

NetBreak

Progetto API Market



Piano di Qualifica

Informazioni sul documento

Nome del documento	PianoDiQualifica 4_0_0.pdf
Data di creazione	20 Dicembre 2016
Ultima modifica	11 Giugno 2017
Versione	4.0.0
Stato	Approvato
Redatto da	Alberto Nicolè Nicolò Scapin Davide Scarparo
Verificato da	Davide Scarparo
Approvato da	Andrea Scalabrin
Uso	Esterno
Distribuzione	NetBreak
Destinato a	Prof. Tullio Vardanega, Prof. Riccardo Cardin, ItalianaSoftware S.r.l.
Email di riferimento	netbreakswe@gmail.com

Abstract

Questo documento contiene le strategie adottate dal team NetBreak per istituire un controllo di qualità e garantire gli standard richiesti per il prodotto API Market.

Changelog

Versione	Data	Autore	Ruolo	Descrizione
4.0.0	2017-06-11	Andrea Scalabrin	Responsabile	Approvazione del documento
3.1.0	2017-06-10	Davide Scarparo	Verificatore	Revisione del documento
3.0.3	2017-06-08	Nicolò Scapin	Verificatore	Aggiornati i test e relativi risultati per RA nella sezione #A.6, #B.2 e #C.2
3.0.2	2017-05-30	Nicolò Scapin	Verificatore	Aggiunte sezioni #4.4: "Test di non regressione" e #4.5: "Test di validazione"
3.0.1	2017-05-28	Nicolò Scapin	Verificatore	Ampliata sezione sezione #4.3: "Test di sistema"
3.0.0	2017-04-12	Davide Scarparo	Responsabile	Approvazione del documento
2.3.0	2017-04-08	Alberto Nicolè	Verificatore	Verifica del documento
2.2.2	2017-04-04	Andrea Scalabrin	Analista	Corrette sezioni #4.1 e #4.2
2.2.1	2017-04-02	Davide Scarparo	Analista	Aggiunta sezione #4.3: "Test di sistema"
2.2.0	2017-03-30	Marco Casagrande	Verificatore	Verifica del documento
2.1.3	2017-03-28	Andrea Scalabrin	Analista	Ampliate sezioni #4.1: "Test unità" e #4.2: "Test integrazione"
2.1.2	2017-03-25	Davide Scarparo	Analista	Aggiunte sezioni #3.8: "Software construction process" e #3.9: "System-Software Qualification Testing Process"
2.1.1	2017-03-23	Davide Scarparo	Analista	Stesura appendice C: "Resoconto qualità di processo"
2.1.0	2017-03-21	Andrea Scalabrin	Verificatore	Verifica del documento
2.0.1	2017-03-19	Nicolò Scapin	Responsabile	Apportate correzioni alle sezioni #3 e #4

Versione	Data	Autore	Ruolo	Descrizione
2.0.0	2017-03-04	Davide Scarparo	Responsabile	Approvazione del documento
1.1.0	2017-03-03	Andrea Scalabrin	Verificatore	Verifica del documento
1.0.4	2017-03-01	Nicolò Scapin	Analista	Stesura appendice B:" Resoconto Qualità di prodotto
1.0.3	2017-02-28	Alberto Nicolè	Analista	Stesura appendice A: "Attività di verifica"
1.0.2	2017-02-24	Davide Scarparo	Analista	Ampliata sezione #4: aggiunti test di integrazione
1.0.1	2017-02-23	Davide Scarparo	Analista	Ristrutturazione documento sezioni #2, #3, #4: nello specifico #2.3, #2.4, #3.2 e #4.1
1.0.0	2017-01-03	Nicolò Scapin	Responsabile	Approvazione del documento
0.1.1	2016-12-31	Davide Scarparo	Analista	Effettuate modifiche secondo verifica
0.1.0	2016-12-29	Marco Casagrande	Verificatore	Verifica del documento
0.0.5	2016-12-28	Davide Scarparo	Analista	Stesura sezione #4: "Test"
0.0.4	2016-12-26	Dan Serbanoiu	Analista	Stesura sezione #3: "Qualità di processo"
0.0.3	2016-12-23	Alberto Nicolè	Analista	Stesura sezione #2: "Qualità di prodotto"
0.0.2	2016-12-22	Alberto Nicolè	Analista	Stesura sezione #1: "Introduzione"
0.0.1	2016-12-20	Andrea Scalabrin	Responsabile	Creato template documento

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	2
2	Qualità di prodotto	3
2.1	Definizione degli obiettivi di qualità	4
2.1.1	Funzionalità	4
2.1.1.1	Obiettivi	4
2.1.1.2	Metriche	4
2.1.1.2.1	Completezza delle funzioni sviluppate	4
2.1.1.2.2	Correttezza delle funzioni sviluppate	4
2.1.1.2.3	Accuratezza rispetto alle aspettative	5
2.1.1.2.4	Controllo degli accessi	5
2.1.2	Affidabilità	5
2.1.2.1	Obiettivi	5
2.1.2.2	Metriche	5
2.1.2.2.1	Chiamate a microservizi corrette	5
2.1.2.2.2	Copertura dei test	6
2.1.2.2.3	Controllo dei guasti	6
2.1.3	Usabilità	6
2.1.3.1	Obiettivi	6
2.1.3.2	Metriche	7
2.1.3.2.1	Comprensibilità delle funzionalità offerte	7
2.1.3.2.2	Controllo e monitoraggio delle operazioni	7
2.1.3.2.3	Qualità della messaggistica	7
2.1.4	Efficienza	8
2.1.4.1	Obiettivi	8
2.1.4.2	Metriche	8
2.1.4.2.1	Tempo di risposta	8
2.1.5	Manutenibilità	8
2.1.5.1	Obiettivi	8
2.1.5.2	Metriche	9
2.1.5.2.1	Impatto delle modifiche	9
2.1.6	Portabilità	9
2.1.6.1	Obiettivi	9
2.1.6.2	Metriche	9
2.1.6.2.1	Supporto differenti versioni dei browser	9
3	Qualità di processo	10
3.1	Infrastructure Management Process	12
3.1.1	Obiettivi	12
3.1.2	Metriche	12
3.1.2.1	Disponibilità <i>NetBreakDB</i>	12
3.2	Project Planning, Assessment & Control Process	13
3.2.1	Obiettivi	13
3.2.2	Metriche	13

3.2.2.1	Schedule Variance	13
3.2.2.2	Budget Variance	14
3.3	Risk Management Process	14
3.3.1	Obiettivi	14
3.3.2	Metriche	14
3.3.2.1	Rischi non preventivati	14
3.4	System/Software Requirements Analysis Process	15
3.4.1	Obiettivi	15
3.4.2	Metriche	15
3.4.2.1	Adempimento requisiti obbligatori	15
3.5	System/Software Architectural Design Process	15
3.5.1	Obiettivi	16
3.5.2	Metriche	16
3.5.2.1	Fan In	16
3.5.2.2	Fan Out	16
3.6	Software Architectural Design Process	16
3.6.1	Obiettivi	17
3.7	Software Detailed Design Process	17
3.7.1	Obiettivi	17
3.7.2	Metriche	17
3.7.2.1	Numero di metodi per classe	17
3.7.2.2	Numero di parametri per metodo	17
3.7.2.3	Numero di attributi per classe	18
3.8	Software Construction Process	18
3.8.1	Obiettivi	18
3.8.2	Metriche	18
3.8.2.1	Complessità Ciclomatica	18
3.8.2.2	Numero di livelli di annidamento	19
3.8.2.3	Linee di commento	19
3.9	System/Software Qualification Testing Process	19
3.9.1	Obiettivi	19
3.9.2	Metriche	20
3.9.2.1	Test di Unità	20
3.9.2.2	Test di Integrazione	20
3.9.2.3	Test di Sistema	20
3.9.2.4	Test di Validazione	21
3.9.2.5	Test superati	21
3.10	Software Documentation Management Process	21
3.10.1	Obiettivi	21
3.10.2	Metriche	22
3.10.2.1	Indice Gulpease	22
3.11	Software Verification Process	22
3.11.1	Obiettivi	22
3.11.2	Metriche	22
3.11.2.1	Code Coverage	22
4	Test	24
4.1	Test di unità	24
4.1.1	Grado di completamento dei test di unità	34
4.2	Test di integrazione	35
4.2.1	Grado di completamento dei test di integrazione	36
4.3	Test di sistema	37

4.3.1	Grado di completamento dei test di sistema	40
4.4	Test di non regressione	42
4.4.1	Grado di completamento dei test di non regressione	42
4.5	Test di validazione	44
4.5.1	Grado di completamento dei test di validazione	49
A	Attività di verifica	51
A.1	Tracciamento	51
A.1.1	Tracciamento Test di Integrazione - Componenti	51
A.1.2	Tracciamento Componenti - Test di Integrazione	52
A.1.3	Tracciamento Test di Sistema - Requisiti	53
A.1.4	Tracciamento Requisiti - Test di Sistema	55
A.1.5	Tracciamento Test di Validazione - Requisiti	57
A.2	Resoconto verifica documenti	58
A.3	Revisione dei Requisiti	58
A.4	Revisione di Progettazione	59
A.5	Revisione di Qualifica	60
A.6	Revisione di Accettazione	61
A.7	Riepilogo	61
B	Resoconto Qualità di Prodotto	66
B.1	Revisione di Qualifica	66
B.2	Revisione di Accettazione	66
C	Resoconto Qualità di Processo	68
C.1	Revisione di Qualifica	68
C.1.1	Numero di metodi per classe	68
C.1.2	Numero di parametri per metodo	69
C.1.3	Numero di attributi per classe	69
C.2	Revisione di Accettazione	69
C.2.1	Fan In	70
C.2.2	Fan Out	70
C.2.3	Numero di metodi per classe	70
C.2.4	Numero di parametri per metodo	70
C.2.5	Numero di attributi per classe	71
C.2.6	Complessità Ciclomatica	71
C.2.7	Numero di livelli di annidamento	71

1 Introduzione

1.1 Scopo del documento

Questo documento descrive le scelte e le strategie attuate per permettere di raggiungere determinati obiettivi di qualità misurabili. A questo scopo, sarà necessario un continuo processo di verifica, orientato ad individuare e correggere errori ed eventuali sprechi di risorse. Per conseguire dei risultati concreti, il processo di verifica dovrà fornire dei dati quantificabili per poter valutare se gli obiettivi sono stati raggiunti o meno. Per facilitarne la valutazione, per ogni metrica saranno indicati due range:

- **Range accettazione:** rappresenta l'intervallo di valori minimi richiesti per il raggiungimento degli obiettivi di qualità definiti;
- **Range ottimale:** rappresenta l'intervallo di valori desiderati, entro cui dovrebbe collocarsi la misurazione. Nel caso in cui non si rientrasse in questo range, sarà necessario effettuare una verifica più accurata, al fine di individuarne le cause e poter applicare le dovute correzioni.

1.2 Scopo del prodotto

Lo scopo del prodotto è la realizzazione di un *API Market_G* per l'acquisto e la vendita di *microservizi_G*. Il sistema offrirà la possibilità di registrare nuove *API_G* per la vendita, permetterà la consultazione e la ricerca di API ai potenziali acquirenti, gestendo i permessi di accesso ed utilizzo tramite creazione e controllo di relative *API key_G*. Il sistema, oltre alla web app stessa, sarà corredato di un *API Gateway_G* per la gestione delle richieste e il controllo delle chiavi, e fornirà funzionalità avanzate di statistiche per il gestore della piattaforma e per i fornitori dei microservizi.

1.3 Riferimenti normativi

- NORMEDIPROGETTO 4_0_0.PDF;
- **Capitolato d'appalto C1:** APIM: An API Market Platform
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C1.pdf>;

1.4 Riferimenti informativi

- PIANODIPROGETTO 4_0_0.PDF;
- **Slide del corso riguardo la qualità di prodotto**
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L10.pdf>;
- **Slide del corso riguardo la qualità di processo**
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L11.pdf>;
- **Standard ISO/IEC 12207:2008**
<https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en>;
- **Standard ISO 9001**
<https://www.iso.org/iso-9001-quality-management.html>;
- **Standard ISO/IEC 9126:2001**
https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Standard ISO/IEC 15504**
https://en.wikipedia.org/wiki/ISO/IEC_15504;

- **Indice Gulpease**

https://it.wikipedia.org/wiki/Indice_Gulpease;

1.5 Glossario

Per semplificare la consultazione e disambiguare alcune terminologie tecniche, le voci indicate con la lettera *G* a pedice sono descritte approfonditamente nel documento GLOSSARIO 3_0_0.PDF e specificate solo alla prima occorrenza all'interno del suddetto documento.

2 Qualità di prodotto

Al fine di garantire una buona qualità di prodotto, il *team_G* ha individuato dallo standard *ISO/IEC 9126_G* le qualità che ritiene più importanti durante tutto il ciclo di vita del prodotto API Market. Per ognuna delle qualità individuate, sono stati definiti obiettivi e metriche coerenti con i livelli di qualità dichiarati.

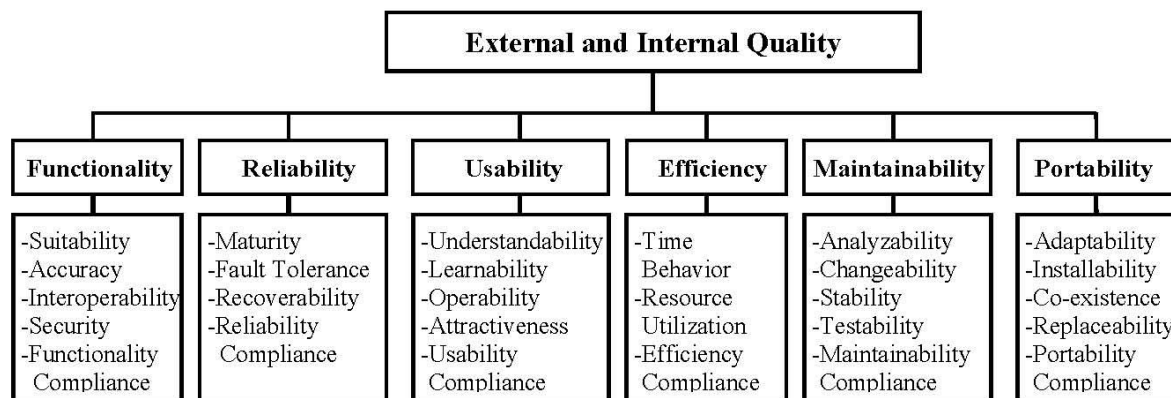


Figura 1: Standard ISO/IEC 9126

Le **metriche interne** si applicano al software non eseguibile, come ad esempio le specifiche tecniche e il codice sorgente, durante i periodi di progettazione e codifica. Esse sono specificate nella norma *ISO/IEC 9126-3_G*.

