

Piano di Qualifica

2023-12-20 — v0.1.0



overture.unipd@gmail.com

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin <i>Zextras</i> Gruppo <i>Overture</i>
Responsabile	Alex Vedovato
Redattori	Eleonora Amadori Michele Bettin Francesco Costantino Bulychov
Verificatori	Eleonora Amadori Francesco Costantino Bulychov Riccardo Fabbian Francesco Furno

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.1.0	2023-12-20	Eleonora Amadori	Francesco Costantino Bulychov	Stesura della sezione 'Iniziative di automiglioramento per la qualità'
0.0.5	2023-12-20	Eleonora Amadori	Francesco Furno	Stesura della sezione 'Cruscotto di valutazione della qualità'
0.0.4	2023-12-20	Eleonora Amadori	Francesco Furno	Stesura della sezione 'Metodologie di testing'
0.0.3	2023-12-19	Michele Bettin	Riccardo Fabbian	Stesura della sezione 'Qualità di prodotto' in 'Obiettivi metrici di qualità'
0.0.2	2023-12-19	Michele Bettin	Riccardo Fabbian	Stesura della sezione 'Qualità di processo' in 'Obiettivi metrici di qualità'
0.0.1	2023-11-15	Francesco Costantino Bulychov	Eleonora Amadori	Struttura di base ed introduzione

Indice

1) Introduzione	5
1.1) Scopo del documento	5
1.2) Scopo del prodotto	5
1.3) Glossario	5
1.4) Riferimenti	5
1.4.1) Riferimenti normativi	5
1.4.2) Riferimenti informativi	5
2) Obiettivi metrici di qualità	6
2.1) Qualità di processo	6
2.1.1) Processi primari	6
2.1.1.1) Fornitura	6
2.1.1.2) Sviluppo	6
2.1.2) Processi di supporto	6
2.1.2.1) Documentazione	6
2.1.2.2) Verifica	6
2.1.2.3) Gestione della qualità	6
2.1.3) Processi organizzativi	7
2.1.3.1) Gestione dei processi	7
2.2) Qualità di prodotto	7
2.2.1) Funzionalità	7
2.2.2) Affidabilità	7
2.2.3) Usabilità	7
2.2.4) Efficienza	7
2.2.5) Manutenibilità	7
3) Metodologie di testing	8
3.1) Codice dei test	8
3.2) Test di unità	8
4) Cruscotto di valutazione della qualità	9
4.1) MPC06 - Estimated at Completion (EAC)	9
4.2) MPC01 - Earned Value (EV) e MPC02 - Planned Value (PV)	10
4.3) MPC03 - Actual Cost (AC) e MPC07 - Estimate to Complete (ETC)	11
4.4) MPC04 - Cost Variance (CV) e MPC05 - Schedule Variance (SV)	12
4.5) MPC08 - Requirements stability index (RSI)	13
4.6) MPC11 - Indice Gulpease	14
4.7) MPC12 - Correttezza Ortografica	15
4.8) MPC13 - Code Coverage	16
4.9) MPC14 - Passed Test Cases Percentage	17
4.10) MPC15 - Quality Metrics Satisfied	18
4.11) MPC16 - Non-Calculated Risk	19
4.12) MPC17 - Efficienza Temporale	20
5) Iniziative di automiglioramento per la qualità	21
5.1) Introduzione	21
5.2) Problemi legati all'organizzazione generale	21
5.3) Valutazione sui ruoli	21
5.4) Valutazione sugli strumenti	22

Lista della immagini

Lista delle tabelle

1) Introduzione

1.1) Scopo del documento

Questo documento presenta le strategie di verifica e validazione implementate per garantire la qualità del prodotto e dei processi coinvolti nel progetto in questione. Data la natura evolutiva del documento, i contenuti saranno ampliati e modificati nel corso del tempo.

Saranno inoltre riportati i risultati delle verifiche effettuate sul prodotto, con l'obiettivo di correggere tempestivamente eventuali problematiche riscontrate.

1.2) Scopo del prodotto

Il prodotto mira principalmente a consentire all'azienda proponente di valutare se sia conveniente dedicare tempo e risorse all'integrazione del protocollo JMAP nel loro prodotto principale chiamato Carbonio, una soluzione di collaborazione online centrata sulla gestione delle email. Infatti JMAP è un protocollo di comunicazione progettato per semplificare l'interazione tra client e server nelle applicazioni di posta elettronica.

Al momento, Carbonio utilizza protocolli standard come IMAP, POP e Exchange Active Sync, perciò l'implementazione di JMAP potrebbe potenzialmente garantire maggiori funzionalità ed efficienza a un costo più contenuto.

1.3) Glossario

Per evitare ambiguità o incomprensioni riguardanti la terminologia usata nel documento, è stato deciso di adottare un glossario in cui vengono riportate le varie definizioni. In questa maniera in esso verranno riportati tutti i termini specifici del dominio d'uso con relativi significati.

La presenza di un termine all'interno del Glossario viene indicata applicando *questo stile*.

1.4) Riferimenti

1.4.1) Riferimenti normativi

- Norme di Progetto
- **Capitolato d'appalto C8:** JMAP, il nuovo protocollo standard per la comunicazione email
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C8.pdf>

1.4.2) Riferimenti informativi

- **ISO/IEC 9126**
https://en.wikipedia.org/wiki/ISO/IEC_9126
- **ISO/IEC 12207**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

2) Obiettivi metrici di qualità

Ogni processo viene valutato mediante l'applicazione di metriche specifiche, le cui definizioni sono dettagliate nelle Norme di Progetto. Questa sezione delinea i criteri che le metriche devono rispettare per essere valutate come accettabili o eccellenti.

2.1) Qualità di processo

La qualità di processo è esigenza primaria nello sviluppo software, difatti per poter avere un prodotto finale di qualità è necessario trovare alla base un'applicazione rigorosa di best practice ben definite che ci permettano di svolgere nel miglior modo possibile l'insieme delle attività da effettuare.

