

Glossario

RAMtastic6

4 aprile 2024



email: ramtastic6@gmail.com

Informazioni sul documento

Versione: 0.2.0
Redattori: Zambon M. Brotto D.
Verificatori: Brotto D. Zaupa R.
Destinatari: T. Vardanega, R. Cardin, Imola Informatica
Uso: Esterno

Registro dei Cambiamenti - Changelog

Versione	Data	Autore	Verificatore	Dettaglio
v.0.2.0	2024-04-03	Michele Z.	Zaupa R.	Cambiati i termini: "Latex" in "LaTeX", "Precondizione" in "Precondizioni", "Postcondizione" in "Postcondizioni", "Processo primario" in "Processi primari", "Processo di supporto" in "Processi di supporto", "Processo organizzativo" in "Processi organizzativi", "Riferimento" in "Riferimenti", "Rischio" in "Rischi", "Sottocaso d'uso" in "Sottocasi d'uso", "Tecnologia" in "Tecnologie", "Verbale esterno" in "Verbalisti esterni", "Verbale interno" in "Verbalisti interni"
v.0.1.1	2024-04-02	Michele Z.	Zaupa R.	Aggiunti i termini: Best practices, Complessita' ciclomatica, Inspection, Modello a V, Pull request, Walkthrough e Way of Working, con le rispettive descrizioni
v.0.1.0	2024-03-22	Brotto D.	Zaupa R.	Corrette le definizioni di PB,RTB e Fornitura. Aggiunto il termine Caption
v.0.0.2	2024-03-20	Brotto D.	Zaupa R.	Stilate le definizioni di tutti i termini, aggiunti alcuni termini ambigui come Prenotazione e Ordinazione e rimosso Acquisizione
v.0.0.1	2024-01-19	Michele Z.	Brotto D.	Creata struttura del glossario e aggiunti i termini

Indice

1	Introduzione	7
2	A	8
2.1	Analisi Dei Requisiti	8
2.2	Analisi Dei Rischi	8
2.3	Applicazione Web Responsive	8
2.4	Attore	8
3	B	9
3.1	Best practices	9
3.2	Branch	9
4	C	10
4.1	Candidatura	10
4.2	Capitolato	10
4.3	Caption	10
4.4	Caso d'Uso	10
4.5	Changelog	10
4.6	Codifica	10
4.7	Complessità ciclomatica	11
4.8	Consuntivo	11
4.9	Controllo di Configurazione	11
5	D	12
5.1	Design	12
5.2	Diagramma dei Casi d'Uso	12
5.3	Diario di Bordo	12
5.4	Discord	12
5.5	Documento Esterno	12
5.6	Documento Interno	13
6	E	14
6.1	Easy Meal	14
6.2	Eccezione	14
6.3	Editor	14
7	F	15
7.1	Feature	15
7.2	Feedback	15
7.3	Fornitura	15
7.4	Framework	15
7.5	Funzionalità	15
8	G	16
8.1	Git	16
8.2	Gitflow	16
8.3	Github	16

9 H	17
10 I	18
10.1 Imola Informatica	18
10.2 Infrastruttura	18
10.3 Ingegneria del software	18
10.4 Inspection	18
11 J	19
11.1 Jira	19
12 K	20
13 L	21
13.1 Latex	21
14 M	22
14.1 Manuale Utente	22
14.2 Milestone	22
14.3 Modello a V	22
15 N	23
15.1 Norme di Progetto	23
16 O	24
16.1 Overleaf	24
16.2 Ordinazione	24
17 P	25
17.1 PB	25
17.2 PDF	25
17.3 Piano di Progetto	25
17.4 Piano di Qualifica	25
17.5 PoC	25
17.6 Posta elettronica	26
17.7 Postcondizioni	26
17.8 Precondizioni	26
17.9 Prenotazione	26
17.10Preventivo	26
17.11Principio di miglioramento continuo	26
17.12Processo	26
17.13Processi primari	27
17.14Processi di supporto	27
17.15Processi organizzativi	27
17.16Progettazione architetturale	27
17.17Progettazione dettagliata	27
17.18Pull request	27
17.19Push	27

18 Q	28
18.1 Qualità di processo	28
18.2 Qualità di prodotto	28
19 R	29
19.1 Redattore	29
19.2 Repository	29
19.3 Requisito	29
19.4 Requisiti funzionali desiderabili	29
19.5 Requisiti funzionali obbligatori	29
19.6 Riferimenti	29
19.7 Riferimenti normativi	29
19.8 Riferimenti informativi	30
19.9 Rilascio	30
19.10Rischi	30
19.11Rischio organizzativo	30
19.12Rischio relativo al prodotto	30
19.13Rischio tecnologico	30
19.14Risorsa Umana	30
19.15RTB	31
20 S	32
20.1 Scenario	32
20.2 Scenario principale	32
20.3 Scenario alternativo	32
20.4 Scrum	32
20.5 Sistema	32
20.6 Software	32
20.7 Sottocasi d'uso	33
20.8 Specifica tecnica	33
20.9 Sprint	33
20.10Strumento	33
21 T	34
21.1 Tecnologie	34
21.2 Telegram	34
21.3 Template	34
21.4 Test	34
21.5 Ticket	34
21.6 Trigger	35
22 U	36
22.1 UML	36
23 V	37
23.1 Verbale	37
23.2 Verbali esterni	37
23.3 Verbali interni	37

23.4	Versionamento	37
24	W	38
24.1	Walkthrough	38
24.2	Way of Working	38
25	X	39
26	Y	40
27	Z	41

1 Introduzione

Il glossario ha lo scopo di raccogliere i termini tecnici usati nel corso del progetto, al fine di facilitare la comprensione della documentazione, sia per i membri del gruppo che per i lettori esterni.

2 A

2.1 Analisi Dei Requisiti

L'analisi dei requisiti è il processo di raccolta, documentazione e analisi delle esigenze e delle funzionalità di un sistema software. Questo processo coinvolge la comprensione approfondita dei requisiti dell'utente, delle necessità del business e delle specifiche tecniche per sviluppare un sistema che soddisfi le aspettative degli stakeholder. Gli obiettivi principali dell'analisi dei requisiti includono la definizione chiara e completa delle funzionalità del sistema, la specifica dei vincoli e delle restrizioni, e l'identificazione dei requisiti non funzionali come quelli relativi alle prestazioni, sicurezza e usabilità.