Durante le fasi di sviluppo del software, i prodotti intermedi sono valutati tramite metriche interne che misurano le proprietà intrinseche del prodotto. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale, in quanto gli attributi interni influenzano le caratteristiche esterne e quelle in uso. Le metriche interne misurano attributi interni del software e forniscono indicazioni sulle caratteristiche esterne del prodotto finale, tramite l'analisi statica dei prodotti intermedi. Le metriche interne si applicano anche alla documentazione del prodotto.

Le **metriche esterne** misurano i comportamenti del prodotto software rilevabili dai test, dall'operatività e dall'osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti. Esse sono specificate nella norma *ISO/IEC 9126-2_G*.

Le metriche per la qualità di prodotto saranno identificate con la seguente sintassi:

$$\text{MPD_}[Target][[Id]$$

dove:

- **Target** rappresenta la qualità di interesse della metrica e può assumere i valori:
 - **F** (Funzionalità);
 - **A** (Affidabilità);
 - **U** (Usabilità);
 - **E** (Efficienza);
 - **M** (Manutenibilità);
 - **P** (Portabilità);
- **Id** rappresenta il codice identificativo progressivo della metrica riguardante la qualità *Target*. Questo indice viene fatto partire da 1 per ogni differente qualità.

2.1 Definizione degli obiettivi di qualità

2.1.1 Funzionalità

Rappresenta la capacità del prodotto nel fornire le funzionalità richieste e soddisfare tutti i requisiti descritti nel documento ANALISIDEREQUISITI 3_0_0.PDF.

2.1.1.1 Obiettivi

- **Adeguatezza:** rappresenta la capacità di fornire un appropriato insieme di funzionalità che permettano agli utenti di svolgere determinati task e raggiungere gli obiettivi prefissati.
- **Accuratezza:** rappresenta la capacità di fornire i risultati e gli effetti attesi con il livello di precisione richiesta.
- **Sicurezza:** rappresenta la capacità di proteggere le informazioni ed i dati, in modo che persone o sistemi non autorizzati non possano accedervi.

2.1.1.2 Metriche

2.1.1.2.1 Completezza delle funzioni sviluppate

Id Metrica: MPD_F1

Descrizione: indica la percentuale di funzionalità sviluppate ritenute complete.

- $\text{fun}_{\text{Compl}}$ = numero di funzioni ritenute complete
- fun_{Tot} = numero di funzioni totali

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{fun}_{\text{Compl}}}{\text{fun}_{\text{Tot}}} * 100$	[90,100] in [0,100]	[90,100] in [0,100]

Tabella 1: Completezza delle funzioni sviluppate

2.1.1.2.2 Correttezza delle funzioni sviluppate

Id Metrica: MPD_F2

Descrizione: indica la percentuale di funzionalità sviluppate ritenute corrette.

- fun_{Corr} = numero di funzioni ritenute corrette
- fun_{Tot} = numero di funzioni totali

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{fun}_{\text{Corr}}}{\text{fun}_{\text{Tot}}} * 100$	100 in [0,100]	100 in [0,100]

Tabella 2: Correttezza delle funzioni sviluppate

2.1.1.2.3 Accuratezza rispetto alle aspettative

Id Metrica: MPD_F3

Descrizione: indica la percentuale di risultati conformi alle aspettative.

- $\text{test}_{\text{Corr}}$ = numero di test ritenuti corretti
- test_{Tot} = numero di test totali

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{test}_{\text{Corr}}}{\text{test}_{\text{Tot}}} * 100$	[90,100] in [0,100]	[95,100] in [0,100]

Tabella 3: Accuratezza rispetto alle aspettative

2.1.1.2.4 Controllo degli accessi

Id Metrica: MPD_F4

Descrizione: indica la percentuale di accessi corretti al sistema.

- $\text{accessi}_{\text{Succ}}$ = numero di accessi controllati con successo dal sistema
- $\text{accessi}_{\text{Tot}}$ = numero di accessi totali

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{accessi}_{\text{Succ}}}{\text{accessi}_{\text{Tot}}} * 100$	[90,100] in [0,100]	100 in [0,100]

Tabella 4: Controllo degli accessi

2.1.2 Affidabilità

Rappresenta la capacità del prodotto software di mantenere il livello di prestazione, quando viene utilizzato in condizioni specificate.

2.1.2.1 Obiettivi

- **Maturità:** rappresenta la capacità di evitare che si verifichino errori o siano prodotti risultati non corretti in fase di esecuzione.
- **Tolleranza agli errori:** rappresenta la capacità di mantenere il livello di prestazioni in caso di errori nel software o di violazione nelle interfacce specificate.

2.1.2.2 Metriche

2.1.2.2.1 Chiamate a microservizi corrette

Id Metrica: MPD_A1

Descrizione: indica il numero di chiamate al microservizio j andate a buon fine.

- $\text{num}_{\text{CF}}(\text{MS}(j))$ = numero di chiamate al microservizio j fallite o avvenute con successo, ma con $t_{\text{R}}(\text{MS}(j)) > t_{\text{MR}}(\text{MS}(j)) + 0.3 * t_{\text{MR}}(\text{MS}(j))$, dove $t_{\text{R}}(\text{MS}(j))$ = tempo di risposta del microservizio j e $t_{\text{MR}}(\text{MS}(j))$ = tempo medio di risposta del microservizio j

- $\text{num}_{\text{CT}}(\text{MS}(j))$ = numero totale di chiamate al microservizio j

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{CF}}(\text{MS}(j))}{\text{num}_{\text{CT}}(\text{MS}(j))} * 100$	[90,100] in [0,100]	100 in [0,100]

Tabella 5: Chiamate a microservizi corrette

2.1.2.2.2 Copertura dei test

Id Metrica: MPD_A2

Descrizione: indica il livello di copertura dei test.

- test_P = numero di test pianificati
- test_N = numero di test necessari a garantire la copertura richiesta o massima

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{test}_P}{\text{test}_N} * 100$	[80,100] in [0,100]	100 in [0,100]

Tabella 6: Copertura dei test

2.1.2.2.3 Controllo dei guasti

Id Metrica: MPD_A3

Descrizione: indica il livello di controllo dei guasti, attraverso il numero di condizioni di errore messe sotto controllo per evitare malfunzionamenti e/o guasti al prodotto.

- num_{CEG} = numero di condizioni d'errore gestite correttamente
- num_{CEP} = numero di condizioni d'errore possibili nel sistema

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{CEG}}}{\text{num}_{\text{CEP}}} * 100$	[80,100] in [0,100]	100 in [0,100]

Tabella 7: Controllo dei guasti

2.1.3 Usabilità

Rappresenta la capacità di un prodotto software di essere comprensibile, di poter essere studiato e di risultare attraente da parte di un utente sotto determinate condizioni.

2.1.3.1 Obiettivi

- **Comprensibilità:** rappresenta la capacità di permettere all'utente di capire le funzionalità del prodotto software e come poterle utilizzare con successo per svolgere particolari task in determinate condizioni di utilizzo.

- **Operabilità:** rappresenta la capacità di permettere all'utente di utilizzare e controllare il prodotto software.
- **Attrattività:** rappresenta la capacità di risultare piacevole per l'utente.

2.1.3.2 Metriche

2.1.3.2.1 Comprensibilità delle funzionalità offerte

Id Metrica: MPD_U1

Descrizione: indica la percentuale di funzioni comprensibili agli utenti.

- num_{FC} = numero di funzionalità comprensibili agli utenti
- num_{FT} = numero di funzionalità totali offerte

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{FC}}}{\text{num}_{\text{FT}}} * 100$	[80,100] in [0,100]	[90,100] in [0,100]

Tabella 8: Comprensibilità delle funzionalità offerte

2.1.3.2.2 Controllo e monitoraggio delle operazioni

Id Metrica: MPD_U2

Descrizione: indica la capacità del prodotto di monitorare lo stato delle operazioni eseguite.

- num_{FC} = numero di funzionalità con adeguato controllo e monitoraggio delle operazioni
- num_{FCT} = numero di funzionalità con controllo totali

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{FC}}}{\text{num}_{\text{FCT}}} * 100$	[80,100] in [0,100]	[90,100] in [0,100]

Tabella 9: Controllo e monitoraggio delle operazioni

2.1.3.2.3 Qualità della messaggistica

Id Metrica: MPD_U3

Descrizione: indica il grado di chiarezza, completezza e correttezza dei messaggi previsti rispetto alle diverse condizioni gestite dal prodotto (ad esempio, il completamento di una funzione, le condizioni di errore, le scelte da effettuare, etc.).

- num_{MC} = numero di messaggi che risultano chiari, completi e corretti
- num_{MT} = numero totale di messaggi previsti

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{MC}}{num_{MT}} * 100$	[70,100] in [0,100]	100 in [0,100]

Tabella 10: Qualità della messaggistica

2.1.4 Efficienza

Rappresenta la capacità di un prodotto software di realizzare le funzioni richieste nel minor tempo possibile ed utilizzando nel miglior modo le risorse necessarie.

2.1.4.1 Obiettivi

- **Comportamento rispetto al tempo:** rappresenta la capacità di fornire appropriati tempi di risposta, tempi di elaborazione e quantità di lavoro eseguendo le funzionalità previste.
- **Utilizzo delle risorse:** rappresenta la capacità di utilizzare un appropriato numero e tipo di risposte quando esegue le funzionalità previste.

2.1.4.2 Metriche

2.1.4.2.1 Tempo di risposta

Id Metrica: MPD_E1

Descrizione: indica il tempo medio che intercorre fra la richiesta di una determinata funzionalità e la restituzione del risultato all'utente. L'unità di misura scelta per esprimere il tempo è il *secondo*.

- t_i = tempo trascorso fra la richiesta i di una funzionalità ed il completamento delle operazioni necessarie alla restituzione del risultato alla richiesta i

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\sum_{i=1}^n (t_i)}{n}$	[0,10]	[0,4]

Tabella 11: Tempo di risposta

2.1.5 Manutenibilità

Rappresenta la capacità di un prodotto software di essere modificato. Le modifiche possono includere correzioni o adattamenti del software a modifiche negli ambienti, nei requisiti e nelle specifiche funzionali.

2.1.5.1 Obiettivi

- **Analizzabilità:** rappresenta la capacità di poter effettuare la diagnosi sul software ed individuare le cause di errori o malfunzionamenti.
- **Modificabilità:** rappresenta la capacità di consentire lo sviluppo di modifiche al codice, alla progettazione e alla documentazione.

- **Stabilità:** rappresenta la capacità di evitare effetti non desiderati a seguito di modifiche al software.
- **Testabilità:** rappresenta la capacità di consentire la verifica e la validazione del software modificato, cioè di eseguire test.

2.1.5.2 Metriche

2.1.5.2.1 Impatto delle modifiche

Id Metrica: MPD_M1

Descrizione: indica la percentuale di modifiche effettuate in risposta a failure, le quali hanno portato all'introduzione di nuove failure in altre componenti del sistema.

- num_{FRNF} = numero di failure risolte con l'introduzione di nuove failure
- num_{FR} = numero di failure risolte

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{FRNF}}}{\text{num}_{\text{FR}}} * 100$	[0,25] in [0,100]	[0,10] in [0,100]

Tabella 12: Impatto delle modifiche

2.1.6 Portabilità

Rappresenta la capacità di un prodotto software di poter essere trasportato da un ambiente ad un altro.

2.1.6.1 Obiettivi

- **Adattabilità:** rappresenta la capacità del prodotto di essere adattato a differenti ambienti, senza richiedere azioni specifiche diverse da quelle previste dal software per tali attività.

2.1.6.2 Metriche

2.1.6.2.1 Supporto differenti versioni dei browser

Id Metrica: MPD_P1

Descrizione: indica la percentuale di versioni dei browser supportate.

- $\text{num}_{\text{VSupp}}$ = numero delle versioni dei browser supportati
- num_{VTot} = numero totale delle versioni che devono essere supportate

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{VSupp}}}{\text{num}_{\text{VTot}}} * 100$	[90,100] in [0,100]	100 in [0,100]

Tabella 13: Supporto differenti versioni dei browser

3 Qualità di processo

Per garantire la qualità del prodotto finale è necessario garantire la qualità dei processi necessari al suo completamento. A questo scopo, si è deciso di adottare lo standard *ISO/IEC 15504G*, denominato *SPICEG*. Esso si fonda sul principio che ogni processo deve essere controllato costantemente, in modo tale da rilevare possibili errori o debolezze, e correggerli prima che essi si diffondano, provocando un aumento del carico di lavoro e dello spreco di risorse. SPICE definisce sei livelli di maturità del processo:

- **Level 0 - Incomplete:** processo non ancora implementato o incapace di raggiungere i suoi obiettivi;
- **Level 1 - Performed:** processo messo in atto e capace di raggiungere i suoi obiettivi. Viene misurato tramite:
 - *Process performance:* capacità di raggiungere i propri obiettivi e di ottenere risultati identificabili.
- **Level 2 - Managed:** processo eseguito sulla base di obiettivi ben definiti. Viene misurato tramite:
 - *Performance management:* capacità di elaborare un prodotto coerente con gli obiettivi attesi;
 - *Work product management:* capacità di elaborare un prodotto documentato, controllato e verificato in modo appropriato.
- **Level 3 - Established:** processo eseguito in base ai principi dell'ingegneria del software. Viene misurato tramite:
 - *Process definition:* capacità di raggiungere i propri obiettivi aderendo agli standard preposti;
 - *Process deployment:* capacità di sfruttare risorse adeguate che gli permettano di essere attuato in maniera efficace.
- **Level 4 - Predictable:** processo attuato all'interno di limiti ben definiti. Viene misurato tramite:
 - *Process measurement:* capacità di utilizzare i risultati raggiunti e le misure ricavate durante l'esecuzione per garantire il raggiungimento degli obiettivi definiti;
 - *Process control:* capacità di correggere e/o migliorare, se necessario, le sue modalità di esecuzione, in seguito a controlli basati sulle misurazioni rilevate.
- **Level 5 - Optimizing:** processo predicibile e capace di adattarsi per raggiungere obiettivi specifici e rilevanti. Viene misurato tramite:
 - *Process innovation:* capacità di tenere sotto controllo tutti i cambiamenti strutturali e di esecuzione;
 - *Process optimization:* capacità di identificare e implementare le modifiche effettuate, per garantire un miglioramento continuo nella realizzazione degli obiettivi fissati.

