



Piano di qualifica

| | |
|---------------------|---|
| Versione | 1.0.0 |
| Approvazione | Matteo Infantino |
| Redazione | Francesco Battistella Simone Innocente |
| Verifica | Andrea Polo |
| Stato | Approvato |
| Uso | Esterno |
| Destinato a | FourCats TealBlue Prof. Tullio Vardanega Prof. Riccardo Cardin |
| Email | fourcats.unipd@gmail.com |

Descrizione

Descrizione delle strategie di verifica e validazione adottate dal gruppo FourCats per garantire la qualità di prodotto e di processo nella realizzazione del capitolato NaturalApi



| Versione | Data | Nominativo | Ruolo | Descrizione | Verificatore | Esito |
|----------|------------|-----------------------|--------------|---|------------------|----------|
| 1.0.0 | 14-05-2020 | Simone Innocente | Verificatore | Aggiornata §4 | Andrea Polo | Positivo |
| | 10-05-2020 | Simone Innocente | Verificatore | Aggiornato Appendice A | Andrea Polo | Positivo |
| | 05-05-2020 | Francesco Battistella | Verificatore | Aggiornato Appendice B | Andrea Polo | Positivo |
| 0.9.0 | 28-04-2020 | Matteo Munari | Verificatore | Aggiornata §4 | Edoardo Tinto | Positivo |
| | 25-04-2020 | Edoardo Tinto | Verificatore | Stesura §4 - Test di integrazione | Matteo Munari | Positivo |
| | 23-04-2020 | Francesco Battistella | Verificatore | Aggiornata appendice B | Matteo Infantino | Positivo |
| | 22-04-2020 | Matteo Munari | Verificatore | Stesura §4 - Test di unità | Matteo Infantino | Positivo |
| 0.0.2 | 06-03-2020 | Simone Innocente | Verificatore | Aggiornati appendici A e B | Edoardo Tinto | Positivo |
| | 06-03-2020 | Giulio Umbrella | Verificatore | Rimosso appendice A, rinominati appendici B in A e C in B | Edoardo Tinto | Positivo |
| | 13-02-2020 | Giulio Umbrella | Verificatore | Aggiornata §4 - Test di sistema | Matteo Munari | Positivo |
| | 08-02-2020 | Simone Innocente | Verificatore | Aggiornata §3 - Qualità di prodotto | Matteo Munari | Positivo |
| | 05-02-2020 | Simone Innocente | Verificatore | Aggiornata §2 - Qualità di processo | Matteo Munari | Positivo |
| | 27-01-2020 | Francesco Battistella | Verificatore | Correzioni generali secondo suggerimenti RR | Matteo Munari | Positivo |
| 0.0.1 | 11-01-2020 | Simone Innocente | Verificatore | Stesura Appendice C - Valutazioni per il miglioramento del team | Andrea Polo | Positivo |

| | | | | | |
|------------|-----------------------|--------------|--|---------------|----------|
| 11-01-2020 | Simone Innocente | Verificatore | Stesura Appendice B - Resoconto attività di verifica | Andrea Polo | Positivo |
| 28-01-2020 | Matteo Munari | Verificatore | Verifica §4 - Test di sistema | Andrea Polo | Positivo |
| 28-12-2019 | Giulio Umbrella | Verificatore | Conclusa §4 - Test di sistema | Andrea Polo | Positivo |
| 24-12-2019 | Giulio Umbrella | Verificatore | Inizio §4 - Test di sistema | Andrea Polo | Positivo |
| 11-12-2019 | Simone Innocente | Verificatore | Aggiunte ad appendice A - Ciclo di Deming | Andrea Polo | Positivo |
| 09-12-2019 | Francesco Battistella | Verificatore | Conclusa §3 - Qualità di prodotto | Matteo Munari | Positivo |
| 07-12-2019 | Simone Innocente | Verificatore | Aggiornati riferimenti della §1 - Introduzione | Matteo Munari | Positivo |
| 07-12-2019 | Francesco Battistella | Verificatore | Aggiunto appendice A - ISO/IEC 9126 | Matteo Munari | Positivo |
| 06-12-2019 | Francesco Battistella | Verificatore | Iniziata stesura §3 - Qualità di Prodotto | Matteo Munari | Positivo |
| 05-12-2019 | Simone Innocente | Verificatore | Conclusa stesura §2 - Qualità di processo | Matteo Munari | Positivo |
| 04-12-2019 | Francesco Battistella | Verificatore | Modifiche §1 - Introduzione | Matteo Munari | Positivo |
| 01-12-2019 | Simone Innocente | Verificatore | Iniziata stesura §2 - Qualità di processo | Matteo Munari | Positivo |
| 01-12-2019 | Simone Innocente | Verificatore | Stesura §1 - Introduzione | Matteo Munari | Positivo |
| 01-12-2019 | Simone Innocente | Verificatore | Creata struttura documento | Matteo Munari | Positivo |

Indice

| | | |
|----------|---|-----------|
| 1 | Introduzione | 5 |
| 1.1 | Scopo del documento | 5 |
| 1.2 | Scopo del prodotto | 5 |
| 1.3 | Riferimenti | 5 |
| 1.3.1 | Riferimenti normativi | 5 |
| 1.3.2 | Riferimenti informativi | 5 |
| 2 | Qualità di processo | 6 |
| 2.1 | Tabella delle metriche di processo | 6 |
| 3 | Qualità di prodotto | 7 |
| 3.1 | Prodotti | 7 |
| 3.2 | Software | 7 |
| 3.3 | Tabella delle metriche di prodotto | 7 |
| 4 | Specifica dei test | 9 |
| 4.1 | Test di Accettazione | 9 |
| 4.1.1 | Specifiche | 9 |
| 4.2 | Test di Sistema | 14 |
| 4.2.1 | Specifiche | 14 |
| 4.3 | Test di Integrazione | 24 |
| 4.3.1 | Specifiche dei test ad ampio scope | 24 |
| 4.3.2 | Specifiche dei test a scope ridotto | 25 |
| 4.4 | Test di Unità | 27 |
| 4.4.1 | Specifiche | 27 |
| A | Resoconto attività di verifica | 40 |
| A.1 | Revisione dei requisiti | 40 |
| A.1.1 | Documenti | 40 |
| A.1.1.1 | Indice di Gulpease | 40 |
| A.1.2 | Processi | 40 |
| A.1.3 | Software | 40 |
| A.2 | Revisione di progettazione | 41 |
| A.2.1 | Documenti | 41 |
| A.2.1.1 | Indice di Gulpease | 41 |
| A.2.2 | Processi | 42 |
| A.2.2.1 | EAC: Estimated at Completion | 42 |
| A.2.2.2 | VAC: Variance at Completion | 42 |
| A.2.2.3 | BV: Budget Variance | 43 |
| A.2.2.4 | AC: Actual Cost | 44 |
| A.2.3 | Software | 45 |
| A.3 | Revisione di qualifica | 46 |
| A.3.1 | Documenti | 46 |
| A.3.1.1 | Indice di Gulpease | 46 |
| A.3.2 | Processi | 47 |
| A.3.2.1 | EAC: Estimated at Completion | 47 |
| A.3.2.2 | VAC: Variance at Completion | 47 |
| A.3.2.3 | BV: Budget Variance | 48 |
| A.3.2.4 | AC: Actual Cost | 48 |
| A.3.3 | Software | 49 |
| A.3.3.1 | Legenda | 49 |
| A.3.3.2 | WMC: Metodi pesati per classe | 49 |

| | | |
|----------|---|-----------|
| A.3.3.3 | DIT: Profondità dell'albero di ereditarietà | 49 |
| A.3.3.4 | NOC: Numero di figli | 49 |
| A.3.3.5 | RFC: Risposta per classe | 50 |
| A.3.3.6 | LCOM: Mancanza di coesione nei metodi | 50 |
| A.3.3.7 | CBO: Accoppiamento tra classi | 51 |
| A.3.3.8 | SonarQube | 51 |
| A.4 | Revisione di accettazione | 52 |
| A.4.1 | Documenti | 52 |
| A.4.1.1 | Indice di Gulpease | 52 |
| A.4.2 | Processi | 53 |
| A.4.2.1 | EAC: Estimated at Completion | 53 |
| A.4.2.2 | VAC: Variance at Completion | 53 |
| A.4.2.3 | BV: Budget Variance | 54 |
| A.4.2.4 | AC: Actual Cost | 55 |
| A.4.3 | Software | 56 |
| A.4.3.1 | Requisiti | 56 |
| A.4.3.2 | NaturalAPI | 57 |
| A.4.3.3 | CBO: Accoppiamento tra classi | 59 |
| A.4.3.4 | NaturalAPI Discover | 60 |
| A.4.3.5 | NaturalAPI Design | 63 |
| A.4.3.6 | NaturalAPI Develop | 66 |
| B | Valutazioni per il miglioramento del team | 69 |
| B.1 | Revisione dei requisiti | 69 |
| B.1.1 | Valutazioni sull'organizzazione del team | 69 |
| B.1.2 | Valutazione sugli strumenti di lavoro | 70 |
| B.2 | Revisione di progettazione | 70 |
| B.2.1 | Valutazione sull'organizzazione del team | 70 |
| B.2.2 | Valutazione sugli strumenti di lavoro | 71 |
| B.3 | Revisione di qualifica | 71 |
| B.3.1 | Valutazione sull'organizzazione del team | 71 |
| B.3.2 | Valutazione sui ruoli | 71 |
| B.3.3 | Valutazione sugli strumenti di lavoro | 72 |
| B.4 | Revisione di accettazione | 72 |
| B.4.1 | Valutazione sull'organizzazione del team | 72 |
| B.4.2 | Valutazione sui ruoli | 72 |
| B.4.3 | Valutazione sugli strumenti di lavoro | 74 |

1 Introduzione

1.1 Scopo del documento

Il Piano di qualifica ha lo scopo di illustrare le strategie di verifica e validazione adottate per garantire la qualità di prodotto e di processo. Per raggiungere tale obiettivo viene applicato un sistema di verifica continua sui processi e sulle attività svolte. Questo consente di rilevare e correggere rapidamente eventuali anomalie riducendo al minimo lo spreco di risorse.

1.2 Scopo del prodotto

Il fine del progetto è la creazione di un *toolkit*, NaturalAPI, in grado di generare automaticamente application programming interfaces (*APIs*) e i relativi unit test per un dato linguaggio di programmazione, partendo da feature e scenari in formato *Gherkin* e da documenti di testo inerenti al dominio di interesse.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- Norme di progetto (NP.1.0.0);
- Capitolato NaturalAPI: <https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C3.pdf>.

1.3.2 Riferimenti informativi

- ISO/IEC 9126: https://it.wikipedia.org/wiki/ISO/IEC_9126
- Indice di Gulpease - Wikipedia: https://it.wikipedia.org/wiki/Indice_Gulpease
- Schedule Variance and Cost Variance:
<https://pmstudycircle.com/2012/05/schedule-variance-sv-cost-variance-cv-in-project-cost-management>
- Clean Architecture A Craftsman's Guide to Software Structure and Design - Robert C. Martin Series - PART V Architecture - ISBN: 978-0134494166
- Ingegneria del software Ian Sommerville - Decima Edizione - ISBN-13: 978-8891902245 - Capitolo 21 Gestione della qualità
- CodeMR.com - Appendix A: Supported Metric List <https://www.codemr.co.uk/documents/#h.ewgasege54gx>
- CacheQuality.com - Metrics <https://www.cachequality.com/docs/cachequality-release/metrics>
- CacheQuality.com - Depth inheritance tree <https://www.cachequality.com/docs/metrics/depth-inheritance-tree>
- mainsqual.squoring.com <https://maisqual.squoring.com/wiki/index.php/Category:Metrics>

2 Qualità di processo

La qualità di un prodotto è influenzata enormemente dai processi utilizzati nell'arco del ciclo di vita del prodotto stesso.

È quindi essenziale ricercare la qualità dei processi in modo sistematico, disciplinato e quantificabile.

Inizialmente i processi, verificati e valutati attentamente, sono stati scelti tra quelli proposti nello standard ISO/IEC 12207:1995. In seguito sono stati adattati secondo le esigenze del gruppo.

Diventa quindi necessario operare con un buon ciclo di vita e una buona gestione del controllo e miglioramento continuo. A questo scopo viene seguito lo schema del ciclo di Deming.

Le metriche adoperate per perseguire la loro qualità sono esposte nella tabella di seguito.

2.1 Tabella delle metriche di processo

| Metrica | Valore Preferibile | Valore Accettabile |
|-------------------------------------|---------------------|---|
| Processo di gestione | | |
| <i>BAC: Budget At Completion</i> | = al preventivo | $\text{preventivo} - 5\% \leq BAC \leq \text{preventivo} + 5\%$ |
| <i>SV: Schedule Variance</i> | ≥ 0 | 0 |
| <i>AC: Actual Cost</i> | $0 \leq AC \leq PV$ | $0 \leq AC \leq \text{budget totale}$ |
| <i>CV: Cost Variance</i> | > 0 | ≥ 0 |
| <i>BV: Budget Variance</i> | > 0 | = 0 |
| <i>EAC: Estimated at Completion</i> | < 0 | = 0 |
| <i>VAC: Variance at Completion</i> | > 0 | = 0 |

Tabella 2: Tabella delle metriche di processo

3 Qualità di prodotto

3.1 Prodotti

Con il termine prodotto è inteso tutto ciò che è concretamente utilizzabile, consultabile o eseguibile. Nel contesto di questo progetto i prodotti sono documenti e software.

3.2 Software

Per garantire la qualità del prodotto il team FourCats ha voluto prendere come riferimento lo standard ISO/IEC 9126. Questo standard fornisce un modello per valutare tutte le caratteristiche da tenere in considerazione per produrre un prodotto di buona qualità. Le caratteristiche sono descritte attraverso dei parametri, che ne quantificano il grado di raggiungimento. Nella tabella di seguito vengono descritte le voci che ad oggi, per il gruppo, sono rilevanti nel contesto del progetto. È stato inoltre deciso di utilizzare lo strumento *SonarQube* per tenere traccia di alcune di queste.

3.3 Tabella delle metriche di prodotto

| Metrica | Valore Preferibile | Valore Accettabile |
|---|--------------------|--------------------|
| Funzionalità | | |
| <i>Completezza dell'implementazione</i> | 100% | 100% |
| <i>PROS: Requisiti obbligatori soddisfatti</i> | 100% | 100% |
| <i>PRD: Requisiti desiderabili</i> | 100% | 80% |
| <i>PROD: Requisiti opzionali e desiderabili</i> | 100% | ≥ 80% |
| Affidabilità | | |
| <i>Condition Coverage</i> | 100% | ≥ 70% |
| <i>Line Coverage</i> | 100% | ≤ 70% |
| <i>Lines to Cover</i> | 0 | ≤ 30% |
| <i>Uncovered Lines</i> | 0 | ≤ 15% |
| <i>Coverage</i> | 90% | ≥ 75% |
| <i>Unit Test Errors</i> | 0 | 0 |
| <i>Unit Test Failures</i> | 0 | 0 |
| <i>Unit Test Success Density</i> | 100 | 100 |
| Manutenibilità | | |
| <i>Facilità di comprensione</i> | ≥ 20 | ≥ 10 |
| <i>Complessità ciclomatica</i> | 0-15 | 0-30 |

| | | |
|--|-----------------------|------------------------|
| <i>CBO: Accoppiamento tra classi</i> | $0 \leq CBO \leq 2$ | $0 \leq CBO \leq 5$ |
| <i>Code Smell</i> | 0 | 30 |
| <i>Technical Debt Ratio</i> | 0% | Maintainability rating |
| <i>Maintainability rating</i> | A | $\leq C$ |
| <i>WMC: Metodi pesati per classe</i> | ≤ 20 | ≤ 30 |
| <i>DIT: Profondità dell'albero di ereditarietà</i> | ≤ 3 | ≤ 4 |
| <i>NOC: Numero di figli</i> | ≤ 3 | ≤ 4 |
| <i>RFC: Risposta per classe</i> | ≤ 50 | ≤ 75 |
| <i>LCOM: Mancanza di coesione nei metodi</i> | 0 | $\leq 0,5$ |
| Documenti | | |
| <i>Indice di Gulpease</i> | $80 \leq IG \leq 100$ | $50 \leq IG \leq 100$ |

Tabella 3: Tabella delle metriche di prodotto

4 Specifica dei test

4.1 Test di Accettazione

I test di accettazione sono concordati con il proponente e hanno lo scopo di verificare la soddisfazione dell'utente che utilizza il prodotto completo. I test seguono la codifica definita nel documento NP.1.0.0.

