

QUIZZIPEDIA



team404swe@gmail.com

Piano di qualifica 2.0

Informazioni sul documento	
Nome Documento	Piano di qualifica
Versione	2.0
Uso	Esterno
Data Creazione	21 dicembre 2015
Data Ultima Modifica	3 luglio 2016
Redazione	Alex Beccaro
Verifica	Davide Bortot - Marco Crivellaro
Approvazione	Martin Vadice Mbouenda
Committente	Zucchetti SPA
Lista di distribuzione	Prof. Vardanega Tullio TEAM404

Registro delle modifiche

Versione	Autore	Data	Descrizione
2.0	M. V. Mbouenda (Responsabile)	14/05/2016	Approvazione documento
1.6.2	M. Crivellaro (Analista/Ver.)	13/05/2016	Verifica finale
1.6.1	A. Beccaro (Verificatore)	12/05/2016	Miglioramento impaginazione tabelle e completamento sezione test
1.6	A. Beccaro (Verificatore)	05/05/2016	Aggiunta sezione test di sistema e requisiti di qualità
1.5	A. Beccaro (Verificatore)	27/04/2016	Rivista sezione gestione amministrativa, aggiunta sezione procedure controllo qualità processi, rifatta sezione organizzazione
1.4	A. Beccaro (Verificatore)	26/04/2016	Inserimento test di accettazione
1.3	A. Beccaro (Verificatore)	25/04/2016	Ampliato paragrafo sui parametri di tolleranza
1.2	A. Beccaro (Verificatore)	17/04/2016	Aggiunto paragrafo Procedure di controllo di qualità processi
1.1	A. Beccaro (Verificatore)	16/04/2016	Modifica struttura documento in seguito alla RR
1.0	D. Bortot (Responsabile)	16/03/2016	Approvazione documento.
0.2	L. Alessio (Verificatore)	15/03/2016	Verifica documento completo.
0.1.1	A. Multineddu (Verificatore)	11/03/2016	Correzione errori di impaginazione e tipografici
0.1	M. Crivellaro (Verificatore)	10/03/2016	Verifica documento.
0.0.7	A. Multineddu (Verificatore)	19/01/2016	Correzione errori nella sezione Misure e metriche
0.0.6	A. Multineddu (Verificatore)	18/01/2016	Aggiunta metriche e tabella resoconto misure
0.0.5	M. Crivellaro (Verificatore)	16/01/2016	Modifica layout e aggiunta file template.tex
0.0.4	A. Multineddu (Verificatore)	07/01/2016	Redazione sezione Misure e metriche
0.0.3	A. Multineddu (Verificatore)	03/01/2016	Redazione sezione Risorse per la verifica
0.0.2	A. Multineddu (Verificatore)	22/12/2015	Redazione sezione Strategia generale di qualifica
0.0.1	A. Multineddu (Verificatore)	21/12/2015	Prima stesura del documento.

Indice

1	Introduzione	2
1.1	Scopo del documento	2
1.2	Scopo del prodotto	2
1.3	Glossario	2
1.4	Riferimenti	3
1.4.1	Normativi	3
1.4.2	Informativi	3
2	Strategia generale di qualifica	4
2.1	Definizione obiettivi	4
2.2	Qualità di processo	4
2.2.1	Standard utilizzati	5
2.3	Qualità di prodotto	6
2.3.1	Standard utilizzati	6
3	Verifica qualitativa sui processi	7
3.1	Infrastructure Management Process	7
3.1.1	Obiettivi e strategie	7
3.1.2	Metriche	7
3.2	Project Planning, Assessment and Control Process	7
3.2.1	Obiettivi e strategie	7
3.2.2	Metriche	8
3.2.2.1	Schedule Variance	8
3.2.2.2	Budget variance	8
3.3	System/Software Requirements Analysis Process	9
3.3.1	Obiettivi e strategie	9
3.3.2	Metriche e misure	9
3.3.2.1	Copertura dei requisiti DA CORREGGERE	9
3.4	System/Software Architectural Design Process	9
3.4.1	Obiettivi	9
3.4.2	Metriche e misure	10
3.4.2.1	Change cost	10
3.4.2.2	Core size	10
3.5	Software Detailed Design Process	10
3.5.1	Obiettivi	10
3.5.2	Metriche e misure	11
3.5.2.1	Parameter count	11
3.5.2.2	XXXNumber of methods	11
3.6	Software Construction Process	11
3.6.1	Obiettivi	11
3.6.2	Metriche e misure	11
3.6.2.1	Cyclomatic complexity	11
3.6.2.2	Metriche di Halstead	11
3.6.2.3	Dependency count	12
3.6.2.4	Maintainability Index	12
3.7	System/Software Qualification Testing Process	13
3.7.1	Obiettivi e strategie	13

3.7.2	Metriche e misure	13
3.7.2.1	Test di Accettazione eseguiti	13
3.7.2.2	Test di Sistema eseguiti	13
3.7.2.3	Test di Integrazione eseguiti	14
3.7.2.4	Test di Unità eseguiti	14
3.8	Software Documentation Management Process	14
3.8.1	Obiettivi e strategie	14
3.8.2	Metriche e misure	15
3.8.3	Indice Gulpease	15
4	Misure e metriche: riepilogo parametri di tolleranza	16
4.1	Processi	16
4.2	Documenti	16
4.3	Software	16
	Appendici	17
A	Specifica test	17
A.1	Livelli di testing	17
A.2	Test di accettazione	18
A.3	Test di sistema	23
A.3.1	Requisiti funzionali	23
A.3.2	Requisiti di qualità	30
A.4	Test di integrazione	31
A.5	Test di unità	32
B	Tracciamento componenti – requisiti	32
C	Tracciamento requisiti – componenti	33
D	Tracciamento metodi – test di unità	33
E	Esito misurazioni	35
E.1	Analisi	35
E.1.1	Processi	35
E.1.2	Documenti	35
E.2	Progettazione Architettuale	36

Elenco delle figure

1	PDCA e miglioramento complessivo	4
2	Riepilogo caratteristiche ISO/IEC 9126:2001	6
3	Fasi di sviluppo	17

Sommario

Il presente documento contiene il *piano di qualifica* del capitolato **Quizzipedia**. Vengono rese note le strategie di verifica qualitativa dei processi e del prodotto utilizzate dal gruppo **Team404**, le strategie che coinvolgono l'attuazione di modelli standard e l'utilizzo di metriche e misure necessarie alla valutazione oggettiva del processo o prodotto in analisi.

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di esporre le strategie che il gruppo **Team404** intende adottare per assicurare la qualità del prodotto software **Quizzipedia** e dei processi coinvolti per il suo sviluppo.

1.2 Scopo del prodotto

Il progetto **Quizzipedia** ha come obiettivo lo sviluppo di un sistema software basato su tecnologie Web (Javascript₆, Node.js₆, HTML5₆, CSS3₆) che permetta la creazione, gestione e fruizione di questionari. Il sistema dovrà quindi poter archiviare i questionari suddivisi per argomento, le cui domande dovranno essere raccolte attraverso uno specifico linguaggio di markup (Quiz Markup Language) d'ora in poi denominato QML₆. In un caso d'uso a titolo esemplificativo, un esaminatore dovrà poter costruire il proprio questionario scegliendo tra le domande archiviate, ed il questionario così composto sarà presentato e fruibile all'esaminando, traducendo l'oggetto QML in una pagina HTML₆, tramite un'apposita interfaccia web. Il sistema presentato dovrà inoltre poter proporre questionari preconfezionati e valutare le risposte fornite dall'utente finale.

Per un'analisi più precisa ed approfondita del progetto si rimanda al documento *analisi_dei_requisiti_2.0.pdf*.

1.3 Glossario

Viene allegato un glossario nel file "*glossario_2.0.pdf*" nel quale viene data una definizione a tutti i termini che in questo documento appaiono con il simbolo '6' a pedice.

1.4 Riferimenti

1.4.1 Normativi

- Capitolato d'appalto Quizzipedia:
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C5.pdf>
- Norme di Progetto: *norme_di_progetto_2.0.pdf*

1.4.2 Informativi

- Corso di Ingegneria del Software anno 2015/2016:
<http://www.math.unipd.it/~tullio/IS-1/2015/>
- Regole del progetto didattico:
<http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/PD01.pdf> <http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/>
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/PD01b.html>
- Metriche di progetto:
https://it.wikipedia.org/wiki/Metriche_di_progetto
- Metriche per il software
<http://www.verifysoft.com/en-software-complexity-metrics.pdf>
- Complexity-report:
<https://github.com/jared-stilwell/complexity-report>

2 Strategia generale di qualifica

2.1 Definizione obiettivi

Il gruppo di lavoro, per garantire la qualità del prodotto software **Quizzipedia** e dei processi coinvolti al suo sviluppo, intende adottare metodologie standard ed il più possibile automatizzabili per ridurre al minimo l'attività umana e favorire processi automatici laddove sia possibile. Vengono di seguito elencati gli obiettivi di qualità che si vogliono raggiungere per quanto riguarda il prodotto finale e i processi coinvolti per lo sviluppo dello stesso, e le strategie necessarie all'attuazione della corretta verifica e validazione del prodotto.

