



SIRIUS

SEQUENZIATORE

Piano Di Qualifica

Versione 2.0.0

Ingegneria Del Software AA 2013-2014

Informazioni documento

Titolo documento:	Piano Di Qualifica
Data creazione:	21 Gennaio 2014
Versione attuale:	2.0.0
Utilizzo:	Interno
Nome file:	<i>PianoDiQualifica_v2.0.0.pdf</i>
Redazione:	Seresin Davide
Verifica:	Botter Marco Quaglio Davide
Approvazione:	Quaglio Davide
Distribuito da:	Sirius
Destinato a:	Prof. Vardanega Tullio Prof. Cardin Riccardo Zucchetti S.p.A.

Sommario

Il documento spiega in dettaglio la strategia di verifica adottata dal gruppo *Sirius* per lo sviluppo del progetto *Sequenziatore*.

Diario delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
3.0.0	2014-06-20	<i>Seresin Davide</i>	<i>Responsabile</i>	Approvazione del documento
2.1.0	2014-04-24	<i>Quaglio Davide</i>	<i>Responsabile</i>	Verifica delle correzioni
2.0.2	2014-04-24	<i>Quaglio Davide</i>	<i>Responsabile</i>	Aggiornamenti paragrafi per evitare discorsività
2.0.1	2014-04-20	<i>Quaglio Davide</i>	<i>Amministratore</i>	Aggiornamento titoli con contenuti
2.0.0	2014-03-23	<i>Quaglio Davide</i>	<i>Responsabile</i>	Approvazione del documento
1.2.0	2014-03-23	<i>Botter Marco</i>	<i>Verificatore</i>	Verifica della nuova definizione test e resoconto attività di verifica
1.1.2	2014-03-23	<i>Quaglio Davide</i>	<i>Amministratore</i>	Aggiunta del resoconto attività di verifica
1.1.1	2014-03-10	<i>Seresin Davide</i>	<i>Amministratore</i>	Modifica della sezione definizione obiettivi
1.1.0	2014-03-11	<i>Quaglio Davide</i>	<i>Verificatore</i>	Verifica delle modifiche
1.0.1	2014-03-10	<i>Seresin Davide</i>	<i>Amministratore</i>	Modifica dello scopo del documento e pianificazione dei test
1.0.0	2014-03-05	<i>Quaglio Davide</i>	<i>Amministratore</i>	Approvazione del documento
0.2.0	2014-03-05	<i>Botter Marco</i>	<i>Verificatore</i>	Verifica delle aggiunte su resoconto dell'attività di verifica
0.1.1	2014-03-01	<i>Seresin Davide</i>	<i>Amministratore</i>	Aggiunta resoconto attività di verifica
0.1.0	2014-02-20	<i>Botter Marco</i>	<i>Verificatore</i>	Verifica del documento e appendice
0.0.3	2014-02-18	<i>Seresin Davide</i>	<i>Amministratore</i>	Aggiunta di informazioni dettagliate e appendice
0.0.2	2014-02-15	<i>Quaglio Davide</i>	<i>Verificatore</i>	Verificato scheletro e bozza documento
0.0.1	2014-02-12	<i>Seresin Davide</i>	<i>Amministratore</i>	Creato lo scheletro del documento

Indice

1	Introduzione	1
1.1	Scopo del Prodotto	1
1.2	Glossario	1
1.3	Riferimenti	1
1.3.1	Normativi	1
1.3.2	Informativi	1
1.4	Scopo del documento	2
2	Visione generale della strategia di verifica	3
2.1	Organizzazione	3
2.2	Pianificazione strategica temporale	4
2.3	Obiettivi	5
2.3.1	Qualità di processo	5
2.3.2	Qualità di prodotto	5
2.4	Procedure di controllo	8
2.4.1	Qualità di processo	8
2.4.2	Qualità di prodotto	8
2.5	Risorse umane e responsabilità	9
2.6	Risorse software	9
2.7	Tecniche di analisi statica	9
2.8	Tecniche di analisi dinamica	10
2.8.1	Test di unità	10
2.8.2	Test di integrazione	10
2.8.3	Test di sistema	10
2.8.4	Test di regressione	10
2.8.5	Test di accettazione	11
3	Gestione amministrativa della revisione	12
3.1	Comunicazione e risoluzione anomalie	12
3.2	Trattamento delle discrepanze	12
4	Pianificazione dei test	13
4.1	Test di sistema	13
4.1.1	Descrizione dei test di sistema	13
4.1.2	Ambito utente	13
4.1.3	Ambito process owner	16
4.2	Requisiti di vincolo	18
4.3	Test di integrazione	19
4.3.1	Descrizione dei test di integrazione	19

4.4	Test di unità	20
A	Appendice	23
A.1	Ciclo di Deming	23
A.2	ISO/IEC 9126	23
A.3	Capability Maturity Model Integration (CMMI)	27
B	Resoconto attività di verifica	29
B.1	Riassunto dell'attività di verifica su RR	29
B.2	Dettaglio dell'attività di verifica su RR	30
B.2.1	Documenti	30
B.2.2	Processi	30
B.3	Riassunto dell'attività di verifica su RP	32
B.4	Dettaglio dell'attività di verifica su RP	33
B.4.1	Documenti	33
B.4.2	Processi	33
B.5	Riassunto dell'attività di verifica su RQ	35
B.6	Dettaglio dell'attività di verifica su RQ	36
B.6.1	Documenti	36
B.6.2	Processi	37
B.6.3	Risultati delle misurazioni sul codice	39

1 Introduzione

1.1 Scopo del Prodotto

Lo scopo del progetto *Sequenziatore*, è di fornire un servizio di gestione di processi definiti da una serie di passi da eseguirsi in sequenza o senza un ordine predefinito, utilizzabile da dispositivi mobili di tipo smaptphone o tablet.

1.2 Glossario

Al fine di rendere più leggibile e comprensibile i documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario_v2.0.0.pdf*.

Ogni occorrenza di vocaboli presenti nel *Glossario* deve essere seguita da una “G” maiuscola in pedice.

1.3 Riferimenti

1.3.1 Normativi

- ISO/IEC Standard 12207:1995;
- ISO/IEC 9126;
- IEEE Std 730TM-2002 (revision of IEEE Std 730-1998) - Standard for Software Quality Assurance Plans;
- **Norme di progetto:** Norme di progetto v2.0.0;
- Capitolato d'appalto C4: *sequenziatore*.

1.3.2 Informativi

- Informazioni sul sito del docente;
- Software Engineering (9th edition) Ian Sommerville Pearson Education Addison-Wesley;
- Ciclo di Deming - Estratto da Software Engineering (9th edition) Ian Sommerville Pearson Education Addison-Wesley;
- Capability Maturity Model Integration (CMMI) - Estratto da Software Engineering (9th edition) Ian Sommerville Pearson Education Addison-Wesley;
- SWEBOK cap.11 - *Software Quality*;
- **Piano di progetto:** Piano Di Progetto v2.0.0;

- Indice di Gulpease.

1.4 Scopo del documento

Il documento si prefigge di illustrare la strategia complessiva di verifica e validazione proposta dal team *Sirius* per pervenire al collaudo del sistema con la massima efficacia_G. In questo documento, inoltre, definiamo gli obiettivi di qualità intesa come il rispetto dei requisiti e prestazioni enunciati esplicitamente, la conformità agli standard di sviluppo esplicitamente documentati e le caratteristiche implicite che si aspetta da un prodotto software. Garantendo in particolar modo ed in modo macroscopico:

- La correttezza del prodotto;
- La verifica continua sulle attività svolte;
- Il soddisfacimento del cliente.

2 Visione generale della strategia di verifica

2.1 Organizzazione

Il team *Sirius* ha deciso di porre al centro di ogni periodo l'attività di verifica in quanto essa certifica la qualità del prodotto. L'attività di verifica sarà continua in tutte le fasi del progetto.

Il processo di qualifica accompagnerà tutte le fasi di ciclo di vita del software. Ogni procedura di verifica sarà schedulata attraverso appositi strumenti e i risultati saranno analizzati in questo documento. Tramite il diario delle modifiche è possibile tenere traccia dell'attività di verifica effettuata ed operare delle verifiche circoscritte ai soli cambiamenti. In particolare le operazioni di controllo verranno istanziate quando il prodotto da analizzare avrà raggiunto uno stato in cui presenti differenze sostanziali rispetto allo stato precedente. Lo schema che rappresenta l'organizzazione e la pianificazione delle attività di verifica è il **modello a V** (V-model). Il modello dimostra la relazione tra ogni periodo del ciclo di vita dello sviluppo del software e il suo periodo di *testing*.

