



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025



Gruppo: SWEg Labs

Email: gruppo.sweg@gmail.com

Glossario

Versione 1.0.0

Stato	Approvato
Redazione	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Verifica	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Approvazione	...
Proprietario	Gruppo <i>SWEg Labs</i>
Uso	Esterno
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin <i>AzzurroDigitale Srl</i>

Registro delle modifiche

Versione	Data	Descrizione	Autore	Verifica
1.0.0	...	Approvazione del documento
...	...	Verifica del documento
...
0.2.0	...	Verifica del documento allo stato attuale
0.1.11	17-11-24	Aggiunti i termini <u>Specifiche funzionali</u> , <u>Specifiche tecniche</u> , <u>Contesto applicativo</u> , <u>Architettura</u> , <u>Accoppiamento</u> , <u>Pattern architetturale</u> , <u>Pull request</u> , <u>Build</u> , <u>Feedback</u> , <u>Google Meet</u> e <u>Issue Tracking System (ITS)</u>	Giacomo Loat	Federica Bolognini

Versione	Data	Descrizione	Autore	Verifica
0.1.10	16-11-24	Aggiunti i termini <u>Codifica</u> , <u>ISO/IEC 9126</u> , <u>Efficienza</u> , <u>Efficacia</u> , <u>Prodotto software</u> e <u>Test</u>	Riccardo Stefani	Federica Bolognini
0.1.9	16-11-24	Aggiunti i termini <u>Angular</u> , <u>NestJS</u> , <u>Node.js</u> , <u>Spring</u> , <u>Spring Boot</u> e <u>Telegram</u>	Riccardo Stefani	Michael Fantinato
0.1.8	13-11-24	Aggiunti i termini <u>Metrica</u>	Michael Fantinato	Filippo Righetto
0.1.9	13-11-24	Aggiunto il termine <u>ISO/IEC/IEEE 12207</u>	Federica Bolognini	Filippo Righetto
0.1.8	13-11-24	Aggiunti i termini <u>Demo</u> e <u>Snippet</u>	Riccardo Stefani	Filippo Righetto
0.1.7	12-11-24	Modificato il termine <u>Processo</u>	Giacomo Loat	Riccardo Stefani
0.1.6	12-11-24	Aggiunti i termini <u>Diario di Bordo</u> , <u>Agile (modello di sviluppo)</u> , <u>Incrementale (modello di sviluppo)</u> , <u>Burndown (diagramma di)</u> , <u>Preventivo</u>	Giacomo Loat	Riccardo Stefani
0.1.5	11-11-24	Aggiunti i termini <u>Capitolato</u> , <u>Caso d'uso</u> , <u>Attore</u> , <u>Scenario d'uso</u> , <u>Intelligenza artificiale</u> , <u>IA</u> , <u>API</u> , <u>Applicazione web</u> , <u>Browser</u> , <u>Confluence</u> , <u>Slack</u> e <u>Onboarding</u>	Filippo Righetto	Riccardo Stefani
0.1.4	11-11-24	Aggiunto il termine <u>ISO/IEC 31000:2018</u>	Federica Bolognini	Michael Fantinato
0.1.3	10-11-24	Aggiunti i termini <u>Baseline</u> , <u>Branch</u> , <u>Feature</u> , <u>Git</u> , <u>IDE</u> , <u>Product Baseline (PB)</u> , <u>Proponente</u> , <u>Proof of Concept (PoC)</u> , <u>Merge</u> , <u>Minimum Viable Product (MVP)</u> , <u>Snake Case</u> e <u>Visual Studio Code (VSC)</u>	Riccardo Stefani	Michael Fantinato
0.1.2	09-11-24	Aggiunti i termini <u>Ciclo di vita del software</u> , <u>Processo</u> , <u>Verifica</u> e <u>Way of Working</u>	Riccardo Stefani	Giacomo Loat
0.1.1	08-11-24	Inserimento dei primi termini, tratti dai primi due verbali interni della RTB	Riccardo Stefani	Filippo Righetto
0.1.0	05-11-24	Creazione del documento	Riccardo Stefani	Filippo Righetto

Tabella 1: Registro delle modifiche

Indice

1	Introduzione	1
1.1	Obiettivo del documento	1
1.2	Struttura del documento	1
A		2
B		4
C		6
D		7
E		9
F		10
F.1	Feedback	10
G		11
H		12
I		13
J		15
K		16
L		17
M		18
N		19
O		20
P		21
Q		23
R		24
S		25
T		27
U		28
V		29
W		30
X		31
Y		32
Z		33

1 Introduzione

1.1 Obiettivo del documento

Questo documento è stato creato con l'intento di raccogliere tutti i termini che possano risultare sconosciuti ad un lettore esterno o che possano creare ambiguità. Lo scopo è quello di creare una base omogenea di conoscenza specifica.

