

KALEIDOSCODE
SWEDesigner
SOFTWARE PER DIAGRAMMI UML

PIANO DI QUALIFICA V3.0.0



Informazioni sul documento

Versione	3.0.0
Data Redazione	09/03/2017
Redazione	Pezzuto Francesco Sanna Giovanni
Verifica	Bonato Enrico
Approvazione	Pace Giulio
Uso	Esterno
Distribuzione	<i>Prof. Vardanega Tullio</i> <i>Prof. Cardin Riccardo</i> <i>Zucchetti s.p.a.</i>

kaleidos.codec6@gmail.com

Diario delle Modifiche

Versione	Data	Autore	Descrizione
2.0.2	22/05/2017	Pezzuto Francesco	Preso atto degli errori segnalati nella valutazione di RP; sistemate/ampliate sezioni: 2.1, 2.2, 3.2, 4
2.0.0	03/05/2017	Pace Giulio	Approvazione del documento
1.1.0	30/04/2017	Bonato Enrico	Verifica del documento
1.0.5	25/04/2017	Sanna Giovanni	Stesura test
1.0.4	20/04/2017	Pezzuto Francesco	Preso atto degli errori segnalati nella valutazione di RR; riorganizzata struttura del documento; sistemate/ampliate sezioni: Obiettivi di qualità, Organizzazione, Scadenze temporali, Misure e metriche, Resoconto attività di verifica (Periodo Analisi e Analisi di dettaglio); inserite appendici Capability Maturity Model, Ciclo di Deming, ISO/IEC 9126
1.0.0	02/04/2017	Sanna Giovanni	Approvazione del documento
0.2.0	31/03/2017	Pezzuto Francesco	Verifica del documento
0.1.1	28/03/2017	Sovilla Matteo	Correzione e integrazione come indicato da verifica
0.1.0	25/03/2017	Pezzuto Francesco	Verifica del documento
0.0.5	23/03/2017	Sanna Giovanni	Stesura parte capitolo 2

Versione	Data	Autore	Descrizione
0.0.4	23/03/2017	Sovilla Matteo	Stesura parte capitolo 3
0.0.3	23/03/2017	Bonato Enrico	Stesura parte capitolo 3
0.0.2	22/03/2017	Pace Giulio	Stesura sezione Obiettivi di qualità
0.0.1	09/03/2017	Bonolo Marco	Creazione scheletro del documento e stesura della sezione Introduzione

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti utili	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	1
2	Visione generale della strategia	3
2.1	Obiettivi di qualità di processo	3
2.1.1	Rispetto della pianificazione concordata	3
2.1.2	Rispetto del budget concordato	4
2.1.3	Leggibilità della documentazione prodotta	4
2.1.4	Rispetto delle norme di progettazione	4
2.1.5	Rispetto delle norme di codifica	4
2.1.6	Corretto funzionamento delle componenti del sistema	4
2.2	Obiettivi di qualità di prodotto	4
2.2.1	Funzionalità	4
2.2.2	Affidabilità	5
2.2.3	Efficienza	5
2.2.4	Manutenibilità	5
2.2.5	Portabilità	5
2.3	Organizzazione	5
2.4	Scadenze temporali	6
3	La strategia di gestione della qualità nel dettaglio	7
3.1	Risorse	7
3.1.1	Necessarie	7
3.1.2	Disponibili	7
3.2	Misure e metriche	7
3.2.1	Obiettivi di qualità di processo	8
3.2.2	Obiettivi di qualità di prodotto	9
4	Resoconto delle attività di verifica	10
4.1	Periodo di Analisi e Analisi di dettaglio	10
4.1.1	Processi	10
4.1.2	Indici di Gulpease	10
4.2	Periodo di Progettazione architetturale	10
4.2.1	Processi	10
4.2.2	Indici di Gulpease	11
4.2.3	Progettazione	11
4.3	Periodo di Progettazione di dettaglio e Codifica	14
4.3.1	Processi	14
4.3.2	Indici di Gulpease	14
4.3.3	Progettazione	14

4.3.4	Codifica	18
A	Test	19
A.1	Test di validazione	19
A.1.1	Test TV1	19
A.1.2	Test TV2	19
A.1.3	Test TV3	20
A.1.4	Test TV3.1	20
A.1.5	Test TV4	20
A.1.6	Test TV5	20
A.1.7	Test TV6	21
A.1.8	Test TV7	21
A.1.9	Test TV8	21
A.1.10	Test TV9	22
A.1.11	Test TV10	22
A.1.12	Test TV11	22
A.1.13	Test TV12	23
A.1.14	Test TV13	23
A.1.15	Test TV14	23
A.1.16	Test TV15	24
A.1.17	Test TV16	24
A.1.18	Test TV17	24
A.1.19	Test TV18	25
A.1.20	Test TV19	25
A.1.21	Test TV20	25
A.1.22	Test TV36	26
A.1.23	Test TV37	26
A.1.24	Test TV38	27
A.1.25	Test TV39	27
A.1.26	Test TV40	27
A.1.27	Test TV41	28
A.2	Tracciamento test di Validazione	28
A.3	Test di Sistema	30
A.4	Tracciamento test di sistema	38
A.5	Test di integrazione	42
A.6	Tracciamento test di integrazione	45
A.7	Test di unità	46
A.8	Tracciamento test di unità	50
A.9	Resoconto stato test	52
B	Capability Maturity Model	53
B.1	Struttura	53
B.2	Livelli	53
C	Ciclo di Deming	54

D ISO/IEC 9126	54
D.1 Modello della qualità	54
D.1.1 Qualità esterna ed interna	54
D.1.2 Qualità in uso	55
D.2 Metriche per la qualità	55
D.2.1 Esterna	55
D.2.2 Interna	55

Elenco delle tabelle

3	Metriche per qualità di processo	8
4	Metriche per qualità di prodotto	9
5	RR - Schedule e budget variance	10
6	RR - Indici di Gulpease calcolati sulla documentazione prodotta	10
7	RP - Schedule e budget variance	10
8	RP - Indici di Gulpease calcolati sulla documentazione prodotta	11
9	RP - Grado di accoppiamento afferente	12
10	RP - Grado di accoppiamento efferente	13
11	RQ - Schedule e budget variance	14
12	RQ - Indici di Gulpease calcolati sulla documentazione prodotta	14
13	RQ - Numero di violazioni delle norme di progettazione	14
14	RQ - Grado di accoppiamento afferente	16
15	RQ - Grado di accoppiamento efferente	17
16	RQ - Fan In e Fan Out	17
17	RQ - Numero di violazioni delle norme di codifica	18
18	Metriche per qualità di prodotto	18
19	Tracciamento test di validazione	29
20	Test di sistema	37
21	Tracciamento test di sistema	41
22	Stato implementazione test sistema	41
24	Tracciamento test di integrazione	45
25	Stato implementazione test integrazione	45
26	Test di Unità	50
27	Tracciamento test di Unità	52
28	Stato implementazione test unità	52
29	Riassunto stato implementazione test	52

Elenco delle figure

1	Diagramma test di integrazione	42
---	--	----

1 Introduzione

1.1 Scopo del documento

Questo documento definisce gli obiettivi e le metodologie che ogni membro del gruppo *KaleidosCode* adotterà per garantire un determinato livello di qualità del prodotto.

A tal proposito ogni membro del gruppo è tenuto a leggere, perseguire e raggiungere gli obiettivi definiti in esso.

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un software di costruzione di diagrammi UML_G con la relativa generazione di codice Java_G e Javascript_G utilizzando tecnologie web. Il prodotto deve essere conforme ai vincoli qualitativi richiesti dal committente.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite sono riportate nel documento *Glossario v2.0.0*.

La prima occorrenza di ciascuno di questi vocaboli è marcata da una “G” maiuscola in pedice.

1.4 Riferimenti utili

1.4.1 Riferimenti normativi

- **Capitolato_G d'appalto:**
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf> (09/03/2017);
- **Norme di progetto:** *Norme di progetto v3.0.0*.

1.4.2 Riferimenti informativi

- **Slide dell'insegnamento di Ingegneria del Software 1° semestre:**
 - Qualità del software:
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L10.pdf> (02/04/2017);
 - Qualità di Processo:
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L11.pdf> (02/04/2017).
- **Slide dell'insegnamento di Ingegneria del Software 2° semestre:**
 - Metodi e obiettivi di quantificazione:
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L15.pdf> (02/04/2017).
- **ISO_G 9001:**
https://it.wikipedia.org/wiki/Norme_della_serie_ISO_9000#ISO_9001 (02/04/2017);

- ISO 9126: https://it.wikipedia.org/wiki/ISO/IEC_9126 (02/04/2017);
- ISO 12207: https://en.wikipedia.org/wiki/ISO/IEC_12207 (02/04/2017);
- Indice Gulpease_G:
https://it.wikipedia.org/wiki/Indice_Gulpease (02/04/2017);
- Complessità ciclomatica:
https://en.wikipedia.org/wiki/Cyclomatic_complexity (02/04/2017);
- Capability Maturity Model (CMM_G):
https://en.wikipedia.org/wiki/Capability_Maturity_Model (02/04/2017);
- Analisi dei requisiti: *Analisi dei requisiti v3.0.0*;
- Piano di progetto: *Piano di progetto v3.0.0*;
- Glossario: *Glossario v2.0.0*.

2 Visione generale della strategia

Per garantire la qualità dei prodotti realizzati durante lo sviluppo del progetto è indispensabile definire e perseguire strategie che assicurino la qualità dei processi adottati nonché il loro continuo miglioramento; inoltre, è necessario definire metriche e pianificare attività che valutino in modo preciso la qualità dei prodotti ottenuti e dei processi adottati. A tal scopo verranno adottate le seguenti strategie:

- Definizione accurata di norme che regolamentano e standardizzano i processi coinvolti nel progetto in termini di:
 - Processo di fornitura;
 - Processo di sviluppo;
 - Processi di supporto;
 - Processi organizzativi.
- Descrizione dettagliata delle strategie di pianificazione adottate per lo sviluppo del progetto in termini di:
 - Modello di sviluppo adottato;
 - Analisi dei rischi che si possono incontrare;
 - Pianificazione delle attività e dei tempi;
 - Stima preventiva delle risorse che saranno impiegate;
 - Assegnazione delle risorse al fine di portare a termine le attività pianificate nei tempi previsti;
 - Consuntivo, durante lo sviluppo del progetto, delle risorse impiegate.
- Ad ogni processo coinvolto nello sviluppo del progetto è stato scelto di applicare il ciclo di PDCA_G affiancato dal CMM.¹ Essi infatti permettono il controllo, la valutazione e il miglioramento continuo dei processi nonché la determinazione del livello di maturità dell'organizzazione nel gestirli.

2.1 Obiettivi di qualità di processo

Prendendo come riferimento la normativa [ISO/IEC 12207], il gruppo *KaleidosCode* ha definito i seguenti obiettivi di qualità di processo che si impegna a perseguire.

2.1.1 Rispetto della pianificazione concordata

È necessario pianificare le attività per la realizzazione del progetto ed è quindi fondamentale rispettare tale pianificazione per garantire la consegna del prodotto secondo le tempistiche concordate.

Misura e metrica adottate sono descritte nella sezione 3.2.1.

¹Per maggiori informazioni, consultare le appendici [B](#) e [C](#)

2.1.2 Rispetto del budget concordato

È necessario stabilire un budget per il costo della realizzazione del prodotto ed è quindi fondamentale far rientrare i costi per le risorse nella spesa prevista.

Misura e metrica adottate sono descritte nella sezione [3.2.1](#).

2.1.3 Leggibilità della documentazione prodotta

Durante la realizzazione del prodotto è necessario redigere la documentazione delle attività di pianificazione, gestione, sviluppo, verifica e validazione oltre che i loro prodotti.

È quindi fondamentale che i documenti prodotti siano, per quanto tecnici, facilmente leggibili.

Misura e metrica adottate sono descritte nella sezione [3.2.1](#).

2.1.4 Rispetto delle norme di progettazione

In progettazione si definiranno i moduli e le componenti del prodotto. Sarà necessario rispettare le norme di progettazione definite nel documento *Norme di progetto*.

Misura e metrica adottate sono descritte nella sezione [3.2.1](#). Inoltre, sarà utilizzata una misura e metrica, descritte nella sezione [3.2.2](#), per valutarne la qualità.

