QUIZZIPEDIA

team404swe@gmail.com

Piano di qualifica 1.0

Informazioni sul documento	
Nome Documento	Piano di qualifica
Versione	1.0
Uso	Esterno
Data Creazione	21 dicembre 2015
Data Ultima Modifica	16 maggio 2016
Redazione	Andrea Multineddu
Verifica	Marco Crivellaro - Luca Alessio
Approvazione	Davide Bortot
Committente	Zucchetti SPA
Lista di distribuzione	Prof. Vardanega Tullio
	TEAM404

Registro delle modifiche

Versione	Autore	Data	Descrizione
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	rivista sezione verifica fasi di sviluppo
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
2.1.1 Qualità di processo . . . 4
2.1.2 Qualità di prodotto . . . 4
2.2 Organizzazione . . . 5
2.2.1 Analisi . . . 5
2.2.2 Progettazione architetturale . . . 5
2.2.3 Progettazione di dettaglio e Codifica . . . 6
2.3 Gestione amministrativa della revisione . . . 7
2.4 Procedure di controllo di qualità di processo . . . 8
2.5 Procedure di controllo qualità di prodotto . . . 8

3 Misure e metriche: parametri di tolleranza **9**
3.1 Processi . . . 9
3.2 Documenti . . . 9
3.3 Software . . . 10

Appendici **11**

A Misure e Metriche **11**
A.1 Di progetto . . . 11
A.1.1 Schedule Variance . . . 11
A.1.2 Budget variance . . . 11
A.2 Per i documenti . . . 11
A.2.1 Indice Gulpease . . . 11
A.3 Per il software . . . 12
A.3.1 Parameter count . . . 12
A.3.2 Cyclomatic complexity . . . 12
A.3.3 Metriche di Halstead . . . 12
A.3.4 Dependency count . . . 13
A.3.5 Maintainability Index . . . 13
A.3.6 First-order density . . . 14
A.3.7 Change cost . . . 14
A.3.8 Core size . . . 14
A.3.9 Copertura dei test . . . 14
A.3.10 W3C - Markup Validation Service . . . 14

B Specifica test **15**
B.1 Livelli di testing 15
B.2 Test di accettazione 16
B.2.1 Requisiti funzionali 16

C Resoconto attività di verifica **21**

Elenco delle figure

1 Riepilogo caratteristiche ISO/IEC 9126:2001 4
2 PDCA e miglioramento complessivo 7
3 Fasi di sviluppo 15

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**. 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 (Javascript$_G$, Node.js$_G$, HTML5$_G$, CSS3$_G$) 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 QML$_G$. 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 HTML$_G$, 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_1.0.pdf* .

1.3 Glossario

Viene allegato un glossario nel file "*glossario_1.0.pdf*" nel quale viene data una definizione a tutti i termini che in questo documento appaiono con il simbolo '$_G$' 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_1.0.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 quindi 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.

2.1.1 Qualità di processo

Per quanto riguarda la valutazione qualitativa dei processi si fa riferimento a due modelli:

- $SPICE_G$: 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.
- $PDCA_G$: metodo di gestione iterativo utilizzato per il controllo e il miglioramento continuo dei processi e dei prodotti.

DA DESCRIVERE IN APPENDICE I DUE MODELLI

2.1.2 Qualità di prodotto

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	Efficenza	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 1: Riepilogo caratteristiche ISO/IEC 9126:2001

DA DESCRIVERE IN APPENDICE

2.2 Organizzazione

Di seguito vengono descritte le attività di verifica da effettuare per ogni fase di sviluppo. Tali fasi seguono la pianificazione definita nel documento allegato *Piano-di-progetto-2.0.pdf*.