4.1.1 Specifiche

| Test | Descrizione | Tool di appartenenza | Stato |
|-------|--|----------------------|-------|
| TA1.1 | L'utente deve poter creare una BDL. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare dei documenti di testo;• Decidere la cartella di destinazione della BDL;• Dare un nome alla BDL;• Visualizzare la BDL. | Discover | I |
| TA2.1 | L'utente deve poter aggiungere dei documenti ad una BDL. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare dei documenti di testo;• Scegliere una BDL esistente;• Visualizzare la BDL;• Verificare che la BDL sia stata aggiornata. | Discover | I |
| TA3.1 | L'utente deve poter rimuovere dei documenti da una BDL. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare dei documenti di testo;• Scegliere una BDL esistente;• Visualizzare la BDL;• Verificare che la BDL sia stata aggiornata. | Discover | I |

| Test | Descrizione | Tool di appartenenza | Stato |
|--------|--|----------------------|-------|
| TA6.1 | L'utente deve poter generare i suggerimenti per la creazione di una BAL. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare dei file .feature in formato Gherkin;• Selezionare una BDL (opzionale);• Generare i suggerimenti. | Design | I |
| TA6.2 | L'utente deve poter visualizzare un messaggio di errore nel caso in cui non abbia inserito nessun file .feature e si tenta di creare i suggerimenti | Design | I |
| TI8.1 | L'utente deve poter rifiutare i suggerimenti ricevuti. All'utente viene chiesto di: <ul style="list-style-type: none">• Eliminare un suggerimento;• Confermare l'eliminazione;• Verificare che il suggerimento non sia più presente. | Design | I |
| TA9.1 | L'utente deve poter modificare il nome di una action. All'utente viene chiesto di: <ul style="list-style-type: none">• Modificare il nome di una action;• Verificare che la modifica sia stata accettata. | Design | I |
| TA9.2 | L'utente deve poter visualizzare un messaggio di errore se il nome della action modificato è uguale ad un nome di un'altra action. | Design | I |
| TA9.3 | L'utente deve poter modificare il tipo di ritorno di una action. All'utente viene chiesto di: <ul style="list-style-type: none">• Modificare il tipo di ritorno di una action;• Verificare che la modifica sia stata accettata. | Design | I |
| TA10.1 | L'utente deve poter creare un tipo custom. All'utente viene chiesto di: <ul style="list-style-type: none">• Inserire il nome per il tipo custom;• Verificare che il tipo custom sia stato creato. | Design | I |

| Test | Descrizione | Tool di appartenenza | Stato |
|--------|---|----------------------|-------|
| TA10.1 | L'utente deve poter inserire il nome e il tipo per ogni attributo del nuovo tipo custom. All'utente viene chiesto di: <ul style="list-style-type: none">• Inserire il nome del nuovo attributo;• Inserire il tipo del nuovo attributo;• Verificare che l'attributo venga aggiunto correttamente. | Design | I |
| TA10.2 | L'utente deve poter visualizzare un messaggio di errore se si tenta di creare un tipo custom con nome uguale ad un altro tipo esistente | Design | I |
| TA11.1 | L'utente deve poter inserire un nuovo parametro. All'utente viene chiesto di: <ul style="list-style-type: none">• Inserire il nome del nuovo parametro;• Inserire il tipo del nuovo parametro;• Verificare che venga aggiunto correttamente il nuovo parametro. | Design | I |
| TA11.2 | L'utente deve poter visualizzare un messaggio di errore se si tenta di creare un parametro con un nome già presente. | Design | I |
| TA12.1 | L'utente deve poter eliminare un parametro presente. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare un parametro da eliminare;• Verificare che il parametro non sia più visualizzato. | Design | I |
| TA13.1 | L'utente deve poter modificare un parametro. All'utente viene chiesto di: <ul style="list-style-type: none">• Modificare il nome di un parametro;• Modificare il tipo di un parametro. | Design | I |

| Test | Descrizione | Tool di appartenenza | Stato |
|--------|--|----------------------|-------|
| TA18.1 | <p>L'utente deve poter generare un BAL. All'utente viene chiesto di:</p> <ul style="list-style-type: none"> • Selezionare i file .feature; • Selezionare un BDL (opzionale); • Richiedere i suggerimenti per il BAL; • Generare il BAL. | Design | I |
| TA19.1 | <p>L'utente deve poter aggiungere nuove feature e scenari ad un BAL. All'utente viene chiesto di:</p> <ul style="list-style-type: none"> • Selezionare i file .feature; • Generare i suggerimenti per il BAL. | Design | I |
| TA20.1 | <p>L'utente deve poter generare dei suggerimenti per le API. All'utente viene chiesto di:</p> <ul style="list-style-type: none"> • Selezionare una BAL; • Selezionare un PLA; • Generare i suggerimenti per le API. | Develop | I |
| TA20.2 | <p>L'utente deve poter visualizzare un messaggio di errore nel momento in cui si cerca di generare dei suggerimenti per le API senza aver selezionato un BAL e un PLA</p> | Develop | I |
| TA21.1 | <p>L'utente deve poter creare un PLA. All'utente viene chiesto di:</p> <ul style="list-style-type: none"> • Inserire l'estensione dei file che verranno generati; • Inserire il template per l'API; • Inserire il template per un tipo custom; • Inserire il template per un test; • Selezionare la cartella di destinazione. | Develop | I |

| Test | Descrizione | Tool di appartenenza | Stato |
|--------|---|----------------------|-------|
| TA22.1 | L'utente deve poter modificare un PLA. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare un PLA esistente;• Modificare il template per l'API;• Modificare il template per un tipo custom;• Modificare il template per un test. | Develop | I |
| TA24.1 | L'utente deve poter generare i suggerimenti per le API. All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare un BAL;• Selezionare un PLA;• Visualizzare i suggerimenti ricevuti. | Develop | I |
| TA24.2 | L'utente deve poter modificare un suggerimenti per le API. All'utente viene chiesto di: <ul style="list-style-type: none">• Visualizzare i suggerimenti;• Modificare i suggerimenti. | Develop | I |
| TA25.1 | L'utente deve poter generare le API e i Test All'utente viene chiesto di: <ul style="list-style-type: none">• Selezionare la cartella di destinazione delle API;• Generare le API e i test. | Develop | I |

Tabella 4: Tabella test di accettazione

4.2 Test di Sistema

I test di sistema hanno lo scopo di verificare che il sistema soddisfi a pieno i requisiti e rispetti le specifiche definite nell'Analisi dei requisiti. Viene controllata l'interazione tra le parti e tutte le funzionalità del sistema. I test seguono la codifica definita nel documento NP.1.0.0.

4.2.1 Specifiche

Tabella 5: Tabella dei test di sistema

| Test | Requisito | Obiettivo | Stato | Esempi |
|------|-----------|--|-------|--|
| TSF1 | RF1 | Se un utente seleziona dei documenti, ottiene un BDL | S | <ul style="list-style-type: none">• Se l'utente non seleziona nessun documento, il <i>BDL</i> è vuoto;• Se l'utente seleziona dei documenti, il <i>BDL</i> non è vuoto. |
| TSF2 | RF1.1 | Verificare che il <i>BDL</i> prodotto contenga un insieme di nomi e la relativa frequenza | S | |
| TSF3 | RF1.2 | Verificare che il <i>BDL</i> prodotto contenga un insieme di verbi e la relativa frequenza | S | |
| TSF4 | RF1.3 | Verificare che il <i>BDL</i> prodotto contenga un insieme di predicati e la relativa frequenza | S | |
| TSF5 | RF2 | Verificare che un documento in una lingua diversa venga tradotto | NI | |
| TSF6 | RF2.1 | Verificare che i documenti in un linguaggio naturale diverso dall'inglese vengano tradotti | NI | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|---|-------|---|
| TSF7 | RF2.1 | Verificare che un documento il quale ha <i>feature</i> e <i>scenari</i> in linguaggio diverso dall'inglese vengano tradotti, mantenendo il formato <i>Gherkin</i> | NI | |
| TSF8 | RF3 | Verificare che la lista di scelta, fatta dall'utente, venga inserita correttamente nel sistema | S | <ul style="list-style-type: none"> • Nessuna scelta porta ad una lista vuota |
| TSF9 | RF3.1 | Dato un <i>BDL</i> e un insieme di documenti, è possibile associare al <i>BDL</i> i documenti usati per produrlo | S | <ul style="list-style-type: none"> • Se il BDL non esiste o è vuoto, non viene associato nessun documento |
| TSF10 | RF3.2 | Dato un <i>BDL</i> e un insieme di documenti, il sistema non riapplica il parser sui documenti già analizzati | NI | <ul style="list-style-type: none"> • Data una lista di documenti da analizzare, controllare che siano tutti associati al BDL |
| TSF11 | RF3.3 | Dato un <i>BDL</i> , è possibile integrarlo con dei nuovi documenti | S | <ul style="list-style-type: none"> • Se non fornisco documenti, il BDL non cambia. Dato dei documenti di cui conosco già il risultato dell'analisi del parser, si controlla che il BDL aggiornato contenga i valori conteggiati. |
| TSF12 | RF3.3 | Speculare a TSF11 | S | |
| TSF13 | RF4 | Il <i>parser</i> utilizzato da <i>NaturalAPI Discover</i> corrisponde a quello scelto dall'utente; | S | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|--|
| TSF14 | RF5 | I suggerimenti per le <i>feature</i> vengono salvati correttamente | S | <ul style="list-style-type: none"> • Se l'utente non accetta una modifica suggerita, il testo di una feature non cambia; • Se l'utente accetta una modifica suggerita, il testo di una <i>feature</i> cambia. |
| TSF15 | RF6 | <i>NaturalAPI Design</i> deve avere gli input necessari per poter funzionare | S | <ul style="list-style-type: none"> • Controllare che con <i>BDL</i>, <i>feature</i> e <i>BO</i>, <i>NaturalAPI Design</i> si avvii; • Controllare che con <i>BDL</i> e <i>feature</i>, <i>NaturalAPI Design</i> si avvii; • Controllare combinazione di input che non fanno partire <i>NaturalAPI Design</i> (e.g. <i>BDL</i> e <i>BO</i> ma senza <i>feature</i>). |
| TSF16 | RF7 | <i>NaturalAPI Design</i> gestisce le <i>feature</i> selezionate dall'utente | S | <ul style="list-style-type: none"> • Se l'utente non seleziona nessuna <i>feature</i>, la lista è vuota; • Se l'utente seleziona un numero arbitrario di <i>feature</i>, tutte e solo quelle che l'utente seleziona, vengono conteggiate e le altre vengono scartate. |
| TSF17 | RF8 | <i>NaturalAPI Design</i> gestisce correttamente la scelta del <i>BDL</i> | S | <ul style="list-style-type: none"> • Se non scelgo nessun <i>BDL</i>, <i>NaturalAPI Design</i> non ne rileva nessuno; • Se scelgo più di un <i>BDL</i> viene segnalato un errore. |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|--|
| TSF18 | RF9 | La scelta della BO rimane opzionale | S | |
| TSF19 | RF10 | <i>NaturalAPI Design</i> produce un elenco di suggerimenti a partire dagli input forniti dall'utente | S | |
| TSF20 | RF10.1 | NaturalAPI fornisce almeno un suggerimento per ogni <i>scenario</i> o un avviso di errore | S | <ul style="list-style-type: none"> • Uno <i>scenario</i> vuoto non produce nessun suggerimento. |
| TSF21 | RF10.2 | I suggerimenti forniti da NaturalAPI hanno elementi riconducibili ad <i>action</i> e <i>object</i> . Se ci sono degli elementi estranei ottengo un errore. | S | <ul style="list-style-type: none"> • Se il suggerimento prodotto contiene solo <i>object</i> o <i>action</i>, il test è superato; • Se il suggerimento prodotto contiene elementi che non sono object o <i>action</i>, ottengo un errore. |
| TSF22 | RF11 | I suggerimenti confermati dall'utente si possono ritrovare nel <i>BAL</i> | S | <ul style="list-style-type: none"> • Se l'utente non conferma nulla, il <i>BAL</i> è vuoto; • Se l'utente conferma solo <i>n</i> suggerimenti, tutti e solo quelli che sono selezionati, vengono conteggiati; • Se l'utente seleziona solo alcuni suggerimenti, vengono conteggiati solo quelli e gli altri vengono scartati. |
| TSF23 | RF11.1 | Se un suggerimento è accettato, allora viene contrassegnato come accettato | S | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|---|-------|---|
| TSF24 | RF12 | Quando un utente rifiuta un suggerimento, non viene inserito nel <i>BAL</i> (speculare TSF21) | S | |
| TSF25 | RF12.1 | Se un suggerimento è rifiutato, allora viene contrassegnato come rifiutato | S | |
| TSF26 | RF13 | Dato un suggerimento, l'utente lo può modificare e i cambiamenti sono persistenti. | S | |
| TSF27 | RF13.1 | Data un' <i>action</i> in un suggerimento, l'utente può modificarne il nome | S | <ul style="list-style-type: none"> • Se l'utente non cambia il nome, il nome del suggerimento nel <i>BAL</i> è invariato; • Se l'utente cambia il nome, il nome del suggerimento nel <i>BAL</i>, riporta il nuovo nome. |
| TSF28 | RF13.2 | Dato un suggerimento, l'utente può aggiungere degli <i>object</i> | S | <ul style="list-style-type: none"> • Se l'utente non aggiunge <i>object</i> al suggerimento, il suggerimento nel <i>BAL</i> ha gli stessi <i>object</i> originali; • Se l'utente aggiunge <i>object</i> al suggerimento, il suggerimento nel <i>BAL</i> ha nuovi <i>object</i>. |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|---|
| TSF29 | RF13.3 | Dato un suggerimento, l'utente può rimuovere un <i>object</i> | S | <ul style="list-style-type: none">• Se l'utente non rimuove <i>object</i> al suggerimento, il suggerimento nel <i>BAL</i> ha gli stessi <i>object</i> originali;• Se l'utente rimuove <i>object</i> al suggerimento, il suggerimento nel <i>BAL</i> non ha gli stessi <i>object</i> rimossi. |
| TSF30 | RF13.4 | Dato un <i>object</i> in un suggerimento, l'utente può modificarne il nome | S | <ul style="list-style-type: none">• Se l'utente non modifica il nome dell'<i>object</i> al suggerimento, il nome nel suggerimento nel <i>BAL</i> ha il nome originale;• Se l'utente modifica il nome dell'<i>object</i> al suggerimento, il nome nel suggerimento nel <i>BAL</i> ha il nuovo nome. |
| TSF31 | RF13.5 | Dato un <i>object</i> in un suggerimento, l'utente può modificarne il tipo | S | <ul style="list-style-type: none">• Se l'utente non modifica il tipo dell'<i>object</i> al suggerimento, il tipo nel suggerimento nel <i>BAL</i> ha il nome del tipo;• Se l'utente modifica il tipo dell'<i>object</i> al suggerimento, il tipo nel suggerimento nel <i>BAL</i> ha il nuovo tipo. |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|---|
| TSF32 | RF14 | L'utente può eseguire il merge dei suggerimenti proposti dal <i>NaturalAPI Design</i> e ricavarne uno solo | NI | <ul style="list-style-type: none"> • Se unisco due suggerimenti, il numero dei suggerimenti diminuisce di uno; • Se unisco un solo suggerimento ottengo un errore. |
| TSF33 | RF14.1 | Ogni suggerimenti ha come elemento base una <i>action</i> . Se l'utente unisce due suggerimenti, la <i>action</i> risultante deve avere il nome che è stato scelto | NI | |
| TSF34 | RF14.2 | I suggerimenti sono <i>action</i> con dentro <i>object</i> . Se l'utente unisce dei suggerimenti, il suggerimento contiene <i>action</i> e <i>object</i> degli originali | NI | <ul style="list-style-type: none"> • Dati due suggerimenti senza <i>object</i>, il suggerimento ottenuto dall'unione non ha <i>object</i>; • Dati due suggerimenti, la loro unione ha solo <i>action</i> e <i>object</i> presenti nei suggerimenti originali. |
| TSF35 | RF15 | Verificare la suddivisione del suggerimento | NI | |
| TSF36 | RF15.1 | Data la suddivisione di un suggerimento, verificare che i suggerimenti ottenuti siano lo stesso numero di quelli impostati dall'utente | NI | |
| TSF37 | RF15.2 | Data una suddivisione di un suggerimento, verificare che i nuovi suggerimenti abbiano i nomi impostati dall'utente | NI | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|---|
| TSF38 | RF15.3 | Verificare che gli <i>object</i> siano suddivisi tra i nuovi suggerimenti creati | NI | |
| TSF39 | RF16 | L'utente può creare dei gruppi | NI | <ul style="list-style-type: none"> • Un gruppo vuoto non ha suggerimenti associati; • Un gruppo che non esiste non ha suggerimenti associati; • Dato un gruppo, è possibile ricavare tutti i suggerimenti ad esso associati e non. |
| TSF40 | RF16.1 | L'utente assegna un nome al gruppo | NI | |
| TSF41 | RF16.2 | Se un utente elimina un gruppo, non è più presente fra i gruppi disponibili | NI | <ul style="list-style-type: none"> • Un gruppo eliminato non può avere suggerimenti associati; • Un gruppo eliminato non è più presente fra i gruppi disponibili. |
| TSF42 | RF16.3 | Verificare che un' <i>action</i> appartenga ad un gruppo dopo l'assegnazione | NI | |
| TSF43 | RF16.4 | Verificare che un' <i>action</i> rimossa da un gruppo non sia più presente tra gli elementi del gruppo | NI | |
| TSF44 | RF17 | Se <i>NaturalAPI Design</i> ha prodotto una <i>BAL</i> , deve essere possibile associare tutte le <i>feature</i> che sono state usate come input | S | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|---|
| TSF45 | RF17.1 | Se l'utente aggiunge <i>feature</i> , possono diventare suggerimenti che poi vengono inseriti nel BAL | S | <ul style="list-style-type: none"> • Se l'utente non aggiunge nulla, il <i>BAL</i> non cambia; • Se l'utente aggiunge una <i>feature</i>, dopo calcolo e approvazione, il suggerimento deve essere presente nel <i>BAL</i>. |
| TSF46 | RF17.2 | Se ho già prodotto un suggerimento a partire da una <i>feature</i> , <i>NaturalAPI Design</i> non deve ricalcolarlo | S | <ul style="list-style-type: none"> • Se l'utente non aggiunge <i>feature</i>, <i>NaturalAPI Design</i> non deve produrre suggerimenti; • Se l'utente aggiunge <i>feature</i>, <i>NaturalAPI Design</i> produce suggerimenti solo per la nuova <i>feature</i>. |
| TSF47 | RF18 | Se l'utente non fornisce il <i>BAL</i> o <i>PLA</i> , <i>NaturalAPI Develop</i> avvisa che mancano gli input necessari | S | |
| TSF48 | RF19 | Se l'utente crea un <i>PLA</i> , diventa un file del sistema operativo | S | |
| TSF49 | RF20 | Se l'utente aggiunge nuove regole, il <i>PLA</i> viene modificato | S | <ul style="list-style-type: none"> • Se l'utente non aggiunge regole, il <i>PLA</i> non cambia; • Se l'utente aggiunge nuove regole, il <i>PLA</i> presenta la nuove regole; |
| TSF50 | RF21 | Se l'utente seleziona un framework, i test vengono prodotti usando quel framework | S | |

| Test | Requisito | Obiettivo | Stato | Esempi |
|-------|-----------|--|-------|---|
| TSF51 | RF22 | Se l'utente fornisce gli input, <i>NaturalAPI Developer</i> produce un output con il formato previsto; | S | |
| TSF52 | RF23 | Se l'utente accetta un suggerimento, le <i>API</i> riportano il suggerimento | S | |
| TSF53 | RF24 | Se l'utente modifica un suggerimento, le <i>API</i> riportano il suggerimento modificato | S | |
| TSF54 | RF25 | <i>NaturalAPI Developer</i> è in grado di associare il <i>BAL</i> alle <i>API</i> prodotte | S | <ul style="list-style-type: none"> • Se le <i>API</i> sono state prodotto da un certo <i>BAL</i>, tentare di associarle con un <i>BAL</i> estraneo causa un errore; • Se le <i>API</i> sono state prodotte da un certo <i>BAL</i>, il test di associazione ha esito positivo. |

4.3 Test di Integrazione

I test di integrazione hanno l'obiettivo di testare moduli composti di diverse unità. In particolare, i test di integrazione sono stati divisi in due categorie a seconda dell'estensione del test, ovvero di quante componenti il test si compone.

- Test ad ampio *scope*: testano funzionalità coinvolgendo diverse unità;
- Test a *scope* ridotto: testano specifici moduli dell'architettura.

I test seguono la codifica definita nel documento NP.1.0.0.

