torchlight

torchlight.swe2324@outlook.com

Glossario

Versione 2.0.0

Redattori	Cappellari Marco
	Filippini Giovanni
	Meneghini Fabio
	Ye Tao Ren Federico
Verifica	Agafitei Ciprian
	Cappellari Marco
	Filippini Giovanni
	Meneghini Fabio
	Pluzhnikov Dmitry
Approvazione	De Laurentis Arianna Pia
Uso	Esterno
Destinatari	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Zucchetti S.p.A.

Registro delle Modifiche

Ver.	Data	Descrizione	Autore	Verifica
2.0.0	2024/04/08	AVerifica finale e convalida del documento	Agafitei Ciprian	Meneghini Fabio
1.0.3	2024/04/07	Aggiunto il termine *Docker*	Filippini Giovanni	Cappellari Marco
1.0.2	2024/03/15	Aggiunta di nuovi termini (*Model View Controller pattern*, *Sprint retrospective*, *Sprint review*)	Meneghini Fabio	De Laurentis Arianna Pia
1.0.1	2024/03/11	Aggiunta di nuovi termini (*Manuale utente*, *Scrum*, *Specifica Tecnica*)	De Laurentis Arianna Pia	Agafitei Ciprian
1.0.0	2024/01/14	Verifica finale e convalida del documento	Tao Ren Federico Ye	Pluzhnikov Dmitry

Ver.	Data	Descrizione	Autore	Verifica
0.2.3	2024/01/14	Aggiunta la definizione agli ultimi termini inseriti (*Diagramma di Burndown*, *Sentence similarity*, *Benchmark*, *Brainstroming*, *Walkthrough*, *Test di regressione*, *Test di integrazione*, *Roadmap*, *Validazione*) e corretti piccoli errori di ortografia	Filippini Giovanni	Cappellari Marco
0.2.2	2024/01/13	Aggiunti i termini *Qualità del software*, *Piano di progetto*, *Glossario*, *Modello a V*, *Test di funzionalità*, *Test di sistema*, *Test di accettazione* e *Checklist*	Cappellari Marco	Cappellari Marco
0.2.1	2024/01/11	Inserimento di nuovi termini apparsi nella documentazione	Ye Tao Ren Federico	Cappellari Marco
0.2.0	2024/01/05	Revisione dei termini e delle definizioni inserite finora	Agafitei Ciprian	Agafitei Ciprian
0.1.5	2023/12/23	Aggiunti i termini *UML*, *Embedded database* e *Vettore*	Meneghini Fabio	Pluzhnikov Dmitry

Ver.	Data	Descrizione	Autore	Verifica
0.1.4	2023/12/21	Aggiunti i termini *Debugging* e *Zero-shot Classification*	Ye Tao Ren Federico	Pluzhnikov Dmitry
0.1.3	2023/12/11	Aggiunto il termine *Architettura*, migliorata la definizione di alcuni termini	Ye Tao Ren Federico	Meneghini Fabio
0.1.2	2023/12/08	Aggiunti i termini *Deployment*, *Design pattern*, *Idioma*	Meneghini Fabio	Filippini Giovanni
0.1.1	2023/12/08	Inserimento di tutti i termini individuati fin'ora	Meneghini Fabio	Filippini Giovanni
0.1	2023/11/06	Creazione del documento	Filippini Giovanni	Pluzhnikov Dmitry

Tabella 1: Registro delle modifiche

Indice

Introduzione al documento

Questo documento contiene le definizioni specifiche dei termini utilizzati nella documentazione del progetto *ChatSQL*. L'obiettivo è eliminare possibili ambiguità e assicurare una chiara comprensione dei concetti adottati nel contesto del progetto.

A

Analisi dei requisiti L'analisi dei requisiti è una fase fondamentale dello sviluppo di un prodotto software. Essa si concentra sulla comprensione e la documentazione delle esigenze, delle funzionalità e dei vincoli che il sistema deve soddisfare. L'obiettivo principale dell'analisi dei requisiti è definire in modo chiaro e completo cosa il software deve fare, fornendo una base solida per tutte le fasi successive dello sviluppo del software.

Architettura L'*architettura del software* si riferisce alla struttura fondamentale di un sistema software. Questa struttura include gli elementi del sistema, le relazioni tra di essi e il modo in cui cooperano per raggiungere gli obiettivi del software. L'architettura del software fornisce una visione ad alto livello del sistema e stabilisce la base per la progettazione e l'implementazione dettagliata.