1.2 Struttura del documento

Il documento è suddiviso in sezioni rappresentanti le lettere dell'alfabeto; all'interno di queste si troveranno i termini ambigui o specifici del progetto che sono stati contrassegnati nel resto della documentazione con un pedice_G alla fine della parola

A

Accoppiamento

Accoppiamento

In informatica, l'accoppiamento è il grado di dipendenza tra due o più componenti di un sistema software. Un accoppiamento elevato indica una forte interdipendenza tra le componenti, mentre un accoppiamento basso indica una minore dipendenza. Un basso accoppiamento è generalmente preferibile, poiché rende il sistema più flessibile, modulare e facile da mantenere.

Agile (modello di sviluppo)

Agile, modello di sviluppo

Approccio alla gestione dei progetti e allo sviluppo del software che enfatizza la flessibilità, la collaborazione e il miglioramento continuo. Ha come principi fondamentali consegna continua, collaborazione stretta tra sviluppatori e clienti, adattabilità ai cambiamenti, comunicazione diretta tra le parti e valutazione e miglioramento continuo del processo e del prodotto.

Analisi dei Requisiti

Processo fondamentale dello sviluppo di un prodotto software che si concentra sulla raccolta, analisi e definizione delle necessità e delle aspettative degli utenti finali, degli stakeholder e del sistema nel suo complesso. Questo processo mira a comprendere e documentare in modo chiaro e completo le esigenze, le funzionalità, le prestazioni e i vincoli che il sistema deve soddisfare. L'obiettivo principale dell'analisi dei requisiti è fornire una base solida per tutte le fasi successive dello sviluppo del software, assicurando che il prodotto finale soddisfi le esigenze degli utenti e raggiunga gli obiettivi del progetto.

Analisi dei Rischi

L'analisi dei rischi è il processo di identificazione, valutazione e prioritizzazione dei rischi in un progetto, sistema o attività, al fine di ridurre o gestire il loro impatto potenziale. Viene utilizzata per prevedere gli eventi negativi che potrebbero influenzare il successo di un progetto e per determinare le azioni preventive o correttive da intraprendere.

Angular

Framework open-source sviluppato da Google per creare applicazioni web dinamiche e scalabili. Basato su TypeScript, utilizza un'architettura a componenti, il data binding bidirezionale e strumenti integrati per lo sviluppo di interfacce utente robuste e performanti.

API

Acronimo di Application Programming Interface, è un insieme di regole e protocolli che consente a diverse applicazioni software di comunicare tra loro per scambiare dati, caratteristiche e funzionalità. Le API fungono da intermediari, permettendo lo scambio di informazioni tra software diversi, semplificando e accelerando lo sviluppo di applicazioni.

Applicazione web

Applicazione accessibile via web tramite un browser (Safari, Chrome, Firefox, Edge etc.) e può funzionare sia da mobile che da desktop. Si tratta di un'architettura tipicamente di tipo client-server, che offre determinati servizi all'utente.

Architettura

In ambito informatico, l'organizzazione di base di un sistema, espressa dai suoi componenti, dalle relazioni tra di loro e con l'ambiente, e dai principi che ne guidano il progetto e l'evoluzione. Essa comprende le strutture del sistema, necessarie per ragionare su di esso, che includono elementi software, le relazioni tra di essi e le loro proprietà.

Attore

Nel contesto dell'analisi dei requisiti e del design del software, il termine attore rappresenta chiunque interagisca con il sistema. Gli attori possono essere utenti, altri sistemi o qualsiasi altra entità che abbia un ruolo nelle interazioni con il sistema modellato.

AzzurroDigitale

AzzurroDigitale è una società italiana con sede a Padova. Si occupa di digitalizzazione di processi sia con prodotti proprietari che di terze parti, e ha l'obiettivo di accompagnare le aziende manifatturiere nella transizione 5.0.

B

Backlog

Insieme di compiti/attività da completare per un certo obiettivo. All'interno del framework Scrum, ne esistono due tipi principali: il product backlog, che è la lista delle funzionalità da implementare, e lo sprint backlog, che contiene le attività da svolgere durante un particolare sprint. Un'attività interna al backlog porta valore ad un progetto perchè possiede:

- Stato, che segnala se l'attività è stata completata, in corso o non ancora iniziata.
- Priorità, che indica l'importanza dell'attività rispetto alle altre.
- Assegnatario, cioè una persona incaricata a svolgere l'attività. Questa assegnazione non è vincolante, infatti se un membro del team ha terminato la sua attività può prendersi a carico un'altra attività presente nel backlog anche se non era stata inizialmente assegnata a lui.
- Scadenza, cioè un termine temporale entro il quale l'attività deve essere svolta.

Baseline

Nel contesto dell'ingegneria del software, stato di avanzamento che rappresenta un insieme di punti di arrivo che ci si pone come obiettivo di raggiungere in una milestone, dimostrando che l'incremento delle modifiche condotte ha portato a un risultato.