2.1.5 Rispetto delle norme di codifica

Nel periodo di codifica, bisognerà sviluppare le unità software ideate in progettazione. Per questo motivo, sarà necessario rispettare le norme di codifica concordate nel documento *Norme di progetto*.

Misura e metrica adottate sono descritte nella sezione [3.2.1](#).

2.1.6 Corretto funzionamento delle componenti del sistema

Sarà necessario sottoporre le unità software sviluppate ad una serie di test² per valutarne il corretto funzionamento.

Misura e metrica adottate sono descritte nella sezione [3.2.1](#).

2.2 Obiettivi di qualità di prodotto

Prendendo come riferimento la normativa [ISO/IEC 9001] ed in particolare [ISO/IEC 9126]³, il gruppo *KaleidosCode* ha definito i seguenti obiettivi di qualità che si impegna a far raggiungere al prodotto *SWEDesigner*.

2.2.1 Funzionalità

Si garantisce che *SWEDesigner* abbia tutte le funzionalità definite e concordate con il *Zucchetti s.p.a.* nel documento *Analisi dei requisiti v3.0.0*. L'implementazione di ogni requisito deve essere quanto più completa ed economica.

Misure e metriche adottate sono descritte nella sezione [3.2.2](#).

²Definiti nella sezione [A](#)

³Per maggiori informazioni, consultare l'appendice [D](#)

2.2.2 Affidabilità

Il sistema deve funzionare nella sua completezza.

Misure e metriche adottate sono descritte nella sezione [3.2.2](#).

2.2.3 Efficienza

Il sistema deve minimizzare l'utilizzo delle risorse impiegate e fornire le funzionalità richieste nel minor tempo possibile.

Misure e metriche adottate sono descritte nella sezione [3.2.2](#).

2.2.4 Manutenibilità

Il codice prodotto per realizzare il sistema deve essere comprensibile ed estensibile.

Misure e metriche adottate sono descritte nella sezione [3.2.2](#).

2.2.5 Portabilità

Il sistema è un applicativo web. Per questo motivo, si garantisce che il front-end_G sarà completamente funzionante ed utilizzabile almeno dal browser_G Google Chrome, dove verrà testato il prodotto durante lo sviluppo. Inoltre, si perseguirà l'obiettivo di garantire la completa funzionalità del prodotto anche su altri browser, in particolare: Mozilla Firefox e Microsoft Edge.

Misure e metriche adottate sono descritte nella sezione [3.2.2](#).

2.3 Organizzazione

La gestione della strategia di verifica si basa sull'attuazione delle relative attività descritte nelle *Norme di progetto v3.0.0*. Tali attività vengono eseguite per ogni processo attuato allo scopo di verifica della qualità del processo stesso e dell'eventuale prodotto ottenuto facendo riferimento anche alle metriche definite nella sezione [3.2](#).

In ogni documento è presente un diario delle modifiche che permette di concentrare l'attività di verifica solo nelle parti modificate dopo l'ultima eseguita.

Data la diversa natura dei prodotti ottenuti dai periodi del progetto si applicherà, per ognuno di essi, una diversa procedura di verifica:

- **Analisi e Analisi di dettaglio:** si effettuerà una prima stesura dei documenti illustrati nel *Piano di progetto v3.0.0*;
 - Verrà controllata la correttezza ortografica con *LanguageTool 3.6*, opportunamente integrato in *TexStudio*;
 - Verrà controllata la correttezza lessicale con un'attenta ed accurata rilettura affiancata dal controllo di *LanguageTool 3.6*;
 - Verrà controllata la correttezza dei contenuti rispetto alle aspettative del documento con un'attenta rilettura;

- Verrà controllato il corretto tracciamento e la corrispondenza di ciascun requisito con un caso d’uso, mediante l’utilizzo dell’applicativo web creato appositamente⁴;
- Verrà controllato che la stesura di ciascun documento rispetti le norme definite in *Norme di progetto v3.0.0*;
- Verranno controllate le rappresentazioni grafiche, figure e tabelle assicurandosi che per ciascuna di esse sia presente un’opportuna didascalia e un relativo indice nel corrispondente documento;
- **Progettazione architettuale:** verrà controllato che tutti i requisiti corrispondano ad un componente individuato in questo periodo e se ne assicurerà la tracciabilità;
- **Progettazione di dettaglio e Codifica:** durante ciascuna delle iterazioni in questo periodo i *Programmatore* svolgeranno l’attività di codifica e di esecuzione dei test previsti per la verifica del codice prodotto. Tali attività avverranno nel modo più automatizzato possibile seguendo le norme descritte in *Norme di progetto v3.0.0*. I *Verificatori* avranno il compito di supervisionare le attività controllando la presenza di eventuali errori;
- **Validazione e verifica:** verrà effettuato il collaudo del prodotto, in modo da assicurarne il corretto funzionamento al momento della consegna.

Per ogni periodo a partire dalla progettazione architettuale verranno inoltre effettuati tutti i controlli opportuni descritti al primo punto di questo paragrafo nei nuovi documenti redatti e in presenza di modifiche o integrazioni ai documenti precedentemente stesi.

2.4 Scadenze temporali

Dato l’obiettivo di rispettare le scadenze fissate nel *Piano di progetto v3.0.0*, è indispensabile pianificare anche l’attività di verifica della documentazione e del codice prodotto in modo che risulti sistematica e organizzata. Grazie all’applicazione di tale strategia l’individuazione e la correzione degli errori avverrà il prima possibile, impedendo la loro rapida diffusione e mitigando la possibilità che gli stessi si ripresentino in futuro, diminuendo così il rischio di ritardi. Tale pianificazione è documentata nel *Piano di progetto v3.0.0* il quale contiene, nella sottosezione “Scadenze”, le scadenze temporali che il gruppo *KaleidosCode* si impegna a rispettare.

⁴Consultare *Norme di progetto v3.0.0* per maggiori informazioni sullo strumento di tracciamento utilizzato.

3 La strategia di gestione della qualità nel dettaglio

3.1 Risorse

3.1.1 Necessarie

Per la realizzazione del prodotto sono necessarie le risorse umane e tecnologiche elencate di seguito.

- **Risorse umane:** sono descritte dettagliatamente nel *Piano di progetto v3.0.0*:
 - *Responsabile di progetto*;
 - *Amministratore*;
 - *Analista*;
 - *Progettista*;
 - *Programmatore*;
 - *Verificatore*.
- **Risorse software:** sono descritte dettagliatamente nelle *Norme di progetto v3.0.0*. Si tratta di software che permettono:
 - la comunicazione e la condivisione del lavoro tra gli elementi del team;
 - la stesura della documentazione in formato $\text{L}^{\text{T}}\text{E}^{\text{X}}_{\text{G}}$;
 - la creazione di diagrammi UML;
 - la codifica nei linguaggi di programmazione scelti;
 - la semplificazione delle attività di verifica;
 - la gestione dei test sul codice.
- **Risorse hardware:** ciascun componente del gruppo deve avere un computer con tutti i software necessari descritti nelle *Norme di progetto v3.0.0*. È necessario avere a disposizione almeno un luogo dove poter effettuare le riunioni interne.

3.1.2 Disponibili

Ogni membro del team ha a disposizione uno o più computer personali dotati degli strumenti necessari.

Le riunioni interne si svolgono presso le aule del dipartimento di Matematica dell'Università degli Studi di Padova.

3.2 Misure e metriche

Il processo di verifica deve essere quantificabile per fornire informazioni utili.

Le metriche adottate sono descritte approfonditamente nel documento *Norme di progetto*. Per alcune di esse si definiranno due intervalli di misure (range_G):

- **Range di accettazione:** intervallo di valori vincolante per l'accettazione del prodotto;

- **Range ottimale:** intervallo di valori entro cui è consigliabile rientri la misurazione. Il mancato rispetto di questa condizione non pregiudica l'accettazione del prodotto, ma richiede verifiche più approfondite in merito.

Di seguito sono definiti gli specifici obiettivi quantitativi da perseguire.

3.2.1 Obiettivi di qualità di processo

Metrica	Range di accettazione	Range ottimale
Numero di violazioni delle norme di progettazione	0 - 10	0 - 5
Numero di violazioni delle norme di codifica	0 - 10	0 - 5
Percentuale di test di validazione effettuati	90% - 100%	100%
Percentuale di test di integrazione effettuati	90% - 100%	100%
Percentuale di test di sistema effettuati	90% - 100%	100%
Percentuale di test di unità effettuati	90% - 100%	100%
Schedule Variance	$\geq -(\text{preventivo} * 10\%)$	≥ 0
Budget Variance	$\geq -(\text{preventivo} * 10\%)$	≥ 0
Indice Gulpease	40 - 100	50 - 100

Tabella 3: Metriche per qualità di processo

3.2.2 Obiettivi di qualità di prodotto

Metrica	Obiettivo
Soddisfacimento dei requisiti obbligatori	100%
Soddisfacimento dei requisiti desiderabili	70% - 100%

Metrica	Range di accettazione	Range ottimale
Percentuale totale di test superati	80% - 100%	90% - 100%
Grado di accoppiamento afferente	0 - 7	0 - 5
Grado di accoppiamento efferente	0 - 7	0 - 5
Linee di commento su linee di codice	≥ 0.25	≥ 0.30
Numero di parametri	0 - 8	0 - 5
Numero di campi dati	0 - 16	0 - 10
Complessità ciclomatica	0 - 10	0 - 6
Livello di annidamento	0 - 6	0 - 4
Chiamate innestate di metodi	0 - 6	0 - 4
Copertura del codice	80% - 100%	90% - 100%
Numero di linee per metodo	≤ 60	≤ 40
Validazione W3C	0 - 10 (per pagina)	0 - 0 (per pagina)

Tabella 4: Metriche per qualità di prodotto

4 Resoconto delle attività di verifica

4.1 Periodo di Analisi e Analisi di dettaglio

4.1.1 Processi

Documento	Schedule variance	Budget variance
<i>Analisi dei requisiti v1.0.0</i>	0%	N.D.
<i>Glossario v1.0.0</i>	0%	N.D.
<i>Norme di progetto v1.0.0</i>	0%	N.D.
<i>Piano di progetto v1.0.0</i>	0%	N.D.
<i>Piano di qualifica v1.0.0</i>	0%	N.D.
<i>Studio di fattibilità v1.0.0</i>	0%	N.D.

Tabella 5: RR - Schedule e budget variance

4.1.2 Indici di Gulpease

Documento	Valutazione	Esito
<i>Analisi dei requisiti v1.0.0</i>	41.08	Accettabile
<i>Glossario v1.0.0</i>	45.95	Accettabile
<i>Norme di progetto v1.0.0</i>	46.95	Accettabile
<i>Piano di progetto v1.0.0</i>	48.23	Accettabile
<i>Piano di qualifica v1.0.0</i>	53.99	Ottimale
<i>Studio di fattibilità v1.0.0</i>	46.73	Accettabile

Tabella 6: RR - Indici di Gulpease calcolati sulla documentazione prodotta

4.2 Periodo di Progettazione architettuale

4.2.1 Processi

Documento	Schedule variance	Budget variance
<i>Analisi dei requisiti v2.0.0</i>	7%	0%
<i>Glossario v2.0.0</i>	0%	0%
<i>Norme di progetto v2.0.0</i>	0%	0%
<i>Piano di progetto v2.0.0</i>	0%	0%
<i>Piano di qualifica v2.0.0</i>	-10%	0%
<i>Specifiche tecniche v1.0.0</i>	8%	-5%

Tabella 7: RP - Schedule e budget variance

4.2.2 Indici di Gulpease

Documento	Valutazione	Esito
<i>Analisi dei requisiti v2.0.0</i>	44.03	Accettabile
<i>Glossario v2.0.0</i>	44.98	Accettabile
<i>Norme di progetto v2.0.0</i>	46.69	Accettabile
<i>Piano di progetto v2.0.0</i>	50.42	Ottimale
<i>Piano di qualifica v2.0.0</i>	52.64	Ottimale
<i>Specifica tecnica v1.0.0</i>	42.84	Accettabile

