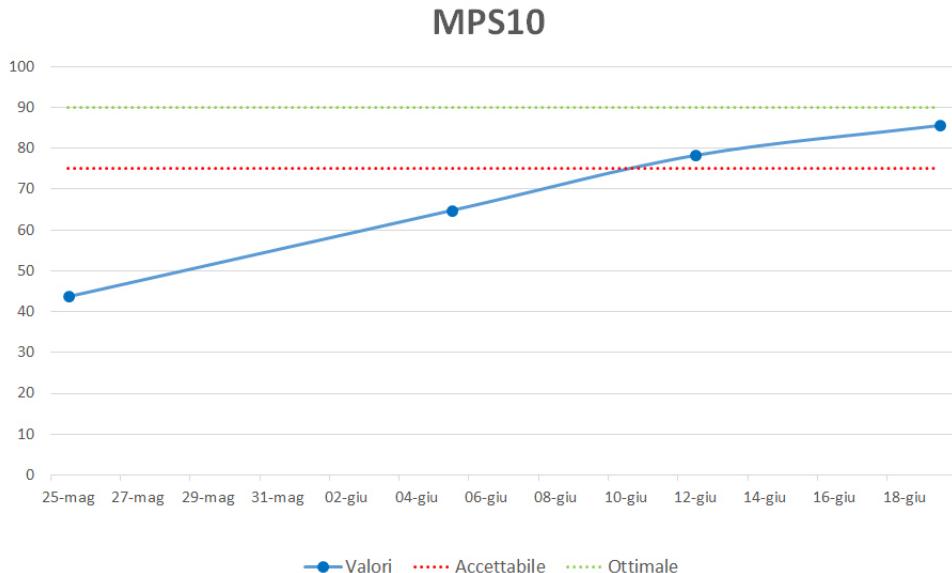


Figura 16: MPS10 - Serie storica

La percentuale di codice coperto dai test inizialmente non era sufficiente, ma nel tempo il gruppo ha lavorato in tale direzione per migliorare questo aspetto. Attualmente la copertura è ampiamente accettabile, con l'obiettivo però di raggiungere un valore ottimale entro la consegna del prodotto.

E.2.3 Efficienza

E.2.3.1 MPS9 - Nested Block Depth

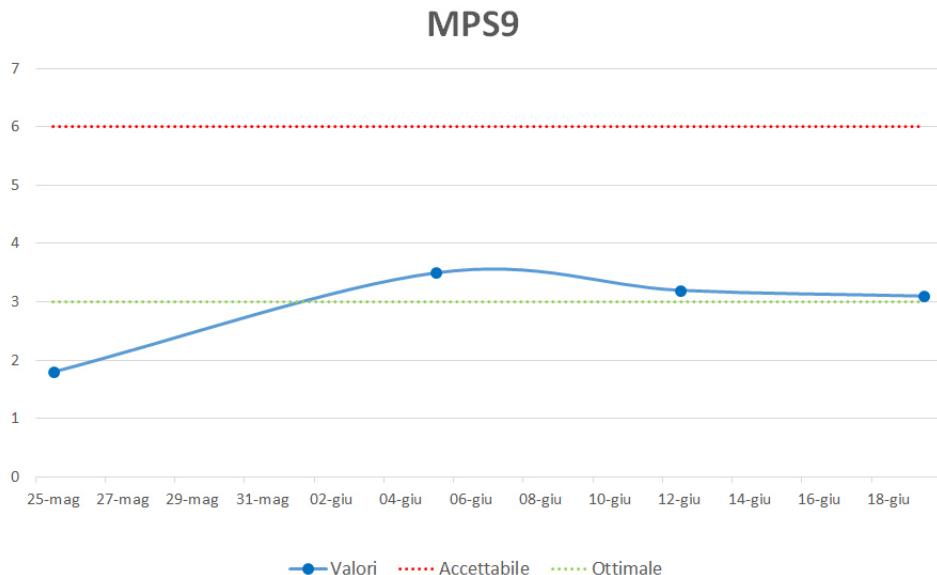


Figura 17: MPS9 - Serie storica

Il livello di annidamento delle strutture di controllo, dopo essere partito da un livello ottimale, è tornato entro una misurazione di accettabilità con l'aumento della complessità del codice durante la codifica. Attualmente è vicino alla soglia di ottimalità, pur non raggiungendola.