Best Practices

Nello sviluppo software, metodologie che attraverso l'esperienza e la sperimentazione sono state identificate come modi efficaci e raccomandati di affrontare determinati problemi o compiti nel processo di sviluppo del software. Queste pratiche sono considerate migliori (best) perché hanno dimostrato di portare a risultati di alta qualità, facilitando la manutenzione del codice e promuovendo una migliore collaborazione nel team di sviluppo.

Branch

Letteralmente "ramo", indica un'entità che si sviluppa o si dirama da un punto principale. Nel contesto di un sistema di controllo delle versioni, un branch rappresenta una linea di sviluppo separata. Può essere utilizzato per sviluppare nuove funzionalità, risolvere bug o implementare modifiche senza influenzare direttamente il ramo principale del codice, noto come master o main.

Browser

Applicazione software progettata per consentire agli utenti di navigare in Internet, visualizzare pagine web e accedere a contenuti online.

Build

Build

Definito anche come costruzione, è il processo di compilazione di un progetto software, in cui il codice sorgente viene trasformato in un formato eseguibile. La build può includere una serie di attività come la compilazione del codice sorgente, la creazione di file di configurazione e la generazione di file di installazione o di pacchetti per la distribuzione del software. Normalmente viene automatizzato tramite l'utilizzo di strumenti affidabili e riproducibili, gestendo le dipendenze del progetto e automatizzando il processo di rilascio e il controllo di versione.

Burndown (diagramma di)

Burndown, diagramma di

Strumento grafico utilizzato nella gestione agile dei progetti per tracciare la quantità di lavoro rimanente nel tempo. Esso mostra la diminuzione progressiva (“burn down”) delle attività o dei punti stima rimanenti nel corso del tempo, consentendo al team di progetto di valutare il proprio progresso e adattare la pianificazione in base alle esigenze. A differenza del diagramma di Gantt, il diagramma di Burndown si concentra sulla visualizzazione dell’avanzamento reale rispetto al piano temporale.

C

Capitolato

Documento privato tra chi commissiona il lavoro e il gruppo (ditta) che lo esegue, in cui viene esposto un problema che il proponente necessita di risolvere e specifica le norme e i vincoli da rispettare per lo sviluppo del prodotto software specifico.

Caso d'uso

Descrizione dettagliata di come un utente (attore) interagisce con l'applicazione per il compimento di un'attività specifica. È uno strumento utilizzato nel contesto dello sviluppo software per individuare i requisiti funzionali del prodotto e per fornire una visuale chiara delle interazioni che possono avvenire all'interno dell'applicazione

Checklist

Lista dettagliata di elementi, attività o criteri specifici che devono essere controllati, esaminati o completati durante le diverse fasi del ciclo di vita del software. E' utilizzata come strumento di controllo e verifica.

Ciclo di vita del software

Serie di fasi attraverso le quali un software passa dal suo concepimento iniziale fino al suo ritiro o dismissione. È un concetto chiave nell'ingegneria del software e fornisce una struttura organizzativa per il processo di sviluppo del software.

Codifica

Fase del processo di sviluppo software in cui gli sviluppatori traducono i requisiti e il design del sistema in linguaggio di programmazione, creando il codice sorgente. Durante questa fase, gli sviluppatori seguono gli standard di codifica e le buone prassi per assicurare la leggibilità, l'efficienza e la manutenibilità del codice.

Confluence

Piattaforma di collaborazione e gestione della conoscenza, utilizzata per creare, condividere e collaborare su documenti, progetti e informazioni all'interno di un team o di un'organizzazione. Confluence offre funzionalità come la creazione di wiki aziendali, la gestione di progetti, la documentazione tecnica e la collaborazione in tempo reale.

Contesto applicativo

Ambito o scenario in cui un'applicazione software è progettata per essere utilizzata. Il contesto applicativo definisce le condizioni, le esigenze e le caratteristiche specifiche dell'ambiente in cui l'applicazione deve operare, influenzando il design, le funzionalità e le prestazioni del software.

D

Demo

Una demo (abbreviazione di "dimostrazione") è una presentazione interattiva o registrata di un software o di una funzionalità specifica all'interno di un progetto. Viene usata per mostrare lo stato attuale dello sviluppo, illustrando come funzionano le caratteristiche implementate e come si comporta il sistema rispetto ai requisiti stabiliti. Le demo sono solitamente presentate a stakeholder, clienti, o team interni per ottenere feedback, verificare funzionalità e orientare lo sviluppo futuro. Le demo sono particolarmente utili nei processi iterativi e agili, dove è importante monitorare e comunicare costantemente i progressi del progetto.

Diagramma di Gantt

Strumento di visualizzazione temporale utilizzato nella gestione dei progetti per rappresentare le attività pianificate nel tempo. È composto da una barra orizzontale che rappresenta l'arco temporale totale del progetto e da barre orizzontali più piccole che rappresentano le singole attività del progetto. Ogni barra è posizionata lungo l'asse temporale in base alle date di inizio e fine previste per l'attività.