4.3.1 Specifiche dei test ad ampio scope

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|--|------------------------|-------|
| TI1.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe CreateBDL | Discover | S |
| TI1.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe CreateBDL | Discover | S |
| TI1.3 | Verifichiamo i collegamenti tra la classe AnalyzeDocument e la classe CreateBDL | Discover | S |
| TI3.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe RemoveDocuments | Discover | S |
| TI3.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe RemoveDocuments | Discover | S |
| TI3.3 | Verifichiamo i collegamenti tra la classe AnalyzeDocument e la classe RemoveDocuments | Discover | S |
| TI2.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe AddDocuments | Discover | S |
| TI2.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe AddDocuments | Discover | S |
| TI2.3 | Verifichiamo i collegamenti tra la classe AnalyzeDocument e la classe AddDocuments | Discover | S |
| TI18.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe GenerateBAL | Designer | S |
| TI18.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe GenerateBAL | Designer | S |
| TI18.3 | Verifichiamo i collegamenti tra l'interfaccia BALAnalyzer e la classe GenerateBAL | Designer | S |

| Test | Obiettivo | Tool di appartenenza | Stato |
|--------|---|----------------------|-------|
| TI6.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe GenerateBALSuggestions | Designer | S |
| TI6.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe GenerateBALSuggestions | Designer | S |
| TI6.3 | Verifichiamo i collegamenti tra l'interfaccia BALAnalyzer e la classe GenerateBALSuggestions | Designer | S |
| TI8.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe DeclineBALSuggestion | Designer | S |
| TI8.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe DeclineBALSuggestion | Designer | S |
| TI9 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe ModifyBALSuggestion | Designer | S |
| TI25.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe GenerateAPI | Develop | S |
| TI25.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe GenerateAPI | Develop | S |
| TI20.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe SuggestApi | Develop | S |
| TI20.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe SuggestApi | Develop | S |
| TI20.3 | Verifichiamo i collegamenti tra l'interfaccia BALAnalyzer e la classe SuggestApi | Develop | S |
| TI24.1 | Verifichiamo i collegamenti tra gli Interface Adapters e la classe ModifyApi | Develop | S |
| TI24.2 | Verifichiamo i collegamenti tra l'interfaccia RepositoryAccess e la classe ModifyApi | Develop | S |
| TI24.3 | Verifichiamo i collegamenti tra l'interfaccia BALAnalyzer e la classe ModifyApi | Develop | S |

Tabella 6: Tabella test di integrazione

4.3.2 Specifiche dei test a scope ridotto

| Test | Obiettivo | Classe di appartenenza | Stato |
|------|--|------------------------|-------|
| TI1 | Testiamo il modulo che implementa l'interfaccia RepositoryAccess | Discover | S |
| TI2 | Testiamo il modulo di parsing | Discover | S |
| TI3 | Testiamo il modulo che implementa l'interfaccia RepositoryAccess | Designer | S |
| TI4 | Testiamo il modulo di parsing | Designer | S |
| TI5 | Testiamo il modulo che implementa l'interfaccia BALAnalyzer | Designer | S |
| TI6 | Testiamo il modulo che implementa l'interfaccia RepositoryAccess | Developer | S |
| TI7 | Testiamo il modulo che implementa l'interfaccia BALAnalyzer | Developer | S |

Tabella 7: Tabella test di integrazione

Tabella 8: Tabella tracciamento ID-Interfaccia

| ID | Classe |
|----|------------------------------|
| 1 | RepositoryAccess (Discover) |
| 2 | TextAnalyzer (Discover) |
| 3 | RepositoryAccess (Designer) |
| 4 | TextAnalyzer (Designer) |
| 5 | BALAnalyzer (Designer) |
| 6 | RepositoryAccess (Developer) |
| 7 | BALAnalyzer (Developer) |

4.4 Test di Unità

I test di unità hanno lo scopo di testare le unità minime in cui il software può essere decomposto in modo che siano verificabili singolarmente. In generale un'unità si identifica con un metodo di una classe. I test seguono la codifica definita nel documento NP.1.0.0.

4.4.1 Specifiche

Tabella 9: Tabella dei test di unità

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|------------------------|-------|
| TU1.1 | Viene verificato che la creazione di un nuovo Bdl avvenga correttamente | Bdl | S |
| TU1.2 | Viene verificato che l'aggiunta di un nome a un Bdl avvenga correttamente: se il nome è già presente nel Bdl si incrementa la frequenza, altrimenti lo si inserisce | Bdl | S |
| TU1.3 | Viene verificato che l'aggiunta di un verbo a un Bdl avvenga correttamente: se il verbo è già presente nel Bdl si incrementa la frequenza, altrimenti lo si inserisce | Bdl | S |
| TU1.4 | Viene verificato che l'aggiunta di un predicato a un Bdl avvenga correttamente: se il predicato è già presente nel Bdl si incrementa la frequenza, altrimenti lo si inserisce | Bdl | S |
| TU1.5 | Viene verificato che l'ordinamento del Bdl avvenga correttamente, ordinando i termini per frequenza decrescente | Bdl | S |
| TU1.6 | Viene verificato che l'aggiunta di un nome con la relativa frequenza a un Bdl avvenga correttamente | Bdl | S |
| TU1.7 | Viene verificato che l'aggiunta di un verbo con la relativa frequenza a un Bdl avvenga correttamente | Bdl | S |
| TU1.8 | Viene verificato che l'aggiunta di un predicato con la relativa frequenza a un Bdl avvenga correttamente | Bdl | S |
| TU1.9 | Viene verificato che la rimozione di un nome da un Bdl avvenga correttamente: la sua frequenza viene diminuita di uno, se è uguale a uno viene rimosso | Bdl | S |
| TU1.10 | Viene verificato che la rimozione di un verbo da un Bdl avvenga correttamente: la sua frequenza viene diminuita di uno, se è uguale a uno viene rimosso | Bdl | S |
| TU1.11 | Viene verificato che la rimozione di un predicato da un Bdl avvenga correttamente: la sua frequenza viene diminuita di uno, se è uguale a uno viene rimosso | Bdl | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|------------------------|-------|
| TU1.12 | Viene verificato che la conversione di un Bdl a stringa di testo avvenga correttamente | Bdl | S |
| TU1.13 | Viene verificato che la lista dei nomi venga di un Bdl ritornata correttamente e non venga restituito un riferimento nullo | Bdl | S |
| TU1.14 | Viene verificato che la lista dei verbi venga di un Bdl ritornata correttamente e non venga restituito un riferimento nullo | Bdl | S |
| TU1.15 | Viene verificato che la lista dei predicati venga di un Bdl ritornata correttamente e non venga restituito un riferimento nullo | Bdl | S |
| TU2.1 | Viene verificato che la creazione di un Document avvenga correttamente | Document | S |
| TU2.2 | Viene verificato che che un Document venga convertito correttamente in formato stringa | Document | S |
| TU3.1 | Viene verificato che la creazione di un nuovo WordCounter tramite parola avvenga correttamente | WordCounter | S |
| TU3.2 | Viene verificato che la creazione di un nuovo WordCounter tramite parola e frequenza avvenga correttamente | WordCounter | S |
| TU3.3 | Viene verificato che l'incremento della frequenza avvenga correttamente | WordCounter | S |
| TU3.4 | Viene verificato che il decremento della frequenza avvenga correttamente | WordCounter | S |
| TU3.5 | Viene verificato che la conversione a stringa avvenga correttamente | WordCounter | S |
| TU4.1 | Viene verificato che la creazione di un Wordtag avvenga correttamente | WordTag | S |
| TU4.2 | Viene verificato che la conversione a stringa avvenga correttamente | WordTag | S |
| TU5.1 | Viene verificato che la creazione di un Dependency avvenga correttamente | Dependency | S |
| TU5.2 | Viene verificato che la restituzione della lista di WordTag avvenga correttamente e non venga restituito un riferimento nullo | Dependency | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|------------------------|-------|
| TU5.3 | Viene verificato che l'aggiunta di un nuovo elemento avvenga correttamente | LemmatizerData | S |
| TU5.4 | Viene verificato che la conversione a stringa avvenga correttamente | LemmatizerData | S |
| TU6.1 | Viene verificato che la creazione di un AnalyzedData avvenga correttamente | AnalyzedData | S |
| TU6.2 | Viene verificato che l'aggiunta del parsing di un elemento avvenga correttamente | AnalyzedData | S |
| TU6.3 | Viene verificato che l'aggiunta della lemmatizzazione di un elemento avvenga correttamente | AnalyzedData | S |
| TU6.4 | Viene verificato che la restituzione della lista dei dati di parsing avvenga correttamente e non venga restituito un riferimento nullo | AnalyzerData | S |
| TU6.5 | Viene verificato che la restituzione della lista degli elementi lemmizzati avvenga correttamente e non venga restituito un riferimento nullo | AnalyzerData | S |
| TU6.6 | Viene verificato che la conversione a stringa avvenga correttamente | AnalyzedData | S |
| TU7.1 | Viene verificato che la creazione dell'interactor CreateBdl avvenga correttamente | CreateBdl | S |
| TU7.2 | Viene verificato che la richiesta di creazione di un BDL fornendo un nome e una lista di titoli di documenti provochi la creazione di un file e la stampa dell'output | CreateBdl | S |
| TU8.1 | Viene verificato che la creazione dell'interactor AddDocuments avvenga correttamente | AddDocuments | S |
| TU8.2 | Viene verificato che la richiesta di aggiunta di una lista di documenti ad un dato BDL, porti alla corretta modifica del file BDL e alla stampa dell'output | AddDocuments | S |
| TU9.1 | Viene verificato che la creazione dell'interactor RemoveDocuments avvenga correttamente | RemoveDocuments | S |
| TU9.2 | Viene verificato che la richiesta di rimozione di una serie di documenti da un BDL, porti alla corretta modifica del file BDL e alla stampa dell'output | RemoveDocuments | S |
| TU10.1 | Viene verificato che la creazione di un AnalyzeDocument avvenga correttamente | AnalyzeDocument | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|---------|--|-----------------------------|-------|
| TU10.2 | Viene verificato che la rimozione di un documento da un oggetto BDL avvenga correttamente | AnalyzeDocument | S |
| TU10.3 | Viene verificato che l'aggiunta di un documento a un oggetto BDL avvenga correttamente | AnalyzeDocument | S |
| TU11.1 | Viene verificato che la creazione di un FileSystemAccess avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.2 | Viene verificato che il caricamento da file di un documento avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.3 | Viene verificato che il caricamento da file di un BDL avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.4 | Viene verificato che il salvataggio su file di un BDL avvenga correttamente e che vengano creati i tre file csv | FileSystemAccess (Discover) | S |
| TU11.5 | Viene verificato che il caricamento da file di un associazione avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.6 | Viene verificato che il salvataggio su file di un associazione avvenga correttamente e che venga creato il file JSON | FileSystemAccess (Discover) | S |
| TU11.7 | Viene verificato che la lettura da file dei nomi di un BDL avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.8 | Viene verificato che la lettura da file dei verbi di un BDL avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.9 | Viene verificato che la lettura da file dei predicati di un BDL avvenga correttamente | FileSystemAccess (Discover) | S |
| TU11.10 | Viene verificato che la scrittura su file di una lista di WordCounter avvenga correttamente | FileSystemAccess (Discover) | S |
| TU12.1 | Viene verificato che la creazione di un oggetto StanfordNlp avvenga correttamente | StanfordNlp | S |
| TU12.2 | Viene verificato che il parsing di un documento restituisca i dati attesi | StanfordNlp | S |
| TU13.1 | Viene verificato che la creazione di un Repository avvenga correttamente | Repository (Discover) | S |
| TU13.2 | Viene verificato che la creazione di un nuovo BDL venga memorizzata correttamente | Repository (Discover) | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|--|------------------------|-------|
| TU13.3 | Viene verificato che la lettura di un BDL da memoria persistente avvenga correttamente | Repository (Discover) | S |
| TU13.4 | Viene verificato che l'update di un BDL già esistente avvenga correttamente | Repository (Discover) | S |
| TU13.5 | Viene verificato che la lettura di un Document da memoria persistente avvenga correttamente | Repository (Discover) | S |
| TU13.6 | Viene verificato che la lettura di un associazione da memoria persistente avvenga correttamente | Repository (Discover) | S |
| TU13.7 | Viene verificato che l'aggiornamento di un associazione avvenga correttamente | Repository (Discover) | S |
| TU14.1 | Viene verificato che la creazione di una CLI avvenga correttamente | CLI (Discover) | S |
| TU14.2 | Viene verificato che la stampa delle operazioni possibili avvenga correttamente | CLI (Discover) | S |
| TU14.3 | Viene verificato che la richiesta di un'operazione scateni il giusto use case | CLI (Discover) | S |
| TU14.4 | Viene verificato che l'inserimento del nome di un BDL da parte dell'utente avvenga correttamente e non sia vuoto | CLI (Discover) | S |
| TU14.5 | Viene verificato che l'inserimento di un nome di un documento avvenga correttamente e non sia vuoto | CLI (Discover) | S |
| TU14.6 | Viene verificato che la stampa del risultato avvenga correttamente | CLI (Discover) | S |
| TU15.1 | Viene verificato che la creazione di un Presenter avvenga correttamente | Presenter | S |
| TU15.2 | Viene verificato che l'aggiornamento dello stato interno venga effettuato in modo corretto | Presenter | S |
| TU16.1 | Viene verificato che la creazione di un Controller avvenga correttamente | Controller (Discover) | S |
| TU16.2 | Viene verificato che la richiesta di creazione di un BDL comporti la chiamata al giusto interactor | Controller (Discover) | S |
| TU16.3 | Viene verificato che la richiesta di aggiunta documenti a un BDL comporti la chiamata al giusto interactor | Controller (Discover) | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|--|------------------------|-------|
| TU16.4 | Viene verificato che la richiesta di rimozione documenti da un BDL comporti la chiamata al giusto interactor | Controller (Discover) | S |
| TU17.1 | Viene verificato che la creazione di una nuova Dependency avvenga correttamente | Dependency | S |
| TU18.1 | Viene verificato che la creazione di una nuova Action tramite nome avvenga correttamente | Action | S |
| TU18.2 | Viene verificato che la creazione di una nuova Action tramite nome e tipo avvenga correttamente | Action | S |
| TU18.3 | Viene verificato che l'aggiunta di ObjectParam avvenga correttamente e che non introduca duplicati | Action | S |
| TU18.4 | Viene verificato che la modifica del nome di un ObjectParam avvenga correttamente | Action | S |
| TU18.5 | Viene verificato che la modifica del tipo di un ObjectParam avvenga correttamente | Action | S |
| TU18.6 | Viene verificato che la modifica del campo required di un ObjectParam avvenga correttamente | Action | S |
| TU18.7 | Viene verificato che la conversione a stringa avvenga correttamente | Action | S |
| TU19.1 | Viene verificato che la creazione di un Actor vuoto avvenga correttamente | Actor | S |
| TU19.2 | Viene verificato che la creazione di un Actor tramite nome avvenga correttamente | Actor | S |
| TU19.3 | Viene verificato che la creazione di un Actor tramite nome e lista di Action avvenga correttamente | Actor | S |
| TU19.4 | Viene verificato che l'aggiunta di una Action avvenga correttamente, senza creare duplicati | Actor | S |
| TU19.5 | Viene verificato che l'aggiunta di una lista di Action avvenga correttamente, senza creare duplicati | Actor | S |
| TU19.6 | Viene verificato che alla richiesta di una Action tramite nome venga restituita quella corretta, oppure venga segnalato che non è presente | Actor | S |
| TU19.7 | Viene verificato che la conversione a stringa avvenga correttamente | Actor | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|------------------------|-------|
| TU20.1 | Viene verificato che la creazione di un BAL vuoto avvenga correttamente | BAL | S |
| TU20.2 | Viene verificato che la creazione di un BAL tramite attore avvenga correttamente | BAL | S |
| TU20.3 | Viene verificato che la creazione di un BAL tramite lista di attori avvenga correttamente, senza creare attori duplicati | BAL | S |
| TU20.4 | Viene verificato che l'aggiunta di un attore a un BAL non crei duplicati | BAL | S |
| TU20.5 | Viene verificato che la conversione a stringa avvenga correttamente | BAL | S |
| TU21.1 | Viene verificato che la creazione di un ObjectParam vuoto avvenga correttamente | ObjectParam | S |
| TU21.2 | Viene verificato che la creazione di un ObjectParam tramite nome avvenga correttamente | ObjectParam | S |
| TU21.3 | Viene verificato che la creazione di un ObjectParam tramite nome e tipo avvenga correttamente | ObjectParam | S |
| TU22.1 | Viene verificato che la creazione di un nuovo Scenario avvenga correttamente | Scenario | S |
| TU22.2 | Viene verificato che la restituzione delle Action collegate allo scenario avvenga correttamente | Scenario | S |
| TU23.1 | Viene verificato che la creazione di una BlackList vuota avvenga correttamente | BlackList | S |
| TU23.2 | Viene verificato che la creazione di una BlackList tramite una lista di termini avvenga correttamente, senza inserire termini duplicati | BlackList | S |
| TU23.3 | Viene verificato che l'aggiunta di un termine avvenga solo se il termine non è presente | BlackList | S |
| TU23.4 | Viene verificato che l'aggiunta di una lista di termini aggiunga effettivamente solo i termini che non sono presenti | BlackList | S |
| TU23.5 | Viene verificato che se si cerca un termine nella BlackList, venga ritornato true solo se è presente | BlackList | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|---------------------------|-------|
| TU24.1 | Viene verificato che il caricamento da file di un BDL avvenga correttamente | FileSystemAccess (Design) | S |
| TU24.2 | Viene verificato che la lettura da file dei nomi di un BDL avvenga correttamente | FileSystemAccess (Design) | S |
| TU24.3 | Viene verificato che la lettura da file dei verbi di un BDL avvenga correttamente | FileSystemAccess (Design) | S |
| TU24.4 | Viene verificato che la lettura da file dei predicati di un BDL avvenga correttamente | FileSystemAccess (Design) | S |
| TU25.1 | Viene verificato che la creazione di un Repository avvenga correttamente | Repository (Design) | S |
| TU25.2 | Viene verificato che la creazione di un BAL venga memorizzata correttamente in memoria persistente | Repository (Design) | S |
| TU25.3 | Viene verificato che la lettura di un BAL da memoria persistente avvenga correttamente | Repository (Design) | S |
| TU25.4 | Viene verificato che la creazione di uno scenario venga memorizzata correttamente in memoria locale | Repository (Design) | S |
| TU25.5 | Viene verificato che la lettura di più scenari da memoria locale avvenga correttamente | Repository (Design) | S |
| TU25.6 | Viene verificato che la cancellazione di un suggerimento dalla memoria locale avvenga correttamente | Repository (Design) | S |
| TU25.7 | Viene verificato che la cancellazione degli scenari da memoria locale avvenga correttamente | Repository (Design) | S |
| TU26.1 | Viene verificato che la creazione di un BalAnalyzerImplementation avvenga correttamente | BalAnalyzerImplementation | S |
| TU26.2 | Viene verificato che la restituzione del BAL avvenga correttamente | BalAnalyzerImplementation | S |
| TU26.3 | Viene verificato che la creazione del JSON avvenga correttamente, restituendo il testo nel giusto formato | BalAnalyzerImplementation | S |
| TU27.1 | Viene verificato che la creazione di un Controller avvenga correttamente | Controller (Design) | S |
| TU27.2 | Viene verificato che la richiesta di generazione suggerimenti BAL comporti la chiamata al giusto interactor | Controller (Design) | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|--|------------------------|-------|
| TU27.3 | Viene verificato che la richiesta di rifiuto suggerimento BAL comporti la chiamata al giusto interactor | Controller (Design) | S |
| TU27.4 | Viene verificato che la richiesta di creazione di un BAL comporti la chiamata al giusto interactor | Controller (Design) | S |
| TU28.1 | Viene verificato che l'aggiornamento dello stato interno venga effettuato in modo corretto | DataPresenter | S |
| TU29.1 | Viene verificato che la creazione di una CLI avvenga correttamente | CLI (Design) | S |
| TU29.2 | Viene verificato che la stampa delle operazioni possibili avvenga correttamente | CLI (Design) | S |
| TU29.3 | Viene verificato che la richiesta di un'operazione scateni il giusto use case | CLI (Design) | S |
| TU29.4 | Viene verificato che la generazione dei suggerimenti avvenga correttamente | CLI (Design) | S |
| TU29.5 | Viene verificato che l'inserimento di un nome di un documento avvenga correttamente e non sia vuoto | CLI (Design) | S |
| TU29.6 | Viene verificato che la stampa del risultato avvenga correttamente | CLI (Design) | S |
| TU30.1 | Viene verificato che la creazione dell'interactor GenerateBal avvenga correttamente | GenerateBal | S |
| TU30.2 | Viene verificato che la richiesta di generazione di un BAL porti alla creazione di un file e alla stampa dell'output | GenerateBal | S |
| TU31.1 | Viene verificato che la creazione dell'interactor GenerateBalSuggestions avvenga correttamente | GenerateBalSuggestions | S |
| TU31.2 | Viene testato il comportamento dello UC quando vengono inseriti input errati | | S |
| TU31.3 | Viene verificato che il comportamento dello UC quando si inseriscono feature corrette è quello atteso | | S |
| TU32.1 | Viene verificato che la creazione dell'interactor DeclineBalSuggestion avvenga correttamente | DeclineBalSuggestion | S |
| TU32.2 | Viene testato il comportamento dello UC quando vengono inseriti input errati | DeclineBalSuggestion | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|--|-------------------------|-------|
| TU33.1 | Viene verificato che l'aggiunta di uno scenario avvenga correttamente, senza creare duplicati | DataKeeper (Design) | S |
| TU33.2 | Viene verificato che la rimozione di un action avvenga correttamente, rimuovendo l'action se presente o non facendo nulla altrimenti | DataKeeper (Design) | S |
| TU34.1 | Viene verificato che la creazione di un oggetto API avvenga correttamente | API | S |
| TU34.2 | Viene verificato che l'aggiunta di un API avvenga correttamente, senza creare duplicati | API | S |
| TU35.1 | Viene verificato che la creazione di un PLA avvenga correttamente | PLA | S |
| TU36.1 | Viene verificato che l'aggiunta di API avvenga correttamente, senza creare duplicati | DataKeeper (Develop) | S |
| TU36.2 | Viene verificato che la rimozione di una API avvenga correttamente, rimuovendola se l'API è presente e non facendo nulla altrimenti | DataKeeper (Develop) | S |
| TU37.1 | Viene verificato che il caricamento di un PLA da file avvenga correttamente | FileSystem (Develop) | S |
| TU37.2 | Viene verificato che il salvataggio delle API su file avvenga correttamente | FileSystem (Develop) | S |
| TU38.1 | Viene verificato che la lettura di un PLA da memoria persistente avvenga correttamente | Repository (Develop) | S |
| TU38.2 | Viene verificato che la creazione di API su memoria persistente avvenga correttamente | Repository (Develop) | S |
| TU38.3 | Viene verificato che l'aggiunta di API su memoria locale avvenga correttamente | Repository (Develop) | S |
| TU38.4 | Viene verificato che la cancellazione di una API da memoria locale avvenga correttamente | Repository (Develop) | S |
| TU39.1 | Viene verificato che la creazione di un Controller avvenga correttamente | Controller (Develop) | S |
| TU39.2 | Viene verificato che la richiesta di creazione suggerimenti API comporti la chiamata al giusto interactor | Controller (Develop) | S |
| TU40.1 | Viene verificato che l'aggiornamento dello stato interno venga effettuato in modo corretto | DataPresenter (Develop) | S |

