

MyTalk

Software di comunicazione tra utenti senza
requisiti di installazione



clockworkTeam7@gmail.com

Piano di Qualifica

v 2.0

Informazioni sul documento

Nome documento	Piano di Qualifica
Versione documento	v 2.0
Data creazione	2012/12/11
Data ultima modifica	2013/01/23
Uso documento	Esterno
Redazione	Ceseracciu Marco Furlan Valentino
Verifica	Palmisano Maria Antonietta Zohouri Haghian Pardis
Approvazione	La Bruna Agostino
Lista distribuzione	gruppo <i>Clockwork</i> Zucchetti SPA Prof. Tullio Vardanega

Sommario

Questo documento si prefigge di regolamentare le operazioni di verifica e validazione del gruppo *Clockwork* nello svolgimento del progetto **MyTalk**.

Diario delle modifiche

Autore	Modifica	Data	Versione
La Bruna Agostino	Verifica complessiva e approvazione	2013/01/23	v 2.0
Zohouri Haghian Pardis	Verifica e correzioni grammaticali, sintattiche e strutturali	2013/01/23	v 1.10
Zohouri Haghian Pardis	Controllo appendice	2013/01/23	v 1.9
Palmisano Maria Antonietta	Verifica contenutistica documento (senza appendice)	2013/01/23	v 1.8
Ceseracciu Marco	Aggiunto nell'appendice il resoconto della fase di progettazione	2013/01/20	v 1.7
Ceseracciu Marco	Completata stesura sezione Test di Integrazione	2013/01/20	v 1.6
Ceseracciu Marco	Inizio stesura sezione Test di Integrazione	2013/01/19	v 1.5
Furlan Valentino	Aggiunta sezione dell'esito della Revisione dei Requisiti, stesura sezione Test di Sistema	2013/01/19	v 1.4
Ceseracciu Marco	Sistemata Appendice e modificati i Riferimenti Informativi	2013/01/18	v 1.3
Furlan Valentino	Modificati contenuti della sezione Qualità	2013/01/18	v 1.2
Furlan Valentino	Modificati contenuti della sezione Visione generale delle strategie di verifica	2013/01/17	v 1.1
Zohouri Haghian Pardis	Approvazione del documento	2012/12/18	v 1.0
Furlan Valentino	Verifica e correzioni grammaticali e sintattiche	2012/12/18	v 0.11
Palmisano Maria Antonietta	Stesura Resoconto attività di verifica	2012/12/17	v 0.10
Palmisano Maria Antonietta	Stesura sezione Procedure di controllo di qualità di processo	2012/12/17	v 0.9
Palmisano Maria Antonietta	Stesura sezione Comunicazione e risoluzione di anomalie, stesura sezione Trattamento delle discrepanze	2012/12/16	v 0.8
Palmisano Maria Antonietta	Completata stesura sezione Strumenti e Tecniche, stesura sezione Misure e Metriche	2012/12/16	v 0.7
Palmisano Maria Antonietta	Stesura preliminare sezione Strumenti e Tecniche	2012/12/14	v 0.6
Palmisano Maria Antonietta	Stesura sezione Qualità	2012/12/13	v 0.5

Palmisano Maria Antonietta	Completata stesura sezione Risorse necessarie e Risorse disponibili	2012/12/13	v 0.4
Palmisano Maria Antonietta	Stesura sezione Responsabilità, stesura preliminare sezione Risorse necessarie e Risorse disponibili	2012/12/12	v 0.3
Palmisano Maria Antonietta	Stesura sezione Organizzazione, stesura sezione Pianificazione strategica generale	2012/12/11	v 0.2
Palmisano Maria Antonietta	Creazione documento, stesura sezione Introduzione	2012/12/11	v 0.1

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Visione generale delle strategie di verifica	3
2.1	Organizzazione e responsabilità	3
2.2	Pianificazione Strategica e Temporale	3
2.2.1	Verifica generale dei documenti	4
2.2.2	Fase di Analisi	4
2.2.3	Fase di Progettazione	6
2.2.4	Fase di Codifica	6
2.2.5	Fase di Collaudo	7
2.3	Risorse necessarie e risorse disponibili	8
2.4	Strumenti	8
2.5	Tecniche	8
2.5.1	Analisi statica	8
2.5.2	Analisi dinamica	9
2.5.3	Procedure dei test	9
2.6	Metodi	10
2.7	Misure e metriche	11
3	Obiettivi di qualità	12
3.1	Qualità di processo	12
3.2	Qualità del prodotto	14
3.2.1	Funzionalità	15
3.2.2	Affidabilità	15
3.2.3	Efficienza	16
3.2.4	Usabilità	16
3.2.5	Manutenibilità	17
3.2.6	Portabilità	17
4	Gestione amministrativa della revisione	18
4.1	Comunicazione e risoluzione di anomalie	18
4.2	Trattamento delle discrepanze	18
4.3	Procedure di controllo di qualità di processo	18
5	Pianificazione ed esecuzione del collaudo	20
5.1	Test di Sistema	20
5.2	Test di Integrazione	23
5.2.1	Test intra-componenti	23

5.2.2	Test inter-componenti	25
6	Dettaglio dell'esito delle revisioni	26
6.1	Revisione dei Requisiti	26
A	Resoconto delle attività di verifica	28
A.1	Revisione dei Requisiti	28
A.2	Revisione di Progettazione	28

Elenco delle figure

1	Modello SPA-I	12
---	-------------------------	----

Elenco delle tabelle

1	Mappamento dei requisiti ambito utente ai test di sistema	23
2	Funzionalità da verificare per ogni componente	25

1 Introduzione

1.1 Scopo del documento

Questo documento si prefigge di illustrare la strategia complessiva di verifica e validazione del gruppo *Clockwork* per la garanzia della qualità del progetto **MyTalk** e dei processi attuati per la sua produzione. Al fine di evidenziare e correggere anomalie, difetti e incongruenze, verranno presentati i vari criteri di qualità adottati, le metriche correlate e le conseguenti soglie che permetteranno di discriminare l'effettiva qualità.

1.2 Scopo del prodotto

Il prodotto denominato **MyTalk** si propone di fornire un software per un sistema di comunicazione audio e video tra utenti. Lo scopo del progetto è poter comunicare con altri utenti tramite il browser, utilizzando solo componenti standard, senza dover installare plugin o programmi esterni. L'utilizzatore dovrà poter chiamare un altro utente, iniziare la comunicazione sia audio che video, svolgere la chiamata e terminare la chiamata ottenendo delle statistiche sull'attività.

1.3 Glossario

Per evitare ambiguità i termini tecnici o di uso non comune, vengono evidenziati, alla loro prima occorrenza nel documento, tramite sottolineatura. Le definizioni di questi termini sono riportate nel documento in allegato **Glossario_v2.0.pdf**.