2.2.1 Analisi

- **Organizzazione interna**:
  - **Assegnazione dei ruoli** secondo la pianificazione specificata nel documento *Piano-di-Progetto-1.0.pdf* redatto dal Responsabile di progetto;
  - **Corretta pianificazione di sviluppo** secondo quanto stilato sempre nel documento *Piano-di-Progetto-1.0.pdf*. In particolare viene verificata la corretta divisione temporale delle diverse fasi di progetto (vedere sezione precedente) e la correttezza del preventivo in base al budget disponibile;
  - **Disponibilità delle risorse umane e tecnologiche** e dell'ambiente di lavoro;
  - **Lettura e sottoscrizione delle Norme di Progetto**. Il responsabile deve accertarsi che ogni membro del gruppo abbia letto e compreso il documento *Norme-di-progetto-1.0*. Il documento deve essere quindi preventivamente redatto.
- **Documentazione**: il processo di verifica per la documentazione deve seguire le linee guida descritte nella sezione 3 (Misure e metriche: parametri di tolleranza).
- **Requisiti e Use Case**:
  - Correttezza dei diagrammi UML per quanto riguarda l'adeguata rappresentazione grafica degli Use Case. Verifica, leggibilità e adeguato livello di granularità;
  - Verifica della correttezza degli Use Case (testuali e grafici) secondo lo standard UML2.0;
  - Verifica 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.

2.2.2 Progettazione architetturale

In questa fase è necessaria la verifica dei nuovi documenti prodotti (la Specifica tecnica) e la correttezza delle modifiche apportate alla documentazione in seguito alla prima revisione. In particolare deve essere verificato che la progettazione ad alto livello soddisfi i requisiti emersi nella precedente fase e descritti nel documento *Analisi-dei-requisiti-2.0.pdf*.
Di fondamentale importanza è la pianificazione dei test di sistema e la verifica che i requisiti siano soddisfatti. Per la Specifica Tecnica, il documento principale di questa fase, deve essere verificata la correttezza di tutti i diagrammi secondo lo standard UML 2.
I diagrammi presenti devono essere verificati secondo le seguenti linee guida:

- **diagrammi dei package:**
  - correttezza delle relazioni tra i package
  - i package devono essere descritti per struttura e a livello logico
- **diagrammi delle classi:**

  - verifica del 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

- **diagrammi di attività:**
  - verifica della correttezza del flusso di ogni diagramma
  - verifica dell'utilizzo delle entità atomiche opportune

- **diagrammi di sequenza:**
  - verifica della corretta individuazione degli attori

2.2.3 Progettazione di dettaglio e Codifica

Durante questa fase la verifica è volta al controllo del corretto funzionamento dell'applicazione tramite tecniche di analisi statica (walkthrough e inspection) e dinamica. Durante lo sviluppo del codice è di fondamentale importanza procedere di pari passo con la verifica tramite i test di unità. Il codice prodotto deve seguire le regole definite nel documento *Norme-di-progetto-2.0.pdf*.

2.3 Gestione amministrativa della revisione

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:

- **P**lan: definizione del problema, cosa deve essere realizzato e come andrà controllato per la verifica qualitativa;
- **D**o: eseguire le attività secondo i piani;
- **C**heck: verifica nel tempo dei risultati conseguiti in seguito a modifiche e migliorie, confronto tra risultati attesi e risultati effettivi;
- **A**ct: 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

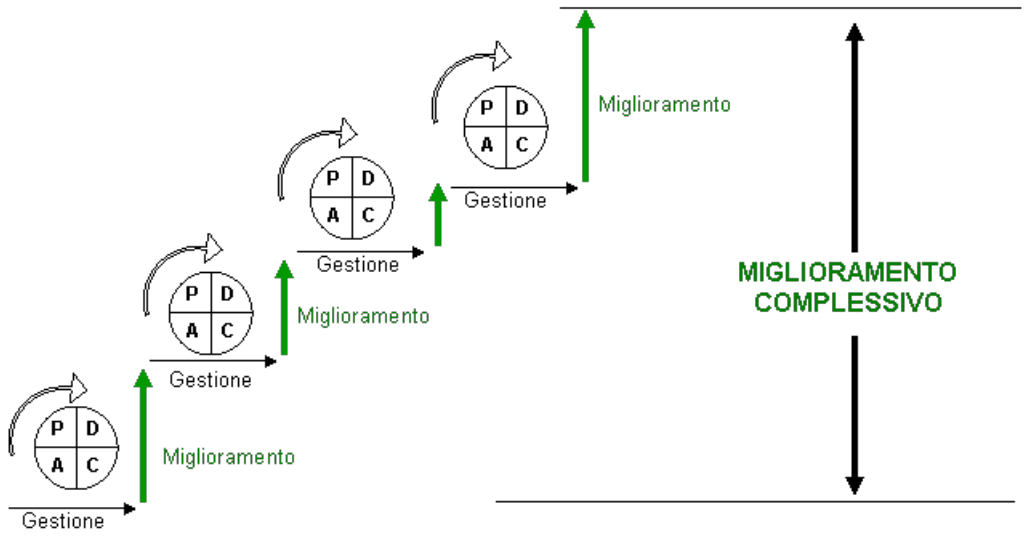

Figura 2: PDCA e miglioramento complessivo

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.

