

NetBreak

Progetto API Market



Norme di Progetto

Informazioni sul documento

Nome del file	NormeDiProgetto 0_1_0.pdf
Data di creazione	06 Dicembre 2016
Ultima modifica e versione	06 Dicembre 2016
	Versione 1
Stato	Approvato
Redatto da	Nicolò Scapin
	Andrea Scalabrin
Verificato da	Davide Scarparo
	Alberto Nicolè
Approvato da	Andrea Scalabrin
Uso	Interno
Distribuzione	NetBreak
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, NetBreak

Abstract

Documento contenente le norme di progetto che il gruppo NetBreak seguirà durante tutte le fasi di realizzazione del prodotto API Market.

Changelog

Descrizione	Autore e Ruolo	Data e versione
Prima versione dell'introduzione	Nicolò Scapin Amministratore	2016-12-06 0.0.2
Creato template	Andrea Scalabrin Responsabile di Progetto	2016-12-06 0.0.1

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Riferimenti normativi	1
1.4	Riferimenti informativi	1
1.5	Glossario	1
2	Processi primari	1
2.1	Fornitura	1
2.1.1	Studio di fattibilità	1
2.2	Sviluppo	2
2.2.1	Analisi dei Requisiti	2
2.2.1.1	Casi d'uso	2
2.2.1.1.1	Nomenclatura	2
2.2.1.1.2	Struttura	2
2.2.1.2	Requisiti progettuali	3
2.2.1.2.1	Nomenclatura	3
2.2.1.2.2	Struttura	3
2.2.2	Progettazione	4
2.2.2.1	Obiettivi	4
2.2.3	Programmazione	4
2.2.3.1	Best practices	4
2.2.3.1.1	Impostazione	4
2.2.3.1.2	Nomenclatura	5
2.2.3.1.3	Commenti	5
2.2.4	Strumenti utilizzati	6
2.2.4.1	TexStudio	6
2.2.4.2	Astah	6
3	Processi di supporto	7
3.1	Documentazione	7
3.1.1	Nomenclatura e versione del documento	7
3.1.2	Classificazione del documento	7
3.1.3	Ciclo di vita del documento	7
3.1.4	Strumenti di sviluppo del documento	7
3.1.5	Struttura del documento	8
3.1.5.1	Frontespizio	8
3.1.5.2	Changelog	8
3.1.5.3	Indice	8
3.1.5.4	Intestazione e piè di pagina	8
3.1.5.5	Norme tipografiche	9
3.1.5.5.1	Stili di testo e punteggiatura	9
3.1.5.5.2	Formato Data	9
3.1.6	Lista documenti da consegnare	9
3.1.6.1	Studio di Fattibilità	9
3.1.6.2	Norme di Progetto	10
3.1.6.3	Analisi dei Requisiti	10
3.1.6.4	Piano di Progetto	10
3.1.6.5	Piano di Qualifica	10

3.1.6.6	Glossario	10
3.2	Verifica	10
3.2.1	Analisi	10
3.2.1.1	Analisi statica	10
3.2.1.2	Analisi dinamica	11
3.2.2	Strumenti	11
4	Processi organizzativi	12
4.1	Gestione Organizzativa	12
4.1.1	Comunicazioni	12
4.1.1.1	Comunicazioni interne	12
4.1.1.2	Comunicazioni esterne	12
4.1.2	Riunioni	12
4.1.2.1	Riunioni interne	12
4.1.2.2	Riunioni esterne	12
4.2	Processo di pianificazione	13
4.2.1	Descrizione	13
4.2.1.1	Responsabile di Progetto	13
4.2.1.2	Amministratore	13
4.2.1.3	Progettista	14
4.2.1.4	Analista	14
4.2.1.5	Verificatore	14
4.2.1.6	Programmatore	14
4.3	Strumenti	14
4.3.1	Google Drive	14
4.3.2	GitHub	15
4.3.3	Gestione delle attività del progetto	15
4.3.4	Task management	15
4.3.4.1	Creazione di un elenco di task	15
4.3.4.2	Creazione di un task	16
4.3.4.3	Modifica di un task	16
4.3.4.4	Completamento di un task	16
4.4	Lista di controllo	17

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è la stesura di norme e regolamenti interni che il team seguirà scrupolosamente per la stesura dei documenti di progetto. Il documento funge da linea guida e da regolamento scrupoloso per ogni membro del gruppo al momento della stesura di ciascun documento o per le comunicazioni esterne. Ciò ha lo scopo di produrre documentazione omogenea, seguendo le medesime linee guida. Nel documento saranno descritti accuratamente: le modalità di stesura dei documenti e di revisione, i ruoli e le mansioni, le metodologie di lavoro, i metodi di comunicazione, i metodi di organizzazione e gli strumenti di lavoro. Ogni sottocategoria sarà comprensiva di eventuali norme e convenzioni utilizzate.

1.2 Scopo del prodotto

Lo scopo del prodotto è la realizzazione di un API Market per l'acquisto e la vendita di microservizi. Il sistema offrirà la possibilità di registrare nuove API per la vendita, permetterà la consultazione e ricerca ai potenziali acquirenti, gestendo i permessi tramite creazione e controllo di relative API key. Il sistema, oltre alla webapp stessa, sarà corredato di un API Gateway per la gestione delle richieste e il controllo delle chiavi, e fornirà funzionalità avanzate di statistiche per il gestore della piattaforma nonché per gli utilizzatori.