Diagramma UML

Acronimo di Unified Modeling Language, un diagramma UML diagramma utilizzato per modellare, descrivere e visualizzare sistemi software e processi di sviluppo software. È uno standard industriale nel campo dell'ingegneria del software e fornisce una serie di diagrammi, ognuno dei quali si concentra su un aspetto specifico del sistema o del processo. Sono diagrammi UML ad esempio i diagrammi dei casi d'uso, i diagrammi delle classi e i diagrammi delle funzionalità.

Diagramma UML dei casi d'uso

Diagramma UML che rappresenta le interazioni tra utenti (attori) e il sistema, descrivendo come gli utenti utilizzano il sistema per raggiungere obiettivi specifici. Ogni caso d'uso rappresenta una funzione o un'attività significativa, utile per descrivere i requisiti funzionali. È spesso il primo passo nella progettazione di un sistema software e aiuta a identificare le funzioni principali e il modo in cui il sistema interagisce con gli utenti.

Diagramma UML delle classi

Diagramma UML che descrive la struttura statica di un sistema, mostrando le classi, i loro attributi, i metodi e le relazioni tra di esse (come ereditarietà, associazioni e aggregazioni). Questo diagramma è fondamentale per la programmazione orientata agli oggetti poiché fornisce una rappresentazione visiva della struttura del codice, aiutando a comprendere le interconnessioni tra le varie entità e a definire i componenti principali.

Diario di Bordo

Nel contesto del progetto didattico, il "diario di bordo" è un'attività settimanale in cui ogni gruppo di progetto documenta e presenta pubblicamente il progresso del proprio lavoro. Questa attività include la descrizione delle attività svolte, delle difficoltà incontrate, dei dubbi e delle incertezze, e dei piani per il periodo successivo. L'obiettivo è promuovere una discussione aperta e riflessiva per migliorare la consapevolezza e la gestione del progetto collaborativo.

Discord

Piattaforma VoIP (Voice over IP: tecnologia che rende possibile effettuare una conversazione sfruttando una connessione internet), messaggistica istantanea e distribuzione digitale progettata per la comunicazione.

Draw.io

Draw.io (ora chiamato diagrams.net) è uno strumento gratuito per la creazione di diagrammi, disponibile sia come applicazione web che come app desktop per vari sistemi operativi. Viene utilizzato ampiamente per progettare e documentare diagrammi di flusso, architetture software, diagrammi UML, organigrammi, mappe mentali, wireframe, e altri tipi di rappresentazioni visive.

E

Efficacia

Capacità di raggiungere gli obiettivi desiderati o di produrre gli effetti previsti.

Efficienza

Capacità di svolgere un compito, un'attività o un processo nel modo più economico e con il minimo spreco di risorse.

F

Feature

Una feature è una specifica funzionalità o capacità di un sistema software che ne arricchisce il comportamento e le possibilità d'uso. Le feature sono aggiunte o miglioramenti che rispondono a esigenze degli utenti, migliorano l'esperienza d'uso, o risolvono specifici problemi. L'introduzione o il miglioramento di una feature segue un ciclo di sviluppo completo, dalla raccolta dei requisiti fino alla verifica e al testing.

F.1 Feedback

Il feedback è un'informazione, un'opinione o una valutazione che viene fornita in risposta a un'azione, a una prestazione o a un comportamento. Ha lo scopo di evidenziare punti di forza e aree di miglioramento, permettendo a chi lo riceve di comprendere meglio come il proprio operato viene percepito e come può migliorare o adattarsi. Il processo di feedback è un processo per cui il risultato dell'azione di un sistema si riflette sul sistema stesso per correggerne o modificarne il comportamento.

Fogli Google

Fogli Google è un'applicazione web di Google, parte della suite di produttività Google Workspace, che consente di creare, modificare e condividere fogli di calcolo online. È uno strumento particolarmente apprezzato per il suo accesso immediato da browser, le funzionalità collaborative in tempo reale, e l'integrazione con altri servizi Google. Così come ogni foglio di calcolo, include la possibilità di creare grafici e diagrammi basati su dati tabellari.

G

Gantt, diagramma di

Vedi *Diagramma di Gantt* [G](#).

Git

Software per il controllo di versione distribuito utilizzabile tramite interfaccia a riga di comando.

GitHub

Servizio di hosting per progetti software. Il sito è principalmente utilizzato da sviluppatori che caricano il codice sorgente di programmi in dei repository e lo rendono scaricabile e migliorabile da altri sviluppatori. Questi ultimi possono interagire con i proprietari dei repository tramite un sistema per inviare segnalazioni di bug o richieste di funzionalità (issue tracker), un sistema per copiare il software in una versione modificabile (fork), un sistema per proporre modifiche agli sviluppatori originali (pull request) e un sistema di discussione legato al codice del repository (commenti).

GitHub Pages

GitHub Pages è un servizio di hosting gratuito offerto da GitHub che permette agli utenti di creare e pubblicare facilmente siti web statici direttamente dai loro repository GitHub. È utilizzato comunemente per creare siti di documentazione, pagine personali o di progetto, blog e portali di portfolio. GitHub Pages è particolarmente apprezzato perché permette di ospitare un sito senza costi e con aggiornamenti automatici ogni volta che il repository viene modificato.