Nella sezione Misure e metriche vengono elencate due metriche (**Schedule variance** e **Budget variance**) necessarie al controllo e alla valutazione dei processi in termini di tempo (schedule) e in termini di costo e risorse impiegate (budget). Tali misurazioni, da effettuare alla

fine di ogni fase, si basano sui valori dei consuntivi di ogni fase (presenti nel documento *piano_di_progetto_1.0.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.
Per il coordinamento delle attività giornaliere ci si affida ad un sistema di **ticketing**$_G$ fornito dalla piattaforma online **Trello**$_G$ (vedere documento *norme_di_progetto_1.0.pdf* per una spiegazione dettagliata del suo utilizzo). Tale sistema permette di tenere costantemente sotto controllo l'andamento dei compiti (**Task**$_G$) e la loro realizzazione.

2.4 Procedure di controllo di qualità di processo

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

Tramite le piattaforme Trello e GitHub 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. In periodi di alta attività il numero di commit per persona deve essere maggiore o uguale a 1 (viene inteso un commit significativo relativo all'intera giornata di lavoro)

2.5 Procedure di controllo qualità di prodotto

- **Software Quality Assurance**: insieme delle attività volte al fine di 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 soddisfa i requisiti.

3 Misure e metriche: parametri di tolleranza

Di seguito vengono elencate le metriche adottate per il controllo qualitativo dei processi, dei documenti e del prodotto software in sviluppo. Per una dettagliata descrizione di ogni metrica vedere la sezione appendici in coda al documento.

3.1 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.

3.2 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	$[50 - 100]$	$[60 - 100]$

Oltre a questo, la documentazione subisce un processo di verifica più approfondito 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-1.0*;
- 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**.

3.3 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
- $10 \leq MI \geq 20$: moderata manutenibilità
- $MI \leq 10$: bassa manutenibilità

A Misure e Metriche

A.1 Di progetto

A.1.1 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:

$$\mathbf{SV = BCWP - BCWS}$$

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: $[\geq 0]$

A.1.2 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:

$$\mathbf{BV = BCWS - ACWP}$$

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: $[\geq 0]$

A.2 Per i documenti

A.2.1 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:

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

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: $[60 - 100]$

A.3 Per il software

In questa sezione vengono elencate le metriche utilizzate dal software di analisi Complexity-report come **Maintainability Index**$_G$, **Metriche di Halstead**$_G$ e **Complessità Ciclomatica**$_G$) e altri indicatori come la percentuale di *Copertura dei test* e la validazione W3C$_G$ dei file HTML$_G$.
Il software complexity-report offre un resoconto delle misurazioni effettuate sull'intero codice, a livello di file, e a livello di funzione presente in ogni file. Di seguito vengono presentate le metriche utilizzate per ogni livello.

A.3.1 Parameter count

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

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

A.3.2 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: $[0 - 10]$

A.3.3 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 Lenght: $N = N_1 + N_2$
- **Volume**: $V = Nlog_2 n$
  - 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: $[0 - 300]$

A.3.4 Dependency count

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

A.3.5 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 Ciclomatica totale

Parametri utilizzati:

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

A.3.6 First-order density

Percentuale di tutte le possibili dipendenze interne tra i moduli del progetto.

- Range di accettazione: $[\leq 20\%]$
- Range ottimale: $[\leq 15\%]$

A.3.7 Change cost

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

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

A.3.8 Core size

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

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

A.3.9 Copertura dei test

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.

*Copertura dei test = Numero funzioni testate * 100/Numero funzioni da testare*

Parametri utilizzati:

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

A.3.10 W3C – Markup Validation Service

Per la validazione delle pagine HTML e i file CSS sviluppati si intende affidarsi allo strumento online W3C Markup Validation Service (https://validator.w3.org/).