| Test | Obiettivo | Classe di appartenenza | Stato |
|--------|---|------------------------|-------|
| TU41.1 | Viene verificato che la creazione di una CLI avvenga correttamente | CLI (Develop) | S |
| TU41.2 | Viene verificato che la stampa di richiesta BAL avvenga correttamente | CLI (Develop) | S |
| TU41.3 | Viene verificato che la stampa di richiesta PLA avvenga correttamente | CLI (Develop) | S |
| TU41.4 | Viene verificato che lettura da input utente del nome del BAL avvenga correttamente | CLI (Develop) | S |
| TU41.5 | Viene verificato che lettura da input utente del nome del PLA avvenga correttamente | CLI (Develop) | S |
| TU41.6 | Viene verificato che la stampa dei risultati avvenga correttamente | CLI (Develop) | S |
| TU42.1 | Viene testato il comportamento dello UC quando vengono inseriti input errati | GenerateApi | S |
| TU43.1 | Viene testato il comportamento dello UC quando vengono inseriti input errati | SuggestApi | S |
| TU44.1 | Viene testato il comportamento dello UC quando vengono inseriti input errati | ModifyApi | S |

Tabella 10: Tabella tracciamento ID-Classe

| ID | Classe |
|----|--------------|
| 1 | BDL |
| 2 | Document |
| 3 | WordCounter |
| 4 | WordTag |
| 5 | Dependency |
| 6 | AnalyzedData |
| 7 | CreateBdl |
| 8 | AddDocuments |

| ID | Classe |
|----|--------------------------------|
| 9 | RemoveDocuments |
| 10 | AnalyzeDocuments |
| 11 | FileSystemAccess (Discover) |
| 12 | StanfordNlp |
| 13 | Repository (Discover) |
| 14 | CLI (Discover) |
| 15 | Presenter |
| 16 | Controller (Discover) |
| 17 | ManualConfig |
| 18 | Action |
| 19 | Actor |
| 20 | BAL |
| 21 | ObjectParam |
| 22 | Scenario |
| 23 | BlackList |
| 24 | FileSystemAccess (Designer) |
| 25 | Repository (Designer) |
| 26 | BalAnalyzerImplementation |
| 27 | Controller (Designer) |
| 28 | DataPresenter |
| 29 | CLI (Designer) |
| 30 | GenerateBal |
| 31 | GenerateBalSuggestions |
| 32 | DeclineBalSuggestion |

| ID | Classe |
|----|---------------------------|
| 33 | DataKeeper (Designer) |
| 34 | API |
| 35 | PLA |
| 36 | DataKeeper (Developer) |
| 37 | FileSystem (Developer) |
| 38 | Repository (Developer) |
| 39 | Controller (Developer) |
| 40 | DataPresenter (Developer) |
| 41 | CLI (Developer) |
| 42 | GenerateApi |
| 43 | SuggestApi |
| 44 | ModifyApi |

Le specifiche dei test di unità vengono aggiornate non appena viene progettata una nuova componente. Non essendo ancora conclusa la progettazione di dettaglio, la tabella subirà ulteriori modifiche.

A Resoconto attività di verifica

In questa sezione vengono descritti e analizzati gli esiti delle attività di verifica, effettuate nei vari periodi di avanzamento del progetto, attraverso le metriche indicate in tale documento.

A.1 Revisione dei requisiti

A.1.1 Documenti

A.1.1.1 Indice di Gulpease

Per i verbali, sia interni che esterni, è stato deciso di tenere conto della media dei loro indici di Gulpease, in modo da rendere più agevole la lettura del diagramma.

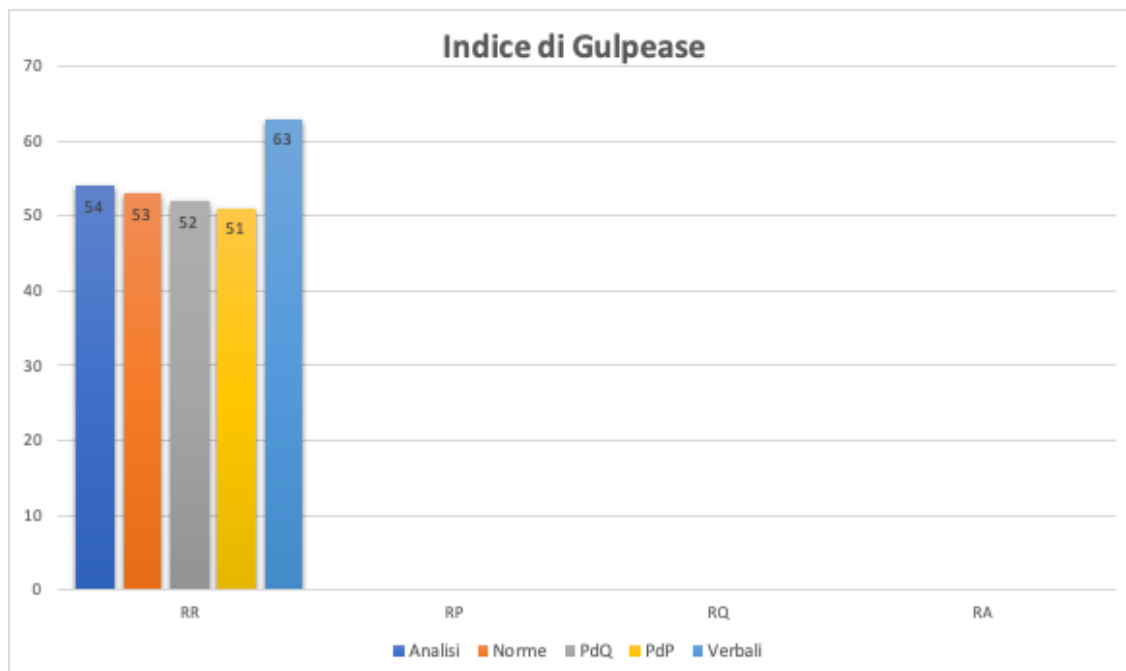


Figura 1: Indice di Gulpease nel periodo di RR

A.1.2 Processi

Essendo il periodo di revisione dei requisiti di puro investimento da parte del gruppo, non è stato necessari calcolare le metriche di processo.

A.1.3 Software

Nel periodo di revisione dei requisiti non c'è stata nessuna attività di codifica, perciò le metriche della qualità riguardanti il prodotto software verranno calcolate nel periodo di codifica di quest'ultimo.

A.2 Revisione di progettazione

A.2.1 Documenti

A.2.1.1 Indice di Gulpease

Per i verbali, sia interni che esterni, è stato deciso di tenere conto della media dei loro indici di Gulpease, in modo da rendere più agevole la lettura del diagramma.

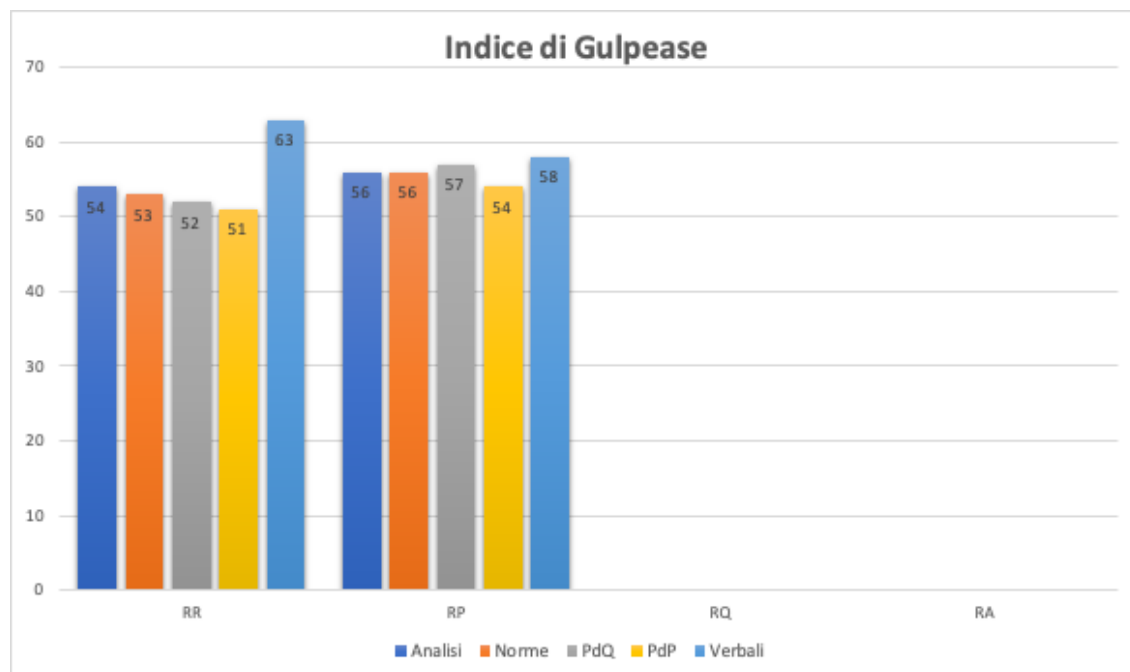


Figura 2: Indice di Gulpease nei periodi di RR e RP

A.2.2 Processi

A.2.2.1 EAC: Estimated at Completion

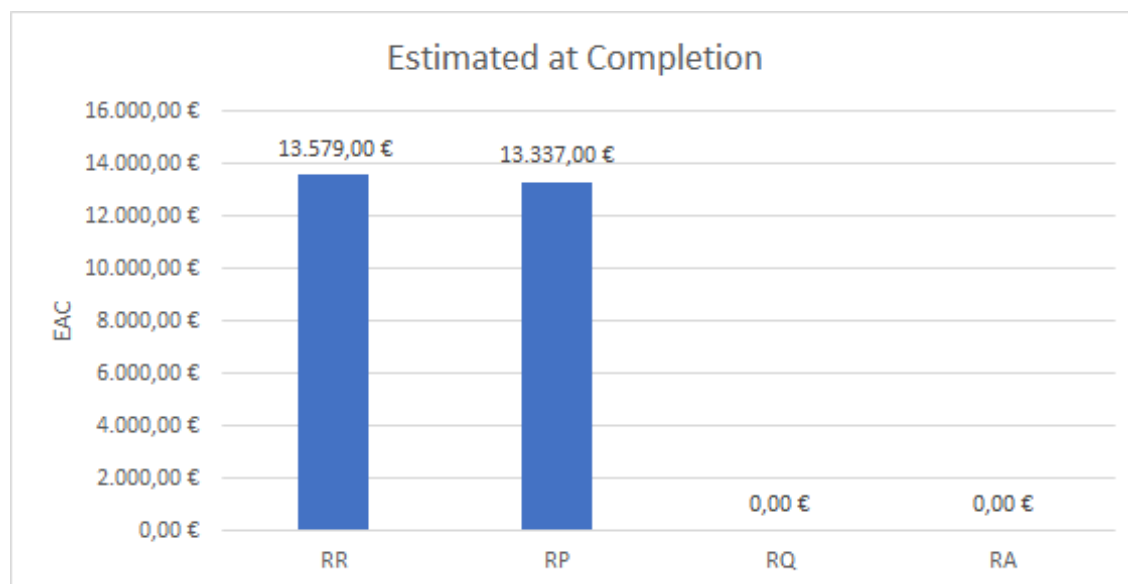


Figura 3: Estimated at Completion nei periodi di RR e RP

Nel primo periodo, il valore dell'Estimated at completion è uguale al valore del preventivo in quanto l'actual cost di questa fase è 0. Nel periodo di RR invece il valore è sceso perchè a consuntivo si è speso meno denaro di quanto pianificato, il valore dell'actual cost in questa fase è cambiato.

A.2.2.2 VAC: Variance at Completion

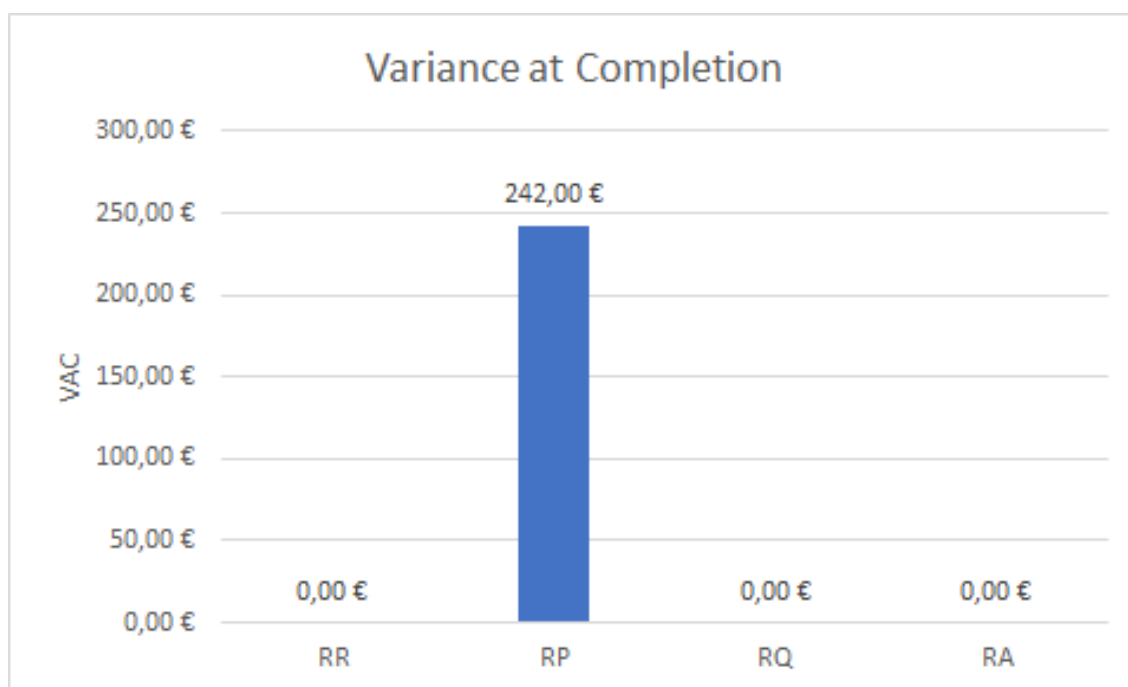


Figura 4: Variance at Completion nei periodi di RR e RP

Il progetto rientra nel budget nel periodo di RP.

A.2.2.3 BV: Budget Variance

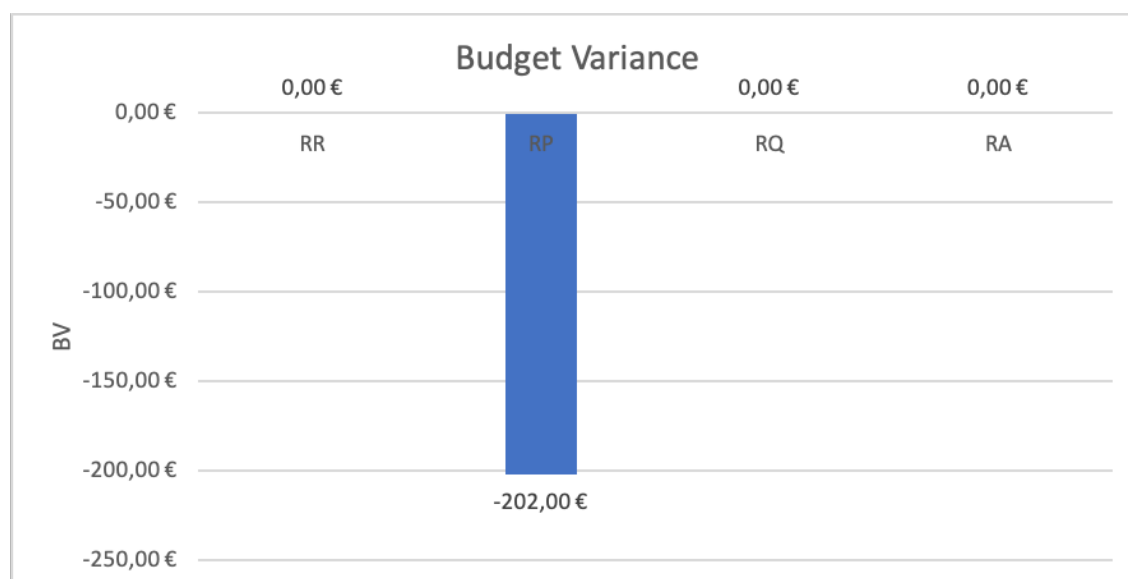


Figura 5: Budget Variance nei periodi di RR e RP

Dal diagramma in figura si nota che nel periodo di RP il valore del Budget Variance è di 202 €, il progetto sta quindi spendendo meno velocemente il budget di quanto è stato pianificato.