Attività Nell'ambito dell'ingegneria del software, un'*attività* è un'azione o un insieme di azioni specifiche e ben definite che vengono eseguite all'interno di un processo per raggiungere determinati obiettivi.

Attore Nel contesto dell'analisi dei requisiti e del design del software, il termine *attore* rappresenta chiunque interagisca con il sistema, sia esso una persona, un'altra applicazione o un dispositivo hardware.

B

Benchmark Nel contesto dell'informatica e dell'ingegneria del software, un *benchmark* è un insieme standardizzato di test o misurazioni utilizzato per valutare le prestazioni, la velocità o altri parametri di un sistema, dispositivo, componente hardware, software o algoritmo.

Best practice Il termine *best practice* si riferisce a un approccio o a un metodo riconosciuto come il più efficace e efficiente per affrontare una determinata situazione o per risolvere un problema specifico.

Brainstorming Il *brainstorming* è una tecnica creativa in cui un gruppo genera liberamente idee su un argomento senza giudizi critici. Favorisce la diversità e la quantità di proposte, stimolando l'innovazione. Le idee vengono registrate e successivamente valutate per identificare soluzioni pratiche. Ampiamente utilizzato in riunioni aziendali e sessioni creative.

Branch Letteralmente "ramo", sta ad indicare 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.

C

Capitolato Documento che contiene le specifiche e le condizioni per lo sviluppo di un progetto software. Il capitolato viene redatto dal proponente e viene presentato ai fornitori o agli sviluppatori interessati a partecipare all'appalto per la realizzazione del prodotto software.

Caso d'uso Un *caso d'uso* (o *use case* in inglese) cattura, descrive e definisce le interazioni tra un sistema e gli attori che interagiscono con esso. Esso rappresenta un insieme di scenari che descrivono come un sistema risponde a una specifica richiesta o interazione da parte di un attore. I casi d'uso sono utilizzati durante la fase di analisi dei requisiti per identificare, documentare e comprendere le funzionalità chiave del sistema dal punto di vista degli utenti o degli altri attori coinvolti.

Checklist Una 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. Utilizzata come strumento di controllo e verifica.

Ciclo di vita del software Il *ciclo di vita del software* si riferisce alla 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 La *codifica* è la 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.

Committente Nell'ambito dell'ingegneria del software, il termine *committente* si riferisce alla persona, all'organizzazione o all'entità che richiede e finanzia lo sviluppo di un sistema software. Il committente è colui che ha un interesse diretto nel progetto e assume il ruolo di chi determina le aspettative per il prodotto software finale. Il committente svolge un ruolo chiave nel processo di sviluppo del software, in quanto fornisce la visione iniziale del progetto, identifica le esigenze degli utenti finali e stabilisce i criteri di successo.

Cruscotto Interfaccia grafica che fornisce una visualizzazione rapida e sintetica di dati, metriche e informazioni chiave relative a un progetto software.

Customer Acceptance (CA) Terza revisione di avanzamento del progetto didattico. Durante questa fase, il prodotto o servizio sviluppato viene presentato al cliente per la sua valutazione e accettazione. In questa fase, vengono valutati gli obiettivi raggiunti, le funzionalità implementate e la conformità alle specifiche concordate.

D

Database Un *database* è una collezione strutturata di dati progettata per consentire l'organizzazione, l'archiviazione e il recupero efficiente delle informazioni. I database sono ampiamente utilizzati in ambito informatico per gestire dati di vario tipo, come informazioni aziendali, dati utente, record finanziari e altro ancora.

Dataset Insieme strutturato di dati che può essere analizzato, elaborato e utilizzato per l'addestramento di modelli, la ricerca, l'analisi statistica e altri scopi.

Debugging Il debugging, in informatica, nell'ambito dello sviluppo software, indica l'attività che consiste nell'individuazione e correzione da parte del programmatore di uno o più errori rilevati nel software, direttamente in fase di programmazione oppure a seguito della fase di testing o dell'utilizzo finale del programma stesso.

Deployment In ingegneria del software, il termine *deployment* si riferisce al processo di distribuzione e installazione di un prodotto software in un ambiente operativo o in una piattaforma di produzione in modo che gli utenti finali possano utilizzarla. In altre parole, è il passaggio dallo sviluppo di un'applicazione al suo effettivo utilizzo da parte degli utenti.