Lo scopo principale è quello di rendere definibili e misurabili tutti i processi di sviluppo e verifica per poterne controllare l'andamento e la produttività.

2.2 Qualità di processo

Ogni processo coinvolto nell'attività di sviluppo ha bisogno di una costante attività di verifica di supporto in grado di identificare possibili miglioramenti o peggioramenti ed apportare eventuali correzioni.

Il modello che il gruppo intende quindi seguire è il cosiddetto PDCA (o Ciclo di Deming), modello volto al miglioramento continuo che definisce quattro fasi cicliche:

- **Plan:** definizione del problema, cosa deve essere realizzato e come andrà controllato per la verifica qualitativa;
- **Do:** eseguire le attività secondo i piani;
- **Check:** verifica nel tempo dei risultati conseguiti in seguito a modifiche e migliorie, confronto tra risultati attesi e risultati effettivi;
- **Act:** applicazione di soluzioni correttive atte al miglioramento.

Ogni processo deve essere sottoposto a verifica in modo da poterlo migliorare all'iterazione successiva. Le due fasi chiave del modello PDCA sono la prima fase di definizione (P) del problema e la fase di verifica dei risultati attesi (C). Il concetto generale fa riferimento anche al

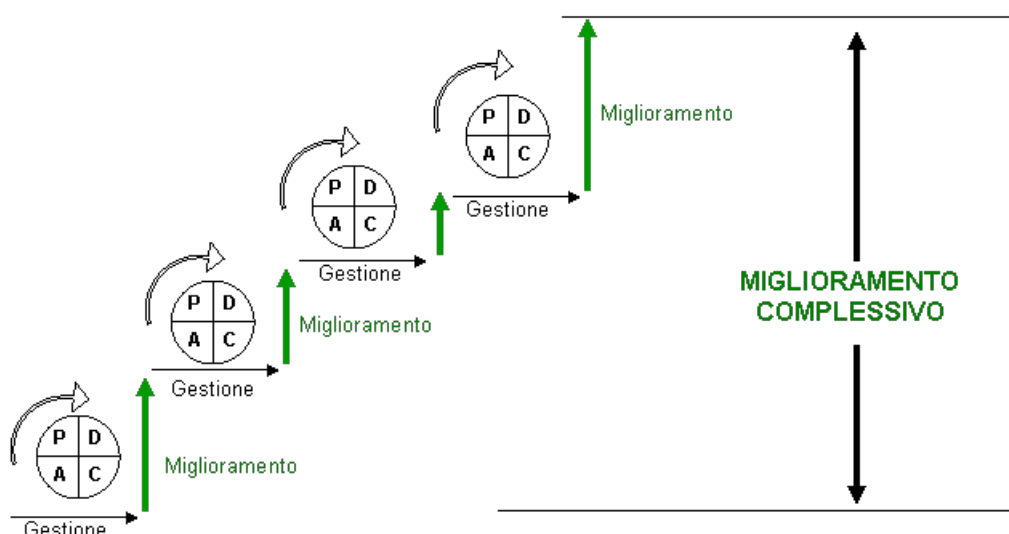


Figura 1: PDCA e miglioramento complessivo

metodo di *Correctness by construction*, che ha come principio base quello di fare in modo di non introdurre errori già dall'inizio e/o di correggere tali errori nel **momento più vicino possibile a quando sono stati introdotti**.

L'attività costante di verifica, quindi, deve servire proprio a far sì che gli errori e le divergenze vengano individuate il prima possibile. Piccoli errori iniziali, se non gestiti, possono portare all'assoluta ingestibilità dell'attività di verifica e quindi compromettere pesantemente il successo finale del progetto. Per evitare ciò, un processo dipendente da un altro non può avere inizio finché il precedente non sia stato verificato e ne sia stata accertata la correttezza.

Le linee guida fornite dal modell PDCA per l'attività di controllo ed il miglioramento continuo dei processi sono:

- Pianificazione dettagliata
- Monitoraggio costante delle attività pianificate in esecuzione
- Definizione delle risorse umane e tecnologiche necessarie per il conseguimento degli obiettivi
- Utilizzo di metriche per quantificare e verificare il miglioramento della qualità del processo

Sui processi, quindi, vanno effettuate delle misurazioni oggettive che possano dare informazioni utili sull'andamento degli stessi, in modo da poter decidere se intervenire in maniera migliorativa. In generale, tali misurazioni tengono conto dei seguenti elementi:

- Tempo impiegato per il completamento
- Risorse utilizzate
- Attinenza alla pianificazione

2.2.1 Standard utilizzati

Per quanto riguarda la valutazione qualitativa dei processi si fa riferimento ai seguenti standard:

- SPICE₆: definito dallo standard ISO/IEC 15504:1998, fornisce un modello per la valutazione del livello di "maturità" dei processi per identificare quali azioni possono essere necessarie per migliorare un processo specifico.
- PDCA₆: metodo di gestione iterativo utilizzato per il controllo e il miglioramento continuo dei processi e dei prodotti.
- ISO/IEC 12207:2008: utilizzato come riferimento per l'individuazione dei processi da sottoporre ad attività di verifica qualitativa:
 - Infrastructure Management Process
 - Risk Management ProcessXXXX
 - System/Software Requirements Analysis Process
 - System/Software Architectural Design Process
 - Software Detailed Design Process
 - Software Construction Process
 - System/Software Integration Process
 - System/Software Qualification Testing Process
 - Software Documentation Management Process
 - Software Verification Process

2.3 Qualità di prodotto

AGGIUNGERE TESTO

- **Software Quality Assurance:** insieme delle attività volte a garantire il raggiungimento degli obiettivi di qualità. Queste attività prevedono l'utilizzo di tecniche di analisi statica e dinamica
- **Verifica:** attività costante durante l'intera fase del progetto; serve a controllare che l'output aspettato dalle entità in oggetto di verifica.
- **Validazione:** la conferma oggettiva che il sistema soddisfi i requisiti.

2.3.1 Standard utilizzati

Come punto di riferimento per la qualità del prodotto software si fa riferimento allo standard ISO/IEC 9126:2001.

L'immagine seguente elenca le principali caratteristiche a cui un prodotto software di qualità deve aspirare:

Funzionabilità	Affidabilità	Efficienza	Usabilità	Manutenibilità	Portabilità
appropriatezza accuratezza interoperabilità conformità sicurezza	maturità tolleranza agli errori recuperabilità aderenza	comportamento rispetto al tempo utilizzo di risorse conformità	comprensibilità apprendibilità operabilità attrattiva conformità	analizzabilità modificabilità stabilità testabilità collaudabilità	adattabilità installabilità conformità sostituibilità

Figura 2: Riepilogo caratteristiche ISO/IEC 9126:2001

3 Verifica qualitativa sui processi

3.1 Infrastructure Management Process

Processo di gestione dell'infrastruttura: ha lo scopo di monitorare e mantenere operativo l'ambiente di lavoro, verificare che siano presenti e funzionanti le risorse necessarie allo sviluppo del progetto. L'infrastruttura è rappresentata da tutte le risorse tecnologiche (hardware, software, metodologie standard) ed umane impiegate nello sviluppo del prodotto.

3.1.1 Obiettivi e strategie

Gli obiettivi specifici di questo processo sono volti alla verifica della disponibilità e al corretto e costante utilizzo (da parte dei membri del team) delle risorse tecnologiche da utilizzare. Il responsabile deve inoltre accertarsi che ogni membro del gruppo abbia letto e compreso il documento (preventivamente redatto) *Norme-di-progetto-2.0*.

Devono quindi essere definiti e correttamente configurati gli strumenti di comunicazione e coordinamento Trello e GitHub. Tramite queste piattaforme i verificatori possono tenere traccia dello storico dei ticket assegnati, valutarne quindi i tempi di completamento e agire in modo propositivo nel caso si verifichino ritardi nella realizzazione di compiti assegnati.

La verifica del completamento dei ticket viene integrata con un controllo altrettanto costante dell'attività dei commit sulla piattaforma GitHub. Sarà quindi necessario osservare e verificare l'andamento dei commit, verificandone sia correttezza che frequenza.

L'infrastruttura quindi dovrà tenere conto dei seguenti obiettivi:

- Corretta configurazione iniziale delle piattaforme Trello e GitHub per l'utilizzo da parte di tutti i membri del gruppo
- Monitoraggio dell'effettivo utilizzo delle piattaforme Trello e GitHub da parte di tutti i membri
- Monitoraggio del tempo trascorso tra l'assegnazione del ticket ed il suo completamento

3.1.2 Metriche

DA FARE

3.2 Project Planning, Assessment and Control Process

Processo che deve fornire la pianificazione dettagliata delle attività di sviluppo del progetto come la suddivisione temporale delle varie fasi di sviluppo, la scelta del modello di ciclo di vita del prodotto, l'assegnazione dei ruoli, i preventivi ed i consuntivi.