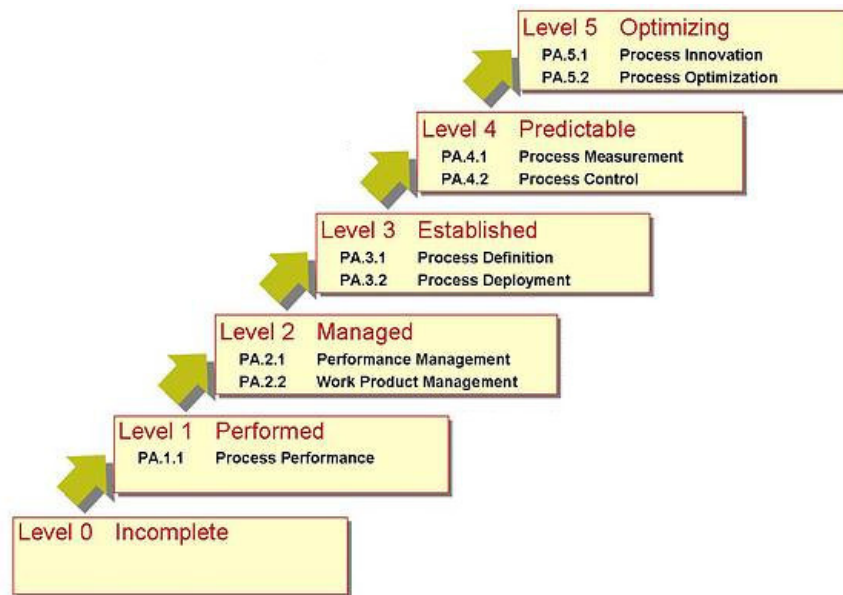


Figura 2: Standard ISO/IEC 15504 (SPICE)

Al fine di perseguire correttamente questo modello, è necessario adottare il principio *PDCA_G*, il quale si compone delle seguenti quattro fasi:

- **Plan:** fase di pianificazione ed individuazione di obiettivi e processi, necessari allo scopo di raggiungere i risultati attesi;
- **Do:** fase di attuazione delle attività pianificate nella precedente fase Plan, e raccolta di dati sulla qualità ottenuta;
- **Check:** fase di verifica dove vengono confrontati i dati in uscita dalla fase Do con quelli pianificati nella fase Plan;
- **Act:** fase in cui si determinano le cause delle differenze fra risultati ottenuti e risultati attesi, in modo da individuare le azioni correttive da effettuare per ottenere un miglioramento della qualità.

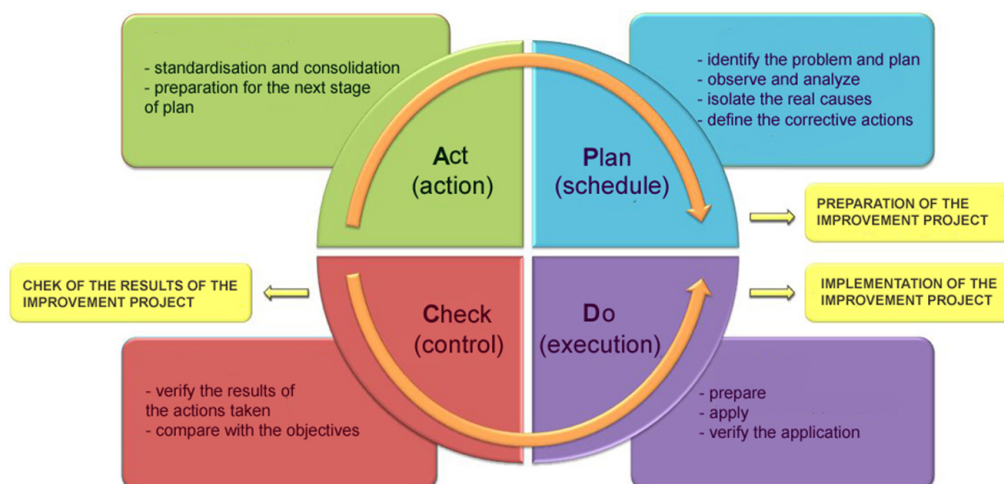


Figura 3: Principio PDCA

Infine, il team ha individuato dallo standard **ISO/IEC 12207:2008** i processi che ritiene più importanti durante tutto il ciclo di vita del prodotto, al fine di garantire una buona qualità di processo. Per ognuno di essi sono stati individuati obiettivi e metriche coerenti con i livelli di qualità perseguiti. Le metriche per la qualità di processo saranno identificate con la seguente sintassi:

$$\text{MPC}[Id]$$

dove **Id** rappresenta il codice identificativo progressivo della metrica di processo interessata.

3.1 Infrastructure Management Process

Questo processo ha lo scopo di fornire, mantenere ed aggiornare l'infrastruttura ed i servizi necessari alla realizzazione del progetto, durante tutto il suo ciclo di vita. Il termine infrastruttura comprende: elementi hardware, software, metodi, strumenti, tecniche e standard impiegati nello sviluppo del prodotto. Essa sarà necessaria allo svolgimento del progetto e dovrà essere mantenuta costantemente aggiornata. L'utilizzo delle metriche scelte, permetterà di individuare eventuali errori all'interno degli strumenti utilizzati, la cui correzione permetterà di produrre dati corretti e coerenti.

3.1.1 Obiettivi

Per tutta la durata del progetto, l'infrastruttura impiegata nello sviluppo dovrà raggiungere i seguenti obiettivi:

- ogni procedura riguardante le attività svolte più frequentemente durante lo sviluppo del progetto, sarà descritta nel documento NORMEDIPROGETTO 4_0_0.PDF;
- ogni riferimento normativo ed informativo sarà completo di informazioni utili al proprio riferimento;
- la piattaforma *NetBreakDB* sarà a disposizione di ogni componente del team, in caso di bisogno di accesso ai dati in essa contenuti;

3.1.2 Metriche

3.1.2.1 Disponibilità *NetBreakDB*

Id Metrica: MPC1

Descrizione: indica la percentuale di disponibilità di utilizzo della piattaforma *NetBreakDB*, rispetto alle richieste di accesso.

- **num_{AL}** = numero di accessi corretti alla pagina di login
- **num_{ALTot}** = numero totale delle richieste di accesso alla pagina di login inoltrate

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{AL}}{\text{num}_{ALTot}} * 100$	[80,100] in [0,100]	100 in [0,100]

Tabella 14: Disponibilità NetBreakDB

3.2 Project Planning, Assessment & Control Process

Questo processo ha lo scopo di produrre dei piani di sviluppo per il progetto, che comprendano:

- la scelta del modello di ciclo di vita del prodotto;
- le descrizioni delle attività e dei compiti da svolgere;
- la pianificazione temporale del lavoro e dei costi da sostenere;
- l'allocazione di compiti e responsabilità;
- le misurazioni per rilevare lo stato del progetto, rispetto alle pianificazioni stilate.

Durante tutta l'attività di progetto, sarà fondamentale mantenere aggiornata la pianificazione effettuata, in modo da essere sempre coerente con la situazione attuale. Nel caso in cui, in una fase di lavoro, venissero rilevati dei valori negativi, attraverso l'utilizzo delle metriche scelte Schedule Variance e Budget Variance, occorrerà compensare tali valori entro la fine dell'attività di progetto, al fine di evitare di eccedere le ore di lavoro totali e, di conseguenza, il preventivo dei costi finale indicato nella pianificazione contenuta nel documento PIANODIPROGETTO 4_0_0.PDF.

3.2.1 Obiettivi

L'intero sviluppo del progetto dovrà seguire la pianificazione prodotta:

- ogni componente dovrà svolgere e portare a termine l'attività assegnatagli, svolgendo tutti i compiti nei quali è stata suddivisa e facendo attenzione a rispettare le scadenze fissate;
- il costo necessario allo svolgimento di un'attività non dovrà eccedere la somma preventivata.

3.2.2 Metriche

3.2.2.1 Schedule Variance

Id Metrica: MPC2

Descrizione: indica se si è o meno in linea con la pianificazione temporale delle attività nella baseline. Se si ottiene un valore maggiore di 0, significa che il team è in anticipo, se si ottiene 0, significa che il team ha rispettato la pianificazione, altrimenti, in caso di valore negativo, significa che il team è in ritardo.

- att_{CompI} = attività completate ad un certo momento
- att_{PCompI} = attività che, secondo la pianificazione, dovrebbero essere state completate a quel momento

Metodo di calcolo	Range accettazione	Range ottimale
$att_{CompI} - att_{PCompI}$	≥ 0	≥ 0

Tabella 15: Schedule Variance

3.2.2.2 Budget Variance

Id Metrica: MPC3

Descrizione: indica se la spesa sostenuta alla data corrente è superiore o inferiore a quella preventivata in sede di pianificazione. Un valore positivo indica che si è speso meno di quanto inizialmente previsto, altrimenti significa che si è speso più della somma preventivata.

- **costo_{Plan}** = costo pianificato per realizzare le attività di progetto alla data corrente
- **costo_{Eff}** = costo effettivamente sostenuto alla data corrente

Metodo di calcolo	Range accettazione	Range ottimale
$costo_{Plan} - costo_{Eff}$	≥ 0	≥ 0

Tabella 16: Budget Variance

3.3 Risk Management Process

Questo processo ha l'obiettivo di identificare, analizzare, trattare e monitorare in modo continuo i rischi che possono insorgere durante l'intera attività di progetto. Il livello di probabilità dei rischi analizzati dovrà essere costantemente monitorato. Nel caso si manifestasse un rischio, a qualsiasi livello di pericolosità, il team dovrà attuare le contromisure previste, al fine di mitigarne gli effetti ed evitare un incremento del livello di pericolosità.

3.3.1 Obiettivi

Il team dovrà gestire correttamente i rischi:

- all'inizio dell'attività di progetto, verranno individuati i principali fattori di rischio riguardanti l'organizzazione delle attività;
- all'inizio di ogni attività, l'analisi dei rischi potrà portare all'individuazione di nuovi specifici rischi per ognuna di esse;
- i rischi analizzati che si manifesteranno, verranno trattati secondo le strategie descritte, così da controllarne l'impatto.

3.3.2 Metriche

3.3.2.1 Rischi non preventivati

Id Metrica: MPC4

Descrizione: indicatore che evidenzia i rischi non preventivati.

- **risk_{NP}** = contatore che viene incrementato nel momento in cui si manifesta un rischio non individuato nell'attività di analisi dei rischi

Metodo di calcolo	Range accettazione	Range ottimale
-------------------	--------------------	----------------

risk _{NP}	[0,5]	0
--------------------	-------	---

Tabella 17: Rischi non preventivati

3.4 System/Software Requirements Analysis Process

Questo processo ha lo scopo di creare un insieme di requisiti tecnici, a partire dall'insieme di requisiti individuati dalle fonti, in modo che diventi la linea guida nella progettazione del prodotto. Ogni requisito individuato dovrà essere inserito correttamente nella piattaforma *NetBreakDB*, la quale si occuperà di effettuare il tracciamento delle fonti dalle quali derivano i requisiti, delle modifiche effettuate e della loro implementazione nel prodotto.

3.4.1 Obiettivi

I requisiti identificati dovranno essere gestiti in modo da raggiungere i seguenti obiettivi:

- per ogni requisito dovrà essere possibile indicare dei test, da effettuare per verificarne il soddisfacimento da parte del prodotto;
- nessun requisito dovrà risultare ambiguo;
- tutti i requisiti che il prodotto andrà a soddisfare, saranno stati precedentemente approvati dal proponente.

3.4.2 Metriche

3.4.2.1 Adempimento requisiti obbligatori

Id Metrica: MPC5

Descrizione: indica la percentuale di requisiti obbligatori soddisfatti dal prodotto.

- **num_{ROS}** = numero di requisiti obbligatori soddisfatti
- **num_{ROI}** = numero di requisiti obbligatori identificati

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{ROS}}{num_{ROI}} * 100$	100 in [0,100]	100 in [0,100]

Tabella 18: Adempimento requisiti obbligatori

3.5 System/Software Architectural Design Process

Il processo si pone come obiettivo quello di identificare una corrispondenza fra requisiti di sistema ed elementi del sistema. Nel corso dell'attività di Progettazione, sia ad alto livello che di dettaglio, le componenti verranno inserite nella piattaforma *NetBreakDB*. Essa si occuperà di effettuare i tracciamenti fra le componenti e i requisiti che soddisfano, ed inoltre, il tracciamento tra le relazioni presenti e le varie componenti.

3.5.1 Obiettivi

Durante lo svolgimento delle attività previste da questo processo, il team punterà a definire un'architettura adatta agli scopi del progetto:

- ogni componente progettato come parte del sistema risulterà essere necessario per il funzionamento del prodotto e, quindi, costantemente tracciabile ai requisiti che soddisfa;
- il sistema dovrà presentare basso accoppiamento ed alta coesione;
- ogni componente dovrà essere progettato puntando su incapsulamento, modularizzazione e riuso di codice.

3.5.2 Metriche

3.5.2.1 Fan In

Id Metrica: MPC6

Descrizione: in riferimento ad un modulo software, misura quanti altri moduli lo utilizzano durante la loro esecuzione. Tale indicazione consente di stabilire il livello di riuso implementato.

- **FI** = contatore che viene incrementato nel momento in cui viene individuato un modulo che, durante la sua esecuzione, chiama il modulo in oggetto

Metodo di calcolo	Range accettazione	Range ottimale
FI	≥ 0	≥ 3

Tabella 19: Fan In

3.5.2.2 Fan Out

Id Metrica: MPC7

Descrizione: in riferimento ad un modulo software, misura quanti moduli vengono utilizzati durante la sua esecuzione. Tale indicazione consente di stabilire il livello di accoppiamento implementato.

- **FO** = contatore che viene incrementato nel momento in cui viene individuato un modulo utilizzato dal modulo in oggetto, durante la sua esecuzione

Metodo di calcolo	Range accettazione	Range ottimale
FO	[0,5]	[0,1]

Tabella 20: Fan Out

3.6 Software Architectural Design Process

Lo scopo del processo è fornire una progettazione di minimo del prodotto che andrà a soddisfare i requisiti individuati. Tutte le attività presenti in questo processo portano alla produzione del documento SPECIFICATECNICA 3_0_0.PDF.

3.6.1 Obiettivi

- fornire un'architettura ad alto livello del prodotto software, in grado di soddisfare i requisiti individuati;
- definire le interfacce interne ed esterne per ogni componente individuato;
- progettare uno schema ad alto livello del database.

3.7 Software Detailed Design Process

Lo scopo del processo è fornire una progettazione dettagliata del prodotto che andrà ad implementare i requisiti individuati. Sarà necessario effettuare un'analisi dettagliata delle componenti individuate durante l'attività di progettazione architetturale, suddividendole in unità che siano facilmente codificabili e testabili.

3.7.1 Obiettivi

Le attività svolte dovranno raggiungere i seguenti obiettivi:

- il livello di dettaglio della progettazione dovrà indicare tutti i metodi, con i relativi parametri e campi dati, forniti da ciascuna classe;
- la struttura a basso livello dell'architettura e le relazioni fra le varie unità software concepite saranno esposte nel documento DEFINIZIONE DI PRODOTTO 2_0_0.PDF, il quale definirà esattamente cosa implementare;

3.7.2 Metriche

3.7.2.1 Numero di metodi per classe

Id Metrica: MPC8

Descrizione: indica il numero di metodi definiti in una classe. Un valore molto alto potrebbe indicare una non buona decomposizione delle funzionalità a livello logico.

