



Piano di Qualifica

Informazioni sul documento

Versione	2.0.0
Redazione	Alberto De Agostini Andrea Giacomo Baldan Francesco Agostini
Verifica	Davide Trevisan Marco Boseggia Michael Munaro
Approvazione	Marco Boseggia
Uso	Esterno
Lista di Distribuzione	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
2.0.0	Marco Boseggia	Responsabile	2016-04-10	Approvazione documento
1.3.0	Michael Munaro	Verificatore	2016-04-05	Verifica incremento consuntivo
1.2.1	Andrea Giacomo Baldan	Analista	2016-04-05	Incremento sezione consuntivo per le attività di progettazione
1.2.0	Davide Trevisan	Verificatore	2016-04-02	Verifica sezioni test di validazione e di integrazione
1.1.1	Alberto De Agostini	Progettista	2016-04-01	Stesura sezioni test di validazione e di integrazione
1.1.0	Marco Boseggia	Verificatore	2016-02-26	Verifica sezioni 2.2, 2.8.1 e sottosezioni
1.0.2	Francesco Agostini	Analista	2016-02-24	Stesura sezione 2.8.1 e sottosezioni
1.0.1	Francesco Agostini	Analista	2016-02-19	Ristesura sezione 2.2 in seguito alle correzioni del docente
1.0.0	Alberto De Agostini	Responsabile	2016-01-21	Approvazione documento
0.6.0	Andrea Giacomo Baldan	Verificatore	2016-01-21	Verifica sezione Resoconto attività di Verifica
0.5.2	Francesco Agostini	Analista	2016-01-21	Stesura sezione Resoconto attività di Verifica
0.5.1	Francesco Agostini	Analista	2016-01-20	Creazione scheletro sezione Resoconto attività di Verifica
0.5.0	Michael Munaro	Verificatore	2016-01-20	Verifica Pianificazione dei test
0.4.2	Marco Boseggia	Analista	2016-01-18	Integrazione sezione Pianificazione dei test
0.4.1	Giacomo Vanin	Analista	2016-01-17	Stesura sezione Pianificazione dei test
0.4.0	Michael Munaro	Verificatore	2016-01-16	Verifica Correzioni sezione Standard di Qualità
0.3.1	Giacomo Vanin	Analista	2016-01-16	Correzioni sezione Standard di Qualità
0.3.0	Andrea Giacomo Baldan	Verificatore	2016-01-12	Verifica sezione Standard di Qualità
0.2.1	Marco Boseggia	Analista	2016-01-09	Stesura sezione Standard di Qualità
0.2.0	Michael Munaro	Verificatore	2016-01-07	Verifica Correzione effettuata
0.1.1	Giacomo Vanin	Analista	2016-01-08	Correzione sezione Gestione amministrativa della revisione
0.1.0	Andrea Giacomo Baldan	Verificatore	2016-01-05	Verifica capitoli stesi in precedenza
0.0.3	Giacomo Vanin	Analista	2016-01-04	Stesura sezione Gestione amministrativa della revisione



0.0.2	Francesco Agostini	Analista	2016-01-02	Stesura sezione Visione generale della strategia di verifica
0.0.1	Andrea Giacomo Baldan	Amministratore	2015-12-16	Creazione scheletro del documento



Indice

1	Sommario	1
1.1	Scopo del documento	1
1.2	Scopo del Prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Visione generale della strategia di verifica	3
2.1	Definizione obiettivi	3
2.1.1	Qualità di processo	3
2.1.2	Qualità di prodotto	3
2.2	Procedure di controllo di qualità di processo	3
2.3	Procedure di controllo di qualità di prodotto	3
2.4	Organizzazione	4
2.5	Pianificazione strategica e temporale	4
2.6	Responsabilità	5
2.7	Tecniche di analisi	5
2.7.1	Analisi statica	5
2.7.1.1	Walkthrough	5
2.7.1.2	Inspection	5
2.7.2	Analisi dinamica	5
2.7.2.1	Test di unità	5
2.7.2.2	Test di integrazione	5
2.7.2.3	Test di sistema	6
2.7.2.4	Test di regressione	6
2.7.2.5	Test di accettazione	6
2.8	Metriche di valutazione	6
2.8.1	Metriche per i processi	6
2.8.1.1	Budget Variance (BV)	6
2.8.1.2	Schedule Variance (SV)	7
2.8.2	Metriche per i documenti	7
2.8.2.1	Indice Gulpease	7
2.8.3	Metriche per il software	8
2.8.3.1	Linee di codice per linee di commento	8
2.8.4	Copertura del codice	8
2.8.5	Numero di attributi per classe	8
2.8.6	Numero di parametri per metodo	8
2.8.7	Complessità ciclomatica	9
2.8.8	Livelli di annidamento	9
A	Standard di qualità	10



A.1	Standard ISO/IEC 15504	10
A.2	Ciclo di Deming	11
A.3	Standard ISO/IEC 9126	12
B	Resoconto attività di verifica	15
B.1	Attività di Analisi	15
B.1.1	Documenti	15
B.2	Attività di Progettazione	15
B.2.1	Documenti	15
C	Pianificazione dei test	16
C.1	Test di sistema	16
C.2	Test di integrazione	20
C.2.1	Descrizione dei test di integrazione	21
C.2.2	Tracciamento componenti-test di integrazione	22
C.3	Test di validazione	22
C.3.1	Test TV1	23
C.3.2	Test TV2	23
C.3.3	Test TV3	23
C.3.4	Test TV4	23
C.3.5	Test TV5	24
C.3.6	Test TV6	24
C.3.7	Test TV7	24
C.3.8	Test TV8	24
C.3.9	Test TV9	25
C.3.10	Test TV10	25
C.3.11	Test TV11	25
C.3.12	Test TV12	25
C.3.13	Test TV13	26
C.3.14	Test TV14	26
C.3.15	Test TV15	26
C.3.16	Test TV16	26
C.3.17	Test TV17	27
C.3.18	Test TV18	27
C.3.19	Test TV19	27
C.3.20	Test TV20	27
C.3.21	Test TV21	28
C.3.22	Test TV22	28
C.3.23	Test TV23	28



1 Sommario

1.1 Scopo del documento

Il seguente documento ha lo scopo di delineare le strategie che il gruppo *ScalateKids* ha deciso di adottare per garantire degli obiettivi qualitativi da applicare ai processi sfruttati per lo sviluppo del progetto **ActorBase**. Per raggiungere tali obiettivi è necessario svolgere un'attività_G di verifica continua in modo da rilevare e correggere errori o malfunzionamenti minimizzando lo spreco di risorse.

1.2 Scopo del Prodotto

Implementazione di un database NoSQL_G di tipo key-value_G orientato alla gestione di grandi moli di dati utilizzando il modello ad attori_G su JVM_G, comprensivo di un *Domain Specific Language* (DSL_G) da utilizzare da riga di comando per poter interagire con il database.

Il progetto dovrà essere pubblicato su *GitHub* sotto licenza *MIT*.

1.3 Glossario

Tutti i termini di carattere tecnico o fraintendibile e gli acronimi sono raccolti nel file [Glossario v1.0.0](#); ogni occorrenza di parole nel *Glossario* è indicata da una "G" in pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Capitolato d'appalto C1:** *Actorbase: a NoSQL DB based on the Actor model*;
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1.pdf>
- **Norme di Progetto:** [Norme di Progetto v2.0.0](#).

1.4.2 Informativi

- **Piano di Progetto:** [Piano di Progetto v2.0.0](#)
- **Dispense fornite dall'insegnamento Ingegneria del Software mod. A:**
<http://www.math.unipd.it/~tullio/IS-1/2015/>
- **Standard ISO_G/IEC_G 15504:**
https://en.wikipedia.org/wiki/ISO/IEC_15504
- **Standard ISO_G/IEC_G 9126:**
https://it.wikipedia.org/wiki/ISO/IEC_9126



- **Ciclo di Deming_c:**

<https://en.wikipedia.org/wiki/PDCA>



2 Visione generale della strategia di verifica

2.1 Definizione obiettivi

2.1.1 Qualità di processo

Per perseguire la qualità del prodotto è necessario garantire la qualità dei processi attuati per la sua creazione. Per raggiungere questo scopo i *Verificatori* con approvazione del *Responsabile* hanno trovato utile adottare lo standard ISO_G/IEC_G 15504¹ denominato SPICE (Software Process Improvement and Capability dEtermination). Il ciclo di Deming_G² (noto anche come ciclo PDCA_G) definisce un metodo di controllo costante sui processi, consentendone un miglioramento continuo. Questo miglioramento consente ai processi di aumentare il livello di maturità nella scala data dallo standard ISO_G/IEC_G 15504.