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

Diagramma di Burndown Un *diagramma di Burndown* è uno 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.

Diagramma di Gantt Un *diagramma di Gantt* è uno 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 Un diagramma UML, acronimo di *Unified Modeling Language*, è un 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.

Dizionario dati File di descrizione della struttura e del contenuto del database. Nel caso di questo progetto, si tratta di un documento in formato JSON.

Docker Docker è un software che permette, mediante la virtualizzazione di processi a livello di sistema operativo, di fornire software in ambienti chiamati container.

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.

Embedded database Tipo di database progettato per essere integrato direttamente in un'applicazione software o in un dispositivo, piuttosto che essere eseguito come un'entità separata e autonoma, non necessitando di un'installazione separata o di una gestione esterna.

F

Feature In ambito informatico e tecnologico, il termine *feature* (caratteristica o funzionalità) si riferisce a un aspetto distintivo, a una capacità o a una funzione specifica di un prodotto software, di un'applicazione o di un dispositivo.

File di descrizione Sinonimo del "*dizionario dati*"

Fornitore In ingegneria del software, il *fornitore* è un'entità o a un'organizzazione che fornisce risorse, servizi, o componenti utilizzati nel processo di sviluppo del software.

Framework Infrastruttura software che fornisce un'architettura generale per lo sviluppo di applicazioni. Un *framework* è progettato per facilitare il processo di sviluppo fornendo strutture, librerie, linee guida e pattern comuni che possono essere seguiti dagli sviluppatori. L'obiettivo è semplificare il lavoro degli sviluppatori fornendo un ambiente predefinito in cui possono costruire le loro applicazioni.

G

Git È un sistema di controllo delle versioni distribuito che consente agli sviluppatori di tenere traccia delle modifiche al codice sorgente durante lo sviluppo del software. Git consente di creare *repository* che contengono la cronologia completa delle modifiche, consentendo agli sviluppatori di collaborare, coordinare e mantenere un registro delle versioni del proprio progetto.

GitHub Piattaforma di hosting per progetti software basati su Git. Essa fornisce un'interfaccia web e funzionalità di collaborazione che facilitano la gestione dei *repository* Git.

Glossario Un elenco organizzato di termini o vocaboli specializzati, spesso accompagnati dalle relative definizioni o spiegazioni. La creazione di un glossario facilita la comunicazione tra gli stakeholder, contribuendo a evitare ambiguità e promuovendo una comprensione comune dei termini utilizzati nel contesto specifico.

Granularità In ambito informatico, è una caratteristica che descrive il livello di dettaglio di un insieme di dati o delle componenti basi di un sistema.

H

Hugging Face Portale con la più grande raccolta di modelli LLM per i più svariati compiti.

I

Idioma Nel contesto dello sviluppo software, il termine *idioma* si riferisce a uno stile o una convenzione specifica utilizzata da programmatori per esprimere concetti o implementare soluzioni in un linguaggio di programmazione specifico. Gli idiomi sono spesso considerati come pratiche consigliate o modelli di programmazione comuni all'interno di una comunità di sviluppatori.

Indice Gulpease Indice di leggibilità utilizzato per valutare la complessità di un testo in lingua italiana. Esso prende in considerazione il numero delle frasi, il numero delle parole e il numero delle lettere presenti nel testo. Tale indice non tiene in considerazione aspetti quali la complessità del vocabolario e la struttura sintattica del testo.

Intelligenza Artificiale L'*intelligenza artificiale* è un campo dell'informatica che si occupa dello sviluppo di sistemi in grado di eseguire compiti che normalmente richiedono l'intelligenza umana. Questi compiti includono il riconoscimento di pattern, la comprensione del linguaggio naturale, la risoluzione di problemi complessi, l'apprendimento automatico e la capacità di adattarsi a nuove situazioni.

ISO/IEC 9126:2001 Serie di normative e linee guida preposte a descrivere un modello di qualità del software. Il modello propone un approccio alla qualità in modo tale che le società di software possano migliorare l'organizzazione e i processi e, quindi come conseguenza concreta, la qualità del prodotto sviluppato.

