



# GLOSSARIO

SHOWROOM3D

 [SmokingFingertips](#) |  [smoking.fingertips@gmail.com](mailto:smoking.fingertips@gmail.com)

<b>Versione</b>	1.0.0
<b>Stato</b>	approvato
<b>Uso</b>	esterno
<b>Responsabile</b>	Sebastien Biollo
<b>Redattori</b>	Edoardo Gasparini Luca Polese Davide Baggio Luca Annicchiarico Alberto Angeloni
<b>Verificatori</b>	Sebastien Biollo Gabriele Saracco
<b>Destinatari</b>	<i>Smoking Fingertips</i> Prof. Tullio Vardanega Prof. Riccardo Cardin <i>Sanmarco Informatica S.p.A</i>
<b>Data di Approvazione</b>	2023-02-25
<b>Anno accademico:</b>	2022/2023

## Sommario:

Questo documento racchiude le definizioni dei termini utilizzati nella documentazione del progetto ShowRoom3D, al fine di chiarire possibili ambiguità.

---

## Storico delle Modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	2023-02-25	Sebastien Biollo	Responsabile	Approvazione del documento
0.2.0	2023-02-23	Gabriele Saracco	Verificatore	Verifica di tutti i termini
0.1.2	2023-01-20	Davide Baggio	Amministratore	Stesura termini
0.1.1	2023-01-12	Luca Annicchiarico	Amministratore	Stesura termini
0.1.0	2022-12-22	Sebastien Biollo	Verificatore	Verifica dei termini stesi fin'ora
0.0.3	2022-12-18	Alberto Angeloni	Amministratore	Stesura termini
0.0.2	2022-12-10	Edoardo Gasparini	Amministratore	Stesura termini
0.0.1	2022-12-02	Luca Polese	Amministratore	Stesura termini

## Indice

<b>A</b>	<b>3</b>
<b>B</b>	<b>4</b>
<b>C</b>	<b>6</b>
<b>D</b>	<b>8</b>
<b>E</b>	<b>9</b>
<b>F</b>	<b>10</b>
<b>G</b>	<b>11</b>
<b>H</b>	<b>12</b>
<b>I</b>	<b>14</b>
<b>J</b>	<b>15</b>
<b>L</b>	<b>16</b>
<b>M</b>	<b>18</b>
<b>N</b>	<b>19</b>
<b>P</b>	<b>21</b>
<b>R</b>	<b>22</b>
<b>S</b>	<b>23</b>
<b>T</b>	<b>25</b>
<b>U</b>	<b>26</b>
<b>V</b>	<b>27</b>
<b>W</b>	<b>28</b>

## A

**API** • (o *Application Programming Interface*) si riferisce ad un set di procedure organizzate in gruppi e finalizzate all'esecuzione di una determinata funzione. Spesso è utilizzato per indicare delle librerie software di un determinato linguaggio di programmazione.

**Asset** • oggetti tridimensionali che vengono utilizzati per creare un'esperienza visiva o interattiva. Possono includere modelli di oggetti, personaggi, ambienti, *texture*, animazioni e così via. In genere, gli asset 3D vengono creati utilizzando software di modellazione e animazione e possono essere esportati in vari formati.

**Attività** • Secondo lo standard ISO 12207:1995, un'attività di ingegneria del software è definita come un'operazione, un'insieme di operazioni o una fase che è necessaria per soddisfare uno o più obiettivi. Tali obiettivi possono includere la corretta comprensione delle richieste dell'utente, la progettazione, la codifica, la verifica, la documentazione, il rilascio, la manutenzione, ecc.

## B

**Baseline** • Punto saldo raggiunto e che fa da "base". Si riferisce a una versione specifica di un prodotto software, che viene stabilita come punto di riferimento per il futuro sviluppo e la gestione del software stesso.

**Blueprint** • È un sistema visuale di programmazione che permette agli sviluppatori di creare e modificare la logica di gioco senza dover scrivere codice. Permette di creare "grafi di flusso" che rappresentano la logica del gioco in modo visuale, attraverso l'uso di blocchi (chiamati nodi) che rappresentano azioni e condizioni.

**Board** • È una sorta di "bacheca digitale" che consente di organizzare le attività in modo visuale, suddividendole in colonne che rappresentano i vari stati di avanzamento. Le colonne tipiche di una board di Jira includono:

- *To Do*: le attività da completare
- *In Progress*: le attività in corso di svolgimento
- *Done*: le attività completate

Oltre a queste colonne, la board di Jira può essere personalizzata per soddisfare le esigenze specifiche della squadra o del progetto in corso. Ad esempio, aggiungendo delle colonne aggiuntive.

**Brain storming** • È una tecnica di gruppo utilizzata per generare idee creative e innovative su un determinato argomento o problema. La tecnica funziona in questa maniera: i partecipanti si riuniscono in un ambiente informale e sotto la guida di un moderatore, cominciano a generare idee, cercando di ampliare il campo di possibilità senza limiti o pregiudizi incoraggiati a costruire sulle idee degli altri aggiungendo o combinando per creare soluzioni innovative.

**Branch** • Una branch rappresenta una linea di sviluppo separata e indipendente all'interno di un repository e consente a più componenti del gruppo di lavorare contemporaneamente sullo stesso progetto evitando le interferenze prodotte dal lavoro in parallelo.

## C

**C++** • un linguaggio di programmazione general purpose sviluppato come evoluzione del linguaggio C inserendo la programmazione orientata agli oggetti; col tempo ha avuto notevoli evoluzioni, come l'introduzione dell'astrazione rispetto al tipo.