2.1.2 Qualità di prodotto

I *Verificatori* con approvazione del *Responsabile* hanno deciso di seguire lo standard ISO_G/IEC_G 9126³ che ha lo scopo di delineare delle metriche per la misurazione della qualità di prodotto. Questa è fondamentale per garantire che il prodotto software_G finale funzioni correttamente e rispetti degli obiettivi di qualità prefissati.

2.2 Procedure di controllo di qualità di processo

Per applicare correttamente il PDCA_G è necessario che le seguenti condizioni siano soddisfatte:

- Ci sia una pianificazione accurata dei processi;
- Le risorse siano divise in maniera chiara nella pianificazione;
- I processi siano controllati.

La pianificazione descritta nel [Piano di Progetto v2.0.0](#) è stata fatta cercando di soddisfare queste condizioni. Per avere una misurazione oggettiva e quantitativa della qualità su ogni processo sono state stabilite delle metriche di valutazione descritte nella sezione [2.8.1](#). Tali misurazioni saranno effettuate periodicamente ogni venti giorni.

L'insieme di queste condizioni e metriche consente l'utilizzo del PDCA_G garantendo un miglioramento continuo della qualità di ogni processo e, di conseguenza, del prodotto finale.

2.3 Procedure di controllo di qualità di prodotto

Il controllo di qualità del prodotto si suddividerà in tre parti:

- Attuazione di tecniche di analisi statica e dinamica descritte nella sezione [2.7](#);

¹Per approfondimenti consultare la sezione [A.1](#).

²Per approfondimenti consultare la sezione [A.2](#).

³Per approfondimenti consultare la sezione [A.3](#).



- Verifica costante dell'output di ogni attività_G, i cui risultati verranno riportati nell'appendice B;
- Validazione, ovvero la conferma che il prodotto risponda correttamente ai requisiti attesi.

Per avere una misurazione oggettiva e quantitativa della qualità su ogni prodotto sono state stabilite delle metriche di valutazione per documenti e software descritte nelle sezioni, rispettivamente, 2.8.2 e 2.8.3. Queste misurazioni saranno effettuate ogniqualvolta viene prodotto un file, sia esso un documento o un file di codice.

2.4 Organizzazione

Per garantire la qualità del prodotto finale verranno effettuate attività_G di verifica su ogni processo attuato. I controlli su ogni processo avranno come obiettivo garantire la qualità sia del processo stesso sia dell'eventuale prodotto da esso ricavato.

Ogni fase descritta in [Piano di Progetto v2.0.0](#) necessita di attività_G di verifica diverse a causa della differenza natura del loro output. Di seguito sono elencate le strategie di verifica usate per ciascuna fase:

- **Analisi:** durante questa fase verrà verificato che ogni documento creato sia conforme a quanto descritto nel documento [Norme di Progetto v2.0.0](#), inoltre sarà necessario controllare che ogni requisito sia associato a un caso d'uso e viceversa;
- **Progettazione:** durante questa fase l'attività_G di verifica si concentrerà sul processo incrementale dei documenti prodotti in fase di Analisi e sui processi attuati per l'attività_G di progettazione assicurandosi che ogni documento e ogni processo sia conforme a quanto descritto nelle [Norme di Progetto v2.0.0](#);
- **Codifica:** durante questa fase l'attività_G di verifica si concentrerà sul processo incrementale dei documenti prodotti in fase di Analisi e sui processi attuati per l'attività_G di codifica assicurandosi che ogni documento e ogni processo sia conforme a quanto descritto nelle [Norme di Progetto v2.0.0](#).
- **Verifica e Validazione:** durante questa fase l'attività_G di verifica continuerà ad essere sul processo incrementale dei documenti prodotti nelle precedenti fasi. Inoltre sarà effettuata una corposa attività_G di verifica per il collaudo del prodotto.

In ogni documento viene inoltre tenuto un diario delle modifiche per permettere di mantenere uno storico delle attività_G svolte su di esso e delle responsabilità per ogni attività_G.

2.5 Pianificazione strategica e temporale

L'attività_G di redazione di ogni documento sarà sempre anticipata da uno studio sulla struttura e sui contenuti del documento interessato. In questo modo si riduce la possibilità di errori concettuali e il tempo effettivo di scrittura del documento viene ridotto.

Il processo di verifica deve essere ben organizzato e pianificato per rispettare le scadenze fissate nel [Piano di Progetto v2.0.0](#).

L'attività di verifica segue le metodologie descritte nelle [Norme di Progetto v2.0.0](#). Esse sono state pensate in maniera tale da consentire l'individuazione e la correzione degli errori il più presto possibile evitando che si diffondano.



2.6 Responsabilità

Per organizzare al meglio il lavoro e in particolare le attività_c di verifica e validazione vengono assegnati questi compiti ai soli *Verificatori* e al *Responsabile*. Inoltre le attività_c di verifica non potranno essere svolte da un membro che ha partecipato allo sviluppo di tale sezione per garantire assenza di conflitto di interessi.

2.7 Tecniche di analisi

2.7.1 Analisi statica

L'analisi statica è una tecnica atta a trovare errori e anomalie. Questa tecnica è applicabile sia alla documentazione sia al codice_c.

2.7.1.1 Walkthrough

Come descritto nelle [Norme di Progetto v2.0.0](#) verrà effettuata la tecnica walkthrough nelle fasi iniziali di ogni attività_c per passare il prima possibile ad attività_c di *Inspection* (paragrafo seguente).

2.7.1.2 Inspection

Successivamente all'attività_c di 2.7.1.1 Walkthrough verrà fatta attività_c di inspection (seguendo *Norme di Progetto* (paragrafo precedente)) rispetto alla lista di controllo ottenuta in precedenza.

2.7.2 Analisi dinamica

L'analisi dinamica si applica solo al prodotto software e serve a trovare errori o difetti nell'implementazione. I test effettuati devono essere *ripetibili*; questa caratteristica è fondamentale poiché un test deve produrre sempre lo stesso risultato dato un input in un ambiente per trovare errori.

2.7.2.1 Test di unità

Test per verificare ogni unità del software prodotto. Per unità si intende la più piccola quantità di software che è utile verificare singolarmente.

L'obiettivo di questi test è trovare ed eliminare possibili errori creati dai *programmatore*.

2.7.2.2 Test di integrazione

Test per verificare che l'integrazione tra due o più unità software funzionino correttamente.

In mancanza di unità questi test possono essere svolti tramite la creazione di una o più componenti fittizie che simulano il comportamento delle unità con cui l'unità creata deve interfacciarsi.



2.7.2.3 Test di sistema

Test per verificare la totale copertura dei requisiti software emersi in fase di *Analisi dei Requisiti*. Questi test verificano la totalità del prodotto software al suo raggiungimento di una versione definitiva.

2.7.2.4 Test di regressione

Test da effettuare per verificare che una modifica del software non pregiudichi il funzionamento del sistema o di altre parti del prodotto prima correttamente funzionanti.

Questa attività_G dev'essere svolta tenendo conto del tracciamento dei requisiti per individuare i giusti test di unità, di integrazione e di sistema che possono essere influenzati dalle modifiche effettuate.

2.7.2.5 Test di accettazione

Questo corrisponde al collaudo vero e proprio del software da svolgersi col *Proponente*.

