Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software
Anno Accademico: 2025/26

BitByBit

Gruppo 17

Nome: BitByBit
Email: swe.bitbybit@gmail.com

Glossario

Stato:	Completato
Versione:	1.0.0
Redattori:	Ferdinando Fracasso
	Riccardo Manisi
	Giovanni Visentin
	Dennis Parolin
Verificatori:	Gabriele Scaggiante
	Dennis Parolin
	Giovanni Visentin
Responsabile:	Giovanni Visentin
Destinatari:	Gruppo BitByBit

Registro delle modifiche

Versione	Data	Autore	Descrizione	Verificatore
1.0.0	2026-03-02	Giovanni Visentin	Documento in versione stabilizzata	Dennis Parolin
0.4.0	2026-02-27	Marco Sanguin	Aggiunti Indice Analitico	Riccardo Manisi
0.3.3	2026-01-11	Dennis Parolin	Aggiunti Termini	Riccardo Manisi
0.3.2	2025-12-23	Riccardo Manisi	Aggiunta sezione info generali documenti e migliorato layout	Dennis Parolin
0.3.1	2025-12-16	Ferdinando Fracasso	Rimosso definizioni di termini non utilizzati e tolta la sezione introduttiva perchè superflua, inoltre modificato definizioni di branch/merge/pull request per chiarezza	Giovanni Visentin
0.3.0	2025-12-12	Ferdinando Fracasso	Ristrutturato layout documento e aggiunto sezione introduttiva	Giovanni Visentin
0.2.1	2025-11-05	Dennis Parolin	Aggiunti termini al documento e sistemato alcuni errori grammaticali	Gabriele Scaggiante
0.2.0	2025-11-06	Riccardo Manisi	Tolta la sezione delle convenzioni di scrittura, aggiunto indice e formattazione, inserito nuovi termini	Dennis Parolin
0.1.1	2025-11-05	Giovanni Visentin	Aggiunti termini al documento	Gabriele Scaggiante
0.1.0	2025-11-04	Giovanni Visentin	Stesura iniziale del glossario	Gabriele Scaggiante

Indice

Indice analitico

Scopo del documento

Il presente documento ha lo scopo di raccogliere i termini tecnici utilizzati all'interno del progetto e fornire le loro definizioni, in modo da permettere una più facile comprensione della documentazione prodotta durante lo svolgimento del progetto.

A

Agile development

Metodo per lo sviluppo sw ideato per produrre sw utile nel minor tempo possibile. Il sistema è sviluppato attraverso una serie di incrementi a cui partecipano anche gli stakeholders e gli utenti finali per una maggiore comunicazione in fase di sviluppo. I risultati di ogni incremento vengono utilizzati come base per la pianificazione del prossimo. Attività centrali sono: il **design** e **l'implementazione**. La stesura di un documento dei requisiti è superflua in quanto i requisiti cambiano così velocemente che il documento è obsoleto non appena viene scritto.

Analisi dei requisiti

Stabilisce esattamente cosa deve essere implementato nel sistema, cioè elenca i requisiti lato soluzione (requisiti software). Può essere parte del contatto con l'acquirente. Può essere soggetto a modifiche durante lo sviluppo del prodotto sw.

Analisi Dinamica

Tecnica di verifica che richiede l'esecuzione del codice software su un processore (reale o virtuale).

- Il suo obiettivo è osservare il comportamento del sistema in risposta a determinati input per rilevare malfunzionamenti (failure).
- Include le attività di testing (Unità, Integrazione, Sistema).

Analisi Statica

Tecnica di verifica che non prevede l'esecuzione del codice.

- Si basa sull'esame dei documenti o del codice sorgente per individuare difetti di forma, sintassi o logica.
- Include attività come l'ispezione, il walkthrough e l'uso di strumenti automatici (linter).

Architettura

Organizzazione generale del sistema e dei suoi componenti.

B

Baseline

Versione approvata di un prodotto di lavoro (di progetto) che può essere modificato solo attraverso procedure formali di controllo delle modifiche.

Black-box

Metodologia di test in cui il verificatore non conosce o ignora la struttura interna del sistema (codice sorgente).

- I test sono progettati basandosi esclusivamente sulle specifiche dei requisiti (input e output attesi).
- È nota anche come testing funzionale.

Board di progetto

Rappresenta un'organizzazione delle attività, nel contesto di GitHub identificate tramite le issue. Può contenere tutte le attività di un progetto o solo un sottoinsieme con caratteristiche precise.

Boundary Analysis

Tecnica di progettazione dei test black-box che si concentra sui valori ai margini delle classi di equivalenza degli input (es. minimo, massimo, appena sopra, appena sotto). È utilizzata perché i difetti si nascondono spesso ai bordi dei range di input validi.