E.2.4 Usabilità

E.2.4.1 MPD1 - Indice Gulpease

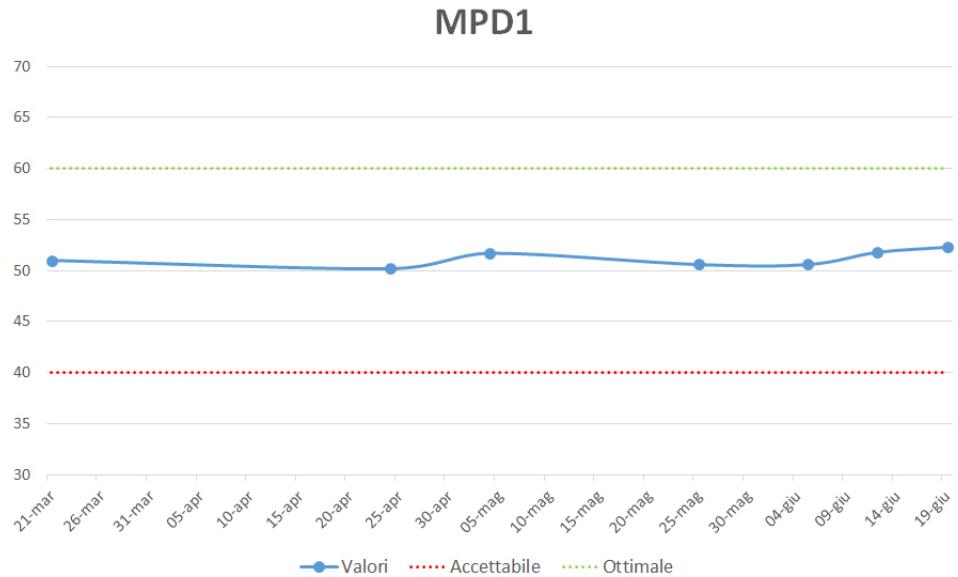


Figura 18: MPD1 - Serie storica

L'indice Gulpease dei documenti si è sempre mantenuto a metà tra il valore accettabile e quello ottimale. In particolare è leggermente peggiorato in corrispondenza del documento *Definizione di Prodotto* per via del contenuto particolarmente tecnico dello stesso. Attualmente è migliorato leggermente, pur senza raggiungere il valore ottimale.