- $\text{num}_{\text{MetCl}}$ = contatore che indica il numero di metodi definiti in una classe

Metodo di calcolo	Range accettazione	Range ottimale
$\text{num}_{\text{MetCl}}$	[1,15]	[1,7]

Tabella 21: Numero di metodi per classe

3.7.2.2 Numero di parametri per metodo

Id Metrica: MPC9

Descrizione: indica il numero di parametri passati ad un metodo. Un valore molto alto potrebbe indicare un'eccessiva complessità del metodo, il quale potrebbe non essere sufficientemente scomposto in sotto-metodi.

- num_{Par} = contatore che indica il numero di parametri passati ad un metodo

Metodo di calcolo	Range accettazione	Range ottimale
-------------------	--------------------	----------------

num _{Par}	[0,8]	[0,4]
--------------------	-------	-------

Tabella 22: Numero parametri per metodo

3.7.2.3 Numero di attributi per classe

Id Metrica: MPC10

Descrizione: indica il numero di attributi di una classe. Un valore molto alto potrebbe suggerire la necessità di suddividere tale classe in più classi differenti correlate tra loro.

- **num_{AttrCl}** = contatore che indica il numero di attributi per una determinata classe

Metodo di calcolo	Range accettazione	Range ottimale
num _{AttrCl}	[0,14]	[2,8]

Tabella 23: Numero di attributi per classe

3.8 Software Construction Process

Questo processo definisce le principali attività volte alla produzione di unità software eseguibili, che rispettino quanto prodotto durante la progettazione. Nell'attività di Codifica, il *Programmatore* dovrà semplicemente attenersi a quanto indicato nel documento DEFINIZIONE DI PRODOTTO 2_0_0.PDF. Inoltre, sarà necessario procedere con la codifica dei test individuati in sede di progettazione, al fine di verificare il corretto funzionamento delle varie unità prodotte.

3.8.1 Obiettivi

Affinchè le unità software prodotte risultino di qualità, il team ha individuato i seguenti obiettivi:

- l'implementazione delle classi e dei metodi definiti in progettazione dovrà produrre codice a bassa complessità, in modo da facilitare l'attività di test;
- l'uso di costrutti e tecniche che creano sdoppiamenti del flusso di esecuzione verrà attuato solo se strettamente necessario;
- il codice prodotto dovrà risultare facilmente manutenibile;
- il codice prodotto risulterà privo di elementi inutilizzati.

3.8.2 Metriche

3.8.2.1 Complessità Ciclomatica

Id Metrica: MPC11

Descrizione: indica la complessità di funzioni, moduli, metodi o classi di un programma, misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. Alti valori di complessità ciclomatica implicano una ridotta manutenibilità del codice; viceversa, bassi valori potrebbero determinare una scarsa efficienza dei metodi.

- **num_{Archi}** = numero di archi del grafo di controllo di flusso
- **num_{Nodi}** = numero di nodi del grafo di controllo di flusso

Metodo di calcolo	Range accettazione	Range ottimale
$\text{num}_{\text{Archi}} - \text{num}_{\text{Nodi}} + 2$	[3,12]	[1,10]

Tabella 24: Complessità Ciclomatica

3.8.2.2 Numero di livelli di annidamento

Id Metrica: MPC12

Descrizione: indica il numero di funzioni o procedure chiamate all'interno di un metodo. Un valore elevato implica un'alta complessità ed un basso livello di astrazione del codice.

- **nesting** = contatore che indica il numero di chiamate a funzioni o procedure presenti all'interno di un metodo

Metodo di calcolo	Range accettazione	Range ottimale
nesting	[1,8]	[1,4]

Tabella 25: Numero di livelli di annidamento

3.8.2.3 Linee di commento

Id Metrica: MPC13

Descrizione: indica la percentuale di linee di commento presenti all'interno del codice sorgente; la loro presenza permette una più semplice comprensione ed un maggior livello di manutenibilità di quanto prodotto.

- **num_{LComm}** = numero di linee di commento presenti nel codice
- **num_{SLOC}** = numero di Source Line Of Code totali prodotte

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{LComm}}}{\text{num}_{\text{SLOC}}} * 100$	[20,100] in [0,100]	[35,100] in [0,100]

Tabella 26: Linee di commento

3.9 System/Software Qualification Testing Process

Lo scopo del processo è quello di assicurare che ogni requisito individuato sia stato implementato nel prodotto. Sarà indispensabile rendere il più possibile automatizzata l'esecuzione dei test di sistema, in modo tale che la loro esecuzione non richieda costi e tempi eccessivi, e allo stesso tempo sia possibile eseguirne un numero sufficiente a garantire un'ottima copertura dei requisiti.

3.9.1 Obiettivi

Durante lo svolgimento delle attività di test, il team dovrà perseguire i seguenti obiettivi:

- le attività di test previste dal processo verranno svolte su un sistema le cui componenti sono verificate e correttamente integrate fra loro;

- il sistema dovrà implementare e soddisfare tutti i requisiti obbligatori individuati durante l'attività di analisi dei requisiti.

3.9.2 Metriche

3.9.2.1 Test di Unità

Id Metrica: MPC14

Descrizione: indica la percentuale di test di unità eseguiti.

- **num_{TUE}** = numero di test di unità eseguiti
- **num_{TUP}** = numero di test di unità pianificati

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{TUE}}{num_{TUP}} * 100$	[95,100] in [0,100]	100 in [0,100]

Tabella 27: Test di Unità

3.9.2.2 Test di Integrazione

Id Metrica: MPC15

Descrizione: indica la percentuale di test di integrazione eseguiti.

- **num_{TIE}** = numero di test di integrazione eseguiti
- **num_{TIP}** = numero di test di integrazione pianificati

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{TIE}}{num_{TIP}} * 100$	[65,100] in [0, 100]	[75,100] in [0,100]

Tabella 28: Test di Integrazione

3.9.2.3 Test di Sistema

Id Metrica: MPC16

Descrizione: indica la percentuale di test di sistema eseguiti.

- **num_{TSE}** = numero di test di sistema eseguiti
- **num_{TSP}** = numero di test di sistema pianificati

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{TSE}}{num_{TSP}} * 100$	[75,100] in [0,100]	[85,100] in [0,100]

Tabella 29: Test di Sistema

3.9.2.4 Test di Validazione

Id Metrica: MPC17

Descrizione: indica la percentuale di test di validazione eseguiti.

- **num_{TVE}** = numero di test di validazione eseguiti
- **num_{TVP}** = numero di test di validazione pianificati

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{TVE}}{num_{TVP}} * 100$	[95,100] in [0,100]	100 in [0,100]

Tabella 30: Test di Validazione

3.9.2.5 Test superati

Id Metrica: MPC18

Descrizione: indica la percentuale di test totali superati.

- **num_{TS}** = numero di test superati
- **num_{TE}** = numero di test eseguiti

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{num_{TS}}{num_{TE}} * 100$	[90,100] in [0,100]	100 in [0,100]

Tabella 31: Test superati

3.10 Software Documentation Management Process

Questo processo ha l'obiettivo di produrre e mantenere le informazioni sul software prodotte dai processi attuati, attraverso un'opportuna documentazione. Ogni documento sarà dotato di:

- numero di versione;
- changelog.

Queste informazioni consentono di tenere traccia di ogni azione effettuata sul documento in oggetto.

3.10.1 Obiettivi

Il processo di documentazione dovrà perseguire le seguenti direttive:

- la documentazione prodotta dovrà essere chiara e comprensibile a tutti gli stakeholder e sarà resa disponibile alle parti interessate per la consultazione;
- ogni forma di ambiguità sul significato di un termine utilizzato verrà eliminata grazie al documento GLOSSARIO 3_0_0.PDF;
- la documentazione prodotta sarà sempre aggiornata ed allineata allo stato attuale del processo di sviluppo del prodotto.

3.10.2 Metriche

3.10.2.1 Indice Gulpease

Id Metrica: MPC19

Descrizione: l'indice Gulpease è un indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri indici, questo ha il vantaggio di utilizzare la lunghezza delle parole in lettere, anziché in sillabe, semplificandone il calcolo automatico. L'indice Gulpease considera due variabili linguistiche:

- la lunghezza della parola;
- la lunghezza della frase, rispetto al numero delle lettere.

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

- $\text{num}_{\text{Frase}}$ = numero di frasi
- $\text{num}_{\text{Lettere}}$ = numero di lettere
- $\text{num}_{\text{Parole}}$ = numero di parole

Metodo di calcolo	Range accettazione	Range ottimale
$89 + \frac{300 * \text{num}_{\text{Frase}} - 10 * \text{num}_{\text{Lettere}}}{\text{num}_{\text{Parole}}}$	[40,100] in [0,100]	[60,100] in [0,100]

Tabella 32: Indice Gulpease

3.11 Software Verification Process

Questo processo ha lo scopo di verificare se un qualsiasi elemento del sistema soddisfa in modo esaustivo i requisiti ad esso associati. Durante le varie attività di revisione della documentazione, gli errori più frequenti rilevati verranno riportati in un documento apposito, in modo tale da velocizzare le successive attività di *Inspection*. Inoltre, per ogni test effettuato verrà tenuto il tracciamento del suo esito.

3.11.1 Obiettivi

Al fine di garantire qualità durante questo processo, il team ha fissato i seguenti obiettivi:

- la documentazione verrà verificata attraverso la tecnica *Inspection*, poiché permette di risparmiare tempo e costi;
- i test dinamici effettuati sui vari elementi saranno automatizzati il più possibile;
- i test dinamici effettuati sui vari elementi del software copriranno una grande parte dei possibili casi d'uso.

3.11.2 Metriche

3.11.2.1 Code Coverage

Id Metrica: MPC20

Descrizione: indica la percentuale di istruzioni che sono eseguite durante i test. Un alto valore di questo indice implica un'alta probabilità che le componenti testate abbiano una ridotta quantità

di errori. Per ottenere un indice basso è necessario che l'implementazione dei metodi sia molto semplice, in modo da non richiedere alcuna attività di testing. Ad esempio, i metodi getter e setter.

- num_{LM} = numero di linee di codice monitorato dai test
- num_{LI} = numero di linee di codice implementate nel software

Metodo di calcolo	Range accettazione	Range ottimale
$\frac{\text{num}_{\text{LM}}}{\text{num}_{\text{LI}}} * 100$	[50,100] in [0,100]	[75,100] in [0,100]

Tabella 33: Code Coverage

4 Test

Al fine di produrre del software di qualità, il team ha strutturato dei test volti a verificare l'efficacia del software prodotto, ovvero che quest'ultimo rispecchi le funzionalità richieste. Tutte le attività di testing prodotte devono poter essere ripetibili e deterministiche, al fine di poter fornire informazioni utili a intraprendere azioni correttive, nel caso si ottengano dei risultati diversi da quelli attesi. Per avere un tracciamento dei test prodotti e dei risultati ottenuti, si è scelto di rappresentare delle tabelle di log di facile consultazione, le quali forniscono un'indicazione degli output delle attività di verifica, eventuali errori e/o risultati non coerenti con quanto fissato.

4.1 Test di unità

Questa tipologia di test serve a verificare ogni singola unità del prodotto software; per unità, si intende la più piccola quantità di software che è utile verificare singolarmente e che viene prodotta da un singolo *Programmatore*.

Tramite questi test si verifica il corretto funzionamento dei moduli che compongono l'intero sistema, in modo da eliminare possibili errori di implementazione da parte dei *Programmatori*. I test di unità saranno descritti nel modo seguente:

$$TU[IdTest]$$

dove **IdTest** rappresenta il codice identificativo progressivo dell'unità presa in considerazione.

Id Test	Descrizione	Stato
TU1	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al corretto Controller o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU2	Verificare che il messaggio d'errore venga costruito in modo coerente rispetto ai dati passati al costruttore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU3	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, verificando che l'utente che esegue la richiesta sia effettivamente un utente autenticato e mantenendo aggiornate le informazioni riguardanti l'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU4	Verificare che, in base ai parametri forniti in input, il metodo mantenga aggiornate le informazioni riguardanti l'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU5	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati i dati dell'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU6	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, leggendo correttamente storico delle transazioni o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU7	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornato lo storico delle transazioni o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU8	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, leggendo correttamente le informazioni del profilo utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU9	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati la gestione delle informazioni del profilo utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU10	Verificare che, in base ai parametri forniti in input, il metodo legga correttamente i record nel relativo DB per i microservizi inseriti dall'utente sviluppatore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU11	Verificare che, in base ai parametri forniti in input, il metodo crei un nuovo record nel relativo DB per il microservizio inserito dall'utente sviluppatore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU12	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, verificando che il microservizio inserito dallo sviluppatore sia effettivamente aggiunto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU13	Verificare che, in base ai parametri forniti in input, i metodi effettuino una corretta lettura delle informazioni riguardanti i microservizi inseriti da uno sviluppatore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU14	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti i microservizi inseriti da uno sviluppatore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU15	Verificare che, in base ai parametri forniti in input, i metodi effettuino una corretta lettura delle informazioni riguardanti le licenze attive relative all'acquisto di una API da parte di un utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU16	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti le licenze attive relative all'acquisto di una API da parte di un utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU17	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, verificando la password inserita in fase di autenticazione e restituendone un messaggio di conferma o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU18	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, leggendo correttamente le informazioni di un microservizio o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU19	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, inserendo un nuovo microservizio o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU20	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiornando le informazioni di un microservizio o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU21	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni di un microservizio acquistato e/o utilizzato in modo corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU22	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, creando il report di SLA per un microservizio acquistato e/o utilizzato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU23	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dettagli dei microservizi, richiamando i metodi del relativo <i>Service</i> . Nello specifico, restituendo i dettagli dei microservizi inseriti da uno sviluppatore o che, nel caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU24	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo i dettagli dei microservizi, richiamando i metodi del relativo <i>Service</i> . Nello specifico, restituendo le licenze attive per ogni microservizio proprio o che, nel caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU25	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo l'autenticazione al sistema, in particolare la richiesta di autenticazione o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU26	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo l'autenticazione al sistema, in particolare la richiesta di registrazione o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU27	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo l'autenticazione al sistema, in particolare la richiesta di recupero della password dimenticata o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU28	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il logout dal sistema o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU29	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il login al sistema o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU30	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la registrazione al sistema o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU31	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la possibilità di effettuare il redirect alla pagina di visualizzazione profilo utente o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU32	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la possibilità di effettuare il redirect alla pagina di gestione profilo utente o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU33	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la possibilità di effettuare il redirect alla pagina di visualizzazione delle proprie transazioni o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU34	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la possibilità di effettuare il redirect alla pagina di gestione del conto o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU35	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo la gestione per l'inserimento di un nuovo microservizio o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU36	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo la gestione per la modifica dei microservizi già inseriti in precedenza e presenti sul marketplace o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU37	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il recupero della password dimenticata da un utente, in particolare la comunicazione con il servizio che si occupa di inviare la mail o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU38	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il recupero della password dimenticata da un utente, in particolare la comunicazione con il servizio che si occupa della gestione dell'evento redirect alla pagina di login o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU39	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il profilo personale dell'utente, in particolare l'invio delle nuove informazioni al service tramite l'apposito metodo.	<i>Superato</i>
TU40	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la registrazione di un utente al sistema, e in particolare che nel caso di buona riuscita deve essere mostrato un messaggio di successo.	<i>Superato</i>