**Capitolato** • Il documento tecnico redatto dal cliente rappresenta una specificazione dei vincoli contrattuali, tra cui il prezzo e le scadenze, per lo sviluppo di un determinato prodotto software. Questo documento viene solitamente presentato all'interno di un bando d'appalto per cercare un fornitore in grado di svolgere il lavoro richiesto.

**Cascata, modello a** • paradigma più antico e diffuso nell'Ingegneria del Software. Propone un approccio sequenziale e sistematico allo sviluppo del software, con fasi ben definite come la comunicazione, la pianificazione, la modellazione, la costruzione e il *deployment<sub>G</sub>*. Durante la fase di comunicazione, vengono raccolti i requisiti del cliente, seguita dalla fase di pianificazione in cui vengono stimati i costi e stabiliti i controlli per monitorare il progetto. Successivamente, durante la fase di modellazione, i requisiti sono analizzati e il software viene progettato, spesso con l'uso di diagrammi *UML<sub>G</sub>*. La fase di costruzione segue con la programmazione vera e propria, seguita dal collaudo del prodotto e infine la consegna del prodotto finito al cliente nella fase di *deployment<sub>G</sub>*, che può essere seguita da una fase di supporto. Nonostante la sua semplicità, il modello a cascata viene considerato superato e non applicabile ai problemi che presentano i progetti odierni nell'Ingegneria del Software. Viene spesso adottato da team poco esperti che hanno bisogno di un punto di partenza per organizzare il proprio flusso di lavoro.

**Ciclo di Deming** • vedi *PDCA<sub>G</sub>*

**Caso d'uso** • I casi d'uso sono un insieme di sequenze di azioni o scenari che hanno uno scopo comune, mirando a descrivere le funzionalità di un sistema dal punto di vista degli utenti, senza fornire dettagli implementativi. Il documento dei casi d'uso definisce e descrive i diversi scenari che possono essere eseguiti dagli attori che interagiscono con il sistema, sia che si tratti di utenti umani che di sistemi informativi esterni. I casi d'uso hanno lo scopo di permettere agli attori principali di raggiungere un obiettivo significativo, fornendo una panoramica completa delle funzionalità del sistema e dei flussi di lavoro. In altre parole, i casi d'uso rappresentano una descrizione degli scenari di utilizzo del sistema, in cui si specificano gli input, le attività svolte dal sistema e le risposte prodotte. L'insieme di tutti i casi d'uso rappresenta una vista globale delle funzionalità del sistema e viene utilizzato come base per la progettazione e lo sviluppo del software.

**Ciclo di vita del software** • serie di fasi attraverso le quali un prodotto software passa dal suo concepimento fino al suo ritiro. Queste fasi includono la pianificazione, la progettazione, lo sviluppo, il testing, l'implementazione, la manutenzione e il ritiro. Il ciclo di vita del software viene utilizzato per gestire il processo di sviluppo del software, fornendo un quadro strutturato per la pianificazione e l'esecuzione delle attività necessarie per la

realizzazione del prodotto software. Ogni fase del ciclo di vita del software ha obiettivi e attività specifiche che devono essere completate per avanzare alla fase successiva.

**code coverage** • è una metrica di analisi del software utilizzata nel testing per misurare l'estensione in cui il codice sorgente di un programma viene eseguito durante l'esecuzione dei test. Questo tipo di metrica viene utilizzata per valutare l'efficacia dei test effettuati e per individuare eventuali parti del codice che non vengono testate e che potrebbero nascondere dei difetti.

**committente** • Il Committente, in campo commerciale o contrattuale, è la persona o l'entità che commissiona o ordina la realizzazione di un prodotto o servizio. In ambito dell'Ingegneria del Software, il Committente è colui che richiede lo sviluppo di un software specifico, definendone i requisiti e i vincoli di progetto.

**competitor** • In ambito economico e politico, il termine "competitor" indica un'entità o un'organizzazione che opera nello stesso mercato o settore e rappresenta un concorrente o un avversario. Questo termine è spesso usato al plurale come "competitors".

**compito** • Specifica attività o responsabilità assegnata a una persona o a un gruppo di persone, che deve essere completata entro un determinato periodo di tempo e in base a specifiche istruzioni o obiettivi. I compiti sono spesso considerati come l'elemento fondamentale del lavoro e possono essere suddivisi in compiti più piccoli o in attività più ampie. Il termine "compito" viene anche utilizzato al plurale per indicare un insieme di attività da svolgere.

**Cruscotto** • Nel contesto dell'informatica e del web, è una schermata grafica che presenta in modo sintetico e immediato le informazioni più importanti e significative relative ad un determinato sistema, servizio o applicazione. Il cruscotto è progettato per fornire una visione d'insieme del sistema, permettendo all'utente di comprendere rapidamente lo stato delle attività in corso, il grado di raggiungimento degli obiettivi, l'andamento di indicatori chiave di performance e così via. Il cruscotto è spesso utilizzato in ambito aziendale e manageriale, ma può essere presente anche in applicazioni di vario genere, come ad esempio i software di monitoraggio di sistemi informatici o di analisi dei dati.

**Customer Acceptance** • Processo attraverso il quale il cliente o l'utilizzatore finale verifica che un prodotto o servizio soddisfi i requisiti e le aspettative concordati con il fornitore. Tale processo avviene generalmente alla fine del ciclo di sviluppo del prodotto o del servizio e può includere test, ispezioni, valutazioni e verifiche da parte del cliente. L'accettazione del cliente è un passaggio critico per il successo del prodotto o del servizio poiché il cliente deve essere soddisfatto del prodotto prima di utilizzarlo e prima di pagare per esso.