1.4 Riferimenti

1.4.1 Normativi

- Capitolato d'Appalto: **MyTalk**, rilasciato da Zucchetti SPA, reperibile all'indirizzo: <http://www.math.unipd.it/~tullio/IS-1/2012/Progetto/C1.pdf>
- Norme di Progetto (allegato **Norme_di_Progetto_v2.0.pdf**)

1.4.2 Informativi

- Analisi dei Requisiti (allegato **Analisi_dei_Requisiti_v2.0.pdf**)
- Specifica Tecnica (allegato **Specifica_Tecnica_v1.0.pdf**)
- Verbale incontro proponente 2012/12/18 (allegato **Verbale20121218.pdf**)
- Software Engineering - Part 5: Verification and Validation, Part 6: Management - Ian Sommerville - 8th ed. (2006)
- ISO/IEC 12207 - Standard per l'organizzazione dei processi

- ISO/IEC 15504:2007 - Standard per la valutazione della qualità dei processi
- ISO/IEC 9126:2001 - Standard per la valutazione della qualità del software
- IEEE 830-1998 - Recommended Practice per la Specifica dei Requisiti Software

2 Visione generale delle strategie di verifica

2.1 Organizzazione e responsabilità

Al fine di sviluppare un prodotto di qualità il gruppo *Clockwork* si impegna ad applicare il processo di verifica e validazione in ogni fase del progetto, sia sul prodotto software che sui documenti del progetto. All'interno del gruppo i ruoli destinati all'attuazione del processo di verifica e di validazione sono i Verificatori e il Responsabile:

- **Verificatore:** ha il compito di applicare le tecniche di verifica descritte successivamente per ogni fase del ciclo di vita. Il Verificatore deve accertare che la documentazione prodotta sia corretta, a livello grammaticale, sintattico e semantico e che rispetti le norme descritte nelle Norme di Progetto. Inoltre deve effettuare un controllo concettuale del documento, se vengono rilevate irregolarità queste verranno segnalate tramite un apposito ticket al redattore. Alla conclusione di tale processo il Verificatore dovrà segnalarlo al Responsabile per l'approvazione finale
- **Responsabile:** ha il dovere di coordinare e gestire l'attuazione del processo di validazione e verifica. È il solo a poter approvare o meno la correttezza di un documento

Al fine di garantire un costante riscontro dell'evoluzione del lavoro, ogni singolo membro del gruppo è tenuto a cronometrare le proprie ore di lavoro dedicate al progetto, per ogni ruolo che sta adempiendo, e renderlo noto al Responsabile, per effettuare un controllo sulle ore di lavoro preventivate e di conseguenza agire per la pianificazione della fase successiva. Inoltre tutti i membri del gruppo hanno l'obbligo di controllare quotidianamente l'emanazione dei ticket.

I documenti prodotti devono essere forniti di un diario delle modifiche, questo consentirà di agevolare la verifica sugli stessi. La conformità alle norme definite nel documento Norme di Progetto consentirà di ridurre gli errori dovuti a incomprensioni che possono nascere durante lo sviluppo del progetto.

Per evitare la propagazione incontrollata degli errori si utilizzerà il **Broken Window Theory** secondo il quale non appena viene riscontrato un errore, sarà segnalato e corretto il prima possibile.

2.2 Pianificazione Strategica e Temporale

Una verifica continua sui processi e sui prodotti, attuata in modo preventivo e non retrospettivo, può garantire un alto livello di qualità al fine di rispettare le aspettative del cliente e il pieno soddisfacimento dei requisiti richiesti dal Capitolato d'Appalto, minimizzando i rischi di fallimento prematuro del progetto e riducendo i costi di correzione degli eventuali errori.

Il Responsabile di Progetto si impegna a definire le attività di verifica finalizzate al collaudo di sistema, e le relative scadenze, assegnando tali attività ai verificatori in modo che ciascuno dei singoli elementi software, documentali o di altro

genere, vengano verificati durante i processi realizzativi ed entro le scadenze previste.

I processi di revisione adottati saranno di due tipi:

- Revisioni formali condotte dal cliente e con effetto sanzionatorio (corrispondenti all'Audit Process di ISO/IEC 12207)
 - Revisione dei Requisiti (RR): 2013/01/09
 - Revisione di Accettazione (RA): da destinarsi, ipotizzata al 2013/03/21
- Revisioni informali di revisione interna (di progresso) con il coinvolgimento del cliente (corrispondenti al Joint Review Process di ISO/IEC 12207)
 - Revisione di Progettazione (RP): 2013/02/05
 - Revisione di Qualifica (RQ): 2013/03/05

Come indicato nel Piano di Progetto il modello di ciclo di vita adottato per lo sviluppo del software è di tipo incrementale. In base a questo modello di ciclo di vita vengono definite le strategie di verifica per ogni fase del ciclo di vita.

2.2.1 Verifica generale dei documenti

Indipendentemente dalla fase che si sta svolgendo, la verifica dei documenti ha la stessa base, e il Verificatore deve controllare che:

- Siano conformi alle norme descritte nel documento `Norme_di_Progetto_v2.0.pdf`
- I termini ambigui siano sottolineati alla loro prima occorrenza e inseriti nel Glossario
- Un documento sia sintatticamente, semanticamente e ortograficamente corretto
- La struttura e i contenuti devono essere conformi agli standard a cui si fa riferimento
- I riferimenti e le etichette riguardanti figure, tabelle e sezioni siano puntuali e corretti

2.2.2 Fase di Analisi

In questa fase i Verificatori hanno il compito di controllare che il documento riguardante l'Analisi dei Requisiti, oltre a soddisfare gli obiettivi qualitativi descritti nella sezione 2.2.1, deve rispettare le seguenti caratteristiche¹:

- **Privo di ambiguità:** il documento non è ambiguo se e solo se ogni requisito ha solo una interpretazione. Il Verificatore per controllare questa qualità dovrà procedere alla lettura del singolo requisito, non è un processo automatizzabile

¹Per una più precisa descrizione delle qualità elencate si rimanda a IEEE 830-1998.