1.3 Riferimenti normativi

- **ISO 8601** (Rappresentazione di date e orari)
https://it.wikipedia.org/wiki/ISO_8601

1.4 Riferimenti informativi

1.5 Glossario

Per semplificare la consultazione e disambiguare alcune terminologie tecniche, alcune voci indicate con la lettera *G* a pedice sono descritte approfonditamente nel documento apposito GLOSSARIO 1_0_0.PDF

2 Processi primari

2.1 Fornitura

2.1.1 Studio di fattibilità

Lo Studio di fattibilità sarà redatto a partire dalle riunioni con tutto il team, in merito alla scelta del miglior capitolato da svolgere. Conseguentemente alla pubblicazione, saranno fissati degli incontri con i membri del gruppo dove saranno stilati internamente una lista di **Pro** e di **Contro** per ciascun progetto disponibile. Gli analisti del team sono le figure incaricate della stesura dei singoli studi, in relazione a quanto emerso dagli incontri con tutto il gruppo. Il documento prenderà in analisi ciascun capitolato, secondo i sottocitati punti:

- **Descrizione:** Introduzione generale del capitolato
- **Dominio Applicativo:** Descrizione del bacino di utenza e dell'ambito in cui il prodotto sarà utilizzato.
- **Tecnologie:** Breve elenco delle tecnologie di interesse e che saranno centrali nel progetto.

- **Aspetti critici:** Descrizione degli aspetti cruciali del prodotto, su cui è necessario porre particolare attenzione, evidenziando in particolare le problematiche e le difficoltà che si possono incontrare.
- **Considerazioni conclusive:** Valutazione finale e soggettiva del gruppo che, prendendo spunto dai punti precedenti, stila le motivazioni per cui il capitolato analizzato è stato scelto oppure no.

2.2 Sviluppo

2.2.1 Analisi dei Requisiti

L'Analisi dei requisiti dovrà seguire la fase di Studio di fattibilità. Questa fase analizzerà in modo quanto più accurato possibile i requisiti necessari per ciascun ambito del progetto da realizzare. Verranno inoltre stilati i casi d'uso del prodotto, corredati da opportuna descrizione e analisi. La struttura del documento sarà organizzata nei seguenti punti:

- Casi d'uso
- Requisiti progettuali

2.2.1.1 Casi d'uso

2.2.1.1.1 Nomenclatura

La scelta del nome per i Casi d'uso avverrà secondo la seguente procedura:

U[Codice categoria].[Codice progressivo]

dove:

- **Codice categoria:** identifica il codice entro cui lo specifico Use case viene raggruppato. Può essere organizzato eventualmente in ulteriori sottocategorie oppure omesso.
- **Codice progressivo:** identificativo dello specifico Use case

2.2.1.1.2 Struttura

L'analisi di ciascun Caso d'uso dovrà essere strutturata come segue, avendo cura di mantenere l'ordine indicato:

- **Sigla e nome:** Identifica il caso d'uso analizzato. Va indicato nel titolo del paragrafo corrispondente.
- **Diagramma:** Immagine del diagramma per il suddetto Caso d'uso
- **Descrizione:** Breve descrizione
- **Attori:** Descrizione degli attori coinvolti.
- **Stato iniziale:** Descrive la pre-condizione che dev'essere verificata antecedentemente.
- **Stato finale:** Descrive la post-condizione che deve valere all'uscita della situazione descritta.
- **Flusso di esecuzione:** Analisi completa del flusso di esecuzione del caso d'uso comprendente, se necessario, eventuali casi d'uso esterni

Gli elementi sopracitati sono da considerarsi standard, mentre si valuterà caso per caso l'eventuale immissione di ulteriori informazioni quali:

- **Estensioni:** Descrive le estensioni per il Caso d'uso descritto
- **Inclusioni:** Descrive le inclusioni per il Caso d'uso descritto
- **Note:** Ulteriori informazioni

2.2.1.2 Requisiti progettuali

Il documento dovrà contenere un elenco di requisiti, redatto dagli Analisti, che indicheranno le peculiarità richieste e informazioni relative alla loro tipologia e rilevanza.

2.2.1.2.1 Nomenclatura

La scelta del nome per i Requisiti progettuali avverrà secondo la seguente procedura:

R[Rilevanza][Tipologia][Codice]

Rilevanza: che può assumere uno seguenti valori, elencati in ordine dal maggiore al minor rilievo

- **1:** Requisito critico, di fondamentale importanza per il progetto.
- **2:** Requisito principale.
- **3:** Requisito secondario, che rappresenta una caratteristica auspicabile ma non esplicitamente necessaria.
- **4:** Requisito facoltativo.

Tipologia: che può assumere uno seguenti valori

- **V:** Requisito di vincolo, imposto dal committente.
- **F:** Requisito di funzionalità.
- **Q:** Requisito qualitativo.
- **P:** Requisito prestazionale.