Tabella 8: RP - Indici di Gulpease calcolati sulla documentazione prodotta

4.2.3 Progettazione

Componente	Accoppiamento afferente
Client::View::MainView	0
Client::View::TitleBarView	1
Client::View::ToolBarView	1
Client::View::AddressView	1
Client::View::EditPanelView	1
Client::View::Paper	1
Client::Model::Command	3
Client::Model::ConcreteCommand	0
Client::Model::State	1
Client::Model::DAO	1
Client::Model::MainModel	4
Client::Model::TitleBarModel	1
Client::Model::ToolBarModel	5
Client::Model::PackageToolbar	0
Client::Model::ClassToolbar	0
Client::Model::ActivityToolbar	0
Client::Model::BubbleToolbar	0
Client::Model::AddressModel	1
Client::Model::EditPanelModel	2
Client::Model::ItemPanel	0
Client::Model::DiagramTree	1
Client::Model::Diagram	5
Client::Model::PackageDiagram	0
Client::Model::ClassDiagram	0

Componente	Accoppiamento afferente
Client::Model::ActivityDiagram	0
Client::Model::BubbleDiagram	0
Client::Model::RequestHandler::Sender	1
Client::Model::RequestHandler::Receiver	1
Server::RequestHandler::Sender	1
Server::RequestHandler::Receiver	1
Server::CodeGenerator::CodeGenerator	1
Server::CodeGenerator::Parser::Parser	2
Server::CodeGenerator::Coder::Coder	2
Server::CodeGenerator::Coder::JavaCoder	0
Server::CodeGenerator::Coder::JavaScriptCoder	0
Server::CodeGenerator::Coder::CoderClass	0
Server::CodeGenerator::Coder::CoderOperation	0
Server::CodeGenerator::Coder::CoderParameter	0
Server::CodeGenerator::Coder::CoderAttribute	0
Server::CodeGenerator::Coder::CoderActivity	0
Server::CodeGenerator::Coder::CodedProg	1
Server::CodeGenerator::Coder::CoderElement	6
Server::CodeGenerator::Builder::Builder	1
Server::CodeGenerator::Zipper::Zipper	1
Server::DAO	1

Tabella 9: RP - Grado di accoppiamento afferente

Tutti i componenti sono almeno sul range accettabile.

Componente	Accoppiamento efferente
Client::View::MainView	9
Client::View::TitleBarView	0
Client::View::ToolBarView	0
Client::View::AddressView	0
Client::View::EditPanelView	0
Client::View::Paper	1
Client::Model::Command	0
Client::Model::ConcreteCommand	2
Client::Model::State	1
Client::Model::DAO	1
Client::Model::MainModel	1

Componente	Accoppiamento efferente
Client::Model::TitleBarModel	0
Client::Model::ToolBarModel	0
Client::Model::PackageToolbar	1
Client::Model::ClassToolbar	1
Client::Model::ActivityToolbar	1
Client::Model::BubbleToolbar	1
Client::Model::AddressModel	0
Client::Model::EditPanelModel	0
Client::Model::ItemPanel	1
Client::Model::DiagramTree	1
Client::Model::Diagram	1
Client::Model::PackageDiagram	1
Client::Model::ClassDiagram	1
Client::Model::ActivityDiagram	1
Client::Model::BubbleDiagram	1
Client::Model::RequestHandler::Sender	1
Client::Model::RequestHandler::Receiver	1
Server::RequestHandler::Sender	1
Server::RequestHandler::Receiver	1
Server::CodeGenerator::CodeGenerator	3
Server::CodeGenerator::Parser::Parser	0
Server::CodeGenerator::Coder::Coder	2
Server::CodeGenerator::Coder::JavaCoder	1
Server::CodeGenerator::Coder::JavaScriptCoder	1
Server::CodeGenerator::Coder::CoderClass	1
Server::CodeGenerator::Coder::CoderOperation	1
Server::CodeGenerator::Coder::CoderParameter	1
Server::CodeGenerator::Coder::CoderAttribute	1
Server::CodeGenerator::Coder::CoderActivity	2
Server::CodeGenerator::Coder::CodedProg	0
Server::CodeGenerator::Coder::CoderElement	0
Server::CodeGenerator::Builder::Builder	1
Server::CodeGenerator::Zipper::Zipper	1
Server::DAO	0

Tabella 10: RP - Grado di accoppiamento efferente

Tutti i componenti sono almeno sul range accettabile.

4.3 Periodo di Progettazione di dettaglio e Codifica

4.3.1 Processi

Documento	Schedule variance	Budget variance
<i>Analisi dei requisiti v3.0.0</i>	0%	0%
<i>Glossario v2.0.0</i>	0%	0%
<i>Norme di progetto v3.0.0</i>	0%	0%
<i>Piano di progetto v3.0.0</i>	0%	0%
<i>Piano di qualifica v3.0.0</i>	-5%	0%
<i>Specifica tecnica v2.0.0</i>	5%	0%
<i>Definizione di prodotto v1.0.0</i>	-3%	8%
Codifica e Debug	-10%	-15%

Tabella 11: RQ - Schedule e budget variance

Si individua una forte anomalia nella budget variance per la codifica e debug. L'avanzo sulle attività di progettazione è sufficiente a mantenere in bilancio il preventivo, ma questo valore è indice di ingenuità significative durante la pianificazione.

4.3.2 Indici di Gulpease

Documento	Valutazione	Esito
<i>Analisi dei requisiti v3.0.0</i>		
<i>Glossario v2.0.0</i>		
<i>Norme di progetto v3.0.0</i>		
<i>Piano di progetto v3.0.0</i>		
<i>Piano di qualifica v3.0.0</i>		
<i>Specifica tecnica v2.0.0</i>		
<i>Definizione di prodotto v1.0.0</i>		

Tabella 12: RQ - Indici di Gulpease calcolati sulla documentazione prodotta

4.3.3 Progettazione

Metrica	Valutazione	Esito
Numero di violazioni delle norme di progettazione		

Tabella 13: RQ - Numero di violazioni delle norme di progettazione

Componente	Accoppiamento afferente
Client::View::MainView	0
Client::View::TitleBarView	1
Client::View::ToolBarView	1
Client::View::AddressView	1
Client::View::EditPanelView	1
Client::View::Paper	1
Client::Model::Command	3
Client::Model::ConcreteCommand	0
Client::Model::State	1
Client::Model::DAO	1
Client::Model::MainModel	4
Client::Model::TitleBarModel	1
Client::Model::ToolBarModel	5
Client::Model::PackageToolbar	0
Client::Model::ClassToolbar	0
Client::Model::ActivityToolbar	0
Client::Model::BubbleToolbar	0
Client::Model::AddressModel	1
Client::Model::EditPanelModel	2
Client::Model::ItemPanel	0
Client::Model::DiagramTree	1
Client::Model::Diagram	5
Client::Model::PackageDiagram	0
Client::Model::ClassDiagram	0
Client::Model::ActivityDiagram	0
Client::Model::BubbleDiagram	0
Client::Model::RequestHandler::Sender	1
Client::Model::RequestHandler::Receiver	1
Server::RequestHandler::Sender	1
Server::RequestHandler::Receiver	1
Server::CodeGenerator::CodeGenerator	1
Server::CodeGenerator::Parser::Parser	2
Server::CodeGenerator::Coder::Coder	2
Server::CodeGenerator::Coder::JavaCoder	0
Server::CodeGenerator::Coder::JavaScriptCoder	0
Server::CodeGenerator::Coder::CoderClass	0

Componente	Accoppiamento afferente
Server::CodeGenerator::Coder::CoderOperation	0
Server::CodeGenerator::Coder::CoderParameter	0
Server::CodeGenerator::Coder::CoderAttribute	0
Server::CodeGenerator::Coder::CoderActivity	0
Server::CodeGenerator::Coder::CodedProg	1
Server::CodeGenerator::Coder::CoderElement	6
Server::CodeGenerator::Builder::Builder	1
Server::CodeGenerator::Zipper::Zipper	1
Server::DAO	1

Tabella 14: RQ - Grado di accoppiamento afferente

Tutti i componenti sono almeno sul range accettabile.

Componente	Accoppiamento efferente
Client::View::MainView	9
Client::View::TitleBarView	0
Client::View::ToolBarView	0
Client::View::AddressView	0
Client::View::EditPanelView	0
Client::View::Paper	1
Client::Model::Command	0
Client::Model::ConcreteCommand	2
Client::Model::State	1
Client::Model::DAO	1
Client::Model::MainModel	1
Client::Model::TitleBarModel	0
Client::Model::ToolBarModel	0
Client::Model::PackageToolbar	1
Client::Model::ClassToolbar	1
Client::Model::ActivityToolbar	1
Client::Model::BubbleToolbar	1
Client::Model::AddressModel	0
Client::Model::EditPanelModel	0
Client::Model::ItemPanel	1
Client::Model::DiagramTree	1
Client::Model::Diagram	1
Client::Model::PackageDiagram	1

Componente	Accoppiamento efferente
Client::Model::ClassDiagram	1
Client::Model::ActivityDiagram	1
Client::Model::BubbleDiagram	1
Client::Model::RequestHandler::Sender	1
Client::Model::RequestHandler::Receiver	1
Server::RequestHandler::Sender	1
Server::RequestHandler::Receiver	1
Server::CodeGenerator::CodeGenerator	3
Server::CodeGenerator::Parser::Parser	0
Server::CodeGenerator::Coder::Coder	2
Server::CodeGenerator::Coder::JavaCoder	1
Server::CodeGenerator::Coder::JavaScriptCoder	1
Server::CodeGenerator::Coder::CoderClass	1
Server::CodeGenerator::Coder::CoderOperation	1
Server::CodeGenerator::Coder::CoderParameter	1
Server::CodeGenerator::Coder::CoderAttribute	1
Server::CodeGenerator::Coder::CoderActivity	2
Server::CodeGenerator::Coder::CodedProg	0
Server::CodeGenerator::Coder::CoderElement	0
Server::CodeGenerator::Builder::Builder	1
Server::CodeGenerator::Zipper::Zipper	1
Server::DAO	0

Tabella 15: RQ - Grado di accoppiamento efferente

Tutti i componenti sono almeno sul range accettabile.

Metriche per la progettazione

Per evitare la creazione di tabelle troppo prolisse e complesse da consultare, sono stati riportati solo i valori minimi e massimi rilevati nelle misurazioni.

Metrica	Range Accettazione	Range Ottimale	04/07/2017	Esito
Fan In			min 0 - max 6	Ottimo
Fan Out			min 0 - max 5	Ottimo

Tabella 16: RQ - Fan In e Fan Out

4.3.4 Codifica

Metrica	Accettazione	Ottimale	04/07/2017	Esito
Numero di violazioni delle norme di codifica	0-10	0-5	4	Ottimale

Tabella 17: RQ - Numero di violazioni delle norme di codifica

Metriche per il codice

Per evitare la creazione di tabelle troppo prolisse e complesse da consultare, sono stati riportati solo i valori minimi e massimi rilevati nelle misurazioni.

Metrica	Accettazione	Ottimale	04/07/2017	Esito
Percentuale totale di test superati	80% - 100%	90% - 100%	96%	Ottimale
Grado di accoppiamento afferente	0 - 7	0 - 5	min 0 - max 6	Accettabile
Grado di accoppiamento efferente	0 - 7	0 - 5	min 0 - max 9	Non Acc.
Linee di commento su linee di codice	≥ 0.25	≥ 0.30	min 26%-max 47%	Accettabile
Numero di parametri	0 - 8	0 - 5	min 0- max 5	Ottimale
Numero di campi dati	0 - 16	0 - 10	min 0 - max 6	Ottimale
Complessità ciclomatica	0 - 10	0 - 6		
Livello di annidamento	0 - 6	0 - 4		
Chiamate innestate di metodi	0 - 6	0 - 4		
Copertura del codice	80% - 100%	90% - 100%		
Numero di linee per metodo	≤ 60	≤ 40	min 3 - max 52	Accettabile
Validazione W3C	0 - 10 (pagina)	0 - 0 (pagina)	min 0 - max 4	Accettabile

Tabella 18: Metriche per qualità di prodotto

A Test

Sono state individuate quattro tipologie di test:

- **Test di unità:** servono alla verifica della correttezza degli algoritmi;
- **Test di integrazione:** servono alla verifica della correttezza delle componenti individuate;
- **Test di sistema:** servono alla verifica del corretto funzionamento dell'architettura e della soddisfazione dei requisiti descritti nell'*Analisi dei requisiti*;
- **Test di validazione:** servono per accertarsi che il prodotto sia conforme con quanto concordato con il Proponente.