3.2.1 Obiettivi e strategie

La verifica per questa fase deve accertarsi che:

- i ruoli e le attività siano correttamente assegnati secondo la pianificazione specificata nel documento *Piano-di-Progetto-2.0.pdf* redatto dal Responsabile.

- i costi effettivi per ogni fase non eccedano rispetto ai costi preventivati

3.2.2 Metriche

Per il controllo e la misurazione dei processi in termini di tempo (schedule) e in termini di costo e risorse impiegate (budget), vengono utilizzate le metriche (**Schedule variance** e **Budget variance**) Tali misurazioni, da effettuare alla fine di ogni fase, si basano sui valori dei consuntivi di ogni fase (presenti nel documento *piano_di_progetto_2.0.pdf*). La valutazione di queste metriche è essenziale per verificare l'andamento di ogni processo misurato e valutarne quindi il miglioramento o il peggioramento complessivo.

3.2.2.1 Schedule Variance

Il valore SV (schedule variance) indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di progetto pianificate nella baseline.

Se $SV > 0$ significa che il progetto sta producendo (ossia rilasciando deliverable) con maggior velocità a quanto pianificato, viceversa se negativo. Formula:

$$SV = BCWP - BCWS$$

Dati:

- **BCWP** (Budgeted Cost of Work Performed): Valore delle attività realizzate alla data corrente.
- **BCWS** (Budgeted Cost of Work Scheduled): Costo pianificato per realizzare le attività di progetto alla data corrente.
- Range-accettazione: $[\geq -(PreventivoFase * 5\%)]$
- Range-ottimale: $[\geq 0]$

3.2.2.2 Budget variance

Indica se alla data corrente si è speso di più o di meno rispetto a quanto previsto a budget alla data corrente.

Formula:

$$BV = BCWS - ACWP$$

Dati:

- **ACWP** (Actual Cost of Work Performed): Costo effettivamente sostenuto alla data corrente.

Se $BV > 0$ significa che il progetto sta spendendo il proprio budget con minor velocità di quanto pianificato, viceversa se negativo. Il fatto di spendere più velocemente il budget non ha nulla a che fare con il risparmio che se ne può avere, rappresentato invece da CV.

Parametri utilizzati:

- Range-accettazione: $[\geq -(PreventivoFase * 10\%)]$
- Range-ottimale: $[\geq 0]$

3.3 System/Software Requirements Analysis Process

Processo che ha lo scopo di definire e catalogare precisamente tutti i requisiti. Produce il documento Analisi dei Requisiti.

3.3.1 Obiettivi e strategie

La verifica su questo processo deve tenere conto di:

- Correttezza dei diagrammi UML per quanto riguarda l'adeguata rappresentazione grafica degli Use Case. Verifica, leggibilità e adeguato livello di granularità;
- Correttezza degli Use Case (testuali e grafici) secondo lo standard UML2.0;
- Correttezza delle precondizioni e delle postcondizioni di ogni Use Case;
- Completezza dei requisiti in base alle esigenze di capitolato;
- Tracciabilità dei requisiti tramite tabella di mappatura requisiti - casi d'uso;
- Controllo eventuali conflitti o imprecisioni nella codifica dei requisiti e dei casi d'uso.

3.3.2 Metriche e misure

3.3.2.1 Copertura dei requisiti DA CORREGGERE

Indica la percentuale dei casi testati rispetto alla totalità dei casi da testare. Una percentuale del 100% può essere auspicabile solo se sono stati ben definiti i casi che necessitano realmente di essere testati.

$$\text{Copertura dei test} = \text{Numero funzioni testate} * 100 / \text{Numero funzioni da testare}$$

Parametri utilizzati:

- Range-accettazione: [70 – 100]
- Range-ottimale: [80 – 100]

3.4 System/Software Architectural Design Process

Processo che produce il documento Specifica Tecnica che descrive l'architettura ad alto livello del sistema secondo i requisiti. La progettazione ad alto livello deve soddisfare i requisiti emersi dall'analisi.

3.4.1 Obiettivi

Deve essere verificato che la progettazione ad alto livello presentata nel documento Specifica Tecnica soddisfi i requisiti.

I diagrammi presenti in tale documento devono essere verificati secondo le seguenti linee guida:

- correttezza delle relazioni tra i package
- i package devono essere descritti per struttura e a livello logico

- verifica della correttezza del flusso di ogni diagramma
- verifica della corretta individuazione degli attori

L'architettura ad alto livello presentata nel documento deve soddisfare i seguenti requisiti:

- basso accoppiamento ed alta coesione
- i componenti devono soddisfare tutti i requisiti
- tracciabilità di ogni componente con il corrispondente requisito

3.4.2 Metriche e misure

3.4.2.1 Change cost

Percentuale dei moduli affetti da cambiamento quando un modulo all'interno del progetto viene modificato.

- Range di accettazione: $[\leq 50\%]$
- Range ottimale: $[\leq 40\%]$

3.4.2.2 Core size

La percentuale dei moduli che hanno molte dipendenze verso (e da) altri moduli.

- Range di accettazione: $[\leq 30\%]$
- Range ottimale: $[\leq 25\%]$

3.5 Software Detailed Design Process

Processo che produce il documento Definizione di Prodotto che descrive l'architettura a basso livello del sistema.

3.5.1 Obiettivi

L'attività di verifica deve avere i seguenti obiettivi:

- corretto utilizzo di interfacce e classi astratte
- le classi inserite non devono essere ridondanti o addirittura inutilizzate
- verifica che nei diagrammi di ogni classe siano presenti i tipi di ogni parametro dei metodi e degli attributi di classe
- verifica mirata al controllo della visibilità dei metodi, dei costruttori (se presenti nel diagramma) e degli attributi. I metodi pubblici devono essere limitati allo stretto necessario

3.5.2 Metriche e misure

3.5.2.1 Parameter count

Numero di parametri per funzione. Valori bassi sono da preferire. Parametri utilizzati:

- Range di accettazione: $[0 - 7]$
- Range ottimale: $[0 - 5]$

3.5.2.2 XXXNumber of methods

Numero di metodi definiti in una classe. Parametri utilizzati:

- Range di accettazione: $[1 - 7]$
- Range ottimale: $[1 - 10]$

3.6 Software Construction Process

Questo processo coinvolge l'attività di codifica vera e propria secondo le direttive definite nel documento *Definizione-di-dettaglio-1.0.pdf*.

3.6.1 Obiettivi

L'attività di verifica deve essere volta al controllo della qualità del codice prodotto. A tal proposito vengono utilizzate le metriche descritte nella sezione successiva. Sarà inoltre compito del programmatore di produrre codice facilmente manutenibile e comprensibile, per facilitare anche l'attività di test su di esso.

3.6.2 Metriche e misure

3.6.2.1 Cyclomatic complexity

Metrica software usata per indicare la complessità ciclomatica di un programma. Rappresenta una misura quantitativa del numero di cammini linearmente indipendenti che si possono percorrere nel codice sorgente. La complessità ciclomatica può essere misurata per funzioni individuali, metodi e classi all'interno di un programma.

- Range di accettazione: $[0 - 15]$
- Range ottimale: $[0 - 10]$

3.6.2.2 Metriche di Halstead

Queste metriche sono state concepite per identificare proprietà misurabili del codice e le relazioni tra di esse. Questi numeri sono staticamente calcolati dal codice sorgente.

Dati:

- n_1 = numero di operatori distinti;
- n_2 = numero di operandi distinti;

- N_1 = numero totale degli operatori;
- N_2 = numero totale degli operandi.

Da questi numeri si possono calcolare le seguenti misure:

- Program Vocabulary: $n = n_1 + n_2$
- Program Length: $N = N_1 + N_2$
- **Volume:** $V = N \log_2 n$
 - * Range di accettazione: [20 – 1500]
 - * Range ottimale: [20 – 1000]
- **Difficulty:** $D = \frac{n_1}{2} * \frac{N_1}{n_2}$
 - * Range di accettazione: [0 – 30]
 - * Range ottimale: [0 – 15]
- **Effort:** $E = D * V$
 - * Range di accettazione: [0 – 400]
 - * Range ottimale: [0 – 300]

3.6.2.3 Dependency count

Conteggio del numero di chiamate di tipo **require**₆ di ogni metodo. Valori bassi sono da preferire.