## D

**Deployment** • É il processo di distribuzione di un'applicazione o di un sistema software in un ambiente di produzione dopo che è stata sviluppata, testata e validata. Questo processo implica la preparazione, l'installazione, la configurazione e la distribuzione di tutti i componenti software e hardware necessari per far funzionare l'applicazione o il sistema in modo efficiente e affidabile nell'ambiente di produzione. Il deployment può essere eseguito manualmente o in modo automatico utilizzando strumenti appositi, come ad esempio gli strumenti di Continuous Delivery.

**Design Pattern** • Una soluzione progettuale standard per risolvere un problema comune in modo efficace e riutilizzabile. Un design pattern è un modello logico descritto per la risoluzione di problemi ricorrenti durante la fase di progettazione e sviluppo del software. L'adozione di design pattern può aiutare a migliorare la qualità del software e a ridurre il debito tecnico.

**Diagramma di Gantt** • Strumento utilizzato nella gestione dei progetti per pianificare e visualizzare le attività in un determinato intervallo di tempo. e' composto da una serie di barre orizzontali che rappresentano le attività del progetto e dalle relative durate. Il diagramma di Gantt mostra anche le relazioni tra le attività, permettendo di identificare le dipendenze tra di esse. In questo modo, i membri del team del progetto possono capire facilmente quali attività devono essere completate prima che inizino altre attività e quale sia lo stato di avanzamento del progetto nel complesso.

**Diagrammi UML** • Sono un linguaggio visuale di modellizzazione del software usato per descrivere, progettare e documentare sistemi software. Questi diagrammi utilizzano una serie di simboli grafici per rappresentare diversi aspetti del sistema, tra cui classi, oggetti, componenti, attività, sequenze di eventi e relazioni tra di essi. I diagrammi UML (Unified Modeling Language) sono comunemente utilizzati durante il processo di sviluppo del software per comunicare e documentare le decisioni di progettazione e le scelte implementative del team di sviluppo.

**DirectX** • E'una collezione di API (Application Programming Interface) sviluppate da Microsoft per semplificare la programmazione di applicazioni multimediali, in particolare per il sistema operativo Windows. DirectX offre funzionalità per la gestione di grafica 2D e 3D, audio e input da dispositivi di gioco come mouse, tastiere e gamepad. Le applicazioni che utilizzano DirectX possono essere eseguite su diverse versioni di Windows e su hardware con diversi livelli di prestazioni, grazie alla capacità di DirectX di adattarsi alle specifiche delle diverse schede video e audio presenti sui diversi dispositivi.

**Discord** • Applicazione gratuita che consente agli utenti di chattare, comunicare tramite microfono e videochiamata a gruppi. Offre funzionalità utili agli utenti, come la comunicazione in tempo reale e un forum. Discord è accessibile da smartphone, tablet e PC.





## E

**Economicità** • Si riferisce alla capacità di produrre risultati desiderati utilizzando il minor numero possibile di risorse o costi. In altre parole, l'economicità si riferisce all'efficienza con cui vengono utilizzate le risorse per raggiungere un obiettivo specifico.

**Efficacia** • Il grado di conformità del prodotto è la misura della sua capacità di raggiungere gli obiettivi prefissati

**Efficenza** • Capacità di mantenere un livello di produttività costante utilizzando il minimo numero possibile di risorse.

## F

**Fase** • In informatica, una fase rappresenta una porzione di lavoro o di attività che viene svolta in modo sequenziale o parallelo durante lo sviluppo di un progetto. Ogni fase è caratterizzata da specifiche attività che devono essere eseguite per raggiungere uno specifico obiettivo. Le fasi possono variare in base alla metodologia di sviluppo adottata e possono includere la pianificazione, l'analisi dei requisiti, la progettazione, lo sviluppo, il testing e la manutenzione.

**Fornitore** • La persona, il team o l'azienda responsabile della realizzazione del prodotto richiesto dal proponente è definito come fornitore. Questo termine può essere utilizzato anche al plurale per riferirsi a più persone, team o aziende coinvolte nella realizzazione del prodotto.

**Framework** • Un framework è un insieme di strumenti, librerie e linee guida utilizzate per sviluppare applicazioni software. In pratica, si tratta di un'infrastruttura predefinita che offre un set di funzionalità comuni, facilitando il lavoro del programmatore. Un framework può includere librerie per la gestione della grafica, per la gestione dei dati, per la comunicazione tra i vari componenti dell'applicazione, e molto altro ancora. L'utilizzo di un framework consente di ridurre il tempo di sviluppo, migliorare l'affidabilità dell'applicazione e mantenere una maggiore coerenza tra il codice scritto dai vari sviluppatori.

## G

**Gantt Project** • E' un'applicazione desktop open source per la gestione di progetti che consente di pianificare e organizzare le attività attraverso l'uso di diagrammi di Gantt. Fornisce una serie di strumenti per la gestione delle risorse, la creazione di rapporti, la gestione dei costi e il monitoraggio dello stato del progetto. È in grado di esportare i diagrammi di Gantt in vari formati, tra cui PDF, HTML, PNG e CSV. GanttProject è disponibile per diverse piattaforme, tra cui Windows, Mac OS X e Linux.

**Git** • E' un sistema di controllo versione distribuito, utilizzato principalmente per gestire il codice sorgente durante lo sviluppo del software. Permette ai team di sviluppo di lavorare contemporaneamente sullo stesso progetto, tenere traccia delle modifiche, gestire i conflitti e riunire le modifiche in un'unica versione del progetto. Git è open source e supporta molte funzionalità come branch e merge, stashing, cherry-pick, e la possibilità di lavorare con repository remoto su server.