La classificazione ed il tracciamento dei test è definito nelle *Norme di progetto v3.0.0*.

A.1 Test di validazione

I test di validazione vengono effettuati con il Proponente e servono per accertarsi che il prodotto realizzato sia conforme alle attese.

Per ogni test è descritta una serie di passi che l'utente deve seguire in modo tale da effettuarlo correttamente.

A.1.1 Test TV1

L'utente vuole verificare che il sistema permetta la creazione di un nuovo progetto. All'utente è richiesto di:

- Comunicare l'intenzione di voler creare un nuovo progetto;
- Inserire un nome per il nuovo progetto;
- Confermare la creazione del progetto;
- Verificare che venga creato e aperto un nuovo progetto vuoto;

A.1.2 Test TV2

L'utente vuole verificare che si possa caricare un progetto precedentemente creato. All'utente è richiesto di:

- Comunicare l'intenzione di voler aprire un progetto precedentemente salvato;
- Selezionare il progetto che intende caricare;
- Confermare il caricamento;
- Verificare che il progetto venga aperto correttamente.

A.1.3 Test TV3

L'utente vuole verificare che si possa salvare il progetto corrente. All'utente è richiesto di:

- Effettuare almeno una modifica al progetto corrente;
- Comunicare l'intenzione di voler salvare il progetto corrente;
- Verificare che il progetto sia stato correttamente salvato.

A.1.4 Test TV3.1

L'utente vuole verificare che si possa salvare il progetto corrente specificandone il nome. All'utente è richiesto di:

- Effettuare almeno una modifica al progetto corrente;
- Comunicare l'intenzione di voler salvare il progetto corrente con un nuovo nome;
- Inserire il nome con cui si intende salvare il progetto;
- Confermare il salvataggio con il nome specificato.
- Verificare che il progetto sia stato correttamente salvato.

A.1.5 Test TV4

L'utente vuole verificare la possibilità di poter annullare un'azione appena eseguita. All'utente è richiesto di:

- Aprire un progetto precedentemente creato (TV2) o crearne uno nuovo (TV1);
- Effettuare almeno una modifica al progetto corrente;
- Comunicare l'intenzione di voler annullare l'ultima azione eseguita;
- Verificare che il progetto si trovi nello stato precedente all'ultima azione eseguita.

A.1.6 Test TV5

L'utente vuole verificare la possibilità di poter ripristinare un'azione appena annullata. All'utente è richiesto di:

- Aprire un progetto precedentemente creato (TV2) o crearne uno nuovo (TV1);
- Effettuare almeno una modifica al progetto;
- Annullare l'ultima azione eseguita (TV6);
- Comunicare l'intenzione di voler ripristinare l'azione appena annullata;
- Verificare che gli effetti dell'azione precedentemente annullata siano stati ripristinati.

A.1.7 Test TV6

L'utente vuole verificare le funzionalità del diagramma dei package aggiungendo un nuovo package. All'utente è richiesto di:

- Aprire un progetto precedentemente creato o crearne uno nuovo;
- Aggiungere un nuovo package al diagramma dei package;
- Inserire il nome e specificare la visibilità per il package;
- Confermare la creazione del package;
- verificare che il package sia presente nel diagramma dei package, e che abbia le proprietà specificate.

A.1.8 Test TV7

L'utente vuole verificare le funzionalità del diagramma dei package modificando un package. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contiene almeno un package o crearne un nuovo nuovo progetto e aggiungere un package;
- Selezionare un package dal diagramma dei package;
- Comunicare di voler modificare il package selezionato;
- Modificare il nome e la visibilità del package;
- Confermare la modifica;
- Verificare che il package selezionato presenti le modifiche apportate.

A.1.9 Test TV8

L'utente vuole verificare le funzionalità del diagramma dei package eliminando un package. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contiene almeno un package o crearne uno nuovo e aggiungere un package;
- Selezionare un package dal diagramma dei package;
- Comunicare l'intenzione di eliminare il package selezionato;
- Confermare la cancellazione del package;
- Verificare che vengano eliminate anche le relazioni associate al package.

A.1.10 Test TV9

L'utente vuole verificare le funzionalità del diagramma dei package aggiungendo una nuova relazione tra package. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno 2 package o crearne uno nuovo e aggiungere 2 package;
- Aggiungere una nuova relazione tra due package al diagramma dei package;

Per fare ciò all'utente è richiesto di:

- Selezionare un primo package;
 - Selezionare un altro package;
 - Selezionare la tipologia di relazione;
 - Inserire molteplicità o eventuali parametri.
- Confermare la creazione della relazione.

A.1.11 Test TV10

L'utente vuole verificare le funzionalità del diagramma dei package apportando una modifica ad una relazione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una relazione tra package o crearne uno nuovo e aggiungere una relazione tra package;
- Selezionare una relazione dal diagramma dei package;
- Comunicare l'intenzione di voler modificare la relazione selezionata;
- Apportare tutte le possibili modifiche alla relazione selezionata;
- Confermare le modifiche alla relazione;
- Verificare che tutte le modifiche apportate siano state applicate correttamente.

A.1.12 Test TV11

L'utente vuole verificare le funzionalità del diagramma dei package eliminando una relazione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una relazione tra package o crearne uno nuovo e aggiungere una relazione tra package;
- Selezionare una relazione dal diagramma dei package;
- Comunicare l'intenzione di voler eliminare la relazione selezionata;
- Confermare la cancellazione della relazione selezionata.

A.1.13 Test TV12

L'utente vuole verificare le funzionalità del diagramma delle classi aggiungendo una nuova classe. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno un package o crearne uno nuovo e aggiungere un package;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Aggiungere una nuova classe al diagramma delle classi;
- Inserire il nome per la classe appena creata;
- Confermare la creazione della classe.

A.1.14 Test TV13

L'utente vuole verificare le funzionalità del diagramma delle classi apportando una modifica ad una classe. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una classe o crearne uno nuovo e aggiungere una classe;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una classe dal diagramma delle classi;
- Comunicare l'intenzione di voler modificare la classe selezionata;
- Apportare tutte le possibili modifiche alla classe selezionata;
- Confermare le modifiche apportate alla classe;
- Verificare che tutte le modifiche apportate siano state applicate correttamente.

A.1.15 Test TV14

L'utente vuole verificare le funzionalità del diagramma delle classi eliminando una classe. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una classe o crearne uno nuovo e aggiungere una classe;
- Selezionare un package al diagramma dei package;
- Selezionare una classe dal diagramma delle classi;
- Comunicare l'intenzione di voler eliminare la classe o premere il tasto Canc;
- Confermare la cancellazione della classe;
- Verificare che vengano eliminate anche le relazioni associate alla classe.

A.1.16 Test TV15

L'utente vuole verificare le funzionalità del diagramma delle classi aggiungendo una nuova interfaccia. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno un package o crearne uno nuovo e aggiungere un package;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Aggiungere una nuova interfaccia al diagramma delle classi;
- Inserire il nome per l'interfaccia appena creata;
- Confermare la creazione dell'interfaccia.

A.1.17 Test TV16

L'utente vuole verificare le funzionalità del diagramma delle classi apportando una modifica ad un' interfaccia. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno un'interfaccia o crearne uno nuovo e aggiungere un'interfaccia;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare un'interfaccia dal diagramma delle classi;
- Comunicare l'intenzione di voler modificare l'interfaccia selezionata;
- Apportare tutte le possibili modifiche all'interfaccia selezionata;
- Confermare le modifiche all'interfaccia.

A.1.18 Test TV17

L'utente vuole verificare le funzionalità del diagramma delle classi eliminando un'interfaccia. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno un'interfaccia o crearne uno nuovo e aggiungere un'interfaccia;
- Selezionare un package al diagramma dei package;
- Selezionare un'interfaccia al diagramma delle classi;
- Comunicare l'intenzione di voler eliminare l'interfaccia selezionata o premendo il tasto Canc;
- Confermare la cancellazione dell'interfaccia
- Verificare che vengano eliminate anche le relazioni associate all'interfaccia.

A.1.19 Test TV18

L'utente vuole verificare le funzionalità del diagramma delle classi aggiungendo una nuova relazione tra classi o interfacce. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno 2 tra classi e/o interfacce nello stesso package o crearne uno nuovo e aggiungere 2 elementi tra classi e/o interfacce;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Aggiungere una nuova relazione tra classi o interfacce; All'utente è richiesto di:
 - Selezionare una prima classe o interfaccia;
 - Selezionare un'altra classe o interfaccia;
 - Comunicare l'intenzione di voler aggiungere una relazione fra i due elementi selezionati;
 - Selezionare la tipologia di relazione;
 - Inserire molteplicità o eventuali parametri.
 - Confermare la creazione della relazione.

A.1.20 Test TV19

L'utente vuole verificare le funzionalità del diagramma delle classi apportando una modifica ad una relazione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una relazione o crearne uno nuovo e creare una relazione;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una relazione dal diagramma delle classi;
- Comunicare l'intenzione di voler modificare la relazione selezionata;
- Apportare modifiche alla relazione;
- Confermare le modifiche alla relazione;
- Verificare che le modifiche apportate siano state applicate correttamente.

A.1.21 Test TV20

L'utente vuole verificare le funzionalità del diagramma delle classi eliminando una relazione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno una relazione o crearne uno nuovo e creare una relazione;

- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una relazione dal diagramma delle classi;
- Comunicare l'intenzione di eliminare la relazione selezionata o premere il tasto **cancel**;
- Confermare la cancellazione della relazione.

A.1.22 Test TV36

L'utente vuole verificare le funzionalità del bubble flowchart aggiungendo una nuova bubble. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga un'attività o crearne uno nuovo e aggiungere un'attività;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una classe e visualizzare il diagramma delle attività relativo a un suo metodo;
- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Aggiungere una nuova bubble al diagramma delle attività;
- Inserire eventuali parametri per la bubble;
- Confermare la creazione della bubble.

A.1.23 Test TV37

L'utente vuole verificare le funzionalità del bubble flowchart apportando una modifica ad una bubble. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga una bubble o crearne uno nuovo e aggiungere una bubble;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una classe e visualizzare il diagramma delle attività relativo a un suo metodo;
- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Selezionare una bubble del bubble flowchart;
- Modificare i parametri della bubble tramite il menù sul lato;
- Confermare le modifiche apportate alla bubble.

A.1.24 Test TV38

L'utente vuole verificare le funzionalità del bubble flowchart eliminando una bubble. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga una bubble o crearne uno nuovo e aggiungere una bubble;
- Selezionare un package al diagramma dei package;
- Selezionare una classe e visualizzare il diagramma delle attività di uno dei suoi metodi;
- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Selezionare una bubble dal bubble flowchart;
- Cancellare la bubble premendo il bottone apposito o il tasto Canc;
- Confermare la cancellazione della bubble.

A.1.25 Test TV39

L'utente vuole verificare le funzionalità del bubble flowchart aggiungendo un nuovo elemento di decisione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga un'attività o crearne uno nuovo e aggiungere un'attività;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una classe e visualizzare il diagramma delle attività relativo a un suo metodo;
- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Aggiungere un nuovo evento temporale al bubble flowchart;
- Inserire il nome e la durata per l'evento temporale appena creato;
- Confermare la creazione dell'elemento di decisione.

A.1.26 Test TV40

L'utente vuole verificare le funzionalità del bubble flowchart apportando una modifica ad un elemento di decisione; All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga almeno un'elemento di decisione o crearne uno nuovo e aggiungere un'elemento di decisione;
- Selezionare un package e visualizzare il relativo diagramma delle classi;
- Selezionare una classe e visualizzare il diagramma delle attività relativo a un suo metodo;

- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Selezionare un'evento temporale del bubble flowchart;
- Modificare almeno un parametro dell'elemento di decisione dal menù al lato;
- Confermare le modifiche apportate all'elemento di decisione.

A.1.27 Test TV41

L'utente vuole verificare le funzionalità del bubble flowchart eliminando un elemento di decisione. All'utente è richiesto di:

- Aprire un progetto precedentemente creato che contenga un'elemento di decisione o crearne uno nuovo e aggiungere un'elemento di decisione;
- Selezionare un package al diagramma dei package;
- Selezionare una classe e visualizzare il diagramma delle attività di uno dei suoi metodi;
- Selezionare un'attività e visualizzare il suo bubble flowchart;
- Selezionare un'elemento di decisione dal bubble flowchart;
- Cancellare l'elemento di decisione premendo il bottone apposito o il pulsante Canc;
- Confermare la cancellazione dell'elemento di decisione.