GitHub Projects

GitHub Projects è uno strumento di gestione dei progetti integrato in GitHub, ideato per aiutare sviluppatori e team a organizzare, pianificare e tracciare il lavoro sui progetti direttamente all'interno dell'ambiente GitHub. Si basa su un sistema flessibile di "project board" simile a Kanban, che offre un modo visuale per coordinare i task e monitorare l'avanzamento del lavoro. GitHub Projects è uno strumento ideale per team che lavorano su progetti di sviluppo software, in quanto permette di gestire l'intero processo di sviluppo all'interno di GitHub stesso. Grazie alla sua integrazione nativa con il codice, le issues e le pull requests, aiuta a mantenere sincronizzati i task e a ridurre il contesto di cambiamento per gli sviluppatori, migliorando il flusso di lavoro e facilitando la collaborazione su GitHub.

Glossario

Elenco organizzato di termini tecnici, acronimi e definizioni utilizzati nel contesto del progetto. Questo documento fornisce una chiara comprensione dei concetti e dei linguaggi specifici impiegati nel progetto, aiutando a ridurre ambiguità e fraintendimenti tra i membri del team e gli stakeholder.

[Google Meet](#)

Google Meet

Servizio di videoconferenza sviluppato da Google. È parte di Google Workspace e offre funzionalità di videochiamata, chat, condivisione dello schermo e registrazione delle riunioni. Google Meet è ampiamente utilizzato per incontri di lavoro, lezioni online, webinar e conferenze, grazie alla sua facilità d'uso, alla stabilità della connessione e alla possibilità di partecipare senza dover scaricare alcun software.

Gulpease, indice di

Indice di leggibilità di un testo specificamente in lingua italiana, che utilizza il numero delle parole, delle frasi e delle lettere per facilitare il calcolo automatico della leggibilità.

H

...

I

IA

Acronimo di intelligenza artificiale

IDE

Acronimo di Integrated Development Environment, è un ambiente di sviluppo, ovvero un software che, in fase di programmazione, supporta i programmatori nello sviluppo e debugging del codice sorgente di un programma, segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a fornire una serie di strumenti e funzionalità di supporto alla fase stessa di sviluppo e debugging.
Incrementale (modello di sviluppo)

Incrementale, modello di sviluppo

Approccio alla creazione di software che suddivide il progetto in piccoli segmenti o incrementi. Ogni incremento rappresenta una versione funzionante del software che include nuove funzionalità o miglioramenti rispetto alla versione precedente.

Intelligenza artificiale

Disciplina che studia come realizzare sistemi informatici in grado di simulare il pensiero umano. Sistemi basati su di essa hanno dunque l'abilità di mostrare capacità umane quali il ragionamento, l'apprendimento, la pianificazione e la creatività.

ISO/IEC 31000:2018

La norma ISO 31000 "Risk management - Principles and guidelines", in italiano UNI ISO 31000 Gestione del rischio - Principi e linee guida. È una guida che fornisce principi e linee guida generali per la gestione del rischio. Può essere utilizzata da qualsiasi organizzazione pubblica, privata o sociale, associazione, gruppo o individuo, e non è specifica per nessuna industria o settore. La ISO 31000 può essere applicata nel corso dell'intero ciclo di vita di un'organizzazione, ed essere adottata per molte attività come la definizione di strategie e decisioni, operazioni, processi, funzioni, progetti, prodotti, servizi e beni. Può inoltre essere applicata a qualsiasi tipo di rischio, sia per conseguenze di tipo positivo che negativo.

ISO/IEC 9126

La norma ISO/IEC 9126 è uno standard internazionale per la valutazione della qualità del software. Definisce un modello di qualità software che si basa su sei caratteristiche generali di qualità (funzionalità, affidabilità, usabilità, efficienza, manutenibilità, portabilità) e su 27 sotto-caratteristiche. Lo standard è stato sostituito dalla norma ISO/IEC 25010:2011, che definisce un modello di qualità software più ampio e aggiornato.

ISO/IEC/IEEE 12207

Lo standard ISO 12207 stabilisce un processo di ciclo di vita del software, compreso processi ed attività relative alle specifiche ed alla configurazione di un sistema. Ad ogni processo corrisponde un insieme di risultati (outcome): in totale ci sono 43 processi, 133 attività, 325 sottoattività e 236 risultati (la nuova ISO/IEC 12207:2008 definisce 43 sistemi e processi software). Lo standard ha come obiettivo principale quello di fornire una struttura comune che permetta a clienti, fornitori, sviluppatori, tecnici, manager di usare gli stessi termini e lo stesso linguaggio per definire gli stessi processi. La struttura dello standard è stata concepita per essere flessibile e modulare in modo che sia adattabile alle necessità di chiunque lo utilizzi.