2.8 Metriche di valutazione

Per rendere quantificabile il processo di verifica verranno usate delle metriche elencate sotto. Soltanto due tipi di range_G saranno accettati:

- **Ottimale:** contiene i valori che si deve cercare di ottenere, valori al di fuori di questo range_G dovranno essere verificati e, se necessario, discussi dal gruppo;
- **Accettazione:** valori per avere un'accettazione del prodotto.

2.8.1 Metriche per i processi

Per garantire un monitoraggio e un controllo dei processi è stato scelto di utilizzare un indice sui costi e un indice temporale.

2.8.1.1 Budget Variance (BV)

Questo indice indica se alla data corrente si è in linea con i costi preventivati.

Per calcolare questo indice basta moltiplicare le ore di lavoro pianificate per i rispettivi costi e sottrarle alla moltiplicazione tra le ore preventivate per i rispettivi costi. Se $BV_G \geq 0$ significa che le spese sostenute sono minori rispetto ai costi preventivati.

Range accettabili:

- Range_G ottimale: ≥ 0 ;
- Range_G di accettazione: ≥ -600 .



2.8.1.2 Schedule Variance (SV)

Questo indice indica se alla data corrente si è in anticipo o in ritardo rispetto alla pianificazione temporale. Per calcolare questo indice basta sottrarre le ore effettive alle ore pianificate. Il risultato indica una quantità di ore lavoro.

Se $SV_G \geq 0$ significa che si è in anticipo rispetto a quanto pianificato.

Range accettabili:

- $Range_G$ ottimale: ≥ 0 ;
- $Range_G$ di accettazione: ≥ -40 .

2.8.2 Metriche per i documenti

Per la stesura dei documenti è stato scelto di utilizzare un *indice di leggibilità* per la lingua italiana.

2.8.2.1 Indice Gulpease

Questo indice indica il grado di leggibilità di un documento.

L'indice di Gulpease prende in considerazione principalmente tre variabili:

- Numero delle frasi;
- Numero delle lettere;
- Numero delle parole.

La formula applicata per calcolare il suddetto indice è:

$$89 + \frac{300 * (\text{numerodellefrasi}) - 10 * (\text{numerodellelettere})}{\text{numerodelleparole}} \quad (1)$$

Lo script applicato su un documento produrrà in output un valore intero compreso tra zero e cento, in particolare i significati sono i seguenti:

- Indice inferiore a 80 rappresenta un documento difficile da leggere per persone con una licenza di scuola elementare;
- Indice inferiore a 60 rappresenta un documento difficile da leggere per persone con una licenza di scuola media inferiore;
- Indice inferiore a 40 rappresenta un documento difficile da leggere per persone con una licenza di diploma superiore;

Range accettabili:

- $Range_G$ ottimale: 60 - 100;
- $Range_G$ di accettazione: 45 - 60.



2.8.3 Metriche per il software

Per perseguire una buona qualità software sono state pensate diverse metriche di valutazione descritte nei prossimi paragrafi. Tuttavia questa sezione è una sezione che potrà subire cambiamenti durante le prossime revisioni.

2.8.3.1 Linee di codice per linee di commento

Applicato su un file produrrà un valore in output che rappresenta la percentuale di righe di commento rispetto alle linee di codice_c.

Range accettabili:

- Range_c ottimale: 30 - 40%;
- Range_c di accettazione: 20 - 30%.

2.8.4 Copertura del codice

La copertura del codice_c indica il numero di istruzioni che vengono eseguite durante i test rispetto alla totalità delle istruzioni. Per avere una minor possibilità di presenza di errori nel codice_c si dovrà avere una percentuale il più alta possibile di istruzioni testate.

Tuttavia metodi molto semplici che non necessitano testing andranno a influire negativamente in questo ambito.

Range accettabili:

- Range_c ottimale: 60 - 100%;
- Range_c di accettazione: 40 - 60%.

2.8.5 Numero di attributi per classe

Un numero troppo elevato di attributi in una classe potrebbe indicare il non soddisfacimento del principio di *single responsibility*_c. Questo potrebbe significare la necessità di una divisione in più classi relazionate tra loro per dividere le diverse funzionalità o un possibile errore di progettazione.

Range accettabili:

- Range_c ottimale: 3 - 6;
- Range_c di accettazione: 0 - 12.

2.8.6 Numero di parametri per metodo

Un numero troppo elevato di parametri in un metodo potrebbe indicare il non soddisfacimento del principio di *single responsibility*_c. Questo potrebbe significare la necessità di una divisione in sotto metodi per dividere le diverse funzionalità o un possibile errore di progettazione.

Range accettabili:



- Range ottimale: 0 - 4;
- Range di accettazione: 4 - 8.

2.8.7 Complessità ciclomatica

La complessità ciclomatica è usata per misurare la complessità di parti di codice_G come funzioni, metodi, classi o moduli.

Questa è calcolata attraverso un grafo di controllo del flusso del programma.

La complessità ciclomatica di una sezione di codice_G è il numero di cammini linearmente indipendenti attraverso il codice_G sorgente. Per esempio, se il codice_G sorgente non contiene punti decisionali come IF o cicli FOR, allora la complessità sarà 1, poiché esiste un solo cammino nel sorgente (e quindi nel grafo). Se il codice_G ha un singolo IF contenente una singola condizione, allora ci saranno due cammini possibili: il primo se l'IF viene valutato a TRUE e un secondo se l'IF viene valutato a FALSE.

Range accettabili:

- Range_G ottimale: 0 - 10;
- Range_G di accettazione: 10 - 15.

2.8.8 Livelli di annidamento

Indica il numero di livelli di annidamento di funzioni e metodi. Un valore alto di questo indice implica un livello di complessità molto elevato, aumentando la probabilità di commettere errori.

Range accettabili:

- Range_G ottimale: 1 - 4;
- Range_G di accettazione 5 - 8.



A Standard di qualità

A.1 Standard ISO/IEC 15504

Lo standard ISO_c/IEC_c 15504 definisce un metodo per controllare continuamente la qualità di ogni processo. Questo metodo sfrutta un modello denominato SPY (Software Process Assessment and Improvement) per la valutazione di processo.

Questo standard fornisce i seguenti nove attributi di qualità di processo:

- **Process performance:** il processo riesce a trasformare input identificabili in output identificabili.;
- **Performance management:** Il processo viene attuato e controllato seguendo una pianificazione che garantisca il raggiungimento degli obiettivi;
- **Work product management:** Il processo viene attuato e controllato seguendo una pianificazione che garantisca risultati documentati e verificati;
- **Process definition:** Il processo viene attuato seguendo standard di processo per garantire i risultati;
- **Process resource:** Il processo ha risorse adeguate per la sua attuazione;
- **Process measurement:** I risultati raggiunti e le misure rilevate durante l'attuazione di un processo sono stati usati per assicurarsi che tale processo supporti il raggiungimento di specifici obiettivi;
- **Process control:** Ogni processo è controllato attraverso la raccolta, analisi ed utilizzo delle misure di prodotto e di processo rilevate, per migliorare, eventualmente, le sue modalità di attuazione;
- **Process change:** Le modifiche alla definizione, gestione e attuazione di ogni processo sono controllate;
- **Continuous improvement:** Le modifiche ad ogni processo sono identificate ed implementate per assicurare il continuo miglioramento nel raggiungimento degli obiettivi rilevanti per l'organizzazione.

Questi attributi vengono utilizzati per suddividere la maturità di un processo nei seguenti sei livelli:

- **Livello 0 - Incompleto:** Il processo non raggiunge i suoi obiettivi;
- **Livello 1 - Attuato:** Il processo raggiunge i suoi obiettivi ma non è controllato. Il raggiungimento di questo livello è dimostrato dal possesso dell'attributo di *Process Performance*;
- **Livello 2 - Gestito:** Il processo è attuato, pianificato, tracciato, verificato e migliorato sulla base di obiettivi ben definiti. Il raggiungimento di questo livello è dimostrato dal possesso dell'attributo di *Performance management* e *Work product management*;
- **Livello 3 - Definito:** Il processo è attuato, pianificato e controllato su procedure basate sui principi del software engineering. Il raggiungimento di questo livello è dimostrato dal possesso dell'attributo di *Process definition* e *Process resource*;
- **Livello 4 - Predicibile:** Il processo è stabilizzato ed è attuato all'interno di definiti limiti riguardo i risultati attesi, le performance e le risorse impiegate. Il raggiungimento di questo livello è dimostrato dal possesso dell'attributo di *Performance measurement* e *Process control*;

- **Livello 5 - Ottimizzante:** Il processo è predicibile ed in grado di adattarsi per raggiungere obiettivi specifici e rilevanti per la organizzazione. Il raggiungimento di questo livello è dimostrato dal possesso dell'attributo di *Process change* e *Continuous improvement*.