**Github** • Il servizio di hosting per progetti software che implementa lo strumento di controllo versione distribuito Git, consentendo agli utenti di gestire, collaborare e condividere il codice sorgente dei propri progetti in modo efficiente e sicuro attraverso il web

**Google Calendar** • Sistema che consente di creare, gestire e condividere più calendari. È anche possibile importare calendari da altri servizi online o dal computer e integrare Google Calendar con gli altri servizi Google grazie all'account Google.

**Google Docs** • Suite per ufficio gratuita basata su Web che include documenti, fogli elettronici e presentazioni. Consente agli utenti di creare, modificare e collaborare in tempo reale sui documenti online, che possono essere importati o inviati tramite e-mail.

**Google Drive** • Servizio Web introdotto da Google che consente di memorizzare e sincronizzare file online. Può essere utilizzato tramite Web o mediante un'applicazione installata sul computer. Inoltre, consente di archiviare anche i documenti creati con Google Docs.

**Google Meet** • Servizio di videoconferenza e comunicazione online sviluppato da Google. Consente a utenti e gruppi di comunicare in tempo reale tramite video, audio e messaggi di testo su diverse piattaforme come PC, tablet o smartphone. Il servizio è utilizzato sia per scopi personali che per attività professionali come riunioni aziendali, lezioni online e collaborazioni a distanza. Google Meet è disponibile come applicazione web e come app per dispositivi mobili.

**Google** • Google Inc ha sviluppato un motore di ricerca per Internet che indicizza e cataloga le risorse online, tra cui foto, newsgroup, mappe, email, shopping, traduzioni, video e programmi specifici.

## H

**Hosting** • Servizio che consiste nel fornire spazio su un server web per memorizzare i file di un sito web, rendendolo accessibile online. Questo servizio può includere anche altri servizi aggiuntivi come la gestione del dominio, la gestione delle email e la sicurezza del sito web

## I

**ISO/IEC 12207:1995** • E'uno standard internazionale per il ciclo di vita del software che descrive un insieme di processi comuni al software. Il modello di ciclo di vita del software descritto in ISO/IEC 12207:1995 comprende un insieme di processi che definiscono attività come l'analisi dei requisiti, la progettazione, la codifica, la verifica e la manutenzione del software. Lo standard definisce anche una serie di attività di supporto, come la gestione della configurazione, la gestione della qualità e la gestione del progetto, che devono essere eseguite durante tutto il ciclo di vita del software.

è uno standard internazionale per il ciclo di vita del software che descrive un insieme di processi comuni al software. Il modello di ciclo di vita del software descritto in ISO/IEC 12207:1995 comprende un insieme di processi che definiscono attività come l'analisi dei requisiti, la progettazione, la codifica, la verifica e la manutenzione del software. Lo standard definisce anche una serie di attività di supporto, come la gestione della configurazione, la gestione della qualità e la gestione del progetto, che devono essere eseguite durante tutto il ciclo di vita del software.

**ISO/8601** • E'uno standard internazionale che stabilisce il formato per la rappresentazione di date, orari e intervalli di tempo. Questo standard definisce un formato di data e ora facilmente leggibile e comprensibile a livello internazionale, che consente una maggiore interoperabilità tra i sistemi informatici di diversi paesi. Il formato standard di ISO 8601 è "YYYY-MM-DDTHH:MM:SS", dove "YYYY" rappresenta l'anno, "MM" il mese, "DD" il giorno, "T" separa la data dall'ora, "HH" l'ora, "MM" il minuto e "SS" il secondo.

**ISO/IEC 31000:2009** • E'uno standard internazionale che definisce i principi e le linee guida per la gestione del rischio. Si applica a qualsiasi tipo di organizzazione e a qualsiasi tipo di rischio, sia esso finanziario, operativo, tecnologico o di altro tipo. Lo standard fornisce un quadro di riferimento per la gestione del rischio che consente alle organizzazioni di identificare, valutare e gestire i rischi in modo sistematico ed efficace.

**ISO/IEC 9126** • Le normative e le linee guida sviluppate dall'Organizzazione internazionale per la normazione (ISO) in collaborazione con la Commissione Elettrotecnica Internazionale (IEC) costituiscono un insieme di standard che descrivono un modello di qualità del software.

**Issue Tracking System** • o ITS e' uno strumento software utilizzato per gestire e tracciare le segnalazioni di problemi, i bug e le richieste di funzionalità in un progetto software. L'ITS fornisce una piattaforma centralizzata per la gestione del ciclo di vita dei problemi, dalla segnalazione iniziale alla risoluzione finale, consentendo al team di sviluppo di collaborare e monitorare lo stato di ogni problema. I vantaggi dell'utilizzo di un ITS includono una maggiore trasparenza, una migliore organizzazione e una maggiore efficienza nella risoluzione dei problemi.

**iterazione** • strumento software utilizzato per gestire e tracciare le segnalazioni di problemi, i bug e le richieste di funzionalità in un progetto software. L'ITS fornisce una piattaforma centralizzata per la gestione del ciclo di vita dei problemi, dalla segnalazione iniziale alla risoluzione finale, consentendo al team di sviluppo di collaborare e monitorare lo stato di ogni problema. I vantaggi dell'utilizzo di un ITS includono una maggiore trasparenza, una migliore organizzazione e una maggiore efficienza nella risoluzione dei problemi.