A.2 Tracciamento test di Validazione

Test di validazione	Requisito
TV1	R0F14.1
TV2	R0F14.2
TV3	R0F14.8
TV3.1	R0F14.8.2
TV4	R0F14.4
TV5	R0F14.5
TV6	R1F18.1
TV7	R1F18.2
TV8	R1F18.3
TV9	R1F18.5
TV10	R0F15.5
TV11	R1F18.6
TV12	R0F15.1
TV13	R0F15.2

Test di validazione	Requisito
TV14	R0F15.3
TV15	R0F15.7
TV16	R0F15.8
TV17	R0F15.9
TV18	R0F15.4
TV19	R0F15.5
TV20	R0F15.6
TV36	R0F20.1
TV37	R0F20.2
TV38	R0F20.3
TV39	R0F20.4
TV40	R0F20.5
TV41	R0F20.6

Tabella 19: Tracciamento test di validazione

A.3 Test di Sistema

Test	Descrizione	Stato
TS1	Verificare che il sistema permetta di generare codice compilabile correttamente	Superato
TS2	Verificare che il sistema permetta di gestire un progetto	Superato
TS2.1	Verificare che il sistema permetta di creare un nuovo progetto	Superato
TS2.1.1	Verificare che il sistema permetta di definire il nome del progetto	Superato
TS2.2	Verificare che il sistema permetta di caricare un progetto	Superato
TS2.3	Verificare che il sistema permetta di chiudere un progetto	Superato
TS2.3.1	Verificare che il sistema permetta, al momento della chiusura, di salvare le modifiche effettuate successivamente all'ultimo salvataggio	Non Superato
TS2.4	Verificare che il sistema permetta di salvare un progetto	Superato
TS2.4.1	Verificare che il sistema permetta di salvare il progetto attuale sovrascrivendolo	Superato
TS2.4.2	Verificare che il sistema permetta di salvare il progetto in una directory scelta dall'utente	Superato
TS3	Verificare che il sistema permetta di editare diagrammi UML	Superato
TS3.1	Verificare che il sistema permetta di editare il diagramma dei package	Superato
TS3.1.1	Verificare che il sistema permetta di creare un nuovo package vuoto nel diagramma dei package	Superato
TS3.1.2	Verificare che il sistema permetta di modificare un package presente nel diagramma dei package	Superato
TS3.1.2.1	Verificare che il sistema permetta di rinominare un package	Superato
TS3.1.2.3	Verificare che il sistema permetta di rimuovere un elemento dal package	Superato
TS3.1.3	Verificare che il sistema permetta di eliminare un package	Superato
TS3.1.4	Verificare che il sistema permetta di passare dal diagramma dei package al diagramma delle classi	Superato
TS3.1.5	Verificare che il sistema permetta di definire una dipendenza tra package	Superato

Test	Descrizione	Stato
TS3.1.6	Verificare che il sistema permetta rimuovere una dipendenza tra package	Superato
TS3.1.7	Verificare che il sistema permetta di riposizionare un elemento all'interno del diagramma dei package	Superato
TS3.2	Verificare che il sistema permetta di editare il diagramma delle classi	Superato
TS3.2.1	Verificare che il sistema permetta di aggiungere una nuova classe al diagramma delle classi	Superato
TS3.2.2	Verificare che il sistema permetta di modificare una classe presente nel diagramma delle classi	Superato
TS3.2.2.1	Verificare che il sistema permetta di rinominare una classe presente nel diagramma delle classi	Superato
TS3.2.2.10	Verificare che il sistema permetta di aggiungere una nuova interfaccia al diagramma delle classi	Superato
TS3.2.2.11	Verificare che il sistema permetta di modificare un'interfaccia presente nel diagramma delle classi	Superato
TS3.2.2.11.1	Verificare che il sistema permetta di rinominare un'interfaccia	Superato
TS3.2.2.11.10	Verificare che il sistema permetta di visualizzare il layer superiore	Superato
TS3.2.2.11.11	Verificare che il sistema permetta di visualizzare il layer inferiore	Superato
TS3.2.2.11.2	Verificare che il sistema permetta di impostare l'importanza di un'interfaccia	Superato
TS3.2.2.11.2.1	Verificare che il sistema permetta di modificare l'importanza di un'interfaccia a "bassa"	Superato
TS3.2.2.11.2.2	Verificare che il sistema permetta di modificare l'importanza di un'interfaccia a "media"	Superato
TS3.2.2.11.2.3	Verificare che il sistema permetta di modificare l'importanza di un'interfaccia a "alta"	Superato
TS3.2.2.11.3	Verificare che il sistema permetta di aprire la schermata di modifica di un'interfaccia	Superato
TS3.2.2.11.4	Verificare che il sistema permetta di eliminare un'interfaccia presente nel diagramma delle classi	Superato
TS3.2.2.11.5	Verificare che il sistema permetta di aggiungere un commento all'interno del diagramma delle classi	Superato
TS3.2.2.11.6	Verificare che il sistema permetta di collegare un commento ad un elemento presente nel diagramma delle classi	Superato

Test	Descrizione	Stato
TS3.2.2.11.7	Verificare che il sistema permetta di modificare un commento presente nel diagramma delle classi	Superato
TS3.2.2.11.8	Verificare che il sistema permetta di eliminare un commento presente nel diagramma delle classi	Superato
TS3.2.2.11.9	Verificare che il sistema permetta di cambiare layer di visualizzazione	Non Superato
TS3.2.2.3	Verificare che il sistema permetta di impostare l'importanza di una classe	Superato
TS3.2.2.4	Verificare che il sistema permetta di modificare l'importanza di una classe	Superato
TS3.2.2.4.1	Verificare che il sistema permetta di modificare l'importanza di una classe a "bassa"	Superato
TS3.2.2.4.2	Verificare che il sistema permetta di modificare l'importanza di una classe a "media"	Superato
TS3.2.2.4.3	Verificare che il sistema permetta di modificare l'importanza di una classe a "alta"	Superato
TS3.2.2.5	Verificare che il sistema permetta di passare alla schermata di modifica di una classe	Superato
TS3.2.2.6	Verificare che il sistema permetta di eliminare una classe presente nel diagramma delle classi	Superato
TS3.2.2.7	Verificare che il sistema permetta di definire una relazione tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.1	Verificare che il sistema permetta di definire una dipendenza tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.2	Verificare che il sistema permetta di definire un'associazione tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.3	Verificare che il sistema permetta di definire un vincolo di ereditarietà tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.4	Verificare che il sistema permetta di definire un'aggregazione tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.5	Verificare che il sistema permetta di definire una composizione tra due elementi presenti nel diagramma delle classi	Superato
TS3.2.2.7.7	Verificare che il sistema permetta di definire la realizzazione di un'interfaccia	Superato

Test	Descrizione	Stato
TS3.2.2.8	Verificare che il sistema permetta di modificare una relazione tra due elementi del diagramma delle classi	Superato
TS3.2.2.8.1	Verificare che il sistema permetta di modificare una dipendenza tra due elementi presenti nel diagramma delle classi	N.I.
TS3.2.2.8.2	Verificare che il sistema permetta di modificare un'associazione tra due elementi presenti nel diagramma delle classi	N.I.
TS3.2.2.8.3	Verificare che il sistema permetta di modificare un vincolo di ereditarietà tra due elementi presenti nel diagramma delle classi	N.I.
TS3.2.2.8.4	Verificare che il sistema permetta di modificare un'aggregazione tra due elementi presenti nel diagramma delle classi	N.I.
TS3.2.2.8.5	Verificare che il sistema permetta di modificare una composizione tra due elementi presenti nel diagramma delle classi	N.I.
TS3.2.2.8.7	Verificare che il sistema permetta di modificare la realizzazione di un'interfaccia	N.I.
TS3.2.2.9	Verificare che il sistema permetta di eliminare una relazione tra due elementi del diagramma delle classi	Superato
TS3.2.3	Verificare che il sistema permetta di modificare una classe mediante la schermata di modifica di una classe	Superato
TS3.2.3.1	Verificare che il sistema permetta di aggiungere un nuovo attributo alla classe	Superato
TS3.2.3.2	Verificare che il sistema permetta di modificare un attributo della classe	Superato
TS3.2.3.2.2	Verificare che il sistema permetta di rinominare l'attributo	Superato
TS3.2.3.2.3	Verificare che il sistema permetta di cambiare il tipo dell'attributo	Superato
TS3.2.3.2.4	Verificare che il sistema permetta di definire un valore di default per l'attributo	Superato
TS3.2.3.3	Verificare che il sistema permetta di eliminare un attributo della classe	Superato
TS3.2.3.4	Verificare che il sistema permetta di aggiungere una nuova operazione alla classe	Superato
TS3.2.3.5	Verificare che il sistema permetta di modificare un'operazione della classe	Superato
TS3.2.3.5.1	Verificare che il sistema permetta di impostare la visibilità dell'operazione	Superato

Test	Descrizione	Stato
TS3.2.3.5.2	Verificare che il sistema permetta di rinominare un'operazione	Superato
TS3.2.3.5.3	Verificare che il sistema permetta di definire la lista dei parametri dell'operazione	Superato
TS3.2.3.5.3.1	Verificare che il sistema permetta di aggiungere un nuovo parametro alla lista	Superato
TS3.2.3.5.3.2	Verificare che il sistema permetta di modificare un parametro della lista	Superato
TS3.2.3.5.3.2.1	Verificare che il sistema permetta di definire la direzione del parametro	Superato
TS3.2.3.5.3.2.2	Verificare che il sistema permetta di rinominare il parametro	Superato
TS3.2.3.5.3.2.3	Verificare che il sistema permetta di definire il tipo del parametro	Superato
TS3.2.3.5.3.2.4	Verificare che il sistema permetta di definire il valore di default del parametro	Superato
TS3.2.3.5.3.3	Verificare che il sistema permetta di eliminare un parametro della lista	Superato
TS3.2.3.5.4	Verificare che il sistema permetta di definire proprietà aggiuntive dell'operazione	Superato
TS3.2.3.6	Verificare che il sistema permetta di eliminare un'operazione	Superato
TS3.2.3.7	Verificare che il sistema permetta di impostare la visibilità della classe	Superato
TS3.2.3.8	Verificare che il sistema permetta di commentare una classe attraverso il collegamento di una classe	Superato
TS3.2.3.9	Verificare che il sistema permetta di passare dalla modifica di una classe al diagramma delle classi	Superato
TS3.2.3.10	Verificare che il sistema permetta di marciare una classe	Superato
TS3.2.3.10.1	Verificare che il sistema permetta di marciare una classe come statica	Superato
TS3.2.3.10.2	Verificare che il sistema permetta di marciare una classe come astratta	Superato
TS3.2.3.10.3	Verificare che il sistema permetta di marciare una classe come finale	Superato
TS3.2.3.10.4	Verificare che il sistema permetta di marciare una classe come frozen	Superato
TS3.2.3.10.5	Verificare che il sistema permetta di marciare una classe come readOnly	Superato

Test	Descrizione	Stato
TS3.2.3.10.6	Verificare che il sistema permetta di marciare una classe come enum	Superato
TS3.2.4	Verificare che il sistema permetta di modificare un'interfaccia mediante la schermata di modifica di un'interfaccia	Superato
TS3.2.4.1	Verificare che il sistema permetta di aggiungere una nuova operazione all'interfaccia	Superato
TS3.2.4.2	Verificare che il sistema permetta di modificare un'operazione dell'interfaccia	Superato
TS3.2.4.2.1	Verificare che il sistema permetta di impostare la visibilità dell'operazione	Superato
TS3.2.4.2.2	Verificare che il sistema permetta di rinominare un'operazione	Superato
TS3.2.4.2.3	Verificare che il sistema permetta di definire la lista dei parametri dell'operazione	Superato
TS3.2.4.2.3.1	Verificare che il sistema permetta di aggiungere un nuovo parametro alla lista	Superato
TS3.2.4.2.3.2	Verificare che il sistema permetta di modificare un parametro della lista	Superato
TS3.2.4.2.3.2.1	Verificare che il sistema permetta di definire la direzione del parametro	Superato
TS3.2.4.2.3.2.2	Verificare che il sistema permetta di rinominare il parametro	N.I.Superato
TS3.2.4.2.3.2.3	Verificare che il sistema permetta di definire il tipo del parametro	Superato
TS3.2.4.2.3.2.4	Verificare che il sistema permetta di definire il valore di default del parametro	Superato
TS3.2.4.2.3.3	Verificare che il sistema permetta di eliminare un parametro della lista	Superato
TS3.2.4.2.4	Verificare che il sistema permetta di definire il tipo di ritorno dell'operazione	Superato
TS3.2.4.2.5	Verificare che il sistema permetta di definire proprietà aggiuntive dell'operazione	Superato
TS3.2.4.2.6	Verificare che il sistema permetta di aprire il bubble flowchart corrispondente	Superato
TS3.2.4.3	Verificare che il sistema permetta di eliminare un'operazione dall'interfaccia	Superato
TS3.2.4.4	Verificare che il sistema permetta di rinominare l'interfaccia	Superato