3.6.2.4 Maintainability Index

Rappresenta l'indice principale dei risultati dell'analisi effettuata da Plato sull'intero codice. Compreso tra 0 e 100, questo indice rappresenta la relativa facilità di manutenzione del codice analizzato. Valori alti rappresentano miglior manutenibilità. Questo indice viene calcolato utilizzando la seguente formula:

$$MI = MAX \left[0, 100 \frac{171 - 5.2 \ln V - 0.23G - 16.2 \ln L}{171} \right]$$

dove:

- **MI** = Maintainability Index
- **V** = Halstead Volume
- **L** = Source Lines of Code (SLOC)
- **G** = Complessità Ciclomantica totale

Parametri utilizzati:

- Range di accettazione: [20 – 100]
- Range ottimale: [70 – 100]

3.7 System/Software Qualification Testing Process

Scopo di questo processo è di assicurare e verificare l'implementazione corretta di ogni requisito individuato.

3.7.1 Obiettivi e strategie

La modalità di verifica comporta la creazione e l'esecuzione di test adeguati. A tal proposito si intende usare il framework di testing Jasmine. I test da pianificare ed effettuare, come spiegato nel documento *Norme-di-progetto-2.0.pdf*, sono divisi in quattro categorie:

- Test di Accettazione (TA)
- Test di Sistema (TS)
- Test di Integrazione (TI)
- Test di Unità (TU)

3.7.2 Metriche e misure

3.7.2.1 Test di Accettazione eseguiti

Indica la percentuale di test di accettazione eseguiti (AE) rispetto al numero di test pianificati.

$$AE = \text{Numero test eseguiti} * 100 / \text{Numero test pianificati}$$

Parametri utilizzati:

- Range-accettazione: [100%]
- Range-ottimale: [100%]

3.7.2.2 Test di Sistema eseguiti

Indica la percentuale di test di sistema eseguiti rispetto al numero di test pianificati. Indica la percentuale di test di sistema eseguiti (SE) rispetto al numero di test pianificati.

$$SE = \text{Numero test eseguiti} * 100 / \text{Numero test pianificati}$$

Parametri utilizzati:

- Range-accettazione: [100%]
- Range-ottimale: [100%]

3.7.2.3 Test di Integrazione eseguiti

Indica la percentuale di test di integrazione eseguiti (IE) rispetto al numero di test pianificati.

$$IE = \text{Numero test eseguiti} * 100 / \text{Numero test pianificati}$$

Parametri utilizzati:

- Range-accettazione: [70%]
- Range-ottimale: [60%]

3.7.2.4 Test di Unità eseguiti

Indica la percentuale di test di unità eseguiti rispetto al numero di test pianificati.

Indica la percentuale di test di unità eseguiti (AE) rispetto al numero di test pianificati.

$$UE = \text{Numero test eseguiti} * 100 / \text{Numero test pianificati}$$

Parametri utilizzati:

- Range-accettazione: [100%]
- Range-ottimale: [100%]

3.8 Software Documentation Management Process

Processo che mira a produrre la documentazione di supporto necessaria allo sviluppo del prodotto.

3.8.1 Obiettivi e strategie

La documentazione subisce un processo di verifica secondo le seguenti linee guida:

- Verifica della presenza di tutti i documenti necessari;
- Attinenza di ogni documento alle specifiche di stile e formattazione presenti nel documento *Norme-di-progetto-2.0*;
- Verifica della presenza e del corretto utilizzo del registro delle modifiche interno ad ogni documento;
- Verifica iniziale dei documenti tramite la tecnica di analisi statica **walkthrough**;
- Verifica dei documenti tramite tecnica di analisi statica di tipo **inspection**. In questo caso l'attenzione del verificatore si focalizza solo su particolari aspetti del testo (per esempio accenti, sillabazione, maiuscole, ecc);
- Monitoraggio da parte dei verificatori del registro delle modifiche in modo da poter attuare la verifica solo alla parte inerente alla sezione modificata;
- **Coerenza** di informazioni tra i diversi documenti;
- Verifica di assenza di **ridondanza** di determinati contenuti tra i diversi documenti;
- Calcolo dell'indice di leggibilità **Gulpease**.

3.8.2 Metriche e misure

3.8.3 Indice Gulpease

Indice di leggibilità di un testo tarato sulla lingua italiana.

Questo indice considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere. La formula per il suo calcolo è la seguente:

$$89 - \frac{(300 * \text{NumeroFrase}) - (10 * \text{NumeroLettere})}{\text{NumeroParole}}$$

I risultati sono compresi tra 0 e 100, dove il valore 100 indica la leggibilità più alta e 0 la leggibilità più bassa.

In generale risulta che testi con un indice:

- inferiore a 80 sono difficili da leggere per chi ha la licenza elementare;
- inferiore a 60 sono difficili da leggere per chi ha la licenza media;
- inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

- Range di accettazione: [50 – 100]
- Range ottimale: [60 – 100]

4 Misure e metriche: riepilogo parametri di tolleranza

Di seguito vengono elencate le metriche adottate per il controllo qualitativo dei processi, dei documenti e del prodotto software in sviluppo.

4.1 Processi

Metriche utilizzate per il controllo dell'andamento dei processi di supporto allo sviluppo:

Metrica	Accettazione	Ottimale
Schedule Variance	$\geq -(P * 5\%)$	≥ 0
Budget Variance	$\geq -(P * 10\%)$	≥ 0

$P = PreventivoFase$

Schedule variance e *budget variance* danno un resoconto in termini di tempo e di costi effettuati confrontati con il preventivo.

I margini di tolleranza di errore sono stati impostati al 5% sulle scadenze prefissate e al 10% sul budget.

4.2 Documenti

La metrica che si è scelto di utilizzare per un'analisi il più possibile oggettiva sul contenuto dei documenti è l'indice Gullpease (vedere appendice X per una descrizione più completa).

Metrica	Accettazione	Ottimale
Gulpease Index	[50 – 100]	[60 – 100]

4.3 Software

Per l'analisi statica del codice prodotto viene utilizzato il software di analisi Complexity-report, per una descrizione dettagliata delle metriche utilizzate dal software vedere appendice B sezione 3.5.

Metrica	Accettazione	Ottimale
Parameters count	[0 – 7]	[0 – 5]
Dependency count	≤ 5	$= 0$
Halstead's Volume	[20 – 1500]	[20 – 1000]
Halstead's Difficulty	[0 – 30]	[0 – 15]
Cyclomatic complexity	[0 – 15]	[0 – 10]
First-order density	$\leq 20\%$	$\leq 15\%$
Change cost	$\leq 50\%$	$\leq 40\%$
Core size	$\leq 30\%$	$\leq 25\%$
Maintainability Index	[20 – 100]	[70 – 100]
Copertura dei test	[70 – 100]	[80 – 100]

La metrica fondamentale di questo software è il **Maintainability Index** che offre un valore sul livello di manutenibilità del codice.

I valori di tolleranza fissati per tale indice sono:

- $MI \geq 20$: alta manutenibilità
- $10 \leq MI \leq 20$: moderata manutenibilità
- $MI \leq 10$: bassa manutenibilità

A Specifica test

La suddivisione temporale delle diverse fasi di sviluppo segue il cosiddetto modello a V ed è descritta nel documento allegato *Piano-di-Progetto-2.0.pdf*. Questo modello prevede un

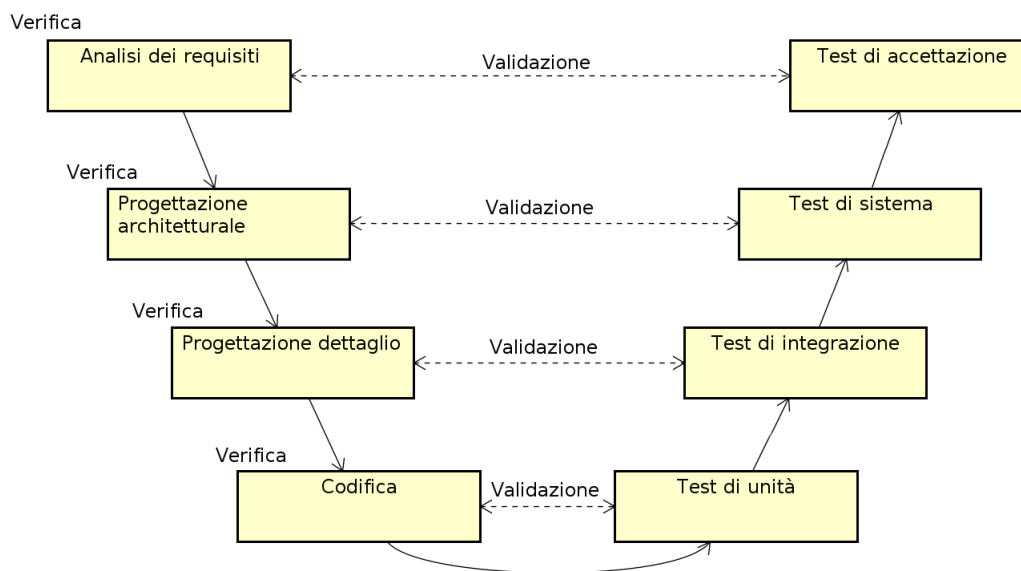


Figura 3: Fasi di sviluppo

lavoro di controllo e verifica corrispondente ad ogni attività per evitare l'accumulo di errori difficilmente gestibili in seguito.