E.2.5 Manutenibilità

E.2.5.1 MPS3 - Average Cyclomatic Complexity

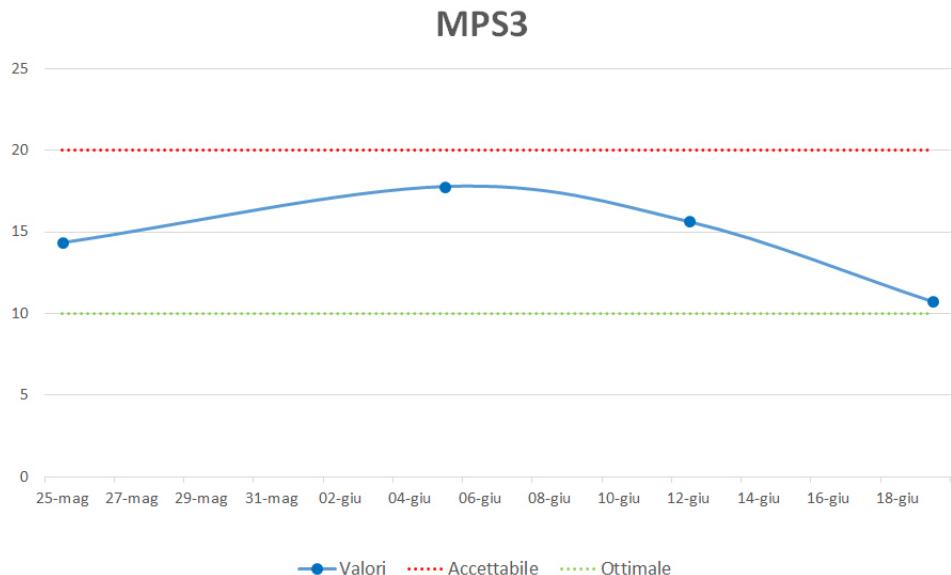


Figura 19: MPS3 - Serie storica

La complessità ciclometrica media si è sempre mantenuta in un intervallo accettabile. C'è stato un iniziale peggioramento dovuto all'aumento della complessità del codice durante la codifica. Per tal motivo sono state apportate modifiche correttive che hanno portato a un costante miglioramento e all'avvicinarsi ad un valore ottimale.

E.2.5.2 MPS4 - Coupling Between Object Classes

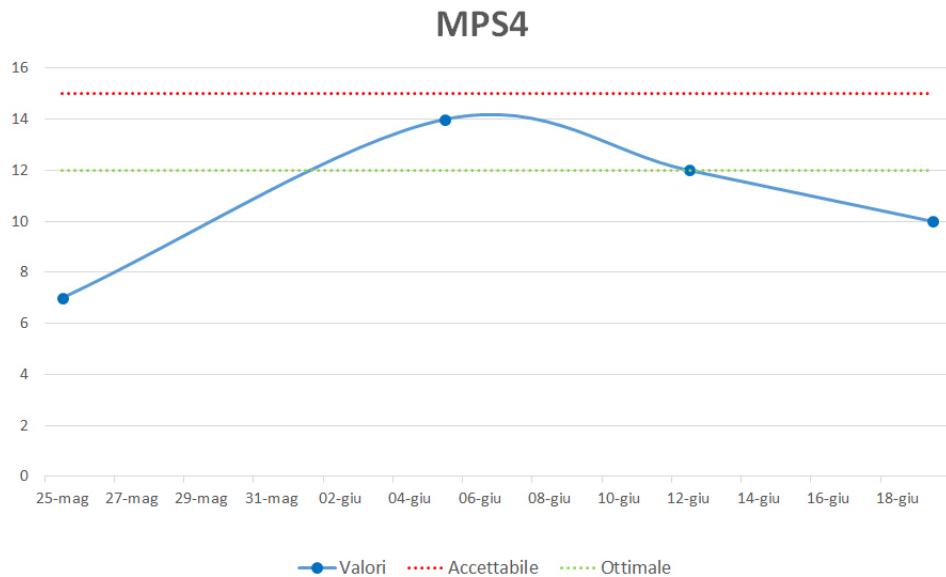


Figura 20: MPS4 - Serie storica

L'accoppiamento medio tra le classi inizialmente era largamente ottimale per via della mancanza di molte componenti del prodotto, ancora non codificate. Con l'incremento del codice l'accoppiamento è salito oltre la soglia ottimale, senza tuttavia risultare non accettabile. Grazie a miglioramenti dell'architettura del sistema e del codice da essa derivante attualmente il livello di accoppiamento è ottimale.