A.2.2.4 AC: Actual Cost

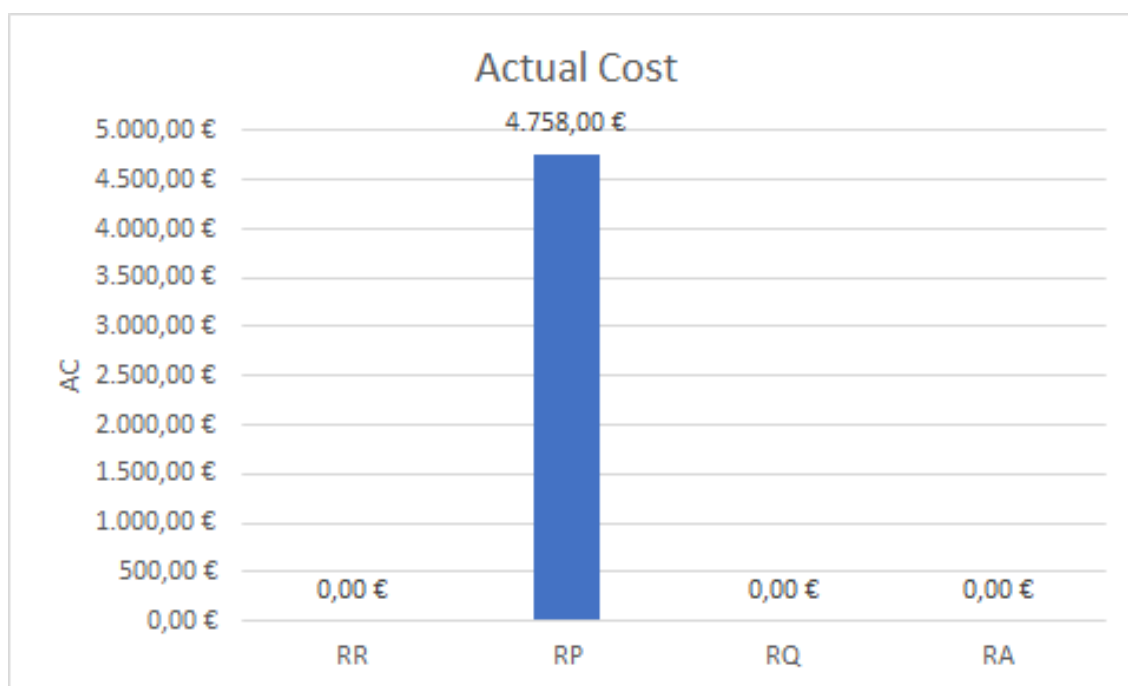


Figura 6: Actual Cost nei periodi di RR e RP

Come si può notare dalla figura, il valore dell'Actual Cost nel periodo di RR è pari a zero, in quanto il periodo di tempo in questa fase è stato da considerare come investimento personale.

A.2.3 Software

In questo periodo non è stato codificato nessun prodotto all'infuori del Proof of Concept il quale per natura è molto mutevole.

È stato quindi deciso di non tenere traccia delle metriche di prodotto software durante il suo sviluppo.

Infatti nella fase di progettazione di dettaglio e codifica il prodotto cambierà radicalmente, e non ci sarà una vera e propria continuità rispetto al PoC.

A.3 Revisione di qualifica

A.3.1 Documenti

A.3.1.1 Indice di Gulpease

Per i verbali, sia interni che esterni, è stato deciso di tenere conto della media dei loro indici di Gulpease, in modo da rendere più agevole la lettura del diagramma.

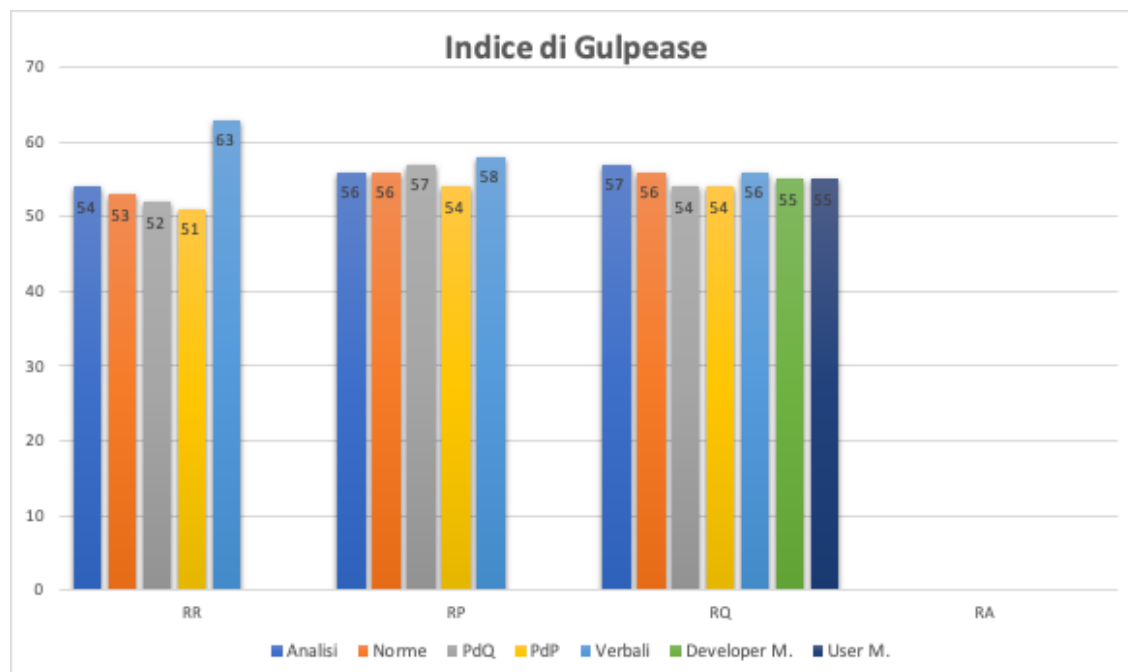


Figura 7: Indice di Gulpease nei periodi di RR, RP e RQ

A.3.2 Processi

A.3.2.1 EAC: Estimated at Completion

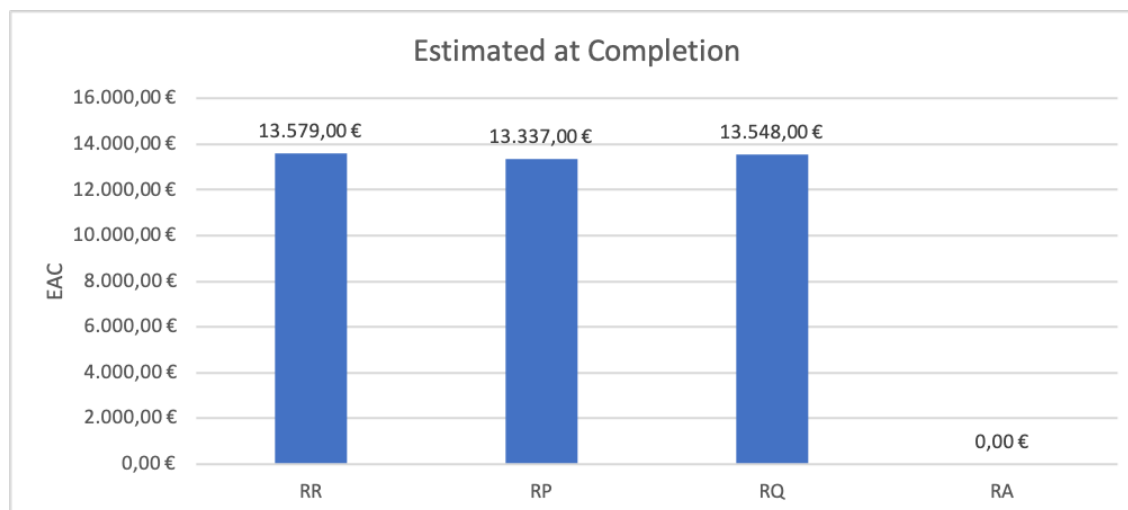


Figura 8: Estimated at Completion nei periodi di RP e RQ

Come si può vedere dal diagramma, il costo del progetto non supera il preventivo iniziale.

A.3.2.2 VAC: Variance at Completion

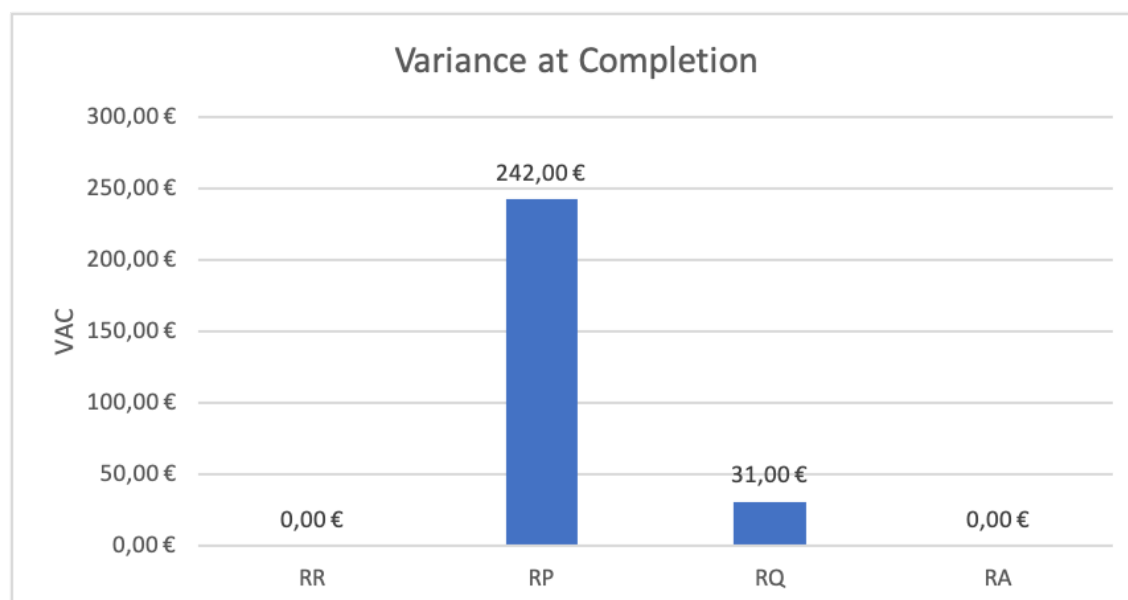


Figura 9: Variance at Completion nei periodi di RP e RQ

Il progetto ha sfiorato il budget nel periodo di RQ. Tuttavia, poichè nel periodo di RP sono stati risparmiati 202 €, il progetto rientra ancora nel budget di 31 €.

A.3.2.3 BV: Budget Variance

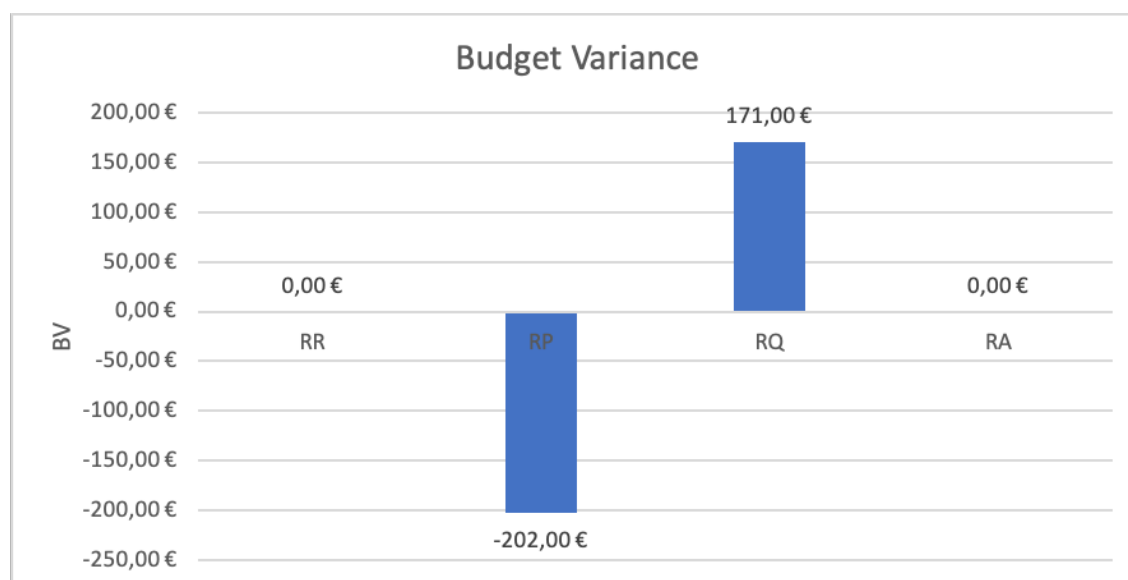


Figura 10: Budget Variance nei periodi di RP e RQ

Dal diagramma in figura si nota che nel periodo di RQ il valore del Budget Variance è di 171 €. Il progetto sta quindi spendendo più velocemente il budget di quanto è stato pianificato.

A.3.2.4 AC: Actual Cost

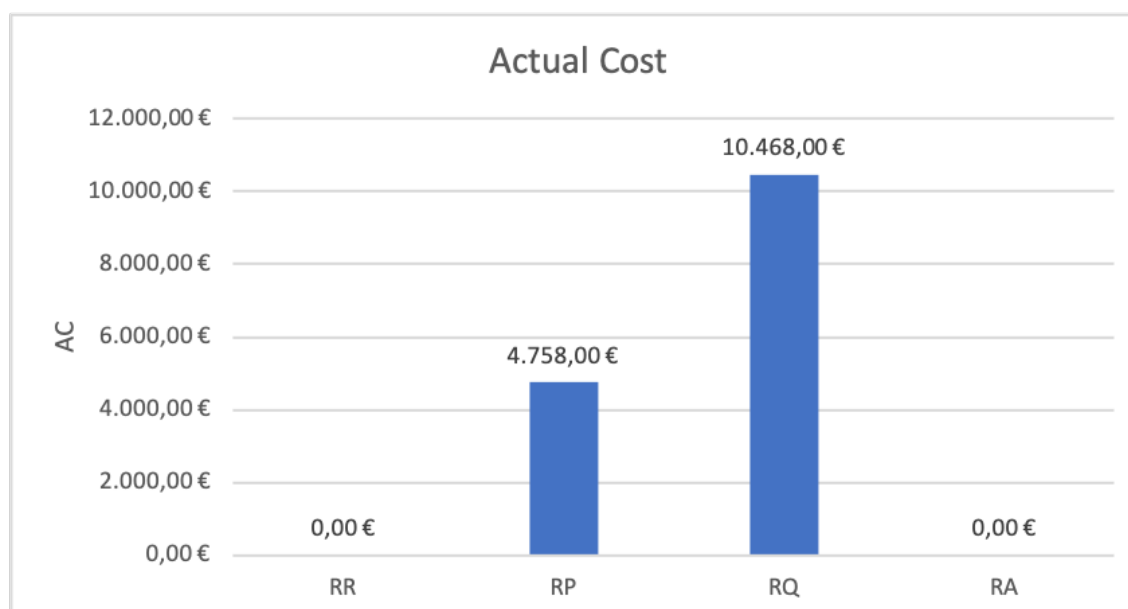


Figura 11: Actual Cost nei periodi di RP e RQ

L'Actual Cost del diagramma è la somma dell'Actual Cost del periodo RP e dell'Actual Cost del periodo RQ.

A.3.3 Software

In questa fase è iniziata la codifica del prodotto. Il gruppo FourCats si è affidato a due strumenti per l'analisi statica del codice, SonarQube e CodeMR.

A.3.3.1 Legenda Le metriche successive sono state calcolate grazie al plugin CodeMR che fornisce dei diagrammi per tenere sotto controllo le metriche.

Qui in figura è mostrata la legenda che indica i vari livelli dei valori.

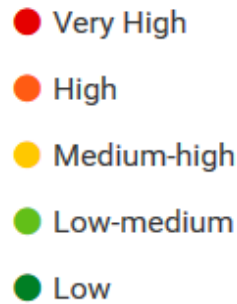


Figura 12: Legenda

A.3.3.2 WMC: Metodi pesati per classe La maggior parte dei metodi delle classi ha un valore molto basso, in particolare, delle 29 classi totali, solo 2 classi hanno un valore medio-basso. Questo significa che ogni classe è ben dimensionata e riutilizzabile.

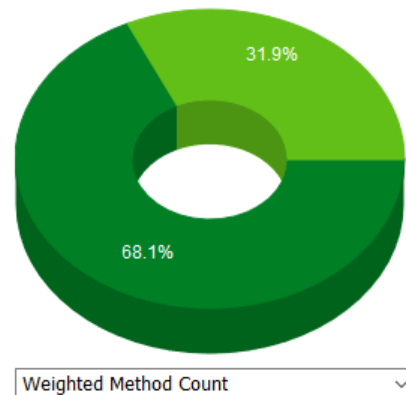


Figura 13: Legenda

A.3.3.3 DIT: Profondità dell'albero di ereditarietà Quasi tutte le classi hanno valore molto basso di profondità dell'albero di ereditarietà. Le classi così risultano meno complesse da sviluppare, testare e mantenere. Diagramma visualizzabile in figura 13.

A.3.3.4 NOC: Numero di figli Tutte le classi hanno un numero di figli basso.

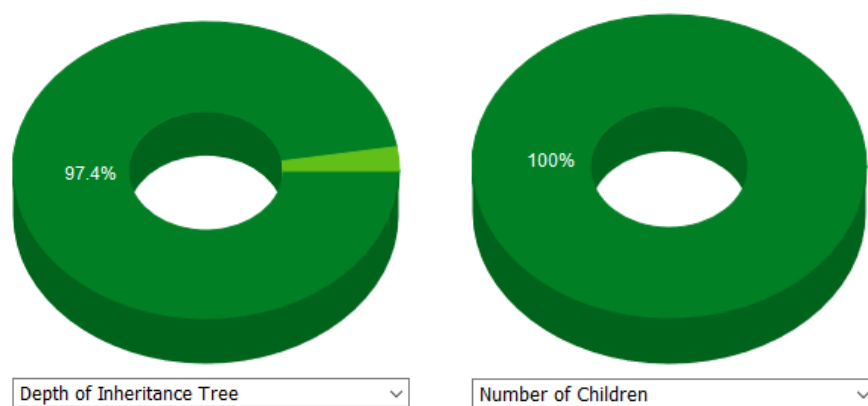


Figura 14: Metrica DIT sulla sinistra, NOC sulla destra

A.3.3.5 RFC: Risposta per classe Il numero di metodi che possono essere invocati in una classe è basso in tutte le classi.