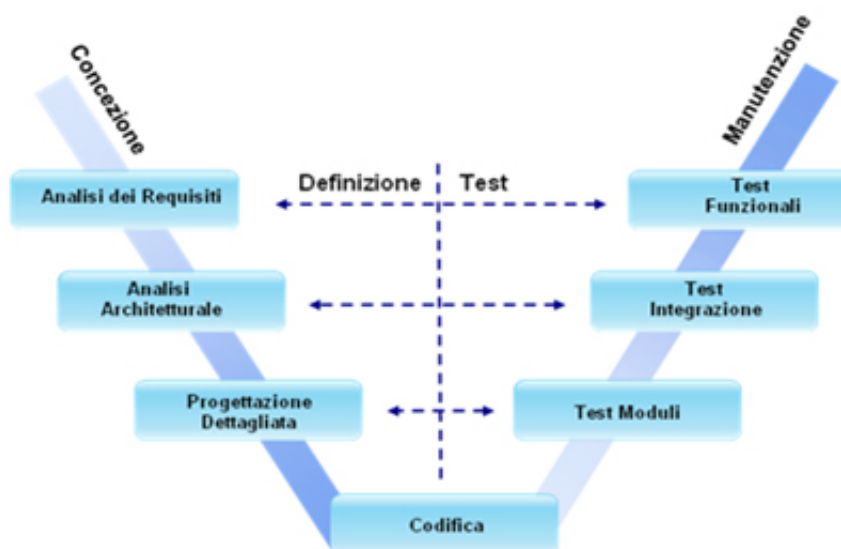


Figura 1: Modello a V

L'organizzazione della strategia di verifica prevede l'attività di verifica in tutti i periodi di avanzamento del prodotto, che sono paralleli alle scadenze definite in Piano Di Progetto v2.0.0.

- **Analisi:** in questa prima fase il compito del *Verificatore* è innanzitutto relativo alla documentazione e alla correttezza del tracciamento dei requisiti. Ogni documento che servirà per la consegna della RR, una volta ultimata la fase di redazione, verrà verificato in modo definitivo seguendo la procedura così definita:

1. Verrà controllata la correttezza dei contenuti rispetto alle aspettative del documento tramite una rilettura accurata;
 2. Verrà controllata la correttezza grammaticale;
 3. Verrà controllato che il documento rispetti le norme definite in Norme di progetto v2.0.0 tramite la lista di controllo presente in tale documento.
 4. Verrà verificato che ogni requisito funzionale rilevato abbia una corrispondenza in almeno un caso d'uso e che questo sia tracciato tramite il software di tracciamento che *Sirius* ha deciso di utilizzare;
 5. Verrà verificato che ogni requisito di vincolo e di qualità sia tracciato tramite il software di tracciamento che *Sirius* ha deciso di utilizzare.
- **Progettazione:** il *Verificatore* ha l'importante compito di controllare il soddisfacimento dei requisiti indicati in fase di analisi. Inoltre, si devono verificare che i processi che portano all'incremento dei documenti redatti nel precedente periodo siano conformi alle procedure e regole descritte in Norme di progetto v2.0.0;
 - **Programmazione:** il *Verificatore* provvederà a controlli periodici e pianificati di porzioni di codice, inizialmente di tipo statico per poi passare a dei controlli di tipo dinamico per valutare la correttezza del software;
 - **Collaudo:** in questa fase le verifiche saranno esclusivamente di tipo dinamico per garantire che il prodotto risponda a tutti i requisiti indicati e a tutte le richieste del committente: sia implicite che esplicite. Per le indicazioni precise circa la procedura di verifica adottata dal gruppo si fa riferimento a Norme di progetto v2.0.0.

Il processo di verifica verterà sulla parte di redazione di documenti, essendo questa un'attività predominante e costante durante tutto il progredire del progetto. Garantendo, altresì che il risultato software sia efficace_G rispetto alla procedura analizzata, non perdendo di efficienza_G contrattuale.

2.2 Pianificazione strategica temporale

Al fine di rispettare in modo ristretto le scadenze citate di seguito e spiegate in modo approfondito in Piano Di Progetto v2.0.0, *Sirius* ha deciso di pianificare in modo approfondito e sistematico l'attività di verifica. Facendo in modo di rilevare e risolvere nel più breve tempo possibile gli errori che vengono rilevati per evitare che questi possano creare maggiori problematiche nell'avanzamento del prodotto software.

Per questo si adottano delle specifiche tecniche in base all'avanzamento del progetto. Ogni attività di redazione dei documenti e di codifica saranno precedute da uno studio

preliminare sulla struttura e sui contenuti degli stessi. *Sirius*, conscio della poca esperienza nella pianificazione e gestione di progetti di questo tipo, ha deciso di inserire degli $slack_G$ temporale durante la pianificazione delle attività. Tale scelta è approfondita in Norme di progetto v2.0.0 che ne definisce la quantità e in Piano Di Progetto v2.0.0 che ne analizza le motivazioni. L'aggiunta di slack temporali, oltre a portare al progetto una pianificazione più precisa, comporta un aumento dei costi che è però commisurato all'aumento della qualità finale.

2.3 Obiettivi

2.3.1 Qualità di processo

Al fine di garantire la qualità di prodotto è necessario ricercare la qualità dei processi che lo definiscono. Per questo *Sirius* ha deciso di adottare lo standard ISO/IEC 15504 denominato SPICE il quale fornisce le indicazioni necessarie a valutare l'idoneità dei processi attualmente in uso. Per applicare correttamente questo modello, ed adattarlo alla gestione attuata dal gruppo di lavoro, si è deciso di utilizzare il ciclo di Deming o PDCA: il quale definisce una metodologia di controllo dei processi durante il loro ciclo di vita permettendo, inoltre, di migliorare in modo continuo la qualità.

2.3.2 Qualità di prodotto

Al fine di aumentare il valore del prodotto e di garantire il corretto funzionamento dello stesso, è necessario fissare degli obiettivi e garantire che questi vengano effettivamente rispettati. Lo standard ISO/IEC 9126, descritto in appendice A.2, è stato redatto con lo scopo di definire obiettivi e di delineare metriche capaci di misurare il raggiungimento degli stesso.

Di seguito elenchiamo le caratteristiche che *Sirius* si impegna a garantire per il prodotto che andrà a realizzare. Oltre alla descrizione della caratteristica qui vengono definite le metriche, i parametri di accettazione.

- **Funzionalità** L'applicazione prodotta deve soddisfare tutti i requisiti obbligatori individuati in Analisi dei requisiti v3.0.0 nel modo più completo ed economico possibile, garantendo la sicurezza del prodotto e dei suoi componenti, e adeguandosi alle norme e alle prescrizioni imposte. Inoltre, data la natura del prodotto *Sequenziatore*, *Sirius* ha deciso di prestare particolare attenzione all'interoperabilità del codice: intesa come la capacità di agire con altri sistemi.
 - **misura:** percentuale di requisiti soddisfatti;
 - **metrica:** la soglia di sufficienza sia il soddisfacimento di tutti i requisiti obbligatori;

- **Affidabilità** L'applicazione deve dimostrarsi robusta, di facile ripristino e recupero in caso di errori, e aderire alle norme e alle prescrizioni stabilite.
 - **misura:** numero di esecuzioni dell'applicazione andate a buon fine;
 - **metrica:** le esecuzioni dovranno spaziare su tutta la gamma delle possibili casistiche. Il numero di esecuzioni andate a buon fine dovrà essere rapportato al numero totale delle casistiche considerate;
- **Usabilità** L'applicazione deve risultare comprensibile, facilmente apprendibile e soprattutto aderire a norme e prescrizioni per garantire facilità d'uso e soddisfacimento delle necessità dell'utente.
 - **misura:** data l'aleatorietà della qualità richiesta, non si riesce a definire un'unità di misura obiettiva;
 - **metrica:** non è stata definita una metrica di usabilità; ma *Sirius* cercherà di offrire la miglior esperienza di utilizzo per tutti coloro che usano il prodotto;
- **Efficienza** L'applicazione deve fornire tutte le funzionalità nel minor tempo possibile e con il minimo utilizzo di risorse.
 - **misura:** il tempo di latenza per ottenere una risposta dal programma; il tempo di latenza per ottenere una risposta simulando un sovraccarico della rete;
 - **metrica:** i tempi di latenza dovranno essere in linea con le tempistiche rilevate con l'utilizzo di architetture dello stesso tipo di quelle definite nel prodotto;
- **Manutenibilità** L'applicazione deve essere analizzabile, facilmente modificabile e verificabile; inoltre dovrà ridurre il rischio di comportamenti inaspettati al seguito dell'effettuazione di modifiche. Inoltre la documentazione prodotta deve essere chiara e comprensibile.
 - **misura:** le misurazioni per garantire questa caratteristica sono diverse e non esclusive e sono descritte in seguito;
 - **metrica:** le metriche da utilizzare sono descritte in seguito;
- **Portabilità** L'applicazione deve essere adattabile e compatibile con ambienti d'uso diversi, e con i quali dovrà coesistere condividendo risorse e anche per questo sarà validata da strumenti forniti dal W3C.
 - **misura:** L'applicazione dovrà essere eseguibile con i browser indicati in Analisi dei requisiti v3.0.0;

- **metrica:** soddisfacimento dei requisiti di compatibilità e validazione tramite strumenti W3C;

Inoltre, *Sirius* ha definito delle altre caratteristiche che andranno ricercate per il prodotto:

- **Semplicità:** realizzazione del prodotto nella maniera più semplice possibile, ma non semplicistica;
- **Incapsulamento:** il codice deve avere visibilità minima e permettere un utilizzo dall'esterno solamente mediante interfacce; ciò aumenta la manutenibilità e la possibilità di riuso del codice;
- **Coesione:** le funzionalità che concorrono allo stesso fine devono risiedere nello stesso componente; favorisce semplicità, manutenibilità, riusabilità e riduce l'indice di dipendenza.