E.2.5.3 MPS5 - Weighted Method Complexity

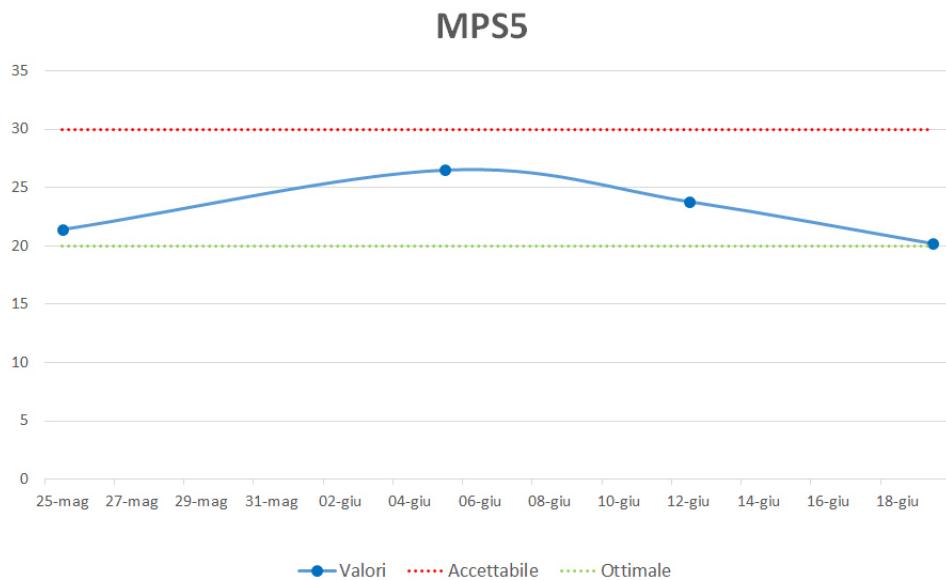


Figura 21: MPS5 - Serie storica

La complessità media dei metodi è stata fin da subito vicina al livello ottimale, con un lieve peggioramento iniziale dovuto all'introduzione durante il ciclo uno di codifica di alcuni dei metodi più complessi dell'applicazione. Con l'avanzare della codifica e l'implementazione di alcuni metodi minori, oltre alla scomposizione dei metodi più complessi, attualmente il livello di complessità raggiunto è ottimale.

E.2.5.4 MPS6 - Class Size Attributes

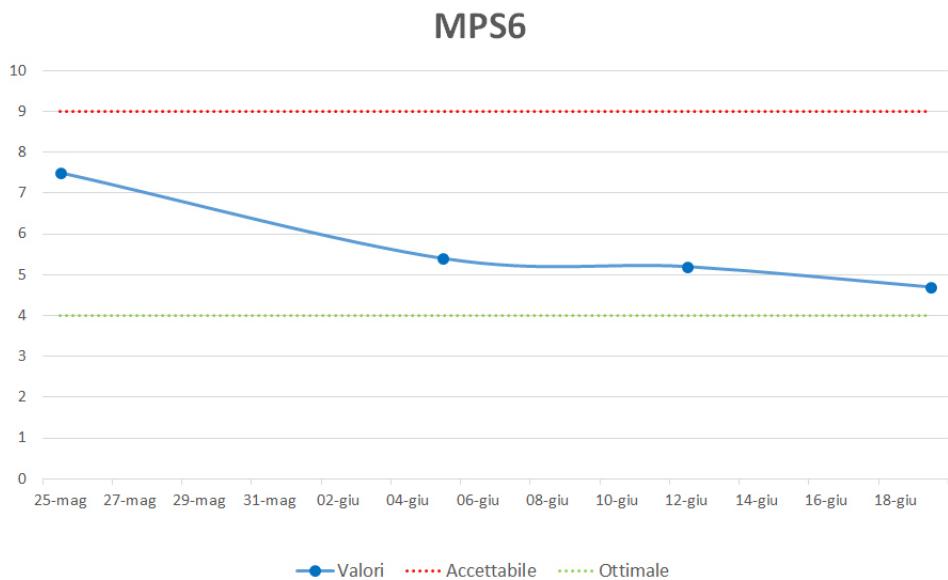


Figura 22: MPS6 - Serie storica

Il numero di campi dati medio per classe si è sempre attestato in un range accettabile, ma è costantemente migliorato nel tempo. Ciò è stato possibile grazie all'introduzione di classi più specializzate nel corso della codifica e alla riprogettazione di alcune delle classi più complesse. Il livello raggiunto attualmente si avvicina all'ottimalità, con ancora però dei margini di miglioramento.