2.1.1) Processi primari

2.1.1.1) Fornitura

Metrica	Nome	Valore accettabile	Valore ottimo
MPC01	Earned value (EV)	≥ 0	$\leq EAC$
MPC02	Planned value (PV)	≥ 0	$\leq \text{Budget At Completion (BAC)}$
MPC03	Actual cost (AC)	≥ 0	$\leq EAC$
MPC04	Cost variance (CV)	$\geq -7.5\%$	$\geq 0\%$
MPC05	Schedule variance (SV)	$\geq -7.5\%$	$\geq 0\%$
MPC06	Estimated at completion (EAC)	Errore del $\pm 3\%$ rispetto al BAC	Esattamente pari al BAC
MPC07	Estimate to complete (ETC)	≥ 0	$\leq EAC$

2.1.1.2) Sviluppo

Metrica	Nome	Valore accettabile	Valore ottimo
MPC08	Requirements stability index (RSI)	$\geq 75\%$	100%
MPC09	Structural Fan-In (SFIn)	-	Va massimizzato
MPC10	Structural Fan-Out (SFOU)	-	Va minimizzato

2.1.2) Processi di supporto

2.1.2.1) Documentazione

Metrica	Nome	Valore accettabile	Valore ottimo
MPC11	Indice Goupease	$\geq 60\%$	$\geq 80\%$
MPC12	Correttezza ortografica	0 errori	0 errori

2.1.2.2) Verifica

Metrica	Nome	Valore accettabile	Valore ottimo
MPC13	Code coverage	$\geq 80\%$	100%
MPC14	Passed test cases percentage	100%	100%

2.1.2.3) Gestione della qualità

Metrica	Nome	Valore accettabile	Valore ottimo
MPC15	Quality metrics satisfied	$\geq 85\%$	100%

2.1.3) Processi organizzativi

2.1.3.1) Gestione dei processi

Metrica	Nome	Valore accettabile	Valore ottimo
MPC16	Non-calculated risk	≤ 3	0
MPC17	Efficienza temporale	≤ 3	≤ 1

2.2) Qualità di prodotto

La qualità di prodotto si riferisce all'insieme delle caratteristiche di un'entità risultante dallo sviluppo software, che ne determinano la capacità di soddisfare esigenze sia esplicite che implicite. In altre parole, essa è la misura in cui un prodotto risponde alle aspettative del cliente o agli standard prestabiliti.

2.2.1) Funzionalità

Metrica	Nome	Valore accettabile	Valore ottimo
MPD01	Copertura dei requisiti obbligatori	100%	100%
MPD02	Copertura dei requisiti desiderabili	$\geq 50\%$	100%
MPD03	Copertura dei requisiti opzionali	$\geq 0\%$	$\geq 50\%$

2.2.2) Affidabilità

Metrica	Nome	Valore accettabile	Valore ottimo
MPD04	Code coverage	$\geq 80\%$	100%
MPD05	Branch coverage	$\geq 50\%$	$\geq 80\%$
MPD06	Statement coverage	$\geq 60\%$	$\geq 80\%$
MPD07	Failure density	100%	100%

2.2.3) Usabilità

Metrica	Nome	Valore accettabile	Valore ottimo
MPD08	Facilità di utilizzo	≤ 3 errori commessi nell'interazione	0 errori commessi nell'interazione
MPD09	Tempo di apprendimento	≤ 15 minuti	≤ 5 minuti

2.2.4) Efficienza

Metrica	Nome	Valore accettabile	Valore ottimo
MPD10	Utilizzo risorse	$\geq 75\%$	100%

2.2.5) Manutenibilità

Metrica	Nome	Valore accettabile	Valore ottimo
MPD11	Complessità ciclomatica	1-10	11-20
MPD12	Code smell	0	0
MPD13	Coefficient of Coupling (COC)	$\leq 30\%$	$\leq 10\%$

3) Metodologie di testing

In questa sezione vengono elencati i test eseguiti sul prodotto, necessari per dimostrare che i vincoli individuati all'interno del documento *Analisi dei Requisiti* siano soddisfatti.

Come anticipato all'interno delle *Norme di Progetto*, i test si dividono in:

- **Test di unità:** definiti sulle unità software autonome più piccole, sono implementati principalmente durante la progettazione;
- **Test di integrazione:** successivi ai test di unità, servono a verificare la corretta integrazione tra le diverse unità software;
- **Test di sistema:** garantiscono il corretto funzionamento del sistema. In particolare, verificano che tutti i requisiti individuati funzionino;
- **Test di accettazione:** alla presenza del committente, servono a verificare che il prodotto finale soddisfi tutti i requisiti.

Per la prima revisione del prodotto (RTB) vengono inseriti esclusivamente i test di **unità** implementati e verificati per il Proof of Concept (PoC).

3.1) Codice dei test

Ogni test è associato ad un codice univoco definito nel seguente formato:

T[Tipologia]-[Numero]

Dove **Tipologia** indica la tipologia del test:

- **U:** di unità;
- **I:** di integrazione;
- **S:** di sistema;
- **A:** di accettazione.

Ogni test ha uno **Stato**, che può essere:

- **V:** verificato. Il test ha esito positivo;
- **E:** non verificato. Il test ha esito negativo;
- **NI:** non implementato.