**Issue** • GitHub<sub>G</sub> offre la funzionalità Issue per tenere traccia di idee, miglioramenti, compiti o errori su cui lavorare. Gli utenti possono assegnare gli issues ad altri utenti, etichettarli per una ricerca più rapida e raggrupparli.

## J

**JavaScript** • Linguaggio di scripting orientato agli oggetti e agli eventi è una tecnologia utilizzata nella programmazione Web sia lato client che lato server.

**Jira** • Strumento di gestione dei progetti e problemi sviluppato da Atlassian. Consente di tenere traccia dei problemi, dei bug e delle attività del progetto in modo organizzato e collaborativo. Jira è utilizzato principalmente in ambito software development, ma può essere adattato anche ad altre aree di lavoro come la gestione di progetti e attività in generale. Tra le sue funzionalità, Jira offre la possibilità di creare, assegnare, tracciare e risolvere problemi, creare e visualizzare report, gestire il workflow del progetto e integrarsi con altre applicazioni e strumenti.



## L

**Latex workshop** • Estensione di Visual Studio Code utilizzata per scrivere e compilare documenti LaTeX. Offre diverse funzionalità, come l'auto completamento del codice LaTeX, la visualizzazione del documento in tempo reale e la possibilità di utilizzare diversi compilatori. La sua interfaccia utente intuitiva e le numerose opzioni di personalizzazione lo rendono uno strumento popolare tra gli utenti che lavorano con LaTeX.

**Linux** • Sistema operativo basato sul kernel Linux, sviluppato nel 1991 da Linus Torvalds come software libero e open source. Linux è una famiglia di sistemi operativi di tipo Unix-like, pubblicati sotto varie distribuzioni, con caratteristiche comuni come l'utilizzo del kernel Linux. Il software del progetto GNU è spesso incluso nelle distribuzioni Linux, rendendo il sistema operativo completamente libero e gratuito. Linux è utilizzato da molte aziende leader nel campo dell'informatica come Google, IBM e Oracle Corporation, che sviluppano e pubblicano sistemi basati su Linux.

## M

**Mantenibile** • Si riferisce alla capacità di un software di essere facilmente modificato e aggiornato nel tempo, in modo da poter mantenere o migliorare le sue funzionalità. Un software che è considerato "mantenibile" è caratterizzato da una struttura modulare e da un codice ben organizzato, che lo rendono facile da comprendere, modificare e correggere in caso di errori o esigenze di cambiamento. Inoltre, un software mantenibile dovrebbe essere in grado di adattarsi alle esigenze dell'utente senza richiedere una completa riscrittura del codice sorgente. La manutenibilità del software è un aspetto importante della sua qualità complessiva e contribuisce alla riduzione dei costi di sviluppo e di manutenzione nel lungo termine.

**Merge** • Operazione che avviene in ambiente di sviluppo software, in particolare in sistemi di controllo versione come Github. Consente di unire le modifiche apportate in un ramo (branch) del codice sorgente con un altro ramo, come ad esempio il ramo principale (master). In questo modo, le modifiche effettuate da più sviluppatori su diverse copie del codice sorgente possono essere integrate in modo coerente e senza conflitti. Il Merge può essere effettuato manualmente o automaticamente a seconda delle impostazioni del sistema di controllo versione utilizzato.

**Metrica** • Serie di regole o criteri per la misurazione di determinate proprietà o caratteristiche del software o delle sue specifiche. Questi criteri, chiamati anche "metriche", sono utilizzati per valutare e monitorare la qualità del software durante il processo di sviluppo e manutenzione.

**Microsoft** • Azienda multinazionale americana di tecnologia con sede a Redmond, nello stato di Washington. Fondata nel 1975 da Bill Gates e Paul Allen, Microsoft è diventata una delle società più grandi e influenti al mondo nel campo del software e dei servizi informatici. Tra i suoi prodotti più noti ci sono il sistema operativo Windows, la suite di produttività Office e il motore di ricerca Bing. Microsoft ha anche investito in diversi settori, tra cui videogiochi, dispositivi mobili e cloud computing.

**Miglioramento** • Nel contesto del project management, si riferisce a un processo continuo di identificazione, analisi e implementazione di azioni che mirano a migliorare le prestazioni del progetto e raggiungere i suoi obiettivi in modo più efficiente ed efficace. Tale processo può coinvolgere sia la gestione del progetto che la gestione delle attività operative del progetto stesso, con l'obiettivo di aumentare la qualità, ridurre i costi, ottimizzare le risorse, accelerare i tempi di consegna o migliorare altri aspetti rilevanti per il successo del progetto. Il miglioramento può essere guidato da metriche e indicatori di prestazione, che consentono di monitorare il progresso e identificare le aree in cui intervenire per ottenere i miglioramenti desiderati.

**Milestone** • Letteralmente "pietra miliare" in inglese, indica un traguardo importante da raggiungere durante lo svolgimento di un progetto. Solitamente è associata a una data

di scadenza e rappresenta un obiettivo chiave che permette di verificare se il progetto sta procedendo come previsto. Una volta raggiunta la milestone, è possibile valutare il lavoro svolto fino a quel momento, eventualmente apportare modifiche e definire le attività necessarie per il raggiungimento delle milestone successive.

**Modello agile** • Approccio di sviluppo software iterativo e incrementale, basato sulla collaborazione tra il team di sviluppo e il cliente o l'utente finale. L'obiettivo principale del modello Agile è quello di consegnare un prodotto funzionante e di alta qualità in modo rapido e flessibile, attraverso il lavoro in team, la comunicazione continua e la risposta alle modifiche dei requisiti del progetto. Gli esempi di metodologie Agile includono Scrum, Kanban e XP (Extreme Programming).