Codice: che assume un numero sequenziale e univoco, necessario a catalogare e riconoscere i Requisiti progettuali.

2.2.1.2.2 Struttura

L'analisi di ciascun requisito dovrà essere strutturata come segue, avendo cura di mantenere l'ordine indicato:

- **Descrizione:** una sintesi del requisito, che lo descriva in modo conciso ma quanto più efficace possibile
- **Origine:** descrive la provenienza del requisito, ovvero da chi è stato sollevato e in che ambito.

2.2.2 Progettazione

La progettazione è la fase di preparazione in cui si realizza un'astrazione di quella che diverrà la struttura software del prodotto. Questa fase è successiva alla produzione di una completa Analisi dei Requisiti, in quanto da essa viene tratto spunto per questa fase. I documenti risultanti da questa fase di progettazione fungeranno in seguito da percorso per la produzione del software vero e proprio.

2.2.2.1 Obiettivi

Questo stadio si prefigge i seguenti obiettivi:

- Fornire una visione macroscopica del percorso da seguire in fase di codifica software
- Realizzare un prodotto in rispetto degli standard prefissati in fase di analisi e di analisi dei requisiti.
- Flessibilità del programma, per poter far fronte in modo agile a repentini cambi o modifiche rilevanti, pur non inficiando il lavoro pregresso
- Soddisfare le richieste del committente.

2.2.3 Programmazione

La fase di codifica si prefigge la realizzazione pratica del prodotto richiesto. I programmatori sono i principali responsabili di questa fase, che tenendo conto degli stadi precedenti e seguendo nel dettaglio quanto precedentemente stilato, realizzano un prodotto che sia in grado di svolgere i compiti richiesti. Ogni programmatore dev'essere, come requisito prioritario, strettamente ancorato alle linee guida stabilite.

2.2.3.1 Best practices

La fase di codifica dev'essere molto scrupolosa, attendendosi alle linee guida definite all'interno di questo documento. Esse compongono gli standard di codifica per la realizzazione del progetto. Gli standard sotto riportati hanno il compito di:

- Fornire uno strumento per garantire una più agevole cooperazione tra i diversi sviluppatori
- Favorire la creazione di un codice valido e ben formato
- Favorire la realizzazione di un software di qualità
- Rendere più agevole la lettura e modifica del codice internamente al team

Le regole stilistiche del codice riguardano le parti non prettamente implementative, ma di ausilio ai vari programmatori per permettere una miglior cooperazione e comprensione del codice. Esse riguardano la nomenclatura, La definizione di tecniche di scrittura è importante per avere una maggiore comprensione del codice sorgente. Le tecniche di scrittura del codice descritte nel seguente documento sono suddivise in tre categorie: Impostazione, Nomenclatura e Commenti.

2.2.3.1.1 Impostazione

L'impostazione stilistica del documento consente una più agevole comprensione e lettura del codice, e pertanto è importante fornire una logica comune. Di seguito sono elencate le principali norme relative alla formattazione del codice:

- I blocchi di codice vanno indentati tramite i rientri standard, evitando spaziature singole. Ogni porzione di codice interna ad un'altra dev'essere rigorosamente indentata in modo corretto.
- Utilizzare gli spazi per separare gli operatori, ove possibile. Aggiungere spazi che aumentino la leggibilità del codice in tutte le situazioni dove ciò è permesso, a patto che il funzionamento non risulti alterato.
- Dividere il codice in modo quanto più modulare possibile tra vari file, evitando file monolitici di ingenti dimensioni e di difficile modifica. Ciò è un requisito fondamentale per un corretto e miglior utilizzo dei software di versionamento.

2.2.3.1.2 Nomenclatura

Nell'elenco sottostante sono elencati gli standard di nomenclatura utilizzati per il codice che verrà prodotto. Tale aspetto è fondamentale per una miglior comprensione del codice.

- Assegnare nomi univoci ai costrutti, evitando possibili ambiguità dovute a nomi simili. I costrutti devono avere nomi quanto più significativi possibili, possibilmente anche se utilizzati in porzioni marginali. Le variabili formate da un solo carattere sono invece consentite qualora si tratti di iterazioni circoscritte.
- Nella scelta dei nomi di files e cartelle, è importante che siano descrittivi del proprio contenuto nel modo più accurato possibile.
- Accertarsi che le parole utilizzate siano scritte nella corretta ortografia della lingua a cui appartiene.
- Nella scrittura di nomi composti utilizzare la forma "Camel Case", ottenuta scrivendo con iniziale minuscola la prima parola di un nome composto, seguito dalle successive parole con iniziale sempre maiuscola.
- Nella descrizione di funzioni, utilizzare come prima voce il verbo dell'azione svolta, seguito dal nome dell'azione o dell'oggetto su cui viene eseguita.
- Accertarsi che non vi sia ambiguità nelle abbreviazioni utilizzate, avendo cura di verificare che non possano venir utilizzate due abbreviazioni simili per funzionalità differenti.