Id Test	Descrizione	Stato
TU41	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la registrazione di un utente al sistema, e in particolare che nel caso di registrazione fallita, deve essere mostrato un messaggio di errore.	<i>Superato</i>
TU42	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo tutti i microservizi inseriti da un utente sviluppatore, recuperando tutte le informazioni a essi associate o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU43	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo tutti i microservizi inseriti da un utente sviluppatore, gestendo l'evento click sul pulsante di modifica di un microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU44	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le licenze per i microservizi tramite concessione di API Key o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU45	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la visualizzazione dei dati di utilizzo di un singolo microservizio, restituendo un'indicazione se la SLA dichiarata dal microservizio è stata rispettata durante le varie chiamate dei clienti o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU46	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo la visualizzazione della homepage dell'applicazione web, ovvero che siano presenti all'interno della home tutte le funzionalità previste o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU47	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la ricerca di microservizi ed API all'interno del marketplace attraverso l'evento click sui pulsanti per visualizzare il microservizio selezionato o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU48	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la ricerca di microservizi ed API all'interno del marketplace attraverso l'evento click sui pulsanti per acquistare una licenza (API key) o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU49	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la ricerca di microservizi ed API all'interno del marketplace attraverso l'evento click sui pulsanti per visualizzarne la documentazione o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU50	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo la generazione di una nuova API key nel caso l'utente abbia perso o dimenticato la precedente fornita al momento della transazione o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU51	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo il rinnovo di una licenza per un microservizio attraverso una nuova transazione o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU52	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le informazioni di un microservizio; in particolare i metodi devono permettere di modificare il nome del microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU53	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le informazioni di un microservizio; in particolare i metodi devono permettere di modificare l'autore del microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU54	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le informazioni di un microservizio; in particolare i metodi devono permettere di modificare la versione del microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU55	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le informazioni di un microservizio; in particolare i metodi devono permettere di modificare la data di ultimo aggiornamento di un microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU56	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le informazioni di un microservizio; in particolare i metodi devono permettere di modificare l'interfaccia di un microservizio ed eventuali avvisi riguardanti il suo funzionamento o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU57	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, memorizzando i dati relativi ad un utente e gestendoli richiamando i relativi <i>Controller</i> . Nello specifico, gestendo l'autenticazione dell'utente al sistema o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU58	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, memorizzando i dati relativi ad un utente e gestendoli richiamando i relativi <i>Controller</i> . Nello specifico, gestendo la ricerca di microservizi e sviluppatori all'interno dell'applicazione web, i dati e le statistiche riguardanti un microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU59	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, memorizzando i dati relativi ad un utente e gestendoli richiamando i relativi <i>Controller</i> . Nello specifico, gestendo i dati e le statistiche riguardanti un microservizio o che, nel caso di un errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU60	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto contenente le informazioni sull'errore, ritornando correttamente il titolo, il messaggio ed il codice dell'errore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU61	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le chiamate ai microservizi, ossia il loro inoltro e salvataggio; in particolare i metodi devono permettere di inviare una richiesta di chiamata ad un microservizio o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU62	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo le chiamate ai microservizi, ossia il loro inoltro e salvataggio; in particolare i metodi devono permettere di registrarne i dati di utilizzo di una chiamata ad un microservizio o, nel caso di un errore, deve garantire che la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU63	Verificare che, in base ai parametri forniti in input, i metodi richiedano in maniera corretta il reperimento e il salvataggio dei dati di un utente registrato (cliente o sviluppatore che sia), che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Tabella 34: Test di unità

4.1.1 Grado di completamento dei test di unità

Di seguito viene fornito un pie chart che rappresenta il grado di completamento dei **test di unità** implementati e superati.



Figura 4: Grado di completamento test di unità

4.2 Test di integrazione

Questa tipologia di test serve a verificare il corretto funzionamento delle singole componenti di sistema progettate durante l'attività di progettazione ad alto livello. Per questa tipologia di test, l'idea è quella di utilizzare un approccio top-down, in modo da sottoporre a test e integrare per primi i moduli di livello più alto. Così facendo, però, sarà necessario simulare le componenti di livello inferiore con degli stub. Una volta codificate, le componenti di livello più basso dovranno essere a loro volta integrate e testate.

L'approccio top-down rientra tra le strategie di integrazione incrementale, che consentono di poter determinare in modo più rapido quale componente è la causa di problemi, poichè i difetti rilevati dai test saranno da attribuirsi all'ultima componente aggiunta. I test di integrazione saranno descritti nel modo seguente:

TI[*IdComponente*]

dove:

- **IdComponente** rappresenta il codice identificativo progressivo del componente preso in considerazione.

Id Test	Descrizione	Stato
TI1	Viene verificato che l'applicazione web API Market gestisca correttamente il front-end del prodotto e le sue interazioni con il back-end.	<i>Superato</i>
TI2	Viene verificato che i <i>Controllers</i> del front-end si integrino correttamente nell'applicazione web.	<i>Superato</i>
TI3	Viene verificato che i <i>Services</i> permettano una corretta interazione con il back-end.	<i>Superato</i>
TI4	Viene verificato che il <i>Model</i> si integri correttamente con i <i>Services</i> e con il resto delle componenti dell'applicazione che lo utilizzano.	<i>Superato</i>
TI5	Viene verificato che le <i>Views</i> si integrino correttamente con i <i>Controllers</i> , in modo da visualizzare senza errori i dati ricevuti.	<i>Superato</i>
TI6	Viene verificato che l'applicazione web gestisca correttamente il back-end del prodotto, in modo tale da fornire al front-end le informazioni richieste.	<i>Superato</i>
TI7	Viene verificato che il <i>Server</i> sia configurato e che sia in grado di caricare correttamente tutte le librerie necessarie all'istanziamento delle classi in modo corretto.	<i>Non implementato</i>
TI8	Viene verificato che la <i>Single Page App</i> si integri correttamente con i microservizi <i>Jolie</i> utilizzati per il back-end.	<i>Superato</i>

Id Test	Descrizione	Stato
TI9	Viene verificato che i <i>Controllers</i> si integrino correttamente e gestiscano le richieste inoltrate dai <i>Routers</i> .	<i>Superato</i>
TI10	Viene verificato che il <i>Model</i> si integri correttamente con i <i>Controllers</i> per la gestione dell'inserimento, della modifica e della cancellazione dei dati.	<i>Superato</i>

Tabella 35: Test di integrazione

4.2.1 Grado di completamento dei test di integrazione

Di seguito viene fornito un pie chart che rappresenta il grado di completamento dei **test di integrazione** implementati e superati.

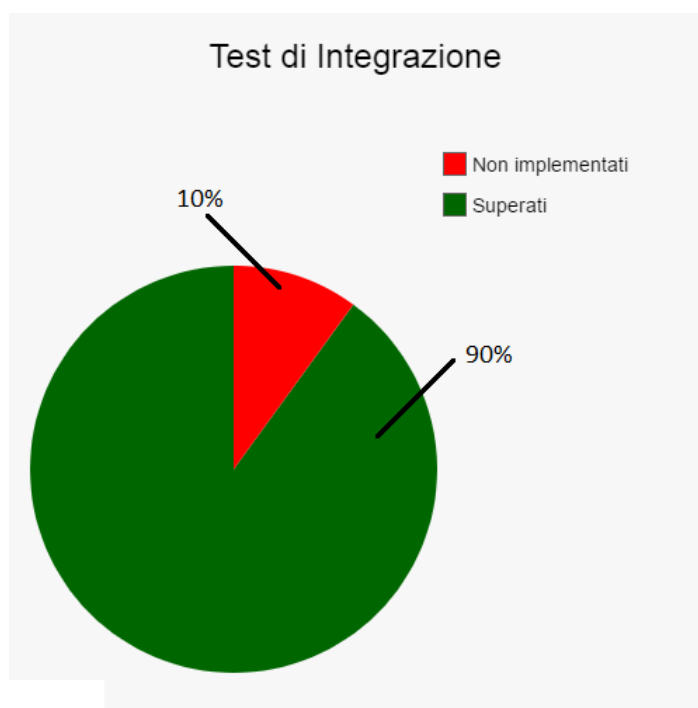


Figura 5: Grado di completamento test di integrazione

4.3 Test di sistema

Questa tipologia di test serve a verificare il corretto comportamento e funzionamento dell'architettura. Tale tipologia di test deve verificare che ci sia la copertura totale dei requisiti software stabiliti durante l'Analisi dei Requisiti. I test di sistema saranno organizzati nel modo seguente:

$$TS[TipologiaRequisito][RilevanzaRequisito][CodiceRequisito]$$

dove:

- **TipologiaRequisito** può assumere valori tra:
 - *V* per i requisiti di vincolo;
 - *F* per i requisiti di funzionalità;
 - *Q* per i requisiti di qualità;
 - *P* per i requisiti prestazionali.
- **RilevanzaRequisito** può assumere valori tra:
 - *O* per i requisiti obbligatori;
 - *D* per i requisiti desiderabili;
 - *F* per i requisiti facoltativi.
- **CodiceRequisito** assume un valore gerarchico che identifica il singolo requisito.

Id Test	Descrizione	Stato
TSFO1	Viene verificato che il sistema registri correttamente un utente.	<i>Superato</i>
TSFO2	Viene verificato che il sistema riesca a far effettuare correttamente il login ad un utente attraverso API Market.	<i>Superato</i>
TSFD3	Viene verificato che il sistema permetta il recupero della password per un utente registrato ad API Market.	<i>Non Implementato</i>
TSFO4	Viene verificato che il sistema ricerchi correttamente una API.	<i>Superato</i>
TSFO4.3	Viene verificato che il sistema restituisca un insieme di risultati coerenti secondo i parametri di ricerca specificati da un utente.	<i>Superato</i>
TSFO5	Viene verificato che il sistema visualizzi correttamente i dati relativi ad una API.	<i>Superato</i>
TSFO5.6	Viene verificato che il sistema consenta la consultazione della documentazione di una API.	<i>Superato</i>
TSFO5.7	Viene verificato che il sistema visualizzi correttamente i dati di utilizzo di una API.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TSFO6.2	Viene verificato che il sistema visualizzi correttamente la lista di API acquistate da un utente.	<i>Superato</i>
TSFO7	Viene verificato che il sistema consenta l'acquisto di una API, con conseguente rilascio di una API Key, per un utente.	<i>Superato</i>
TSFO7.5.3	Viene verificato che il sistema crei e visualizzi correttamente l'API Key associata ad una API, per cui è stata acquistata una licenza da un utente.	<i>Superato</i>
TSFO8	Viene verificato che il sistema gestisca correttamente le API inserite da uno sviluppatore.	<i>Superato</i>
TSFO8.2	Viene verificato che il sistema visualizzi correttamente la lista delle API registrate da uno sviluppatore.	<i>Superato</i>
TSFO8.2.4	Viene verificato che il sistema modifichi correttamente le informazioni relative ad una API registrata da uno sviluppatore.	<i>Superato</i>
TSFO8.2.8	Viene verificato che il sistema visualizzi correttamente il guadagno netto, in base alla policy, di una API da lui inserita.	<i>Non Implementato</i>
TSFO8.2.9	Viene verificato che il sistema elimini correttamente una API su richiesta dello sviluppatore autore.	<i>Superato</i>
TSFO9	Viene verificato che il sistema inserisca correttamente nel marketplace una nuova API.	<i>Superato</i>
TSFO10.1	Viene verificato che il sistema permetta di gestire correttamente ad un utente le informazioni del proprio profilo.	<i>Superato</i>
TSFO10.1.1	Viene verificato che il sistema visualizzi correttamente le informazioni del profilo di un utente.	<i>Superato</i>
TSFO10.1.2	Viene verificato che il sistema modifichi correttamente le informazioni di un utente.	<i>Superato</i>
TSFD10.1.2.9	Viene verificato che il sistema effettui correttamente l'upgrade dell'account di un utente.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TSFO10.2	Viene verificato che il sistema permetta di gestire correttamente ad un utente il proprio conto.	<i>Superato</i>
TSFO10.2.1	Viene verificato che il sistema visualizzi correttamente il saldo attuale del conto di un utente.	<i>Superato</i>
TSFO10.2.2	Viene verificato che il sistema permetta ad un utente di ricaricare il saldo del proprio conto attraverso PayPal.	<i>Non Implementato</i>
TSFO10.3	Viene verificato che il sistema visualizzi correttamente lo storico delle transazioni di un utente.	<i>Superato</i>
TSFO10.3.2	Viene verificato che il sistema visualizzi correttamente la lista delle transazioni concluse di un utente.	<i>Superato</i>
TSFO11	Viene verificato che il sistema riesca a far effettuare correttamente il logout ad un utente.	<i>Superato</i>
TSFO12.1.1.1	Viene verificato che il sistema visualizzi correttamente i dati avanzati di una API da parte di un amministratore.	<i>Non Implementato</i>
TSFO12.1.1.2	Viene verificato che il sistema sospenda correttamente una API su richiesta di un amministratore.	<i>Superato</i>
TSFD12.1.1.3	Viene verificato che il sistema elimini correttamente una API su richiesta di un amministratore.	<i>Superato</i>
TSFO12.2.1.1	Viene verificato che il sistema permetta ad un amministratore di sospendere correttamente un utente.	<i>Superato</i>
TSFO12.2.1.2	Viene verificato che il sistema permetta ad un amministratore di sospendere correttamente i pagamenti verso un utente.	<i>Non Implementato</i>
TSFO12.2.1.3	Viene verificato che il sistema permetta ad un amministratore di revocare correttamente la sospensione ad un utente sospeso.	<i>Superato</i>
TSFO12.2.1.5	Viene verificato che il sistema permetta ad un amministratore di eliminare correttamente un utente.	<i>Superato</i>
TSVO1	Viene verificato che il sistema abbia un'architettura a microservizi.	<i>Superato</i>

Id Test	Descrizione	Stato
TSVO2	Viene verificato che il sistema utilizzi il linguaggio <i>Jolie</i> per l'API Gateway, il back-end dell'applicazione web e le interfacce dei microservizi.	<i>Superato</i>
TSVO3	Viene verificato che il sistema utilizzi il linguaggio di markup <i>HTML5</i> , unito a fogli di stile in <i>CSS3</i> e linguaggio di scripting <i>JavaScript</i> .	<i>Superato</i>
TSVO4	Viene verificato che il sistema utilizzi il DBMS di tipo relazionale <i>MySQL</i> .	<i>Superato</i>
TSVO5	Viene verificato che il sistema funzioni su <i>Google Chrome</i> versione 55.0 o superiore	<i>Superato</i>
TSVO6	Viene verificato che il sistema funzioni su <i>Mozilla Firefox</i> versione 51.0 o superiore	<i>Superato</i>
TSVO7	Viene verificato che il sistema funzioni su <i>Safari</i> versione 10.0 o superiore	<i>Superato</i>
TSVD8	Viene verificato che il sistema funzioni su <i>Opera</i> versione 42.0 o superiore	<i>Superato</i>
TSVD9	Viene verificato che il sistema funzioni su <i>Internet Explorer</i> versione 11.0 o superiore	<i>Superato</i>
TSVO10	Viene verificato che il sistema funzioni su <i>Microsoft Edge</i> versione 38.0 o superiore	<i>Superato</i>
TSVO11	Viene verificato che il sistema funzioni su <i>Android Browser</i> versione 5.1 o superiore	<i>Superato</i>
TSVO12	Viene verificato che il sistema funzioni su <i>Safari</i> per iOS 10 o versioni superiori	<i>Superato</i>
TSVF13	Viene verificato che il sistema funzioni su <i>Google Chrome</i> per iOS versione 56.0 o superiore	<i>Non Implementato</i>
TSVD14	Viene verificato che il sistema funzioni su <i>Google Chrome</i> per Android versione 56.0 o superiore	<i>Non Implementato</i>

Tabella 36: Test di sistema

4.3.1 Grado di completamento dei test di sistema

Di seguito viene fornito un pie chart che rappresenta il grado di completamento dei **test di sistema** implementati e superati.