- **Corretto:** il documento è corretto se e solo se un requisito rilevato è un requisito che il software deve avere. Il Verificatore per controllare questa qualità deve avere una buona conoscenza dei documenti forniti dal proponente, inoltre questo controllo sarà agevolato da un corretto tracciamento dei requisiti
- **Completo:** il documento è completo se e solo se include tutti i requisiti fondamentali e definisce come reagisce il software agli input. Il Verificatore dovrà quindi procedere a una lettura attenta del documento
- **Verificabile:** il documento è verificabile se e solo se per ogni suo requisito è definito il modo con cui si possa verificare che il software soddisfi tale requisito, ovvero che sia tracciabile e atomico, in quanto i requisiti non atomici sono difficilmente verificabili. Questa attività non è automatizzabile, pertanto si richiede una attenta lettura
- **Consistente:** il documento è consistente se non genera contraddizioni al suo interno e con documenti di più alto livello. Si richiede una attenta lettura da parte del Verificatore dell'intero documento
- **Modificabile:** il documento è modificabile se la sua struttura consente di effettuare delle modifiche facilmente e in modo consistente. La conformità delle norme facilita il raggiungimento di questa qualità
- **Tracciabile:** il documento è tracciabile se ogni requisito ha una fonte chiara e se il documento definisce il modo con cui riferirsi al requisito. Il Verificatore, nella tabella riassuntiva dei requisiti, dovrà controllare che per ogni requisito sia stata definita la sua provenienza

Per la verifica del documento principalmente sarà usata la tecnica walkthrough, inoltre, se possibile, verrà utilizzata la tecnica inspection per individuare errori, o non conformità, già riscontrati dal committente nelle versioni precedenti del documento. Le tipologie di tecniche sono descritte nella sezione 2.5. Inoltre i Verificatori rispetto ai diagrammi UML dovranno accertare che essi siano:

- Tracciabili
- Atomici
- Corretti ortograficamente e sintatticamente

Ci si aspetta che una buona progettazione e verifica in questa fase individui i requisiti necessari al raggiungimento delle seguenti qualità:

- **Sufficienza:** i portatori di interesse si aspettano che il software sia capace di soddisfare tutti i requisiti
- **Comprensibilità:** il software deve essere facilmente comprensibile ai portatori di interesse, qualità raggiunta se in questa fase saranno individuati completamente i requisiti, in particolare anche quelli mirati alla usabilità

2.2.3 Fase di Progettazione

In questa fase, oltre agli obiettivi qualitativi generali descritti nella sezione 2.2.1, si dovrà accertare che ogni requisito scoperto nella fase di Analisi venga correttamente tradotto nell'architettura del sistema software, descritta nella Specifica Tecnica.

In questa fase bisogna accertare che la progettazione effettuata rispetti i principi di:

- **Disaccoppiamento:** le classi definite devono essere il più possibile indipendenti
- **Packaging:** le classi devono essere raggruppate per criterio logico
- **Coesione:** le classi devono contenere solo il necessario per raggiungere gli obiettivi

Tale attività sarà agevolata da un corretto tracciamento dei requisiti. Una buona progettazione e verifica in questa fase porterà al raggiungimento delle seguenti qualità:

- **Modularità:** poiché è richiesto che il software sia migliorabile deve essere diviso in parti chiare e distinte. I Verificatori dovranno quindi verificare il livello di disaccoppiamento e di incapsulamento delle classi
- **Semplicità:** l'architettura del software deve contenere solo il necessario, niente di superfluo

Dato che la verifica in questa fase non è automatizzabile, i Verificatori porranno la massima attenzione in questa fase, in quanto è di fondamentale importanza per il successo della fase successiva.

2.2.4 Fase di Codifica

In questa fase, oltre al soddisfacimento degli obiettivi qualitativi descritti nella sezione 2.2.1, si dovranno esaminare le unità di codice prodotte per accertarsi che non contengano errori di programmazione. Si deve controllare che ogni requisito sia tradotto in unità di codice e che non ci sia codice non associato ad alcun requisito².

Poiché il modello di ciclo di vita deciso è quello *incrementale* ([Piano_di_Progetto_v2.0.pdf](#), sez. Ciclo di vita), si avranno degli incrementi i quali hanno lo scopo di sviluppare unità software che verranno successivamente integrate tra loro. La priorità di sviluppo delle varie unità sarà determinata da una visione “bottom-up”, ovvero si seguirà questa procedura:

1. Prodotte principalmente le unità di codice con livello minore di dipendenza dalle altre, per ridurre la produzione di driver per l'attività di test

²Questo sarà possibile tramite un buon tracciamento dei requisiti.

2. Ogni unità dovrà compilare correttamente³
3. Ogni unità deve essere sottoposta ad analisi statica (vedasi sezione 2.5) mediante i software che si è deciso di adottare (*Norme_di_Progetto_v2.0.pdf*, sez. Ambiente di verifica e validazione)
4. Se richiesta l'esecuzione di prova, si effettueranno i test funzionali (black-box)
5. Se richiesto, si effettueranno test strutturali (white-box) (*Norme_di_Progetto_v2.0.pdf*, sez Analisi dinamica), supportati dagli strumenti che si è deciso di adottare (*Norme_di_Progetto_v2.0.pdf*, sez Ambiente di verifica e validazione)

Nel caso si riscontrino malfunzionamenti dopo una integrazione sarà necessario:

- Individuare l'unità responsabile del malfunzionamento
- Ripristinare l'ultima versione stabile del sistema, avvisando il Programmatore che ha introdotto il problema e il Responsabile
- Se il compilatore non genera errori dopo l'integrazione di una nuova unità, si effettueranno test di integrazione descritti nella sezione 5.2

Ci si aspetta che, dall'attuazione di una buona codifica, associata a un buon metodo di verifica, il software raggiunga le seguenti qualità:

- **Robustezza:** poiché il software è destinato anche a persone con limitate conoscenze informatiche, deve essere capace di sopportare ingressi errati. Mediante i test, si dovrà verificare che il software resti stabile anche con input errati
- **Efficienza:** per dare un riscontro veloce con l'utente, per migliorarne l'usabilità, il software deve essere efficiente in termini di tempo. I Verificatori valuteranno questa qualità tramite l'uso di test automatizzati

2.2.5 Fase di Collaudo

In questa fase, oltre a dover soddisfare gli obiettivi qualitativi generali definiti nella sezione 2.2.1, si effettueranno dei test di accettazione sui requisiti definiti nella sezione 5, e ci si assicurerà che non si presentino malfunzionamenti durante l'utilizzo del prodotto.

La strategia di verifica adottata è descritta nella sezione 2.5.

Verranno testati:

- Il corretto funzionamento del programma
- La facilità da parte dell'utente di compiere le operazioni per cui il software si è posto il compito

³Le unità in questione non dovranno provocare warning.