2.2.3.1.3 Commenti

La fase di commento deve essere ampia e uniforme, seguendo i criteri e le linee guida elencate:

- Produrre o mantenere aggiornata un'intestazione per ciascun file, comprensiva di una descrizione di ciò che è incluso in quello specifico documento.
- Indicare pre e post condizione di ogni funzionalità di rilievo, tali da favorire un riutilizzo di eventuale codice già scritto. Al più, si introduca brevemente ciascuna porzione con una descrizione concisa.
- Inserire commenti nella riga precedente al codice o porzione che desidero commentare. Non si commenti, dunque, in modalità "inline".
- Stilare commenti completi, di senso compiuto e grammaticalmente corretti. Documentare inoltre in modo tempestivo, durante la stesura del codice, per evitare di dover rianalizzare una porzione al solo fine di documentarla.

2.2.4 Strumenti utilizzati

2.2.4.1 TexStudio

In seguito a una breve fase di test, la scelta del team per l'editor di documenti Latex è ricaduta su **TexStudio**. Esso è un noto fork di un altrettanto famoso editor, TexMaker. TexStudio è un editor completamente gratuito e OpenSource. Esso è disponibile per i più diffusi sistemi operativi. L'editor consente un agevole utilizzo dei file tex, inclusa una gestione dei documenti inclusi. Tale peculiarità gioca a nostra favore per una semplificata predisposizione alla modularità dei documenti. L'editor scelto possiede numerose features rilevanti, quali autocompletamento e controllo della sintassi, anteprema incorporata, e controllo ortografico. Per le motivazioni elencate, l'editor scelto sarà il principale strumento tramite cui verranno redatti i documenti necessari al progetto.

2.2.4.2 Astah

Per la stesura e modellazione di diagrammi in linguaggio UML, la scelta è ricaduta sull'editor Astah. Esso è un editor multi-piattaforma gratuito. Il suo punto forte rispetto ad altre soluzioni è la semplicità di utilizzo. Tra le funzionalità interessanti ai nostri fini c'è la possibilità di esportare i diagrammi creati direttamente in un file immagine. Il software Astah sarà utilizzato per la creazione di tutti i diagrammi necessari in sede di progettazione ed analisi

3 Processi di supporto

3.1 Documentazione

In questa sezione del documento verrà spiegata la struttura, classificazione e metodo di redazione dei documenti del gruppo *NetBreak*.

3.1.1 Nomenclatura e versione del documento

Ogni documento deve rispettare una denominazione comune il più possibile chiara affinché venga individuato facilmente il nome e la versione del documento. Il gruppo ha scelto di utilizzare la seguente nomenclatura

$$\text{NomeDelDocumento } X_Y_Z.pdf$$

All'interno del nome X_Y_Z indica la versione del documento in questo modo:

- X : indica il numero di pubblicazioni del documento. Questo indice è incrementato esclusivamente dal *Responsabile di Progetto* in seguito all'approvazione finale da parte sua. L'incremento di tale indice, azzerà automaticamente Y e Z ;
- Y : indica il numero di verifiche e viene incrementato esclusivamente dai *Verificatori*. L'incremento di tale indice, azzerà automaticamente Z ;
- Z : indica il numero di aggiornamenti minori effettuati prima o in seguito a una verifica o approvazione. Viene incrementato progressivamente e ritorna a zero solo in seguito a una modifica di X e Y .

Ogni modifica della versione del documento deve riflettersi nel changelog, nel nome del documento e nel frontespizio nella voce "Ultima modifica e versione".

3.1.2 Classificazione del documento

Tutti i documenti redatti dal gruppo devono avere la classificazione, che può essere **Interna** o **Esterna**. Un documento viene classificato come **Interno** se la sua consultazione è limitata al gruppo. Al contrario, se il documento è destinato a una consultazione esterna al gruppo, è classificato come **Esterno**.

3.1.3 Ciclo di vita del documento

Un documento può trovarsi in tre stati.

- Documento in Bozza: il/i redattore/i è in fase di stesura del documento;
- Documento in attesa di verifica: il/i redattore/i ha concluso la stesura, il documento passa in stato di verifica in attesa del *Verificatore*;
- Documento approvato: il *Verificatore* ha concluso il suo lavoro e passa il documento dal *Responsabile di Progetto* che ha il compito di approvarlo definitivamente.

3.1.4 Strumenti di sviluppo del documento

Per la stesura dei documenti è stato scelto di utilizzare \LaTeX_G , un linguaggio semplice e modulare. \LaTeX_G è in grado di evitare possibili conflitti provenienti dall'utilizzo di software e piattaforme differenti.

3.1.5 Struttura del documento

Ogni documento redatto dal gruppo avrà una struttura chiara e di facile comprensione per chiunque. Di seguito viene riportata la struttura che sarà utilizzata nei documenti.

3.1.5.1 Frontespizio

- Nome del gruppo;
- Nome del progetto;
- Logo del gruppo;
- Nome del documento;
- Nome del file;
- Data di creazione;
- Ultima modifica e versione;
- Stato;
- Nome e cognome del/i redattore/i;
- Nome e cognome del verificatori;
- Classificazione del documento;
- Distribuzione;
- Destinatari del documento;

3.1.5.2 Changelog

La seconda sezione del documento è una lista delle modifiche effettuate al documento, dalla sua creazione alla sua approvazione finale. Le modifiche devono essere inserite con questo schema:

- Descrizione: questo campo indica cosa è stato cambiato nel documento;
- Autore e ruolo: questo campo indica l'autore e ruolo della persona che ha effettuato la modifica;
- Data e versione: questo campo indica la data di modifica e la nuova versione del documento in questione.