**Modello incrementale** • Modello di sviluppo del software in cui il prodotto finale viene sviluppato attraverso una serie di incrementi funzionali. In ogni incremento, una nuova funzionalità viene aggiunta al software, creando una versione migliorata del prodotto. Questo processo continua fino a quando il software raggiunge la sua forma finale. Il modello incrementale è caratterizzato da un approccio iterativo e si concentra sulla consegna continua di funzionalità utili per il cliente, consentendo di rispondere in modo flessibile ai cambiamenti nei requisiti durante il processo di sviluppo. Le prime funzionalità implementate solitamente sono quelle più importanti, in questo modo, con i successivi incrementi queste funzionalità tendono a stabilizzarsi per prime.

## N

**Needs** • Nell'ambito del project management, "needs" (necessità o "esigenze") si riferisce ai requisiti o alle richieste che un cliente o un utente finale ha per un prodotto o un servizio. Identificare e soddisfare le esigenze del cliente è un aspetto fondamentale del processo di gestione del progetto.

**Norme** • Insieme di regole, linee guida o standard che stabiliscono come comportarsi in un determinato contesto o settore. In ambito informatico, le norme possono riguardare ad esempio la sicurezza informatica, la gestione dei dati personali, lo sviluppo del software, la qualità dei processi produttivi e così via. Le norme possono essere stabilite da organizzazioni internazionali, governative, o da organismi di settore, e hanno lo scopo di garantire la sicurezza, la qualità, l'efficienza e l'efficacia delle attività svolte.

## P

**Package** • (in italiano *pacchetto*) è un insieme di file, contenenti istruzioni che forniscono funzionalità aggiuntive a L<sup>A</sup>T<sub>E</sub>X, come nuovi comandi, nuove funzioni, nuovi stili e nuovi ambienti. Possono essere usati per modificare il layout di un documento o aggiungere funzionalità (come tabelle e figure). Sono forniti con l'installazione di L<sup>A</sup>T<sub>E</sub>X, ma è anche possibile scaricare e installare dei package da repository online.

**PascalCase** • E' una convenzione di denominazione di programmazione in cui la prima lettera di ogni parola nel nome composto è maiuscola. Viene spesso utilizzato per rendere più facile da leggere il codice. Si differenzia dal *camelCase*, poiché in PascalCase la prima lettera di tutte le parole che compongono il nome dell'oggetto o della funzione devono essere maiuscole.

**PDCA** • Acronimo di Plan-Do-Check-Act, è un metodo iterativo a quattro fasi utilizzato per il controllo e miglioramento continuo di processi e prodotti. Le quattro fasi del PDCA sono: Plan: definizione degli obiettivi e delle attività necessarie per raggiungerli. Do: esecuzione delle attività pianificate. Check: valutazione dei risultati ottenuti e confronto con gli obiettivi prefissati. Act: applicazione delle correzioni e delle azioni di miglioramento necessarie. Il PDCA è spesso utilizzato nel contesto della gestione della qualità per garantire il miglioramento continuo dei processi e dei prodotti.

**Precondizione** • Condizione che deve essere soddisfatta prima dell'esecuzione di un'operazione. Una condizione o un predicato che deve essere sempre verificato prima dell'esecuzione di una sezione di codice o di un'operazione in una specifica formale. Viene usato anche al plurale con il termine 'precondizioni'.

**Processo** • Insieme di attività interconnesse che trasformano gli input (bisogni) in output (prodotti), utilizzando risorse e seguendo regole prestabilite. Deve essere condotto in modo sistematico, disciplinato e misurabile

**Product Baseline** • E' la successiva Baseline del prodotto dopo la Technology Baseline, in cui si definiscono eventuali design pattern, diagrammi delle classi e diagrammi delle attività. In questa fase vengono anche identificati tutti i requisiti e le funzionalità del prodotto, insieme ai relativi test di verifica e validazione.

**Proof of Concept** • o PoC, è un'attività preliminare di sviluppo di un prototipo software che ha lo scopo di dimostrare la fattibilità tecnica e la validità di un'idea o di un concept. Solitamente viene utilizzata come strumento di validazione iniziale prima di avviare lo sviluppo effettivo del progetto.

**Proponente** • Ente o azienda che presenta il capitolato d'appalto per un determinato progetto, indicando le specifiche richieste e le modalità di svolgimento dei lavori.

**Postcondizione** • Condizione o un predicato che deve essere sempre vero al termine dell'esecuzione di una sezione di codice o di un'operazione in una specifica formale. Il termine "post" si riferisce al fatto che la condizione viene verificata dopo l'esecuzione dell'azione specificata. Usato anche con il termine plurale "postcondizioni".

**Pull request** • Funzionalità di GitHub che consente a un collaboratore di richiedere ai membri del team la revisione e l'approvazione delle modifiche apportate ai file e pubblicate su un branch specifico. Durante il processo di pull request, i membri del team possono esaminare le modifiche e fornire commenti o suggerimenti. Una volta che la pull request è stata approvata, le modifiche possono essere integrate nel branch principale del progetto. In alcuni casi, potrebbe essere necessario apportare ulteriori modifiche ai file prima di poter pubblicare la pull request sul branch principale, al fine di evitare conflitti con la versione corrente del progetto.

## R

**Repository** • In informatica, è una raccolta di file e directory che costituiscono la versione corrente di un progetto software o di un insieme di dati. Un repository consente di gestire e controllare la cronologia delle modifiche apportate ai file e di mantenere traccia dei cambiamenti effettuati da ogni membro del team. Permette inoltre di collaborare con altri utenti, in modo che possano contribuire al progetto e apportare le loro modifiche in modo controllato e coordinato.