Test	Descrizione	Stato
TS3.2.4.5	Verificare che il sistema permetta di impostare la visibilità dell'interfaccia	Superato
TS3.2.4.6	Verificare che il sistema permetta di marciare l'interfaccia	Superato
TS3.2.4.7	Verificare che il sistema permetta di passare dalla modifica di un'interfaccia al diagramma delle classi	Superato
TS3.2.4.8	Verificare che il sistema permetta di commentare l'interfaccia	Superato
TS3.3	Verificare che il sistema permetta di editare il bubble diagram	Superato
TS3.3.1	Verificare che il sistema permetta di aggiungere un nuovo elemento nel bubble diagram	Superato
TS3.3.2	Verificare che il sistema permetta di modificare un'elemento presente nel bubble diagram	Superato
TS3.3.2.1	Verificare che il sistema permetta di rinominare una custom bubble	Non Superato
TS3.3.2.2	Verificare che il sistema permetta di commentare una bubble	Superato
TS3.3.5	Verificare che il sistema permetta di aggiungere un nuovo elemento di decisione al bubble diagram	Superato
TS3.3.6	Verificare che il sistema permetta di modificare un elemento di decisione del bubble diagram	Superato
TS3.3.7	Verificare che il sistema permetta di eliminare un elemento di decisione del bubble diagram	Superato
TS3.3.11	Verificare che il sistema permetta di riposizionare un elemento all'interno del bubble diagram	Superato
TS3.3.18	Verificare che il sistema permetta di passare dal bubble diagram al diagramma delle classi	Superato
TS3.3.19	Verificare che il sistema permetta di lavorare con custom bubble sul bubble digram	Superato
TS3.3.19.1	Verificare che il sistema permetta di aggiungere una nuova custom bubble al bubble diagram	Superato
TS3.3.19.3	Verificare che il sistema permetta di eliminare una custom bubble presente nel bubble diagram	Superato
TS3.3.19.3	Verificare che il sistema permetta di modificare una custom bubble presente nel bubble diagram	Superato
TS3.3.19.3	Verificare che il sistema permetta di innestare una bubble presente nel bubble diagram all'interno di un'altra	Superato

Test	Descrizione	Stato
TS3.3.19.3	Verificare che il sistema permetta di disinnestare una bubble innestata all'interno di un'altra	Superato
TS5	Verificare che il sistema permetta di gestire il codice generato	Superato
TS5.1	Verificare che il sistema permetta di leggere il codice generato	Superato
TS5.2	Verificare che il sistema permetta di esportare il codice generato	Superato

Tabella 20: Test di sistema

A.4 Tracciamento test di sistema

Test di sistema	Requisito
TS2	R0F14
TS2.1	R0F14.1
TS2.1.1	R0F14.1.1
TS2.2	R0F14.2
TS2.3	R0F14.3
TS2.3.1	R0F14.3.1
TS2.4	R0F14.8
TS2.4.1	R0F14.8.1
TS2.4.2	R0F14.8.2
TS3	R0F1
TS3.1	R1F18
TS3.1.1	R1F18.1
TS3.1.2	R1F18.2
TS3.1.2.1	R1F18.2.1
TS3.1.2.3	R1F18.2.4
TS3.1.3	R1F18.3
TS3.1.4	R1F18.4
TS3.1.5	R1F18.5
TS3.1.6	R1F18.6
TS3.1.7	R1F18.7
TS3.2	R0F15
TS3.2.1	R0F15.1
TS3.2.2	R0F15.2
TS3.2.2.1	R0F15.2.1
TS3.2.2.10	R0F15.7
TS3.2.2.11	R0F15.8
TS3.2.2.11.1	R0F15.8.1
TS3.2.2.11.10	R0F15.14.1
TS3.2.2.11.11	R0F15.14.2
TS3.2.2.11.2	R0F15.8.2
TS3.2.2.11.2.1	R0F15.8.2.1
TS3.2.2.11.2.2	R0F15.8.2.2
TS3.2.2.11.2.3	R0F15.8.2.3
TS3.2.2.11.3	R0F15.8.3
TS3.2.2.11.4	R0F15.9

Test di sistema	Requisito
TS3.2.2.11.5	R0F15.10
TS3.2.2.11.6	R0F15.11
TS3.2.2.11.7	R0F15.12
TS3.2.2.11.8	R0F15.13
TS3.2.2.11.9	R0F15.14
TS3.2.2.3	R0F15.2.3
TS3.2.2.4	R0F15.2.4
TS3.2.2.4.1	R0F15.2.4.1
TS3.2.2.4.2	R0F15.2.4.2
TS3.2.2.4.3	R0F15.2.4.3
TS3.2.2.5	R0F15.2.5
TS3.2.2.6	R0F15.3
TS3.2.2.7	R0F15.4
TS3.2.2.7.1	R0F15.4.1
TS3.2.2.7.2	R0F15.4.2
TS3.2.2.7.3	R0F15.4.3
TS3.2.2.7.4	R0F15.4.4
TS3.2.2.7.5	R0F15.4.5
TS3.2.2.7.7	R0F15.4.7
TS3.2.2.8	R0F15.5
TS3.2.2.8.1	R0F15.5.1
TS3.2.2.8.2	R0F15.5.2
TS3.2.2.8.3	R0F15.5.3
TS3.2.2.8.4	R0F15.5.4
TS3.2.2.8.5	R0F15.5.5
TS3.2.2.8.7	R0F15.5.7
TS3.2.2.9	R0F15.6
TS3.2.3	R0F16
TS3.2.3.1	R0F16.1
TS3.2.3.2	R0F16.2
TS3.2.3.2.2	R0F16.2.2
TS3.2.3.2.3	R0F16.2.3
TS3.2.3.2.4	R0F16.2.4
TS3.2.3.3	R0F16.3
TS3.2.3.4	R0F16.4
TS3.2.3.5	R0F16.5

Test di sistema	Requisito
TS3.2.3.5.1	R0F16.5.1
TS3.2.3.5.2	R0F16.5.2
TS3.2.3.5.3	R0F16.5.3
TS3.2.3.5.3.1	R0F16.5.3.1
TS3.2.3.5.3.2	R0F16.5.3.2
TS3.2.3.5.3.2.1	R0F16.5.3.2.1
TS3.2.3.5.3.2.2	R0F16.5.3.2.2
TS3.2.3.5.3.2.3	R0F16.5.3.2.3
TS3.2.3.5.3.2.4	R0F16.5.3.2.4
TS3.2.3.5.3.3	R0F16.5.3.3
TS3.2.3.5.4	R0F16.5.4
TS3.2.3.6	R0F16.6
TS3.2.3.7	R0F16.7
TS3.2.3.8	R0F16.8
TS3.2.3.9	R0F16.9
TS3.2.3.10	R0F16.10
TS3.2.3.10.1	R0F16.10.1
TS3.2.3.10.2	R0F16.10.2
TS3.2.3.10.3	R0F16.10.3
TS3.2.3.10.4	R0F16.10.4
TS3.2.3.10.5	R0F16.10.5
TS3.2.3.10.6	R0F16.10.6
TS3.2.4	R0F17
TS3.2.4.1	R0F17.1
TS3.2.4.2	R0F17.2
TS3.2.4.2.1	R0F17.2.1
TS3.2.4.2.2	R0F17.2.2
TS3.2.4.2.3	R0F17.2.3
TS3.2.4.2.3.1	R0F17.2.3.1
TS3.2.4.2.3.2	R0F17.2.3.2
TS3.2.4.2.3.2.1	R0F17.2.3.2.1
TS3.2.4.2.3.2.2	R0F17.2.3.2.2
TS3.2.4.2.3.2.3	R0F17.2.3.2.3
TS3.2.4.2.3.2.4	R0F17.2.3.2.4
TS3.2.4.2.3.3	R0F17.2.3.3
TS3.2.4.2.4	R0F17.2.4

Test di sistema	Requisito
TS3.2.4.2.5	R0F17.2.5
TS3.2.4.2.6	R0F17.2.6
TS3.2.4.3	R0F17.3
TS3.2.4.4	R0F17.4
TS3.2.4.5	R0F17.5
TS3.2.4.6	R0F17.6
TS3.2.4.7	R0F17.7
TS3.2.4.8	R0F17.8
TS3.3	R0F19
TS3.3.1	R0F19.1
TS3.3.2	R0F19.2
TS3.3.2.1	R0F19.2.1
TS3.3.2.2	R0F19.2.2
TS3.3.5	R0F19.4
TS3.3.6	R0F19.5
TS3.3.7	R0F19.6
TS3.3.11	R0F19.10
TS3.3.18	R0F19.17
TS3.3.19	R0F20
TS3.3.19.1	R0F20.1
TS3.3.19.2	R0F20.2
TS5	R0F22
TS5.1	R0F14.6
TS5.2	R0F14.7

Tabella 21: Tracciamento test di sistema

Nella seguente tabella sono riportate le percentuali di successo e implementazione per i test di sistema.

Numero di Test	Percentuale Successo	Percentuale Implementati
142	94%	96%

Tabella 22: Stato implementazione test sistema

A.5 Test di integrazione

In questa sezione vengono descritti i test di integrazione, da utilizzare per testare i vari componenti descritti nella progettazione ad alto livello, che permettono di verificare la corretta integrazione ed il corretto flusso dei dati all'interno del sistema.

Si è deciso di utilizzare una strategia di integrazione incrementale che permette di verificare più componenti in parallelo.

Seguendo questa strategia eventuali difetti rilevati da un test sono da attribuirsi, con maggior probabilità, all'ultima parte aggiunta; ciò permette, a patto di tenere i vari passi reversibili, di retrocedere verso uno stato noto e sicuro.

È stato utilizzato il metodo bottom-up per poter integrare prima le parti con minore dipendenza funzionale e maggiore funzionalità che corrispondono ai componenti per requisiti obbligatori, in questo modo è possibile avere una versione funzionante delle parti obbligatorie dell'applicazione il prima possibile.

Inoltre con questo metodo i componenti legati a parti obbligatorie vengono testate ad ogni integrazione, diminuendo la possibilità che si presentino errori in essi.

Il seguente diagramma, che non segue il formalismo UML, è utilizzato per semplificare la spiegazione della strategia di integrazione.