2.2 Analisi Dei Rischi

L'analisi dei rischi è il processo di identificazione, valutazione e gestione dei potenziali problemi e delle minacce che potrebbero influenzare il successo di un progetto o di un'attività. Questo processo coinvolge la valutazione dei rischi potenziali, la determinazione della loro probabilità di accadere e dell'impatto che potrebbero avere sul progetto, nonché lo sviluppo di strategie per mitigare o gestire tali rischi. L'obiettivo dell'analisi dei rischi è identificare precocemente le potenziali problematiche e adottare misure preventive o correttive per ridurre al minimo gli impatti negativi sul progetto.

2.3 Applicazione Web Responsive

Un'applicazione web progettata per adattarsi automaticamente e fornire un'esperienza utente ottimale su una varietà di dispositivi e dimensioni dello schermo, inclusi computer desktop, laptop, tablet e smartphone, utilizzando tecniche di progettazione e sviluppo che consentono all'interfaccia utente di adattarsi dinamicamente alle dimensioni del dispositivo dell'utente.

2.4 Attore

Un attore è entità che interagisce con il sistema svolgendo delle attività, intesa sia come persona che come sistema terzo/esterno. Ciascuna entità è caratterizzata dall'insieme delle azioni che può compiere.

3 B

3.1 Best practices

Le best practices nel contesto della produzione software sono l'applicazione di metodi e procedure che nel corso del tempo e attraverso l'esperienza pratica hanno dimostrato di essere le migliori in termini di efficienza ed efficacia nel raggiungimento degli obiettivi prefissati.

3.2 Branch

Un ramo (branch) in GitHub è una versione separata del codice sorgente di un progetto software. Consente agli sviluppatori di lavorare su nuove funzionalità o correzioni di bug senza influenzare direttamente il codice principale (ramo principale o 'master'). I branch sono utilizzati per sviluppare in modo isolato, consentendo agli sviluppatori di sperimentare liberamente senza compromettere la stabilità del codice principale. Una volta completate le modifiche su un ramo, è possibile integrare (effettuare il 'merge') le modifiche nel ramo principale preferibilmente tramite una richiesta di pull (pull request) per incorporare le modifiche nel codice principale.

4 C

4.1 Candidatura

La candidatura in un contesto di progetto didattico indica l'atto attraverso il quale uno studente o un gruppo di studenti manifesta il proprio interesse a partecipare allo sviluppo di un progetto proposto dall'azienda o dall'istituzione didattica. Essa è un'azione volontaria che implica la presentazione di una proposta o di un interesse formale per essere considerati come partecipanti al progetto.

4.2 Capitolato

Un capitolato è un documento tecnico allegato a un contratto di appalto, che definisce le specifiche tecniche delle opere da eseguire nel contesto del contratto stesso. Solitamente parte integrante del contratto, il capitolato stabilisce i diritti, i doveri e le particolarità relative all'esecuzione dei lavori tra le parti coinvolte. Serve da riferimento per delineare gli accordi tra i soggetti privati coinvolti nel progetto.

4.3 Caption

La caption di una tabella o di un'immagine in una documentazione è una breve descrizione o etichetta che fornisce informazioni sul contenuto della tabella o dell'immagine stessa. Ha lo scopo di identificare e spiegare il contenuto visivo o i dati presentati nella tabella, mentre nel caso delle immagini può includere una breve descrizione o un titolo per aiutare il lettore a comprendere il contesto della stessa.

4.4 Caso d'Uso

Un caso d'uso definisce le interazioni tra gli attori (utenti) e il sistema software, rappresentando sequenze di azioni che guidano l'utilizzo del sistema per raggiungere determinati obiettivi. Queste interazioni descrivono come il sistema deve essere utilizzato e quali funzionalità devono essere esposte agli utenti.

4.5 Changelog

Un changelog è una sezione di documentazione che tiene traccia delle modifiche apportate a un software o a un progetto nel tempo. Esso elenca le variazioni, le correzioni di bug, le nuove funzionalità e altre modifiche significative, fornendo un registro storico delle revisioni effettuate. Il changelog è utile per gli sviluppatori e gli utenti finali per comprendere le modifiche apportate a una determinata versione del software oppure di un documento e per monitorare l'evoluzione del prodotto nel tempo.

4.6 Codifica

La codifica è il processo attraverso il quale si traducono le specifiche di progettazione in istruzioni eseguibili da un computer. Questo coinvolge la scrittura e la programmazione del codice sorgente utilizzando un linguaggio di programmazione specifico. Durante la codifica, gli sviluppatori scrivono algoritmi e istruzioni che definiscono il comportamento del software, implementando le funzionalità richieste e risolvendo i problemi identificati durante la fase di progettazione. La codifica è un

passaggio cruciale nello sviluppo del software e richiede attenzione ai dettagli e competenze tecniche specifiche.

4.7 Complessità ciclomatica

La complessità ciclomatica è una metrica software utilizzata per misurare la complessità di un programma. Essa viene calcolata utilizzando il grafo di controllo di flusso del programma, dove i nodi del grafo rappresentano gruppi indivisibili di istruzioni e gli archi orientati connettono due nodi se il secondo può essere eseguito immediatamente dopo il primo. La complessità ciclomatica di una sezione di codice esprime il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso del programma.

4.8 Consuntivo

Il consuntivo è un rapporto che confronta le previsioni o le stime con i dati effettivi o reali, al fine di valutare le variazioni e le differenze tra ciò che era previsto e ciò che è stato effettivamente raggiunto. Nell'ambito dello sviluppo del software o di altri progetti, il consuntivo può essere utilizzato per analizzare il bilancio finanziario, il tempo impiegato, le risorse utilizzate e altre metriche di progetto. Fornisce una valutazione critica delle prestazioni e dell'efficacia del progetto, consentendo di prendere decisioni informate e di apportare eventuali correzioni di rotta necessarie per mantenere il progetto sul giusto percorso.

4.9 Controllo di Configurazione

Il controllo di configurazione è un processo utilizzato nel campo dello sviluppo del software per gestire le modifiche alla configurazione e alla struttura del software stesso. Coinvolge l'identificazione, la registrazione, la revisione e l'approvazione delle modifiche apportate al codice sorgente, alla documentazione, ai file di configurazione e ad altri elementi del progetto. Il controllo di configurazione aiuta a mantenere la coerenza e l'integrità del sistema software durante l'evoluzione del progetto, garantendo che le modifiche vengano gestite in modo ordinato e tracciabile. Questo processo è essenziale per il mantenimento della qualità del software e per garantire che tutte le versioni e le varianti del sistema siano correttamente documentate e controllate.