Tramite i giudizi degli utenti valuteremo l'usabilità del prodotto **MyTalk**. Per il soddisfacimento dei requisiti di qualità definiti nel documento *Analisi dei Requisiti*, si adotteranno tecniche di walkthrough e inspection. Per il soddisfacimento dei vincoli sarà effettuata una prova ad hoc. Al termine della fase di beta-test si passerà alla *Revisione di Accettazione*. La buona riuscita di questa fase porterà al raggiungimento della seguente qualità:

- **Affidabilità:** i portatori di interesse si aspettano che il software funzioni correttamente. I Verificatori valuteranno questa caratteristica eseguendo personalmente le funzionalità del prodotto

2.3 Risorse necessarie e risorse disponibili

Tali risorse serviranno per la realizzazione del progetto, esse sono distinte in risorse umane e risorse tecnologiche. Le risorse umane sono costituite dalle figure di Responsabile, Responsabile standard, Amministratore, Analista, Progettista, Verificatore, Programmatore (*Norme_di_Progetto_v2.0.pdf*, sez. Ruoli). Le risorse tecnologiche che serviranno per lo sviluppo del progetto sono i pc dei singoli componenti del gruppo, dotati della strumentazione software (*Norme_di_Progetto_v2.0.pdf*, sez. Ambiente di verifica e validazione).

2.4 Strumenti

I software che si utilizzeranno per facilitare e automatizzare l'attività di Verifica e Validazione sono descritti nel documento *Norme_di_Progetto_v2.0.pdf*, sez. Ambiente di verifica e validazione.

2.5 Tecniche

I Verificatori dovranno adottare le seguenti tecniche di analisi dove è richiesto:

- Analisi statica (vedasi 2.5.1)
- Analisi dinamica (vedasi 2.5.2)

2.5.1 Analisi statica

Tipologia di analisi che non richiede l'esecuzione del codice e si può applicare ad ogni prodotto, non solo al codice.

Per la verifica di prodotti semplici si può ricorrere ai metodi di lettura (desk check), distinti in:

- **Inspection:** lettura mirata all'individuazione di un errore presupposto
- **Walkthrough:** lettura a largo spettro per rilevamento di errori non presupposti

Per effettuare analisi statica sulle unità di codice si ricorre a metodi formali (*Norme_di_Progetto_v2.0.pdf*, sez. Analisi statica), che dimostrano una certa proprietà del prodotto.

Nelle prime fasi del progetto si adotterà principalmente la tecnica walkthrough. Gli errori rilevati più frequentemente, dai Verificatori e dal committente, forniranno al gruppo esperienza, per consentire successivamente, nelle attività di verifica, di utilizzare una tecnica di inspection per l'individuazione di tali errori.

2.5.2 Analisi dinamica

Tipologia di analisi che richiede l'esecuzione del codice, che si distingue in:

- Unità
- Aggregato
- Sistema completo

Le esecuzioni che applicano analisi dinamica sono denominate test o prove.

Le tecniche che si utilizzeranno per svolgere i test sono le seguenti:

- **Test di unità**
- **Test di integrazione**
- **Test di regressione**
- **Test di sistema e collaudo**

Per ulteriori chiarimenti riguardanti i test consultare *Norme_di_Progetto_v2.0.pdf*, sez. Analisi dinamica.

Si inizierà col effettuare test sulle singole unità del sistema, e, una volta superati tali test, saranno integrate per svolgere altri test che verificheranno le interazioni tra queste unità.

Infine, si eseguiranno una serie di test di accettazione per assicurare che il sistema funzioni nella maniera attesa.

Il processo di test del software ha due obiettivi:

- Dimostrare allo sviluppatore che il software soddisfa i requisiti
- Scoprire errori e/o difetti nel software⁴

2.5.3 Procedure dei test

I passi per effettuare test sono:

1. Definizione del test sul componente
2. Preparazione del caso di prova

⁴Ovvero dove il comportamento del software è errato, indesiderato o non conforme alle sue specifiche.

3. Esecuzione del test sul componente
4. Analisi dei risultati ottenuti con quelli attesi
5. Verifica dei risultati

Definizione del test sul componente

- Se è un componente attivo, e non sono state ancora prodotte le componenti passive pilotate da tale componente, il test dovrà definire gli stub necessari
- Se è un componente passivo e non sono state ancora prodotte le componenti attive che pilotano il componente soggetto a test, il test dovrà definire i driver necessari

Preparazione del caso di prova L'esecuzione del test deve avvenire in un definito ambiente, con dati in ingresso controllati, e devono essere definiti gli output attesi.

I dati in input devono essere distinti in 3 classi di equivalenza:

- Interni al dominio
- Al limite del dominio
- Al di fuori del dominio

Verifica dei risultati

- Se i risultati ottenuti differiscono da quelli attesi, si comunicheranno gli errori tramite i ticket, successivamente si correggeranno gli errori e si dovrà rieseguire il test
- Se i risultati ottenuti non differiscono da quelli attesi, si procederà con la registrazione dei risultati

2.6 Metodi

Poter garantire un software di qualità, capace quindi di soddisfare le proprietà richieste, è un obiettivo di fondamentale importanza, reso possibile allocando risorse per effettuare controllo sul prodotto.

Tuttavia le risorse sono limitate e comportano dei costi, e quindi è importante saperle gestire in modo da mantenere un buon rapporto fra qualità del prodotto e risorse necessarie.

Pertanto, in base alla complessità del prodotto, e alle caratteristiche che esso deve avere, il gruppo *Clockwork* adotterà le seguenti tipologie di analisi statica:

- **Analisi di flusso di controllo**
- **Analisi di flusso dei dati**

- **Analisi di flusso di informazione**
- **Verifica formale del codice**

Tali metodologie vengono definite nel documento `Norme_di_Progetto_v2.0.pdf`, sez. Analisi statica.

2.7 Misure e metriche

Le misure rilevate dal processo di verifica, che permettono di ottenere informazioni quantitative relative ai prodotti di ogni processo produttivo (sia esso progettuale o di codifica), devono essere stabilite a priori quando si hanno sufficienti conoscenze dell'argomento.

Qualora ci fossero metriche, ancora incerte e approssimative, si dovrà cercare di stabilirle il più precisamente possibile durante il corso della progettazione e realizzazione del prodotto, attraverso l'analisi di ciò che sarà stato realizzato fino a quel momento. Le tipologie di metriche che adotteremo sono:

- Complessità ciclomatica
- Numeri di parametri
- Numero di campi dati per classe
- Numero di livelli di annidamento
- Grado di accoppiamento
 - Indice di utilità
 - Indice di dipendenza

Tali metriche, con corrispettive misurazioni, vengono definite nel documento `Norme_di_Progetto_v2.0.pdf`, sez. Metriche.

3 Obiettivi di qualità

3.1 Qualità di processo

La qualità dei processi che determinano la realizzazione del prodotto è necessaria per garantire la qualità del prodotto finale.

Si adotta lo standard ISO/IEC 15504:2007 SPICE “Software Process Improvement Capability Determination” per garantire la qualità dei processi utilizzati. Esso definisce un modello per la valutazione della qualità dei processi nello sviluppo software SPA-I “Software Process Assessment & Improvement”. Tutti i processi saranno quindi valutati per misurarne la qualità e permetterne il miglioramento.