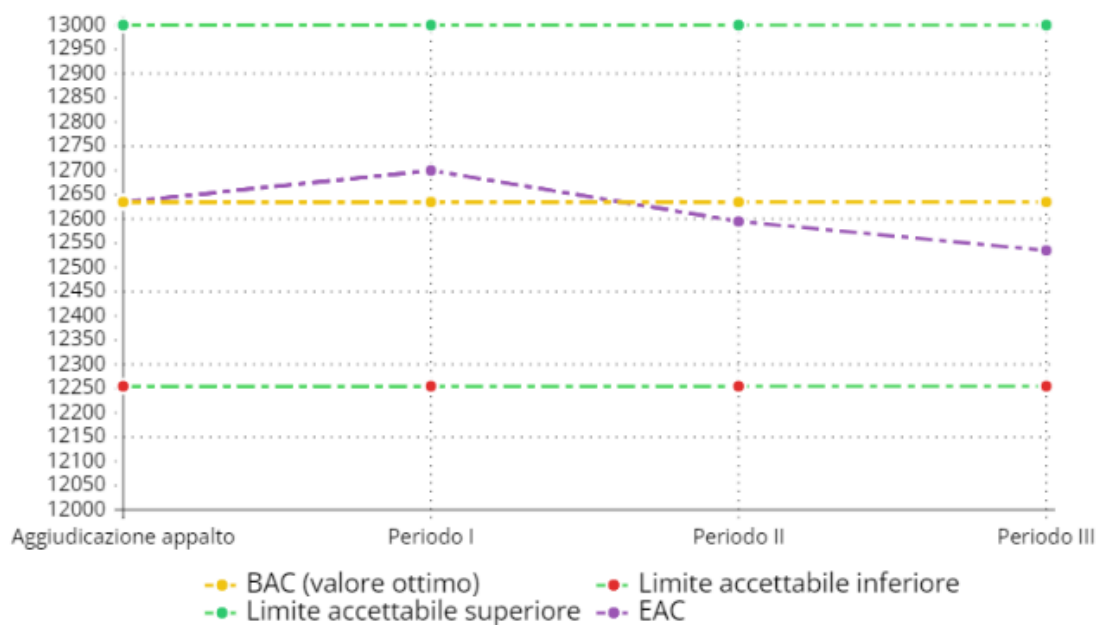
3.2) Test di unità

Sono stati utilizzati i framework di unit testing **JUnit** e **Mockito**.

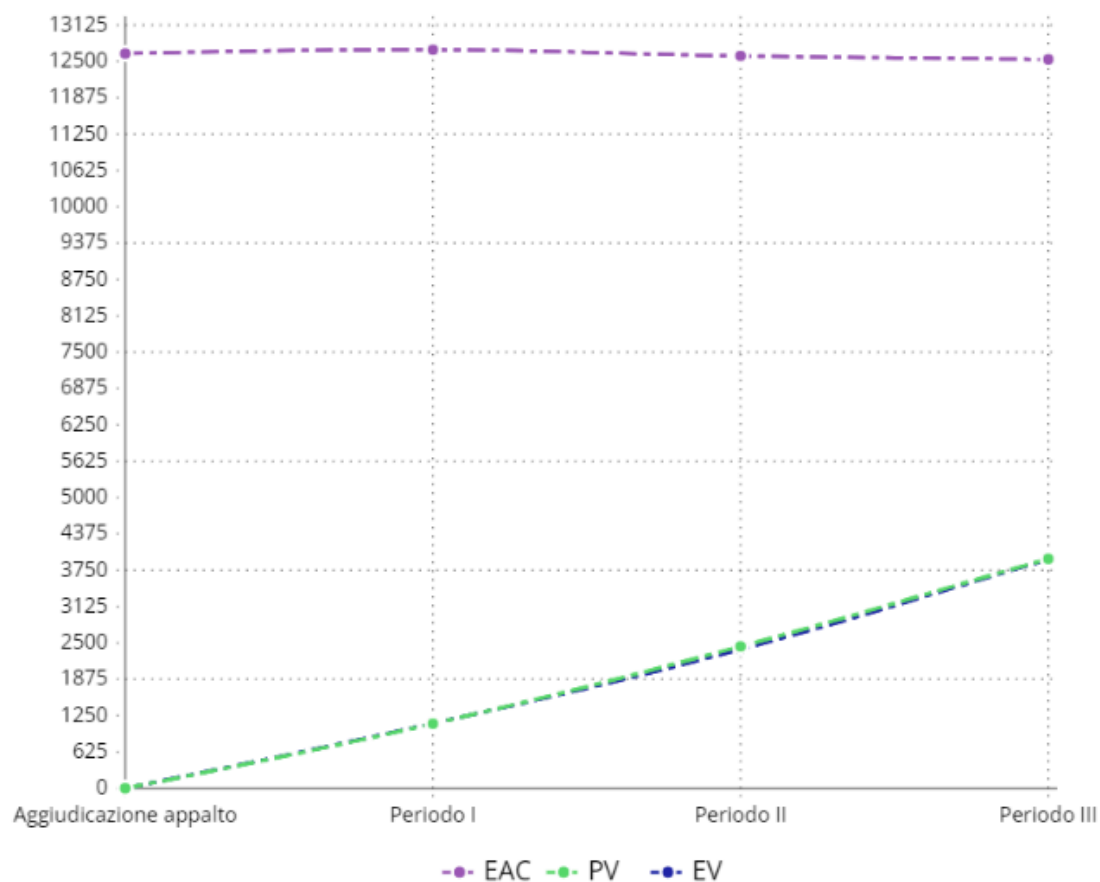
Codice	Descrizione	Stato
TU-1	Verificare che l'oggetto mail venga inserito correttamente nel database	V
TU-2	Verificare che l'oggetto mail inviato dal database venga ricevuto correttamente	V

