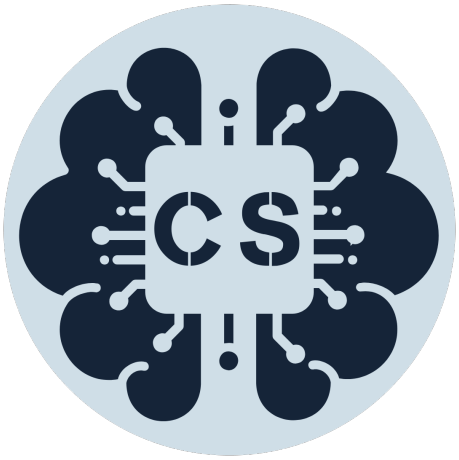
Glossario

CyberSorcerers Team



Membri del team:
Sabrina Caniato
Giulia Dentone
Nicola Lazzarin
Giovanni Moretti
Andrea Rezzi
Samuele Vignotto

Informazioni sul documento	
Destinatari:	Prf. Tullio Vardanega, Prf. Riccardo Cardin
Caratteristiche	Parole in ordine alfabetico
	Indicate nei documenti con una "G" al pedice

Indice

1 Attori 4
2 Baseline 4
3 Branch 4
4 Casi d’uso 4
5 Changelog 4
6 Ciclo di vita 5
7 Ciclo PDCA 5
8 Design Thinking 6
9 Epic stories 6
10 Feature 6
11 Framework Scrum 6
12 Flow di Funzionamento 7
13 IA 7
14 Issue 7
15 Issue tracking system 7
16 Mappa dell’empatia 8
17 Middleware 8
18 Modello Agile 8
19 Performance 9
20 Plugin 9
21 Project Manager 9
22 Proof of Concept 9
23 Pull request 10
24 Repository 10

25 Requisiti di Business **10**

26 Stakeholder **10**

27 Task **10**

28 Tag **11**

29 Template **11**

30 Testing **11**

31 Trade Off **11**

32 User Stories **11**

33 Versione **11**

34 Way of working **12**

35 Web App **12**

1 Attori

Nel contesto di un diagramma dei casi d'uso, un attore rappresenta un ruolo o una classe di utenti esterni al sistema che interagisce con il sistema stesso.

2 Baseline

Rappresenta uno stato o una configurazione di un sistema o di un progetto in un determinato momento, che viene utilizzato come riferimento fondamentale per misurare il progresso, valutare le modifiche e gestire la configurazione. Una baseline può includere documenti, codice sorgente, specifiche di progetto, pianificazioni, e altri elementi chiave. Le baseline vengono stabilite in fasi cruciali di un progetto e servono come punti di riferimento stabili per la gestione del cambiamento. Le principali baseline in un progetto software possono includere la baseline dei requisiti, la baseline dell'architettura, la baseline del design e la baseline del codice. Ogni baseline rappresenta uno stato verificato e accettato del progetto in un dato momento, fornendo una base solida per valutare eventuali modifiche future e garantire la coerenza e l'integrità del sistema nel corso del tempo.

3 Branch

Si riferisce a una diramazione del codice sorgente che viene creata per lavorare su una specifica issue o per sviluppare una nuova funzionalità.

4 Casi d'uso

Un diagramma dei casi d'uso è uno strumento di modellazione utilizzato nell'ambito dell'ingegneria del software per rappresentare graficamente le interazioni tra un sistema e gli attori esterni che interagiscono con esso.

5 Changelog

Un "changelog" è un documento o un registro che traccia e descrive le modifiche apportate a un software o a un progetto nel corso del tempo. Questo registro fornisce una cronologia dettagliata delle versioni del software, elencando le nuove funzionalità aggiunte, le correzioni di bug, le miglioramenti e qualsiasi altra modifica rilevante. Il changelog è un componente essenziale nella gestione del software, poiché fornisce una visione chiara e strutturata delle modifiche effettuate. Queste informazioni sono utili per gli sviluppatori, gli utenti e altri stakeholder, in quanto consentono di comprendere come il software è evoluto nel tempo e forniscono indicazioni su cosa aspettarsi in ogni nuova versione.

6 Ciclo di vita

Il "ciclo di vita del software" è il periodo che inizia con la concezione di un'applicazione o di un sistema software e si estende attraverso le fasi di sviluppo, test, implementazione, manutenzione e, infine, il ritiro o l'aggiornamento del software. Questo concetto rappresenta il percorso completo che un software percorre dalla sua creazione fino alla sua conclusione o evoluzione successiva. Le fasi principali del ciclo di vita del software includono:

- Concezione: Identificazione delle esigenze e dei requisiti del software, definizione degli obiettivi e della portata del progetto.
- Sviluppo: Progettazione e implementazione del software in base ai requisiti definiti nella fase di concezione.
- Test: Validazione e verifica del software attraverso test funzionali, di sistema e di accettazione per garantire che soddisfi gli standard di qualità e risponda alle aspettative degli utenti.
- Implementazione: Distribuzione e installazione del software negli ambienti operativi previsti.
- Manutenzione: Fase in cui vengono apportate modifiche, correzioni di bug e miglioramenti al software in risposta a nuovi requisiti o a cambiamenti nell'ambiente operativo.
- Ritiro o Aggiornamento: Il software viene ritirato se non è più utile o supportato, oppure viene sottoposto a aggiornamenti o evoluzioni per adeguarlo a nuovi requisiti o tecnologie.