Figura 1: Modello SPA-I

Per la misurazione della maturità del processo lo SPICE definisce i seguenti attributi di processo:

1. **Process performance:** capacità di un processo di raggiungere i suoi obiettivi, trasformando input identificabili in output identificabili
2. **Process management:** insieme di attività di pianificazione e monitoraggio delle performance di un processo

3. **Work product management:** lo sviluppo del processo è correttamente pianificato e monitorato per consentire la corretta verifica e documentazione dei risultati ottenuti
4. **Process definition:** l'attuazione di un processo si basa su approcci standardizzati
5. **Process resource:** il processo ha a disposizione sufficienti risorse umane e tecnologiche per poter essere attuato
6. **Process deployment:** i risultati ottenuti sono misurabili al fine di determinare il raggiungimento degli obiettivi desiderati
7. **Process control:** vengono raccolte ed analizzate delle misurazioni allo scopo di poter applicare le correzioni necessarie alla corretta attuazione del processo
8. **Process innovation:** riguarda l'attenzione al miglioramento e alla realizzazione degli obiettivi attuali e futuri pianificati dall'azienda. I cambiamenti proposti ai processi esistenti devono essere valutati rispetto agli obiettivi di business attuali e futuri al fine di selezionare le modifiche più appropriate da implementare
9. **Process optimization:** le modifiche al processo vengono definite ed implementate in modo da garantire il continuo incremento delle performance per il raggiungimento degli obiettivi prefissati

Per ciascuno dei seguenti attributi viene assegnata una valutazione che ne definisce il livello di possesso dell'attributo da parte del processo. Lo standard definisce le possibili valutazioni assegnabili:

- **N - Non posseduto** (0 - 15% di possesso): non c'è evidenza oppure ce n'è poca del possesso di un attributo
- **P - Parzialmente posseduto** (16 - 50% di possesso): c'è evidenza di approccio sistematico al raggiungimento del possesso di un attributo e del raggiungimento di tale possesso, ma alcuni aspetti del possesso possono essere non prevedibili
- **L - Largamente posseduto** (51 - 85% di possesso): vi è evidenza di approccio sistematico al raggiungimento del possesso di un attributo e di un significativo livello di possesso di tale attributo, ma l'attuazione del processo può variare nelle diverse unità operative della organizzazione
- **F - (Fully) Pienamente posseduto** (86 - 100% di possesso): vi è evidenza di un totale e sistematico approccio e di un completo raggiungimento del possesso dell'attributo e non esistono significative differenze nel modo di attuare il processo tra le diverse unità operative

I livelli di maturità che SPICE prevede per la valutazione sono:

- **Livello 0 - Processo incompleto:** il processo non è implementato o non raggiunge gli obiettivi. Non vi è evidenza di approcci sistematici agli attributi definiti
- **Livello 1 - Processo semplicemente attuato:** il processo viene messo in atto e raggiunge i suoi obiettivi. Non vi è evidenza di approcci sistematici agli attributi definiti. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process performance”
- **Livello 2 - Processo gestito:** il processo è attuato, ma anche pianificato, tracciato, verificato ed aggiustato se necessario, sulla base di obiettivi ben definiti. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process management” e “Work product management”
- **Livello 3 - Processo definito:** il processo è attuato, pianificato e controllato sulla base di procedure ben definite, basate sui principi del software engineering. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process definition” e “Process resource”
- **Livello 4 - Processo predicibile:** il processo è stabilizzato ed è attuato all'interno di definiti limiti riguardo i risultati attesi, le performance, le risorse impiegate ecc. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process deployment” e “Process control”
- **Livello 5 - Processo ottimizzante:** il processo è predicibile ed in grado di adattarsi per raggiungere obiettivi specifici e rilevanti per l'organizzazione. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process innovation” e “Process optimization”

L'applicazione dello standard ISO/IEC 15504 porta benefici agli sviluppatori, agli utilizzatori o acquirenti del software. In particolare:

- **Sviluppatori:** vantaggi nell'ottimizzazione dell'uso delle risorse, che comporta un contenimento dei costi, maggiore tempestività di consegna del prodotto ultimato, migliore stima dei rischi ed impegno e la possibilità di confrontarsi con delle best practice
- **Utenti:** maggiore facilità nella selezione dei fornitori, migliore valutazione dei rischi di progetto e controllo dello stato di avanzamento in corso d'opera, una riduzione dei costi di correzione degli errori ed un controllo dei rischi e delle varianti in corso d'opera

3.2 Qualità del prodotto

Riguardo alla qualità del prodotto si seguiranno le linee guida dettate dallo standard ISO/IEC 9126, che è composto da varie caratteristiche e sotto-caratteristiche.

Successivamente per ogni caratteristica generale sarà specificato quale sarà il grado minimo di raggiungimento per definire un determinato obiettivo di qualità conseguito.

Il prodotto si potrà considerare completo e di qualità solo quando tutte le caratteristiche saranno soddisfatte almeno ad un grado minimo.

3.2.1 Funzionalità

Il prodotto deve avere la capacità di offrire apposite funzionalità che siano in grado di esaudire le richieste dell'utente sotto condizioni specifiche. In particolare il prodotto dovrà essere appropriato e accurato.

- **Appropriatezza:** costituisce la capacità del software di offrire un insieme di funzioni per i compiti ed obiettivi prefissati all'utente
- **Accuratezza:** capacità del software di fornire i risultati concordati o i precisi effetti richiesti
- **Interoperabilità:** capacità del prodotto software di interagire ed operare con specifici sistemi
- **Conformità:** capacità del prodotto software di aderire a standard, convenzioni e regolamentazioni rilevanti al settore operativo a cui vengono applicate
- **Sicurezza:** la capacità del prodotto software di proteggere informazioni e dati impedendo che persone o sistemi non autorizzati possano accedervi o modificarli, mentre garantire queste operazioni a utenti o sistemi autorizzati

Quantificazione del raggiungimento dell'obiettivo di qualità: la misurazione del raggiungimento di questo obiettivo si calcolerà verificando la quantità di requisiti soddisfatti che avranno un riscontro in elementi funzionali nell'applicazione prodotta.

Il soddisfacimento di tutti i requisiti obbligatori previsti dal Capitolato d'Appalto rappresenterà la soglia di sufficienza.

3.2.2 Affidabilità

L'affidabilità misura la capacità del prodotto di mantenere un determinato livello di prestazioni se usato in determinate condizioni e per un certo periodo.