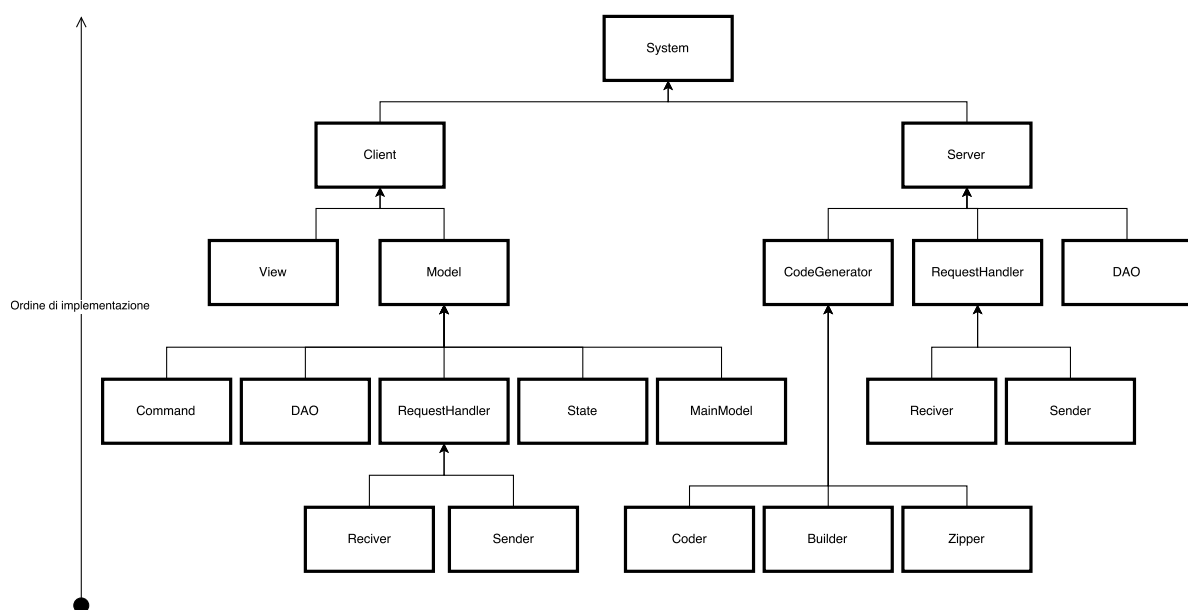


Figura 1: Diagramma test di integrazione

TI1	Viene verificata l'integrazione finale per le componenti del sistema, in particolare tra Server e Client.	SweDesigner.	Non implementato.
TI1.1	Viene verificata l'integrazione finale per le componenti del Client, in particolare tra Model e View.	Client.	Non implementato.

TI1.1.1	Viene verificato che il sistema gestisca correttamente le componenti della View in particolare l'integrazione tra la TitlebarView, la ToolbarView, l'PathView, l'EditPanelView e la ProjectView.	View.	Non implementato.
TI1.1.2	Viene verificato che il sistema gestisca correttamente le componenti della View in particolare l'integrazione dei vari moduli con la libreria esterna JointJS _G .	View.	Non implementato.
TI1.1.3	Viene verificata l'integrazione per le componenti del Model, in particolare il DataManager, RequestHendler, Project, ProjectModel e ToolbarModel.	Model.	Non implementato.
TI1.1.3.2	Viene verificato che il DataManager all'interno del Model carichi correttamente un progetto.	DataManager.	Superato.
TI1.1.3.3	Viene verificato che il DataManager all'interno del Model salvi correttamente un progetto.	DataManager.	Superato.
TI1.1.3.4.1	Viene verificato che il RequestHandler all'interno del Client invii le richieste per bubble correttamente al Server.	RequestHandler.	Superato.
TI1.1.3.4.2	Viene verificato che il RequestHandler all'interno del Client invii correttamente al Server le richieste per la creazione di codice a partire dal file JSON _G .	RequestHandler.	Superato.
TI1.1.3.5	Viene verificato che il RequestHandler all'interno del Client riceva correttamente le bubble dal Server.	RequestHandler.	Superato.
TI1.1.3.6	Viene verificata l'integrazione all'interno del Client con i vari moduli del Model collegati ai moduli della View, in particolare con il Project, il ProjectModel, il DataManager, il RequestHandler e la ToolbarModel.	Client.	Superato.
TI1.2	Viene verificata l'integrazione per le componenti del Server, in particolare RequestHandler, DAO e CodeGenerator.	Server.	Superato.
TI1.2.1.1	Viene verificata l'integrazione tra RequestHandler del server e il DAO.	RequestHandler.	Non Implementato.

TI1.2.1.2	Viene verificato che il RequestHandler all'interno del Server invii correttamente al Client il file .zip contenente il codice generato.	RequestHandler.	Non implementato.
TI1.2.1.3	Viene verificato che il RequestHandler all'interno del Server riceva correttamente dal Client il file JSON necessario alla generazione del codice.	RequestHandler.	Non implementato.
TI1.2.2	Viene verificato che il DAO all'interno del Server restituisca correttamente una bubble.	DAO.	Non Implementato.
TI1.2.3	Viene verificato che il DAO all'interno del Server salvi correttamente una bubble.	DAO.	Non Implementato.
TI1.2.4	Viene verificata l'integrazione per le componenti del CodeGenerator del Server, in particolare Coder, Builder e Zipper.	CodeGenerator.	Superato.
TI1.2.5	Viene verificata l'integrazione per le componenti del CodeGenerator del Server con il metodo Javascript JSON.parse().	CodeGenerator.	Superato.
TI1.2.6	Viene verificato che il Coder all'interno del Server costruisca correttamente i file a partire dagli oggetti costruiti tramite il metodo JSON.parse().	Coder.	Superato.
TI1.2.7	Viene verificato che il Builder all'interno del Server organizzi i file in cartelle seguendo le istruzioni contenute nel file JSON inviato dal Client.	Builder.	Superato.
TI1.2.8	Viene verificato che lo Zipper del Server crei correttamente il file .zip a partire da file sorgenti e cartelle generati.	Zipper.	Superato

A.6 Tracciamento test di integrazione

Test di integrazione	Componente
TI1	SweDesigner
TI1.1	SweDesigner::Client
TI1.1.1	SweDesigner::Client::View
TI1.1.2	SweDesigner::Client::View
TI1.1.3	SweDesigner::Client::Model
TI1.1.3.2	SweDesigner::Client::Model::DataManager
TI1.1.3.3	SweDesigner::Client::Model::DataManager
TI1.1.3.4.1	SweDesigner::Client::Model::RequestHandler
TI1.1.3.4.2	SweDesigner::Client::Model::RequestHandler
TI1.1.3.5	SweDesigner::Client::Model::RequestHandler
TI1.1.3.6	SweDesigner::Client
TI1.2	SweDesigner::Server
TI1.2.1.1	SweDesigner::Server::RequestHandler
TI1.2.1.2	SweDesigner::Server::RequestHandler
TI1.2.1.3	SweDesigner::Server::RequestHandler
TI1.2.2	SweDesigner::Server::DAO
TI1.2.3	SweDesigner::Server::DAO
TI1.2.4	SweDesigner::Server
TI1.2.5	SweDesigner::Server::CodeGenerator
TI1.2.6	SweDesigner::Server::CodeGenerator::Coder
TI1.2.7	SweDesigner::Server::CodeGenerator::Builder
TI1.2.8	SweDesigner::Server::CodeGenerator::Zipper

Tabella 24: Tracciamento test di integrazione

Nella seguente tabella sono riportate le percentuali di successo e implementazione per i test di integrazione.

Numero di Test	Percentuale Successo	Percentuale Implementati
27	100%	100%

Tabella 25: Stato implementazione test integrazione

A.7 Test di unità

In questa sezione vengono descritti i test di unità, da utilizzare per testare che ogni singola unità funzioni correttamente; per unità intendiamo la più piccola quantità di software che conviene verificare da sola.

Test	Descrizione	Classe	Stato
TU1	Si vuole verificare che l'unità DataManager sia in grado di salvare il progetto corrente e di aprire un progetto esistente.	DataManager	Superato
TU2	Si vuole verificare che l'unità Project sia in grado di restituire una operazione o una classe presente nel progetto, dato l'identificativo dell'operazione/classe in input; inoltre, si verifica che sia possibile eliminare una operazione o una classe presente nel progetto, dato l'identificativo dell'operazione/classe in input.	Project	Superato
TU3	Si vuole verificare che l'unità ProjectModel sia in grado di creare e gestire il modello di un progetto.	ProjectModel	Superato
TU4	Si vuole verificare che l'unità TitlebarModel sia in grado di creare e gestire il modello di una titlebar	TitlebarModel	Superato
TU5	Si vuole verificare che l'unità ToolbarModel sia in grado di creare e gestire il modello di una toolbar.	ToolbarModel	Superato
TU7	Viene verificato che vengano richiamati i giusti metodi per la creazione del codice Java partendo da un'array contenente i giusti campi	JavaCoder	Superato
TU8	Si vuole verificare che l'unità ProjectView sia in grado di visualizzare un modello di progetto.	ProjectView	Superato
TU9	Si vuole verificare che l'unità TitlebarView sia in grado di visualizzare il modello di una titlebar.	TitlebarView	Superato
TU10	Si vuole verificare che l'unità ToolbarView sia in grado di visualizzare il modello di una toolbar.	ToolbarView	Superato
TU11	Si vuole verificare che l'unità EditPanelView sia in grado di visualizzare il pannello per editare i diagrammi.	EditPanelView	Superato

TU12	Si vuole verificare che l'unità EditPanelView sia in grado di visualizzare il pannello per editare i diagrammi.	EditPanelView	Superato
TU13	Si vuole verificare che l'unità Parser sia in grado di produrre un oggetto contenente tutte le informazioni del file JSON di input, necessarie a generare un programma.	Parser	Superato
TU14	Si vuole verificare che l'unità JavaCoder sia in grado di produrre un oggetto CodedProgram, e che contenga il codice sorgente in Java corrispondente all'oggetto in input, il quale viene restituito dal Parser.	JavaCoder	Superato
TU15	Si vuole verificare che l'unità JavascriptCoder sia in grado di produrre un oggetto SWEDesigner::Server::CodeGenerator::Coder::CodedProgram, e che contenga il codice sorgente in Javascript corrispondente all'oggetto in input, il quale viene restituito da Parser.	JavascriptCoder	Superato
TU16	Si vuole verificare che l'unità CodedProgram permetta di memorizzare le informazioni per produrre il contenuto di un file del programma; inoltre tali informazioni devono poter essere restituite.	CodedProgram	Superato
TU17	Si vuole verificare che la unità CoderClass sia in grado di produrre il codice sorgente, in Java o Javascript, relativo all'istestazione di una classe le cui informazioni sono contenute nell'oggetto di input.	CoderClass	Superato
TU18	Si vuole verificare che la unità CoderAttribute sia in grado di produrre il codice sorgente, in Java o Javascript, relativo all'attributo di una classe le cui informazioni sono contenute nell'oggetto di input.	CoderAttribute	Superato
TU19	Si vuole verificare che la unità CoderOperation sia in grado di produrre il codice sorgente, in Java o Javascript, relativo all'istestazione del metodo/operazione le cui informazioni sono contenute nell'oggetto di input.	CoderOperation	Superato

TU20	Si vuole verificare che la unità <code>CoderParameter</code> sia in grado di produrre il codice sorgente, in Java o Javascript, relativo al parametro del metodo/funzione le cui informazioni sono contenute nell'oggetto di input.	<code>CoderParameter</code>	Superato
TU21	Si vuole verificare che la unità <code>CoderActivity</code> sia in grado di produrre il codice sorgente, in Java o Javascript, relativo all'implementazione di un metodo/funzione le cui informazioni sono contenute nell'oggetto di input.	<code>CoderActivity</code>	Superato
TU22	Si vuole verificare che l'unità <code>Builder</code> sia in grado di creare una directory su disco, contenente il programma codificato e organizzato secondo le informazioni contenute nel <code>CodedProgram</code> di input.	<code>Builder</code>	Superato
TU23	i vuole verificare che l'unità <code>Zipper</code> sia in grado di creare un pacchetto in formato zip su disco, corrispondente alla directory specificata dal path in input.	<code>Zipper</code>	Superato
TU24	Si vuole verificare che l'unità <code>RequestHandler</code> sia in grado di gestire una richiesta di generazione di codice sorgente, restituendo un pacchetto in formato zip contenente il programma, corrispondente al contenuto del file JSON associato alla richiesta.	<code>RequestHandler</code>	Superato
TU25	Si vuole verificare che l'unità <code>DAO</code> sia in grado di restituire le informazioni di una bubble memorizzata nel database.	<code>DAO</code>	Superato
TU26	Si vuole verificare che l'unità <code>Package</code> sia in grado di creare il modello per un package del diagramma.	<code>Package</code>	Superato
TU27	Si vuole verificare che l'unità <code>PkgComment</code> sia in grado di creare il modello per un commento di package del diagramma.	<code>PkgComment</code>	Superato
TU28	Si vuole verificare che l'unità <code>PkgCommentLink</code> sia in grado di creare il modello per il collegamento fra un package e un commento del diagramma.	<code>PkgCommentLink</code>	Superato