E.2.5.5 MPS7 - Number of Parameters

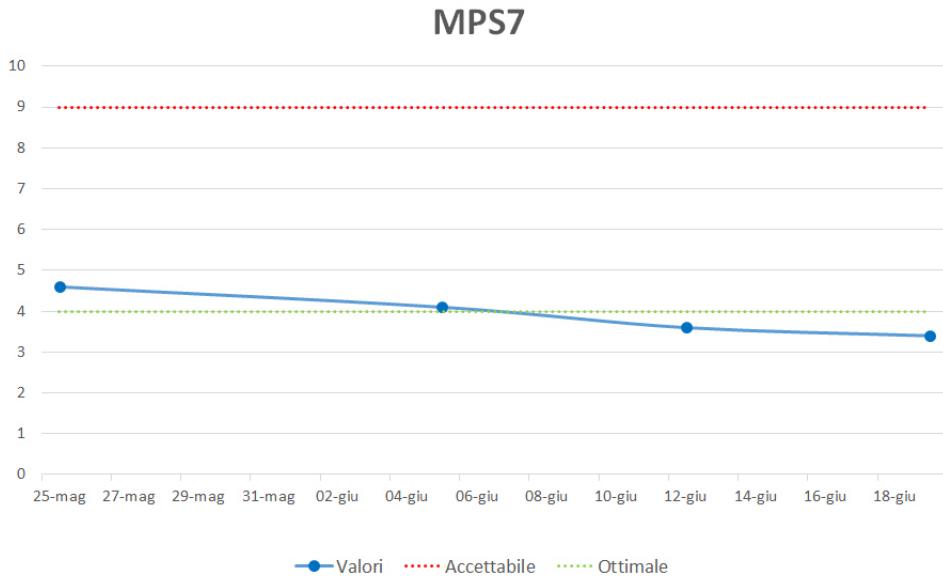


Figura 23: MPS7 - Serie storica

Il numero medio di parametri associati ad un metodo è sempre stato largamente accettabile. Durante la codifica le misurazioni sono gradualmente migliorate grazie l'introduzione di vari metodi di supporto con pochi parametri. Attualmente il livello raggiunto è ottimale.

E.2.5.6 MPS8 - Non-comment Lines of Code

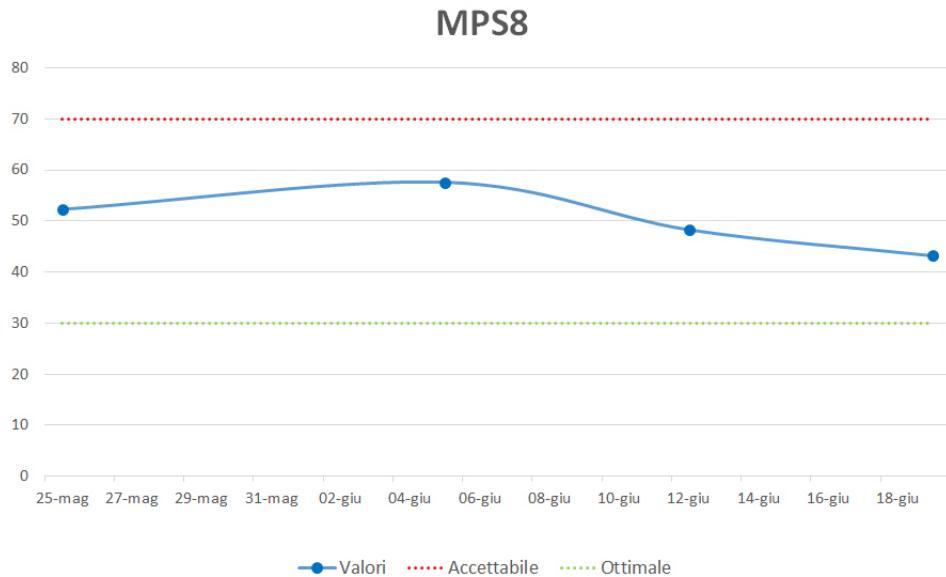


Figura 24: MPS8 - Serie storica

Il numero medio di linee di codice per metodo, escludendo i commenti, ha sempre mantenuto un valore accettabile. Inizialmente ha subito un lieve peggioramento, per via dell'introduzione durante la codifica di alcuni metodi particolarmente lunghi. Con la successiva introduzione di metodi di supporto e la modifica di alcuni dei metodi con più linee di codice, i riscontri di tale metrica stanno attualmente migliorando. Non è ancora stato però raggiunto il livello ottimale.