Branch

Versione parallela del repository. È contenuto all'interno del repository, ma non ha effetti sul branch principale. Permette di lavorare senza interrompere la versione stabile corrente.

Bugfix

Intervento di manutenzione correttiva volto a risolvere un difetto (bug) riscontrato nel software. Include la diagnosi del problema, la modifica del codice e la successiva verifica.

C

Capitolato

Documento che contiene le specifiche di un progetto software. Viene redatto dal proponente e presentato ai fornitori interessati a partecipare all'appalto. Presenta aspettative, vincoli, e suggerimenti. Descrive i requisiti lato bisogno (requisiti utente).

Checklist

Lista di controllo strutturata utilizzata durante le ispezioni per guidare il verificatore. Contiene un elenco di criteri ed errori comuni da cercare nell'artefatto esaminato, garantendo che nessun aspetto critico venga trascurato.

CI/CD

Insieme di pratiche e strumenti che automatizzano le fasi di integrazione, test e rilascio del software.

- **CI**: (Continuous Integration) integra frequentemente le modifiche di codice in un repository condiviso, eseguendo build e test automatici.
- **CD**: (Continuous Delivery) automatizza il rilascio del software negli ambienti di test o produzione.

Ciclo di vita

È composto da un insieme di stati che descrivono l'avanzamento del prodotto software dalla concezione all'utilizzo fino al ritiro. Si riferisce alla completa durata dello sviluppo del prodotto, dal concepimento alla fine, che deve essere garantita. È composto da stati in cui si ha una certa sequenza di passaggi da seguire. La transizione da uno stato all'altro avviene come conseguenza di un processo di ciclo di vita. In seguito al ritiro il prodotto può essere rimesso in vita, per questo viene indicato con ciclo.

Committente

Colui che ordina l'esecuzione di un prodotto ad un ente esterno, il fornitore. È colui che ha il maggior peso contrattuale.

Configuration Item

Identifica una componente o un elemento infrastrutturale che deve essere gestito e monitorato per garantire la corretta erogazione dei servizi software.

Continuous Verification

Approccio che integra la verifica (statica e dinamica) in ogni fase del flusso di sviluppo, anziché posticiparla alla fine. Nel contesto del gruppo, è implementata tramite policy che impediscono il merge di codice non verificato.

Contratto

Documento (legale) stilato tra committente e fornitore.

D

Deployment

Il processo di messa in opera di un sistema software o di un aggiornamento in un ambiente specifico (es. ambiente di staging o di produzione), rendendolo disponibile per l'uso o per il collaudo.

Driver

Componente software fittizio utilizzato nei test di integrazione (strategia Bottom-Up). Simula il comportamento di un modulo di livello superiore che deve chiamare e pilotare il modulo sotto test, passandogli gli input necessari.

E

F

G

Git Flow

Modello di gestione dei branch in Git che definisce ruoli specifici per i diversi rami (es. `main`, `develop`, `feature/*`) e regole precise per la loro interazione, facilitando il lavoro parallelo e il rilascio di versioni.

GitHub Actions

Piattaforma di integrazione continua (CI) e distribuzione continua (CD) integrata in GitHub, che permette di automatizzare i flussi di lavoro di compilazione, test e deployment direttamente dal repository.

H

I

IDE

Applicazione software che fornisce agli sviluppatori strumenti completi per lo sviluppo software. Include solitamente un editor di codice, strumenti di build automation, un debugger e funzionalità di intellisense (es. Visual Studio Code).

Ispezione

Tecnica formale di analisi statica in cui un verificatore esamina un artefatto (codice o documento) in autonomia, utilizzando una checklist per individuare discrepanze rispetto agli standard. È un processo rigoroso e documentato.

Issue

Unità di lavoro tracciata su un sistema di gestione progetti (es. GitHub). Nel contesto della validazione, viene utilizzata per tracciare anomalie o bug riscontrati durante i test.

J

K

L

Linter

Strumento di analisi statica automatizzata che analizza il codice sorgente per segnalare errori di programmazione, bug, errori stilistici e costrutti sospetti.

M

Matrice di Tracciamento

Tabella che mappa le relazioni tra i requisiti e altri artefatti del progetto, in particolare i test. Serve a garantire che ogni requisito abbia almeno un test associato (copertura) e che ogni test sia giustificato da un requisito.

Milestone

Strumento utilizzato nella gestione di progetto per fissare una data di calendario che descrive degli obiettivi che portano un avanzamento nello sviluppo di progetto. Ogni milestone deve essere:

- specifica
- raggiungibile
- misurabile
- traducibile in compiti assegnabili
- dimostrabile dagli stakeholders

Merge

Metodo per integrare i cambiamenti fatti in un branch di lavoro in un altro branch, definito di destinazione. Può essere eseguito con le pull request o tramite la command line.