- **Maturità:** capacità del software di evitare che si verifichino errori, malfunzionamenti o la produzione di risultati non coerenti
- **Tolleranza agli errori:** il software deve riuscire a mantenere determinati livelli di prestazioni, nonostante la presenza di errori, malfunzionamenti o usi scorretti

- **Recuperabilità:** capacità di un prodotto di ripristinare il livello appropriato di prestazioni e di essere in grado di recuperare le informazioni rilevanti in seguito ad un malfunzionamento
- **Aderenza:** capacità di aderire a standard, regole e convenzioni inerenti l'affidabilità

Quantificazione del raggiungimento dell'obiettivo di qualità: il confronto fra il numero di esecuzioni totale con quelle andate a buon fine e che hanno mantenuto un livello di prestazione da poter permettere l'utilizzo previsto del prodotto, sarà la misurazione del raggiungimento di questo obiettivo.

3.2.3 Efficienza

L'efficienza è determinata dalla capacità di fornire adeguate prestazioni in relazione alla quantità di risorse usate.

- **Comportamento rispetto al tempo:** capacità di fornire adeguati tempi di risposta, elaborazione e velocità, sotto condizioni determinate
- **Utilizzo delle risorse:** capacità di utilizzare in maniera adeguata la giusta quantità e tipologia di risorse
- **Conformità:** capacità di aderire a standard e specifiche sull'efficienza

Quantificazione del raggiungimento dell'obiettivo di qualità: il tempo necessario per ottenere una risposta dal servizio (sia in condizioni normali che in condizioni di sovraccarico) determinerà la misurazione del raggiungimento di questo obiettivo.

3.2.4 Usabilità

L'usabilità è determinata dalla capacità del software di essere capito, appreso e usato dall'utente in certe condizioni.

- **Comprensibilità:** rappresenta la facilità di comprensione dei concetti del prodotto, permettendo all'utente di comprendere se il software è appropriato
- **Apprendibilità:** capacità di diminuire l'impegno richiesto agli utenti per imparare ad utilizzare l'applicazione
- **Operabilità:** capacità di porre gli utenti in condizioni tali da utilizzare il prodotto per i propri scopi e controllarne l'uso
- **Conformità:** capacità del software di aderire a standard o convenzioni relativi all'usabilità

Quantificazione del raggiungimento dell'obiettivo di qualità: la capacità del prodotto di adattarsi ai vari tipi di ambienti in cui esso verrà eseguito, costituirà la misurazione del raggiungimento di questo obiettivo. L'usabilità sarà poi ritenuta raggiunta fornendo un'interfaccia il più possibile chiara, semplice ed intuitiva.

3.2.5 Manutenibilità

La manutenibilità rappresenta la capacità del software di essere modificato, includendo correzioni, miglioramenti e/o adattamenti.

- **Facilità di analisi:** rappresenta la facilità con la quale è possibile analizzare il codice per localizzare un eventuale errore
- **Modificabilità:** capacità del prodotto software di permettere l'implementazione di una specifica modifica
- **Stabilità:** capacità del software di evitare effetti inaspettati derivanti da modifiche errate
- **Testabilità:** capacità del software di essere facilmente testato per validare le modifiche apportate

Quantificazione del raggiungimento dell'obiettivo di qualità: la misurazione del raggiungimento di questo obiettivo sarà legata al rispetto delle misure metriche descritte nella sezione 2.2.

3.2.6 Portabilità

La portabilità è la capacità di un software d'essere portato da un ambiente di lavoro ad un altro.

- **Adattabilità:** capacità del software di essere adattato per differenti ambienti operativi senza dover applicare modifiche diverse da quelle fornite per il software considerato
- **Conformità:** capacità del software di aderire a standard e convenzioni relative alla portabilità
- **Sostituibilità:** capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti

Quantificazione del raggiungimento dell'obiettivo di qualità: il sistema deve essere funzionante con il browser Google Chrome e andrà testato sugli altri principali browser (come Firefox, Opera, Safari, Internet Explorer).

4 Gestione amministrativa della revisione

4.1 Comunicazione e risoluzione di anomalie

Un'**anomalia** consiste in una deviazione del prodotto dalle aspettative prefissate. Per la gestione e risoluzione di anomalie ci si affida allo strumento di ticketing messo a disposizione dal servizio Github nell'apposita sezione "Issue". Il Verificatore, per ogni nuova anomalia riscontrata dovrà aprire un nuovo ticket (*Norme_di_Progetto_v2.0.pdf*, sez. Gestione delle attività).

4.2 Trattamento delle discrepanze

Una **discrepanza** è un tipo di anomalia non grave, non concerne il corretto funzionamento del prodotto, ma può riguardare un allontanamento dai requisiti attesi specificati nel Capitolato d'Appalto o una violazione delle Norme di Progetto. Le modalità di comunicazione delle discrepanze è del tutto simile alle modalità specificate per le anomalie; la risoluzione presenta invece modalità differenti. Una volta creato il ticket sarà compito del verificatore riconoscere quale tipo di discrepanza si tratta e, nel caso riguardi una violazione delle Norme di Progetto, provvederà a comunicare il problema all'Amministratore che prenderà provvedimenti. Se invece il problema riguarda un allontanamento dai requisiti, una volta identificata l'origine della discrepanza, il Verificatore solleciterà l'Analista per valutare la gravità e i costi per risolverla.

4.3 Procedure di controllo di qualità di processo

L'organizzazione interna dei processi si basa sul principio PDCA, in grado di garantire un miglioramento continuo della qualità di tutti i processi (compreso quello di verifica stesso) e, conseguentemente, dei prodotti risultanti. I processi devono essere pianificati dettagliatamente rispetto ai requisiti richiesti e alle risorse disponibili, e quindi attuati secondo il piano. La verifica sui processi avviene attraverso l'analisi costante delle misurazioni sul prodotto di ciascun processo e sul processo stesso. Se l'analisi di tali misure evidenzia valori che si discostano, in modo peggiorativo, dai piani prefissati, questo denoterà la presenza di un problema: per ognuno di essi che verrà identificato, si provvederà a stabilirne le cause e le possibili soluzioni, intervenendo in modo correttivo sul processo ed, eventualmente, sul piano iniziale ad esso relativo. Le misurazioni sul processo consistono principalmente in:

- Tempo impiegato per essere completato
- Cicli iterativi interni al processo
- Attinenza ai piani stabiliti
- Soddisfazione dei requisiti richiesti
- Risorse utilizzate e/o consumate durante il processo

Se non vengono identificati problemi relativi ad un processo, è possibile intervenire comunque in modo migliorativo: tale miglioramento consiste evidentemente nel ridurre il numero di cicli iterativi, risorse e tempo utilizzati, comunque garantendo che l'esecuzione del processo sia fedele al piano e soddisfi i requisiti, aumentandone quindi l'efficienza e determinando una variazione non negativa nel grado di efficacia.