**Requirements and Technology Baseline** • E'una  $baseline_G$  di un progetto software che definisce i requisiti e le tecnologie utilizzate per realizzarli. In particolare, la Requirements and Technology Baseline (acronimo RTB) identifica i requisiti funzionali e non funzionali del sistema, le tecnologie, le librerie e le piattaforme utilizzate per implementare tali requisiti, così come le strategie di sviluppo e di testing del software. La RTB costituisce un punto di riferimento fondamentale per il controllo del progetto e per la verifica della sua conformità ai requisiti del cliente.

**Requisito** • Rappresenta le specifiche delle funzionalità, dei vincoli e delle performance che il sistema deve soddisfare per adempiere a un obbligo o risolvere un problema. Essi definiscono ciò di cui il sistema ha bisogno e possono essere visti da due punti di vista: dal punto di vista del bisogno del cliente e dal punto di vista della soluzione del fornitore.

**Roll back** • Operazione di reversibilità eseguita in risposta ad un evento indesiderato durante l'esecuzione di una transazione, operazione o modifica del sistema. Consiste nell'annullamento delle modifiche apportate e nel ripristino dello stato precedente l'operazione. Il Rollback viene spesso utilizzato in situazioni di emergenza o di fallimento del sistema per evitare la perdita di dati o il deterioramento delle prestazioni.

**Ruolo** • Funzione o compito specifico assegnato a una persona all'interno di un progetto o di un'organizzazione.

## S

**Showroom** • Spazio espositivo, utilizzato dalle aziende per presentare i propri prodotti o servizi ai clienti ed esposti in modo tale da attirare l'attenzione del pubblico e permettere ai potenziali acquirenti di vederli, toccarli e valutarli in modo più accurato

**Showroom 3d** • Ambiente virtuale interattivo tridimensionale che consente di presentare prodotti o servizi in modo realistico e coinvolgente. È un'evoluzione del tradizionale showroom fisico, con l'aggiunta di tecnologie digitali avanzate come la realtà virtuale e la realtà aumentata. Gli utenti possono esplorare il prodotto o il servizio in modo intuitivo, vedere come funziona e scoprire tutte le sue caratteristiche senza doversi spostare fisicamente.

**Stakeholder** • Indica una persona o un'organizzazione che ha un interesse o una relazione con un progetto o un prodotto e avere un'opinione rilevante ai fini realizzativi del prodotto. Questo può includere investitori, fornitori, clienti, dipendenti e altre parti interessate dell'organizzazione.

**Standard** • Insieme di norme regole, raccomandazioni o specifiche che sono convenzionali e predefinite da un'autorità competente e riconosciute come tali. Lo scopo di uno standard è quello di fornire una base di riferimento per la creazione di tecnologie che siano compatibili e interoperabili tra loro.

**Stato dell arte** • Situazione attuale e il livello di sviluppo raggiunto in un determinato campo di conoscenza o di tecnologia.



## T

**Task** • vedi definizione di  $\text{compito}_G$ .

**Telegram** • Servizio di messaggistica istantanea basato su cloud fornito da Telegram LLC. Utilizzato da utenti in tutto il mondo, permette lo scambio di messaggi di testo, vocali, multimediali e file di qualsiasi tipo, senza fini di lucro. Telegram offre anche funzionalità avanzate come le chat segrete e la crittografia end-to-end.

**Test** • processo di valutazione di un sistema o un componente di sistema al fine di verificare se soddisfa le specifiche e i requisiti previsti. Esistono diverse tipologie di test: test funzionale che verifica il comportamento del sistema rispetto a casi di prova specifici, test di regressione che verifica che eventuali modifiche non abbiano impattato negativamente funzionalità già testate e test di accettazione che viene effettuato dal cliente o utente finale per verificare se il sistema soddisfa le proprie aspettative e requisiti.

**TeXLive** • Distribuzione del sistema di composizione tipografica TeX. Include molti pacchetti e programmi aggiuntivi per aiutare a scrivere documenti TeX. La distribuzione TeXLive è disponibile per diverse piattaforme, tra cui Windows, Linux e macOS, ed viene utilizzata soprattutto per scrivere documenti scientifici, tesi di laurea e articoli tecnici.

**Texture** • In informatica e grafica, una texture si riferisce a un'immagine o a una mappa di pixel che viene applicata a una superficie 3D per simulare l'aspetto di una superficie più complessa o dettagliata. Le texture sono spesso utilizzate nei videogiochi, nelle animazioni, nei film e in altre forme di media digitali per creare l'illusione di dettagli o di superfici più realistiche.

**Ticketing** • Sistema utilizzato per la gestione del supporto tecnico all'interno di un'organizzazione, che consente agli utenti di segnalare problemi o richieste di assistenza attraverso l'apertura di un "ticket". Questo ticket viene quindi gestito dal personale dell'helpdesk, che monitora lo stato della richiesta e fornisce una soluzione al problema segnalato.