La divisione in diverse fasi, e l'attività di verifica abbinata ad ogni fase e processo, hanno come scopo quello di facilitare l'integrazione e il corretto funzionamento delle parti che comporranno il sistema finale. Il ramo discendente descritto nella precedente figura rappresenta la successione delle fasi di sviluppo: ciascuna fase è accompagnata da una costante attività di verifica in modo da poter permettere il passaggio alla fase successiva esclusivamente quando si è sicuri che non ci siano errori. Ciò è essenziale durante la delicata attività di raccolta e documentazione dei requisiti, nella progettazione architetturale ad alto livello ed in seguito nella progettazione di dettaglio.

Del ramo ascendente che attraversa le attività di testing è importante sottolineare l'utilizzo dell'approccio bottom-up. Dalla fase di codifica quindi, per mezzo di test specifici sulle singole componenti (test di unità), si è in grado di garantire la correttezza del lavoro svolto prima che le singole componenti vengano integrate tra loro. Ovviamente l'esito positivo di ogni test su ogni unità non garantisce la correttezza dell'integrazione tra le stesse.

A.1 Livelli di testing

I test da pianificare ed effettuare, come spiegato nel documento *Norme-di-progetto-2.0.pdf*, sono divisi in quattro categorie:

- **Test di Accettazione (TA)**
- **Test di Sistema (TS)**
- **Test di Integrazione (TI)**

- Test di Unità (TU)

A.2 Test di accettazione

Tabella A.1: Test di accettazione

ID requisito/i	ID test	Descrizione	Stato
F 1 F 1.1 F 1.1.1 F 1.1.2 F1.1.2.1	TA 1	Viene verificato che il sistema fornisca la possibilità ad un utente non autenticato di registrarsi compilando il form di registrazione. L'utente deve poter: <ol style="list-style-type: none"> 1. inserire la propria mail 2. inserire la password 3. ricevere un messaggio di errore se la stringa password ha meno di 8 caratteri 4. inviare i dati di registrazione 	Pianificato
F 1.2 F 1.2.1 F 1.2.1.1 F 1.2.1.2 F 1.2.1.3	TA 1.1	La sezione dedicata alla registrazione deve permettere l'invio dei dati inseriti. L'utente deve poter ricevere un messaggio di errore se: <ol style="list-style-type: none"> 1. i dati non sono stati inseriti 2. la mail inserita risulta già in uso Se i dati non sono corretti il sistema deve bloccare l'invio.	Pianificato
F 1.2.2 F 1.2.3	TA 1.2	Quando i dati corretti vengono inviati viene verificato che: <ol style="list-style-type: none"> 1. il sistema memorizzi un nuovo account con i dati inseriti 2. venga effettuato automaticamente il login al completamento della registrazione 	Pianificato
F 2 F 2.1 F 2.1.1 F2.1.2	TA 2	Verifica che il sistema fornisca la possibilità ad un utente non autenticato di autenticarsi. All'utente dev'essere permesso di: <ol style="list-style-type: none"> 1. inserire la propria mail 2. inserire la propria password 3. 	Pianificato

Tabella A.1: Requisiti funzionali

ID requisito	ID test	Descrizione	Stato
F 2.2 F 2.2.1 F 2.2.2 F 2.2.2.1 F 2.2.2.2 F 2.2.3	TA 2.1	<p>Verifica che la sezione dedicata all'autenticazione fornisca una procedura di recupero password. il sistema deve:</p> <ol style="list-style-type: none"> 1. permettere l'inserimento della mail 2. controllare che la mail risulti registrata e in caso negativo visualizzare un messaggio di errore 3. bloccare l'invio dei dati se la mail non risulta registrata 4. mandare una mail all'indirizzo inserito con la password dimenticata 	Pianificato
F 2.3 F 2.3.1 F 2.3.1.1 F 2.3.2	TA 2.2	<p>Verifica che la sezione dedicata all'autenticazione permetta l'invio dei dati inseriti. Il sistema deve:</p> <ol style="list-style-type: none"> 1. effettuare un controllo di correttezza sui dati inseriti 2. visualizzare un messaggio di errore se la combinazione e-mail/password non è registrata 3. autenticare l'utente se la combinazione mail/password è corretta 	Pianificato
F 3 F 3.1 F 3.1.1 F 3.1.2 F 3.1.3	TA 3	<p>Verifica che il sistema permetta la navigazione e selezione dei questionari per categoria. Il sistema deve:</p> <ol style="list-style-type: none"> 1. fornire una sezione con un elenco di categorie 2. fornire l'elenco dei questionari presenti per la categoria selezionata 3. fornire il numero di questionari della categoria 	Pianificato

Tabella A.1: Requisiti funzionali

ID requisito	ID test	Descrizione	Stato
F 3.2 F 3.2.1 F 3.2.2 F 3.2.3 F 3.2.4	TA 3.1	<p>Verifica che il sistema fornisca una sezione con un elenco dei questionari relativi ad una categoria. Viene verificato che:</p> <ol style="list-style-type: none"> 1. quando l'utente seleziona un questionario venga visualizzata la sezione della compilazione del questionario scelto 2. per ogni elemento dell'elenco venga visualizzato il nome del questionario 3. per ogni elemento dell'elenco venga visualizzato il numero di domande del questionario 4. per ogni elemento dell'elenco venga visualizzato l'autore 	Pianificato
F 4 F 4.1 F 4.2 F 4.2.1 F 4.2.2 F 4.2.3 F 4.2.4 F 4.2.4.1 F 4.2.5 F 4.2.5.1 F 4.2.6 F 4.2.6.1 F 4.3 F 4.4	TA 4	<p>Verifica che il sistema dia la possibilità di compilare un questionario. L'utente deve poter:</p> <ol style="list-style-type: none"> 1. navigare liberamente tra le domande del questionario 2. rispondere alle domande 3. visualizzare la risposta data 4. selezionare possibili risposte in caso di risposte multiple 5. associare elementi in caso di risposte di tipo associativo 6. selezionare le risposte in caso di domanda di tipo completamento testo 7. cambiare le risposte già date 8. rispondere alle domande in qualsiasi ordine 9. consegnare il questionario completato 	Pianificato

Tabella A.1: Requisiti funzionali

ID requisito	ID test	Descrizione	Stato
F 4.5.1 F 4.5.2 F 4.5.2.1 F 4.5.2.1.1 F 4.5.2.1.2 F 4.5.2.1.3	TA 4.1	<p>Verifica che alla consegna del questionario il sistema:</p> <ol style="list-style-type: none"> 1. avvisi l'utente se alcune risposte non sono state compilate 2. valuti il questionario e ne visualizzi il risultato 3. visualizzi il punteggio ottenuto 4. visualizzi la percentuale di risposte corrette 	Pianificato
F 4.6 F 4.6.1 F 4.6.2 F 4.7	TA 4.2	<p>Verifica che la compilazione del questionario abbia un tempo massimo. Il sistema deve:</p> <ol style="list-style-type: none"> 1. consegnare automaticamente il questionario allo scadere del tempo massimo anche se incompleto 2. visualizzare il tempo restante 	Pianificato
F 5 F 5.1 F 5.1.1 F 5.1.2 F 5.2	TA 5	<p>Verifica che il sistema dia la possibilità ad un utente autenticato di creare una nuova domanda. L'utente deve potere:</p> <ol style="list-style-type: none"> 1. scegliere la categoria della domanda 2. inserire i dati necessari alla compilazione della domanda in formato QML 3. inviare i dati inseriti 	Pianificato

Tabella A.1: Requisiti funzionali

ID requisito	ID test	Descrizione	Stato
F 5.2.1 F 5.2.1.1 F 5.2.1.2 F 5.2.1.3 F 5.2.2 F 5.2.2.1 F 5.2.2.2	TA 5.1	<p>Verifica che il sistema effettui un controllo sulla correttezza dei dati prima della memorizzazione. Il sistema deve:</p> <ol style="list-style-type: none"> 1. visualizzare un messaggio di errore se i dati obbligatori non risultano inseriti 2. visualizzare un messaggio di errore se il codice QML inserito non risulta valido 3. visualizzare un messaggio nel caso che i dati siano corretti e la memorizzazione della domanda è andata a buon fine. In caso contrario deve annullare la creazione della domanda ed avvisare l'utente tramite un messaggio a schermo 	Pianificato
F 6 F 6.1 F 6.1.1 F 6.1.2 F 6.1.3 F 6.1.3.1 F 6.1.3.2 F 6.2	TA 6	<p>Verifica che il sistema fornisca la possibilità ad un utente autenticato di creare un nuovo questionario.</p> <p>L'utente deve poter:</p> <ol style="list-style-type: none"> 1. compilare il nome del questionario 2. inserire la categoria 3. visualizzare la lista delle domande della categoria 4. selezionare qualsiasi domanda visualizzata 5. inviare il nuovo questionario 	Pianificato
F 6.2.1 F 6.2.1.1 F 6.2.1.2 F 6.2.1.3 F 6.2.2 F 6.2.2.1 F 6.2.2.2	TA 6.1	<p>Verifica che il sistema, prima di inviare i dati, effettui un controllo su di essi. Nello specifico si verifica che il sistema:</p> <ol style="list-style-type: none"> 1. avvisi con un messaggio che i campi obbligatori non sono stati compilati o non è stata selezionata nessuna domanda 2. crei il nuovo questionario se i dati inseriti sono corretti, avvisando l'utente del successo 3. avvisi l'utente con un messaggio se la creazione del nuovo questionario non è andata a buon fine 	Pianificato

Tabella A.1: Requisiti funzionali

A.3 Test di sistema

Nella seguente tabella vengono elencati i test di sistema. Sulla quarta colonna viene specificato il tipo di test, se dinamico o statico.