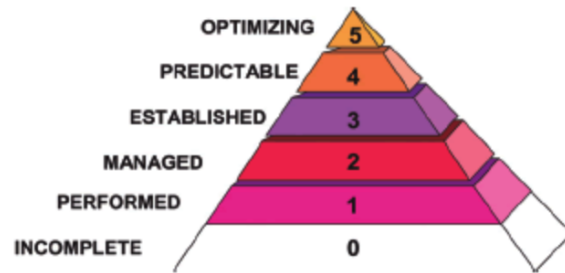


Figura 1: Visione piramidale dei sei livelli di maturità di processo.

Infine per stabilire in che livello si trovi un processo ci sono quattro livelli di possesso per ogni attributo:

- **N:** Non posseduto;
- **P:** Parzialmente posseduto;
- **L:** Largamente posseduto;
- **F:** Pienamente (Fully) posseduto.

A.2 Ciclo di Deming

Il ciclo di Deming_G, anche chiamato PDCA_G, è un modello che consente il miglioramento continuo dei processi. Si suddivide nelle seguenti quattro attività_G:

- **Plan:** attività_G di pianificazione di attività_G, risorse, scadenze e responsabilità;
- **Do:** attività_G di esecuzione delle attività_G pianificate;
- **Check:** attività_G di verifica dei risultati dell'attività_G di esecuzione, questi risultati verranno poi confrontati con i risultati pianificati nella fase di pianificazione;
- **Act:** attività_G di messa in pratica del miglioramento dei processi sfruttando i risultati dell'attività_G di verifica per trovare e modificare gli aspetti critici.

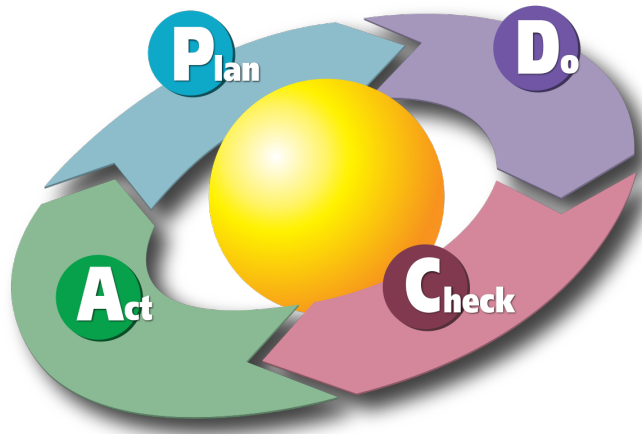


Figura 2: Ciclo di Deming

A.3 Standard ISO/IEC 9126

Lo standard ISO_c/IEC_c 9126 definisce un modello per migliorare l'organizzazione e i processi, avendo come conseguenza il miglioramento della qualità del prodotto finale. I criteri qualitativi sono divisi in tre macro_c aree:

- **Qualità interna:** è la qualità del prodotto software vista dall'interno. Si riferisce alla valutazione del codice_c sorgente e dell'architettura prima che il prodotto sia ultimato;
- **Qualità esterna:** è la qualità del prodotto software vista dall'esterno. Si riferisce al risultato dei test condotti sul prodotto ultimato in un ambiente di prova;
- **Qualità in uso:** è la qualità del prodotto vista dall'utilizzatore finale del prodotto. Si riferisce alla qualità dopo il rilascio dell'applicazione.

Non potendo testare la qualità in uso nell'ambito di questo progetto si è scelto di concentrarsi sulla qualità interna ed esterna. Queste qualità nello standard ISO_c/IEC_c 9126 sono ulteriormente suddivise in sei categorie:

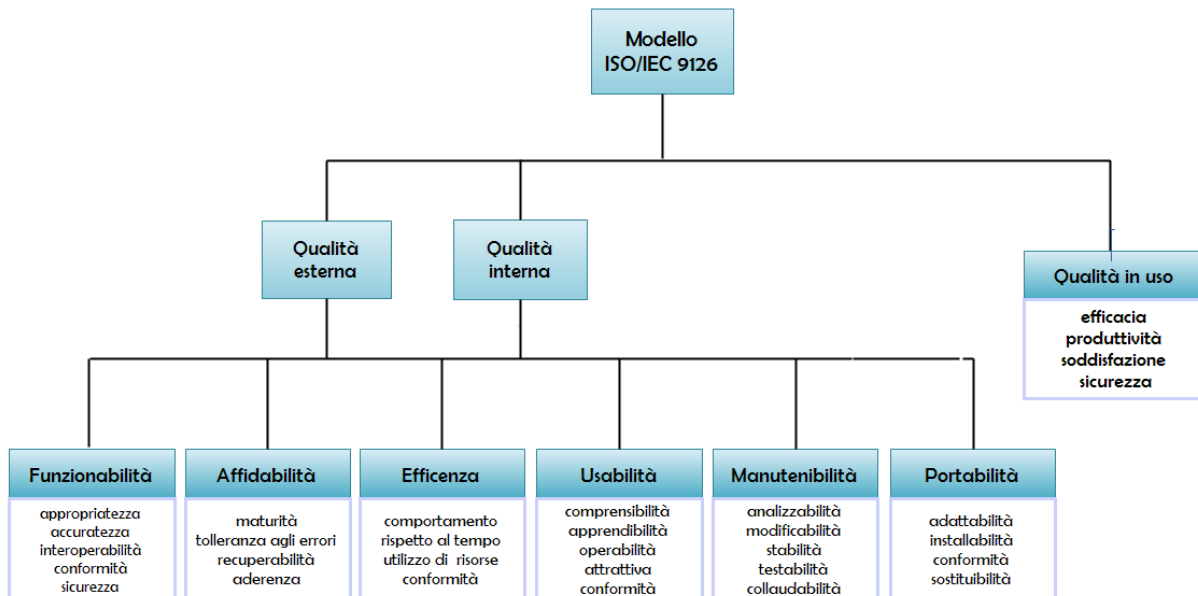


Figura 3: Modello dello standard ISO/IEC 9126

- **Funzionalità:** capacità del prodotto di fornire funzioni che soddisfino esigenze stabilite. Questa categoria si suddivide in:
 - **Appropriatezza:** capacità del prodotto di fornire un appropriato insieme di funzioni che rispondano a determinati requisiti;
 - **Accuratezza:** capacità del prodotto di fornire dei risultati corretti o con precisione concordata;
 - **Interoperabilità:** capacità del prodotto di interagire con uno o più sistemi esterni;
 - **Conformità:** capacità del prodotto di aderire a standard del settore di applicazione;
 - **Sicurezza:** capacità del prodotto di proteggere i dati, ad esempio negando l'accesso a utenti non autorizzati.
- **Affidabilità:** capacità del prodotto di mantenere uno specificato livello di prestazioni durante un determinato periodo. Questa categoria si suddivide in:
 - **Maturità:** capacità del prodotto di evitare che si verifichino risultati non corretti a causa di errori nel software;
 - **Tolleranza agli errori:** capacità del prodotto di mantenere uno specificato livello di prestazioni anche in presenza di errori nel software o di uso scorretto del prodotto;