F Esiti delle revisioni

F.1 Revisione dei Requisiti

Successivamente alla prima revisione formale, il gruppo ha deciso di apportare varie modifiche ai documenti, basandosi sulle segnalazioni ricevute. Di seguito vengono descritte brevemente tali modifiche:

- **Studio di Fattibilità:** nel documento è stata modificata la descrizione del capitolo scelto.
- **Norme di Progetto:** ampliata la sezione del processo di sviluppo nelle parti riguardanti la progettazione e la codifica. Incremento del processo di verifica con l'inserimento delle nuove metriche e la rimozione di quelle ritenute poco significative. Inserimento di una sezione dedicata al processo di validazione contenente indicazioni sui test.
- **Piano di Qualifica:** notevole ristrutturazione della struttura del documento che ha portato a collocare la sezione dedicata alle metriche nel documento *Norme di progetto* e a porre degli obiettivi di qualità che facciano richiamo a tali metriche in maniera più attinente allo scopo del documento. Ampliamento delle sezioni riguardanti gli standard di qualità adottati, ora divise in tre appendici distinte. Aggiunta di un'appendice dedicata ai test.
- **Piano di Progetto:** modifica della struttura dell'analisi dei rischi in forma tabellare, oltre che spostamento della loro rilevazione in appendice dedicata. Consistente cambio nella pianificazione per meglio denotare l'adozione di un *modello incrementale_G*, oltre che per passaggio ad RP min. Eliminato il termine "fase" mal collocato. Riportata l'attualizzazione dei rischi e il *consuntivo_G* per le nuove attività svolte. Espansa l'analisi del preventivo a finire.
- **Analisi dei Requisiti:** Approfondimento sulle funzionalità del prodotto. Introduzione del *caso d'uso_G* UC0 e corretto il flusso base degli eventi di UC1. Aumentata la fruibilità di UC5.2 inserendo dei livelli intermedi di astrazione. Correzione dei casi d'uso e dei requisiti segnalati. Inserito riferimento alle *Norme di Progetto* per la sintassi dei requisiti. Rimosso un requisito prestazionale sulla latenza del server poiché difficile da garantire. Inserite le versione di *Node.js_G* e di JavaScript utilizzate.

F.2 Revisione di Progettazione

A seguito della Revisione di Progettazione minima sostenuta il gruppo ha deciso di apportare varie modifiche ai documenti, basandosi sulle segnalazioni ricevute. Di seguito vengono descritte brevemente tali modifiche:

- **Norme di Progetto:** individuate nuove metriche di processo e spostamento della semantica adottata per casi d'uso, requisiti, $package_G$ e classi nella sottosezione documentazione relativa ai processi di supporto;
- **Piano di Qualifica:** integrazione degli obiettivi di qualità individuati in 3.1, inoltre è stato introdotto l'uso di serie storiche al fine di rendere più significativi i riscontri, ora più frequenti;
- **Piano di Progetto:** introdotta nuova pianificazione con introduzione di misurazioni settimanali delle metriche al fine di perseguire gli obiettivi di qualità individuati dal gruppo nel documento *Piano di Qualifica*;
- **Analisi dei Requisiti:** corretta *Fig. 2*, corrette le inclusioni di UC4 e sanati errori sintattici individuati;
- **Specifiche Tecniche:** approfonditi vantaggi e svantaggi di tutte le tecnologie individuate, eliminate immagini ritenute di scarso valore descrittivo aggiuntivo, individuati più controller nel package `Front-end::ViewModel`, utilizzata corretta direttiva UML per i nomi delle classi astratte e corretto tracciamento tra requisiti e componenti logiche.