La definizione delle metriche e gli strumenti utilizzati per la rilevazione delle stesse sono specificate in Norme di progetto v2.0.0. Di seguito per ogni metrica indichiamo le misure desiderabili: divise in range ottimale e range di accettazione. D.S indica che sarà definito in seguito.

Metrica	Accettazione	Ottimale
SV ¹	>-(ore prev x5%)	>0
BV ²	>-(costo prev x10%)	>0
Gulpease ³	40-100	50-100
Complessità ciclomatica ⁴	1-15	1-10
Livelli di annidamento ⁵	1-6	1-3
Attributi per classe ⁶	0-16	3-8
Parametri per metodo ⁷	0-8	0-4
Linee di codice per linee di commento ⁸	>0,25	>0,30
Accoppiamento	D.S.	D.S.
Copertura	80%-100%	85%-100%

Tabella 1: parametri delle metriche adottate.

I parametri ottimali e di accettazione sono riferiti a:

1. Parametro calcolato da *Sirius* valutando l'inesperienza del gruppo;
2. Parametro calcolato da *Sirius* valutando l'inesperienza del gruppo;
3. Parametro calcolato in base all'utenza della documentazione;
4. Il valore 10 come massimo fu raccomandato da T.J.McCabe, l'inventore di tale metrica;

5. Valore ideale valutando la chiarezza di codifica;
6. Valore ideale valutando la chiarezza di codifica;
7. Valore ideale valutando la chiarezza di codifica;
8. Il valore 0,30 è ricavato dal rapporto 22/78. Valori ricavati dalle medie dichiarate da Ohlo(Open source network).

2.4 Procedure di controllo

2.4.1 Qualità di processo

Al fine di garantire la qualità di processo, *Sirius* adotterà il principio PDCA, descritto nella sezione A.1. Tramite tale tecnica, sarà garantito un miglioramento continuo dei processi e quindi della qualità degli stessi. Come diretta conseguenza si otterrà il miglioramento qualitativo del prodotto risultante. Per ottenere questo è necessario che il processo sia in controllo e quindi:

- Effettuare una dettagliata pianificazione dei processi;
- Pianificare il numero di risorse da utilizzare, e ripartire le stesse in modo chiaro;

Verranno utilizzate due metriche per monitorare la qualità di processo e per mantenere il processo in controllo, queste sono: BV che monitora il consumo del budget nel tempo e SV che valuta lo stato di avanzamento temporale attuale rispetto a quello pianificato. Le specifiche di queste due metriche sono indicate in Norme di progetto v2.0.0.

Valori ottimali di BV ed SV indicano un elevato grado di conoscenza ed integrazione del processo come indicato nel CMMI nella sezione A.3.

La qualità dei processi viene monitorata inoltre tramite la qualità del prodotto: infatti un prodotto di bassa qualità indica un processo da migliorare.

Se su un processo non vengono rilevati problemi, è possibile apportare dei miglioramenti: riducendo il numero di cicli iterativi, il tempo o le risorse ma garantendo che l'esecuzione del processo sia fedele al piano e soddisfi i requisiti. In questo modo si aumenta l'efficienza del processo e se ne determina un'adattamento positivo alla realtà del team di lavoro valutabile in efficacia.

2.4.2 Qualità di prodotto

Il controllo di qualità di prodotto verrà garantito da:

- *Quality Assurance*: insieme delle attività atte a garantire il raggiungimento degli obiettivi di qualità. Prevede tecniche di analisi statica e dinamica descritte in 2.8;

- *Verifica*: processo che determina se l'output è consistente, corretto e completo. Il processo di verifica sarà eseguito durante l'intera durata del progetto. I risultati delle attività di verifica saranno riportate in appendice B;
- *Validazione*: certificazione che attesta la conformità del sistema ai requisiti.

2.5 Risorse umane e responsabilità

Al fine di garantire l'efficacia e la sistematicità del processo di verifica vengono attribuite delle responsabilità a degli specifici ruoli di progetto. Per il processo di verifica le responsabilità sono attribuite a *Responsabile di Progetto* ed ai *Verificatori*. Mentre compito dell' *Amministratore* è quello di supporto a tutte le attività fornendo una solida infrastruttura software anche per il processo di verifica in ogni fase lavorativa. Per una descrizione più approfondita di ruoli e responsabilità si rimanda a Norme di progetto v2.0.0.

2.6 Risorse software

Al fine di effettuare la fase di verifica e validazione nel modo più sistematico possibile sono stati messi a disposizione di tutti i *Verificatori* dall' *Amministratore* un pacchetto di prodotti software il più specifico possibile rispetto alle esigenze del team. Inoltre, è sempre compito dell' *Amministratore* formare ogni verificatore all'utilizzo dei prodotti che permettano la verifica, evidenziando, se richiesto le funzionalità non utilizzate per ogni prodotto. *Sirius* ha deciso di adottare questo tipo di formazione per valorizzare il lavoro di chi effettivamente utilizza i prodotti di verifica, dando la possibilità che proprio da queste figure nascano idee e proposte di miglioramento che saranno poi valutate dal *Responsabile di progetto* congiuntamente all'*Amministratore*. Gli strumenti necessari al raggiungimento degli obiettivi definiti è indicato in Norme di progetto v2.0.0, che ne definiscono anche le specifiche tecniche e l'utilizzo.

2.7 Tecniche di analisi statica

Questa tecnica di analisi applicabile sia alla documentazione che al codice e permette di effettuare la verifica di quanto prodotto individuando errori ed anomalie. Le tecniche di analisi statica utilizzate sono di tipo **inspection** per la documentazione, solo a seguito di una prima analisi di tipo **walkthrough**. Per il codice verrà adottata principalmente una tipologia di analisi **inspection** per gli errori che sono più ricorrenti o che sono stati rilevati nelle verifiche precedenti. Per la descrizione si faccia riferimento a Norme di progetto v2.0.0.

2.8 Tecniche di analisi dinamica

L'analisi dinamica si applica solamente al prodotto software e ciò consiste nell'esecuzione del codice mediante l'uso di test predisposti per verificarne il funzionamento o rilevare possibili difetti di implementazione eseguendo tutto o solo una parte del codice. Al fine di garantire l'utilità del test è necessario che il test sia *ripetibile*. Per *ripetibile* si intende che dato un certo input per la stessa porzione di codice, questo produca sempre lo stesso output sulla stesso ambiente. Per questo motivi saranno definiti a priori:

- **Ambiente:** si tratta sia del sistema hardware che di quello software sui quali è stato pianificato l'utilizzo del prodotto; di essi deve essere specificato lo stato iniziale dal quale poter far partire il test;
- **Specifica:** definire quali sono gli input e quali dovranno essere gli output attesi;
- **Procedure:** definire come sono svolti i test, l'ordine e come devono essere analizzati i risultati;

I test verteranno su test di unità per le porzioni di codice prodotte, test di integrazione per le componenti aggiunte in modo incrementale; test di sistema per verificare la corretta esecuzione del sistema; eventuali test di regressione per le modifiche apportate a componenti già testati; e infine test di accettazione per validare il prodotto finale.

2.8.1 Test di unità

Verifica di ogni singola unità del prodotto tramite l'utilizzo di stub, driver e logger.

2.8.2 Test di integrazione

Verifica dei componenti di sistema che andranno ad incrementare la parte già testata del prodotto; con l'obiettivo di testare la combinazione di più unità garantendone la funzionalità del risultato. Per l'esecuzione di tali test dovranno essere aggiunte delle componenti fittizie a sostituzione di quelle che non sono ancora state sviluppate, facendo in modo di non influenzare l'esito dell'analisi.

2.8.3 Test di sistema

Consiste nella validazione del prodotto software dal momento che lo si ritiene giunto ad una versione definitiva. Tale test ha lo scopo di verificare che la il prodotto copra tutti i requisiti stabiliti in fase di analisi.

2.8.4 Test di regressione

Si intende la nuova esecuzione di test a fronte di una modifica di alcune componenti software, per garantire la conformità del sistema post modifica.

2.8.5 Test di accettazione

Definito come il collaudo del prodotto. Questo test viene eseguito in presenza del proponente ed in caso di esito positivo si può procedere al rilascio ufficiale del prodotto sviluppato.

3 Gestione amministrativa della revisione

3.1 Comunicazione e risoluzione anomalie

Un'anomalia consiste in una deviazione del prodotto dalle aspettative prefissate. Per la gestione e risoluzione di anomalie ci si affida allo strumento di *ticketing* adottato da *Sirius*, e normato in Norme di progetto v2.0.0. Il *Verificatore*, per ogni anomalia riscontrata, dovrà aprire un nuovo ticket indirizzato al *Responsabile di Progetto*, il quale, dopo aver valutato l'impatto costi/benefici lo approverà e aprirà in ticket per il *Programmatore* che ha sviluppato quella parte di software o redatto il documento. Per la procedura di creazione del ticket si rimanda a Norme di progetto v2.0.0.