3.1.5.3 Indice

In ogni documento deve essere presente l'indice di tutte le sezioni del documento; ogni immagine o tabella deve essere indicata con il relativo indice.

3.1.5.4 Intestazione e piè di pagina

Tutte le pagine del documento, eccetto la prima hanno una intestazione e piè di pagina. L'intestazione contiene nell'angolo sinistro il nome del progetto e del documento in questione. Nell'angolo destro è presente il nome del capitolo contenente quella sezione, ad esclusione del changelog. Il piè di pagina contiene nell'angolo sinistro il nome del gruppo e l'indirizzo email ufficiale del gruppo. Nell'angolo destro è presente una numerazione in numeri romani del changelog e dell'indice, mentre per le pagine rimanenti è utilizzata una numerazione con numeri arabi, che indica la pagina attuale e il numero totale di pagine del documento.

3.1.5.5 Norme tipografiche

Questa sezione del documento contiene i criteri riguardanti l'ortografia e la tipografia utilizzate nel corso dello sviluppo del progetto.

3.1.5.5.1 Stili di testo e punteggiatura

- **Grassetto**: il grassetto deve essere utilizzato per parole importanti all'interno di frasi o elenchi puntati;
- **Corsivo**: il corsivo deve essere utilizzato per:
 - Citazioni;
 - Parola del glossario (unitamente ad una G maiuscola in pedice);
 - Riferimenti ai ruoli;
 - Riferimento all'intero gruppo.
- **MAIUSCOLO**: viene utilizzato unicamente per scrivere acronimi e macro \LaTeX presenti nei documenti;
- **MAIUSCOLETTO**: viene utilizzato per i riferimenti ad altri documenti;
- **\LaTeX** : viene usato il comando `\LaTeX` per ogni occorrenza del termine \LaTeX .
- **Punteggiatura**: Ogni simbolo di punteggiatura è seguito da uno spazio, ad eccezione del punto, del punto interrogativo, del punto esclamativo è seguito da uno spazio e lettera maiuscola. Ogni voce di un elenco puntato deve terminare con punto e virgola, ad eccezione dell'ultima che termina con un punto.

3.1.5.5.2 Formato Data

All'interno di ogni documento, tutte le date seguiranno lo standard ISO_G 8601:2004_G:

YYYY-MM-DD

Dove:

- YYYY: indica l'anno;
- MM: indica il mese;
- DD: indica il giorno;

3.1.6 Lista documenti da consegnare

3.1.6.1 Studio di Fattibilità

- **Classificazione**: Interno;
- **Destinazione**: gruppo e committenti;
- **Contenuto**: questo documento contiene lo studio effettuato dal gruppo su tutti i capitoli e le motivazioni che hanno portato all'accettazione o meno del progetto.

3.1.6.2 Norme di Progetto

- Classificazione: Interno;
- Destinazione: gruppo e committenti;
- Contenuto: l'intento del documento è raccogliere tutte le convenzioni, strumenti e regole che il gruppo adotterà durante la realizzazione del progetto.

3.1.6.3 Analisi dei Requisiti

- Classificazione: Esterno;
- Destinazione: gruppo, committenti e proponente;
- Contenuto: questo documento si prefigge lo scopo di dare una visione generale dei requisiti e di tutti i possibili casi d'uso dell'API Market.

3.1.6.4 Piano di Progetto

- Classificazione: Esterno;
- Destinazione: gruppo, committenti e proponente;
- Contenuto: questo documento descrive come il gruppo ha impiegato tempo e risorse umane.

3.1.6.5 Piano di Qualifica

- Classificazione: Esterno;
- Destinazione: gruppo, committenti e proponente;
- Contenuto: questo documento contiene come il gruppo intende raggiungere gli obiettivi di qualità prefissati all'inizio del progetto.

3.1.6.6 Glossario

- Classificazione: Esterno;
- Destinazione: gruppo, committenti e proponente;
- Contenuto: questo documento intende fornire una definizione di tutti i termini tecnici e acronimi, rendendo la lettura facile a tutti i destinatari.

3.2 Verifica

3.2.1 Analisi

3.2.1.1 Analisi statica

L'analisi statica è applicata ai documenti di testo, consiste nel trovare errori nella sintassi e ortografica. Una prima verifica viene effettuata dai *Verificatori* e successivamente dal *Responsabile di Progetto*. Le due tecniche scelte sono la **Formal Walkthrough_G** e la **Fagan Inspection_G**.

- **Formal Walkthrough**: questa tecnica è focalizzata nel trovare errori di sintassi e ortografici, senza concentrarsi sul trovare errori in particolare. Questa tecnica sarà utilizzata dai *Verificatori*, i quali trascriveranno gli errori più frequenti in una apposita lista, necessaria per la tecnica di Fagan Inspection;

- **Fagan Inspection:** Questa tecnica si basa su una lettura attenta dei documenti, basandosi sulla lista degli errori stilata durante la **Formal Walkthrough**. Questo processo acquisirà rilevanza in concomitanza con l'incremento della lista di possibili errori stilata dai *Verificatori*.