Issue

Una issue su GitHub (e altre piattaforme di gestione del codice e dei progetti) è un elemento utilizzato per tracciare problemi, richieste di funzionalità, idee o miglioramenti relativi a un progetto. È uno strumento fondamentale per organizzare il lavoro collaborativo e garantire che tutti i membri del team siano aggiornati sui task e le priorità. Ogni issue rappresenta un singolo elemento che richiede attenzione o azione, e fornisce un luogo centralizzato per discuterlo, seguirlo e risolverlo.

Issue Tracking System (ITS)

Issue Tracking System (ITS)

Un Issue Tracking System (ITS) è un software o un sistema di gestione che consente di creare, monitorare e gestire issue (problemi, task, richieste o bug) all'interno di un progetto. Questo strumento è utilizzato principalmente per facilitare la collaborazione tra membri del team e per garantire che tutte le problematiche e le richieste siano gestite in modo organizzato, trasparente e tracciabile. Un ITS è spesso impiegato da team di sviluppo software, ma può essere utile anche in contesti di supporto clienti, gestione di progetto, e qualsiasi ambito dove occorra tracciare richieste di lavoro o problemi.

J

Jira

Jira è uno strumento di gestione dei progetti e di issue tracking sviluppato da Atlassian, ampiamente utilizzato per la pianificazione, il monitoraggio e il controllo di progetti, in particolare nell'ambito dello sviluppo software. Nato come strumento di gestione dei bug e dei problemi, Jira è diventato uno dei principali strumenti per il project management, soprattutto per le organizzazioni che adottano metodologie agili come Scrum e Kanban.

K

...

L

L^AT_EX

Linguaggio di marcatura per la preparazione di testi, basato sul programma di composizione tipografica TeX. LaTeX è ampiamente utilizzato per la creazione di documenti scientifici e tecnici grazie alla sua capacità di gestire formule matematiche complesse e alla sua alta qualità tipografica.

M

Merge

Operazione fondamentale nei sistemi di controllo delle versioni. Essa è utilizzata per combinare le modifiche apportate in due branch separati in un singolo branch.

Metrica

Misura quantitativa utilizzata per valutare, quantificare e analizzare diversi aspetti del processo di sviluppo del software, del prodotto software stesso o della gestione del progetto. Le metriche forniscono dati numerici che consentono di valutare l'andamento del progetto, la qualità del software, l'efficacia dei processi e altri aspetti rilevanti.

Milestone

In ingegneria del software e nella gestione dei progetti, punto di riferimento o traguardo significativo che sancisce il termine di un periodo nel ciclo di vita di un progetto. Le milestone rappresentano generalmente eventi chiave, compimenti o obiettivi importanti che indicano il progresso del progetto. L'obiettivo che ci si pone durante una milestone è realizzare una baseline.

Minimum Viable Product (MVP)

Versione ridotta di un prodotto, che incorpora solo le funzioni essenziali per soddisfare le esigenze di base. Viene utilizzato per rilasciare un prodotto come test e ricevere feedback dall'utenza per migliorare poi il prodotto finito con tutte le funzionalità.

N

NestJS

Framework Node.js per lo sviluppo di applicazioni lato server, basato su TypeScript. Utilizza un'architettura modulare e concetti come iniezione delle dipendenze e decoratori, ispirandosi a framework come Angular, per creare applicazioni scalabili e facilmente mantenibili.

Node.js

Runtime JavaScript open-source e multiplatforma, progettato per l'esecuzione di codice JavaScript lato server. Basato sul motore V8 di Google Chrome, permette di creare applicazioni veloci, scalabili e basate su eventi, sfruttando un modello non bloccante per la gestione delle operazioni I/O.

Norme di Progetto

Insieme di linee guida, procedure e regole stabilite per regolare e standardizzare l'approccio, il processo e l'output del lavoro all'interno del progetto. Queste norme possono riguardare diversi aspetti del progetto, come la gestione del codice, la documentazione, la comunicazione e la gestione dei rischi. L'obiettivo delle norme di progetto è promuovere la coerenza, la qualità e l'efficienza nel processo di sviluppo del software, consentendo al team di lavorare in modo più efficace e collaborativo.

O

Onboarding

Il processo di integrazione e formazione di un nuovo dipendente, collaboratore o utente in un'organizzazione o piattaforma. L'onboarding ha l'obiettivo di fornire tutte le informazioni e le risorse necessarie per adattarsi al nuovo ambiente di lavoro, comprendere la cultura aziendale, acquisire competenze specifiche e diventare operativi nel più breve tempo possibile.

P

Pattern architetturale

Pattern architetturale

Soluzione generale e riutilizzabile a un problema che si verifica comunemente nell'ambito dello sviluppo del software. I pattern architetturali forniscono un approccio testato e comprovato per la progettazione e l'implementazione del software.