5 Pianificazione ed esecuzione del collaudo

5.1 Test di Sistema

Questa sezione è dedicata alla pianificazione dei test di sistema che verranno attuati prima del rilascio del prodotto finito. Tramite questi test si vorrebbe dimostrare che il prodotto soddisferà le aspettative del cliente, fornendo esempi di test da attuare per confermare che i requisiti obbligatori emersi ed elencati nel documento Analisi dei Requisiti sono stati soddisfatti.

Codice requisito	Codice verifica	Modalità di verifica	Stato della verifica
RUFO 1.1	TS-RUFO 1.1a	Prova dinamica: viene verificato se a fronte dell'inserimento di una username valida e non ancora utilizzata da altri utenti il sistema risponde in maniera positiva	Pianificata
RUFO 1.1	TS-RUFO 1.1b	Prova dinamica: viene verificato se a fronte dell'inserimento di una username non valida e/o già utilizzata da altri utenti il sistema risponde in maniera negativa	Pianificata
RUFO 1.2	TS-RUFO 1.2a	Prova dinamica: viene verificato se a fronte dell'inserimento di una password conforme alle richieste il sistema risponde in maniera affermativa	Pianificata
RUFO 1.2	TS-RUFO 1.2b	Prova dinamica: viene verificato se a fronte dell'inserimento di una password non conforme alle richieste il sistema risponde in maniera negativa	Pianificata
RUFO 1.3	TS-RUFO 1.3a	Prova dinamica: viene verificato se a fronte del reinserimento della password conforme alla password scelta il sistema risponde in maniera affermativa	Pianificata

RUFO 1.3	TS-RUFO 1.3b	Prova dinamica: viene verificato se a fronte del reinserimento della password non conforme alla password scelta il sistema risponde in maniera negativa	Pianificata
RUFO 1.4	TS-RUFO 1.4	Prova dinamica: viene verificato se il sistema effettivamente risponde in maniera positiva all'inserimento del nome da parte dell'utente	Pianificata
RUFO 1.5	TS-RUFO 1.5	Prova dinamica: viene verificato se il sistema effettivamente risponde in maniera positiva all'inserimento del cognome da parte dell'utente	Pianificata
RUFO 2.1	TS-RUFO 2.1a	Prova dinamica: viene verificato se a fronte dell'inserimento di uno username valido (relativo ad un utente) il sistema risponde in maniera positiva	Pianificata
RUFO 2.1	TS-RUFO 2.1b	Prova dinamica: viene verificato se a fronte dell'inserimento di uno username non valido il sistema risponde in maniera negativa	Pianificata
RUFO 2.2	TS-RUFO 2.2a	Prova dinamica: viene verificato se a fronte dell'inserimento di una password valida (relativa ad un utente) il sistema risponde in maniera positiva	Pianificata
RUFO 2.2	TS-RUFO 2.2b	Prova dinamica: viene verificato se a fronte dell'inserimento di una password non valida il sistema risponde in maniera negativa	Pianificata

RUFO 5.1	TS-RUFO 5.1	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di visualizzare gli utenti attualmente collegati ad un server	Pianificata
RUFO 6.1	TS-RUFO 6.1	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di selezionare il destinatario conoscendone l'indirizzo IP	Pianificata
RUFO 6.2	TS-RUFO 6.2	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di chiamare un utente che si è registrato presso un server	Pianificata
RUFO 6.3	TS-RUFO 6.3	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di scegliere un destinatario da una lista di utenti	Pianificata
RUFO 6.4	TS-RUFO 6.4	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di effettuare chiamate audio verso un altro utente	Pianificata
RUFO 6.5	TS-RUFO 6.5	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di effettuare chiamate video verso un altro utente	Pianificata
RUFO 6.15	TS-RUFO 6.15	Prova dinamica: viene verificato che il sistema permetta effettivamente ad un utente di abbandonare la chiamata in qualsiasi momento	Pianificata

RUFO 8	TS-RUFO 8	Prova dinamica: viene verificato se effettivamente dopo che l'utente ha terminato la sessione il sistema non lo riconosce più come utente autenticato	Pianificata
--------	-----------	---	-------------

Tabella 1: Mappamento dei requisiti ambito utente ai test di sistema

5.2 Test di Integrazione

Questa sezione viene dedicata alla pianificazione dei test da utilizzare per i vari componenti definiti durante la progettazione di alto livello del sistema. Tramite queste prove si vuol verificare e quindi dimostrare la corretta interazione ed il corretto flusso dei dati all'interno del sistema.

5.2.1 Test intra-componenti

I componenti saranno testati e collaudati seguendo una strategia incrementale mediante il principio di bottom-up.

Saranno collaudati inizialmente i componenti con minori dipendenze funzionali, proseguendo successivamente con l'albero delle dipendenze.

Sarà necessario avere funzionanti i database (compresa la loro gestione), successivamente si proseguirà con la parte back-end e front-end.

Ogni componente sosterrà dei test, ai quali ci si potrà aspettare un esito positivo o negativo.

Componente	Funzionalità da verificare
CCLI1 (autenticazione)	Ogni volta che un utente tenta un'autenticazione i dati devono essere cercati all'interno del server
	Se i dati che l'utente fornisce sono presenti nel database il suo indirizzo IP deve essere memorizzato
	Ogni volta che un utente effettua un logout, l'indirizzo IP legato al suo account deve essere impostato a 0
CCLI2 (gestione dati account)	Ogni volta che viene tentata una registrazione deve essere verificato che lo username non sia già utilizzato da un altro account
	Ogni volta che viene effettuata una registrazione e i dati devono essere inseriti nel server

	Ogni volta che un utente modifica i dati del proprio account questi devono essere aggiornati nel server
CCLI3 (tutorial)	Ogni volta che viene visualizzata la lista dei tutorial i link devono essere reperiti dal server
	Ogni volta che viene visualizzato un tutorial, questo deve essere integrato nella pagina attraverso l'apposito servizio di Youtube
	Ogni volta che viene visualizzato un tutorial i link per passare al video precedente e successivo devono essere generati dinamicamente prendendo i dati presenti nel server
CCLI4 (visualizzazione lista utenti)	Ogni volta che l'utente visualizza la lista utenti, tale lista viene creata dinamicamente tramite i dati presenti nel server
CCLI5 (gestione notifiche)	Ogni volta che il server segnala una notifica questa deve essere visualizzata nel modo corretto a seconda della tipologia
CICLI6 (chiamata)	Ogni volta che un utente tenta di iniziare una chiamata, i dati necessari all'inizializzazione della chiamata devono essere inviati al server
CICLI7 (Messaggio differito)	Ogni volta che l'utente registra un messaggio differito, tale messaggio viene salvato nel server in attesa di essere notificato al destinatario
CICLI8 (chat)	Ogni volta che viene inviato un messaggio, deve essere notificato al destinatario
CICLI9 (invio file)	Ogni volta che viene inviato un file, deve essere notificato al destinatario
	Ogni volta che il destinatario accetta di ricevere un file, deve iniziare il trasferimento
CSER1 (autenticazione)	Ogni volta che un utente effettua l'autenticazione, il server deve mandare un messaggio ai client per aggiornare la lista degli utenti
	Ogni volta che un utente effettua un logout, il server deve mandare un messaggio ai client per aggiornare la lista degli utenti
CSER2 (gestione dati utenti)	Ogni volta che un utente si registra, il server deve mandare un messaggio al client dell'avvenuta registrazione
	Ogni volta che un utente modifica i propri dati, il server deve mandare un messaggio al client dell'avvenuto aggiornamento