A.3.1 Requisiti funzionali

Tabella A.2: Test di sistema req. funzionali

ID req.	ID test	Da verificare	Tipo	Stato
F 1	TS 1	Il sistema fornisce la possibilità ad un utente non autenticato di registrarsi	Dinamico	Pianificato
F 1.1	TS 1.1	La sezione dedicata alla registrazione permette l'inserimento dei dati necessari	Dinamico	Pianificato
F 1.1.1	TS 1.1.1	La sezione dedicata alla registrazione deve permettere l'inserimento di una e-mail	Dinamico	Pianificato
F 1.1.2	TA 1.1.2	La sezione dedicata alla registrazione deve permettere l'inserimento di una password	Dinamico	Pianificato
F 1.1.2.1	TS 1.1.2.1	Se la password inserita ha meno di 8 caratteri viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 1.2	TS 1.2	La sezione dedicata alla registrazione deve permettere l'invio dei dati inseriti	Dinamico	Pianificato
F 1.2.1	TS 1.2.1	Prima di inviare i dati viene effettuato un controllo su di essi	Dinamico	Pianificato
F 1.2.1.1	TS 1.2.1.1	Se i dati richiesti non sono stati inseriti viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 1.2.1.2	TS 1.2.1.2	Se la email inserita risulta già in uso viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 1.2.1.3	TS 1.2.1.3	Se i dati non risultano corretti ne viene bloccato l'invio	Dinamico	Pianificato
F 1.2.2	TS 1.2.2	Quando dei dati corretti vengono inviati il sistema memorizza un nuovo account con essi	Dinamico	Pianificato
F 1.2.3	TS 1.2.3	Quando viene creato un nuovo account con la procedura di registrazione viene automaticamente effettuato il login con quell'account	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 2	TS 2	Il sistema deve fornire la possibilità ad un utente non autenticato di autenticarsi	Dinamico	Pianificato
F 2.1	TS 2.1	La sezione dedicata all'autenticazione deve permettere l'inserimento dei dati necessari	Dinamico	Pianificato
F 2.1.1	TS 2.1.1	La sezione dedicata all'autenticazione deve permettere l'inserimento di una e-mail	Dinamico	Pianificato
F 2.1.2	TS 2.1.2	La sezione dedicata all'autenticazione deve permettere l'inserimento di una password	Dinamico	Pianificato
F 2.2	TS 2.2	La sezione dedicata all'autenticazione deve fornire una procedura di recupero password	Dinamico	Pianificato
F 2.2.1	TS 2.2.1	La procedura di recupero password deve permettere l'inserimento della e-mail	Dinamico	Pianificato
F 2.2.2	TS 2.2.2	La procedura di recupero password deve effettuare un controllo sulla e-mail inserita	Dinamico	Pianificato
F 2.2.2.1	TS 2.2.2.1	Se la e-mail non risulta registrata viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 2.2.2.2	TS 2.2.2.2	Se la e-mail non risulta registrata viene bloccato l'invio della mail a quell'indirizzo	Dinamico	Pianificato
F 2.2.3	TS 2.2.3	La procedura di recupero password deve mandare una mail all'indirizzo inserito con la password dimenticata	Dinamico	Pianificato
F 2.3	TS 2.3	La sezione dedicata all'autenticazione deve permettere l'invio dei dati inseriti	Dinamico	Pianificato
F 2.3.1	TS 2.3.1	Prima di inviare i dati viene effettuato un controllo su di essi	Dinamico	Pianificato
F 2.3.1.1	TS 2.3.1.1	Se la combinazione e-mail/password non è registrata viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 2.3.2	TS 2.3.2	Se la combinazione e-mail/password è registrata l'utente viene autenticato	Dinamico	Pianificato
F 3	TS 3	Il sistema deve permettere la navigazione dei questionari per categoria	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 3.1	TS 3.1	Il sistema deve fornire una sezione con un elenco delle categorie di questionari presenti nel database	Dinamico	Pianificato
F 3.1.1	TS 3.1.1	Quando l'utente seleziona una categoria viene visualizzata la sezione che elenca i questionari della categoria scelta	Dinamico	Pianificato
F 3.1.2	TS 3.1.2	Per ogni elemento dell'elenco viene visualizzato il nome della categoria	Dinamico	Pianificato
F 3.1.3	TS 3.1.3	Per ogni elemento dell'elenco viene visualizzato il numero di questionari della categoria	Dinamico	Pianificato
F 3.2	TS 3.2	Il sistema deve fornire una sezione con un elenco dei questionari relativi ad una categoria	Dinamico	Pianificato
F 3.2.1	TS 3.2.1	Quando l'utente seleziona un questionario viene visualizzata la sezione della compilazione del questionario scelto	Dinamico	Pianificato
F 3.2.2	TS 3.2.2	Per ogni elemento dell'elenco viene visualizzato il nome del questionario	Dinamico	Pianificato
F 3.2.3	TS 3.2.3	Per ogni elemento dell'elenco viene visualizzato il numero di domande del questionario	Dinamico	Pianificato
F 3.2.4	TS 3.2.4	Per ogni elemento dell'elenco viene visualizzato l'autore	Dinamico	Pianificato
F 4	TS 4	Il sistema deve fornire la possibilità di compilare un questionario	Dinamico	Pianificato
F 4.1	TS 4.1	Durante la compilazione del questionario un utente può navigare liberamente tra le domande che lo compongono	Dinamico	Pianificato
F 4.2	TS 4.2	L'utente può rispondere alle domande	Dinamico	Pianificato
F 4.2.1	TS 4.2.1	Viene visualizzato il testo della domanda	Dinamico	Pianificato
F 4.2.2	TS 4.2.2	Viene visualizzata la risposta data	Dinamico	Pianificato
F 4.2.3	TS 4.2.3	Se la domanda richiede una risposta inserita dall'utente allora viene fornito un campo per l'inserimento della risposta	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 4.2.4	TS 4.2.4	Se la domanda ha delle risposte pre-impostate di qualsiasi tipo queste vengono visualizzate	Dinamico	Pianificato
F 4.2.4.1	TS 4.2.4.1	Deve essere possibile selezionare una risposta	Dinamico	Pianificato
F 4.2.5	TS 4.2.5	Se la domanda permette associazioni tra elementi deve visualizzare tutti gli elementi associabili	Dinamico	Pianificato
F 4.2.5.1	TS 4.2.5.1	Deve essere possibile associare due elementi di insiemi diversi	Dinamico	Pianificato
F 4.2.6	TS 4.2.6	Se la domanda prevede il completamento del testo con delle parole vengono visualizzate le possibili parole	Dinamico	Pianificato
F 4.2.6.1	TS 4.2.6.1	Deve essere possibile selezionare una parola dalla lista	Dinamico	Pianificato
F 4.3	TS 4.3	L'utente può cambiare la risposta di una domanda in qualsiasi momento prima della consegna del questionario	Dinamico	Pianificato
F 4.4	TS 4.4	Le domande possono essere compilate in un ordine qualsiasi	Dinamico	Pianificato
F 4.5	TS 4.5	L'utente può consegnare il questionario	Dinamico	Pianificato
F 4.5.1	TS 4.5.1	Se alla consegna una o più domande non sono state compilate il sistema avverte l'utente di ciò	Dinamico	Pianificato
F 4.5.2	TS 4.5.2	Quando un questionario viene confermato e consegnato viene valutato dal sistema	Dinamico	Pianificato
F 4.5.2.1	TS 4.5.2.1	La valutazione del questionario viene visualizzata all'utente	Dinamico	Pianificato
F 4.5.2.1.1	TS 4.5.2.1.1	Viene visualizzato il nome del questionario	Dinamico	Pianificato
F 4.5.2.1.2	TS 4.5.2.1.2	Viene visualizzato il punteggio ottenuto	Dinamico	Pianificato
F 4.5.2.1.3	TS 4.5.2.1.3	Viene visualizzata la percentuale delle risposte corrette	Dinamico	Pianificato
F 4.6	TS 4.6	La compilazione del questionario ha un tempo massimo	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 4.6.1	TS 4.6.1	Allo scadere del tempo il questionario verrà consegnato automaticamente anche se incompleto	Dinamico	Pianificato
F 4.6.2	TS 4.6.2	Il tempo inizia a scorrere da quando l'utente accede alla pagina	Dinamico	Pianificato
F 4.7	TS 4.7	Viene visualizzato il tempo restante	Dinamico	Pianificato
F 5	TS 5	Il sistema deve fornire la possibilità ad un utente autenticato di creare una nuova domanda	Dinamico	Pianificato
F 5.1	TS 5.1	La sezione dedicata alla creazione di una domanda deve permettere l'inserimento dei dati necessari	Dinamico	Pianificato
F 5.1.1	TS 5.1.1	La sezione dedicata alla creazione di una domanda deve permettere la scelta di una o più categorie	Dinamico	Pianificato
F 5.1.2	TS 5.1.2	La sezione dedicata alla creazione di una domanda deve permettere l'inserimento del codice QML che definisce la domanda	Dinamico	Pianificato
F 5.2	TS 5.2	La sezione dedicata alla creazione di una domanda deve permettere l'invio dei dati inseriti	Dinamico	Pianificato
F 5.2.1	TS 5.2.1	Prima di inviare i dati viene effettuato un controllo su di essi	Dinamico	Pianificato
F 5.2.1.1	TS 5.2.1.1	Se i dati obbligatori non risultano inseriti viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 5.2.1.2	TS 5.2.1.2	Se il codice QML inserito non risulta valido viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 5.2.1.3	TS 5.2.1.3	Se i dati obbligatori non risultano inseriti o il codice QML non è valido la creazione della nuova domanda viene bloccata	Dinamico	Pianificato
F 5.2.2	TS 5.2.2	Quando dei dati che superano i controlli vengono inviati viene creata una nuova domanda con essi e inserita nel database	Dinamico	Pianificato
F 5.2.2.1	TS 5.2.2.1	Se la creazione di una nuova domanda va a buon fine il sistema avvisa l'utente	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 5.2.2.2	TS 5.2.2.2	Se la creazione di una nuova domanda non va a buon fine il sistema visualizza un messaggio di errore	Dinamico	Pianificato
F 6	TS 6	Il sistema deve fornire la possibilità ad un utente autenticato di creare un nuovo questionario	Dinamico	Pianificato
F 6.1	TS 6.1	La sezione dedicata alla creazione di un questionario deve permettere l'inserimento dei dati necessari	Dinamico	Pianificato
F 6.1.1	TS 6.1.1	La sezione dedicata alla creazione di un questionario deve permettere l'inserimento del nome del questionario	Dinamico	Pianificato
F 6.1.2	TS 6.1.2	La sezione dedicata alla creazione di un questionario deve permettere l'inserimento della categoria del questionario	Dinamico	Pianificato
F 6.1.3	TS 6.1.3	La sezione dedicata alla creazione di un questionario deve visualizzare la lista delle domande della categoria selezionata	Dinamico	Pianificato
F 6.1.3.1	TS 6.1.3.1	Se nessuna categoria è stata selezionata non viene visualizzata alcuna domanda	Dinamico	Pianificato
F 6.1.3.2	TS 6.1.3.2	Ogni domanda deve essere selezionabile	Dinamico	Pianificato
F 6.2	TS 6.2	La sezione dedicata alla creazione di un nuovo questionario deve permettere l'invio dei dati inseriti	Dinamico	Pianificato
F 6.2.1	TS 6.2.1	Prima di inviare i dati viene effettuato un controllo su di essi	Dinamico	Pianificato
F 6.2.1.1	TS 6.2.1.1	Se i dati obbligatori non risultano inseriti viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 6.2.1.2	TS 6.2.1.2	Se nessuna domanda è selezionata viene visualizzato un messaggio di errore	Dinamico	Pianificato
F 6.2.1.3	TS 6.2.1.3	Se i dati obbligatori non risultano inseriti non è valido nessun dato viene inviato	Dinamico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 6.2.2	TS 6.2.2	Quando dei dati che superano i controlli vengono inviati viene creato un nuovo questionario con essi e inserito nel database	Dinamico	Pianificato
F 6.2.2.1	TS 6.2.2.1	Se la creazione di un nuovo questionario va a buon fine il sistema avvisa l'utente	Dinamico	Pianificato
F 6.2.2.2	TS 6.2.2.2	Se la creazione di un nuovo questionario non va a buon fine il sistema visualizza un messaggio di errore	Dinamico	Pianificato
F 7	TS 7	Il sistema deve fornire la possibilità ad un utente autenticato di uscire facendo il logout	Dinamico	Pianificato
F 8	TS 8	Il sistema deve essere in grado di interpretare il codice QML	Dinamico	Pianificato
F 8.1	TS 8.1	QML definisce domande	Statico	Pianificato
F 8.1.1	TS 8.1.1	Una domanda è formata dal testo del quesito e da una o più risposte	Statico	Pianificato
F 8.1.2	TS 8.1.2	Una domanda deve avere almeno una risposta esatta	Statico	Pianificato
F 8.1.3	TS 8.1.3	Una domanda può avere una lista di possibili risposte	Statico	Pianificato
F 8.1.4	TS 8.1.4	QML permette la definizione di diverse tipologie di domande	Statico	Pianificato
F 8.1.4.1	TS 8.1.4.1	QML definisce domande di tipo vero o falso	Statico	Pianificato
F 8.1.4.2	TS	QML definisce domande a scelta multipla con una sola risposta esatta	Statico	Pianificato
F 8.1.4.3	TS 8.1.4.3	QML definisce domande a scelta multipla con più risposte esatte	Statico	Pianificato
F 8.1.4.4	TS 8.1.4.4	QML definisce domande a risposta testuale	Statico	Pianificato
F 8.1.4.5	TS 8.1.4.5	QML definisce domande di associazione tra elementi di due insiemi	Statico	Pianificato
F 8.1.4.5.1	TS 8.1.4.5.1	Gli insiemi possono contenere elementi non associabili di disturbo	Statico	Pianificato
F 8.1.4.6	TS 8.1.4.6	QML definisce domande a completamento testuale a scelta da un gruppo di parole date	Statico	Pianificato