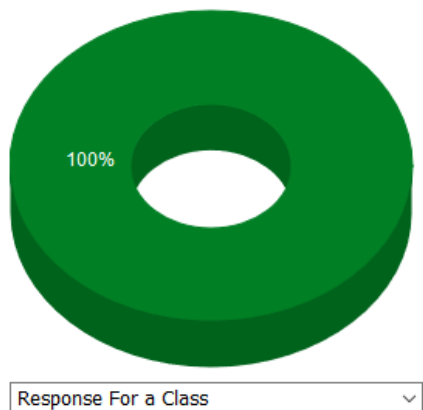


Figura 15: Risposta per classe

A.3.3.6 LCOM: Mancanza di coesione nei metodi Come si vede in figura, c'è una bassa mancanza di coesione nei metodi nelle varie classi. Tuttavia è presente comunque un elevato valore medio-basso.

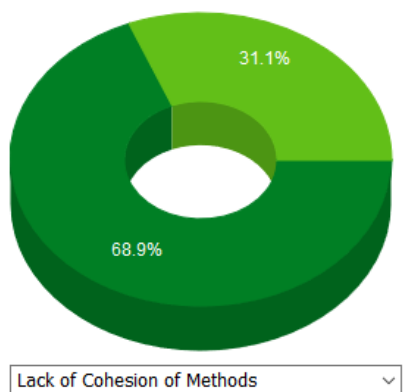


Figura 16: Mancanza di coesione nei metodi

A.3.3.7 CBO: Accoppiamento tra classi Tra le classi del progetto, esiste un basso accoppiamento tra classi. In pochi casi il valore raggiunge il livello medio-basso.

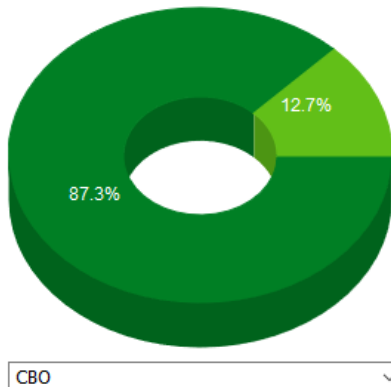


Figura 17: Accoppiamento tra classi

A.3.3.8 SonarQube Avendo avuto alcune difficoltà con l'utilizzo di SonarQube, questo sono le metriche calcolate al momento.

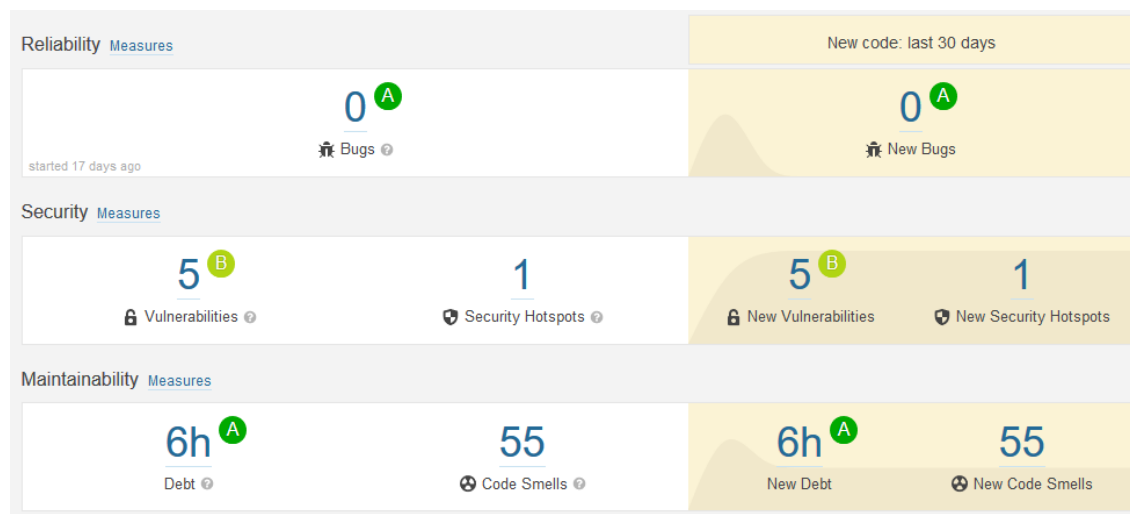


Figura 18: Alcune metriche calcolate da SonarQube

A.4 Revisione di accettazione

A.4.1 Documenti

A.4.1.1 Indice di Gulpease

Per i verbali, sia interni che esterni, è stato deciso di tenere conto della media dei loro indici di Gulpease, in modo da rendere più agevole la lettura del diagramma.

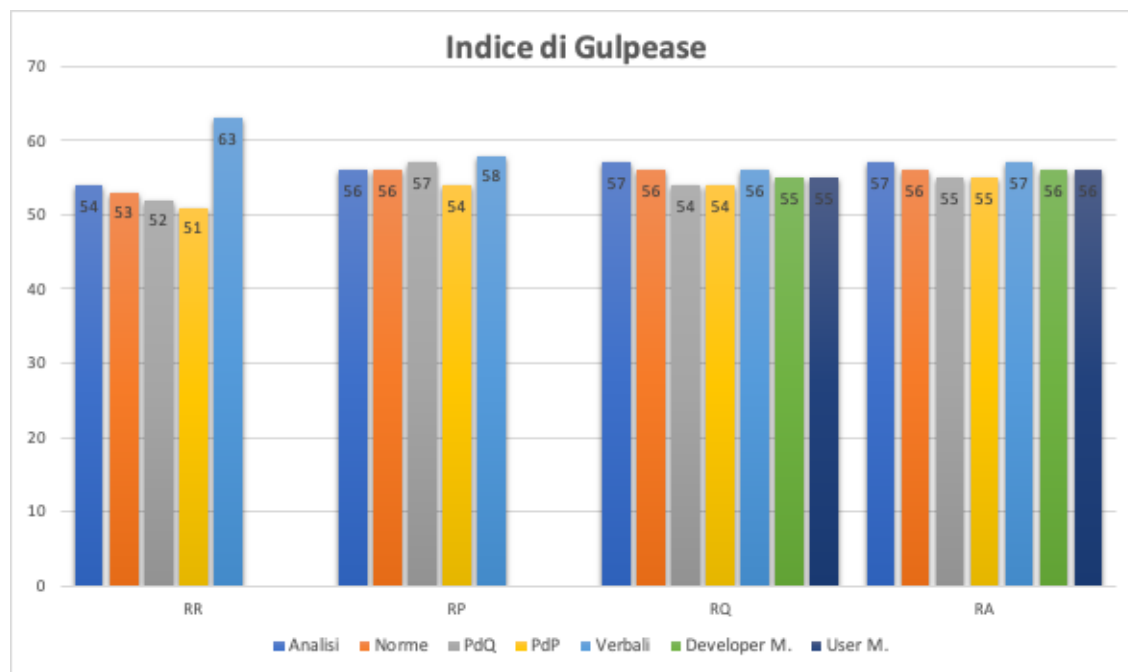


Figura 19: Indice di Gulpease nei periodi di RR, RP, RQ e RA

A.4.2 Processi

A.4.2.1 EAC: Estimated at Completion

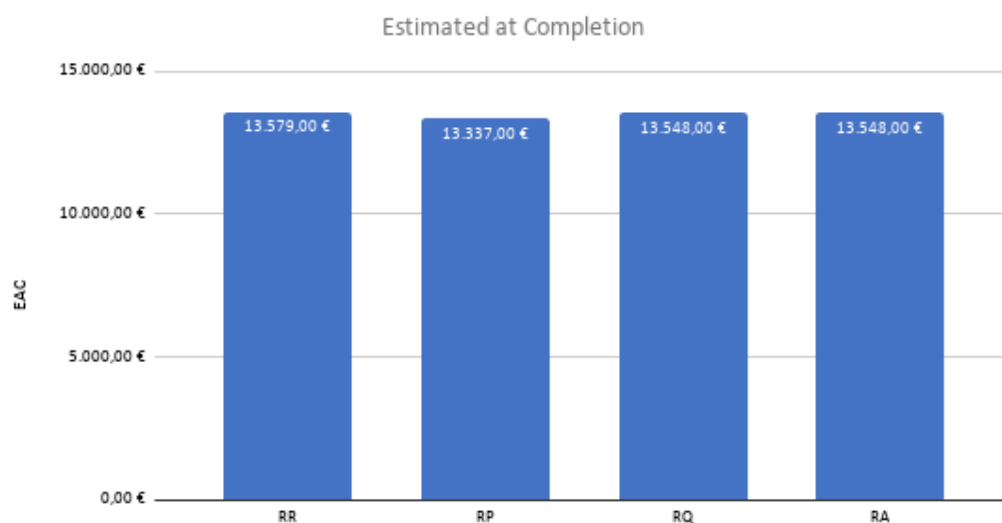


Figura 20: Estimated at Completion nei periodi di RP, RQ e RA

Il costo finale del progetto non ha superato il preventivo iniziale bensì c'è stato un risparmio di 31 €.

A.4.2.2 VAC: Variance at Completion

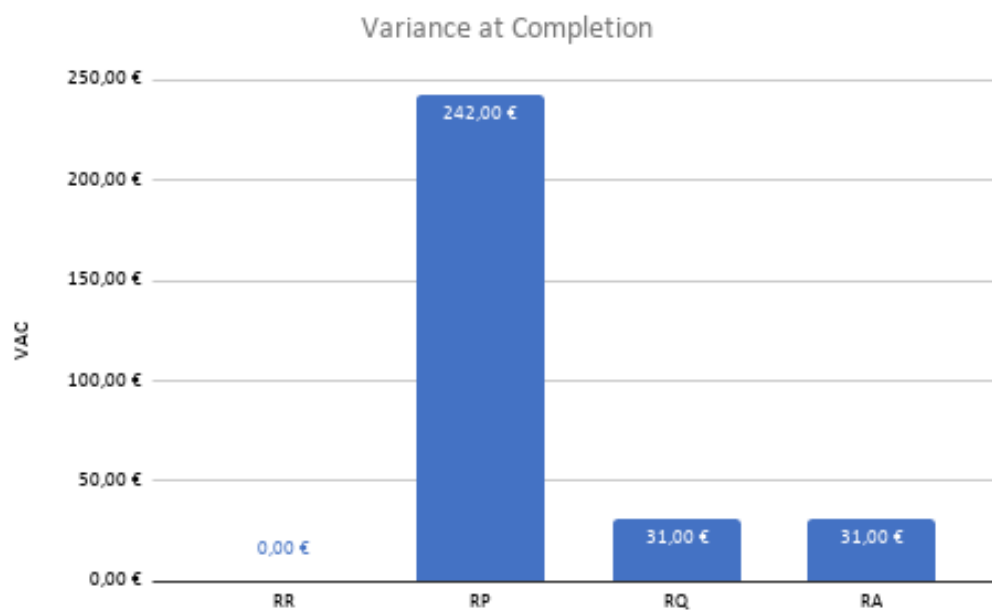


Figura 21: Variance at Completion nei periodi di RP, RQ e RA

Il progetto non ha sforato il budget nel periodo di RA andando in pari con i costi preventivati.

A.4.2.3 BV: Budget Variance

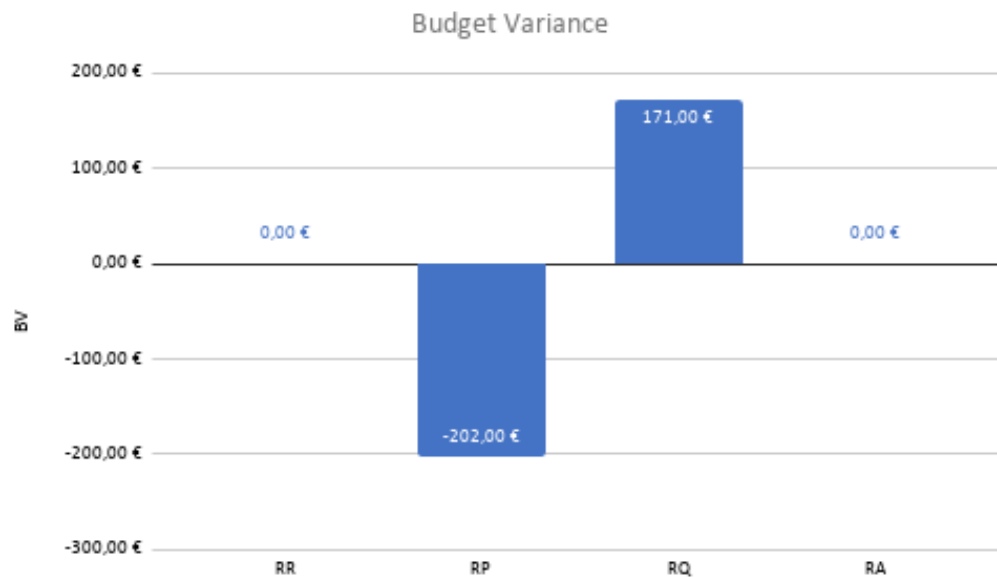


Figura 22: Budget Variance nei periodi di RP, RQ e RA

Nel periodo di RA il valore del Budget Variance è pari a 0 €: il progetto ha quindi speso il costo pianificato.

A.4.2.4 AC: Actual Cost

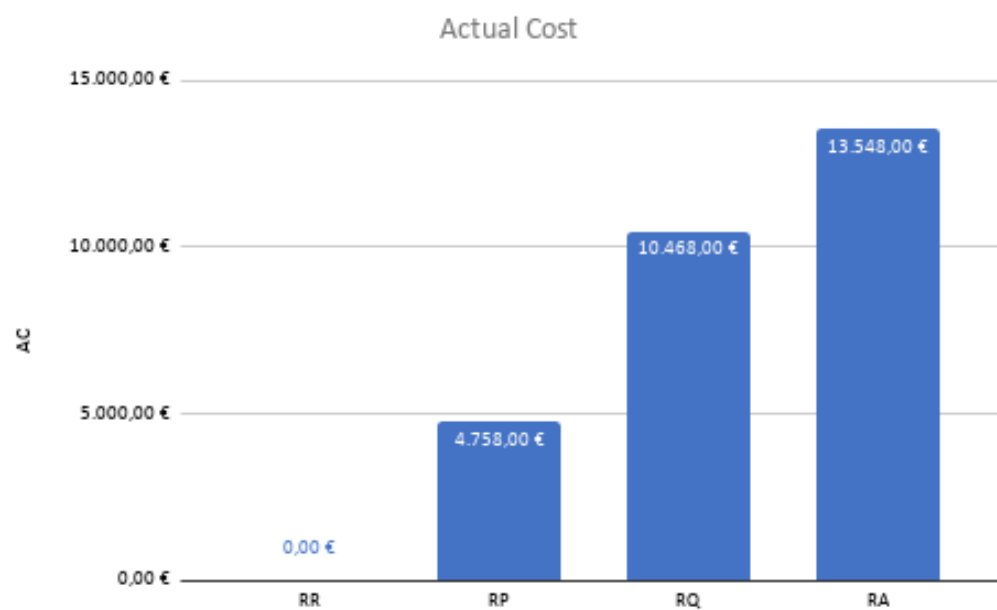


Figura 23: Actual Cost nei periodi di RP, RQ e RA

L'Actual Cost del diagramma è il prezzo finale del progetto.

A.4.3 Software

A.4.3.1 Requisiti

Su un totale di 77 requisiti totali, 14 non sono stati soddisfatti che tuttavia comprendono solamente requisiti opzionali e desiderabili.

In particolare i requisiti desiderabili non soddisfatti sono 11, e i restanti 3 sono requisiti opzionali.

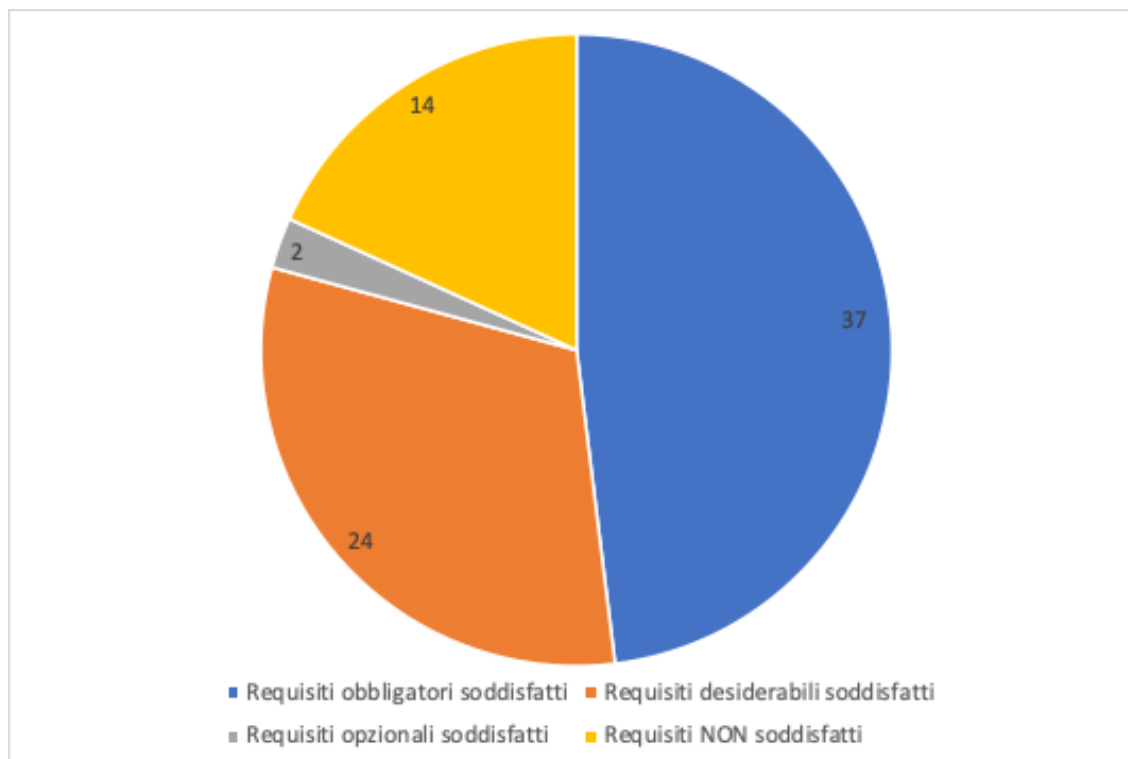


Figura 24: Requisiti soddisfatti NaturalAPI

A.4.3.2 NaturalAPI

Per convenienza è stato deciso di raggruppare i tre moduli in uno solo per alcune metriche, in particolare quelle calcolate con il plugin CodeMR.

WMC: Metodi pesati per classe

La maggior parte dei metodi delle classi ha un valore molto basso, in particolare, delle 75 classi totali, solo 1 classe ha un valore medio-alto. Questo significa che le classi sono ben dimensionate e riutilizzabili.

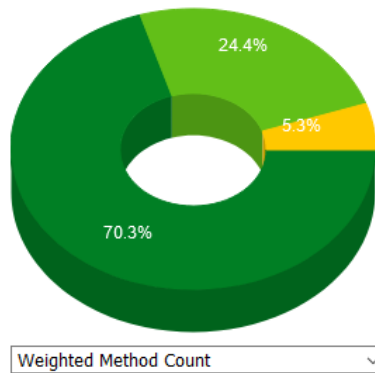


Figura 25: Metrica WMC

DIT: Profondità dell'albero di ereditarietà

Quasi tutte le classi hanno valore molto basso di profondità dell'albero di ereditarietà. Una sola classe raggiunge un valore medio-alto. Le classi così risultano meno complesse da sviluppare, testare e mantenere. Diagramma visualizzabile in figura 25.