5 D

5.1 Design

Il design nel contesto dello sviluppo del software è la fase in cui vengono definiti l'architettura, la struttura e le specifiche dettagliate del sistema software da creare. Questo processo include la progettazione dell'interfaccia utente, la scelta delle tecnologie e dei linguaggi di programmazione, nonché la pianificazione dell'implementazione del sistema. È una fase cruciale che assicura un'organizzazione efficiente e logica delle funzionalità del software, garantendo la sua scalabilità, manutenibilità e facilità d'uso.

5.2 Diagramma dei Casi d'Uso

Il diagramma dei casi d'uso è una rappresentazione grafica delle interazioni tra gli attori esterni al sistema software e il sistema software stesso. È composto da attori, casi d'uso e le relazioni tra di essi, e fornisce una panoramica visiva delle funzionalità offerte dal sistema e dei suoi utilizzi da parte degli attori. Questo strumento aiuta a comprendere i requisiti funzionali del sistema, le interazioni tra gli utenti e il sistema stesso, e fornisce una base per la progettazione e lo sviluppo del software.

5.3 Diario di Bordo

Il diario di bordo è un documento utilizzato per registrare in modo sistematico le attività svolte, le decisioni prese, gli eventi significativi e le osservazioni durante lo sviluppo di un progetto. Serve come registro dettagliato del lavoro svolto, delle sfide affrontate e delle soluzioni adottate nel corso del tempo. Il diario di bordo aiuta a tracciare il progresso del progetto, a monitorare le scadenze, a identificare eventuali problemi e a documentare le lezioni apprese per migliorare i processi futuri.

5.4 Discord

Discord è una piattaforma di comunicazione vocale e testuale progettata per la collaborazione remota, la formazione online e la comunicazione tra gruppi di persone. Offre funzionalità come chat di testo e vocali, canali organizzati per argomento, server personalizzabili, integrazioni con altre applicazioni e molto altro. Discord è ampiamente apprezzato per la sua facilità d'uso, la scalabilità e la ricchezza di funzionalità, rendendolo una scelta popolare per la comunicazione online.

5.5 Documento Esterno

Un documento esterno è un'opera o un insieme di documenti che sono destinati a essere condivisi con terzi al di fuori del gruppo di sviluppo o dell'organizzazione. Questi documenti possono includere proposte di progetto, rapporti di avanzamento, documenti di requisiti, manuali utente, documenti di consegna o qualsiasi altra documentazione destinata ad aziende proponenti, clienti, i professori o altri stakeholder esterni al processo di sviluppo. La creazione e la gestione accurata dei documenti esterni sono cruciali per garantire una comunicazione chiara, trasparente ed efficace con gli interessati esterni al progetto.

5.6 Documento Interno

Un documento interno è un file o un insieme di documenti utilizzati all'interno del gruppo di sviluppo o dell'organizzazione per scopi di comunicazione, pianificazione, gestione o documentazione. Questi documenti possono includere specifiche tecniche, documenti di progettazione, diagrammi, report di progresso, verbali delle riunioni, linee guida interne o qualsiasi altra documentazione destinata esclusivamente all'uso all'interno del team di lavoro. La creazione e la gestione efficace dei documenti interni sono fondamentali per garantire una collaborazione efficiente, un flusso di lavoro organizzato e una condivisione accurata delle informazioni all'interno del gruppo.

6 E

6.1 Easy Meal

Easy Meal è un progetto proposto da Imola Informatica e sviluppato con l'obiettivo di semplificare il processo di prenotazione dei tavoli nei ristoranti, offrendo agli utenti la possibilità di pianificare in anticipo il proprio pasto e ordinarlo in modo collaborativo con altri utenti. Con Easy Meal, gli utenti possono prenotare facilmente un tavolo, selezionare le loro preferenze alimentari e ordinare il cibo desiderato prima del giorno desiderato, consentendo loro di godersi un'esperienza gastronomica senza problemi e senza attese. Grazie a questa piattaforma intuitiva e user-friendly, prenotare un tavolo e decidere cosa mangiare diventa un'esperienza semplice e piacevole per tutti i clienti dei ristoranti.

6.2 Eccezione

Un'eccezione è un'anomalia o evento imprevisto durante l'esecuzione di un programma che interrompe il flusso normale di esecuzione. Può essere causata da errori di programmazione, input non validi o condizioni impreviste. Richiede una gestione appropriata per mantenere la stabilità del software.

6.3 Editor

Un editor è un'applicazione software utilizzata per creare, modificare e gestire file di testo o codice sorgente. Fornisce funzionalità come formattazione del testo, ricerca e sostituzione, e strumenti per la gestione dei progetti.

7 F

7.1 Feature

Una feature è una caratteristica o funzionalità specifica di un prodotto software. Le feature definiscono il comportamento o le capacità del software e possono includere qualsiasi cosa, dalle semplici azioni dell'utente alle complesse operazioni di sistema. Le feature sono progettate per soddisfare determinati requisiti o fornire determinati vantaggi agli utenti finali. Possono essere descritte attraverso specifiche funzionali, requisiti o casi d'uso e sono spesso evidenziate durante lo sviluppo del software per garantire che il prodotto finale soddisfi le aspettative degli utenti.

7.2 Feedback

I feedback sono opinioni, valutazioni o commenti forniti dagli utenti o dagli stakeholder su un prodotto, un servizio o un'esperienza. Il feedback può riguardare diversi aspetti, come l'usabilità, le prestazioni, le funzionalità o il design di un'applicazione software. È importante raccogliere, analizzare e utilizzare il feedback per migliorare continuamente il prodotto o il servizio, soddisfare le esigenze degli utenti e garantire una migliore esperienza complessiva.

7.3 Fornitura

La fornitura indica il processo primario che riguarda l'approvvigionamento, l'acquisizione e la consegna di prodotti o servizi esterni necessari per il completamento del progetto, nonché la gestione delle relazioni con i fornitori esterni.

7.4 Framework

Un framework è un'infrastruttura software che fornisce un'organizzazione predefinita per lo sviluppo di applicazioni. Essenzialmente, è una struttura di supporto su cui è possibile costruire e organizzare il codice. I framework forniscono una serie di librerie, modelli di progettazione, componenti e strumenti che semplificano lo sviluppo di software, consentendo agli sviluppatori di concentrarsi sulla logica dell'applicazione anziché dover reinventare ogni singolo aspetto. Spesso sono progettati per affrontare problemi comuni nello sviluppo di software, come la gestione delle richieste web, la manipolazione dei dati, la sicurezza e altro ancora. I framework possono essere generici o specifici per determinati tipi di applicazioni o linguaggi di programmazione.