4) Cruscotto di valutazione della qualità

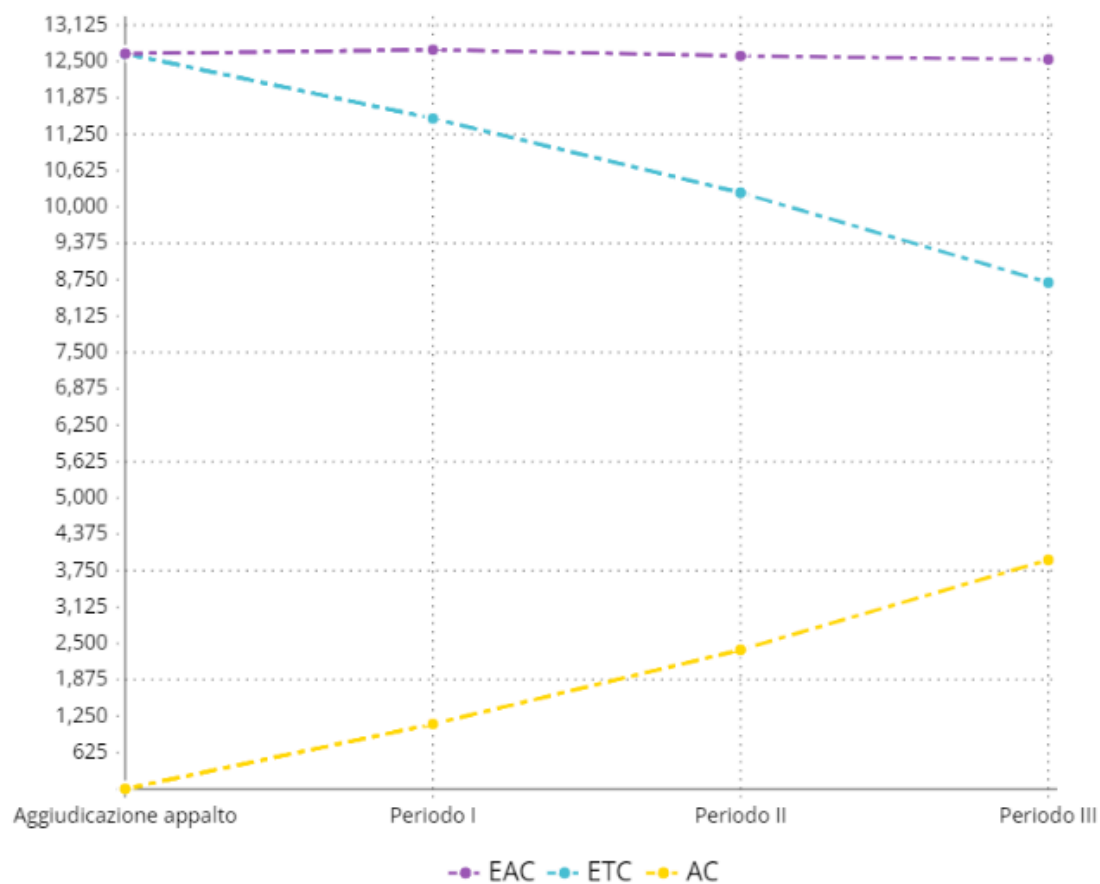
4.1) MPC06 - Estimated at Completion (EAC)



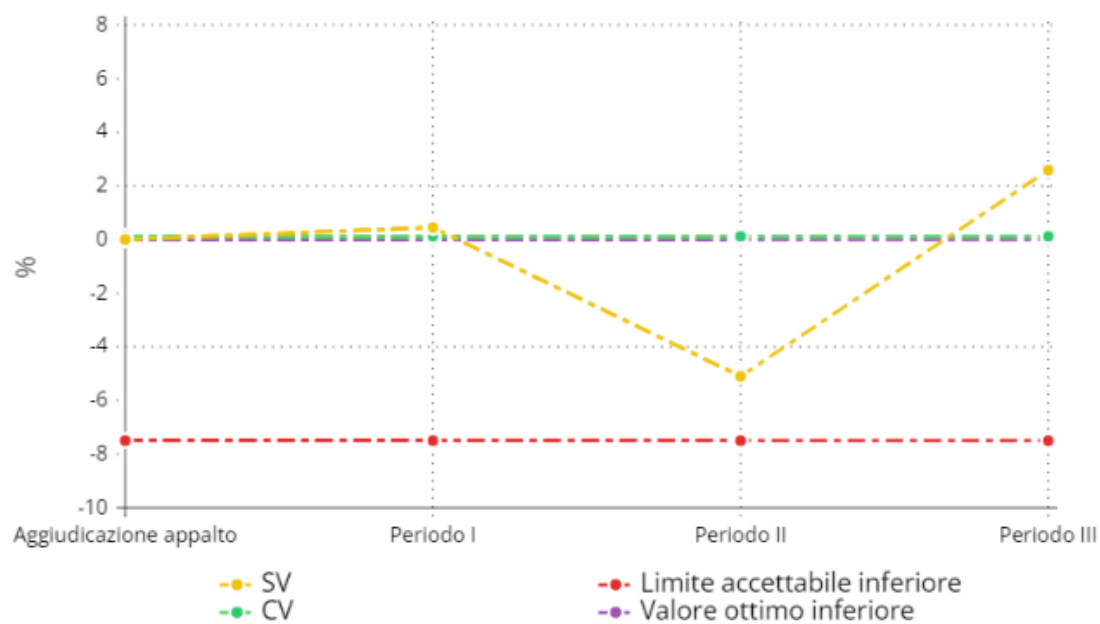
4.2) MPC01 - Earned Value (EV) e MPC02 - Planned Value (PV)



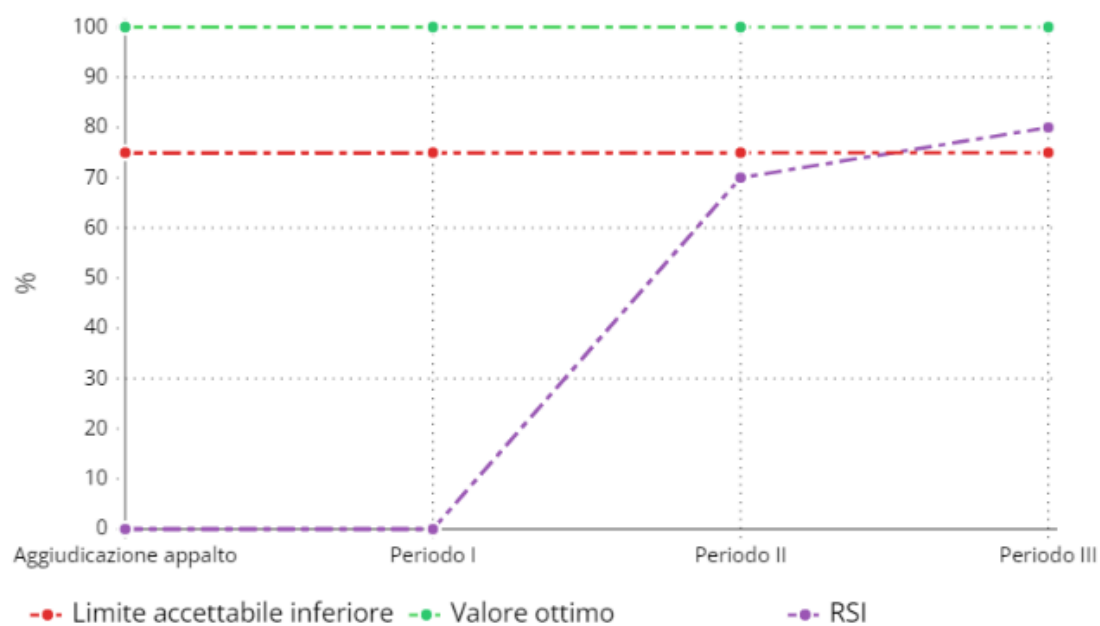
4.3) MPC03 - Actual Cost (AC) e MPC07 - Estimate to Complete (ETC)