- **Recuperabilità:** capacità del prodotto di ristabilire un livello adeguato di prestazioni in seguito a un malfunzionamento;
- **Aderenza:** capacità del prodotto di a standard riguardanti l'affidabilità.
- **Efficienza:** capacità del prodotto di fornire prestazioni appropriate in relazione alle risorse usate. Questa categoria si suddivide in:
 - **Comportamento rispetto al tempo:** capacità del prodotto di fornire i risultati in tempi adeguati;
 - **Utilizzo delle risorse:** capacità del prodotto di utilizzare le risorse in modo adeguato;
 - **Conformità:** capacità del prodotto di aderire a standard sull'efficienza.
- **Usabilità:** capacità del prodotto di essere capito e usato dall'utente. Questa categoria si suddivide in:
 - **Comprensibilità:** facilità di comprensione del prodotto;
 - **Apprendibilità:** capacità di ridurre l'impegno richiesto all'utente per imparare come usare correttamente il prodotto finale;
 - **Operabilità:** capacità del prodotto di fare in modo che gli utenti riescano a farne uso per i propri scopi;
 - **Attrattiva:** capacità del prodotto di essere piacevole per l'utente;
 - **Conformità:** capacità del prodotto di aderire a standard sull'usabilità.
- **Manutenibilità:** capacità del prodotto di essere modificato e migliorato. Questa categoria si suddivide in:
 - **Analizzabilità:** facilità di analisi del codice, per la localizzazione di errori;
 - **Modificabilità:** capacità del prodotto di permettere la sostituzione di componenti per implementare una determinata modifica;
 - **Stabilità:** capacità del prodotto di evitare errori derivanti da modifiche errate;
 - **Testabilità:** capacità del prodotto di essere facilmente testato rispetto alle modifiche apportate.
- **Portabilità:** capacità del prodotto di essere trasportato in un altro ambiente di lavoro. Questa categoria si suddivide in:
 - **Adattabilità:** capacità del prodotto di essere adattato per ambienti operativi diversi senza modifiche diverse da quelle fornite;
 - **Installabilità:** capacità del prodotto di essere installato in specifici ambienti;
 - **Conformità:** capacità del prodotto di aderire a standard sulla portabilità;
 - **Sostituibilità:** capacità del prodotto di essere utilizzato al posto di un altro software per svolgere gli stessi compiti.



B Resoconto attività di verifica

B.1 Attività di Analisi

B.1.1 Documenti

Nella tabella di seguito sono riportati i valori degli indici di Gulpease per ogni documento prodotto durante l'attività_c di Analisi. L'esito della verifica dipenderà dal valore di questo indice come descritto nella sezione [2.8.2.1](#).

Documento	Valore indice	Esito
<i>Studio di Fattibilità v1.0.0</i>	54.18	Superato
<i>Norme di Progetto v1.0.0</i>	52.04	Superato
<i>Piano di Progetto v1.0.0</i>	54.87	Superato
<i>Piano di Qualifica v1.0.0</i>	52.82	Superato
<i>Analisi dei Requisiti v1.0.0</i>	68.06	Superato
<i>Glossario v1.0.0</i>	47.42	Superato

B.2 Attività di Progettazione

B.2.1 Documenti

Nella tabella di seguito sono riportati i valori degli indici di Gulpease per ogni documento prodotto durante l'attività_c di Progettazione. L'esito della verifica dipenderà dal valore di questo indice come descritto nella sezione [2.8.2.1](#).

Documento	Valore indice	Esito
<i>Specifica Tecnica v1.0.0</i>	68	Superato
<i>Norme di Progetto v2.0.0</i>	61	Superato
<i>Piano di Progetto v2.0.0</i>	64	Superato
<i>Piano di Qualifica v2.0.0</i>	64	Superato
<i>Analisi dei Requisiti v2.0.0</i>	88	Superato
<i>Glossario v2.0.0</i>	52	Superato



C Pianificazione dei test

Di seguito sono descritti i test di sistema. In futuro questa sezione verrà aggiornata con i test di integrazione e di unità.

C.1 Test di sistema

I test di sistema permettono di verificare il comportamento del sistema rispetto ai requisiti descritti nell'[Analisi dei requisiti v2.0.0](#).

La tabella seguente descrive i test dei requisiti ritenuti meritevoli di verifica.

La colonna "Stato" nella tabella è lasciata vuota in quanto i test saranno effettuati successivamente.

Test	Descrizione	Stato	Requisito
TS.OBF1	Viene verificato che il sistema dovrà fornire la possibilità di distribuire il carico all'interno di un cluster scalabile		OBF1
TS.OBF1.1	Viene verificato che il cluster _c dovrà fornire la possibilità di connettersi ai singoli nodi _c del cluster _c		OBF1.1
TS.OBF1.1.1	Viene verificato che il singolo nodo _c dovrà fornire un server per ricevere comandi dall'esterno e inviare risposte		OBF1.1.1
TS.OBF1.1.10.1	Viene verificato che il sistema ad attori dovrà permettere di autenticarsi dall'esterno mediante interazioni tra Clientactor _c , Mainactor _c , Storefinder _c e Userkeeper _c		OBF1.1.10.1
TS.OBF1.1.10.2.1	Viene verificato che il sistema ad attori dovrà permettere di creare collezioni _c mediante l'interazione tra Clientactor _c , Mainactor _c , Storefinder _c , Storekeeper _c e Userkeeper _c		OBF1.1.10.2.1
TS.OBF1.1.10.2.2	Viene verificato che il sistema ad attori dovrà permettere di visualizzare una lista delle collezioni _c mediante l'interazione con Clientactor _c		OBF1.1.10.2.2
TS.OBF1.1.10.2.3	Viene verificato che il sistema ad attori dovrà permettere di modificare il nome delle collezioni _c mediante l'interazione tra Clientactor _c , Mainactor _c , Storefinder _c e Userkeeper _c		OBF1.1.10.2.3
TS.OBF1.1.10.2.4	Viene verificato che il sistema ad attori dovrà permettere di cancellare collezioni _c mediante l'interazione tra Clientactor _c , Mainactor _c , Storefinder _c , Storekeeper _c , Userkeeper _c e Ninja _c		OBF1.1.10.2.4
TS.OBF1.1.10.2.5	Viene verificato che il sistema ad attori dovrà permettere di aggiungere collaboratori a collezioni _c		OBF1.1.10.2.5
TS.OBF1.1.10.2.6	Viene verificato che il sistema ad attori dovrà permettere di rimuovere collaboratori da collezioni _c mediante l'interazione tra Clientactor _c , Mainactor _c , Storefinder _c , Storekeeper _c , Userkeeper _c e Ninja _c		OBF1.1.10.2.6
TS.OBF1.1.10.2.7	Viene verificato che il sistema ad attori dovrà permettere di esportare collezioni _c mediante l'interazione tra Clientactor _c , Mainactor _c , Storefinder _c , Storekeeper _c , Userkeeper _c e Ninja _c		OBF1.1.10.2.7