7.5 Funzionalità

Le funzionalità si riferiscono alle capacità o agli attributi specifici di un software, di un'applicazione o di un sistema che consentono agli utenti di eseguire determinate azioni o di ottenere determinati risultati. Le funzionalità sono le varie operazioni, servizi o caratteristiche che un prodotto software offre agli utenti per soddisfare determinati scopi o necessità. Possono includere azioni interattive, processi automatizzati, visualizzazioni dei dati, strumenti di gestione e altro ancora. Le funzionalità sono progettate per rispondere ai requisiti e alle esigenze degli utenti, migliorare l'usabilità del software e offrire un'esperienza soddisfacente agli utenti finali.

8 G

8.1 Git

Git è un sistema di controllo di versione distribuito utilizzato principalmente nello sviluppo software. Consente agli sviluppatori di tenere traccia delle modifiche apportate al codice sorgente nel corso del tempo, coordinare il lavoro con altri membri del team e gestire le diverse versioni di un progetto in modo efficiente. Git consente di creare repository (archivi di file sorgente e cronologia delle modifiche) sia localmente che su server remoti, facilitando la collaborazione tra sviluppatori distribuiti geograficamente. È ampiamente utilizzato nell'industria del software per gestire progetti di qualsiasi dimensione e complessità.

8.2 Gitflow

GitFlow è un modello di branching e di workflow basato su Git, progettato per gestire progetti software con un ciclo di sviluppo complesso e ramificato. Questo approccio fornisce una struttura chiara per la gestione delle diverse fasi dello sviluppo, inclusi i rilasci, le correzioni di bug e le nuove funzionalità. GitFlow prevede l'utilizzo di due branch principali, 'master' e 'develop', oltre a una serie di branch temporanei per le funzionalità in sviluppo denominati 'feature branch', i bugfix e i rilasci. Questo modello facilita il lavoro in team e la collaborazione su progetti complessi, consentendo di mantenere una storia di versionamento pulita e organizzata.

8.3 Github

GitHub è una piattaforma di hosting di codice sorgente basata su Git, che fornisce strumenti per la gestione dei repository, la collaborazione tra sviluppatori e il controllo delle versioni dei progetti software. Su GitHub, gli sviluppatori possono caricare i propri progetti, tenere traccia delle modifiche tramite commit, creare e gestire branch, collaborare con altri utenti tramite problemi e richieste di pull, e distribuire le proprie applicazioni. È ampiamente utilizzato sia da sviluppatori individuali che da team di sviluppo per lo sviluppo di software open source e progetti privati.

9 H

10 I

10.1 Imola Informatica

Imola Informatica, proponente del progetto Easy Meal, è una società indipendente di consulenza IT. Tutto ciò che riguarda il mondo dell'information technology la riguarda, gli interessa e appassiona. La società in gioco ogni volta in cui una azienda pubblica o privata vuole migliorare i propri servizi, innovare i propri processi di lavoro e gli approcci di management per cogliere le opportunità business offerte dalla trasformazione digitale. Sono a servizio dei principali gruppi finanziari e assicurativi e ogni giorno sono a fianco di grandi aziende e piccole startup nel gestire il cambiamento tecnologico e culturale.

10.2 Infrastruttura

L'infrastruttura rappresenta l'insieme degli elementi fisici e tecnologici necessari per il funzionamento di un sistema o di un'applicazione. Questi elementi includono hardware, software, reti, server, database e tutte le risorse e le tecnologie utilizzate per supportare le operazioni di un'organizzazione o di un progetto. Nell'ambito del progetto Easy Meal, l'infrastruttura potrebbe comprendere i server per l'hosting dell'applicazione, i database per la gestione dei dati dei ristoranti e degli utenti, nonché la rete e gli strumenti di sicurezza necessari per garantire il corretto funzionamento e la protezione del sistema.

10.3 Ingegneria del software

L'ingegneria del software è un insieme di principi e pratiche utilizzati per progettare, sviluppare, mantenere, testare e valutare il software per computer. Le relative tecniche vengono impiegate per guidare il processo di sviluppo del software, che comprende la definizione, l'implementazione, la valutazione, la misurazione, la gestione, il cambiamento e il miglioramento del ciclo di vita del software. Questa disciplina pone un forte accento sulla gestione della configurazione del software, che implica il controllo sistematico delle modifiche alla configurazione e il mantenimento dell'integrità e della tracciabilità della configurazione e del codice durante tutto il ciclo di vita del sistema, mediante l'uso del versioning del software.

10.4 Inspection

L'Inspection è una tipologia di analisi statica che consiste nella revisione di parti specifiche del codice e della documentazione attraverso liste di controllo (checklist). Viene eseguita quando si ha già un'idea di dove potrebbero esserci possibili problemi, al fine di intervenire tempestivamente e sistematicamente.

11 J

11.1 Jira

Jira è una piattaforma di gestione del lavoro e dei progetti utilizzata per tracciare le attività, assegnare compiti, pianificare progetti e collaborare tra membri del team.

12 K

13 L

13.1 Latex

LaTeX viene spesso utilizzato per redigere documenti tecnici, come specifiche dei requisiti, documenti di progetto, manuali utente e rapporti di testing. La sua struttura modulare, la gestione avanzata della formattazione e la facilità nel gestire equazioni matematiche e tabelle lo rendono una scelta popolare tra gli sviluppatori e i professionisti del settore. LaTeX consente inoltre di mantenere una formattazione uniforme e di automatizzare alcune operazioni, rendendo più efficiente il processo di documentazione del progetto.

14 M

14.1 Manuale Utente

Il Manuale Utente è un documento che fornisce istruzioni dettagliate su come utilizzare un determinato prodotto software o sistema. È destinato agli utenti finali e fornisce indicazioni chiare e concise su come eseguire le diverse operazioni, navigare nell'interfaccia utente e sfruttare le funzionalità offerte dal software. Il manuale utente può includere tutorial, guide passo-passo, spiegazioni delle funzionalità e istruzioni per la risoluzione dei problemi comuni. L'obiettivo principale del manuale utente è quello di consentire agli utenti di utilizzare il software in modo efficace ed efficiente, migliorando così l'esperienza complessiva dell'utente.

14.2 Milestone

Una milestone è un punto di riferimento significativo o un obiettivo importante all'interno di un progetto, utilizzato per misurare il progresso e il raggiungimento di determinati traguardi. Le milestone sono solitamente associate a date specifiche o a completamenti di specifiche attività o fasi di un progetto. Servono come punti di controllo per valutare se il progetto sta procedendo secondo i piani e per identificare eventuali ritardi o problemi. Le milestone possono essere utilizzate anche per comunicare i progressi del progetto alle parti interessate e per stabilire scadenze chiare e tangibili per il completamento delle attività.