Tabella A.2: Requisiti funzionali

ID req.	ID test	Descrizione	Tipo	Stato
F 8.1.4.6.1	TS 8.1.4.6.1	Il gruppo di parole può contenere elementi di disturbo	Statico	Pianificato
F 8.1.4.7	TS 8.1.4.7	QML definisce domande di associazione tra date poste in una timeline e avvenimenti	Statico	Pianificato
F 8.2	TS 8.2	QML gestisce immagini all'interno di domande e risposte preimpostate	Statico	Pianificato

Tabella A.2: Requisiti funzionali

A.3.2 Requisiti di qualità

Tabella A.3: Test di sistema req. qualità

ID req.	ID test	Da verificare	Tipo	Stato
Q 1	TQ 1	Ogni file contiene codice scritto in un solo linguaggio per separare struttura, contenuto e comportamento	Statico	Pianificato
Q 1.1	TQ 1.1	Il codice html si occupa della struttura delle pagine web	Statico	Pianificato
Q 1.2	TQ 1.2	Il codice css si occupa dello stile delle pagine web	Statico	Pianificato
Q 1.3	TQ 1.3	Il codice javascript si occupa del comportamento delle pagine web	Statico	Pianificato
Q 2	TQ 2	Il codice HTML passa il test di validazione del W3C	Dinamico	Pianificato
Q 3	TQ 3	Il codice CSS passa il test di validazione del W3C	Dinamico	Pianificato