B 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-1.0.pdf*. Questo modello prevede un

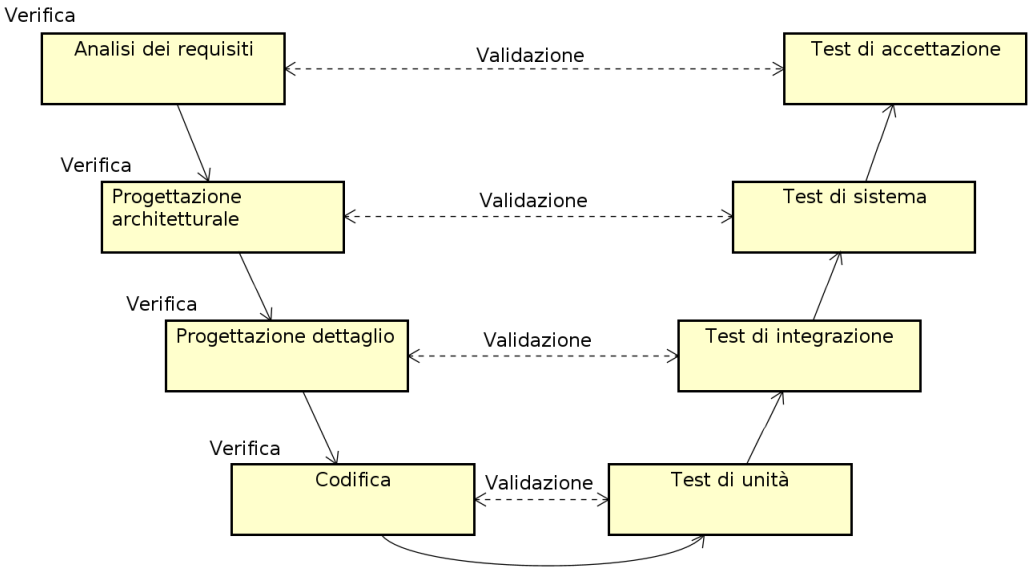

Figura 3: Fasi di sviluppo

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.

B.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)**

B.2 Test di accettazione

B.2.1 Requisiti funzionali

Tabella A.1: Test di accettazione

ID requisito	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: 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 1.2	La sezione dedicata alla registrazione deve permettere l'invio dei dati inseriti. L'utente deve poter ricevere un messaggio di errore se: 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 1.2.2	Quando i dati corretti vengono inviati viene verificato che: 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: 1. inserire la propria mail 2. inserire la propria password 3.	Pianificato

Tabella A.1: Requisiti funzionali

ID	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 2.2	Verifica che la sezione dedicata all'autenticazione fornisca una procedura di recupero password. il sistema deve: 1. permettere l'nserimento 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 2.3	Verifica che la sezione dedicata all'autenticazione permetta l'invio dei dati inseriti. Il sistema deve: 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	Verifica che il sistema permetta la navigazione e selezione dei questionari per categoria. Il sistema deve: 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	Descrizione	Stato	
F 3.2 F 3.2.1 F 3.2.2 F 3.2.3 F 3.2.4	TA 3.2	Verifica che il sistema fornisca una sezione con un elenco dei questionari relativi ad una categoria. Viene verificato che: 1. uando 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	Verifica che il sistema dia la possibilità di compilare un questionario. L'utente deve poter: 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	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 4.5.1	Verifica che alla consegna del questionario il sistema: 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 4.6	Verifica che la compilazione del questionario abbia un tempo massimo. Il sistema deve: 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	Verifica che il sistema dia la possibilità ad un utente autenticato di creare una nuova domanda. L'utente deve potere: 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	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 5.2.1	Verifica che il sistema effettui un controllo sulla correttezza dei dati prima della memorizzazione. Il sistema deve: 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	Verifica che il sistema fornisca la possibilità ad un utente autenticato di creare un nuovo questionario. L'utente deve poter: 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 questinario	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 6.2.1	Verifica che il sistema, prima di inviare i dati, effettui un controllo su di essi. Nello specifico si verifica che il sistema: 1. avvisi con un messaggio che i campi obbligatori non sono stati compilati o non è stata selezionata nessuna domanda 2. crei il nuovo questinario 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

C Resoconto attività di verifica

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