TS.OBF1.1.10.3.1	Viene verificato che il sistema ad attori dovrà permettere di inserire item _G mediante l'interazione tra Clientactor _G , Mainactor _G , Storefinder _G , Storekeeper _G , Ninja _G e Manager _G		OBF1.1.10.3.1
TS.OBF1.1.10.3.2	Viene verificato che il sistema ad attori dovrà permettere di creare una collezione _G nel caso di inserimento item _G a collezione _G inesistente mediante interazioni tra Clientactor _G , Mainactor _G e Userkeeper _G		OBF1.1.10.3.2
TS.OBF1.1.10.3.3	Viene verificato che il sistema ad attori dovrà inviare un messaggio d'errore verso l'esterno nel caso in cui la chiave dell'item _G da inserire senza sovrascrittura sia già presente, mediante interazioni tra Clientactor _G , Mainactor _G , Storefinder _G e Storekeeper _G		OBF1.1.10.3.3
TS.OBF1.1.10.3.4	Viene verificato che il sistema ad attori dovrà permettere di cancellare item _G da collezioni _G mediante interazioni tra Clientactor _G , Main _G , Storefinder _G , Storekeeper _G e Ninja _G		OBF1.1.10.3.4
TS.OBF1.1.10.3.5	Viene verificato che il sistema ad attori dovrà inviare un messaggio d'errore verso l'esterno nel caso in cui il nome della collezione _G a cui inserire nuovi item _G sia inesistente, mediante interazioni tra Clientactor _G ed esterno		OBF1.1.10.3.5
TS.OBF1.1.10.4	Viene verificato che il sistema ad attori dovrà permettere di ricercare item _G tra collezioni _G mediante interazioni tra Clientactor _G , Main _G , Storefinder _G e Storekeeper _G		OBF1.1.10.4
TS.OBF1.1.10.5	Viene verificato che il sistema ad attori dovrà permettere di modificare la password utente di uno username _G mediante interazioni tra Clientactor _G , Main _G , Storefinder _G , Userkeeper _G		OBF1.1.10.5
TS.OBF1.1.10.6	Viene verificato che il sistema ad attori dovrà permettere la disconnessione di client esterni mediante interazioni con Clientactor _G		OBF1.1.10.6
TS.OBF1.1.10.7.1	Viene verificato che il sistema ad attori dovrà permettere di aggiungere nuovi utenti all'interno del database _G mediante interazioni tra Clientactor _G , Main _G , Storefinder _G e Userkeeper _G		OBF1.1.10.7.1
TS.OBF1.1.10.7.2	Viene verificato che il sistema ad attori dovrà permettere la rimozione di un utente dal database _G mediante interazioni tra Clientactor _G , Main _G , Storefinder _G		OBF1.1.10.7.2
TS.OBF1.1.10.7.3	Viene verificato che il sistema ad attori dovrà permettere di effettuare un reset della password di un utente mediante interazioni tra Clientactor _G , Main _G , Storefinder _G , Userkeeper _G		OBF1.1.10.7.3
TS.OBF2	Viene verificato che il sistema dovrà fornire una console _G per inviare comandi al server (CLI)		OBF2
TS.OBF2.1	Viene verificato che il sistema dovrà fornire un Domain Specific Language per comunicare con il database _G		OBF2.1
TS.OBF2.1.1	Viene verificato che il DSL _G dovrà fornire un comando per effettuare l'autenticazione utilizzabile mediante CLI		OBF2.1.1



TS.OBF2.1.2.1	Viene verificato che il DSL _G dovrà fornire un comando per la creazione di una nuova collezione _G utilizzabile mediante CLI		OBF2.1.2.1
TS.OBF2.1.2.2	Viene verificato che il DSL _G dovrà fornire un comando per elencare i nomi delle collezioni _G presenti all'interno del database _G utilizzabile mediante CLI		OBF2.1.2.2
TS.OBF2.1.2.3	Viene verificato che il DSL _G dovrà fornire un comando per cancellare una o più collezioni _G utilizzabile mediante CLI		OBF2.1.2.3
TS.OBF2.1.2.4	Viene verificato che il DSL _G dovrà fornire un comando per modificare il nome delle collezioni utilizzabile mediante CLI		OBF2.1.2.4
TS.OBF2.1.2.5	Viene verificato che il DSL _G dovrà fornire un comando per aggiungere collaboratori _G ad una collezione _G del sistema utilizzabile mediante CLI		OBF2.1.2.5
TS.OBF2.1.2.6	Viene verificato che il DSL _G dovrà fornire un comando per rimuovere un collaboratore _G da una collezione _G del sistema utilizzabile mediante CLI		OBF2.1.2.6
TS.OBF2.1.2.7	Viene verificato che il DSL _G dovrà fornire un comando per l'esportazione di collezioni _G su file JSON _G utilizzabile mediante CLI		OBF2.1.2.7
TS.OBF2.1.3.1	Viene verificato che il DSL _G dovrà fornire un comando per inserire un nuovo item _G utilizzabile mediante CLI		OBF2.1.3.1
TS.OBF2.1.3.1.1	Viene verificato che il DSL _G dovrà fornire un comando per inserire un item _G specificandone gli attributi da CLI		OBF2.1.3.1.1
TS.DEF2.1.3.1.2	Viene verificato che il DSL _G dovrà fornire un comando per inserire nuovi item _G da file JSON _G utilizzabile mediante CLI		DEF2.1.3.1.2
TS.OBF2.1.3.6	Viene verificato che il DSL _G dovrà fornire un comando per cancellare un item _G specificandone gli attributi da CLI		OBF2.1.3.6
TS.DEF2.1.4	Viene verificato che il DSL _G dovrà fornire un comando per richiedere aiuto generale sul sistema utilizzabile mediante CLI		DEF2.1.4
TS.DEF2.1.5	Viene verificato che il DSL _G dovrà fornire un comando per richiedere aiuto sull'utilizzo di uno specifico comando del sistema utilizzabile mediante CLI		DEF2.1.5
TS.OBF2.1.6	Viene verificato che il DSL _G dovrà fornire un comando per permettere di effettuare una ricerca su una o più collezioni _G all'interno del sistema; utilizzabile mediante CLI		OBF2.1.6
TS.OBF2.1.7	Viene verificato che il DSL _G dovrà fornire un comando per la modifica della propria password utilizzabile mediante CLI		OBF2.1.7
TS.OBF2.1.8	Viene verificato che il DSL _G dovrà fornire un comando per effettuare il logout dal sistema utilizzabile mediante CLI		OBF2.1.8
TS.OBF2.1.9.1	Viene verificato che il DSL _G dovrà fornire un comando per l'aggiunta di un nuovo utente all'interno del sistema utilizzabile mediante CLI		OBF2.1.9.1
TS.OBF2.1.9.2	Viene verificato che il DSL _G dovrà fornire un comando per la rimozione di un utente dal sistema		OBF2.1.9.2



TS.OBF2.1.9.3	Viene verificato che il DSL _G dovrà fornire un comando per effettuare il reset della password di un utente alla password standard		OBF2.1.9.3
TS.DEF3	Viene verificato che dovrà essere fornito un driver _G Scala _G per interfacciarsi con il database _G		DEF3
TS.DEF3.1	Viene verificato che il driver _G dovrà permettere di effettuare l'autenticazione all'interno del sistema.		DEF3.1
TS.DEF3.2.1	Viene verificato che il driver _G dovrà permettere un la creazione di una nuova collezione _G		DEF3.2.1
TS.DEF3.2.2	Viene verificato che il driver _G dovrà permettere di elencare i nomi delle collezioni _G presenti all'interno del database _G		DEF3.2.2
TS.DEF3.2.3	Viene verificato che il driver _G dovrà permettere di cancellare una o più collezioni _G		DEF3.2.3
TS.DEF3.2.4	Viene verificato che il driver _G dovrà permettere di modificare il nome delle collezioni		DEF3.2.4
TS.DEF3.2.5	Viene verificato che il driver _G dovrà permettere di aggiungere collaboratori _G ad una collezione _G del sistema		DEF3.2.5
TS.DEF3.2.6	Viene verificato che il driver _G dovrà permettere di rimuovere un collaboratore _G da una collezione _G del sistema		DEF3.2.6
TS.DEF3.2.7	Viene verificato che il driver _G dovrà permettere di esportare collezioni _G su file JSON _G		DEF3.2.7
TS.DEF3.3.1	Viene verificato che il driver _G dovrà permettere di inserire un nuovo item _G		DEF3.3.1
TS.DEF3.3.1.1	Viene verificato che il driver _G dovrà permettere di inserire un nuovo item _G specificandone gli attributi		DEF3.3.1.1
TS.DEF3.3.1.2	Viene verificato che il driver _G dovrà permettere di inserire nuovi item _G da file JSON _G		DEF3.3.1.2
TS.DEF3.3.2	Viene verificato che il driver _G dovrà permettere di cancellare uno o più item _G dal sistema		DEF3.3.2
TS.DEF3.4	Viene verificato che il driver _G dovrà permettere di effettuare ricerche su una o più collezioni _G all'interno del sistema		DEF3.4
TS.DEF3.5	Viene verificato che il driver _G dovrà permettere di effettuare il logout dal sistema		DEF3.5
TS.DEF3.6.1	Viene verificato che il driver _G dovrà permettere a utenti amministratori di aggiungere un nuovo utente al sistema		DEF3.6.1
TS.DEF3.6.2	Viene verificato che il driver _G dovrà permettere a utenti amministratori di rimuovere un utente dal sistema		DEF3.6.2
TS.DEF3.6.3	Viene verificato che il driver _G dovrà permettere a utenti amministratori di effettuare il reset della password ad un utente all'interno del sistema		DEF3.6.3
TS.DEF3.7	Viene verificato che il driver _G dovrà permettere di modificare la propria password		DEF3.7