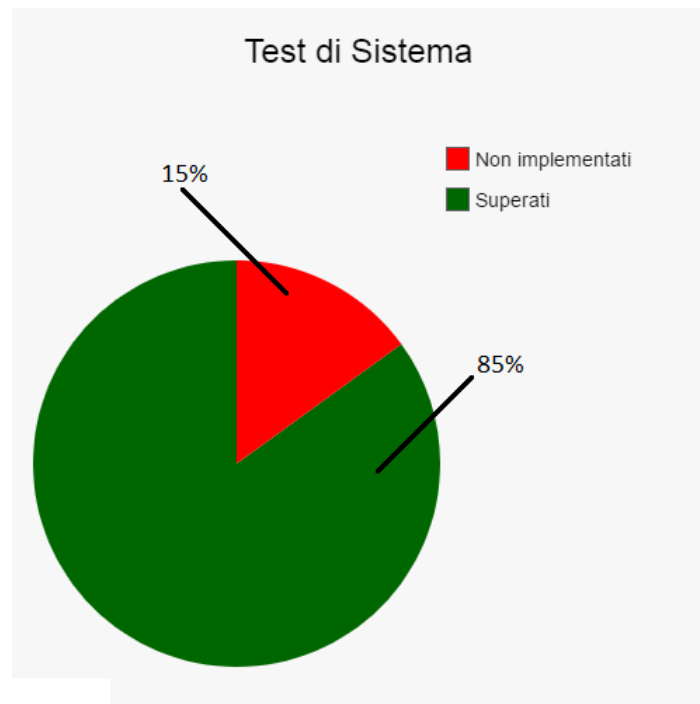


Figura 6: Grado di completamento test di sistema

4.4 Test di non regressione

Questa tipologia di test consiste nell'eseguire nuovamente i test che coinvolgono le componenti software che hanno subito modifiche, in modo da verificare che i cambiamenti apportati non compromettano il funzionamento di componenti che non sono stati aggiornati e che, precedentemente, non erano soggetti ad errori.

I test di non regressione saranno descritti nel modo seguente:

$$\mathbf{TNR}[IdTest]$$

dove ***IdTest*** rappresenta il codice identificativo progressivo della modifica apportata, la quale ha prodotto il corrispondente test di non regressione.

Id Test	Descrizione	Stato
TNR1	Si verifica che i servizi attivi e già implementati reagiscano correttamente (senza nuove problematiche) all'aggiunta di un nuovo servizio	<i>Superato</i>
TNR2	Si verifica che il gateway gestisca correttamente opportune modifiche e aggiunte di servizi all'applicazione	<i>Superato</i>
TNR3	Si verifica che eventuali modifiche ai servizi relativi agli utenti non vadano a impattare sugli attuali utenti e i dati ad essi correlati	<i>Superato</i>
TNR4	Si verifica che eventuali modifiche ai servizi relativi alla SLA non vadano a impattare sulle statistiche pregresse e già registrate	<i>Superato</i>
TNR5	Si verifica che eventuali modifiche ai servizi relativi ai microservizi non vadano a impattare sui dati pregressi	<i>Superato</i>

Tabella 37: Test di non regressione

4.4.1 Grado di completamento dei test di non regressione

Di seguito viene fornito un pie chart che rappresenta il grado di completamento dei **test di non regressione** implementati e superati.

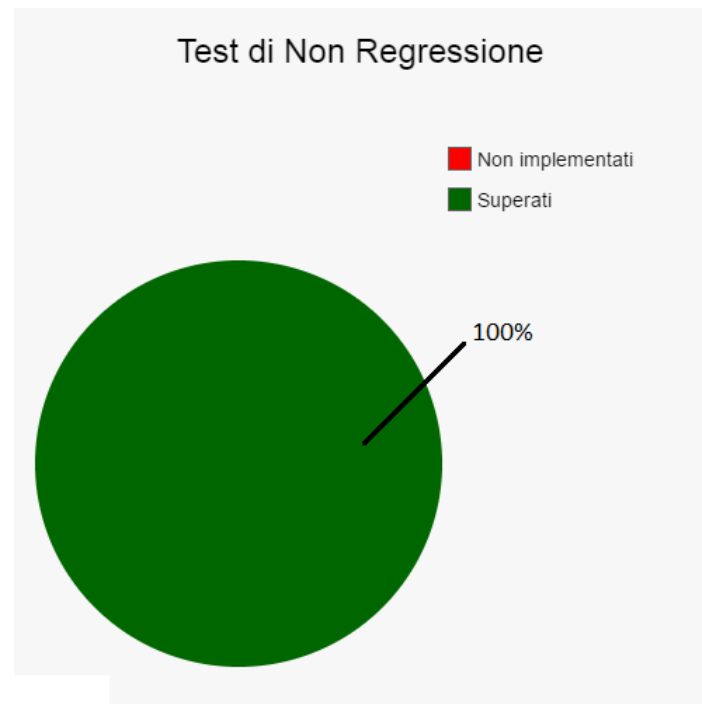


Figura 7: Grado di completamento test di non regressione

4.5 Test di validazione

Questa tipologia di test serve a verificare che il prodotto soddisfi le richieste del proponente attraverso le funzionalità implementate.

Per questo motivo, occorrerà simulare il comportamento generale dell'applicativo e dell'utente che interagisce con esso, attraverso delle macro azioni.

I test di validazione saranno organizzati nel modo seguente:

$$\mathbf{TV}[TipologiaRequisito][RilevanzaRequisito][CodiceRequisito]$$

dove:

- **TipologiaRequisito** può assumere valori tra:
 - *V* per i requisiti di vincolo;
 - *F* per i requisiti di funzionalità;
 - *Q* per i requisiti di qualità;
 - *P* per i requisiti prestazionali.
- **RilevanzaRequisito** può assumere valori tra:
 - *O* per i requisiti obbligatori;
 - *D* per i requisiti desiderabili;
 - *F* per i requisiti facoltativi.
- **CodiceRequisito** assume un valore gerarchico che identifica il singolo requisito.

Id Test	Descrizione	Stato
TVFO1	L'utente intende registrarsi alla piattaforma API Market. All'utente è richiesto di: <ul style="list-style-type: none">• Trovarsi nella sezione apposita;• Compilare il form di registrazione;• Premere il pulsante di conferma;• Verificare attraverso l'autenticazione che la registrazione sia avvenuta correttamente.	<i>Superato</i>

Id Test	Descrizione	Stato
TVFO2	<p>L'utente intende autenticarsi alla piattaforma API Market. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella sezione apposita; • Essere in possesso delle credenziali richieste; • Inserire le credenziali nell'apposito form; • Premere il pulsante di autenticazione; • Verificare che l'autenticazione sia effettivamente avvenuta. 	<i>Superato</i>
TVFO4	<p>L'utente autenticato intende ricercare una API sul marketplace. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella sezione apposita; • Ricercare una API digitando le keywords; • Visualizzazione dei risultati della ricerca, secondo le keywords specificate. 	<i>Superato</i>
TVFO5	<p>L'utente autenticato intende visualizzare le informazioni relative ad una API. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella sezione apposita; • Selezionare l'API di cui vuole visualizzare le informazioni; • Visualizzazione dei dati relativi alla API selezionata. 	<i>Superato</i>
TVFO5.6	<p>L'utente autenticato intende consultare la documentazione relativa ad una API. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella sezione apposita; • Selezionare l'API di interesse; • Visualizzazione della documentazione dell'API. 	<i>Superato</i>

Id Test	Descrizione	Stato
TVFO7	<p>L'utente autenticato intende acquistare l'API che sta visualizzando. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Selezionare la policy di vendita;• Confermare l'acquisto attraverso l'apposito pulsante;• Verificare la transazione nel proprio storico.	<i>Superato</i>
TVFO8.2	<p>L'utente sviluppatore intende visualizzare l'elenco di API da lui caricate sulla piattaforma API Market. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Verificare che vengano visualizzati tutte le proprie API inserite.	<i>Superato</i>
TVFO8.2.3	<p>L'utente sviluppatore intende visualizzare il numero di licenze attive per le API da lui caricate sulla piattaforma API Market. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Selezionare una API dall'elenco visualizzato;• Verificare che venga il numero di licenze attive per la propria API registrata.	<i>Superato</i>

Id Test	Descrizione	Stato
TVFO8.2.4	<p>L'utente sviluppatore intende modificare i dati relativi ad una sua API caricata.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante "Modifica API";• Modificare i dati dell'API selezionata;• Premere il pulsante di conferma modifica;• Verificare che sia stata modificata l'API.	<i>Superato</i>
TVFO9	<p>L'utente sviluppatore intende caricare una propria API sulla piattaforma API Market.</p> <p>All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante "Inserisci API";• Inserire i dati necessari alla pubblicazione della propria API;• Premere il pulsante di conferma inserimento nuova API;• Verificare che sia stata aggiunta al marketplace l'API.	<i>Superato</i>
TVFO10.1.1	<p>L'utente autenticato intende visualizzare il proprio profilo. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Visualizzare il proprio profilo.	<i>Superato</i>

Id Test	Descrizione	Stato
TVFO10.1.2	<p>L'utente autenticato intende modificare i propri dati. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Modificare i campi dati consentiti; • Premere il tasto "Conferma Modifiche"; • Visualizzare il profilo dell'utente modificato. 	<i>Superato</i>
TVFD10.1.2.9	<p>L'utente autenticato intende modificare la tipologia di utenza, effettuando un upgrade dell'account. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Effettuare l'upgrade dell'account, passando da "cliente" a "sviluppatore"; • Verificare la modifica effettuata. 	<i>Superato</i>
TVFO10.2.1	<p>L'utente intende visualizzare il saldo del proprio conto associato all'account. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Premere il pulsante "Conto Virtuale"; • Visualizzazione del saldo attuale del proprio conto. 	<i>Superato</i>
TVFO10.3	<p>L'utente autenticato intende visualizzare lo storico delle transazioni effettuate. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella sezione apposita; • Visualizzare lo storico delle transazioni. 	<i>Superato</i>

Id Test	Descrizione	Stato
TVFO11	<p>L'utente intende disconnettersi dalla piattaforma API Market. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Premere il pulsante di logout;• Verificare che la disconnessione sia effettivamente avvenuta.	<i>Superato</i>
TVFO12.2.1	<p>L'utente amministratore della piattaforma API Market intende moderare un determinato utente registrato. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella sezione apposita;• Selezionare l'utente che si vuole moderare;• Inserire i dati del rapporto di moderazione;• Verificare che l'utente sia stato effettivamente moderato.	<i>Superato</i>

Tabella 38: Test di validazione

4.5.1 Grado di completamento dei test di validazione

Di seguito viene fornito un pie chart che rappresenta il grado di completamento dei **test di validazione** implementati e superati.



Figura 8: Grado di completamento test di validazione

A Attività di verifica

In questa sezione del documento, vengono descritti e analizzati gli esiti delle attività di verifica svolte su tutti i documenti che vengono consegnati nelle varie revisioni di avanzamento del progetto.

A.1 Tracciamento

Il team NetBreak ha deciso di utilizzare la piattaforma *NetBreakDB* al fine di facilitare il tracciamento di classi, requisiti, casi d'uso, test, etc.

A.1.1 Tracciamento Test di Integrazione - Componenti

Test	Componenti
TI1	APIM::FrontEnd::Index
TI2	APIM::FrontEnd::App::Controllers
TI3	APIM::BackEnd::Services
TI4	APIM::FrontEnd::App::Models APIM::BackEnd::Services
TI5	APIM::FrontEnd::App::Views APIM::FrontEnd::Controllers
TI6	APIM::BackEnd::Services
TI7	APIM::FrontEnd::node_modules APIM::FrontEnd::App::AppRun
TI8	APIM::FrontEnd::App::AppRouter APIM::BackEnd::Services APIM::BackEnd::Gateway
TI9	APIM::FrontEnd::App::Controllers
TI10	APIM::FrontEnd::App::Models APIM::FrontEnd::App::Controllers

Tabella 39: Tracciamento Test di Integrazione-Componenti

A.1.2 Tracciamento Componenti - Test di Integrazione

Componenti	Test
APIM::FrontEnd::Index	TI1
APIM::FrontEnd::App::Controllers	TI2
APIM::BackEnd::Services	TI3
APIM::FrontEnd::App::Models	TI4
APIM::BackEnd::Services	TI4
APIM::FrontEnd::App::Views	TI5
APIM::FrontEnd::Controllers	TI5
APIM::BackEnd::Services	TI6
APIM::FrontEnd::node_modules	TI7
APIM::FrontEnd::App::AppRun	TI7
APIM::FrontEnd::App::AppRouter	TI8
APIM::BackEnd::Services	TI8
APIM::BackEnd::Gateway	TI8
APIM::FrontEnd::App::Controllers	TI9
APIM::FrontEnd::App::Models	TI10
APIM::FrontEnd::App::Controllers	TI10

Tabella 40: Tracciamento Componenti-Test di Integrazione

A.1.3 Tracciamento Test di Sistema - Requisiti

Test	Requisito
TSFO1	RFO1
TSFO2	RFO2
TSFD3	RFD3
TSFO4	RFO4
TSFO4.3	RFO4.3
TSFO5	RFO5
TSFO5.6	RFO5.6
TSFO5.7	RFO5.7
TSFO6.2	RFO6.2
TSFO7	RFO7
TSFO7.5.3	RFO7.5.3
TSFO8	RFO8
TSFO8.2	RFO8.2
TSFO8.2.4	RFO8.2.4
TSFO8.2.8	RFO8.2.8
TSFO8.2.9	RFO8.2.9
TSFO9	RFO9
TSFO10.1	RFO10.1
TSFO10.1.1	RFO10.1.1
TSFO10.1.2	RFO10.1.2
TSFD10.1.2.9	RFD10.1.2.9
TSFO10.2	RFO10.2
TSFO10.2.1	RFO10.2.1
TSFO10.2.2	RFO10.2.2
TSFO10.3	RFO10.3
TSFO10.3.2	RFO10.3.2
TSFO11	RFO11