ISO/IEC 12207:1995 *ISO/IEC 12207:1995* è lo standard di riferimento per la gestione del ciclo di vita del software. Definisce tutte le attività svolte nel processo di sviluppo e mantenimento del software. Questo standard stabilisce un processo di ciclo di vita del software, compreso processi ed attività relative alle specifiche ed alla configurazione di un sistema.

ISO/IEC 31000:2018 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. 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. Essendo una linea guida, non è certificabile.

Issue Tracking System (ITS) Strumento software progettato per registrare, monitorare e gestire problemi o richieste di miglioramento all'interno di un progetto o di un'organizzazione. Questi problemi possono includere bug nel software, richieste di nuove funzionalità, attività di sviluppo, miglioramenti di processo e altro ancora.

J

Jira Jira è una piattaforma di gestione del lavoro e tracciamento dei progetti sviluppata da Atlassian. È ampiamente utilizzata per facilitare la gestione dei progetti, la collaborazione del team e la tracciabilità delle attività in vari settori, in particolare nello sviluppo del software e nella gestione dei progetti agili.

JSON (JavaScript Object Notation) È un formato di scambio dati leggero e basato su testo. JSON viene utilizzato solitamente per rappresentare dati strutturati, inoltre è indipendente dal linguaggio di programmazione.

L

Large Language Model (LLM) Il termine *large language model* si riferisce a modelli di linguaggio avanzati e complessi che sono stati addestrati su enormi quantità di dati testuali. Questi modelli, basati su tecniche di intelligenza artificiale come il *deep learning*, in grado di comprendere e generare testo in modo più sofisticato rispetto a modelli più piccoli.

M

Manuale Utente Si tratta di un documento contenenti le istruzioni per l'utilizzo e le funzionalità fornite dall'applicativo. L'utente sarà quindi a conoscenza dei requisiti minimi necessari per il corretto funzionamento dello stesso, di come installarlo in locale e di come farne un utilizzo consapevole.

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*.

Metadato Un metadato è un dato che descrive una qualche proprietà di un altro dato. In senso lato è un'informazione a proposito di un'altra informazione.

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, una *milestone* è un punto di riferimento o un traguardo significativo durante il ciclo di vita di un progetto. Le milestone rappresentano generalmente eventi chiave, compimenti o obiettivi importanti che indicano il progresso del progetto.

Modello a V Un approccio di sviluppo del software che visualizza il processo di sviluppo come una forma di "V", rappresentando le attività di progettazione e verifica. Le fasi sul lato sinistro della "V" rappresentano le attività di specifica e progettazione, mentre le fasi sul lato destro rappresentano le attività di verifica e validazione. Le fasi superiori del modello includono la definizione dei requisiti, la progettazione e la codifica, mentre le fasi inferiori rappresentano le attività di test, integrazione e manutenzione. Il Modello a V enfatizza la correlazione tra le fasi di sviluppo e di verifica, sottolineando l'importanza di testare e validare ciascuna fase prima di procedere alla successiva.

Model View Controller pattern (MVC) Un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti e in applicazioni web, in grado di separare la logica di presentazione dei dati

dalla logica di business. Questo pattern si posiziona nel livello logico o di business e di presentazione in una architettura multi-tier. Il componente centrale del MVC, il modello, cattura il comportamento dell'applicazione in termini di dominio del problema, indipendentemente dall'interfaccia utente (view). Il modello gestisce direttamente i dati, la logica e le regole dell'applicazione. Una View può essere una qualsiasi rappresentazione in output di informazioni, come un grafico o un diagramma. Sono possibili viste multiple delle stesse informazioni, come ad esempio un grafico a barre per la gestione e la vista tabellare per l'amministrazione. La terza parte, il Controller, accetta l'input e lo converte in comandi per il modello e/o la vista.

Minimum Viable Product (MVP) Versione di un prodotto che approssima quello atteso, dotato di caratteristiche e funzionalità appena sufficienti per essere utilizzabile dai primi clienti, i quali possono quindi fornire feedback per lo sviluppo futuro del prodotto stesso.

N

Norme di Progetto Le *Norme di Progetto* sono regole e linee guida stabilite all'interno di un progetto per garantire coerenza e qualità nelle attività svolte. Definiscono standard e procedure, come documentazione, gestione delle versioni e criteri di codifica, per assicurare uniformità nell'approccio e nel risultato finale.