Piano di Progetto

Documento formale che delinea in dettaglio la pianificazione, l'esecuzione, il monitoraggio e il controllo di tutte le attività coinvolte nella realizzazione di un progetto. Questo documento fornisce una roadmap chiara e organizzata, comprensiva di obiettivi, risorse, scadenze e strategie di gestione dei rischi. Essenziale per la gestione efficace di un progetto, il piano di progetto serve come guida per il team di lavoro e gli stakeholder, fornendo una struttura che facilita il coordinamento delle attività e l'assegnazione delle risorse.

Piano di Qualifica

Documento che stabilisce gli standard di qualità, i processi e le attività di testing che saranno implementati durante lo sviluppo di un progetto. Contiene una descrizione dettagliata delle strategie di testing, delle metriche di valutazione e dei criteri di accettazione del prodotto finale. L'obiettivo principale del Piano di Qualifica è garantire che il prodotto soddisfi gli standard di qualità prefissati e che il processo di sviluppo segua procedure coerenti ed efficaci.

Preventivo

Preventivo

Stima dei costi e delle risorse necessarie per completare un determinato lavoro o progetto.

Processo

Processo

Insieme di attività correlate e coese che trasformano ingressi (bisogni) in uscite (prodotti) secondo regole date, consumando risorse nel farlo.

Prodotto software

Un prodotto software è un'applicazione, un sistema o un programma informatico risultante dal processo di sviluppo del software. In altre parole, è il risultato tangibile e funzionante di attività di progettazione, sviluppo, test e manutenzione svolte dal team di sviluppo.

Product Baseline (PB)

E la seconda revisione di avanzamento del progetto didattico. Comprende un prodotto software con design definitivo, chiamato *Minimum Viable Product (MVP)_G*.

Proof of Concept (PoC)

Versione preliminare di un'applicazione o di una soluzione software che viene sviluppata per dimostrare la fattibilità tecnica di un'idea o di un concetto. Viene utilizzata per testare rapidamente l'efficacia di un approccio, identificare eventuali limitazioni delle tecnologie scelte e valutare se l'idea può essere realizzata in modo pratico.

Proponente

Nel contesto dell'ingegneria del software, colui che presenta un'idea, un progetto o una proposta e ne sostiene la realizzazione. Il gruppo *SWEg Labs* ha come proponente l'azienda *AzzurroDigitale G*.

Pull request

Pull request

Nel sistema di versionamento integrato da GitHub, operazione che permette di proporre modifiche al codice sorgente di un progetto. Una pull request consente agli sviluppatori di richiedere l'integrazione delle loro modifiche nel branch principale del repository, facilitando la revisione e la collaborazione.

Q

Qualità

Insieme delle caratteristiche di un'entità che ne determinano la capacità di soddisfare esigenze sia espresse che implicite. Si parla di qualità del prodotto software in termini di:

- **Qualità Intrinseca:** conformità ai requisiti, idoneità all'uso;
- **Qualità Relativa:** soddisfazione del cliente;
- **Qualità Quantitativa:** misurazione oggettiva, per confronto.

R

Repository

In termini informatici, un luogo o un archivio dove vengono conservati e gestiti dati, documenti o, nel contesto del software, il codice sorgente di un progetto. Nell'ambito dei sistemi di controllo delle versioni come Git, un repository è una struttura dati che archivia anche la cronologia completa delle modifiche apportate al codice sorgente di un progetto.

Requirement and Technology Retrospective (RTB)

La prima revisione di avanzamento del progetto didattico. Fissa i requisiti da soddisfare in accordo con il proponente, motiva le tecnologie, i framework e le librerie adottate dimostrandone adeguatezza e compatibilità tramite il Proof of Concept (PoC).

Retrospettiva

Vedi *Sprint Retrospective*_G.

S

Scenario d'uso

Uno scenario d'uso è una descrizione narrativa di come un utente interagisce con un sistema software per raggiungere un determinato obiettivo. Rappresenta una sequenza di azioni e reazioni tra l'utente e il sistema in un contesto specifico e fornisce una visione dettagliata del comportamento atteso del software in una situazione reale.

Script

Uno script è un insieme di istruzioni scritte in un linguaggio di scripting (come Python, JavaScript, Bash, ecc.) progettato per automatizzare compiti specifici o per eseguire operazioni in modo diretto su un sistema, applicazione o ambiente web. A differenza di un programma complesso, uno script tende a essere più leggero e adatto per operazioni mirate e ripetibili, senza richiedere una fase di compilazione lunga o complessa.

Slack

Piattaforma di messaggistica istantanea e collaborazione progettata per team e organizzazioni. Permette di comunicare tramite canali tematici, inviare messaggi diretti, condividere file, e integrare altre applicazioni e strumenti di lavoro. Slack favorisce la comunicazione in tempo reale, migliorando la produttività e la gestione dei progetti, e viene utilizzato soprattutto in ambienti di lavoro agili e collaborativi.

Snake Case

Lo Snake Case è uno stile di scrittura di nomi in cui le parole sono separate da un trattino basso o underscore (-) e tutte le lettere sono generalmente minuscole (es.: nome_variabile_uno). Viene spesso contrapposto allo stile Camel Case, in cui invece le parole successive alla prima vengono concatenate con l'iniziale maiuscola (es.: nomeVariabileUno).