TS.DEF3.8	Viene verificato che il Driver _G dovrà strutturare i dati in output in maniera navigabile		DEF3.8
TS.OBV4	Viene verificato che il sistema dovrà funzionare in ambienti Ubuntu 14.04 o superiore		OBV4
TS.OBV5	Viene verificato che il sistema dovrà funzionare su JVM _G versione 8		OBV5
TS.OBV6	Viene verificato che il sistema dovrà utilizzare la libreria Akka versione 2.4.2		OBV6
TS.OBV7	Viene verificato che il sistema dovrà essere implementato utilizzando il linguaggio Scala versione 2.11		OBV7
TS.DEV8	Viene verificato che il sistema dovrà funzionare in ambienti Windows 10 o superiori		DEV8
TS.DEV9	Viene verificato che il sistema dovrà funzionare su sistemi OSX El Capitan o superiori		DEV9
TS.OBQ10	Viene verificato che dovrà essere fornito un manuale utente redatto in lingua inglese		OBQ10
TS.OBQ11	Viene verificato che dovranno essere rispettate tutte le norme e le metriche sulla stesura del codice come riportato nelle <i>Norme di Progetto</i>		OBQ11
TS.OBQ12	Viene verificato che dovrà essere prodotta documentazione in Scaladoc del prodotto		OBQ12

C.2 Test di integrazione

I test di integrazione permettono di verificare la corretta integrazione ed il corretto flusso dei dati all'interno del sistema, sia relazioni tra package_G sia il funzionamento di ogni singolo package_G di ogni componente.

L'approccio utilizzato è di tipo bottom-up_G, in questo modo è possibile integrare prima le parti con minore dipendenza funzionale e, nel caso di inserimento di parti difettose, è possibile retrocedere ad uno stato testato e sicuro in maniera semplice. Il diagramma seguente non rispetta il formalismo UML_G 2.0 ed è utilizzato per semplificare l'illustrazione della strategia di integrazione:

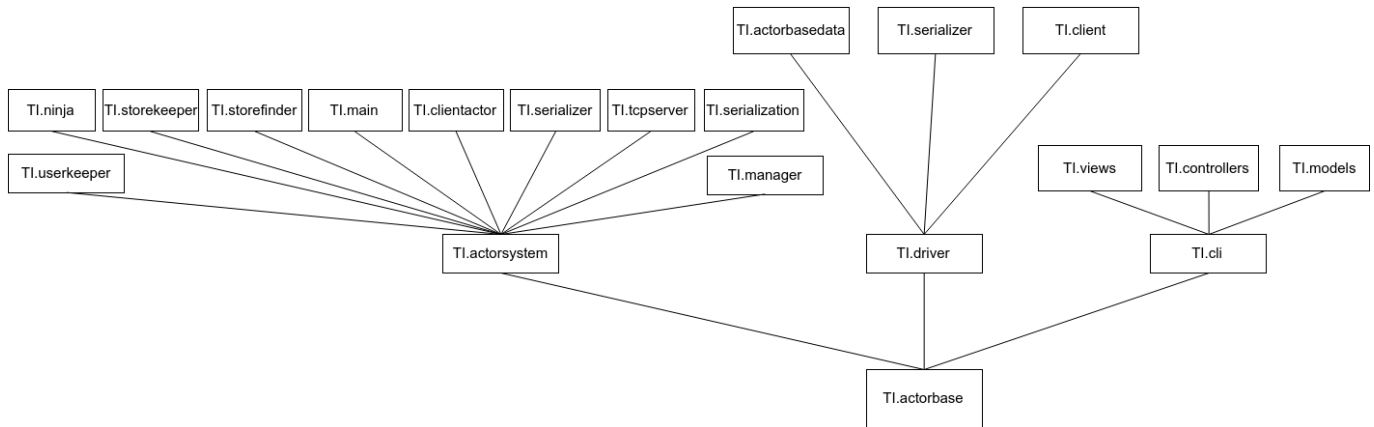


Figura 4: Diagramma informale della strategia di integrazione

C.2.1 Descrizione dei test di integrazione

La colonna "Stato" nella tabella è lasciata vuota in quanto i test saranno effettuati successivamente.

Test	Descrizione	Componente	Stato
TI.actorbase	Test di integrazione finale per le componenti cli, driver e actorsystem	actorbase	
TI.cli	Test di integrazione finale per models, views e controllers	cli	
TI.models	Verifica della corretta esecuzione dei comandi mediante integrazione con componente driver	models	
TI.controllers	Verifica del parsing dei comandi e della corretta integrazione con models e views	controllers	
TI.views	Verifica del corretto funzionamento delle procedure di input e output	views	
TI.driver	Test di integrazione finale per client, actorbasedata e serializer	driver	
TI.client	Verifica della comunicazione tra client e server e corretto funzionamento dei comandi	client	
TI.serializer	Test di controllo sulla corretta esecuzione delle procedure di serializzazione _c e deserializzazione _c secondo il protocollo RESP _c	serializer	
TI.actorbasedata	Verifica della corretta integrazione con le componenti client e serializer	actorbasedata	
TI.actorsystem	Test di integrazione finale per tcpserver, storefinder, storekeeper, ninja, manager, userkeeper, main, clientactor, serialization, warehouseman	actorsystem	
TI.serialization	Verifica delle corrette procedure di serializzazione _c e deserializzazione _c secondo i protocolli specificati	serialization	
TI.tcpserver	Test di verifica del corretto funzionamento del server tcp	tcpserver	



TI.clientactor	Test di verifica del parsing dei comandi ricevuti dall'esterno e della corretta esecuzione dei messaggi	clientactor	
TI.main	Verifica della corretta esecuzione dei messaggi dedicati	main	
TI.storefinder	Verifica della corretta esecuzione dei messaggi dedicati	storefinder	
TI.storekeeper	Verifica della corretta esecuzione dei messaggi dedicati	storekeeper	
TI.userkeeper	Verifica della corretta esecuzione dei messaggi dedicati	userkeeper	
TI.manager	Verifica della corretta esecuzione dei messaggi dedicati	manager	
TI.ninja	Verifica della corretta esecuzione dei messaggi dedicati	ninja	
TI.warehouseman	Verifica della corretta esecuzione dei messaggi dedicati	warehouseman	

C.2.2 Tracciamento componenti-test di integrazione

Componente	Test
actorbase	TI.actorbase
actorbase::cli	TI.cli
actorbase::cli::models	TI.models
actorbase::cli::controllers	TI.controllers
actorbase::cli::views	TI.views
actorbase::driver	TI.driver
actorbase::driver::client	TI.client
actorbase::driver::serializer	TI.serializer
actorbase::driver::actorbasedata	TI.actorbasedata
actorbase::actorsystem	TI.actorsystem
actorbase::actorsystem::serialization	TI.serialization
actorbase::actorsystem::tcpserver	TI.tcpserver
actorbase::actorsystem::clientactor	TI.clientactor
actorbase::actorsystem::main	TI.main
actorbase::actorsystem::storefinder	TI.storefinder
actorbase::actorsystem::storekeeper	TI.storekeeper
actorbase::actorsystem::userkeeper	TI.userkeeper
actorbase::actorsystem::manager	TI.manager
actorbase::actorsystem::ninja	TI.ninja
actorbase::actorsystem::warehouseman	TI.warehouseman

C.3 Test di validazione

I test di validazione servono all'accertamento che il prodotto realizzato sia conforme alle attese. Per ogni test vengono descritti i passi che un utente deve eseguire per testare i requisiti ad esso associati. Si noti che i seguenti test eccetto **TV1**, **TV3** e **TV4** rappresentano le operazioni possibili sul database, per questo motivo tutti questi test saranno effettuabili sia tramite l'utilizzo della CLI_c sia tramite l'utilizzo del Driver_c. Nei test di validazione in cui è previsto l'inserimento di parametri per l'esecuzione di una operazione sul data-



base si è scelto di non descrivere dettagliatamente tutti i parametri da inserire poiché questa descrizione è già stata redatta nell'[Analisi dei Requisiti 2.0.0](#).