P

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

Piano di Progetto Un 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 Il *Piano di Qualifica* è un 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 e efficaci.

Plan Do Check Act (PDCA) Il ciclo PDCA, noto anche come ciclo di Deming o ciclo di miglioramento continuo, è un modello di gestione e miglioramento continuo che si basa su quattro fasi chiave: Plan (Pianificare), Do (Eseguire), Check (Verificare) e Act (Agire). Questo approccio iterativo consente alle organizzazioni di adattarsi ai cambiamenti, risolvere i problemi in modo efficiente e migliorare costantemente le proprie operazioni e processi.

Processo Nell'ambito dell'ingegneria del software, un *processo* è una sequenza di attività strutturate che sono finalizzate a produrre un risultato specifico. I processi sono fondamentali per organizzare e guidare le attività coinvolte nello sviluppo del software in modo sistematico ed efficiente.

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

Proof of Concept (PoC) Letteralmente ”prova di concetto”, è una dimostrazione pratica per verificare la fattibilità o la validità di un'idea, di un concetto o di un progetto specifico. Nel contesto dello sviluppo software, un PoC potrebbe coinvolgere la creazione di una versione semplificata di un'applicazione per testare un nuovo approccio tecnologico o per dimostrare la fattibilità di una funzionalità specifica.

Progettazione L'attività di *progettazione* nel processo di sviluppo mira a definire l'architettura di un prodotto in grado di soddisfare le esigenze di tutti gli *stakeholder*. Questa fase fornisce documentazione dettagliata sulla struttura del prodotto, specifiche tecniche e scelte tecnologiche adottate per la sua realizzazione.

Prompt Nel contesto dei chatbot, il termine *prompt* si riferisce a un messaggio, una domanda o un input iniziale fornito al sistema per avviare o stimolare una risposta. In altre parole, il prompt è ciò che il chatbot utilizza come input iniziale per generare una risposta appropriata.

Proponente Nel contesto dell'ingegneria del software, il termine *proponente* si riferisce a colui che presenta un'idea, un progetto o una proposta e ne sostiene la realizzazione.

Q

Qualità del software Si riferisce alla misura in cui un sistema software soddisfa i requisiti specificati e le aspettative degli utenti. Questo concetto comprende diversi attributi, tra cui l'affidabilità, l'efficienza, la manutenibilità, l'usabilità e la sicurezza del software. La gestione della *qualità del software* coinvolge l'implementazione di processi e pratiche che mirano a garantire che il prodotto software sia conforme agli standard di qualità definiti e che soddisfi le necessità degli utenti finali.

Query Nel contesto dei database, una *query* è una richiesta o un'istanza di interrogazione formulata per ottenere informazioni specifiche dai dati archiviati. Le query sono ampiamente utilizzate nei database relazionali e sono scritte utilizzando linguaggi di query come SQL (*Structured Query Language*).

R

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

Requirement coverage La *requirement coverage* (copertura dei requisiti) è una misura utilizzata nel contesto del software testing per valutare quanto bene i requisiti specificati per un sistema software sono coperti dai test effettuati. In altre parole, indica in che misura i test case eseguiti durante il processo di testing affrontano e verificano gli obiettivi e le specifiche delineate nei requisiti del software.

Requirements and Technology Baseline (RTB) È 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).

Roadmap Una *roadmap* è una rappresentazione visuale di un piano o di attività pianificate nel tempo. Serve a comunicare in modo chiaro le tappe chiave, milestone e obiettivi lungo un percorso temporale. Spesso utilizzata in contesti aziendali o di sviluppo, aiuta a mantenere il focus e a coordinare il team verso un obiettivo comune.

S

Scenario alternativo Nel contesto dell'analisi dei requisiti, lo *scenario alternativo* rappresenta una sequenza di azioni che si verifica quando si verificano condizioni eccezionali o situazioni non standard durante l'interazione tra l'utente e il sistema. In altre parole, lo scenario alternativo si concentra su come il sistema dovrebbe gestire situazioni non previste o errori.

Scenario principale Nel contesto dell'analisi dei requisiti, lo *scenario principale* si riferisce a un percorso tipico o a una sequenza di azioni che rappresenta il flusso principale di interazioni tra un utente (o un sistema esterno) e il sistema che sta per essere sviluppato. Gli scenari principali sono utilizzati per documentare come gli utenti interagiranno con il sistema in situazioni standard.