3.2.1.2 Analisi dinamica

L'analisi dinamica è applicata esclusivamente sul software prodotto, in quanto consiste nell'effettuare test per verificare il corretto funzionamento dell'applicativo.

3.2.2 Strumenti

- W3C markup validator service_G: validatore online di codice HTML_G, utile per trovare eventuali errori nel codice. L'indirizzo web di riferimento è: <https://validator.w3.org/>;
- CSS Validation Service_G: validatore online di codice CSS_G, utile per trovare eventuali errori nel codice. L'indirizzo web di riferimento è: <https://jigsaw.w3.org/css-validator/>;
- SQLFiddle: validatore online di codice MySQL_G. Utile per verificare la consistenza del codice del database_G. L'indirizzo web di riferimento è: <http://sqlfiddle.com/>;
- PHP Code Checker_G: strumento online che verifica la presenza di eventuali errori di sintassi del codice PHP_G. L'indirizzo web di riferimento è: <http://phpcodechecker.com/>;

4 Processi organizzativi

4.1 Gestione Organizzativa

4.1.1 Comunicazioni

Al fine di ottenere comunicazioni chiare e concise il *Responsabile di Progetto* deve gestirle in modo strutturato, utilizzando la forma che meglio si adatta alla situazione. Le comunicazioni possono essere interne o esterne al *team_G*.

4.1.1.1 Comunicazioni interne

Le comunicazioni interni sono uno strumento ad uso esclusivo dei membri del gruppo e sono in stile informale in forma scritta o orale. Se la comunicazione avvengono in forma orale tra alcuni membri del gruppo, è loro dovere riportarla in forma scritta se è una informazione di rilievo riguardo lo sviluppo del software, utilizzando l'applicazione di instant messaging_G Telegram_G, sulla quale è presente la chat del gruppo.

4.1.1.2 Comunicazioni esterne

Per tutte le comunicazioni esterne il gruppo si è dotato di una email ufficiale, necessaria per gestire le comunicazioni con i committenti e proponenti. La gestione di tale casella di posta elettronica è affidata al *Responsabile di Progetto*, il quale ha il compito di tenere informati i membri riguardo le comunicazioni importanti. La casella di posta elettronica è:

`netbreak.swe@gmail.com`

4.1.2 Riunioni

Le riunioni sono un elemento essenziale per il corretto svolgimento del progetto. Hanno il compito di confrontare i membri del gruppo tra di loro e con il proponente e committenti. Una riunione condotta correttamente permette di risparmiare tempo e gestire al meglio le risorse umane.

4.1.2.1 Riunioni interne

Il *Responsabile di Progetto* ha facoltà di decidere e convocare la riunione del gruppo. Tale convocazione deve essere inviata tramite posta elettronica a tutti i membri del gruppo con almeno due giorni di anticipo e deve contenere l'ordine del giorno. Tutti i membri del gruppo sono obbligati a confermare la loro presenza o meno entro ventiquattro ore dalla ricezione della comunicazione. Se impossibilitati a partecipare, devono motivare tale impedimento al *Responsabile di Progetto*. Il *Responsabile di Progetto* ha il compito di dirigere la riunione, tenendo fede all'ordine del giorno precedentemente inviato, ed evitare perdite di tempo. Tutti i partecipanti devono presentarsi alla riunione preparati, aver rispettato le scadenze decise alla riunione precedente, visibili su sulla App multiplatforma Asana_G e caricato il materiale assegnato sulla piattaforma di Web Storage_G indicata. All'inizio di ogni riunione il *Responsabile di Progetto* avrà il compito di nominare un segretario. Il segretario avrà il compito di tenere la minuta dell'incontro, stilare un verbale informale a disposizione esclusivamente del gruppo e creare le task con le scadenze concordate su Asana_G.

4.1.2.2 Riunioni esterne

Queste riunioni sono fondamentali per il corretto svolgimento del progetto. Prima di ogni riunione il *Responsabile di Progetto* nominerà un segretario, incaricato di prendere nota dei temi discussi e redigere un verbale, evidenziando gli aspetti più rilevanti e/o modifiche da apportare. Tale

documento è ad uso esclusivo del gruppo. A questi incontri parteciperanno tutti i membri del gruppo (salvo impedimenti di rilievo) e sarà compito del *Responsabile di Progetto* coordinare e gestire l'interazione tra il gruppo e l'interlocutore. Durante la riunione i membri del gruppo potranno effettuare domande e/o chiedere chiarimenti.

4.2 Processo di pianificazione

4.2.1 Descrizione

Durante l'intero sviluppo del progetto didattico ogni componente del gruppo dovrà obbligatoriamente cimentarsi in tutti i ruoli elencati di seguito.

Inoltre non potrà mai accadere che un membro del gruppo risulti redattore e verificatore di un medesimo documento. In questo modo si tende ad evitare il conflitto di interessi che potrebbe sorgere se la responsabilità della stesura e della verifica di un documento fosse affidata ad un'unica persona. Un membro può inoltre ricoprire più ruoli contemporaneamente.