Metrica

Qualsiasi tipo di misurazione relativa a un sistema software, processo o documentazione. Permette di quantificare un attributi di un prodotto o un processo.

MVP

Sigla per **Minimum Viable Product**. Artefatto che definisce la versione di un prodotto dotata di funzionalità sufficienti per essere distribuito efficacemente. Lo scopo è quello di ottenere dei feedback da parte dal committente e/o dei potenziali utenti utilizzatori. La creazione di questo prodotto deve consumare la minima quantità di risorse del fornitore, in quanto può essere scartato completamente.

N

O

P

PB

Sigla per **Product Baseline**. Baseline di progetto per validare l'architettura del prodotto software e la sua realizzazione. Espone il design definitivo del prodotto descritto nel documento delle specifiche tecniche.

Piano di progetto

Documento ufficiale di tipo esterno, soggetto ad approvazioni, con il quale si descrivono gli obiettivi di progetto e gli elementi necessari al loro raggiungimento. Si vedono le risorse disponibili e le loro assegnazioni alle attività con scansione nel tempo. Principalmente è scomposto in:

- Pianificazione (preventiva): Team, Analisi dei rischi
- Consuntivazione: valutazione retrospettiva delle attività svolte

Struttura tipica del PdP:

- Introduzione (scopo e struttura)
- Organizzazione del progetto
- Analisi dei rischi
- Risorse disponibili
- Suddivisione del lavoro (work breakdown)
- Calendario delle attività
- Meccanismo di controllo e rendicontazione

Piano di qualifica

Documento per la specifica degli obiettivi di qualità. Espone gli standard adottati per la misurazione della qualità e le componenti che devono essere oggetto di test di verifica. Vengono presentate anche le misurazioni effettuate tramite il cruscotto della qualità durante gli sprint di progetto.

PoC

Sigla per **Proof of Concept**. Artefatto che serve a valutare la fattibilità tecnologica del prodotto che si intende sviluppare. Le tecnologie adottate devono essere decise in base alle specifiche funzionalità individuate a partire dai bisogni del committente.

Processo

Un insieme strutturato di attività, metodi, pratiche e trasformazioni utilizzate per sviluppare, manutenere e gestire un prodotto software o un sistema. Definisce cosa deve essere fatto, chi lo fa, quando e come, per raggiungere un obiettivo specifico.

Product backlog

Lista dinamica che evolve nel tempo con il prodotto. Vengono rappresentati gli item da eseguire nei prossimi rilasci con un livello di priorità assegnato a ciascuno. Gli item presenti rappresentano le macroattività che devono essere eseguite dal team.

Proponente

Azienda esterna all'università che propone un capitolato da sviluppare.

Project

È un insieme ordinato di attività sviluppate sulla base dei processi di ciclo di vita che soddisfano gli obiettivi dati.

- Le attività sono suddivise in compiti assegnabili a singoli individui.
- Le attività sono pianificate prima di essere svolte.

Pull Request

Modifiche proposte da un repository inviate da un utente e accettate o scartate da un collaboratore interno al repository.

Q

Quality Gate

Punto di controllo nel ciclo di vita del software che il progetto deve superare per passare alla fase successiva. Nel contesto della CI/CD, è un insieme di criteri automatici (es. "nessun test fallito") che bloccano il merge se non soddisfatti.

R

Refactoring

Processo di ristrutturazione del codice esistente senza modificarne il comportamento esterno. L'obiettivo è migliorare la leggibilità, ridurre la complessità e facilitare la manutenzione futura.

Regressione

Malfunzionamento introdotto involontariamente in una funzionalità precedentemente funzionante a seguito di una modifica al codice (es. bugfix o nuova feature).

Release

Una versione particolare di un configuration item resa disponibile per uno scopo specifico.

Repository

Archivio centralizzato dove viene conservato e gestito il codice sorgente del progetto.

Requisito

Condizione o capacità che il sistema deve soddisfare.

Requisito funzionale

Funzionalità che il sistema deve offrire.

Requisito non funzionale

Vincolo di qualità (prestazioni, sicurezza, usabilità, manutentibilità).

Requisito software

Descrizioni dettagliate delle funzioni, servizi e vincoli di sistema, inseriti nel documento di analisi dei requisiti.

Requisito utente

Requisiti astratti di alto livello. Dichiarazioni in linguaggio naturale (e anche diagrammi) dei servizi che il sistema dovrebbe fornire agli utenti e dei vincoli che deve rispettare.