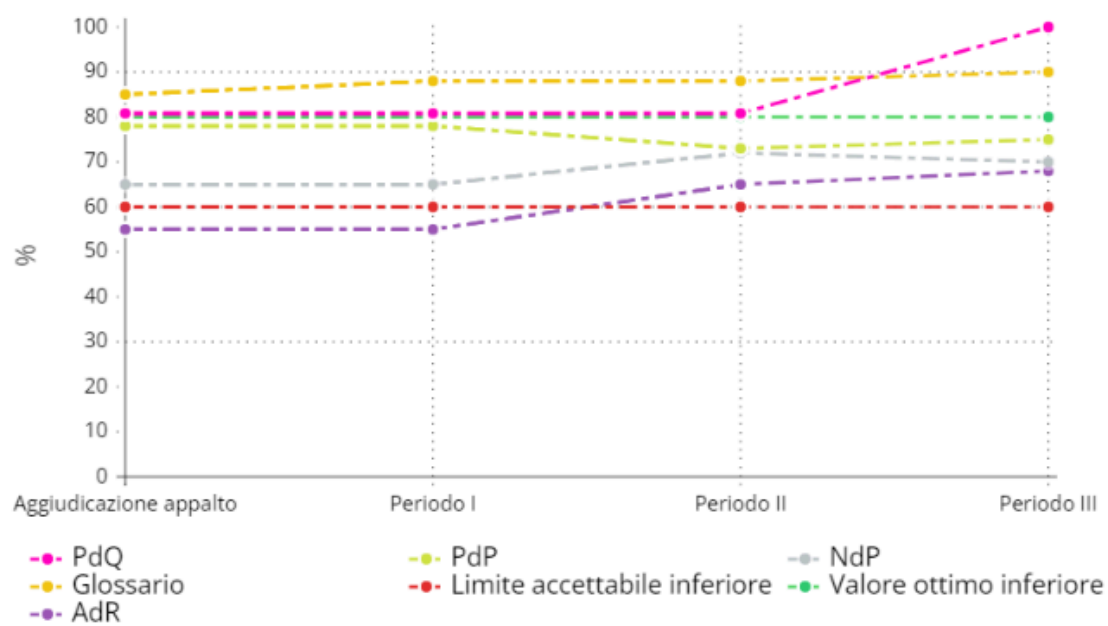
4.4) MPC04 - Cost Variance (CV) e MPC05 - Schedule Variance (SV)



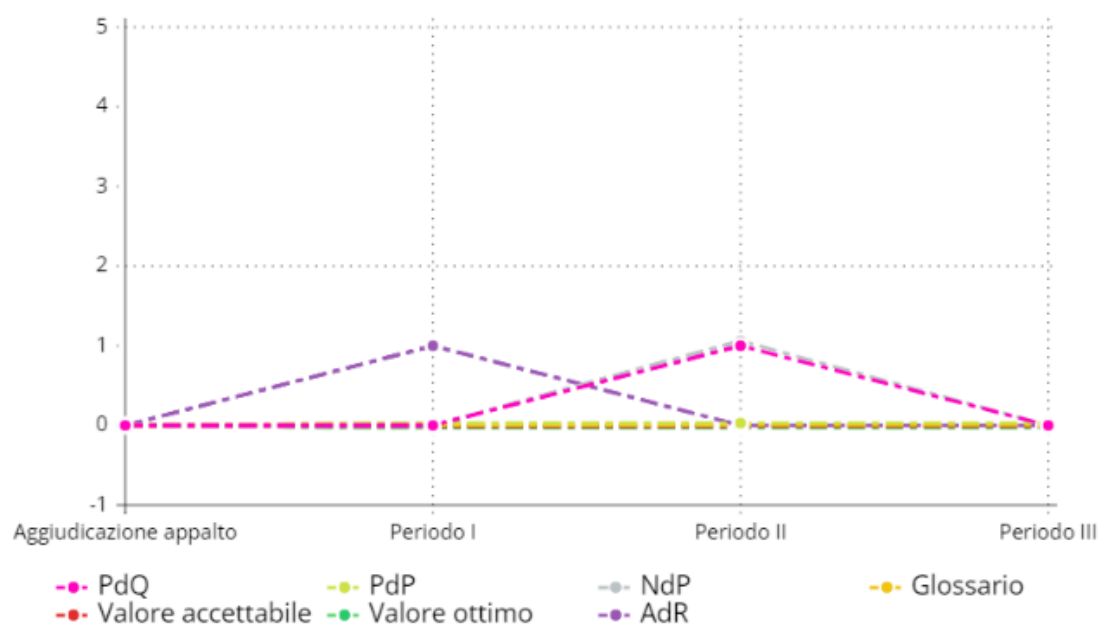
4.5) MPC08 - Requirements stability index (RSI)