**Tracciamento** • E' un processo che consente di tenere traccia delle informazioni relative a un particolare elemento all'interno di un sistema. Nel contesto dei servizi online, il tracciamento si riferisce alla raccolta di informazioni sull'utente, che può essere utilizzata per creare un profilo dell'utente stesso. Questo tipo di tracciamento può essere parte integrante di sistemi organizzativi o di qualsiasi altro servizio, come ad esempio a fini statistici. Nel contesto dello sviluppo software, il tracciamento dei requisiti è il processo di recupero della fonte di ogni requisito e di prevederne gli effetti. Ogni requisito deve essere ricondotto ai requisiti e agli stakeholder da cui ha avuto origine, mentre ogni prodotto realizzato (o requisito derivato) deve essere riconducibile al requisito che soddisfa. Il tracciamento è un processo costante che si svolge lungo l'intero ciclo di vita del prodotto, e permette di assicurare che i requisiti concordati siano tutti e soli quelli necessari e sufficienti. Il ri-

sultato del tracciamento è un grafo aciclico e diretto, che viene costantemente aggiornato.

## U

**UML** • (o Unified Modeling Language) è un linguaggio di modellazione e specifica che si basa sul paradigma orientato agli oggetti. Utilizza una notazione visuale, semi-grafica e semi-formale e fornisce una serie di diagrammi, elementi testuali formali ed elementi di testo libero che compongono un linguaggio unificato. Lo scopo di UML è semplificare la progettazione e la programmazione orientata agli oggetti, offrendo uno strumento standardizzato e condiviso tra i membri del team di sviluppo software

**UML 2** • La sigla UML 2 indica la seconda versione del linguaggio UML<sub>G</sub>. A differenza della precedente versione 1.5, offre un quadro di riferimento più ampio, con l'inclusione di nuovi tipi di strumenti e diagrammi per semplificare la progettazione e la programmazione orientata agli oggetti.

**Unity** • Motore di gioco multipiattaforma utilizzato per sviluppare videogiochi e altre applicazioni interattive, come esperienze in realtà virtuale e aumentata. È caratterizzato dalla sua facilità d'uso, permettendo anche a sviluppatori meno esperti di creare giochi di alta qualità con funzionalità avanzate. Unity supporta numerose piattaforme, tra cui PC, console di gioco, dispositivi mobili e web.

**Unreal Editor** • Software di sviluppo di giochi e simulazioni 3D basato sull'Unreal Engine. Offre una vasta gamma di strumenti per creare, modificare e testare il contenuto di un gioco, inclusi gli strumenti di modellazione, di animazione, di scrittura di codice e di rendering. Unreal Editor fornisce un ambiente di sviluppo integrato (IDE) per la scrittura di codice in C++, visual scripting e blueprint, che consente agli sviluppatori di creare funzionalità personalizzate e personalizzazioni avanzate per giochi e simulazioni.

**Unreal Engine** • Motore di gioco sviluppato da Epic Games utilizzato per creare videogiochi e applicazioni interattive in 3D. Caratterizzato dalla presenza di un editor visuale, chiamato Unreal Editor, consente di creare ambienti, effetti speciali e elementi grafici all'interno di un ambiente di sviluppo integrato (IDE) basato su node e blueprint. Questo motore dispone di una vasta gamma di funzionalità per la gestione di fisica, luci, suoni e animazioni, nonché di un supporto esteso per la realtà virtuale e aumentata

## V

**VCS** • acronimo di Version Control System, è un sistema software che consente di gestire le versioni dei file di un progetto, permettendo ai diversi membri del team di lavorare in modo collaborativo, sincronizzato e controllato sul codice sorgente, sulla documentazione o su qualsiasi altro tipo di file. Il VCS registra tutte le modifiche effettuate sui file e consente di ripristinare qualsiasi versione precedente, in modo da avere uno storico completo delle modifiche apportate nel tempo. Consente inoltre di gestire conflitti tra versioni diverse di uno stesso file e di lavorare su branch separati per sviluppare funzionalità in parallelo senza interferire tra di loro.

**Versionare** • Il termine "versionare" si riferisce all'attività di gestione delle versioni di un software o di un progetto, ovvero al processo di creazione, organizzazione e controllo delle varie versioni o revisioni di un file o di un insieme di file che compongono un progetto. L'obiettivo principale del versioning è quello di tracciare le modifiche apportate ai file nel tempo, facilitando il lavoro collaborativo, la risoluzione dei conflitti e il ripristino di versioni precedenti. I sistemi di versioning più comuni sono i VCS (version control system), come Git, SVN e Mercurial.

**Visual Studio Code** • Noto come VS Code, è un editor di codice sorgente sviluppato da Microsoft per Windows, Linux e macOS. E' dotato di numerose funzionalità per lo sviluppo di software, tra cui la sintassi del codice highlighting, il debugging, la gestione del controllo di versione, l'integrazione con Git, l'autocompletamento, la formattazione del codice, la visualizzazione della documentazione del codice e molto altro. Visual Studio Code è estendibile e personalizzabile.

## W

**Walkthrough** • Tecnica di verifica utilizzata per eseguire una lettura completa e un controllo di un documento, con l'obiettivo di individuare eventuali errori. In questa tecnica, il verificatore esegue una revisione completa del documento senza conoscere in anticipo gli errori potenziali, al fine di individuare i problemi e proporre eventuali miglioramenti.

**Way of Working** • Il modo in cui le attività di progetto vengono organizzate, pianificate e gestite, al fine di renderle sistematiche, disciplinate e misurabili.

**Windows** • E' una famiglia di sistemi operativi sviluppati e commercializzati da Microsoft. Caratterizzati da una interfaccia utente a finestre, i sistemi operativi Windows sono utilizzati su una vasta gamma di dispositivi, tra cui personal computer, tablet, smartphone, server e altri dispositivi embedded. Windows supporta diverse architetture di processori, diverse lingue e molteplici funzionalità, tra cui la gestione di file e cartelle, la connessione a reti e internet, la gestione di periferiche hardware e l'installazione di applicazioni software.