CSER3 (inizializzazione chiamata)	Ogni volta che un client invia i dati per effettuare una chiamata, deve venire inviato un messaggio alla componente CSER6
	Se il destinatario accetta la chiamata, deve venire tentata la connessione tra i gli utenti
CSER4 (salva messaggio in differita)	La prima volta che il destinatario di un messaggio differito presente nel server si autentica, deve venire inviato un messaggio alla componente CSER6
	Se il destinatario accetta il messaggio differito, tale messaggio differito deve venire visualizzato
CSER5 (tutorial)	-
CSER6 (notification)	Ogni volta che un'altra componente del server richiede di inviare una notifica, questa deve essere inoltrata al corretto destinatario

Tabella 2: Funzionalità da verificare per ogni componente

5.2.2 Test inter-componenti

Mentre il collaudo prosegue verranno aggiunti test e prove sui componenti in maniera incrementale. Ad ogni collaudo verrà testato il corretto funzionamento inter-componenti, tenendo conto delle loro dipendenze.

6 Dettaglio dell'esito delle revisioni

Ad ogni revisione che il gruppo intende sostenere, il committente richiede la conduzione di una presentazione, successiva alla visione della documentazione, da parte del gruppo, al fine di poter verificare lo stato di avanzamento del lavoro. Successivamente, entro la settimana lavorativa, il committente stabilisce una valutazione generale e dettagliata per ciascun documento. Sarà quindi compito del gruppo apportare immediatamente le modifiche e le correzioni richieste, allineandosi alle aspettative del committente, in maniera tale da proseguire il lavoro con la certezza di non perpetuare eventuali errori.

6.1 Revisione dei Requisiti

Il lavoro svolto per la fase di Revisione dei Requisiti è stato valutato discretamente, dimostrando che il materiale prodotto era sufficientemente adeguato, ma nei documenti di Analisi dei Requisiti, Piano di Progetto, Piano di Qualifica e Norme di Progetto sono stati rilevati la maggior parte degli errori. Viene riportato in dettaglio quanto è stato effettuato dal gruppo per risolvere le problematiche riscontrate:

- **Analisi dei Requisiti:** dettagliato maggiormente l'intero documento e apportate in particolare le seguenti correzioni:
 - Riportata per esteso la sigla TCO
 - Aggiunti i riferimenti informativi
 - Specificate le necessità degli utenti a cui il prodotto è rivolto
 - Descritti in maggiore dettaglio i vincoli
 - Inserite didascalie più descrittive per le figure
 - Rivisti, modificati e dettagliati maggiormente i casi d'uso e le loro descrizioni
 - Eliminati, modificati e aggiunti alcuni requisiti
 - Sistemate le relazioni tra i casi d'uso
 - Riviste le pre-condizioni e le post-condizioni
 - Effettuato il tracciamento dei requisiti - casi d'uso
 - Rieffettuato il tracciamento dei casi d'uso - requisiti
- **Piano di Progetto:** sono state corrette una serie di imprecisioni di minore importanza e migliorata la presentazione dei dati di *analisi dei rischi*, *pianificazione* e *preventivo*. Inoltre le maggiori correzioni che sono state effettuate sono le seguenti:
 - Migliorato il capitolo riguardante il ciclo di vita
 - Spiegata meglio la fase di Verifica e Validazione

- Chiariti la tipologia dei documenti Specifica Tecnica (ST) e Definizione di Prodotto (DdP)
- Chiarito che le attività svolte fino al 2012/12/21 non sono a carico del committente
- **Piano di Qualifica:** in generale sono state ampliate le strategie, in particolare sono state apportate le seguenti modifiche:
 - Modificate e ampliate le sezioni *Organizzazione della strategia di verifica* e *Obiettivi di qualità*
 - Trasformate in strategie di verifica le sezioni segnalate come materia di NdP
 - Usato il comando `LyX` per creare l'appendice
- **Norme di Progetto:** è stata migliorata la struttura del documento e in particolare sono state apportate le seguenti correzioni:
 - Inseriti i riferimenti normativi e informativi
 - Corretto il formato della data
 - Regolamentato l'avanzamento di versione
 - Modificate le regole di verifica in caso di mancanza di verificatori
 - Definite regole per la rotazione dei ruoli
- **Glossario:** è stata sistemata l'inclusione del diario delle modifiche
- **Verbale 2012/12/18:** inserita sintesi dei punti salienti dell'incontro

A Resoconto delle attività di verifica

A.1 Revisione dei Requisiti

Nella fase di Revisione dei Requisiti è stata effettuata un'attività di verifica sui documenti prodotti. Nel dettaglio è stata effettuata un'analisi statica come indicato nella sezione 2.5.1 tramite walkthrough prima ed inspection poi, verificando che i documenti rispettino i punti individuati e riportati in tale sezione inerenti la formattazione del testo, effettuando le dovute procedure per la correzione degli errori rilevati. Nel dettaglio, i documenti sono stati revisionati effettuando quindi il controllo ortografico tramite Hunspell ed Enchant, plugin per LyX, mentre il controllo grammaticale, sintattico e lessicale è avvenuto tramite un'accurata rilettura da parte dei revisori. Sono stati poi controllati anche i contenuti tabellari e grafici. La segnalazione di irregolarità è avvenuta tramite Ticket che sono state prese in carico successivamente dal redattore e risolte. Infine, per effettuare il doppio tracciamento, si sono controllate le apposite tabelle come descritto nella sezione 2.2.

A.2 Revisione di Progettazione

Anche in questa fase, è stata effettuata analisi statica sui documenti in maniera analoga a quella della precedente revisione. Si è prestata maggiore attenzione ai diagrammi UML, in quanto sono stati riscontrati molti errori ed imprecisioni nella documentazione in ingresso alla precedente revisione. Nei documenti di Specifica Tecnica e Piano di Qualifica è poi stato fondamentale effettuare un'adeguata ed attenta fase di tracciamento. Come in precedenza sono stati infine utilizzati i Ticket per segnalare eventuali irregolarità ed errori che sono stati poi presi in carico dal redattore e risolti.