3.2 Trattamento delle discrepanze

Una discrepanza è un errore di coerenza tra il prodotto realizzato e quello atteso. *Sirius* interpreta la discrepanza come una forma di anomalia non grave, e per questo verrà trattata come tale.

4 Pianificazione dei test

Di seguito elenchiamo tutti i test di validazione, sistema ed integrazione previsti, prevedendo però ad un successivo aggiornamento per i test di unità. Per quanto riguarda le tempistiche di esecuzione dei test si faccia riferimento a Piano Di Progetto v2.0.0. Il valore **N.A** è da intendersi come non applicato, in quanto tali test saranno eseguiti successivamente nello svolgimento del progetto.

4.1 Test di sistema

In questa sezione vengono descritti i test di sistema che consentiranno a *Sirius* di verificare il comportamento dinamico del sistema rispetto ai requisiti descritti in Analisi dei requisiti v3.0.0. I test sotto riportati sono relativi ai requisiti software individuati e meritevoli di test.

4.1.1 Descrizione dei test di sistema

4.1.2 Ambito utente

Test	Requisito	Descrizione	Stato
TU1	FOBU 1	Verificare che il sistema permetta all'utente di registrarsi	Pianificato
TU1.1	FOBU 1.1	Verificare che lo <i>username</i> utente lo identifichi univocamente all'interno del sistema	Pianificato
TU1.1.1	FOBU 1.1.1	Verificare che lo <i>username</i> inserito dall'utente sia composto da almeno 6 caratteri	Pianificato
TU1.2	FOBU 1.2	Verificare che l'utente debba inserire una <i>password</i> d'accesso	Pianificato
TU1.2.1	FOBU 1.2.1	Verificare che la <i>password</i> sia composta almeno da 8 caratteri alfanumerici	Pianificato
TU1.5	FOBU 1.5	Verificare che il campo data di nascita sia un campo obbligatorio	Pianificato
TU1.5	FOBU 1.5.1	Verificare che la data di nascita inserita dall'utente sia antecedente alla data di iscrizione	Pianificato
TU1.6	FOBU 1.6	Verificare che la <i>email</i> utente sia un campo obbligatorio	Pianificato

TU1.6.1	FDEU 1.6.1	Verificare che la <i>email</i> inserita corrisponda ad un indirizzo di posta elettronica esistente	Pianificato
TU2	FOBU 2	Verificare che il sistema permetta all'utente di autenticarsi	Pianificato
TU2.1	FOBU 2.1	Verificare che il sistema non permetta il login con dati non presenti sul <i>server_G</i>	Pianificato
TU3.1	FOPL 3.1	Verificare che l'utente autenticato possa visualizzare le proprie credenziali	Pianificato
TU3.2	FOPL 3.2	Verificare che l'utente autenticato possa modificare i propri dati	Pianificato
TU4.1	FOBL 4.1	Verificare che l'utente autenticato possa scegliere un processo da una lista selezionata o da risultati di una ricerca	Pianificato
TU4.2	FOBL 4.2	Verificare che l'utente autenticato possa visualizzare la descrizione di un processo selezionato	Pianificato
TU4.3	FOBL 4.3	Verificare che un utente autenticato possa iscriversi a un processo precedentemente selezionato	Pianificato
TU4.4	FOBL 4.4	Verificare che un utente autenticato possa eseguire il processo scelto a cui è iscritto	Pianificato
TU4.4.1	FOBL 4.4.1	Verificare che all'utente autenticato sia concesso di visualizzare i criteri di terminazione di un processo	Pianificato
TU4.4.2	FOBL 4.4.2	Verificare che l'utente autenticato possa visualizzare le informazioni sullo stato corrente di avanzamento del processo selezionato	Pianificato
TU4.4.4	FOBL 4.4.4	Verificare che il sistema permetta all'utente autenticato di eseguire un passo del processo scelto	Pianificato
TU4.4.4.1	FOBL 4.4.4.1	Verificare che l'utente autenticato possa visualizzare le informazioni del passo in esecuzione	Pianificato

TU4.4.4.2	FOBL 4.4.4.2	Verificare che all'utente autenticato sia concesso visualizzare i vincoli da rispettare per superare il passo in esecuzione	Pianificato
TU4.4.4.3	FOBL 4.4.4.3	Verificare che il sistema permetta all'utente autenticato di inserire i dati richiesti per l'esecuzione del passo in corso	Pianificato
TU4.4.4.4	FOBL 4.4.4.4	Verificare che l'utente autenticato possa inviare al sistema i dati richiesti per l'esecuzione del passo in corso	Pianificato
TU4.4.4.4.6	FOPL 4.4.4.4.6	Verificare che il sistema permetta all'utente autenticato di raccogliere i dati in assenza di connessione e di inviarli a collegamento ripristinato	Pianificato
TU4.4.4.5	FOBL 4.4.4.5	Verificare che il sistema notifichi all'utente autenticato se i dati che ha inviato sono corretti, se non soddisfano i vincoli di superamento del passo o se sono in attesa di approvazione	Pianificato
TU4.4.4.6	FOBL 4.4.4.6	Verifica se il sistema permetta all'utente autenticato di concludere un passo del quale ha ricevuto l'approvazione sui dati da parte del sistema o dall'process owner	Pianificato
TU4.4.4.7	FOPL 4.4.4.7	Verifica che il sistema permetta all'utente autenticato di saltare il passo in esecuzione se facoltativo	Pianificato
TU4.4.5.1	FOPL 4.4.5.1	Verifica che all'utente autenticato sia concessa la creazione di un report finale su un processo terminato o del quale ha eseguito tutti i passi	Pianificato

TU4.4.5.2	FOBL 4.4.5.2	Verifica che all'utente autenticato sia permesso di eliminare un processo, un processo terminato o del quale ha eseguito tutti i passi, dalla lista dei processi gestiti	Pianificato
TU4.5	FOBL 4.5	Verificare che il sistema permetta all'utente autenticato di disiscriversi da un processo a cui è iscritto	Pianificato
TU5	FOBL 5	Verifica che l'utente possa terminare la propria sessione, diventando utente generico	Pianificato

Tabella 1: Tabella dei requisiti utente

4.1.3 Ambito process owner

Test	Requisito	Descrizione	Stato
TA1	FOBA 1	Verificare che al process owner sia consentita la creazione di processi	Pianificato
TA1.1	FOBA 1.1	Verificare che il process owner debba inserire un nome che identifichi univocamente il processo che vuole creare	Pianificato
TA1.2	FOBA 1.2	Verificare che il process owner debba inserire la descrizione del processo che vuole creare	Pianificato
T2.4.1	FOBA 2.4.1	Verificare che il sistema permetta al process owner di visualizzare i dati inviati dagli utenti che richiedono la sua approvazione	N.A
T1.3	FOBA 1.3	Verificare che il process owner debba definire i criteri di terminazione di un processo durante la sua creazione	Pianificato
T1.4	FOBA 1.4	Verificare che il sistema permetta al process owner di gestire i passi del processo in creazione	Pianificato

T1.4.1	FOBA 1.4.1	Verificare che il sistema permetta al process owner di creare un passo del processo in creazione	Pianificato
T1.4.2	FOBA 1.4.2	Verificare che il process owner possa visualizzare la lista dei passi creati durante la creazione di un nuovo processo	Pianificato
T1.4.3	FDEA 1.4.3	Verificare che il process owner, durante la creazione di un nuovo processo, potrà modificare un passo esistente	Pianificato
T1.4.4	FDEA 1.4.4	Verificare che il sistema permetta al process owner di eliminare un passo del processo in creazione	Pianificato
T1.5	FOBA 1.5	Verificare che il sistema permetta al process owner di avviare un processo in creazione che contiene almeno un passo	Pianificato
T2	FDEA 2	Verificare che il sistema permetta al process owner la gestione dei processi creati	Pianificato
T2.1	FDEA 2.1	Verificare che il process owner possa scegliere e gestire un processo avviato	Pianificato
T2.1.2	FOPA 2.1.2	Verificare che al process owner sia concesso di ricercare un processo inserendone il nome	Pianificato
T2.1.3	FDEA 2.1.3	Verificare che il sistema permetta al process owner di selezionare un processo da gestire	Pianificato
T2.2	FOPA 2.2	Verificare che il sistema permetta al process owner di selezionare gli utenti a cui permettere l'iscrizione al processo gestito	Pianificato

T2.4	FDEA 2.4 e FOBA 2.4.1	Verificare che al process owner sia concesso di controllare e visualizzare i dati inviati dagli utenti che richiedono la sua approvazione	Pianificato
T2.4.2	FDEA 2.4.2	Verificare che il process owner possa approvare i dati controllati	Pianificato
T2.4.3	FDEA 2.4.3	Verificare che il sistema permetta all'process owner di respingere i dati controllati	Pianificato
T2.4.4	FDEA 2.4.4	Verificare che il sistema invii l'esito del controllo agli utenti che hanno inviato dei dati di richiesta di approvazione	Pianificato
T2.5	FDEA 2.5	Verificare che il sistema permetta all'process owner di terminare un processo avviato	Pianificato
T2.6	FDEA 2.6	Verificare che al process owner sia concesso di eliminare un processo terminato dall'insieme dei processi creati	Pianificato

Tabella 2: Tabella dei requisiti process owner

4.2 Requisiti di vincolo

Test	Requisito	Descrizione	Stato
TV1	VOB1	Verificare che il sistema sia compatibile con il <i>browser_G Google Chrome_G</i> versione 27 e successive	Pianificato
TV5	VOB5	Verificare che il sistema sia compatibile con il <i>browser_G mobile Android_G</i> versione 2.3 e successive	Pianificato
TV6	VOB6	Verificare che il sistema sia compatibile con il <i>browser_G Google Chrome per Android_G</i> a partire dalla versione 18 inclusa	Pianificato
TV7	VOB7	Verificare che il sistema sia compatibile con il <i>browser_G mobile Internet Explorer_G per Windows Phone 8_G</i> a partire dalla versione 10 inclusa	Pianificato

Tabella 3: Tabella dei requisiti di vincolo di compatibilità

4.3 Test di integrazione

In questa sezione verranno descritti i test di integrazione, da utilizzare per 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 bottom-up che permette di sviluppare e verificare le componenti in parallelo.