Snippet

Uno snippet è un piccolo blocco di codice riutilizzabile che svolge una funzione specifica o rappresenta un esempio di utilizzo di una determinata funzionalità. Gli snippet sono utilizzati per risparmiare tempo durante la scrittura del codice, poiché forniscono pezzi di codice già pronti che possono essere facilmente inseriti e adattati nel contesto del progetto.

Specifiche funzionali

Specifiche funzionali

Le specifiche funzionali sono una descrizione dettagliata delle funzionalità e dei comportamenti che un sistema software deve avere per soddisfare i requisiti degli utenti. Le specifiche funzionali definiscono cosa il software deve fare, come deve rispondere a determinate azioni e quali risultati deve produrre in risposta a determinati input.

Specifiche tecniche

Specifiche tecniche

Le specifiche tecniche sono una descrizione dettagliata delle caratteristiche tecniche e delle prestazioni che un sistema software deve avere per soddisfare i requisiti tecnici e funzionali. Le specifiche tecniche definiscono come il software deve essere progettato, implementato e testato per garantire che funzioni correttamente e risponda alle esigenze degli utenti.

Spring

Framework Java che fornisce un ecosistema modulare per sviluppare applicazioni lato server. Facilita la gestione delle dipendenze, configurazioni, sicurezza, accesso ai dati e integrazioni, promuovendo un'architettura scalabile e flessibile basata su principi come l'inversione del controllo (IoC) e la programmazione orientata agli aspetti (AOP).

Spring Boot

Framework Java che semplifica lo sviluppo di applicazioni Spring, ovvero applicazioni basate sul framework Spring per la gestione di componenti, dipendenze e funzionalità lato server come sicurezza, accesso ai dati e integrazione. Spring Boot offre configurazioni predefinite, un server embedded e una struttura modulare, permettendo di creare applicazioni pronte all'uso con meno configurazioni manuali.

Sprint

Periodo di tempo prefissato entro il quale lavorare producendo dei risultati documentati. Gli sprint sono al centro delle metodologie Agile, atte a produrre risultati piccoli e in maniera costante.

Sprint Retrospective

Incontro finalizzato ad analizzare l'andamento dello sprint quando questo è terminato, per migliorare la performance futura del team di sviluppo. La riunione retrospettiva dello sprint è quindi propedeutica allo sprint successivo.

T

Telegram

Applicazione multiplatforma che permette una facile comunicazione con gruppi/canali, organizzando la comunicazione sulla base di strumenti e funzionalità automatiche offerte (bot), condividendo facilmente file e messaggi in un canale unico di comunicazione.

Test

In ingegneria del software, processo sistematico e controllato per valutare un sistema software o una sua componente allo scopo di individuare eventuali difetti, errori o comportamenti indesiderati.

Tracciamento

Il tracciamento è il processo di monitoraggio e documentazione dell'evoluzione e delle modifiche di un sistema, dei suoi requisiti e dei suoi componenti, all'interno di un progetto. Questo processo è essenziale per mantenere una visione chiara dello stato attuale del sistema, delle sue modifiche, e delle relazioni tra diversi elementi (come requisiti, attività di design, e componenti di codice).

U

...

V

Verifica

Processo che include un insieme di attività volte a garantire che il lavoro svolto durante lo sviluppo del software rispetti gli standard, i requisiti e le aspettative stabilite. La verifica è essenziale per garantire che il software sia di alta qualità, risponda alle esigenze degli utenti e riduca il rischio di errori e difetti. Si svolge a periodi regolari durante il corso del progetto.

Versionamento

Processo che realizza il cosiddetto “controllo di versione”, stabilendo la storia cronologica delle azioni fatte per una certa attività, tracciando i cambiamenti occorsi e permettendo di tornare a uno stadio precedente qualora necessario.

Visual Studio Code (VSC)

IDE _G libero e gratuito sviluppato da Microsoft per Windows, Linux e macOS. Permette sia di scrivere codice sorgente per un prodotto software sia di scrivere testo di documentazione.

W

Way of working

Il termine “Way of working” (modo di lavorare) si riferisce al modo in cui un individuo, un team o un’organizzazione svolge le proprie attività lavorative. Questo concetto può includere processi, metodologie, abitudini, strumenti e culture aziendali che influenzano la gestione del lavoro e la collaborazione. Un modo di lavorare efficace ed efficiente può migliorare la produttività, la qualità del lavoro e la soddisfazione dei membri del team, promuovendo un ambiente di lavoro positivo e collaborativo.

Il way of working deve essere:

- *Sistematico*, cioè non deve dipendere dalle singole persone, bensì deve essere adattabile anche a futuri membri del team.
- *Disciplinato*, cioè deve far seguire norme di comportamento che blocchino l’istinto e favoriscano il ragionamento.
- *Quantificabile*, cioè si deve poter verificare concretamente la sua implementazione.

X

...

Y

...

Z

...