14.3 Modello a V

Il Modello a V è una rappresentazione grafica del ciclo di vita di un progetto di sviluppo software. Esso mostra in maniera dettagliata le relazioni fra le fasi di analisi e pregettazione, la produzione e il testing e ciò permette di minimizzare i rischi di progetto e di migliorare e garantire la qualità del prodotto.

15 N

15.1 Norme di Progetto

Nel documento Norme di Progetto sono riportate tutte le linee guida tecniche e procedurali stabilite per regolare l'esecuzione e il controllo di un progetto. Questo documento definisce standard e procedure relative all'organizzazione del lavoro, alla metodologia di sviluppo, alla gestione dei documenti, delle configurazioni e dei rischi, nonché alla pianificazione e al monitoraggio delle attività del progetto.

16 O

16.1 Overleaf

Overleaf è una piattaforma online per la scrittura collaborativa di documenti LaTeX. Consente agli utenti di creare e modificare documenti LaTeX direttamente nel browser, senza la necessità di installare software aggiuntivo sul proprio computer. Overleaf offre strumenti per la gestione dei progetti, il controllo delle versioni, la compilazione automatica dei documenti e la condivisione dei lavori con altri collaboratori. È particolarmente popolare nell'ambito accademico e scientifico per la sua facilità d'uso e la possibilità di collaborare in tempo reale.

16.2 Ordinazione

Nella fase di ordinazione dell'applicazione Easy Meal, l'utente seleziona e richiede un piatto specifico dal menu virtuale disponibile sull'applicazione o sul sito web. Durante questa fase, l'utente esplora le opzioni del menu, visualizza le descrizioni dei piatti, seleziona gli elementi desiderati e invia la propria richiesta di ordine al ristorante.

17 P

17.1 PB

La Product Baseline (PB) è un insieme consolidato di documenti e materiali che rappresentano lo stato approvato di un prodotto software in un determinato momento del ciclo di vita del progetto. Include specifiche di progettazione, documenti di requisiti, codice sorgente e altri artefatti pertinenti. La Product Baseline serve come riferimento per valutare il progresso del progetto e assicurare che il prodotto soddisfi i requisiti stabiliti durante la fase di pianificazione. È il risultato di un'analisi e di una revisione approfondita dei requisiti e delle specifiche del progetto, ed è approvata dal cliente o dagli stakeholder.

17.2 PDF

Il Portable Document Format (PDF) è un formato di file sviluppato da Adobe che consente di rappresentare documenti in modo indipendente dall'hardware e dal software utilizzati per crearli o visualizzarli. I file PDF mantengono la formattazione originale del documento, inclusi testo, immagini e layout, consentendo la condivisione e la visualizzazione coerenti su diverse piattaforme e dispositivi.

17.3 Piano di Progetto

Il Piano di Progetto è un documento che definisce gli obiettivi, le risorse, le attività e le scadenze necessarie per completare con successo un progetto. Include la pianificazione delle attività, la suddivisione dei compiti, la stima dei tempi e dei costi, nonché le strategie per gestire i rischi e rispettare le scadenze. Il Piano di Progetto fornisce una guida dettagliata per il team di progetto e per gli stakeholder, consentendo una gestione efficace e un monitoraggio delle attività lungo tutto il ciclo di vita del progetto.

17.4 Piano di Qualifica

Il Piano di Qualifica è un documento che definisce le strategie, le attività e le risorse necessarie per garantire che il prodotto soddisfi i requisiti di qualità stabiliti. Esso descrive le procedure di verifica e validazione che verranno utilizzate per valutare e confermare che il prodotto soddisfi gli standard di qualità definiti. Include anche gli obiettivi di qualità, le metriche di valutazione, i criteri di accettazione e le responsabilità del team per garantire che il prodotto finale sia conforme alle aspettative del cliente. Il Piano di Qualifica è essenziale per garantire che il prodotto sia testato in modo rigoroso e che soddisfi gli standard di qualità richiesti.

17.5 PoC

Il Proof of Concept (PoC) è una realizzazione preliminare o parziale di un progetto, finalizzata a dimostrare la fattibilità delle idee e delle tecnologie coinvolte nel progetto stesso, basandosi su principi o concetti fondamentali del prodotto finale. Essa costituisce un prototipo, utilizzato per valutare la fattibilità del progetto.

17.6 Posta elettronica

La posta elettronica è un sistema di comunicazione che consente agli utenti di inviare e ricevere messaggi tramite internet. I messaggi sono inviati e ricevuti elettronicamente attraverso server di posta elettronica, consentendo agli utenti di comunicare in modo rapido e efficiente a distanza.

17.7 Postcondizioni

Le postcondizioni sono degli stato del sistema o le condizioni che devono essere soddisfatte dopo l'esecuzione di un caso d'uso di una funzione specifica. Indicano lo stato o le modifiche che il sistema dovrebbe avere raggiunto al termine del caso d'uso o della funzione presa in esame

17.8 Precondizioni

Le precondizioni sono degli stato o condizioni del sistema che devono essere soddisfatte affinché un caso d'uso o una funzione possa essere eseguita con successo. Indicano lo stato iniziale o le circostanze necessarie perché il caso d'uso o la funzione presa in esame possa essere avviata e completata correttamente.

17.9 Prenotazione

La prenotazione è la fase del processo in cui l'utente desidera prenotare un tavolo presso un ristorante utilizzando l'applicazione. Durante questa fase, l'utente inserisce i dettagli della prenotazione, come la data, l'orario, il numero di persone e eventuali altre richieste. Una volta completata la prenotazione, il sistema invia una conferma all'utente e al ristorante, cercando di fare in modo che il tavolo sia riservato per il momento richiesto.

17.10 Preventivo

Il preventivo è un'analisi stimata dei costi e delle risorse necessarie per lo sviluppo e l'implementazione di un progetto software. Include valutazioni delle ore di lavoro, delle risorse di sviluppo e delle eventuali spese. Il preventivo serve a pianificare e gestire le risorse disponibili per il progetto, nonché a garantire che il lavoro venga completato entro i limiti temporali e finanziari stabiliti.