Test	Requisito
TSFO12.1.1.1	RFO12.1.1.1
TSFO12.1.1.2	RFO12.1.1.2
TSFD12.1.1.3	RFD12.1.1.3
TSFO12.2.1.1	RFO12.2.1.1
TSFO12.2.1.2	RFO12.2.1.2
TSFO12.2.1.3	RFO12.2.1.3
TSFO12.2.1.5	RFO12.2.1.5
TSVO1	RVO1
TSVO2	RVO2
TSVO3	RVO3
TSVO4	RVO4
TSVO5	RVO5
TSVO6	RVO6
TSVO7	RVO7
TSVD8	RVD8
TSVD9	RVD9
TSVO10	RVO10
TSVO11	RVO11
TSVO12	RVO12
TSVF13	RVF13
TSVD14	RVD14

Tabella 41: Tracciamento Test di Sistema-Requisiti

A.1.4 Tracciamento Requisiti - Test di Sistema

Requisito	Test
RFO1	TSFO1
RFO2	TSFO2
RFD3	TSFD3
RFO4	TSFO4
RFO4.3	TSFO4.3
RFO5	TSFO5
RFO5.6	TSFO5.6
RFO5.7	TSFO5.7
RFO6.2	TSFO6.2
RFO7	TSFO7
RFO7.5.3	TSFO7.5.3
RFO8	TSFO8
RFO8.2	TSFO8.2
RFO8.2.4	TSFO8.2.4
RFO8.2.8	TSFO8.2.8
RFO8.2.9	TSFO8.2.9
RFO9	TSFO9
RFO10.1	TSFO10.1
RFO10.1.1	TSFO10.1.1
RFO10.1.2	TSFO10.1.2
RFD10.1.2.9	TSFD10.1.2.9
RFO10.2	TSFO10.2
RFO10.2.1	TSFO10.2.1
RFO10.2.2	TSFO10.2.2
RFO10.3	TSFO10.3
RFO10.3.2	TSFO10.3.2
RFO11	TSFO11

Requisito	Test
RFO12.1.1.1	TSFO12.1.1.1
RFO12.1.1.2	TSFO12.1.1.2
RFD12.1.1.3	TSFD12.1.1.3
RFO12.2.1.1	TSFO12.2.1.1
RFO12.2.1.2	TSFO12.2.1.2
RFO12.2.1.3	TSFO12.2.1.3
RFO12.2.1.5	TSFO12.2.1.5
RVO1	TSVO1
RVO2	TSVO2
RVO3	TSVO3
RVO4	TSVO4
RVO5	TSVO5
RVO6	TSVO6
RVO7	TSVO7
RVD8	TSVD8
RVD9	TSVD9
RVO10	TSVO10
RVO11	TSVO11
RVO12	TSVO12
RVF13	TSVF13
RVD14	TSVD14

Tabella 42: Tracciamento Requisiti-Test di Sistema

A.1.5 Tracciamento Test di Validazione - Requisiti

Test	Requisito
TVFO1	RFO1
TVFO2	RFO2
TVFO4	RFO4
TVFO5	RFO5
TVFO5.6	RFO5.6
TVFO7	RFO7
TVFO8.2	RFO8.2
TVFO8.2.3	RFO8.2.3
TVFO8.2.4	RFO8.2.4
TVFO9	RFO9
TVFO10.1.1	RFO10.1.1
TVFO10.1.2	RFO10.1.2
TVFD10.1.2.9	RFO10.1.2.9
TVFO10.2.1	RFO10.2.1
TVFO10.3	RFO10.3
TVFO11	RFO11
TVFO12.2.1	RFO12.2.1

Tabella 43: Tracciamento Test di Validazione-Requisiti

A.2 Resoconto verifica documenti

L'analisi dei documenti mediante la tecnica *Walkthrough* ha reso possibile individuare alcuni errori frequenti, che si sono aggiunti alla lista di controllo stilata ed aggiornata nelle varie revisioni, utile per applicare la tecnica *Inspection* nelle future attività di verifica.

Il team, attraverso il software interno *NetBreakDB*, è riuscito ad effettuare i tracciamenti per le componenti di interesse previste da ogni revisione (casi d'uso-requisiti, requisiti-fonti, requisiti-componenti, requisiti-classi, etc.).

Inoltre, questo software è stato utilizzato per generare le tabelle dei vari test e dei relativi tracciamenti con requisiti, componenti, classi e metodi.

A.3 Revisione dei Requisiti

Di seguito, sono riportati gli esiti delle verifiche sottoposte a tutti i documenti, per il calcolo della metrica *Indice Gulpease* (MPC19).

Documento	Indice Gulpease	Esito
<i>Norme di Progetto 1.0.0</i>	49	<i>Superato</i>
<i>Piano di Progetto 1.0.0</i>	50	<i>Superato</i>
<i>Piano di Qualifica 1.0.0</i>	42	<i>Superato</i>
<i>Analisi dei Requisiti 1.0.0</i>	68	<i>Superato</i>
<i>Studio di Fattibilità 1.0.0</i>	54	<i>Superato</i>
<i>Glossario 1.0.0</i>	43	<i>Superato</i>
<i>Verbale Interno - 28/11/2016</i>	60	<i>Superato</i>
<i>Verbale Interno - 01/12/2016</i>	63	<i>Superato</i>
<i>Verbale Interno - 12/12/2016</i>	61	<i>Superato</i>
<i>Verbale Esterno - 22/12/2016</i>	59	<i>Superato</i>
<i>Verbale Interno - 28/12/2016</i>	61	<i>Superato</i>

Tabella 44: Resoconto verifiche documenti - RR

A.4 Revisione di Progettazione

Di seguito, sono riportati gli esiti delle verifiche sottoposte a tutti i documenti, per il calcolo della metrica *Indice Gulpease* (MPC19).

Documento	Indice Gulpease	Esito
<i>Specifica Tecnica 1.0.0</i>	67	<i>Superato</i>
<i>Norme di Progetto 2.0.0</i>	57	<i>Superato</i>
<i>Piano di Progetto 2.0.0</i>	55	<i>Superato</i>
<i>Piano di Qualifica 2.0.0</i>	54	<i>Superato</i>
<i>Analisi dei Requisiti 2.0.0</i>	70	<i>Superato</i>
<i>Glossario 2.0.0</i>	49	<i>Superato</i>
<i>Verbale Interno - 27/01/2017</i>	55	<i>Superato</i>
<i>Verbale Interno - 04/02/2017</i>	64	<i>Superato</i>
<i>Verbale Interno - 13/02/2017</i>	58	<i>Superato</i>
<i>Verbale Esterno - 16/02/2017</i>	60	<i>Superato</i>
<i>Verbale Interno - 17/02/2017</i>	57	<i>Superato</i>
<i>Verbale Interno - 21/02/2017</i>	63	<i>Superato</i>
<i>Verbale Interno - 02/03/2017</i>	61	<i>Superato</i>
<i>Verbale Interno - 03/03/2017</i>	61	<i>Superato</i>

Tabella 45: Resoconto verifiche documenti - RP

A.5 Revisione di Qualifica

Di seguito, sono riportati gli esiti delle verifiche sottoposte a tutti i documenti, per il calcolo della metrica *Indice Gulpease* (MPC19).

Documento	Indice Gulpease	Esito
<i>Definizione di Prodotto 1.0.0</i>	72	<i>Superato</i>
<i>Manuale Utente 1.0.0</i>	68	<i>Superato</i>
<i>Specifica Tecnica 2.0.0</i>	69	<i>Superato</i>
<i>Norme di Progetto 3.0.0</i>	61	<i>Superato</i>
<i>Piano di Progetto 3.0.0</i>	60	<i>Superato</i>
<i>Piano di Qualifica 3.0.0</i>	63	<i>Superato</i>
<i>Analisi dei Requisiti 3.0.0</i>	71	<i>Superato</i>
<i>Glossario 3.0.0</i>	50	<i>Superato</i>
<i>Tracciamento Verbali Interni RQ</i>	68	<i>Superato</i>
<i>Tracciamento Verbali Esterni RQ</i>	67	<i>Superato</i>
<i>Verbale Interno 2017-03-16</i>	57	<i>Superato</i>
<i>Verbale Interno 2017-03-31</i>	61	<i>Superato</i>
<i>Verbale Interno 2017-04-28</i>	59	<i>Superato</i>
<i>Verbale Esterno 2017-03-17</i>	60	<i>Superato</i>
<i>Verbale Esterno 2017-03-30</i>	65	<i>Superato</i>
<i>Verbale Esterno 2017-04-06</i>	62	<i>Superato</i>
<i>Verbale Esterno 2017-04-10</i>	64	<i>Superato</i>
<i>Verbale Esterno 2017-04-13</i>	61	<i>Superato</i>
<i>Verbale Esterno 2017-04-19</i>	65	<i>Superato</i>
<i>Verbale Esterno 2017-04-24</i>	59	<i>Superato</i>

Tabella 46: Resoconto verifiche documenti - RQ

A.6 Revisione di Accettazione

Di seguito, sono riportati gli esiti delle verifiche sottoposte a tutti i documenti, per il calcolo della metrica *Indice Gulpease* (MPC19).

Documento	Indice Gulpease	Esito
<i>Definizione di Prodotto 2.0.0</i>	73	<i>Superato</i>
<i>Manuale Utente 2.0.0</i>	72	<i>Superato</i>
<i>Manuale Sviluppatore 1.0.0</i>	xx	<i>Superato</i>
<i>Specifica Tecnica 3.0.0</i>	70	<i>Superato</i>
<i>Norme di Progetto 4.0.0</i>	65	<i>Superato</i>
<i>Piano di Progetto 4.0.0</i>	63	<i>Superato</i>
<i>Piano di Qualifica 4.0.0</i>	66	<i>Superato</i>
<i>Analisi dei Requisiti 3.0.0</i>	71	<i>Superato</i>
<i>Glossario 3.0.0</i>	50	<i>Superato</i>
<i>Verbale Interno 2017-xx-yy</i>	xx	<i>Superato</i>
<i>Verbale Esterno 2017-xx-yy</i>	xx	<i>Superato</i>
<i>Verbale Interno 2017-xx-yy</i>	xx	<i>Superato</i>

Tabella 47: Resoconto verifiche documenti - RA

A.7 Riepilogo

Infine, per ogni documento principale, viene riportato un grafico che mostra la progressione e il rispettivo miglioramento del documento secondo l'*indice Gulpease*.