Assemblando le componenti in modo incrementale i difetti rilevati da un test sono da attribuirsi, con maggior probabilità, all'ultima parte aggiunta e si rende ogni passo di integrazione reversibile consentendo di retrocedere verso uno stato noto e sicuro. In questo modo i componenti base vengono testati più volte riducendo la possibile presenza di errori.

Nel diagramma

4.3.1 Descrizione dei test di integrazione

Di seguito sono elencati i test di integrazione relativo ai componenti indicati in Specifica Tecnica v1.0.0.

Test	Descrizione	Componente	Stato
TU1	login utente e login process owner	mcomponente	N.A.
TU2	creazione processo	N.A.	
TU3	visualizzazione dati processo	mcomponente	N.A.
TU4	Approvazione passo	mcomponente	N.A.

Tabella 4: Tabella dei test di integrazione process owner

Test	Descrizione	Componente	Stato
TU5	gestione dati	mcomponente	N.A.
TU6	iscrizione/disiscrizione	N.A.	
TU7	esecuzione passo	mcomponente	N.A.
TU8	Visualizzazione report finale	mcomponente	N.A.

Tabella 5: Tabella dei test di integrazione user

4.4 Test di unità

I test di unità sulle view sono effettuati creando una nuova classe basata sulla precedente e definita testnome ed invocando il metodo test con i rispettivi parametri.

4.4.0.1 TestLogin

- **Descrizione:** *Template HTML* che permette di gestire l'interfaccia grafica relativa alle richieste di autenticazione al sistema;
- **Metodo:** TestView();

4.4.0.2 TestMainUser

- **Descrizione:** Classe che permette la gestione delle principali componenti dell'interfaccia grafica dell'utente;
- **Metodo:** TestView();

4.4.0.3 TestRegister

- **Descrizione:** *Template HTML* che permette di gestire dell'interfaccia grafica relativa alle richieste di registrazione da parte dell'utente;
- **Metodo:** TestView();

4.4.0.4 TestUserData

- **Descrizione:** *Template HTML* che permette la realizzazione dei *widget* che consentono visualizzazione e modifica dei dati dell'utente;
- **Metodo:** TestView();

4.4.0.5 TestOpenProcess

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* per consentire l'apertura di un processo tramite ricerca o selezionandolo da una lista;
- **Metodo:** TestView();

4.4.0.6 TestManagementProcess

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* per consentire la visualizzazione dello stato del processo selezionato e i vincoli per concludere il passo in corso;
- **Metodo:** TestView();

4.4.0.7 TestSendData

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* per consentire l'invio dei dati richiesti per la conclusione del passo in esecuzione;
- **Metodo:** `TestView()`;

4.4.0.8 TestSendText

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* che consentono di inserire il testo da inviare per concludere il passo in esecuzione;
- **Metodo:** `TestView()`;

4.4.0.9 TestSendNumb

- **Descrizione:** *Template HTML* che permette agli oggetti che la implementano di realizzare i *widget* che consentono di inserire i dati numerici da inviare per concludere il passo in esecuzione;
- **Metodo:** `TestView()`;

4.4.0.10 TestSendPosition

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* che consentono di inviare la posizione geografica richiesta per la conclusione del passo in esecuzione;
- **Metodo:** `TestView()`;

4.4.0.11 TestSendImage

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* che consentono di inserire le immagini richieste per concludere il passo in esecuzione;
- **Metodo:** `TestView()`;

4.4.0.12 TestPrintProcess

- **Descrizione:** *Template HTML* che permette di realizzare i *widget* che consentono il salvataggio dei *report* sull'esecuzione del processo;
- **Metodo:** `TestView()`;

4.4.0.13 TestMainProcessOwner

- **Descrizione:** Componente che permette la gestione delle principali componenti dell'interfaccia grafica dell'utente *process owner_G*;
- **Metodo:** TestView();

4.4.0.14 TestNewProcess

- **Descrizione:** *Template HTML* che permette di gestire l'interfaccia grafica che consente di creare nuovi processi;
- **Metodo:** TestView();

4.4.0.15 TestAddStep

- **Descrizione:** *Template HTML* che permette di gestire l'interfaccia grafica che consente di definire un nuovo passo del processo in creazione;
- **Metodo:** TestView();

4.4.0.16 TestOpenProcess

- **Descrizione:** *Template HTML* che permette di realizzare *iwidget* che consentono di aprire un processo tramite ricerca o selezionandolo da una lista;
- **Metodo:** TestView();

4.4.0.17 TestManageProcess

- **Descrizione:** *Template HTML* che permette di realizzare *iwidget* che consentono di gestire l'accesso ai dati inviati *alserver_G* dagli utenti;
- **Metodo:** TestView();

4.4.0.18 TestCheckStep

- **Descrizione:** *Template HTML* che permette di realizzare *iwidget* che consentono di gestire l'approvazione dei passi che richiedono intervento umano;
- **Metodo:** TestView();

A Appendice

A.1 Ciclo di Deming

Alla luce delle informazioni sopra citate il team ha deciso di adottare la politica del ciclo PDCA per le attività da svolgere. Lo stesso, oltre a fornire supporto nella pianificazione garantisce un elevato standard qualitativo tramite il *Miglioramento continuo*, che è alla base del ciclo di Deming.



Figura 2: Ciclo di Deming

- **Plan:** pianificazione che prevede la definizione di procedure, risorse, scadenze e responsabilità ;
- **Do:** esecuzione delle attività pianificate;
- **Check:** controllo dei risultati ottenuti e confronto con quelli pianificati;
- **Act:** Analisi dei risultati ottenuti e modifica o definizione di nuove procedure che permettano di evitare gli aspetti critici dei processi in esame.

L'adozione del PDCA garantisce un continuo arricchimento dei processi tramite dei cambiamenti e delle riorganizzazioni. Alla base di questo, ci deve essere una conoscenza specifica delle Norme di progetto v2.0.0 da parte di tutti i componenti del team. Inoltre, queste migliorie aumentano i costi di gestione e per questo devono essere valutati dal *Responsabile di progetto*.

A.2 ISO/IEC 9126

Lo standard ISO/IEC 9126 descrive gli obiettivi qualitativi di prodotto e delinea in generale le metriche per misurare il raggiungimento di tale obiettivo (figura 3). In questo standard i criteri sono divisi in 3 aree diverse:

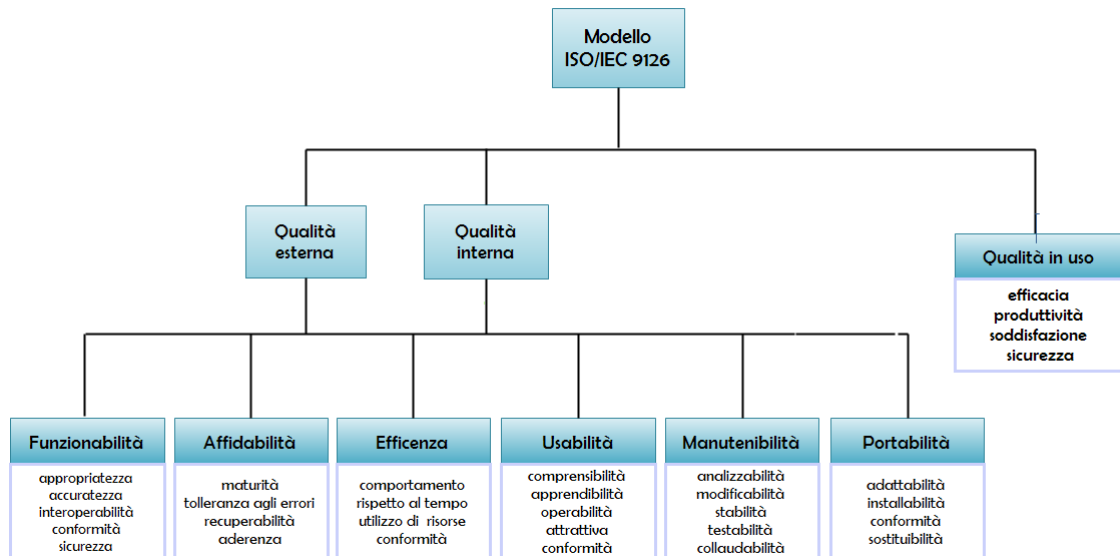


Figura 3: Caratteristiche qualitative definite dal modello ISO/IEC 9126

- **Qualità in uso**: è la qualità del software dal punto di vista dell'utilizzatore;
- **Qualità esterna**: è la qualità del software dal punto di vista esterno nel momento in cui esso viene eseguito e testato in ambiente di prova;
- **Qualità interna**: è la qualità del software vista dall'interno e quindi sono le caratteristiche implementative del software quali architettura e codice che ne deriva.