4.2.1.1 Responsabile di Progetto

Il *Responsabile di Progetto* è colui che detiene la responsabilità del lavoro svolto dall'intero *team_G*. Rappresenta inoltre colui che mantiene i contatti diretti presso il fornitore e il cliente, ovvero gli enti esterni. Più in dettaglio, ha responsabilità su:

- Pianificazione, coordinamento e controllo generale delle attività;
- Gestione delle risorse;
- Analisi e gestione dei rischi;
- Gestione e approvazione della documentazione;
- Contatti con gli enti esterni.

Il *Responsabile di Progetto* redige l'*organigramma_G*, si assicura che tutte le attività vengano svolte seguendo rigorosamente le *Norme di Progetto*, si assicura che vengano rispettati i ruoli assegnati nel *Piano di Progetto* e che non si presentino conflitti di interesse tra redattori e verificatori. Ha inoltre l'incarico di creare, assegnare ad ogni membro e gestire i singoli task. Redige il *Piano di Progetto* e collabora alla stesura del *Piano di Qualifica*. Il *Responsabile di Progetto* è l'unica persona in grado di approvare in modo definitivo un documento.

4.2.1.2 Amministratore

L'*Amministratore* è il responsabile di tutto ciò che riguarda l'ambiente di lavoro. Più in dettaglio, egli si occupa di:

- Controllo dell'ambiente di lavoro;
- Gestione del versionamento della documentazione tramite l'uso di database;
- Controllo delle versioni e delle configurazioni del prodotto;
- Risoluzione dei problemi legati alla gestione dei processi.

L'*Amministratore* redige le *Norme di Progetto* e collabora alla stesura del *Piano di Progetto*.

4.2.1.3 Progettista

Il *Progettista* è il responsabile di tutto ciò che riguarda la progettazione. Più in dettaglio, egli si occupa di:

- Produrre una soluzione attuabile, robusta e semplice entro i limiti di tempo stabiliti;
- Effettuare scelte progettuali volte a garantire la manutenibilità e la modularità del prodotto software.

Il *Progettista* redige la *Specifica Tecnica*, la *Definizione di Prodotto* e la parte programmatica del *Piano di Qualifica*.

4.2.1.4 Analista

L'*Analista* si occupa di tutto ciò che riguarda l'analisi del problema da affrontare. Le mansioni principali sono quelle di:

- Studiare a fondo e capire le problematiche del prodotto da realizzare;
- Produrre una specifica di progetto comprensibile per il *proponente*, per il *committente* e per il *Progettista*.

L'*Analista* redige lo *Studio di Fattibilità*, l'*Analisi dei Requisiti* e parte del *Piano di Qualifica*.

4.2.1.5 Verificatore

Il *Verificatore* è responsabile di tutto ciò che riguarda l'attività di verifica. Effettua la verifica dei documenti utilizzando gli strumenti e i metodi proposti nel *Piano di Qualifica* e seguendo rigorosamente quanto descritto nelle *Norme di Progetto*. Egli ha il compito di garantire la conformità rispetto le *Norme di Progetto* dei documenti da lui verificati. Si occupa di redigere la sezione del *Piano di Qualifica* che illustra l'esito delle verifiche effettuate sui documenti.

4.2.1.6 Programmatore

Il *Programmatore* si occuperà di implementare le soluzioni del *Progettista*, è quindi responsabile dell'attività di codifica. In dettaglio, i suoi compiti sono:

- implementare le soluzioni descritte dal *Progettista* in maniera rigorosa;
- Scrivere il codice rispettando le convenzioni prese nel presente documento;
- Implementare i test per il codice scritto da utilizzare per l'attività di verifica.

Il *Programmatore* redige il *Manuale Utente* e produce la documentazione del codice.

4.3 Strumenti

4.3.1 Google Drive

Google Drive_G è un servizio di storage online utilizzato dal *team_G* per condividere file e documenti che non necessitano di tracciamento. *Google Drive_G* consente anche il lavoro concorrente su uno stesso file.

4.3.2 GitHub

Gli strumenti software per il controllo versione sono ritenuti molto spesso necessari per la maggior parte dei progetti di sviluppo software. Per questo motivo, il $team_G$ ha deciso di creare un $repository_G$, ospitato all'interno di un server, per gestire la continua evoluzione dei documenti digitali come il codice sorgente del software, la documentazione testuale e altre informazioni importanti. Il software di versionamento scelto è Git_G perchè presenta molti più aspetti positivi rispetto ai $repository_G$ centralizzati. Git_G permette di lavorare anche in assenza di connettività con il server centrale. I membri del $team_G$ possono lavorare sulla propria copia locale del $repository_G$ e rendere pubbliche le modifiche caricandole nel $repository_G$ centrale. Questo approccio risulta molto semplice e veloce per poter collaborare con i membri del $team_G$ simultaneamente. Inoltre online è possibile reperire moltissima documentazione a riguardo per un rapido apprendimento. Il servizio scelto per il mantenimento dei dati è $GitHub_G$.