TU29	Si vuole verificare che l'unità PkgDependencies sia in grado di creare il modello per una dipendenza tra package del diagramma.	PkgDependencies	Superato
TU30	Si vuole verificare che l'unità Class sia in grado di creare il modello per una classe del diagramma.	Class	Superato
TU31	Si vuole verificare che l'unità Interface sia in grado di creare il modello per un'interfaccia del diagramma.	Interface	Superato
TU32	Si vuole verificare che l'unità ClComment sia in grado di creare il modello per un commento di classe del diagramma.	ClComment	Superato
TU33	Si vuole verificare che l'unità ClCommentLink sia in grado di creare il modello per il collegamento fra un commento ed una classe del diagramma.	ClCommentLink	Superato
TU34	Si vuole verificare che l'unità Generalization sia in grado di creare il modello per una generalizzazione tra due componenti UML del diagramma.	Generalization	Superato
TU35	Si vuole verificare che l'unità Association sia in grado di creare il modello per una associazione tra due componenti UML del diagramma.	Implementation	Superato
TU36	Si vuole verificare che l'unità Implementation sia in grado di creare il modello per una implementazione tra due componenti UML del diagramma.	Implementation	Superato
TU37	Si vuole verificare che l'unità Aggregation sia in grado di creare il modello per una aggregazione tra due componenti UML del diagramma.	Aggregation	Superato
TU38	Si vuole verificare che l'unità Composition sia in grado di creare il modello per una composizione tra due componenti UML del diagramma.	Composition	Superato
TU39	Si vuole verificare che l'unità bubbleLink sia in grado di creare il modello per un collegamento tra elementi del diagramma delle bubble.	bubbleLink	Superato

TU40	Si vuole verificare che l'unità customBubble sia in grado di creare il modello per una bubble il cui contenuto è impostato dall'utente.	customBubble	Superato
TU41	Si vuole verificare che l'unità bubbleIf sia in grado di creare il modello per una bubble che rappresenta l'istruzione condizionale 'if'.	bubbleIf	Superato
TU42	Si vuole verificare che l'unità bubbleElse sia in grado di creare il modello per una bubble che rappresenta l'istruzione condizionale 'else'.	bubbleElse	Superato
TU43	Si vuole verificare che l'unità bubbleFor sia in grado di creare il modello per una bubble che rappresenta l'istruzione condizionale 'for'.	bubbleFor	Superato
TU44	Si vuole verificare che l'unità bubbleReturn sia in grado di creare il modello per una bubble che rappresenta l'istruzione per il ritorno di un valore.	bubbleReturn	Superato
TU45	Si vuole verificare che l'unità bubbleStart sia in grado di creare il modello per una bubble che rappresenta il punto da cui inizia l'implementazione dell'attività.	bubbleStart	Superato
TU46	Si vuole verificare che l'unità bubbleWhile sia in grado di creare il modello per una bubble che rappresenta l'istruzione condizionale 'while'.	bubbleWhile	Superato

Tabella 26: Test di Unità

A.8 Tracciamento test di unità

Test di unità	Componente
TU1	SWEDesigner::Client::Model::DataManager
TU2	SWEDesigner::Client::Model::Project
TU3	SWEDesigner::Client::Model::ProjectModel
TU4	SWEDesigner::Client::Model::TitlebarModel
TU5	SWEDesigner::Client::Model::ToolbarModel
TU7	SWEDesigner::Client::PathView
TU8	SWEDesigner::Client::ProjectView
TU9	SWEDesigner::Client::TitlebarView

Test di unità	Componente
TU10	SWEDesigner::Client::ToolBarView
TU11	SWEDesigner::Client::EditPanelView
TU12	SWEDesigner::Server::CodeGenerator::codeGenerator
TU13	SWEDesigner::Server::CodeGenerator::Parser::Parser
TU14	SWEDesigner::Server::CodeGenerator::Coder::JavaCoder
TU15	SWEDesigner::Server::CodeGenerator::Coder::JavascriptCoder
TU16	SWEDesigner::Server::CodeGenerator::Coder::CodedProgram
TU17	SWEDesigner::Server::CodeGenerator::Coder::CoderElement::CoderClass
TU18	SWEDesigner::Server::CodeGenerator::Coder::CoderElement::CoderAttribute
TU19	SWEDesigner::Server::CodeGenerator::Coder::CoderElement::CoderOperation
TU20	SWEDesigner::Server::CodeGenerator::Coder::CoderElement::CoderParameter
TU21	SWEDesigner::Server::CodeGenerator::Coder::CoderElement::CoderActivity
TU22	SWEDesigner::Server::CodeGenerator::Builder::Builder
TU23	SWEDesigner::Server::CodeGenerator::Zipper::Zip
TU24	SWEDesigner::Server::RequestHandler::RequestHandler
TU25	SWEDesigner::Server::DAO::DAO
TU26	SWEDesigner::Client::Model::Items::Package
TU27	SWEDesigner::Client::Model::Items::PkgComment
TU28	SWEDesigner::Client::Model::Items::PkgCommentLink
TU29	SWEDesigner::Client::Model::Items::PkgDependencies
TU30	SWEDesigner::Client::Model::Items::Class
TU31	SWEDesigner::Client::Model::Items::Interface
TU32	SWEDesigner::Client::Model::Items::ClComment
TU33	SWEDesigner::Client::Model::Items::ClCommentLink -
TU34	SWEDesigner::Client::Model::Items::Generalization
TU35	SWEDesigner::Client::Model::Items::Association
TU36	SWEDesigner::Client::Model::Items::Implementation
TU37	WEDesigner::Client::Model::Items::Aggregation
TU38	SWEDesigner::Client::Model::Items::Composition
TU39	SWEDesigner::Client::Model::Items::bubbleLink
TU40	SWEDesigner::Client::Model::Items::customBubble
TU41	SWEDesigner::Client::Model::Items::bubbleIf
TU42	SWEDesigner::Client::Model::Items::bubbleElse
TU43	SWEDesigner::Client::Model::Items::bubbleFor
TU44	SWEDesigner::Client::Model::Items::bubbleReturn
TU45	SWEDesigner::Client::Model::Items::bubbleStart

Test di unità	Componente
TU46	SWEDesigner::Client::Model::Items::bubbleWhile

Tabella 27: Tracciamento test di Unità

Nella seguente tabella sono riportate le percentuali di successo e implementazione per i test di unità.

Numero di Test	Percentuale Successo	Percentuale Implementati
46	100%	100%

Tabella 28: Stato implementazione test unità

A.9 Resoconto stato test

Di seguito sono riportati le percentuali di successo e di implementazione dei test, prima divisi per categorie, seguiti da un resoconto totale.

Tipo di Test	numero di Test	Percentuale Successo	Percentuale Implementati
Test di integrazione	27	100%	100%
Test di sistema	142	94%	96%
Test di unità	46	100%	100%
Totale	215	96%	98%

Tabella 29: Riassunto stato implementazione test

B Capability Maturity Model

Il Capability Maturity Model (CMM), divenuto CMMI (I per Integration), è un modello per il miglioramento dei processi di sviluppo del software il cui obiettivo è di aiutare un'organizzazione a migliorare le sue prestazioni in termini di qualità del software prodotto, produttività dell'organizzazione e riduzione dei tempi di sviluppo.

Le lettere dell'acronimo indicano:

- **Capability:** si indica la misura di quanto un singolo processo è adeguato allo scopo per il quale è stato definito e determina il range del risultato raggiungibile utilizzando quel processo in termini di efficienza ed efficacia;
- **Maturity:** si indica la misura che comunica quanto è governato l'insieme di processi aziendali ed è influenzata dalle capability dei processi coinvolti;
- **Model:** si indica l'insieme dei requisiti che vanno ad essere sempre più stringenti per valutare il miglioramento dei processi aziendali;
- **Integration:** si indica l'architettura di integrazione delle diverse discipline (system, hardware, software) e tipologie di attività delle aziende.

B.1 Struttura

Il modello coinvolge cinque aspetti:

- **Livelli di maturità:** un processo di continua maturità a cinque livelli dove il più alto indica lo stato ideale dove i processi sono sistematicamente gestiti da una combinazione di ottimizzazione e continuo miglioramento di processi;
- **Aree chiave di processo:** un'area chiave di processo identifica un insieme di attività che raggiungono obiettivi ritenuti importanti quando svolte assieme;
- **Obiettivi:** gli obiettivi di un'area chiave di processo indicano lo scopo, i confini e le intenzioni di quest'ultima;
- **Caratteristiche comuni:** includono pratiche che implementano e istituzionalizzano un'area chiave di processo; sono suddivise in cinque tipi: impegno nell'operare, abilità nell'operare, attività svolte, misurazioni ed analisi, verifica dell'implementazione;
- **Pratiche chiave:** descrivono gli elementi di infrastruttura e pratiche che contribuiscono maggiormente nell'implementazione ed istituzionalizzazione dell'area.

B.2 Livelli

I livelli di maturità sopra citati sono i seguenti:

1. **Iniziale:** i processi sono imprevedibili, insufficientemente controllati e reattivi; risultano non documentati e perciò non sono ripetibili;
2. **Ripetibile:** i processi sono sufficientemente documentati tanto da renderli ripetibili;

3. **Definito**: i processi sono definiti, documentati e ripetibili;
4. **Gestito**: i processi sono controllati e gestiti attraverso analisi e utilizzo di metriche concordate;
5. **Efficiente**: la gestione dei processi punta al loro miglioramento/ottimizzazione.

C Ciclo di Deming

Il ciclo di Deming o ciclo PDCA (Plan-Do-Check-Act) è un metodo di gestione iterativo focalizzato sul miglioramento continuo dei processi; l'acronimo "PDCA" definisce i passi in cui è diviso:

- Pianificare (**Plan**): definizione delle attività e dei processi da migliorare secondo misurazioni effettuate nonché scadenze e risorse utili al raggiungimento del miglioramento;
- Fare (**Do**): attuazione delle azioni pianificate al passo precedente con conseguente misurazione e raccolta di dati utili ai passi successivi;
- Verificare (**Check**): studio dei risultati misurati e raccolti nel passo precedente e confronto con i risultati attesi;
- Agire (**Act**): standardizzazione dei cambiamenti apportati nell'esecuzione di processo (solamente se l'esito del passo precedente è positivo).

Per poter applicare il ciclo PDCA è necessario che i processi siano documentati, analizzabili e ripetibili per poter individuare gli eventuali errori da correggere.

D ISO/IEC 9126

La normativa [ISO/IEC 9126] si suddivide in:

- Modello della qualità del software;
- Metriche per la qualità esterna;
- Metriche per la qualità interna;
- Metriche per la qualità in uso.

D.1 Modello della qualità

D.1.1 Qualità esterna ed interna

Il modello di qualità stabilito dallo standard è classificato da sei caratteristiche generali:

- **Funzionalità**: capacità del prodotto di fornire l'insieme di funzioni per soddisfare le richieste e gli obiettivi dell'utente;

- **Affidabilità:** capacità del prodotto di mantenere un certo livello di prestazioni quando utilizzato in particolari condizioni in un periodo temporale definito;
- **Efficienza:** capacità del prodotto di fornire determinate prestazioni in relazione alla quantità di risorse utilizzate;
- **Usabilità:** capacità del prodotto di essere capito e usato dall'utente in specifiche condizioni;
- **Manutenibilità:** capacità del software di poter essere modificabile, correggendolo, migliorandolo o adattandolo;
- **Portabilità:** capacità del software di essere trasportato da un ambiente di lavoro ad un altro.

Sono presenti anche varie sotto-caratteristiche misurabili attraverso metriche.

D.1.2 Qualità in uso

La qualità in uso rappresenta la qualità del prodotto software dal punto di vista dell'utente ed è classificata da quattro caratteristiche:

- **Efficacia:** capacità del software di mettere in grado gli utenti di raggiungere i loro obiettivi con accuratezza e completezza;
- **Produttività:** capacità di mettere in grado gli utenti di utilizzare una quantità di risorse relativamente all'efficacia ottenuta in uno specifico contesto d'uso;
- **Soddisfazione:** capacità del prodotto di soddisfare gli utenti;
- **Sicurezza:** capacità del prodotto di raggiungere accettabili livelli di rischio di danni a persone, software, strumenti o all'ambiente operativo.

D.2 Metriche per la qualità

D.2.1 Esterna

Le metriche esterne misurano i comportamenti del software rilevati da test, operatività e osservazione durante la sua esecuzione in un contesto tecnico rilevante.

D.2.2 Interna

Le metriche interne sono applicate a software non eseguibile (codice sorgente) e documentazione durante la progettazione e la codifica. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale data l'influenza degli attributi interni su quelli esterni e quelli in uso. Le metriche interne permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato effettivamente il codice eseguibile.