Non avendo modo di verificare la qualità in uso, *Siriusha* deciso di lavorare su qualità interna ed esterna definendo apposite metriche.

- **Funzionalità** è la capacità di un prodotto software di fornire funzioni che soddisfano esigenze stabilite, necessarie per operare sotto condizioni specifiche.
 - **Appropriatezza**: rappresenta la capacità del prodotto software di fornire un appropriato insieme di funzioni per gli specificati compiti ed obiettivi prefissati all'utente.
 - **Accuratezza**: la capacità del prodotto software di fornire i risultati concordati o i precisi effetti richiesti;
 - **Interoperabilità**: è la capacità del prodotto software di interagire ed operare con uno o più sistemi specificati;
 - **Conformità**: la capacità del prodotto software di aderire a standard, convenzioni e regolamentazioni rilevanti al settore operativo a cui vengono applicate;

- Sicurezza: la capacità del prodotto software di proteggere informazioni e dati negando in ogni modo che persone o sistemi non autorizzati possano accedervi o modificarli, e che a persone o sistemi effettivamente autorizzati non sia negato l'accesso ad essi.
- **Affidabilità:** è la capacità del prodotto software di mantenere uno specificato livello di prestazioni quando usato in date condizioni per un dato periodo.
 - Maturità: è la capacità di un prodotto software di evitare che si verificano errori, malfunzionamenti o siano prodotti risultati non corretti;
 - Tolleranza agli errori: è la capacità di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto;
 - Recuperabilità: è la capacità di un prodotto di ripristinare il livello appropriato di prestazioni e di recupero delle informazioni rilevanti, in seguito a un malfunzionamento. A seguito di un errore, il software può risultare non accessibile per un determinato periodo di tempo, questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità;
 - Aderenza: è la capacità di aderire a standard, regole e convenzioni inerenti all'affidabilità.
- **Usabilità:** è la capacità del prodotto software di essere capito, appreso, usato e benaccetto dall'utente, quando usato sotto condizioni specificate.
 - Comprensibilità: esprime la facilità di comprensione dei concetti del prodotto, mettendo in grado l'utente di comprendere se il software è appropriato.
 - Apprendibilità: è la capacità di ridurre l'impegno richiesto agli utenti per imparare ad usare la sua applicazione;
 - Operabilità: è la capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso;
 - Attrattiva: è la capacità del software di essere piacevole per l'utente che ne fa uso;
 - Conformità: è la capacità del software di aderire a standard o convenzioni relativi all'usabilità.
- **Efficienza:** è la capacità di fornire appropriate prestazioni relativamente alla quantità di risorse usate.
 - Comportamento rispetto al tempo: è la capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento, sotto condizioni determinate;

- Utilizzo delle risorse: è la capacità di utilizzo di quantità e tipo di risorse in maniera adeguata.
- Conformità: è la capacità di aderire a standard e specifiche sull'efficienza_G.
- **Manutenibilità:** è la capacità del software di essere modificato, includendo correzioni, miglioramenti o adattamenti.
 - Analizzabilità: rappresenta la facilità con la quale è possibile analizzare il codice per localizzare un errore nello stesso;
 - Modificabilità: la capacità del prodotto software di permettere l'implementazione di una specificata modifica (sostituzioni componenti);
 - Stabilità: la capacità del software di evitare effetti inaspettati derivanti da modifiche errate;
 - Testabilità: la capacità di essere facilmente testato per validare le modifiche apportate al software.
- **Portabilità:** è la capacità del software di essere trasportato da un ambiente di lavoro ad un altro.
 - Adattabilità: la capacità del software di essere adattato per differenti ambienti operativi senza dover applicare modifiche diverse da quelle fornite per il software considerato;
 - Installabilità: la capacità del software di essere installato in uno specificato ambiente;
 - Conformità: la capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità;
 - Sostituibilità: è la capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso.

A.3 Capability Maturity Model Integration (CMMI)

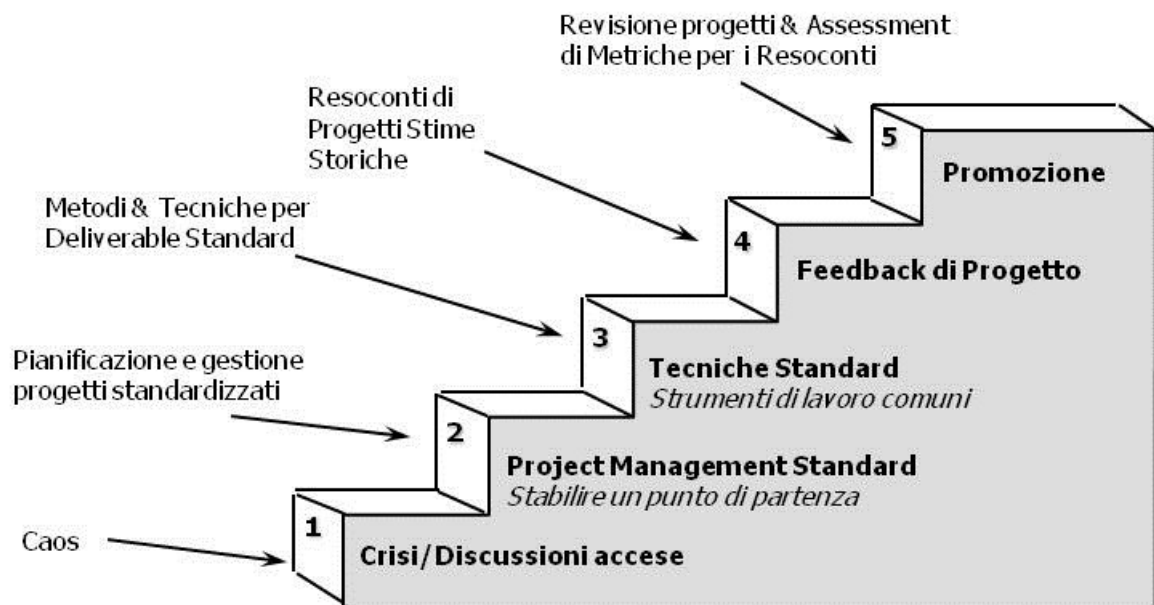


Figura 4: Livello di maturità delle procedure

Il modello identifica cinque livelli di maturità dei processi all'interno di un'organizzazione: dal Livello 1, il processoG più immaturo, o caotico, al Livello 5, il processoG più maturo, o di qualità

- **Livello di maturità 1**

Partendo dall'assunzione che una pratica non può essere migliorata se non è ripetibile, il livello di maturità iniziale vede l'organizzazione effettuare la gestione delle persone tramite procedure ad hoc, spesso informali e non ripetibili se non sporadicamente. Un esempio tipico è data dall'impossibilità, da parte delle persone, di assicurare la data di rilascio del software, indipendentemente dalle tecnologie utilizzate o dalla preparazione delle persone. Un'altra conseguenza tipica è la gestione incontrollata delle modifiche ai requisiti con conseguenze negative sui piani di lavoro. L'attività principale da compiere in questo periodo è quella di aiutare l'organizzazione a rimuovere ogni impedimento alla ripetibilità delle pratiche;

- **Livello di maturità 2**

Al livello di maturità 2, l'organizzazione stabilisce una politica per divulgare presso tutti i gruppi di lavoro i processi stabiliti. Prima di pensare ad ogni miglioramento, l'organizzazione deve assicurare un ambiente di lavoro stabile in cui eseguire in maniera ripetibile i propri processi. Finché si opera in una modalità non strutturata, il management è troppo occupato nel controllo quotidiano

delle operazione per poter pensare a qualsivoglia cambiamento in ottica di miglioramento. L'obiettivo principale del livello 2 è quindi quello di permettere alle persone di svolgere il proprio lavoro in maniera ripetibile, in base a quanto già fatto in passato ed in base all'esperienza maturata. A questo livello il management lascia ai responsabili dei singoli gruppi il compito di controllare il lavoro quotidiano, dedicandosi a sua volta al controllo dei risultati finali e della baseline (ed alle rispettive modifiche). Solo quando le pratiche stabilite saranno eseguite con naturalezza dall'intero gruppo, questo potrà iniziare il periodo successivo di utilizzo di processi comuni a tutto il team;