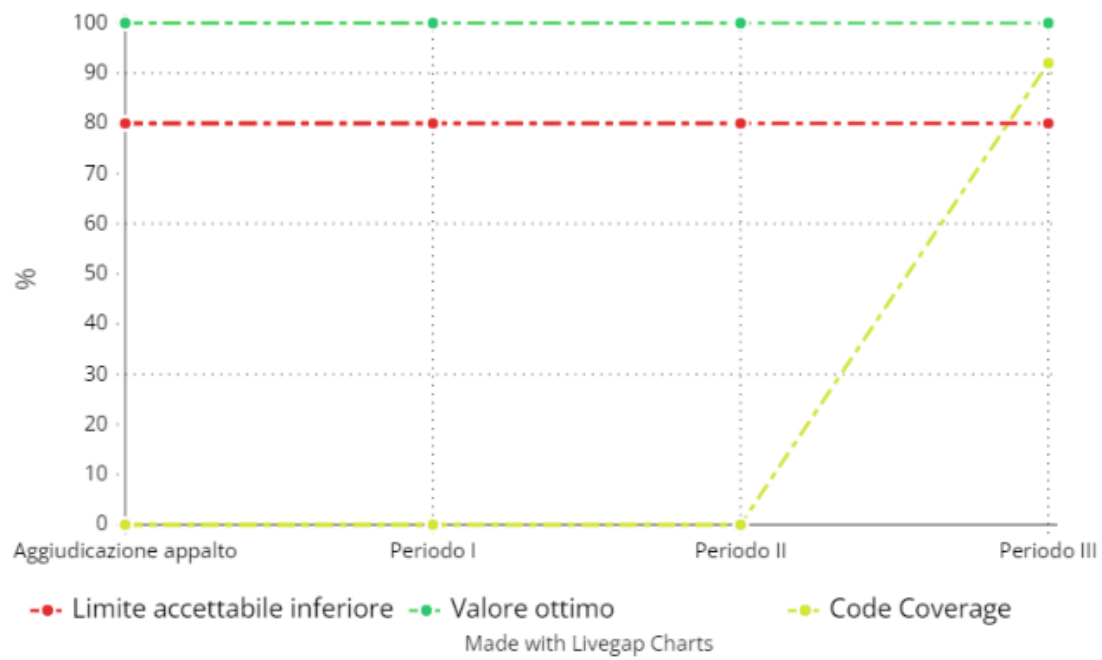
4.6) MPC11 - Indice Gulpease



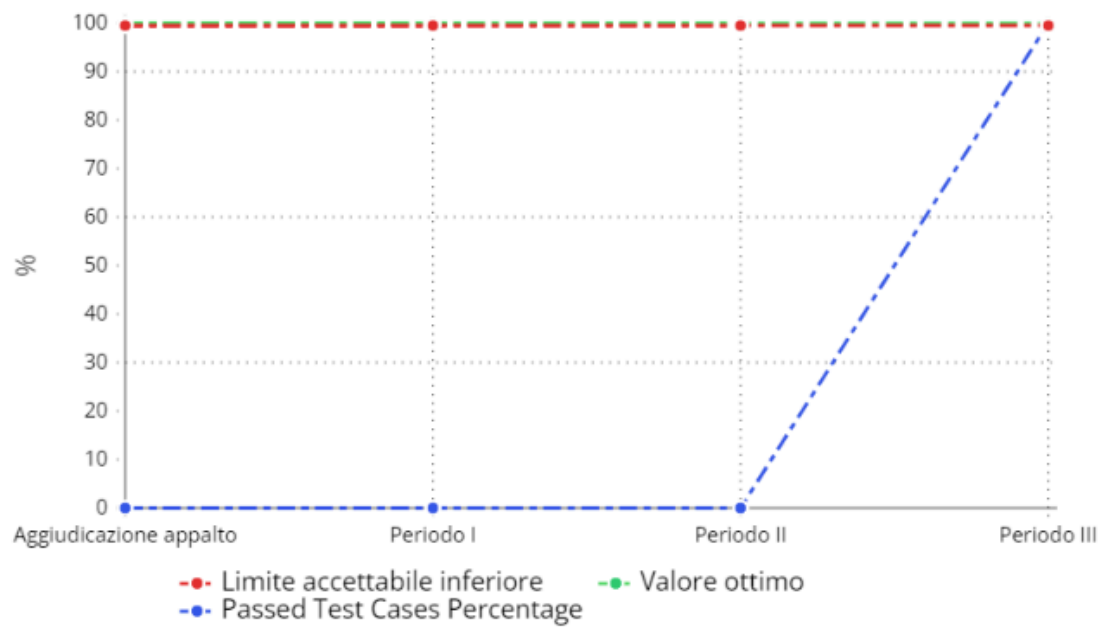
4.7) MPC12 - Correttezza Ortografica



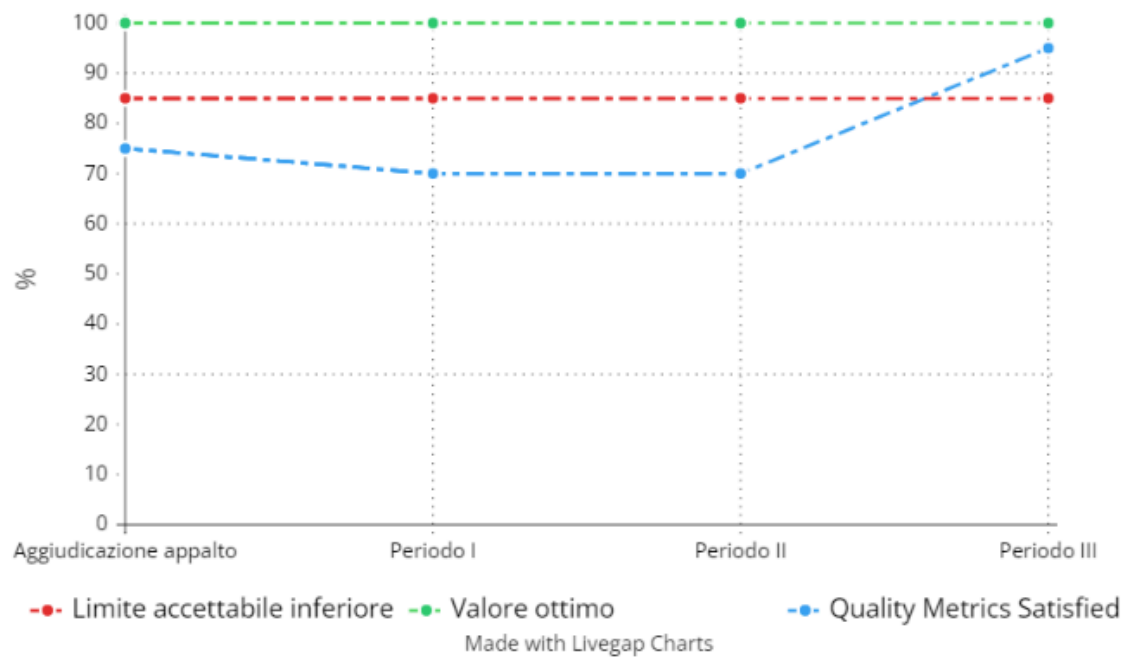
4.8) MPC13 - Code Coverage



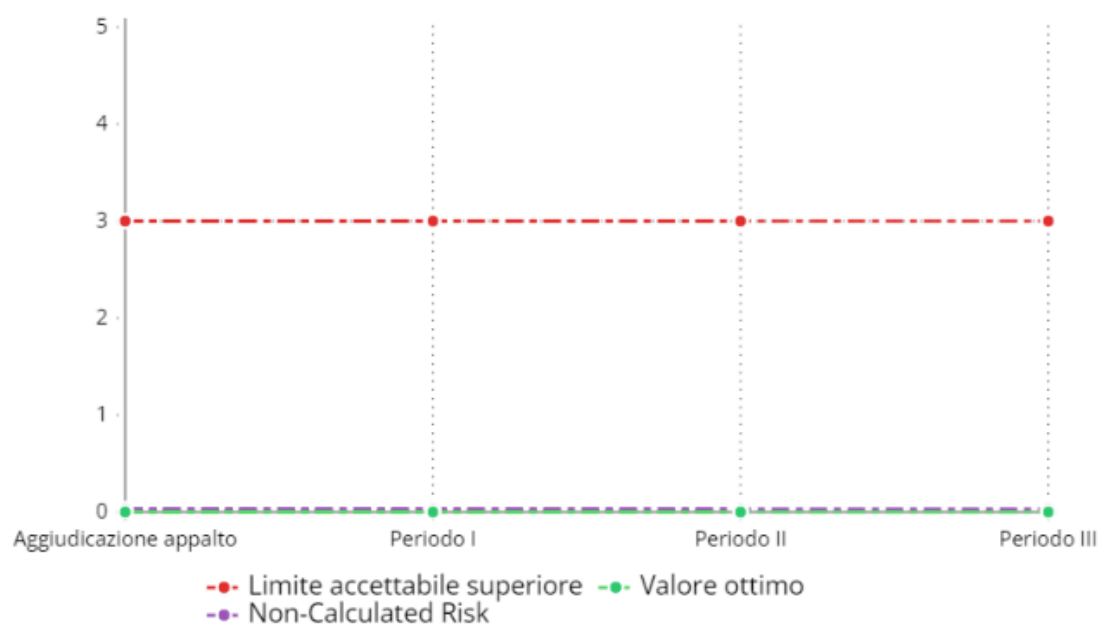
4.9) MPC14 - Passed Test Cases Percentage



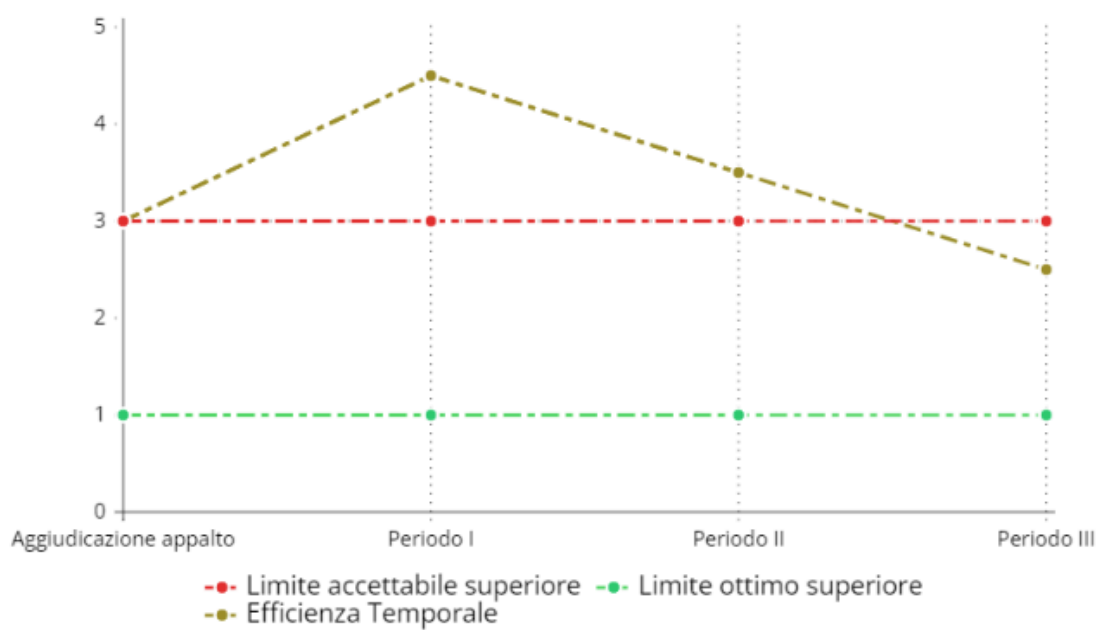
4.10) MPC15 - Quality Metrics Satisfied



4.11) MPC16 - Non-Calculated Risk



4.12) MPC17 - Efficienza Temporale



5) Iniziative di automiglioramento per la qualità

5.1) Introduzione

Un team di progetto è in grado di completare i compiti in modo efficiente a mantenendo un grado elevato di qualità, riducendo i costi e aumentando i profitti solo se mette la produttività in cima alla scala dei valori.

Di seguito infatti ci dedichiamo al tema del miglioramento della produttività dei processi coinvolti nella realizzazione del prodotto richiesto dal Capitolato C8. Essendo questo il primo progetto realistico affrontato dai membri del gruppo, è probabile che si verifichino problemi di natura organizzativa interna, di adempimento efficace dei ruoli assegnati e di giusto utilizzo degli strumenti scelti.