17.11 Principio di miglioramento continuo

Il principio di miglioramento continuo si riferisce alla pratica di costante riflessione e aggiornamento delle metodologie, dei processi e degli strumenti impiegati nel ciclo di vita del software. Si adotta questo principio per valutare costantemente il proprio lavoro, identificare punti di forza e di debolezza e apportare miglioramenti incrementali. Questo approccio mira a ottimizzare l'efficienza, la qualità e l'efficacia del lavoro svolto, consentendo al team di adattarsi ai cambiamenti, risolvere eventuali problemi e perseguire un costante progresso nel raggiungimento degli obiettivi del progetto.

17.12 Processo

Il processo è una successione di eventi/attività che producono un risultato visibile e concreto. Deve essere condotto in modo sistematico, disciplinato e misurabile.

17.13 Processi primari

I processi primari si riferiscono alle attività direttamente coinvolte nella creazione del prodotto software o nel fornire valore diretto al cliente. Questi processi sono centrali per il raggiungimento degli obiettivi del progetto e includono attività come analisi dei requisiti, progettazione, implementazione, test e consegna del software.

17.14 Processi di supporto

I processi di supporto forniscono assistenza e risorse per i processi primari e organizzativi. Questi processi includono attività come la gestione della configurazione, la gestione della qualità, la formazione, la gestione delle risorse e la gestione dei rischi. L'obiettivo dei processi di supporto è quello di garantire che i processi primari e organizzativi siano eseguiti in modo efficiente ed efficace.

17.15 Processi organizzativi

I processi organizzativi sono responsabili della gestione e del coordinamento delle attività all'interno dell'organizzazione. Questi processi includono la definizione delle politiche e delle procedure aziendali, la pianificazione strategica, la gestione dei progetti e la comunicazione interna ed esterna. Gli obiettivi dei processi organizzativi sono quello di garantire che l'organizzazione abbia una struttura e un'infrastruttura adeguate per supportare l'esecuzione dei processi primari e di supporto.

17.16 Progettazione architetturale

La progettazione architetturale è il processo di definizione della struttura di base di un sistema software, che include l'identificazione dei componenti, delle loro interazioni e delle decisioni di progettazione chiave. L'obiettivo è garantire che il software soddisfi i requisiti funzionali e non funzionali.

17.17 Progettazione dettagliata

La progettazione dettagliata è la fase del processo di sviluppo del software in cui i dettagli tecnici dell'architettura definita nella progettazione architetturale vengono elaborati in modo più specifico. Questo include la definizione delle singole componenti, dei loro comportamenti, delle interfacce e delle relazioni tra di esse. La progettazione dettagliata fornisce una guida chiara per l'implementazione del sistema software.

17.18 Pull request

Le Pull request sono un meccanismo che consente a un membro del gruppo di notificare agli altri membri il completamento di una funzionalità all'interno di un determinato branch. Questo permette di avviare i processi di verifica della funzionalità e, eventualmente, di procedere con il merge.

17.19 Push

Il push è un'operazione utilizzata in Git per inviare le modifiche locali apportate al repository locale verso un repository remoto. Questo consente di condividere le modifiche con altri membri del team o di archivarle su un server remoto come GitHub.

18 Q

18.1 Qualità di processo

La qualità di processo si riferisce alla misura in cui un processo software è ben pianificato, gestito ed eseguito per raggiungere gli obiettivi di sviluppo del software in modo efficiente ed efficace.

18.2 Qualità di prodotto

La qualità di prodotto si riferisce alla misura in cui un prodotto software soddisfa i requisiti specificati e le aspettative degli utenti, fornendo funzionalità affidabili, sicure e adatte all'uso previsto.

19 R

19.1 Redattore

Un redattore è un individuo responsabile della stesura di documenti tecnici o di altro genere all'interno di un progetto. Il redattore assicura che i documenti siano chiari, accurati e coerenti.

19.2 Repository

Un repository è uno spazio di archiviazione digitale utilizzato per conservare, gestire e condividere file e dati relativi a un progetto. Può essere utilizzato per memorizzare codice sorgente, documentazione, risorse grafiche e altri file pertinenti al progetto. Un repository facilita la collaborazione tra membri del team consentendo loro di accedere, modificare e aggiornare i file in modo coordinato.

19.3 Requisito

Un requisito è una specifica o una condizione necessaria che un sistema, un prodotto o un servizio deve soddisfare per essere accettabile, utile o in grado di raggiungere determinati obiettivi. I requisiti vengono definiti durante il processo di analisi dei requisiti per guidare lo sviluppo e valutare il successo del progetto.

19.4 Requisiti funzionali desiderabili

I requisiti funzionali desiderabili sono quelli che, se soddisfatti, migliorerebbero l'esperienza dell'utente o aggiungerebbero valore al sistema, ma non sono considerati essenziali per il suo funzionamento di base.

19.5 Requisiti funzionali obbligatori

I requisiti funzionali obbligatori sono quei requisiti indispensabili per il corretto funzionamento del sistema o del prodotto, senza i quali il sistema non può essere considerato soddisfacente.

19.6 Riferimenti

Un riferimento in un documento si riferisce a un'informazione, un'idea o una fonte esterna che viene citata, menzionata o utilizzata come punto di riferimento per comprendere o supportare il contenuto del documento stesso. Può includere link ipertestuali, citazioni bibliografiche, numeri di pagina o altre forme di indicazioni che facilitano la consultazione o il richiamo ad altre fonti o parti del documento.

19.7 Riferimenti normativi

I riferimenti normativi sono quei riferimenti che indicano norme, standard o regolamenti ufficiali che devono essere rispettati durante lo svolgimento di un progetto o l'elaborazione di un documento. Questi possono includere leggi, direttive settoriali, standard di qualità o normative specifiche del settore.

19.8 Riferimenti informativi

I riferimenti informativi si riferiscono a documenti, pubblicazioni o fonti informative che forniscono ulteriori dettagli, contestualizzazioni o approfondimenti relativi al contenuto del documento principale, ma non sono considerati vincolanti o obbligatori per il progetto o l'elaborazione del documento stesso.

19.9 Rilascio

Il rilascio è un'istanza specifica di un prodotto software che è stato messo a disposizione degli utenti finali o dei clienti. Indica il momento in cui una determinata versione del software è pronta per essere distribuita e utilizzata dagli utenti. Un rilascio può includere nuove funzionalità, correzioni di bug, miglioramenti delle prestazioni o altre modifiche significative rispetto alle versioni precedenti del software.