La struttura base del $repository_G$:

- RR;
 - Interni;
 - Esterni.
- RQ;
- RP;
- RA;
- Template.

4.3.3 Gestione delle attività del progetto

Per gestire nella maniera più opportuna la divisione del lavoro, si è scelto di utilizzare il sistema di pianificazione delle attività $Zoho_G$.

4.3.4 Task management

Vengono creati dal *Responsabile di Progetto* e sono assegnati a due singoli membri del gruppo, il primo in qualità di redattore e il secondo in qualità di *Verificatore*. Per una gestione più chiara di questa divisione delle attività, sono stati creati su $Zoho_G$ due insiemi di $task_G$ per ogni documento, definiti:

- *Nome del documento*;
- *Nome del documento - da verificare*.

Una volta che il proprietario ritiene il suo $task_G$ completato, deve spostarlo nella relativa sezione *da verificare* in modo tale che il verificatore interessato possa controllarlo e segnalarlo come completato, se risulta idoneo.

4.3.4.1 Creazione di un elenco di task

Un elenco di $task_G$ rappresenta l'insieme dei $task_G$ necessari per la realizzazione di un intero documento. Per la creazione di un nuovo insieme di $task_G$ bisogna seguire le seguenti istruzioni:

1. Dalla HomePage di $Zoho_G$ selezionare il progetto interessato (API Market);
2. Selezionare la voce **Compiti e Pietre miliari** dal menù laterale;

3. Selezionare la voce **Nuovo elenco di compiti** e compilare l'elenco di $task_G$ nel seguente modo:

- **Elenco dei compiti:** assegnare il nome del documento che si vuole rappresentare;
- **Pietra miliare collegata:** selezionare a quale pietra miliare si vuole collegare la realizzazione di questo insieme di $task_G$. Corrispondono alla versione finale del documento.

4.3.4.2 Creazione di un task

Per la creazione di un nuovo singolo $task_G$ bisogna seguire le seguenti istruzioni:

1. Dalla HomePage di *ZohoG* selezionare il progetto interessato (API Market);
2. Selezionare la voce **Compiti e pietre miliari** dal menù laterale;
3. Selezionare la voce **Nuovo compito** e compilare il $task_G$ nel seguente modo:
 - **Nome task:** assegnare un nome identificativo al $task_G$ seguito dalla $milestone_G$ corrispondente;
 - **Aggiungi descrizione:** inserire una descrizione breve ma concisa del $task_G$, deve avere la seguente forma:
 - Redattore: nome del redattore;
 - Verificatore: nome del verificatore.
 - **Elenco dei compiti:** selezionare uno degli elenchi di $task_G$, creati precedentemente, che hanno come fine ultimo la realizzazione di un documento;
 - **Chi è il responsabile?:** inserire l'intestatario del $task_G$ come primo utente, nel secondo invece inserire il verificatore assegnato;
 - **Data di conclusione:** Inserire la data ultima per il completamento del $task_G$;
 - **Priorità:** inserire, opzionalmente, una priorità al $task_G$.

4.3.4.3 Modifica di un task

Per modificare un $task_G$ seguire le seguenti istruzioni:

- Selezionare il $task_G$ da modificare;
- Dalla pagina proposta selezionare il campo che si vuole modificare;
- Completata la modifica premere il pulsante **Invio**.

4.3.4.4 Completamento di un task

Dopo che il verificatore ha appurato che il $task_G$ soddisfa i requisiti, e quindi è stato redatto secondo le *Norme di Progetto*, può procedere con il suo completamento seguendo le seguenti istruzioni:

- Selezionare il $task_G$ da segnare come completato;
- Nella parte inferiore della pagina proposta selezionare la voce **Contrassegna come completato**.

Dopo questa operazione, il $task_G$ viene spostato automaticamente nella lista dei $task_G$ completati. Da qui, in casi eccezionali, può essere rispostato nei $task_G$ da verificare. Questi casi eccezionali possono essere:

- Il *Responsabile di Progetto* non ha approvato il documento e quindi deve essere rivisto;
- Il *Verificatore*, dopo un secondo controllo sui $task_G$ a lui assegnati, può accorgersi di errori e/o incompletezze. Deve quindi essere rivisto.

4.4 Lista di controllo

Durante l'applicazione della tecnica del *Walkthrough_G* ai documenti sono stati riportati più frequentemente i seguenti errori:

- **Norme stilistiche:**
 - La prima parola di una voce dell'elenco puntato non inizia con una lettera maiuscola;
 - La voce dell'elenco puntato termina con un punto anziché con un punto e virgola o viceversa;
 - I due punti in grassetto;
 - Errori di battitura.
- **Italiano:**
 - Maiuscole usate impropriamente.
- **L^AT_EX:**
 - Mancato utilizzo dei comandi L^AT_EX personalizzati;
 - Scambiato il comando `\textit{...}` con `\textbf{...}`.
- **Casi d'Uso:**
 - Mancato rispetto del template stabilito per i punti trattati nei casi d'uso.
- **Glossario:**
 - Sono stati evidenziati dei termini che non andavano nel *Glossario*;
 - Non sono stati evidenziati dei termini che sono presenti nel *Glossario*.