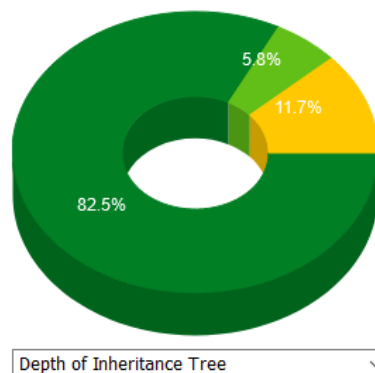


Figura 26: Metrica DIT

NOC: Numero di figli

Tutte le classi hanno un numero di figli basso. In particolare 3 classi hanno un valore leggermente più alto rimanendo comunque medio-basso.

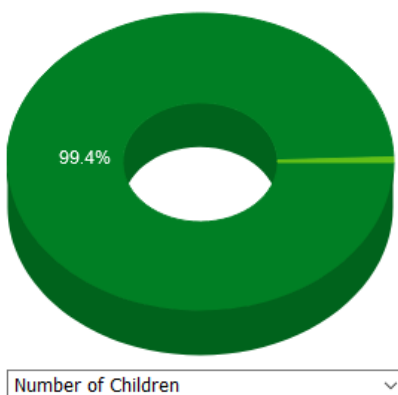


Figura 27: Metrica NOC

RFC: Risposta per classe

Il numero di metodi che possono essere invocati in una classe è basso in tutte le classi

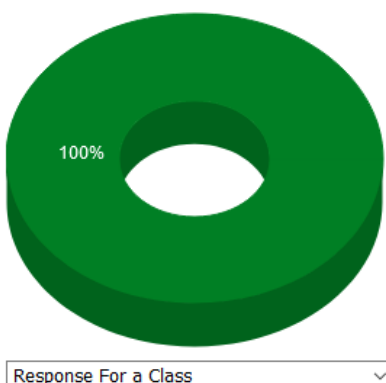


Figura 28: Metrica RFC

LCOM: Mancanza di coesione nei metodi

Come si vede in figura, c'è una bassa mancanza di coesione nei metodi nelle varie classi. Tuttavia è presente comunque un elevato valore medio-basso e una classe con valore medio-alto.

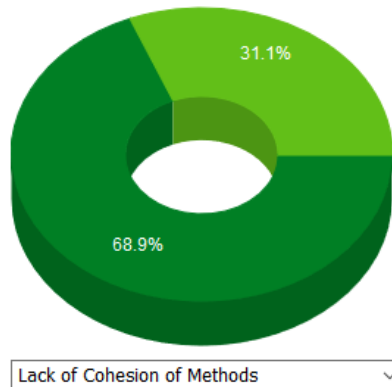


Figura 29: Metrica LCOM

A.4.3.3 CBO: Accoppiamento tra classi

Tra le classi del progetto, esiste un basso accoppiamento tra classi. In pochi casi il valore raggiunge il livello medio-basso, e per solamente 2 classi raggiunge il valore medio-alto.

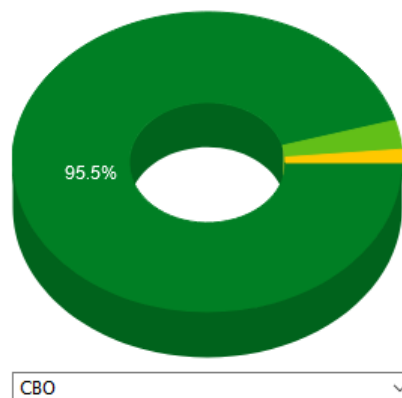


Figura 30: Metrica CBO

A.4.3.4 NaturalAPI Discover

Coverage - SonarQube

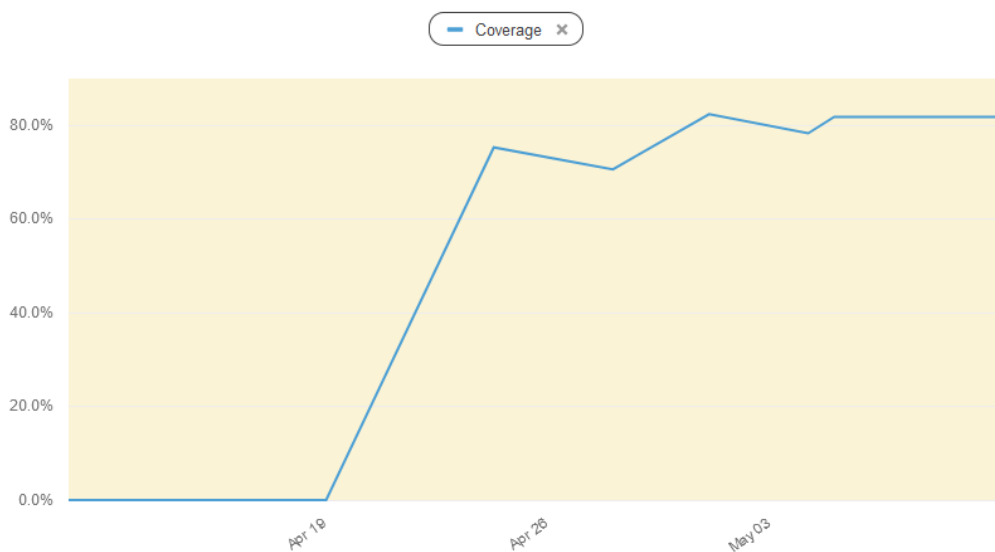


Figura 31: Coverage NaturalAPI Discover

Dalla figura si può notare che il coverage è sempre migliorato con il passare del tempo.

Complessità Ciclomatica - SonarQube

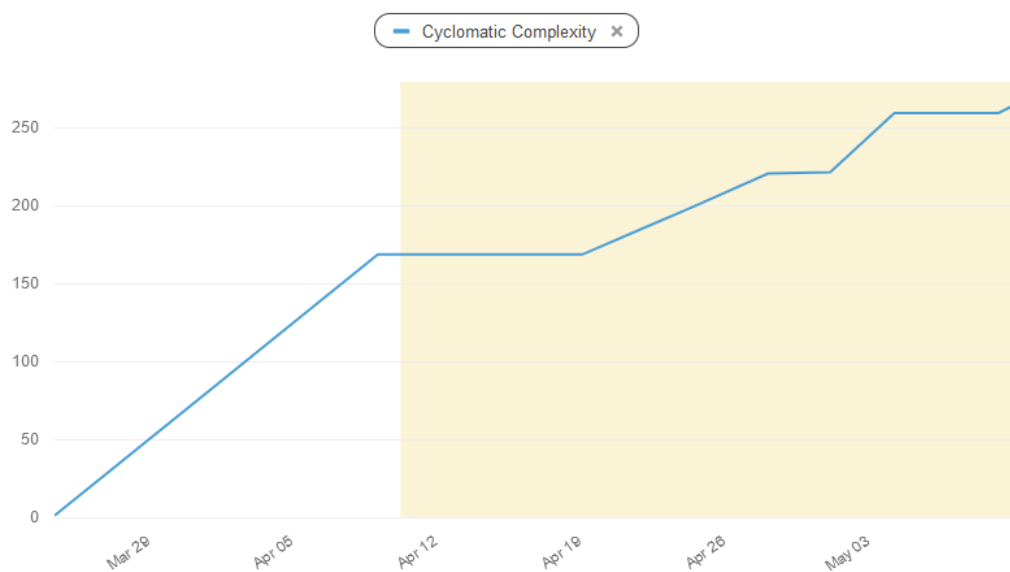


Figura 32: Complessità ciclomatica NaturalAPI Discover

Il dato mostrato in figura è dato dalla somma della complessità ciclomatica di tutte le classi. Questo valore va diviso per il numero delle classi presenti all'interno di NaturalAPI Discover che posseggono un valore di complessità ciclomatica (quindi escludendo le varie interfacce).

Avendo 21 classi, la complessità ciclomatica media per ogni classe sarà quindi all'incirca 13.

Unit Test - SonarQube

| Tests | |
|------------|------|
| Unit Tests | 73 |
| Errors | 0 |
| Failures | 0 |
| Skipped | 0 |
| Success | 100% |
| Duration | 2s |

Figura 33: Unit test NaturalAPI Discover

Maintainability rating - SonarQube

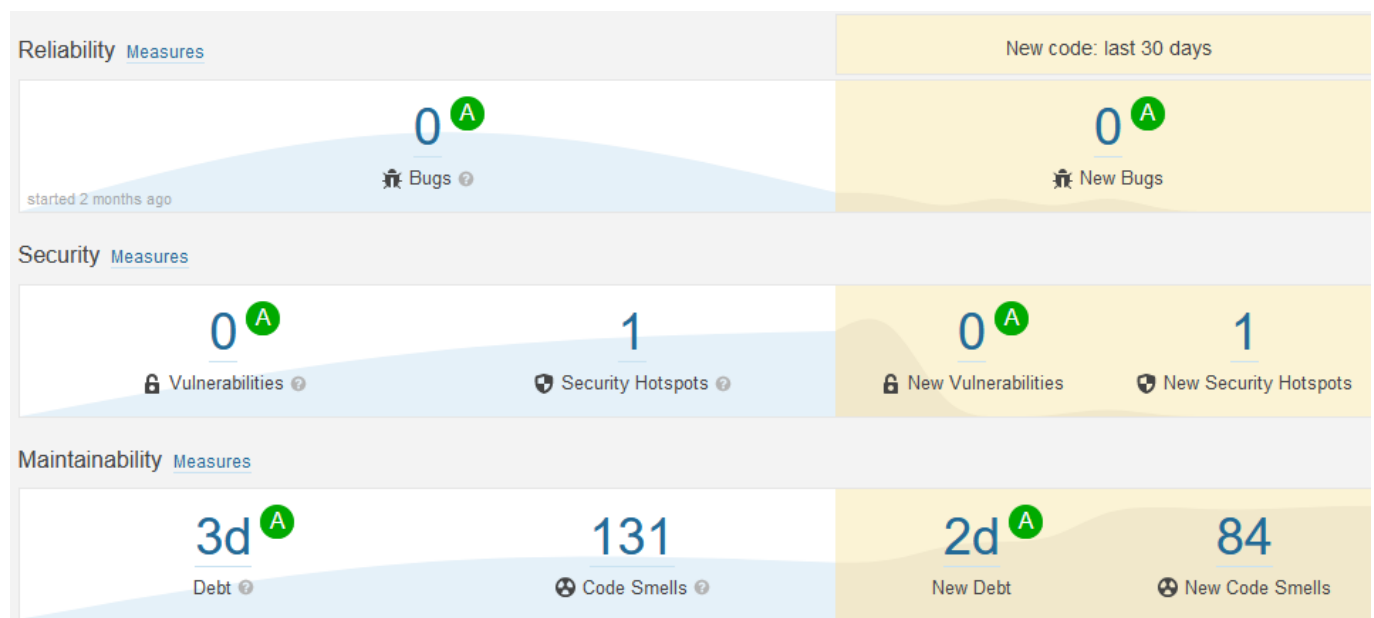


Figura 34: Maintainability rating, Code smells, bug NaturalAPI Discover

Analizzando i code smells, è stato notato che molti di loro non sono particolarmente gravi perché identificano dei problemi minori che possono essere velocemente risolti in una seconda fase.

Ad esempio, ogni package che non è stato nominato con solamente lettere minuscole genera un code smell. Oppure i System.out, comando molto usato per la CLI e per il quale viene segnalato un code smell. Abbiamo ritenuto che

il logger, alternativa segnalata da SonarQube, non è adatto al nostro scopo perchè fa visualizzare una scritta INFO precedente i print, cosa non permessa per come era stata pensata l'implementazione della CLI.

Inoltre, molti di loro sono stati generati in seguito all'integrazione nel sorgente del codice dell'UI Designer messo a disposizione da IntelliJ: quest'ultimo integra nel codice sorgente delle parti di codice per le quali non vi è la possibilità di effettuare modifiche ma sulle quali SonarQube segnala dei code smells.

Ad ogni modo, si tratta di modifiche utili ma non indispensabili e il team ha deciso quindi di non considerare alcuni di loro.

Lines Coverage - SonarQube

| | |
|----------------------|-------|
| Coverage | 80.0% |
| Lines to Cover | 634 |
| Uncovered Lines | 112 |
| Line Coverage | 82.3% |
| Conditions to Cover | 176 |
| Uncovered Conditions | 50 |
| Condition Coverage | 71.6% |

Figura 35: Lines to cover, Uncovered lines, Condition coverage NaturalAPI Discover

A.4.3.5 NaturalAPI Design

Coverage - SonarQube

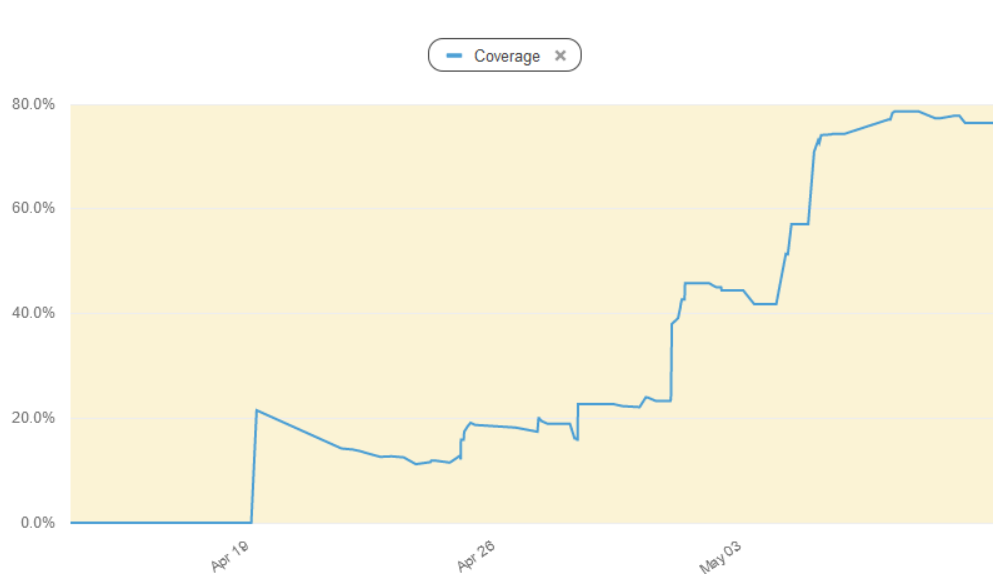


Figura 36: Coverage NaturalAPI Design

Complessità Ciclomatica - SonarQube

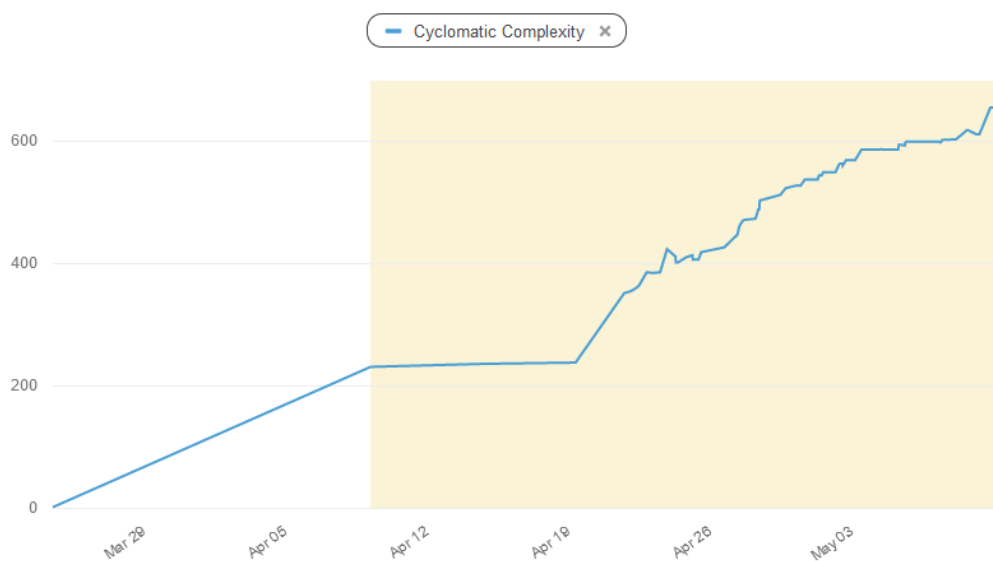


Figura 37: Complessità ciclomatica NaturalAPI Design

Il dato mostrato in figura è dato dalla somma della complessità ciclomatica di tutte le classi. Questo valore va diviso per il numero delle classi presenti all'interno di NaturalAPI Design che posseggono un valore di complessità

cicломatica (quindi escludendo le varie interfacce).

Avendo 41 classi, la complessità cicломatica media per ogni classe sarà quindi all'incirca 16.

Unit Test - SonarQube

| Tests | |
|------------|------|
| Unit Tests | 204 |
| Errors | 0 |
| Failures | 0 |
| Skipped | 0 |
| Success | 100% |
| Duration | 1min |

Figura 38: Unit test NaturalAPI Design

Maintainability rating - SonarQube

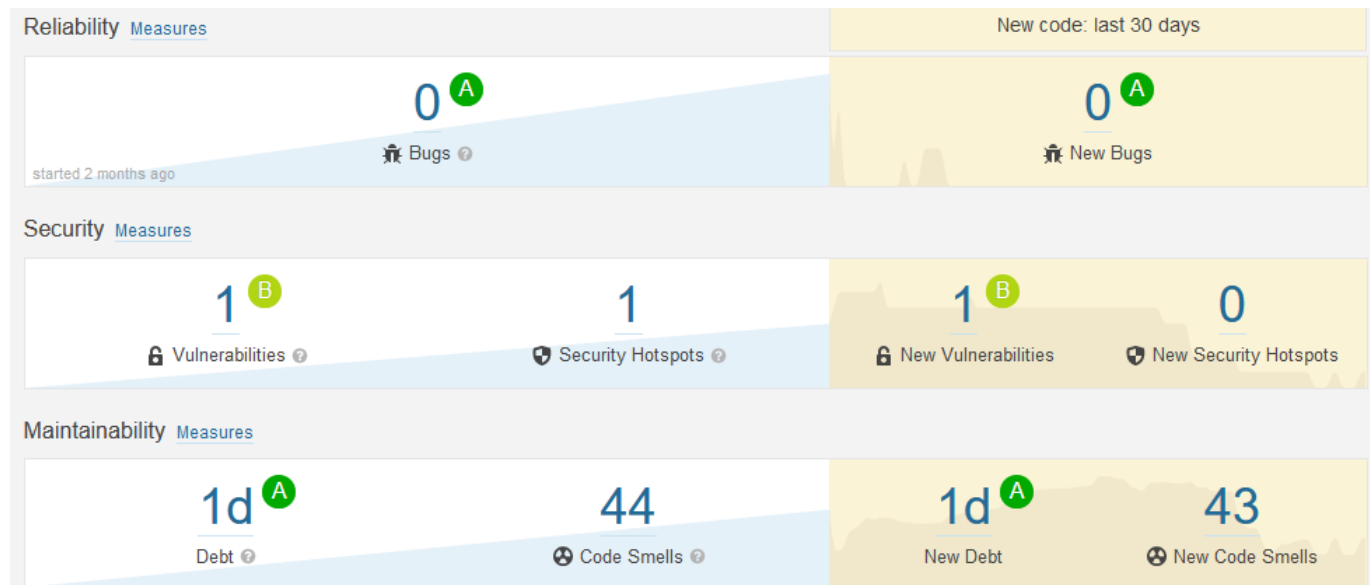


Figura 39: Maintainability rating, Code smells, bug NaturalAPI Design

Analizzando i code smells, è stato notato che molti di loro non sono particolarmente gravi perché identificano dei problemi minori che possono essere velocemente risolti in una seconda fase.