In questa sezione quindi andremo ad elencare i problemi più importanti che abbiamo riscontrato e le relative contromisure prese per risolvere o contenere i rallentamenti causati.

5.2) Problemi legati all'organizzazione generale

Fase	Problema Rilevato	Contromisura
Intero Progetto	È stato particolarmente complicato effettuare delle riunioni per fare il punto della situazione sul progetto dove fossero presenti tutti i membri del gruppo sia in presenza (poiché non si trovano aule libere che permettano ai membri di riunirsi e non tutti i componenti del gruppo seguono con frequenza le lezioni) che online (ognuno ha impegni diversi e dunque alle riunioni manca sempre qualche componente).	Si è deciso dunque che per comodità tutte le riunioni devono essere svolte online (sulla piattaforma Discord) cosicché ogni componente può collegarsi in qualunque situazione (capitava spesso di membri in viaggio o sui mezzi pubblici). Qualora uno dei componenti del gruppo non potesse comunque essere presente il responsabile provvederà a fornirgli un riassunto vocale della discussione tramite il contatto Telegram.

5.3) Valutazione sui ruoli

Fase	Ruolo	Problema Rilevato	Contromisura
Intero Progetto	Verificatore	Dato che questo è il primo progetto in termini di complessità della documentazione che ogni membro del nostro gruppo affronta abbiamo notato (ed è comprensibile) che spesso i documenti pronti ad essere rilasciati nel repository pubblico avevano degli errori di vario genere (esempio: di formattazione del documento, errori semantici, etc.)	Avvertire i verificatori di questo fenomeno e non assegnare loro altri compiti durante lo svolgimento del processo di verifica in quanto il loro lavoro è molto oneroso (non si tratta di approvare e basta le nuove modifiche al repository ma di verificare, spesso in argomenti a loro sconosciuti, se ciò che è stato prodotto sia valido o meno).
Fase iniziale	Responsabile	Nelle fasi iniziali del progetto è capitato che le atti-	Il responsabile di turno deve ricontrollare l'assegna-

		<p>vità venissero assegnate senza valutare in modo corretto il loro costo in termini di tempo ai vari componenti del gruppo. Questo ha portato chiaramente ad un sovraccarico per alcuni ruoli ed eccessivo tempo libero per altri.</p>	<p>zione dei compiti per assicurarsi che siano equamente distribuiti tra i membri. In questo modo, si evitano rallentamenti dovuti a ridistribuzioni degli oneri di progetto.</p>
Analisi dei requisiti	Analista	<p>Per svolgere l'analisi dei requisiti abbiamo dovuto analizzare in modo approfondito il capitolato ma ci è risultato difficile comprendere fin dall'inizio quale fosse il suo vero scopo, in particolare capire quali fossero le parti che realmente il proponente si aspettava realizzassimo.</p> <p>Non potevamo lasciare nulla al caso perché un'imprecisione in questa fase sarebbe stata riflessa immediatamente nei diagrammi dei casi d'uso e se non fosse stata risolta in tempo si sarebbe riflessa anche sul PoC.</p>	<p>Fissare a calendario degli incontri periodici con l'azienda proponente per chiarire quegli aspetti del progetto che ancora non erano ben espressi sul capitolato e successivamente riservare degli incontri anche con il docente Cardin per discutere alcuni aspetti tecnici del progetto (ad esempio: dato un caso d'uso, come sarebbe giusto formalizzarlo nei diagrammi dei casi d'uso seguendo lo standard).</p>
Analisi dei requisiti PoC	Analista Programmatore	<p>Spesso durante diverse attività di progetto colui che doveva svolgere il proprio compito si sentiva afflitto da una miriade di dubbi con granularità fine su come procedere con il proprio lavoro (es. l'analista nella stesura dei casi d'uso oppure il programmatore durante la realizzazione del PoC).</p>	<p>Dopo aver parlato di questo con il proponente siamo giunti alla conclusione di inserire uno dei referenti del progetto nel nostro canale privato Discord cosicché da poter chiarire in maniera informale e immediata qualsiasi dubbio con granularità fine, lasciando i problemi più importanti ai meeting ufficiali con verbale esterno annesso.</p>

5.4) Valutazione sugli strumenti

Fase	Strumento	Problema Rilevato	Contromisura
Intero progetto	Typst	Trovare un metodo per cui tutti i membri del gruppo potessero redigere docu-	Siamo giunti alla conclusione di usare Typst, un linguaggio che da le stesse fun-

		<p>menti indipendentemente dal dispositivo in uso, dal sistema operativo e anche dalle proprie conoscenze di base in materia (LaTeX non era una tecnologia conosciuta da tutti i membri del gruppo).</p>	<p>zionalità di LaTeX, ma con una facilità maggiore e una documentazione più ricca per chi non lo conoscesse. Inoltre per redigere i documenti abbiamo a disposizione una repository web messa a disposizione dallo stesso typst che ci permette di scrivere i documenti in maniera collaborativa dove ognuno può vedere in diretta cosa sta svolgendo un altro membro del gruppo.</p>
Fase iniziale	Git, Github, Typst e altri	<p>Nelle fasi iniziali del progetto è capitato spesso che venisse introdotta una nuova tecnologia da dei membri del gruppo (es. scelta di git e github) che la sapevano utilizzare con destrezza (grazie al corso di laurea Metodi e Tecnologie per lo sviluppo software) e che avrebbe dovuto essere utilizzata in futuro da dei componenti che non avevano un nessuna conoscenza sulla materia in questione.</p>	<p>Piuttosto che creare disallineamenti il gruppo ha preso una pausa, dove il responsabile garantiva che tutti i membri del gruppo avessero una conoscenza delle tecnologie tale da lavorare in maniera ottimale. Questo spesso veniva raggiunto individuando all'interno del gruppo i soggetti più preparati su un argomento e chiedendo loro di realizzare dei video tutorial su una determinata tematica.</p>