C.3.1 Test TV1

L'utente intende aggiungere un nodo al cluster_G.

L'utente deve:

- Avviare il server actorbase (TV1.1);
- Adattare il file di configurazione per potersi connettere al cluster_G (TV1.2);
- Avviare actorbase sul nuovo nodo (TV1.3);
- Verificare tramite console l'avvenuta connessione al cluster_G (TV1.4).

C.3.2 Test TV2

L'utente intende autenticarsi presso il sistema.

L'utente deve:

- Inserire il comando per effettuare l'autenticazione (TV2.1);
- Inserire i parametri per effettuare l'autenticazione (TV2.2);
- Verificare che l'autenticazione ha avuto successo controllando la risposta ricevuta dal server o eseguendo un comando disponibile solo ad utenti autenticati (TV2.3).

C.3.3 Test TV3

L'utente intende richiedere aiuto generale.

L'utente deve:

- Inserire il comando di aiuto generale attraverso l'uso della CLI_G (TV3.1);
- Verificare che l'output della richiesta di aiuto è stato stampato in output sulla CLI_G (TV3.2).

C.3.4 Test TV4

L'utente intende richiedere aiuto specifico di un comando.

L'utente deve:

- Inserire il comando di aiuto attraverso l'uso della CLI_G (TV4.1);
- Inserire il nome del comando di cui vuole ricevere informazioni (TV4.2);
- Verificare che l'output della richiesta di aiuto è stato stampato in output sulla CLI_G (TV4.3).



C.3.5 Test TV5

L'utente intende creare una collezione_G.

L'utente deve:

- Inserire il comando per la creazione collezione_G (TV5.1);
- Inserire il nome della collezione_G da creare (TV5.2);
- Verificare l'avvenuta creazione della collezione_G controllando la risposta ricevuta dal server o usando il comando di visualizzazione lista delle collezioni (TV5.3).

C.3.6 Test TV6

L'utente intende visualizzare la lista delle collezioni_G.

L'utente deve:

- Inserire il comando per la visualizzazione lista delle collezioni_G (TV6.1);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV6.2).

C.3.7 Test TV7

L'utente intende modificare nome ad una propria collezione_G.

L'utente deve:

- Inserire il comando per la modifica nome collezione_G (TV7.1);
- Inserire i parametri per cambiare il nome della collezione_G (TV7.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o usando il comando di visualizzazione lista delle collezioni (TV7.3).

C.3.8 Test TV8

L'utente intende cancellare una collezione_G.

L'utente deve:

- Inserire il comando per la cancellazione collezione_G (TV8.1);
- Inserire i parametri per cancellare la collezione_G (TV8.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o usando il comando di visualizzazione lista delle collezioni (TV8.3).



C.3.9 Test TV9

L'utente intende aggiungere un collaboratore ad una collezione_G.

L'utente deve:

- Inserire il comando per 'aggiunta collaboratore a collezione_G (TV9.1);
- Inserire i parametri per aggiungere il collaboratore alla collezione_G (TV9.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV9.3).

C.3.10 Test TV10

L'utente intende rimuovere un collaboratore da una collezione_G.

L'utente deve:

- Inserire il comando per la rimozione di un collaboratore da un collezione_G (TV10.1);
- Inserire i parametri per rimuovere un collaboratore da una collezione_G (TV10.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV10.3).

C.3.11 Test TV11

L'utente intende esportare una o più collezioni_G su file.

L'utente deve:

- Inserire il comando per l'esportazione di collezioni_G su file (TV11.1);
- Inserire i parametri per esportare collezioni_G su file (TV11.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o controllando il file salvato su disco (TV11.3).

C.3.12 Test TV12

L'utente intende inserire un item_G. Se la collezione_G su cui si vuole inserire l'item_G non è presente nel sistema essa verrà creata.

L'utente deve:

- Inserire il comando per l'inserimento di item_G (TV12.1);
- Inserire i parametri per inserire l'item_G (TV12.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o usando il comando di ricerca sulla collezione_G giusta con la chiave appena usata (TV12.3).



C.3.13 Test TV13

L'utente intende inserire un $item_G$ importando da file.

L'utente deve:

- Inserire il comando per l'importazione di $item_G$ da file (TV13.1);
- Inserire i parametri per inserire gli $item_G$ tramite importazione (TV13.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o usando il comando di ricerca sugli $item_G$ appena inseriti (TV13.3).

C.3.14 Test TV14

L'utente intende cancellare un $item_G$.

L'utente deve:

- Inserire il comando per la cancellazione di $item_G$ (TV14.1);
- Inserire i parametri per cancellare l' $item_G$ (TV14.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o usando il comando di ricerca sulla collezione G giusta con la chiave appena usata per effettuare la cancellazione (TV14.3).

C.3.15 Test TV15

L'utente intende effettuare una ricerca di $item_G$ su una o più collezioni G .

L'utente deve:

- Inserire il comando per la ricerca di $item_G$ (TV15.1);
- Inserire i parametri per la ricerca di un $item_G$ su una o più collezioni G (TV15.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV15.3).

C.3.16 Test TV16

L'utente intende visualizzare tutti gli $item_G$ contenuti in una collezione G .

L'utente deve:

- Inserire il comando per la ricerca di $item_G$ (TV16.1);
- Inserire i parametri per la ricerca di tutti gli $item_G$ contenuti in una collezione G (TV16.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV16.3).

**C.3.17 Test TV17**

L'utente intende effettuare una ricerca di un item_G sull'intero database (secondo i suoi permessi come specificato nel documento [Analisi dei Requisiti 2.0.0](#)).

L'utente deve:

- Inserire il comando per la ricerca di item_G (TV17.1);
- Inserire i parametri per la ricerca di un item_G sull'intero database (TV17.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV17.3).

C.3.18 Test TV18

L'utente intende visualizzare l'intero database (secondo i suoi permessi come specificato nel documento [Analisi dei Requisiti 2.0.0](#)).

L'utente deve:

- Inserire il comando per la ricerca di item_G (TV18.1);
- Inserire i parametri per la ricerca per l'intero database (TV18.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV18.3).

C.3.19 Test TV19

L'utente intende effettuare modificare la propria password.

L'utente deve:

- Inserire il comando per la modifica password (TV19.1);
- Inserire i parametri per la modifica password (TV19.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o effettuando una deautenticazione seguita da una autenticazione con la nuova password (TV19.3).

C.3.20 Test TV20

L'utente intende effettuare deautenticazione dal server.

L'utente deve:

- Inserire il comando per effettuare il logout (TV20.1);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server o eseguendo un comando disponibile solo ad utenti autenticati (che dovrà non essere disponibile) (TV20.2).



C.3.21 Test TV21

L'utente amministratore intende aggiungere un utente al sistema.

L'utente deve:

- Inserire il comando per aggiungere un utente al sistema (TV21.1);
- Inserire i parametri per l'aggiunta utente (TV21.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV21.3).

C.3.22 Test TV22

L'utente amministratore intende rimuovere un utente dal sistema.

L'utente deve:

- Inserire il comando per la rimozione utente (TV22.1);
- Inserire i parametri per la rimozione utente (TV22.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV22.3).

C.3.23 Test TV23

L'utente amministratore intende effettuare un reset della password di un utente.

L'utente deve:

- Inserire il comando per effettuare il reset password (TV23.1);
- Inserire i parametri per il reset password (TV23.2);
- Verificare l'avvenuta esecuzione del comando controllando la risposta ricevuta dal server (TV23.3).



Elenco delle figure

1	Visione piramidale dei sei livelli di maturità di processo.	11
2	Ciclo di Deming	12
3	Modello dello standard ISO/IEC 9126	13
4	Diagramma informale della strategia di integrazione	21