- **Livello di maturità 3**

Al livello di maturità 3, l'organizzazione seleziona le migliori pratiche e le include in un processo comune. Operando tutti con le stesse pratiche definite, l'organizzazione sarà in grado di valutare le pratiche con migliori performance nell'ambiente comune. Documentate nell'ambito del processo comune le pratiche, queste diventano anche lo strumento di apprendimento per le nuove persone. Le misure effettuate sulle pratiche di maggiore criticità sono registrate in un archivio ed utilizzare per effettuarne l'analisi. In tale modo si è creato il fondamento per una cultura di base comune all'organizzazione: un processo comune conosciuto ed applicato da tutti. E' il fondamento della cultura professionale di base dell'organizzazione;

- **Livello di maturità 4**

Al livello di maturità 4, l'organizzazione inizia a gestire i processi in base ai risultati utilizzando l'analisi delle misure effettuate. Le attività sono svolte secondo i processi comuni definiti ed i risultati sono quindi più controllabili in base all'esperienza storica. Le deviazioni dai risultati attesi sono analizzate, le cause delle deviazioni individuate e le azioni correttive prese di conseguenza. I processi sono quindi gestiti quantitativamente ed i risultati sono prevedibili con maggiore cura. I risultati del business sono controllati da valori e non più dalle *milestone* come prima. Si crea quindi la cultura per un vero miglioramento dei processi e quindi delle performance reali;

- **Livello di maturità 5**

Al livello di maturità 5, l'organizzazione opera utilizzando in maniera ripetitiva i propri processi, ne valuta le performance quantitativamente ed opera per migliorarli di continuo. Gli eventuali difetti sono analizzati e le cause che li generano sono rimosse per evitare il loro ripetersi. Le persone sono culturalmente abituate ad eseguire i processi conosciuti ed il management a gestirli quantitativamente ed a migliorarli. Si crea anche la cultura dell'accettazione del cambiamento. L'organizzazione entra in un circolo virtuoso di miglioramento continuo;

B Resoconto attività di verifica

B.1 Riassunto dell'attività di verifica su RR

Sirius, ha deciso di valorizzare soprattutto i requisiti desiderabili per fare in modo di tenere alto l'indice di efficienza_G. Valutando, durante lo stato di avanzamento, quali di questi requisiti saranno successivamente sviluppati nella realizzazione del prodotto software. Questo tipo di scelte risultano infatti difficoltose allo stato attuale in quanto una pianificazione approfondita necessiterebbe di *lead time*_G precisi che non sono al momento tra le conoscenze dei componenti del gruppo. Nel periodo di tempo che ha portato *Sirius* alla consegna di questa revisione sono stati verificati i **documenti** ed i **processi**.

I **documenti** sono stati verificati anche durante le operazioni di redazione per portare a conoscenza dei contenuti tutti i componenti del gruppo di lavoro. L'analisi statica, in primo luogo utilizzando la tecnica del *walkthrough*, ha portato alla redazione di una lista di controllo che verrà poi incrementata ed utilizzata nell'analisi finale del documento prima di procedere alla consegna. Una volta rilevati gli errori questi sono stati notificati al redattore che ha proceduto alla correzione, evidenziando gli errori frequenti che sono stati utilizzati per migliorare il processo di verifica. *Sirius* adotta il ciclo PDCA per rendere più efficiente_G ed efficace_G nel tempo il processo di verifica.

L'attività di verifica, inoltre, utilizzando la tecnica *inspection* è stata utilizzata principalmente per la verifica dei grafici dei casi d'uso. Per verificare la correttezza dei requisiti richiesti e la successiva completezza ci si è affidati ad un particolare strumento di tracciamento definito in Norme di progetto v2.0.0. L'avanzamento dei processi, dettato dal Piano Di Progetto v2.0.0, è stato mantenuto in controllo tramite una costante verifica delle metriche definite in questo documento e di cui troviamo una rappresentazione grafica in seguito.

B.2 Dettaglio dell'attività di verifica su RR

B.2.1 Documenti

Indice di *Gulpease* per i documenti redatti:

Documento	Valore di accettazione	Esito
Piano Di Progetto v2.0.0	>40	<i>Positivo</i>
Norme di progetto v2.0.0	>40	<i>Positivo</i>
Analisi dei requisiti v3.0.0	>40	<i>Positivo</i>
Piano Di Qualifica v2.0.0	>40	<i>Positivo</i>
Studio Di Fattibilità v2.0.0	>40	<i>Positivo</i>

Tabella 2: esito del calcolo indice di Gulpease per ogni documento.

B.2.2 Processi

Sirius ha condotto l'attività di verifica per i processi. In questo primo periodo il processo di documentazione è predominante nella pianificazione delle risorse. Di seguito viene riportato l'indice **SV** (*schedule variance*) per le attività eseguite e i risultati sono i seguenti:

Attività	Ore pianificate	Ore rilevate	SV rilevato	SV accettazione
Norme di Progetto	17 H	17 H	0 H	> -1 H
Studio di Fattibilità	8 H	14 H	-6 H	> -1 H
Analisi dei Requisiti	70 H	68 H	2 H	> -4 H
Piano di Progetto	37 H	35 H	2 H	> -2 H
Piano di Qualifica	26 H	22 H	4 H	> -2 H

Tabella 3 : Indice SV per le attività.

Da una prima analisi, si denota che *Sirius* ha pianificato in modo preciso le attività. L'attività di Studio di Fattibilità, essendo stato uno dei primi documenti che *Sirius* ha redatto, la pianificazione non è stata precisa questo ha portato ad un SV dell'attività fuori dal range di accettazione. Le cause di questo problema sono da ricercare anche nella poca confidenza con gli strumenti di *editor* testi e con gli strumenti di condivisione. La singola occorrenza del problema, non è quindi indice di allarme per gli altri processi che saranno pianificati nell'avanzamento del prodotto.

- SV-totale = 2 H;

SV-totale maggiore di zero denota che *Sirius* stà producendo più velocemente rispetto a quanto pianificato. Questo può essere una diretta conseguenza dell'aggiunta di uno *slack* temporale nella pianificazione delle attività. Il team ha valutato la possibilità di ridurre il tempo di *slack*, per fare in modo che la pianificazione corrisponda alla realtà; ma data la variabilità delle attività che *Sirius* intende svolgere nel *Sirius* ha effettuato l'analisi dei documenti e ha rilevato la conformità controllando l'indice di Gulpease di tutti i documenti prodotti o modificati in questo periodo di tempo. proseguo del progetto e la poca esperienza, è stato deciso di non modificare tale valore.

Di seguito viene riportato l'indice **BV** (*budget variance*) per le attività eseguite e i risultati sono i seguenti:

Attività	Costo pianificato	Costo consuntivo	BV rilevato	BV limite
Norme di Progetto	325.00 €	325.00 €	0.00 €	> -32.50 €
Studio di Fattibilità	180.00 €	310.00 €	-130.00 €	> -18.00 €
Analisi dei Requisiti	1630.00 €	1600.00 €	30.00 €	> -16.30 €
Piano di Progetto	1005.00 €	945.00 €	60.00 €	> -10.05 €
Piano di Qualifica	490.00 €	420.00 €	70.00 €	> -49.00 €

Tabella 4: Indice BV per le attività.

Come descritto sopra per SV, anche BV denota che il preventivo di costo previsto per le attività svolte è stato corretto. In particolare nell'attività di Studio di Fattibilità il costo a consuntivo è stato maggiore rispetto a quello preventivato. Questo è da collegare al costo orario dell'amministratore e non meno alle cause elencate sopra per l'indice SV che ne è strettamente collegato. Complessivamente *Sirius* ha ottenuto:

- BV-totale = 30.00 €.

il risultato ottenuto è una diretta conseguenza di un preventivo appropriato, e quindi ad un piccolo margine di guadagno nel budget di spesa dell'intero progetto.

B.3 Riassunto dell'attività di verifica su RP

Il processo di progettazione è per *Sirius* un importante strumento per giungere ai periodi successivi con maggiore sicurezza del lavoro da fare e con una stima precisa delle risorse e degli obiettivi che si è deciso di perseguire. Nonostante le tempistiche ridotte, l'organizzazione del lavoro ha voluto che prima di procedere alla progettazione vera e propria si correggessero le opportunità di miglioramento riscontrate in RR, con l'obiettivo di consolidare il grado di maturità dei processi esistenti prima di aggiungerne di nuovi. Nella progettazione, descritta nella Specifica Tecnica v1.0.0, si è provveduto a delineare la struttura del software provando a pianificare la tipologia di test che *Sirius* andrà ad eseguire nei periodi a seguire.

B.4 Dettaglio dell'attività di verifica su RP

B.4.1 Documenti

Indice di *Gulpease* per i documenti redatti:

Documento	Valore di accettazione	Esito
Piano Di Progetto v2.0.0	>40	<i>Positivo</i>
Norme di progetto v2.0.0	>40	<i>Positivo</i>
Analisi dei requisiti v3.0.0	>40	<i>Positivo</i>
Piano Di Qualifica v2.0.0	>40	<i>Positivo</i>
Studio Di Fattibilità v2.0.0	>40	<i>Positivo</i>
Specifica Tecnica v1.0.0	>40	<i>Positivo</i>

Tabella 5: esito del calcolo indice di Gulpease per ogni documento.