Tabella A.3: Requisiti di qualità

A.4 Test di integrazione

Tabella A.4: Test di integrazione

ID Test	Descrizione	Stato
T11	Viene verificato che Model::Database gestisca correttamente le interazioni con la ViewModel in entrata per le interazioni sugli utenti e sulla modifica o creazione di nuovi quiz e questionari, ed in uscita per la notifica alla ViewModel di cambiamenti dei dati nel database.	Pianificato
T12	Viene verificato che Model::Parser interagisca correttamente con Model::Database nel controllo della correttezza del codice QML e per la visualizzazione delle domande.	Pianificato
T13	Viene verificato che Model::Statistics interagisca correttamente con Model::Database e ViewModel per la memorizzazione delle statistiche e la loro visualizzazione.	Pianificato
T14	Viene verificato che Model::Publishers ...	Pianificato
T15	Viene verificato che View::Pages interagisca correttamente con View::Templates per la composizione delle pagine web da visualizzare	Pianificato
T16	Viene verificato che View::Templates interagisca correttamente con View::Pages	Pianificato
T17	Viene verificato che ViewModel::Controllers interagisca correttamente con i package View, ViewModel::Interpreter e ViewModel::Router per il data-binding; e con i package ViewModel::Subscribers, ViewModel::Methods che interagiscono con Model per richiedere ed aggiornare i dati	Pianificato
T18	Viene verificato che ViewModel::Subscribers interagisca correttamente con Model e ViewModel::Controllers	Pianificato
T19	Viene verificato che ViewModel::Methods interagisca correttamente con Model e ViewModel::Controllers	Pianificato
T110	Viene Verificato che ViewModel::Interpreter interagisca correttamente con ViewModel::Controllers per la traduzione di testo QML in codice HTML5 da visualizzare	Pianificato
T111	Viene verificato che ViewModel::Router interagisca correttamente con ViewModel::Controllers	Pianificato

Tabella A.4: Test di integrazione

A.5 Test di unità

Tabella A.5: Test di unità

ID Test	Descrizione	Stato
TU1	Viene verificato che	Pianificato
TU2	Viene verificato	Pianificato
TU3	Viene verificato	Pianificato
TU4	Viene verificato	Pianificato
TU5	Viene verificato	Pianificato
TU6	Viene verificato	Pianificato
TU7	Viene verificato	Pianificato
TU8	Viene verificato	Pianificato
TU9	Viene verificato	Pianificato
TU10	Viene verificato	Pianificato
TU11	Viene verificato	Pianificato
TU12	Viene verificato	Pianificato
TU13	Viene verificato	Pianificato
TU14	Viene verificato	Pianificato
TU15	Viene verificato	Pianificato
TU16	Viene verificato	Pianificato
TU17	Viene verificato	Pianificato
TU18	Viene verificato	Pianificato
TU19	Viene verificato	Pianificato
TU20	Viene verificato	Pianificato
TU21	Viene verificato	Pianificato
TU22	Viene verificato	Pianificato
TU23	Viene verificato	Pianificato

Tabella A.5: Test di unità

B Tracciamento componenti – requisiti

Tabella A.6: Tracciamento Componenti - Test di integrazione

Componente	ID Test
Model::Database	TI1
Model::Parser	TI2

Tabella A.6: Tracciamento Componenti - Test di integrazione

Componente	ID Test
Model::Statistics	TI3
Model::Publishers	TI4
View::Pages	TI5
View::Templates	TI6
ViewModel::Controllers	TI7
ViewModel::Subscribers	TI8
ViewModel::Methods	TI9
ViewModel::Interpreter	TI10
ViewModel::Router	TI11

Tabella A.6: Tracciamento Componenti - Test di integrazione

C Tracciamento requisiti – componenti

D Tracciamento metodi – test di unità

Tabella A.7: Tracciamento Componenti - Test di integrazione

Metodo	ID Test
Model::Database::QuizManager::addQuiz() Model::Database::QuizManager::modifyQuiz() Model::Database::QuizManager::removeQuiz()	TU1
Model::Database::QuestionManager::addQuestion() Model::Database::QuestionManager::modifyQuestion() Model::Database::QuestionManager::removeQuestion()	TU2
Model::Parser::Parser::check()	TU3
Model::Statistics::Statistics::updateQuestionStats() Model::Statistics::Statistics::updateQuizStats() Model::Statistics::Statistics::updateUserStatitics()	TU4
Model::Publishers::UserPublishers::publishUser()	TU5
Model::Publishers::QuizPublishers::publishQuiz() Model::Publishers::QuestionPublishers::publishQuestion()	TU6
ViewModel::Controllers::NewQuestionController::NewQuestionController() ViewModel::Controllers::NewQuestionController::saveQuestion()	TU7
ViewModel::Controllers::NewQuiz Controller::newQuiz Controller() ViewModel::Controllers::NewQuiz Controller::saveQuiz ()	TU8
ViewModel::Controllers::QuizListController::quizListController() ViewModel::Controllers::QuizListController::quizzes()	TU9

Tabella A.7: Tracciamento Componenti - Test di integrazione

Metodo	ID Test
ViewModel::Controllers::QuizListController::quizDetails() ViewModel::Controllers::QuizListController::orderBy()	
ViewModel::Controllers::QuizDetailsController::quizDetailsController()	TU10
ViewModel::Controllers::DeleteQuestionController::DeleteQuestionController() ViewModel::Controllers::DeleteQuestionController::deleteQuestion()	TU11
ViewModel::Controllers::DeleteQuizController::DeleteQuizController() ViewModel::Controllers::DeleteQuizController::deleteQuiz()	TU12
ViewModel::Controllers::QuizManagementController::QuizManagementController() ViewModel::Controllers::QuizManagementController::nextQuestion() ViewModel::Controllers::QuizManagementController::previousQuestion() ViewModel::Controllers::QuizManagementController::endQuiz()	TU13
ViewModel::Controllers::QuestionsManagementController::QuestionsManagementController() ViewModel::Controllers::QuestionsManagementController::getQuestion()	TU14
ViewModel::Controllers::QMLEditorController::QMLEditorController() ViewModel::Controllers::QMLEditorController::check()	TU 15
ViewModel::Subscribers::QuestionsSubscriber::getCollection() ViewModel::Subscribers::QuestionsSubscriber::subscribe()	TU16
ViewModel::Subscribers::QuizSubscriber::getCollection() ViewModel::Subscribers::QuizSubscriber::subscribe()	TU17
ViewModel::Subscribers::UsersSubscriber::getCollection() ViewModel::Subscribers::UsersSubscriber::subscribe()	TU18
ViewModel::Methods::QuestionMethods::insert() ViewModel::Methods::QuestionMethods::remove()	TU19
ViewModel::Methods::QuizMethods::insert() ViewModel::Methods::QuizMethods::remove()	TU20
ViewModel::Methods::UserMethods::insert() ViewModel::Methods::UserMethods::remove()	TU21
NON SO SE VA MESSO ViewModel::Interpreter::Interpreter::translate()	TU22
ViewModel::Interpreter::InterpreterFactory::createInterpreter()	TU23
ViewModel::Interpreter::QMLInterpreterFactory::QMLInterpreterFactory() ViewModel::Interpreter::QMLInterpreterFactory::getInstance()	TU24
ViewModel::Interpreter::QMLInterpreter::translate()	TU25
ViewModel::Interpreter::QML2HTMLInterpreterFactory::translate()	TU26
ViewModel::Router::Router::state()	TU27
View::Pages::Page::show()	TU28
View::Pages::LoginPage::show()	TU29
View::Pages::RegistrationPage::show()	TU30

Tabella A.7: Tracciamento Componenti - Test di integrazione

Metodo	ID Test
View::Pages::PasswordRecoveryPage::show()	TU30
View::Pages::QuizCreationPage::show()	TU31
View::Pages::QuestionUpdatePage::show()	TU32
View::Pages::QuestionCreationPage::show()	TU33
View::Pages::QuestionManagementPage::show()	TU34
View::Pages::QuestionManagementPage::sort()	
View::Pages::QuestionManagementPage::search()	
View::Pages::QuizResultsPage::show()	TU35
View::Pages::QuizExecutionPage::show()	TU36
View::Pages::QuizExecutionPage::nextQuestion()	
View::Pages::QuizExecutionPage::previousQuestion()	
View::Pages::QuizListPage::show()	TU37
View::Pages::QuizListPage::sort()	
View::Pages::QuizListPage::search()	
View::Pages::CategoryListPage::show()	TU38
View::Pages::CategoryListPage::sort()	
View::Pages::CategoryListPage::search()	

Tabella A.7: Tracciamento Componenti - Test di integrazione

E Esito misurazioni

Di seguito viene fornito il resoconto dell'attività di verifica inerente ad ogni fase di sviluppo.

E.1 Analisi

E.1.1 Processi

Di seguito vengono riportati i valori degli indici di Schedule e Budget Variance per la validazione dei processi misurati. I valori sono risultati da colcoli sul consuntivo presentato nel documento piano-di-progetto-2.0.pdf.

Metrica	Valore	Accettazione	Ottimale
Schedule Variance	2 ore	Superato	Superato
Budget Variance	95 euro	Superato	Superato

E.1.2 Documenti

Metrica	Gulpease	Accettazione	Ottimale
Norme di Progetto	53	Superato	Superato
Piano di Progetto	50	Superato	Superato
Piano di Qualifica	43	non superato	non superato
Analisi dei Requisiti	58	Superato	Superato
Studio di Fattibilità	60	Superato	Superato

E.2 Progettazione Architettuale

Metrica	Valore	Accettazione	Ottimale
Schedule Variance	41 ore	Superato	Superato
Budget Variance	764 euro	Superato	Superato