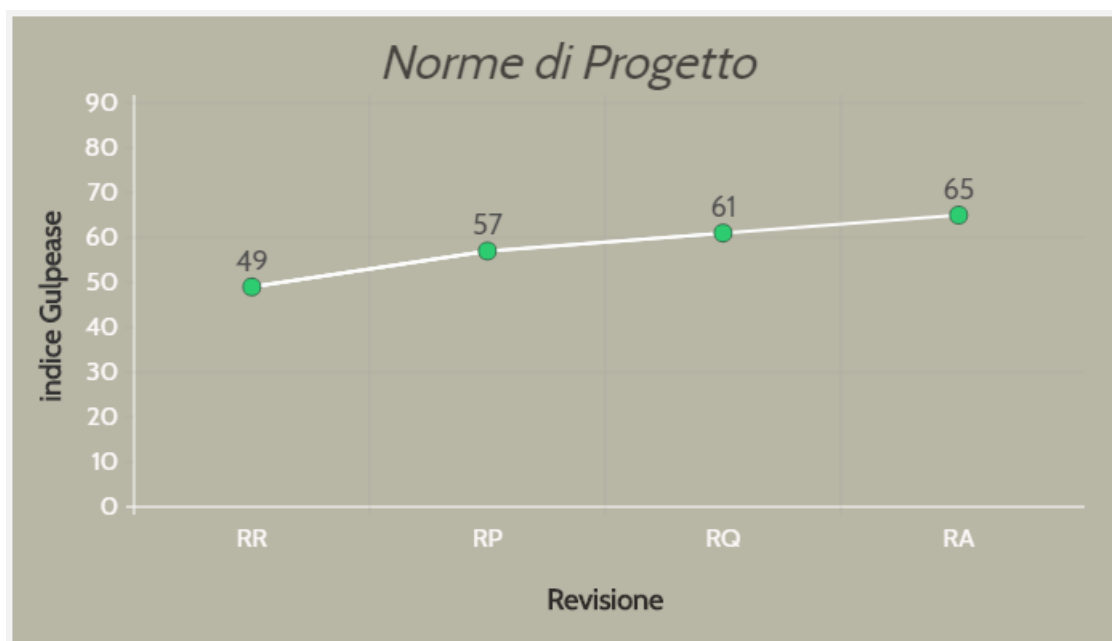


Figura 9: Miglioramento Norme di Progetto

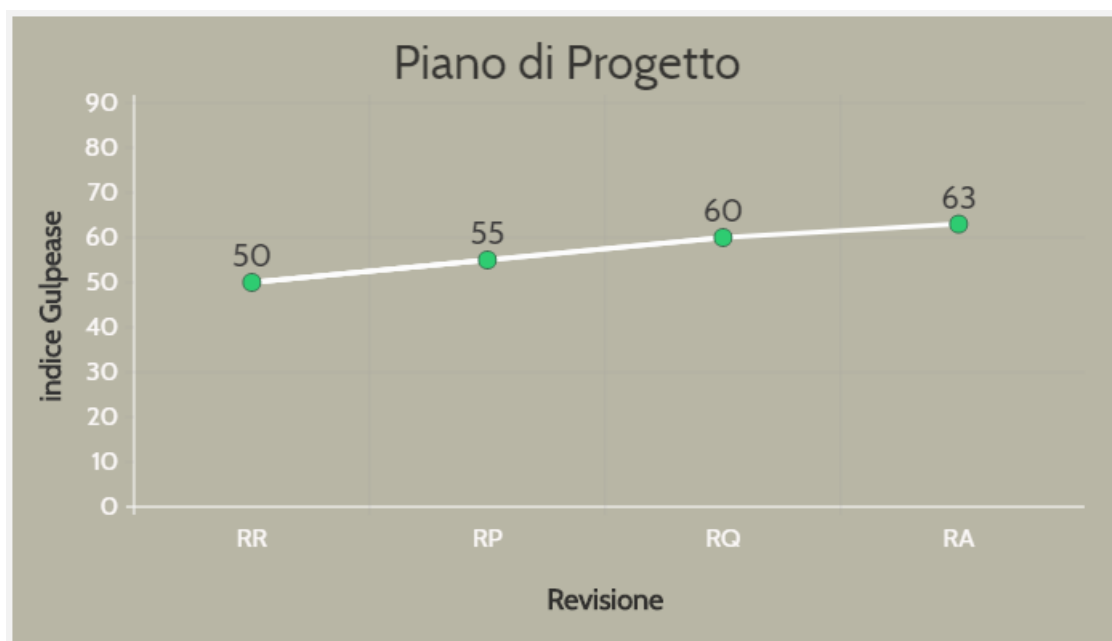


Figura 10: Miglioramento Piano di Progetto



Figura 11: Miglioramento Piano di Qualifica

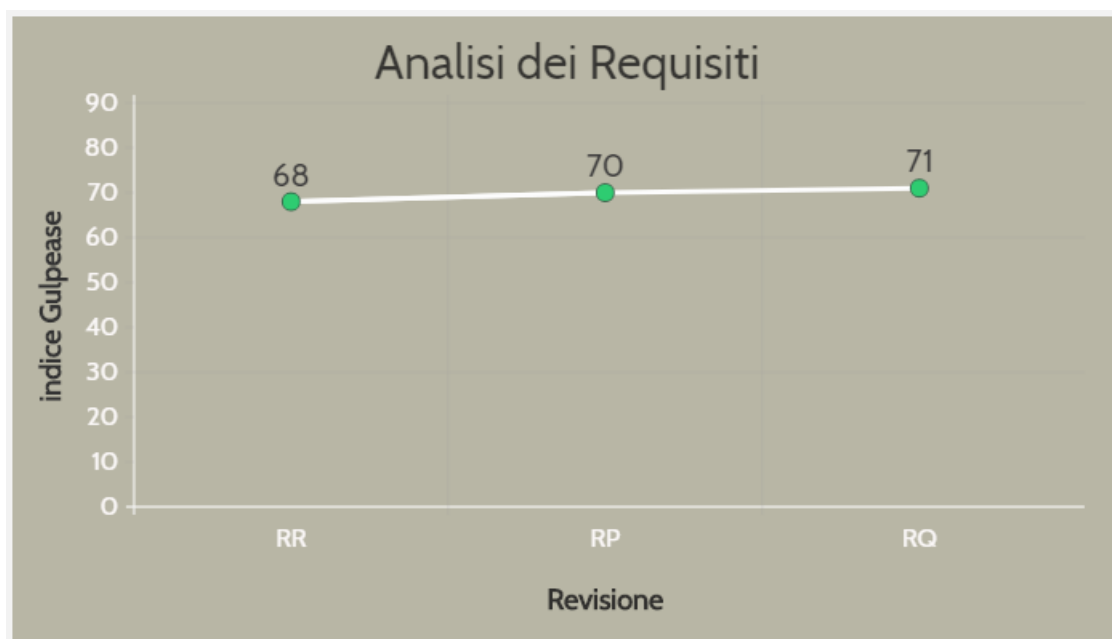


Figura 12: Miglioramento Analisi dei Requisiti

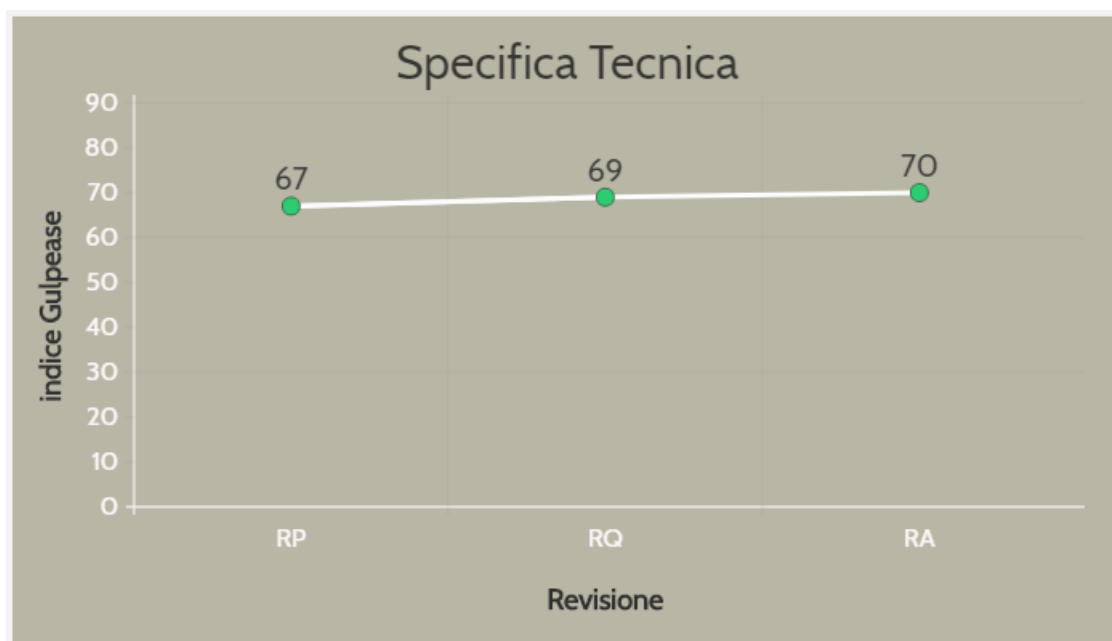


Figura 13: Miglioramento Specifica Tecnica



Figura 14: Miglioramento Definizione di Prodotto



Figura 15: Miglioramento Glossario

B Resoconto Qualità di Prodotto

In questa sezione del documento vengono descritti e analizzati gli esiti delle metriche previste per il controllo della qualità di prodotto, in base alla revisione di progetto.

B.1 Revisione di Qualifica

Id	Metrica	Valore	Esito
MPD_F1	<i>Completezza delle funzioni sviluppate</i>	93%	<i>Superato</i>
MPD_F2	<i>Correttezza delle funzioni sviluppate</i>	100%	<i>Superato</i>
MPD_F3	<i>Accuratezza rispetto alle aspettative</i>	91%	<i>Superato</i>
MPD_F4	<i>Controllo degli accessi</i>	91%	<i>Superato</i>
MPD_A1	<i>Chiamate a microservizi corrette</i>	96%	<i>Superato</i>
MPD_A2	<i>Copertura dei test</i>	83%	<i>Superato</i>
MPD_A3	<i>Controllo dei guasti</i>	84%	<i>Superato</i>
MPD_U1	<i>Comprensibilità delle funzionalità offerte</i>	85%	<i>Superato</i>
MPD_U2	<i>Controllo e monitoraggio delle operazioni</i>	89%	<i>Superato</i>
MPD_U3	<i>Qualità della messaggistica</i>	81%	<i>Superato</i>
MPD_E1	<i>Tempo di risposta</i>	4	<i>Superato</i>
MPD_M1	<i>Impatto delle modifiche</i>	20%	<i>Superato</i>
MPD_P1	<i>Supporto differenti versioni dei browser</i>	90%	<i>Superato</i>

Tabella 48: Resoconto esiti metriche - Qualità di prodotto
RQ

B.2 Revisione di Accettazione

Id	Metrica	Valore	Esito
MPD_F1	<i>Completezza delle funzioni sviluppate</i>	96%	<i>Superato</i>
MPD_F2	<i>Correttezza delle funzioni sviluppate</i>	100%	<i>Superato</i>

MPD_F3	<i>Accuratezza rispetto alle aspettative</i>	97%	<i>Superato</i>
MPD_F4	<i>Controllo degli accessi</i>	98%	<i>Superato</i>
MPD_A1	<i>Chiamate a microservizi corrette</i>	100%	<i>Superato</i>
MPD_A2	<i>Copertura dei test</i>	94%	<i>Superato</i>
MPD_A3	<i>Controllo dei guasti</i>	91%	<i>Superato</i>
MPD_U1	<i>Comprensibilità delle funzionalità offerte</i>	97%	<i>Superato</i>
MPD_U2	<i>Controllo e monitoraggio delle operazioni</i>	95%	<i>Superato</i>
MPD_U3	<i>Qualità della messaggistica</i>	92%	<i>Superato</i>
MPD_E1	<i>Tempo di risposta</i>	3	<i>Superato</i>
MPD_M1	<i>Impatto delle modifiche</i>	13%	<i>Superato</i>
MPD_P1	<i>Supporto differenti versioni dei browser</i>	97%	<i>Superato</i>

Tabella 49: Resoconto esiti metriche - Qualità di prodotto
RA

C Resoconto Qualità di Processo

In questa sezione del documento vengono descritti e analizzati gli esiti delle metriche previste per il controllo della qualità di processo, in base alla revisione di progetto.

La prima tabella contiene le metriche con associato esito applicate sull'intero sistema, mentre le successive tabelle sono una per ogni metrica che si applica a singole componenti.

N.B.: per una consultazione più semplice e rapida, per le metriche *Numero di metodi per classe*, *Numero di parametri per metodo* e *Numero di attributi per classe* viene riportata una semplice tabella indicante il valore minimo, massimo e medio individuati tra tutte le classe o metodi implementate/i.

Nel caso in cui il valore massimo e/o minimo non rientri nel range di accettazione della metrica in analisi, verrà indicata la classe o il metodo che necessita delle dovute correzioni, affinché rientri nel range di accettazione prestabilito.

C.1 Revisione di Qualifica

Id	Metrica	Valore	Esito
MPC1	<i>Disponibilità NetBreakDB</i>	95%	<i>Superato</i>
MPC2	<i>Schedule Variance</i>	0	<i>Superato</i>
MPC3	<i>Budget Variance</i>	75	<i>Superato</i>
MPC4	<i>Rischi non preventivati</i>	1	<i>Superato</i>
MPC5	<i>Adempimento requisiti obbligatori</i>	100%	<i>Superato</i>
MPC13	<i>Linee di commento</i>	20%	<i>Superato</i>
MPC14	<i>Test di Unità</i>	96%	<i>Superato</i>
MPC15	<i>Test di Integrazione</i>	70%	<i>Superato</i>
MPC16	<i>Test di Sistema</i>	80%	<i>Superato</i>
MPC18	<i>Test superati</i>	88%	<i>Superato</i>
MPC20	<i>Code Coverage</i>	67%	<i>Superato</i>

Tabella 50: Resoconto esiti metriche - Qualità di processo RQ

C.1.1 Numero di metodi per classe

La metrica *Numero di metodi per classe* (MPC8) si applica ad ogni classe progettata ed implementata.

Valore minimo	Valore massimo	Valore medio	Esito
1	15	5	<i>Superato</i>

Tabella 51: Resoconto esiti metrica Numero di metodi per classe - RQ

C.1.2 Numero di parametri per metodo

La metrica *Numero di parametri per metodo* (MPC9) si applica ad ogni metodo progettato ed implementato.

Valore minimo	Valore massimo	Valore medio	Esito
0	7	3	<i>Superato</i>

Tabella 52: Resoconto esiti metrica Numero di parametri per metodo - RQ

C.1.3 Numero di attributi per classe

La metrica *Numero di attributi per classe* (MPC10) si applica per ogni classe progettata ed implementata.

Valore minimo	Valore massimo	Valore medio	Esito
0	14	6	<i>Superato</i>

Tabella 53: Resoconto esiti metrica Numero di attributi per classe - RQ

C.2 Revisione di Accettazione

Id	Metrica	Valore	Esito
MPC1	<i>Disponibilità NetBreakDB</i>	98%	<i>Superato</i>
MPC2	<i>Schedule Variance</i>	0	<i>Superato</i>
MPC3	<i>Budget Variance</i>	xx	<i>Superato</i>
MPC4	<i>Rischi non preventivati</i>	1	<i>Superato</i>
MPC5	<i>Adempimento requisiti obbligatori</i>	100%	<i>Superato</i>
MPC13	<i>Linee di commento</i>	35%	<i>Superato</i>
MPC14	<i>Test di Unità</i>	100%	<i>Superato</i>
MPC15	<i>Test di Integrazione</i>	90%	<i>Superato</i>
MPC16	<i>Test di Sistema</i>	85%	<i>Superato</i>
MPC17	<i>Test di Validazione</i>	100%	<i>Superato</i>
MPC18	<i>Test superati</i>	94%	<i>Superato</i>
MPC20	<i>Code Coverage</i>	85%	<i>Superato</i>

Tabella 54: Resoconto esiti metriche - Qualità di processo RA

C.2.1 Fan In

La metrica *Fan In* (MPC6) si applica ad ogni modulo software per misurare e stabilire il livello di riuso implementato.

Per una consultazione più semplice e rapida, di seguito viene riportata una tabella indicante il valore minimo di *Fan In* individuato tra tutti i moduli implementati.

Valore minimo	Esito
2	<i>Superato</i>

Tabella 55: Resoconto esiti metrica Fan In - RA

C.2.2 Fan Out

La metrica *Fan Out* (MPC7) si applica ad ogni modulo software per misurare e stabilire il livello di accoppiamento implementato.

Per una consultazione più semplice e rapida, di seguito viene riportata una tabella indicante il valore massimo di *Fan Out* individuato tra tutti i moduli implementati.

Valore massimo	Esito
4	<i>Superato</i>

Tabella 56: Resoconto esiti metrica Fan Out - RA

C.2.3 Numero di metodi per classe

La metrica *Numero di metodi per classe* (MPC8) si applica ad ogni classe progettata ed implementata.

Valore minimo	Valore massimo	Valore medio	Esito
1	15	4	<i>Superato</i>

Tabella 57: Resoconto esiti metrica Numero di metodi per classe - RA

C.2.4 Numero di parametri per metodo

La metrica *Numero di parametri per metodo* (MPC9) si applica ad ogni metodo progettato ed implementato.

Valore minimo	Valore massimo	Valore medio	Esito
0	5	2	<i>Superato</i>

Tabella 58: Resoconto esiti metrica Numero di parametri per metodo - RA

C.2.5 Numero di attributi per classe

La metrica *Numero di attributi per classe* (MPC10) si applica per ogni classe progettata ed implementata.

Valore minimo	Valore massimo	Valore medio	Esito
0	13	4	<i>Superato</i>

Tabella 59: Resoconto esiti metrica Numero di attributi per classe - RA

C.2.6 Complessità Ciclomatica

La metrica *Complessità Ciclomatica* (MPC11) si applica a metodi, funzioni, moduli, classi, etc.. Per una consultazione più semplice e rapida, di seguito viene riportata una tabella indicante il valore massimo di *Complessità Ciclomatica*, individuato tra tutti/e i/le metodi, funzioni, moduli e classi implementati/e.

N.B.: nel caso in cui il valore massimo non rientri nel range di accettazione di tale metrica, verrà indicato/a il metodo/funzione/classe/modulo che necessita delle dovute correzioni.

Valore massimo	Esito
10	<i>Superato</i>

Tabella 60: Resoconto esiti metrica Complessità Ciclomatica - RA

C.2.7 Numero di livelli di annidamento

La metrica *Numero di livelli di annidamento* (MPC12) si applica ad ogni metodo implementato. Per una consultazione più semplice e rapida, di seguito viene riportata una tabella indicante il valore massimo di *Numero di livelli di annidamento*, individuato tra tutti i metodi implementati.

N.B.: nel caso in cui il valore massimo non rientri nel range di accettazione di tale metrica, verrà indicato il metodo (con relativa classe che lo contiene) che necessita delle dovute correzioni.

Valore massimo	Esito
6	<i>Superato</i>

Tabella 61: Resoconto esiti metrica Numero di livelli di annidamento - RA