19.10 Rischi

Un rischio è un evento o una condizione che, se si verificasse, potrebbe influenzare negativamente il raggiungimento degli obiettivi di un progetto. I rischi possono riguardare varie aree, come i tempi di consegna, i costi, la qualità del prodotto o la soddisfazione del cliente. Gli sforzi di gestione dei rischi sono finalizzati a identificare, valutare e mitigare i potenziali impatti negativi che i rischi potrebbero avere sul progetto.

19.11 Rischio organizzativo

Con il termine rischio organizzativo si riferisce alla possibilità che problemi interni all'organizzazione, come mancanza di risorse, conflitti tra membri del team o cambiamenti nell'organigramma aziendale, possano influenzare negativamente il successo del progetto.

19.12 Rischio relativo al prodotto

Il rischio relativo al prodotto riguarda i potenziali difetti, malfunzionamenti o mancanze nel prodotto software stesso. Potrebbe includere problemi di qualità del software, non conformità ai requisiti utente o problemi di sicurezza.

19.13 Rischio tecnologico

Si riferisce alla possibilità che le tecnologie utilizzate nel progetto, come linguaggi di programmazione, framework o strumenti, possano rivelarsi inadeguate, obsolete o incapaci di soddisfare i requisiti del progetto, causando ritardi o problemi durante lo sviluppo.

19.14 Risorsa Umana

Risorsa Umana si riferisce al personale coinvolto nel progetto, composto da membri del team coinvolti nell'implementazione, nell'esecuzione e nel controllo delle attività previste. Questa risorsa è essenziale per il successo del progetto, in quanto contribuisce con le proprie competenze, esperienze

e sforzi per raggiungere gli obiettivi stabiliti. La gestione efficace delle risorse umane include l'allocazione delle persone giuste ai compiti appropriati, la formazione e lo sviluppo del personale, la risoluzione dei conflitti e la creazione di un ambiente lavorativo collaborativo e motivante.

19.15 RTB

La Requirements Technology Baseline (RTB) è una linea di base tecnologica di un progetto software che rappresenta lo stato degli strumenti, delle tecniche e dei processi utilizzati per la gestione dei requisiti funzionali e non del sistema. Include gli strumenti software utilizzati per la raccolta, l'analisi, la tracciabilità e la gestione dei requisiti, nonché le procedure e le pratiche per l'identificazione, la documentazione e la validazione dei requisiti. La RTB fornisce una base stabile e affidabile per il processo di sviluppo del software, garantendo che gli strumenti e le tecnologie utilizzate siano adatti e siano stati compresi per lo scopo e soddisfino le esigenze del progetto.

20 S

20.1 Scenario

Nel documento di Analisi dei Requisiti, uno scenario rappresenta una sequenza di azioni o eventi che descrivono un possibile flusso di interazione tra l'utente e il sistema. Gli scenari sono utilizzati per comprendere e illustrare come il sistema dovrebbe comportarsi in determinate situazioni o sotto determinate condizioni. Ogni scenario include una serie di passi che descrivono le azioni compiute dagli utenti e le risposte del sistema, consentendo di visualizzare in modo chiaro e dettagliato il comportamento previsto del software. Gli scenari possono essere utilizzati per identificare requisiti funzionali, individuare casi d'uso e validare il sistema durante lo sviluppo e il testing.

20.2 Scenario principale

Lo scenario principale rappresenta il flusso di interazione tipico tra l'utente e il sistema per raggiungere l'obiettivo principale del caso d'uso o della funzionalità in esame. Esso descrive la sequenza di passi che si verificano nella maggior parte dei casi.

20.3 Scenario alternativo

Gli scenari alternativi, al contrario, rappresentano situazioni non standard o eccezionali che possono verificarsi durante l'esecuzione del caso d'uso. Questi scenari descrivono le deviazioni dal flusso principale causate da input inaspettati, condizioni particolari o errori. Gli scenari alternativi forniscono istruzioni su come gestire tali situazioni e ripristinare il flusso normale delle operazioni.

20.4 Scrum

Scrum è un framework Agile utilizzato per lo sviluppo di prodotti complessi. Si basa su un approccio iterativo e incrementale, in cui il lavoro viene diviso in intervalli di tempo definiti chiamati sprint. Durante uno sprint, il team si impegna a completare un insieme di attività prioritarie stabilite durante la pianificazione dello sprint. Gli sprint sono seguiti da sessioni di revisione per valutare il lavoro svolto e identificare eventuali miglioramenti.

20.5 Sistema

Il Sistema nel documento di Analisi dei Requisiti si riferisce all'insieme di componenti software, hardware e di sistema che collaborano per fornire le funzionalità richieste dal prodotto. Il sistema può comprendere diverse parti, come applicazioni software, database e server.

20.6 Software

Il termine "Software" si riferisce all'insieme di programmi, procedure e documentazione associata utilizzati su un sistema informatico per eseguire determinate operazioni o risolvere specifici problemi. Questo include sia il codice sorgente dei programmi che il software precompilato eseguibile. Il software può essere progettato per scopi diversi, come gestione dati, elaborazione informatica, comunicazione, intrattenimento e molti altri. In un contesto di sviluppo software, il termine "software" si riferisce al prodotto finale ottenuto dal processo di sviluppo, che può essere installato e utilizzato dagli utenti per scopi specifici.

20.7 Sottocasi d'uso

Il termine "Sottocaso d'uso" si riferisce a un'istanza specifica di un caso d'uso più ampio. Mentre un caso d'uso descrive un'interazione completa tra gli attori e il sistema per raggiungere un obiettivo specifico, un sottocaso d'uso dettaglia uno specifico scenario o flusso di lavoro all'interno di quel caso d'uso più ampio. I sottocasi d'uso vengono utilizzati per suddividere i casi d'uso complessi in unità gestibili e per fornire una descrizione più dettagliata delle azioni e dei passaggi coinvolti in un'interazione specifica con il sistema.

20.8 Specifica tecnica

La Specifica tecnica è la descrizione dettagliata della progettazione tecnica del software. Questa descrizione include informazioni sulle decisioni architetturali prese, i design pattern utilizzati, la struttura dei componenti software e le interazioni tra di essi, le tecnologie impiegate e altre specifiche rilevanti per l'implementazione del sistema. La specifica tecnica fornisce una guida chiara agli sviluppatori durante la fase di codifica e contribuisce a garantire coerenza e qualità nel prodotto finale.

20.9 Sprint

Lo Sprint è una fase di sviluppo iterativa e incrementale nel framework Agile, come ad esempio Scrum. Durante uno Sprint, il team di sviluppo lavora su una serie di attività concordate all'inizio dello Sprint, con l'obiettivo di consegnare un incremento di funzionalità completamente funzionante e potenzialmente consegnabile al termine del periodo prestabilito, generalmente di durata fissa, solitamente da una a quattro settimane. Gli Sprint offrono una struttura chiara per la pianificazione, l'esecuzione e il monitoraggio del lavoro del team, consentendo un rapido adattamento ai cambiamenti dei requisiti o delle priorità.