B.4.2 Processi

Sirius ha condotto l'attività di verifica per i processi. In questo primo periodo il processo di documentazione è predominante nella pianificazione delle risorse. Di seguito viene riportato l'indice **SV** (*schedule variance*) per le attività eseguite e i risultati sono i seguenti:

Attività	Ore pianificate	Ore rilevate	SV rilevato	SV accettazione
Norme di Progetto	5 H	9 H	-4 H	> -1 H
Analisi dei Requisiti	53 H	25 H	27 H	> -3 H
Piano di Progetto	42 H	25 H	17 H	> -2 H
Piano di Qualifica	7 H	23 H	-16 H	> -1 H
Specifica Tecnica	74 H	90 H	-32 H	> -4 H

Tabella 6 : Indice SV per le attività.

Da una prima analisi, si denota che *Sirius* ha pianificato in modo preciso le attività. Questo ha fatto in modo che la progettazione architettuale sia stata svolta più velocemente rispetto a quanto pianificato. La causa principale può essere ricercata nella velocità di accordo tra i progettisti di *Sirius*, che hanno definito in modo rapido la progettazione del prodotto. Questo ha portato al risultato positivo.

- SV-totale = 2 H;

SV-totale maggiore di zero denota che *Sirius* stà producendo più velocemente rispetto a quanto pianificato. Questo può essere una diretta conseguenza dell'aggiunta di uno *slack* temporale nella pianificazione delle attività. Il team ha valutato la possibilità di ridurre il tempo di *slack*, per fare in modo che la pianificazione corrisponda alla realtà; ma data la variabilità delle attività che *Sirius* intende svolgere nel proseguo del progetto e la poca esperienza, è stato deciso di non modificare tale valore.

Di seguito viene riportato l'indice **BV** (*budget variance*) per le attività eseguite e i risultati sono i seguenti:

Attività	Costo pianificato	Costo consuntivo	BV rilevato	BV limite
Norme di Progetto	61.00 €	105.00 €	-44.00 €	> -6.10 €
Analisi dei Requisiti	1253.00 €	578.00 €	675.00 €	> -125.30 €
Piano di Progetto	1182.00 €	705.00 €	477.00 €	> -118.20 €
Piano di Qualifica	71.00 €	582.00 €	-511.00 €	> -7.10 €
Specifica Tecnica	1584.00 €	2132.00 €	-548.00 €	> -158.40 €

Tabella 7: Indice BV per le attività.

Come descritto sopra per SV, anche BV denota che il preventivo di costo previsto per le attività svolte è stato corretto. In particolare la stesura della specifica tecnica ha richiesto più budget di quello preventivato, le altre attività non hanno però consumato del tutto il proprio budget. Questo nonostante le correzioni da effettuare sulla documentazione prodotta in RR. Complessivamente *Sirius* ha ottenuto:

- BV-totale = 49.00 €.

il risultato ottenuto è una diretta conseguenza di un preventivo appropriato, e quindi ad un piccolo margine di guadagno nel budget di spesa dell'intero progetto.

B.5 Riassunto dell'attività di verifica su RQ

Sirius, ha deciso di valorizzare soprattutto i requisiti desiderabili per fare in modo di tenere alto l'indice di efficienza_G. Valutando, durante lo stato di avanzamento, quali di questi requisiti saranno successivamente sviluppati nella realizzazione del prodotto software. Questo tipo di scelte risultano infatti difficoltose allo stato attuale in quanto una pianificazione approfondita necessiterebbe di *lead time*_G precisi che non sono al momento tra le conoscenze dei componenti del gruppo. Nel periodo di tempo che ha portato *Sirius* alla consegna di questa revisione sono stati verificati i **documenti** ed i **processi**.

I **documenti** sono stati verificati anche durante le operazioni di redazione per portare a conoscenza dei contenuti tutti i componenti del gruppo di lavoro. L'analisi statica, in primo luogo utilizzando la tecnica del *walkthrough*, ha portato alla redazione di una lista di controllo che verrà poi incrementata ed utilizzata nell'analisi finale del documento prima di procedere alla consegna. Una volta rilevati gli errori questi sono stati notificati

al redattore che ha proceduto alla correzione, evidenziando gli errori frequenti che sono stati utilizzati per migliorare il processo di verifica. *Sirius* adotta il ciclo PDCA per rendere più efficiente_G ed efficace_G nel tempo il processo di verifica.

L'attività di verifica, inoltre, utilizzando la tecnica *inspection* è stata utilizzata principalmente per la verifica dei grafici dei casi d'uso. Per verificare la correttezza dei requisiti richiesti e la successiva completezza ci si è affidati ad un particolare strumento di tracciamento definito in Norme di progetto v2.0.0. L'avanzamento dei processi, dettato dal Piano Di Progetto v2.0.0, è stato mantenuto in controllo tramite una costante verifica delle metriche definite in questo documento e di cui troviamo una rappresentazione grafica in seguito.

B.6 Dettaglio dell'attività di verifica su RQ

B.6.1 Documenti

Indice di *Gulpease* per i documenti redatti:

Documento	Valore di accettazione	Esito
Piano Di Progetto v2.0.0	>40	<i>Positivo</i>
Norme di progetto v2.0.0	>40	<i>Positivo</i>
Analisi dei requisiti v3.0.0	>40	<i>Positivo</i>
Piano Di Qualifica v2.0.0	>40	<i>Positivo</i>
Studio Di Fattibilità v2.0.0	>40	<i>Positivo</i>
Definizione Di Prodotto v1.0.0	>40	<i>Positivo</i>
Manuale utente v1.0.0	>40	<i>Positivo</i>
Manuale process owner v1.0.0	>40	<i>Positivo</i>
Specifica Tecnica v1.0.0	>40	<i>Positivo</i>

Tabella 2: esito del calcolo indice di Gulpease per ogni documento.

B.6.2 Processi

Sirius ha condotto l'attività di verifica per i processi. Di seguito è riportato l'indice **SV** (*schedule variance*) per le attività eseguite.

Attività	Ore pianificate	Ore rilevate	SV rilevato	SV accettazione
Norme di Progetto	8 H	6 H	2 H	> -1 H
Piano di Progetto	13 H	6 H	7 H	> -1 H
Piano di Qualifica	19 H	27 H	-8 H	> -1 H
Specifica Tecnica	7 H	56 H	-49 H	> -1 H
Definizione di Prodotto	78 H	84 H	-6 H	> -4 H
Codifica	51 H	48 H	3 H	> -3 H
Manuale Utente	17 H	15 H	2 H	> -1 H
Attività di Verifica	93 H	40 H	53 H	> -5 H

Tabella 3 : Indice SV per le attività.

Dall'analisi della tabella sopra si denota come la pianificazione del periodo sia stata poco efficace. Questo è dovuto ad una attività di progettazione svolta in modo non preciso, che ha portato ad una correzione e nuova stesura di alcuni documenti. Inoltre, si vede come la pianificazione dei test e la progettazione di dettaglio siano state sottovalutate nella pianificazione del periodo. Del resto l'attività di verifica e di codifica, sulla quale si era pianificato un notevole monte ore, sono risultate essere attività sopravvalutate.

- SV-totale = 4 H;

Di seguito viene riportato l'indice **BV** (*budget variance*) per le attività eseguite e i risultati sono i seguenti:

Attività	Costo pianificato	Costo consuntivo	BV rilevato	BV limite
Norme di Progetto	160.00 €	120.00 €	40.00 €	> -16.00 €
Piano di Progetto	390.00 €	180.00 €	210.00 €	> -39.00 €
Piano di Qualifica	475.00 €	675.00 €	-200.00 €	> -47.00 €
Specifica Tecnica	154.00 €	1232.00 €	-1078.00 €	> -15.00 €
Definizione di Prodotto	1759.00 €	1812.00 €	-53.00 €	> -176.00 €
Codifica	765.00 €	720.00 €	45.00 €	> -77.00 €
Manuale Utente	374.00 €	330.00 €	44.00 €	> -37.00 €
Verifica	1395.00 €	600.00 €	795.00 €	> -140.00 €

Tabella 4: Indice BV per le attività.

Complessivamente *Sirius* ha ottenuto:

- BV-totale = -197.00 €.

B.6.3 Risultati delle misurazioni sul codice

Di seguito i risultati dei test di analisi statica e dinamica effettuati sul codice fin'ora prodotto. Per ogni metrica viene indicata la media e il valore massimo, giustificando gli eventuali valori fuori range.

Metrica	Media	Massimo	Accettazione
Complessità ciclomatica	1	10	
Livelli di annidamento	1	10	Acc.
Attributi per classe	1	10	Acc.
Parametri per metodo	1	10	Acc.
Linee di codice per linee commento	1	10	Acc.
Accoppiamento	1	10	Acc.
Copertura	53%		N. Acc.

Tabella 4: Risultati misurazioni sul codice.