Scrum Scrum è un framework per la gestione dei progetti che enfatizza il lavoro di squadra, la responsabilità e il progresso iterativo verso un obiettivo ben definito. Il framework inizia con una semplice premessa: iniziare con ciò che si può vedere o conoscere. Dopodiché, si deve tenere traccia dei progressi e modificarli, se necessario.

Sentence similarity La *similarità delle frasi* è un concetto utilizzato nell'ambito del trattamento automatico del linguaggio naturale (NLP) per misurare quanto due frasi siano simili o correlate dal punto di vista semantico. Tecniche e algoritmi di similarità delle frasi valutano la vicinanza di significato tra due sequenze di parole, considerando la semantica, la struttura grammaticale e il contesto delle frasi. Questa misura è fondamentale in applicazioni come il riconoscimento di intenti, il raggruppamento di testi simili e la traduzione automatica, contribuendo a migliorare la comprensione del significato intrinseco delle frasi da parte dei sistemi informatici.

Snake case Stile di denominazione che consiste nello scrivere le parole in minuscolo e separate da caratteri di *undesrcore* ("_").

Sottocaso d'uso Descrive un'azione o un flusso di lavoro specifico all'interno di un caso d'uso più generale. Può includere dettagli come le azioni degli attori, i passaggi necessari per completare un'attività e gli input e output coinvolti, e sono utili per suddividere un caso

d'uso complesso in unità più gestibili e comprensibili.

Specifica Tecnica Si tratta di un documento formale con lo scopo di servire da linea guida per gli sviluppatori che andranno ad estendere o mantenere il prodotto: al suo interno troverà tutte le informazioni riguardanti i linguaggi e le tecnologie utilizzate, l'architettura del sistema e le scelte progettuali effettuate per il prodotto.

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..

Sprint review Incontro tra il team di sviluppo, il Product Owner, il management e/o i committenti per presentare lo stato dello sprint e confrontarlo con l'impegno assunto all'inizio dei lavori.

SQL Acronimo di *Structured Query Language*, è un linguaggio di programmazione specializzato progettato per la gestione e l'interrogazione dei database relazionali.

Stakeholder Individui, gruppi o entità che hanno un interesse o un coinvolgimento in un progetto, un'organizzazione o un'iniziativa specifica. Essi possono influenzare o essere influenzati dalle attività e dalle decisioni associate al progetto o all'organizzazione. La gestione degli stakeholder è una parte fondamentale del processo di pianificazione e esecuzione di progetti ed è essenziale per il successo complessivo di un'iniziativa.

Streamlit Libreria *open-source* di Python che facilita la creazione e lo sviluppo di applicazione web personalizzate, ideale per progetti di *data science* e *machine learning*.

Structural coverage La *structural coverage* (copertura strutturale) è una misura utilizzata nel contesto del software testing per valutare la completezza dei test eseguiti rispetto alla struttura interna del codice sorgente.

T

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

Test case Nel contesto dell'ingegneria del software, un *test case* è un insieme di condizioni o variabili sotto le quali un tester determina se un'applicazione o sistema software risponde correttamente o meno. Il meccanismo per determinare se un programma software o un sistema ha superato un test è conosciuto come *oracolo di test*.

Test di accettazione È l'ultima fase del processo di verifica del software, in cui il sistema viene valutato per determinare se soddisfa i criteri di accettazione definiti dagli utenti o dagli stakeholder. L'obiettivo principale è confermare che il sistema sia pronto per essere consegnato e utilizzato in un ambiente operativo reale.

Test di funzionalità È una fase di verifica nel processo di sviluppo del software in cui vengono valutate le funzionalità specifiche di un sistema. L'obiettivo principale è assicurare che il software esegua le operazioni previste in modo corretto e conforme ai requisiti specificati. Durante il test di funzionalità, vengono eseguiti scenari di prova progettati per coprire le diverse funzionalità del software. Gli errori, le discrepanze o le deviazioni dalle specifiche vengono identificati e registrati per essere corretti. Questo tipo di test contribuisce a garantire che il prodotto software soddisfi le aspettative degli utenti e funzioni come previsto, migliorando così la qualità complessiva del sistema.