20.10 Strumento

Uno strumento nell'ambito dello sviluppo del software è un'applicazione, un programma o una risorsa che supporta specifiche attività o processi nel ciclo di vita del software. Gli strumenti possono includere ambienti di sviluppo integrati (IDE), sistemi di gestione di progetto come Jira, framework di test come Selenium, sistemi di controllo versione come Git, e molti altri. L'uso degli strumenti giusti può migliorare l'efficienza e la qualità del processo di sviluppo del software.

21 T

21.1 Tecnologie

La "tecnologia" nel contesto dello sviluppo del software si riferisce agli strumenti, ai linguaggi di programmazione, ai framework e ad altre risorse utilizzate per progettare, sviluppare e implementare soluzioni software. Questo termine può anche comprendere hardware, piattaforme di hosting, servizi cloud e altri componenti tecnologici utilizzati nel processo di sviluppo e distribuzione del software. La scelta delle tecnologie giuste può influenzare significativamente le prestazioni, la scalabilità, la manutenibilità e altre caratteristiche del prodotto software.

21.2 Telegram

Telegram è un'applicazione di messaggistica istantanea e di comunicazione. Consente agli utenti di scambiare messaggi di testo, foto, video, file e altro ancora tramite Internet. Telegram offre anche funzionalità come chat di gruppo, chiamate vocali e video, canali pubblici, bot automatizzati e crittografia end-to-end per garantire la privacy e la sicurezza delle comunicazioni. È stato utilizzato dal gruppo per la messaggistica istantanea e per fissare meeting di progetto anche con l'uso di sondaggi a scelta multipla e variabile.

21.3 Template

Un template, in ambito documentale, è un modello predefinito che fornisce una struttura organizzativa e visiva per la creazione di documenti. Questi modelli includono solitamente sezioni già definite, formattazione standard, stili di testo preimpostati e elementi grafici coerenti (usati ad esempio nei powerpoint dei diari di bordo e per le prime pagine di ogni documento, ovvero copertina e indice). L'uso di un template consente di uniformare l'aspetto e la struttura dei documenti prodotti da diversi autori, facilitando la creazione e la lettura dei contenuti.

21.4 Test

Il termine "test" si riferisce a un'attività mirata a valutare o verificare le funzionalità o le prestazioni di un sistema, un'applicazione o un componente software. I test possono essere condotti per individuare difetti, verificare il corretto funzionamento di una funzionalità o garantire il rispetto dei requisiti specificati.

21.5 Ticket

Un "ticket" in Jira è un elemento di tracciamento utilizzato per registrare e monitorare il lavoro in un progetto software. Ogni ticket rappresenta un'attività specifica da completare, come risolvere un bug, implementare una nuova funzionalità o completare un compito di manutenzione. I ticket contengono informazioni dettagliate sul lavoro da svolgere, inclusi i requisiti, le assegnazioni, le scadenze e lo stato di avanzamento. Possono essere assegnati a membri del team e seguiti nel tempo per garantire che vengano completati in modo efficace e tempestivo.

21.6 Trigger

Il "trigger" nell'analisi dei requisiti si riferisce a un evento o una condizione che innesci l'avvio di un determinato caso d'uso o processo all'interno del sistema software. È un elemento chiave per comprendere quando e come un particolare comportamento o funzionalità deve essere attivato nell'applicazione. I trigger possono essere attivati da azioni degli utenti, come clic su pulsanti o input di dati, o da eventi interni al sistema, come arrivo di notifiche o cambiamenti di stato. Identificare correttamente i trigger è fondamentale per definire in modo completo e accurato il comportamento del sistema.

22 U

22.1 UML

UML, acronimo di Unified Modeling Language, è un linguaggio di modellazione standardizzato utilizzato per descrivere, progettare e documentare sistemi software. Fornisce una serie di diagrammi grafici che consentono agli sviluppatori di visualizzare diversi aspetti di un sistema, inclusi struttura, comportamento, interazioni e flussi di dati.

23 V

23.1 Verbale

Un verbale è un documento che riporta in modo dettagliato le discussioni, le decisioni prese e gli eventi avvenuti durante una riunione, un incontro o una sessione di lavoro. Serve a registrare e documentare in maniera accurata ciò che è stato discusso e deciso durante l'incontro, consentendo ai partecipanti di ricordare e consultare le informazioni trattate in seguito. Un verbale può includere una lista degli argomenti affrontati, i partecipanti presenti, le decisioni prese, le azioni da intraprendere e le eventuali osservazioni o commenti aggiuntivi.

23.2 Verbali esterni

Un verbale esterno è un documento che riporta le discussioni, le decisioni e gli eventi avvenuti durante una riunione o un incontro con soggetti esterni al team di lavoro o all'organizzazione. È utilizzato nel nostro caso per comunicare in modo formale e documentato con aziende proponenti, professori, o altre parti interessate esterne.

23.3 Verbali interni

Un verbale interno è documento che riporta le discussioni, le decisioni e gli eventi avvenuti durante una riunione o un incontro all'interno del team di lavoro o dell'organizzazione stessa. Viene utilizzato per registrare e documentare le attività e le decisioni prese all'interno del gruppo di lavoro.

23.4 Versionamento

Il versionamento è un processo di gestione delle diverse versioni di un prodotto software o di altri documenti, che consente di tracciare e registrare le modifiche apportate nel tempo. Il versionamento permette di tenere traccia delle variazioni, delle correzioni di bug e delle nuove funzionalità implementate, facilitando il controllo delle revisioni e garantendo una corretta gestione delle versioni.

24 W

24.1 Walkthrough

Il Walkthrough è una tipologia di analisi statica che coinvolge sia il verificatore che l'autore del prodotto e che consiste nella revisione nel suo complesso del codice e della documentazione forniti, seguita da una discussione degli eventuali problemi individuati.

24.2 Way of Working

Il Way of Working è un insieme di processi operativi, regole e specifiche che definiscono il modo di lavorare del team e come vengono gestiti i processi all'interno del progetto. Un Way of Working efficace fornisce una struttura chiara e coesa per il team, promuove la coerenza, la trasparenza e l'efficienza nelle operazioni quotidiane, contribuendo così al successo complessivo del progetto.

25 X

26 Y

27 Z