Ad esempio, ogni package che non è stato nominato con solamente lettere minuscole genera un code smell. Oppure i System.out, comando molto usato per la CLI e per il quale viene segnalato un code smell. Abbiamo ritenuto che il logger, alternativa segnalata da SonarQube, non è adatto al nostro scopo perché fa visualizzare una scritta INFO precedente i print, cosa non permessa per come era stata pensata l'implementazione della CLI.

Inoltre, molti di loro sono stati generati in seguito all'integrazione nel sorgente del codice dell'UI Designer messo

a disposizione da IntelliJ: quest'ultimo integra nel codice sorgente delle parti di codice per le quali non vi è la possibilità di effettuare modifiche ma sulle quali SonarQube segnala dei code smells.

Ad ogni modo, si tratta di modifiche utili ma non indispensabili e il team ha deciso quindi di non considerare alcuni di loro.

Lines Coverage - SonarQube

| | |
|----------------------|-------|
| Coverage | 80.8% |
| Lines to Cover | 1,079 |
| Uncovered Lines | 166 |
| Line Coverage | 84.6% |
| Conditions to Cover | 304 |
| Uncovered Conditions | 100 |
| Condition Coverage | 71.1% |

Figura 40: Lines to cover, Uncovered lines, Condition coverage NaturalAPI Design

A.4.3.6 NaturalAPI Develop

Coverage - SonarQube

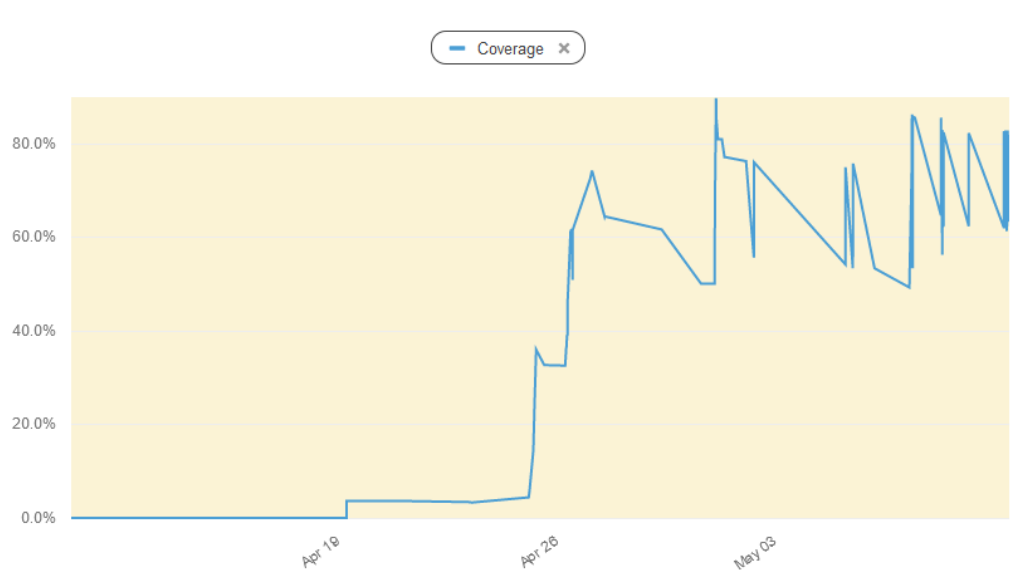


Figura 41: Coverage NaturalAPI Develop

Come si può vedere dal diagramma, questo modulo è stato più ostico rispetto agli altri; il coverage ha avuto dei rallentamenti.

Complessità Ciclomatica - SonarQube

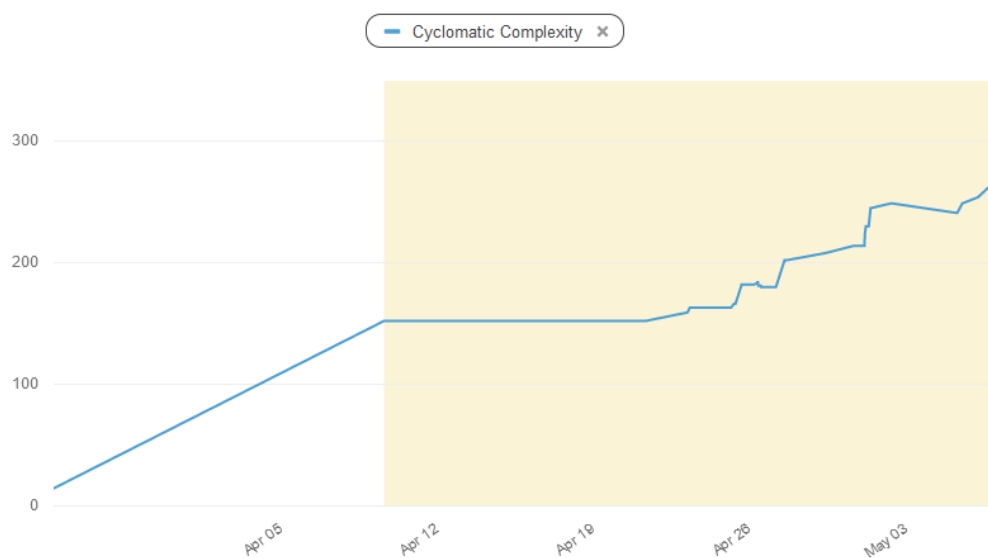


Figura 42: Complessità ciclomatica NaturalAPI Develop

Il dato mostrato in figura è dato dalla somma della complessità ciclomatica di tutte le classi. Questo valore va diviso per il numero delle classi presenti all'interno di NaturalAPI Develop che posseggono un valore di complessità ciclomatica (quindi escludendo le varie interfacce).

Avendo 25 classi, la complessità ciclomatica media per ogni classe sarà quindi all'incirca 12.

Unit Test - SonarQube

| Tests | |
|------------|------|
| Unit Tests | 204 |
| Errors | 0 |
| Failures | 0 |
| Skipped | 0 |
| Success | 100% |
| Duration | 1min |

Figura 43: Unit test NaturalAPI Design

Maintainability rating - SonarQube

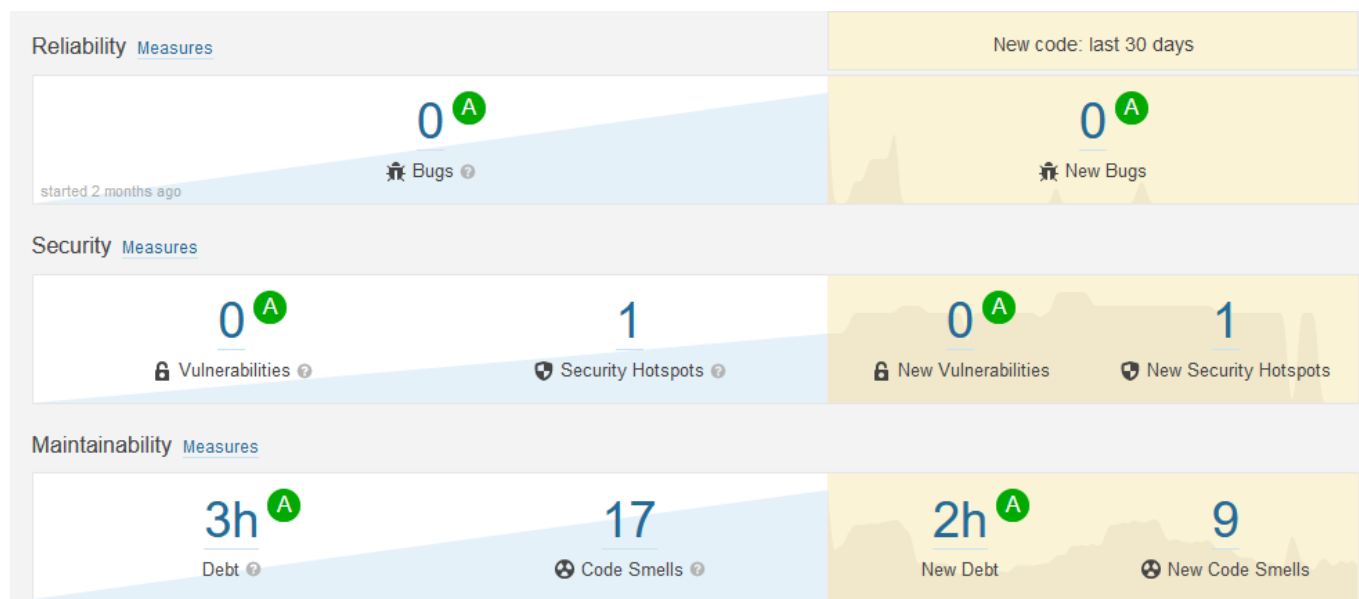


Figura 44: Maintainability rating, Code smells, bug NaturalAPI Develop

Analizzando i code smells, è stato notato che molti di loro non sono particolarmente gravi perché identificano dei problemi minori che possono essere velocemente risolti in una seconda fase.

Ad esempio, ogni package che non è stato nominato con solamente lettere minuscole genera un code smell. Oppure i System.out, comando molto usato per la CLI e per il quale viene segnalato un code smell. Abbiamo ritenuto che il logger, alternativa segnalata da SonarQube, non è adatto al nostro scopo perché fa visualizzare una scritta INFO

precedente i print, cosa non permessa per come era stata pensata l'implementazione della CLI.

Inoltre, molti di loro sono stati generati in seguito all'integrazione nel sorgente del codice dell'UI Designer messo a disposizione da IntelliJ: quest'ultimo integra nel codice sorgente delle parti di codice per le quali non vi è la possibilità di effettuare modifiche ma sulle quali SonarQube segnala dei code smells.

Ad ogni modo, si tratta di modifiche utili ma non indispensabili e il team ha deciso quindi di non considerare alcuni di loro.

Lines Coverage - SonarQube

| | |
|----------------------|-------|
| Coverage | 79.5% |
| Lines to Cover | 772 |
| Uncovered Lines | 138 |
| Line Coverage | 82.1% |
| Conditions to Cover | 170 |
| Uncovered Conditions | 55 |
| Condition Coverage | 67.6% |

Figura 45: Lines to cover, Uncovered lines, Condition coverage NaturalAPI Develop

B Valutazioni per il miglioramento del team

In questa sezione viene riportata la valutazione fatta dal team sul lavoro svolto.

Lo scopo è evidenziare i problemi sorti in questa prima fase di lavoro e valutare come sono stati risolti dal gruppo affinché non si ripetano in futuro. Sono stati tracciati i problemi riguardanti:

- **Organizzazione del team:** sono presenti i problemi sorti sulla comunicazione all'interno del gruppo e con il proponente;
- **Ruoli:** sono presenti i problemi sorti durante lo svolgimento di alcuni ruoli;
- **Strumenti di lavoro:** sono presenti i problemi sorti durante l'uso di alcuni strumenti.

La qualità e l'accuratezza del lavoro è garantita dalla comunicazione e dalla produzione collettiva. Pertanto, ognuno dei membri apporta un proprio contributo per risolvere eventuali problemi. In questo modo ciascuno è libero di esprimere la propria opinione personale e di apportare soluzioni alle difficoltà riscontrate dal gruppo.

Per avere una valutazione quantificabile dello stato delle problematiche rilevate, viene assegnato un valore da 1 a 3 che indica la gravità di ogni problema.

La sezione verrà aggiornata di volta in volta man mano che si riscontrano nuove problematiche.

B.1 Revisione dei requisiti

B.1.1 Valutazioni sull'organizzazione del team

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|--|---|
| Incontro di gruppo: si è riscontrata una certa difficoltà nell'organizzare gli incontri in modo che fossero presenti tutti i membri. | 2 | Necessità di organizzare riunioni in conference call. | Si è deciso di utilizzare "Hangouts" per effettuare riunioni in videocall in modo da essere più flessibili su orari e luoghi di ritrovo. |
| Comunicazione del gruppo: si è riscontrata un po' di confusione nel trattare i vari argomenti nel gruppo <i>Telegram</i> creato inizialmente per conoscerci tra noi membri. | 1 | Necessità di utilizzare un servizio di chat più elastico sui canali. | Si è deciso di passare a <i>Slack</i> per poter dividere i vari canali e non avere i messaggi mescolati tra i vari argomenti. |
| Comunicazioni con il proponente: si è riscontrato il problema di come comunicare con il proponente in modo veloce. | 2 | Necessità di trovare un sistema comune per una comunicazione rapida. | Si è deciso, insieme al proponente, di creare un canale <i>Slack</i> per le comunicazioni più veloci e informali con il fine di organizzarsi per un incontro più formale tramite Skype o Hangout. |

Tabella 11: Tabella valutazione sull'organizzazione del team

B.1.2 Valutazione sugli strumenti di lavoro

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|---|---|
| GitHub: si sono riscontrati diversi conflitti sui file in cui si stava lavorando e alcune problematiche sull'uso di github. | 1 | Necessità di regolamentare meglio l'utilizzo di GitHub. | I membri del gruppo più esperti hanno affiancato e istruito gli altri membri al fine di minimizzare errori di questo tipo. |
| LaTeX: a causa dell'inesperienza di molti sull'uso di questo strumento si sono riscontrate diverse difficoltà nel produrre documenti simili, nell'inserimento di figure e nella costruzione di tabelle. | 1 | Necessità di regolamentare meglio l'utilizzo di LaTeX | I membri del gruppo che sapevano già utilizzare i comandi di Latex hanno affiancato i membri meno esperti dandogli la possibilità di apprendere come effettuare e sviluppare il lavoro. |

Tabella 12: Tabella valutazione sugli strumenti di lavoro

B.2 Revisione di progettazione

B.2.1 Valutazione sull'organizzazione del team

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|---|--|
| Sessione esami: la sessione d'esami ha portato a dedicare meno tempo al progetto, il team si è ritrovato a comprimere diversi compiti in un periodo più ristretto e a doverli svolgere in modo celere. | 2 | Necessità di sapere il tempo che ogni membro del gruppo può dedicare al progetto. | Essenziale è avere ben chiari i giorni di disponibilità di ogni membro del gruppo fino alla fine del progetto, per effettuare una pianificazione che riscontri sempre meno variazioni. |
| Accelerazione compiti: la velocità dello svolgimento dei compiti ha portato i membri del gruppo a non conoscere chiaramente tutte le componenti del progetto, in quanto il lavoro è stato parallelizzato. | 3 | Necessità di spiegare a tutti i membri del team il lavoro svolto | I membri del gruppo che effettuavano modifiche nei prodotti spiegavano agli altri in cosa consistevano. |

Tabella 13: Tabella valutazione sull'organizzazione del team

B.2.2 Valutazione sugli strumenti di lavoro

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|---|--|
| GitHub: l'aggiunta del contenuto del PoC ha creato confusione nel repository. | 1 | Necessità di maggior ordine all'interno della repository. | Si è deciso di creare un nuovo account GitHub e una nuova repository separata per il codice. |

Tabella 14: Tabella valutazione sugli strumenti di lavoro

B.3 Revisione di qualifica

B.3.1 Valutazione sull'organizzazione del team

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|--|---|
| Chiusura Università: la chiusura dell'Università ha privato il team della possibilità di riunirsi fisicamente in un'aula per confrontarsi di persona su diverse scelte progettuali. | 2 | Necessità di avere spesso confronti tra i membri del team per alcune decisioni di progettazione. | Il gruppo ha deciso di trovarsi quasi tutti i pomeriggi su Hangouts in modo informale per poter discutere le soluzioni pensate. |

Tabella 15: Tabella valutazione sull'organizzazione del team

B.3.2 Valutazione sui ruoli

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|--|--|
| Ruolo di Progettista: lo studio della Clean Architecture e l'identificazione dei design pattern ha occupato la maggior parte del tempo di lavoro. È stato dedicato del tempo allo studio degli stessi. Alcune ore, invece, sono state dedicate alla realizzazione dei diagrammi delle classi e di sequenza. | 3 | Necessità di aiuto per i Progettisti per completare il lavoro. | Per evitare ritardi sul lavoro, chi ha svolto il ruolo di Analista e ha avanzato ore, ne ha dedicato alcune allo studio dei design pattern e, nel tempo rimanente, ha affiancato i Progettisti nella loro realizzazione. |

Tabella 16: Tabella valutazione dei ruoli

B.3.3 Valutazione sugli strumenti di lavoro

| Problema | Gravità | Soluzione | Miglioria attuata |
|--|---------|---|--|
| GUI Github: con l'aggiunta di tanti nuovi elementi nella repo si sono riscontrate diverse difficoltà sul tracciamento e l'organizzazione dei commit. | 3 | Necessità di maggior organizzazione sui commit. | Il team ha deciso di passare a GitKraken e i membri che già lo conoscevano hanno istruito il resto del gruppo. |
| IDE: in seguito alla TB il prof. Cardin ha consigliato di non utilizzare Eclipse come IDE. | 2 | Necessità di passare a un nuovo IDE. | Il team ha deciso di lavorare su IntelliJ IDEA e i membri che già lo conoscevano hanno istruito il resto del gruppo. |

Tabella 17: Tabella valutazione sugli strumenti di lavoro

B.4 Revisione di accettazione

B.4.1 Valutazione sull'organizzazione del team

| Problema | Gravità | Soluzione | Miglioria attuata |
|--|---------|---|---|
| Velocità di confronto con il proponente: a causa dei ristretti tempi di consegna si è cercato un modo rapido per avere un riscontro con il proponente. | 3 | Necessità di avere un riscontro in tempi brevi. | Il team ha deciso di velocizzare le modalità di comunicazione con il proponente creando dei video commentati da potergli mandare per avere un feedback sul lavoro svolto senza dover organizzare una videocall. |

Tabella 18: Tabella valutazione sull'organizzazione del team

B.4.2 Valutazione sui ruoli

| Problema | Gravità | Soluzione | Miglioria attuata |
|---|---------|---|--|
| Ruolo di Verificatore: a causa della poca esperienza, questo ruolo è stato più ostico del previsto e ha comportato maggior lavoro per lo studio delle tecnologie atte ad effettuare i test. | 2 | Necessità di aiuto per i Verificatori per completare il lavoro. | Per evitare ritardi sul lavoro, chi ha svolto il ruolo di Responsabile e Amministratore e ha avanzato ore, ne ha dedicato alcune per studiare le tecnologie per la verifica ed ha affiancato i Verificatori nel loro lavoro. |

Tabella 19: Tabella valutazione dei ruoli

B.4.3 Valutazione sugli strumenti di lavoro

| Problema | Gravità | Soluzione | Miglioria attuata |
|--|---------|---|--|
| Strumenti per l'integrazione continua: i Programmatori devono essere affiancati da strumenti per il rilevamento degli errori in modo rapido. | 2 | Necessità di utilizzare degli strumenti per il rilevamento di errori. | Utilizzando SonarQube e Travis CI sono stati velocizzati i rilevamenti degli errori così da essere corretti rapidamente dai Programmatori. |

Tabella 20: Tabella valutazione sugli strumenti di lavoro