RTB

Sigla per **Requirements and Technology Baseline**. Rappresenta una baseline di progetto. Produce la lista dei requisiti funzionali e non funzionali da soddisfare, stabiliti con la proponente e il PoC.

S

Self-documenting code

Codice scritto in modo talmente chiaro (tramite nomi significativi di variabili e funzioni) da rendere superflui i commenti esplicativi per comprenderne il funzionamento logico.

SemVer

Sigla per **Semantic Versioning**. Metodologia composta da un insieme di regole e vincoli che specifica come i numeri di versione sono incrementati e assegnati.

Sprint

Breve periodo di tempo, dalla durata prefissata, in cui un team lavora per completare le attività elencate nello sprint backlog di quel preciso sprint.

Sprint backlog

Lista degli item riferita ad un particolare sprint di progetto. Gli item al suo interno possono essere scomposti in attività più piccole, ma non possono essere mai eliminati.

Sprint Retrospective

Riunione che viene fatta con la sola presenza dei membri del team per identificare le attività portate a termine e i problemi insorti durante la loro esecuzione. Ha l'obiettivo di identificare i potenziali errori fatti.

Sprint Review

Riunione, fatta alla fine di uno sprint, che consta di una revisione per identificare il progresso eseguito sul prodotto. Partecipano anche gli stakeholders.

Stakeholder

Tutti coloro che a vario titolo hanno influenza sul prodotto e sul progetto: utenti, committente, fornitore, regolatori.

Statement Coverage

Metrica di copertura del codice che indica la percentuale di istruzioni (righe di codice) eseguite almeno una volta durante i test automatici.

Stub

Componente software fittizio utilizzato nei test di integrazione (strategia Top-Down). Simula il comportamento di un modulo di livello inferiore non ancora implementato, fornendo risposte predefinite alle chiamate del modulo sotto test.

T

Test di Accettazione

Livello di test finalizzato a validare il sistema rispetto ai bisogni dell'utente.

- Viene eseguito in un ambiente simile a quello reale.
- Ha lo scopo di ottenere l'approvazione formale del cliente per il rilascio (Validazione).

Test di Integrazione

Livello di test che verifica le interazioni tra componenti software o sistemi diversi. L'obiettivo è rilevare difetti nelle interfacce e nello scambio di dati tra moduli che singolarmente funzionano.

Test di Regressione

Riesecuzione di test già superati in precedenza per assicurarsi che le recenti modifiche al codice non abbiano introdotto nuovi difetti in parti del software già verificate.

Test di Sistema

Livello di test che verifica il comportamento dell'intero sistema integrato rispetto ai requisiti specificati. Copre aspetti funzionali e non funzionali (prestazioni, sicurezza).

Test di Unità

Livello di test che verifica le più piccole parti testabili del software (es. singole funzioni o metodi) in isolamento dal resto del sistema.

To-Do

Attività che derivano da decisioni prese dai componenti del gruppo. Ogni To-Do deve essere tracciabile all'interno del contesto del progetto tramite la sua formalizzazione in issue GitHub.

U

Scenari d'uso

Descrizioni narrative di come un utente interagisce con il sistema per raggiungere un obiettivo specifico. Sono utilizzati durante la validazione per verificare che il flusso di lavoro supporti le esigenze reali.

Utente

La persona che usa e interagisce direttamente col sistema (spesso si identifica col committente).

V

Validazione

Processo di conferma, attraverso evidenze oggettive, che il software soddisfi le esigenze dell'utente e l'uso previsto. Risponde alla domanda: "Abbiamo costruito il prodotto giusto?".

Verbale

Documento riassuntivo di un incontro. Deve specificare Ordine del Giorno, Discussioni, Decisioni, Issue e To Do. Ogni verbale deve essere verificato da un membro diverso da chi l'ha scritto.

Verifica

Processo di valutazione per determinare se i prodotti di una fase di sviluppo soddisfano le condizioni imposte all'inizio di quella fase. Risponde alla domanda: "Stiamo costruendo il prodotto nel modo giusto?".

W

Walkthrough

Tecnica di analisi statica informale in cui l'autore di un artefatto guida i membri del team attraverso il documento o il codice per individuare errori e condividere conoscenza.

Way of working

Comprende le metodologie di sviluppo (modello di sviluppo), la spartizione dei ruoli e delle attività, gli strumenti di lavoro (GitHub, GitLab, Trello, Jira, Slack, Discord, Obsidian, Google Docs), convenzioni standard e procedure condivise dai membri del team.

White-box

Metodologia di test in cui il verificatore conosce la struttura interna del codice e la utilizza per progettare i casi di test.

- Mira a coprire i percorsi logici interni (es. rami if-else, cicli).
- È nota anche come testing strutturale.

X

Y

Z