Test di integrazione Il *test di integrazione* è una fase del processo di testing software in cui i singoli moduli o componenti di un'applicazione vengono combinati e testati come un gruppo. L'obiettivo è verificare che le diverse parti del sistema interagiscano correttamente e che l'intera applicazione funzioni come previsto. Questo tipo di test è cruciale per identificare e risolvere problemi di interoperabilità tra le componenti e garantire la coerenza del sistema nel suo complesso.

Test di regressione Il *test di regressione* è un tipo di test software mirato a garantire che le modifiche apportate a una parte del codice non abbiano effetti indesiderati sulle funzionalità

esistenti del software. Dopo ogni modifica o aggiornamento, il test di regressione viene eseguito per assicurare che le funzionalità precedentemente testate rimangano intatte. Ciò aiuta a prevenire il verificarsi di nuovi bug o problemi a seguito delle modifiche apportate durante lo sviluppo del software.

Test di sistema È una fase critica del processo di verifica del software, mirato a valutare il comportamento integrato di un sistema completo. In questa fase, il software viene testato nel suo ambiente operativo, verificando che tutti i componenti funzionino correttamente insieme. Gli obiettivi principali includono la validazione delle funzionalità del sistema, la conformità ai requisiti specificati e la verifica delle prestazioni, della sicurezza e della robustezza complessiva. I test di sistema sono progettati per simulare condizioni del mondo reale e identificare eventuali problemi di integrazione o errori sistematici che potrebbero emergere solo quando tutte le parti del sistema interagiscono tra loro.

Test di unità È una pratica dell'ingegneria del software che aiuta a garantire la qualità e l'affidabilità del codice attraverso la verifica delle singole unità. Tali unità si riferiscono alle parti più piccole e atomiche del software, come funzioni, metodi, classi o moduli. L'obiettivo principale del test di unità è isolare e testare ogni singola unità di codice in modo indipendente dagli altri componenti del sistema. Questo consente agli sviluppatori di individuare e risolvere eventuali difetti o bug in modo più efficiente e tempestivo.

Timeline In italiano "linea del tempo", è una rappresentazione visuale di eventi disposti in ordine cronologico. Questo strumento grafico consente di visualizzare e comprendere facilmente la successione temporale di eventi, attività o cambiamenti nel corso del tempo. Nei progetti software, le timeline sono utili per pianificare e monitorare le attività del progetto nel tempo.

U

UML acronimo di ”Unified Modeling Language”, è un linguaggio di modellazione visuale ampiamente utilizzato nel campo dell'ingegneria del software per progettare e descrivere sistemi software. È un linguaggio standardizzato che fornisce una notazione grafica per rappresentare i diversi aspetti di un sistema, dalla progettazione concettuale alla realizzazione fisica.

V

Validazione La *validazione* è il processo di conferma che un prodotto o sistema soddisfi i requisiti specificati e risponda alle esigenze dell'utente.

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.

Vettore Rappresentazione di una parola come un vettore di numeri reali, dove il valore di ogni numero dipende dalle informazioni sia semantiche che sintattiche della parola.

W

Walkthrough Nel contesto dello sviluppo software, un *walkthrough* è un approfondito esame e revisione di un sistema o di un processo. Coinvolge una valutazione dettagliata per individuare errori, problemi di progettazione o miglioramenti potenziali. A differenza di altre revisioni più mirate, il walkthrough è una panoramica completa e metodica, spesso coinvolgendo diversi partecipanti con competenze specifiche per garantire una revisione accurata e completa.

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.

Web app Una *web app*, o applicazione web, è un'applicazione software che viene eseguita su un browser web. A differenza delle applicazioni tradizionali che devono essere scaricate e installate localmente su un dispositivo, le web app sono accessibili attraverso un browser internet e possono essere utilizzate su diverse piattaforme e dispositivi.

Z

Zero-shot classification Si tratta di una tecnica di classificazione che permette alla macchina di riconoscere e classificare oggetti o concetti mai visti prima, basandosi su descrizioni o relazioni semantiche anziché su esempi specifici.

Zoom È una piattaforma *cloud-based* con software *peer-to-peer* che offre servizi di videotelefonia e chat online. È utilizzato per teleconferenze, telelavoro, formazione a distanza e relazioni sociali.

Zucchetti S.p.A Zucchetti S.p.A. è un'azienda italiana che produce soluzioni software, hardware e servizi per aziende, banche, assicurazioni, professionisti e associazioni di categoria.