Queste fasi possono variare a seconda del modello di sviluppo del software adottato, come il modello a cascata, il modello a spirale, o l'approccio agile. La gestione efficace del ciclo di vita del software è essenziale per garantire il successo del progetto, la qualità del prodotto e la soddisfazione degli utenti.

7 Ciclo PDCA

Il "ciclo PDCA" è un modello di gestione ciclico che rappresenta un approccio iterativo per il miglioramento continuo di processi e prodotti. L'acronimo PDCA sta per Plan-Do-Check-Act, che sono le quattro fasi chiave del ciclo:

- Plan (Pianificare): In questa fase, si pianificano e si stabiliscono gli obiettivi e le attività necessarie per raggiungerli. Si identificano i problemi, si definiscono le soluzioni, e si sviluppa un piano d'azione.
- Do (Eseguire): Durante questa fase, si mette in atto il piano d'azione sviluppato nella fase di pianificazione. Si implementano le soluzioni e si raccoglie dati e informazioni pertinenti.

- Check (Verificare): In questa fase, si valutano i risultati ottenuti mediante la raccolta e l'analisi di dati. Si confrontano i risultati con gli obiettivi prefissati, si identificano le eventuali deviazioni e si valutano le cause di eventuali problemi.
- Act (Agire): Basandosi sui risultati della fase di verifica, si prendono azioni correttive o preventive. Queste azioni possono includere miglioramenti ai processi, l'aggiornamento delle procedure o la revisione dei piani di azione.

8 Design Thinking

Il "design thinking" è un approccio innovativo e centrato sull'utente alla risoluzione dei problemi e alla generazione di soluzioni creative. Questa metodologia si basa sulla collaborazione interdisciplinare, integrando aspetti provenienti dal design, dall'analisi delle esigenze degli utenti e dalla prototipazione rapida. L'obiettivo principale del design thinking è comprendere profondamente le necessità e le prospettive degli utenti, stimolare la creatività del team e iterare rapidamente attraverso diverse soluzioni per raggiungere una risposta ottimale al problema in esame. Inoltre, il design thinking promuove l'empatia, il pensiero aperto e la sperimentazione, incoraggiando i professionisti a considerare molteplici punti di vista al fine di sviluppare soluzioni più innovative e orientate al contesto utente.

9 Epic stories

Un' Epic story è una storia di dimensioni maggiori rispetto a una singola User Story. Le Epic story forniscono una visione ad alto livello dei requisiti di un progetto.

10 Feature

Si riferisce a una funzionalità specifica o a un attributo distintivo di un software, di un'applicazione o di un sistema. Le features sono elementi distinti che contribuiscono alle capacità complessive di un prodotto e possono includere aspetti come la gestione dei dati, le interfacce utente, le prestazioni, la sicurezza, o qualsiasi altra funzione che fornisca un valore specifico agli utenti o ai gestori del sistema.

11 Framework Scrum

Lo Scrum è un framework agile ampiamente utilizzato nel campo dello sviluppo software e della gestione dei progetti. Creato per favorire la flessibilità, la trasparenza e la collaborazione all'interno di un team, lo Scrum offre una struttura organizzativa che facilita la gestione di complessi processi di sviluppo. Il framework Scrum si basa su principi chiave, come la suddivisione del lavoro in brevi iterazioni chiamate "sprint", solitamente della durata di due settimane, e la definizione di ruoli chiave all'interno del team. Le pratiche fondamentali dello Scrum includono

la pianificazione degli sprint, la revisione dello sprint e la retrospectiva dello sprint. La pianificazione del lavoro avviene attraverso la creazione di un backlog di prodotto, una lista priorizzata di funzionalità o attività da svolgere. Durante lo sprint, il team si impegna a completare una porzione di questo backlog, producendo un incremento di prodotto al termine di ogni iterazione.

12 Flow di Funzionamento

Il "flow di funzionamento" si riferisce al flusso di lavoro o alla sequenza di passaggi che vengono eseguiti per completare un compito o un processo specifico.

13 IA

Acronimo per "Intelligenza Artificiale"

14 Issue

Un'issue rappresenta un singolo problema, un'idea, una richiesta di funzionalità o un bug nel contesto di un progetto software ospitato su piattaforme di gestione del codice come GitHub. Ogni issue ha un numero univoco, un titolo descrittivo e un corpo testuale dettagliato che fornisce ulteriori informazioni sul problema in questione.

15 Issue tracking system

Un "issue tracking system" (sistema di tracciamento delle problematiche) è uno strumento software utilizzato per registrare, monitorare e gestire le problematiche, i problemi o i task all'interno di un progetto o di un sistema. Le caratteristiche principali di un issue tracking system includono:

- Consentire agli utenti di registrare in modo strutturato le problematiche, fornendo dettagli come descrizione, priorità, assegnazione, e altri dati pertinenti.
- Assegnare le problematiche ai membri del team responsabili e monitorare lo stato di avanzamento nel tempo. Questo aiuta a garantire che ogni problema venga gestito in modo efficace e tempestivo.
- Fornire la possibilità di aggiungere commenti, note e allegati alle problematiche, consentendo una comunicazione efficace e il condividere di informazioni pertinenti.
- Permettere di classificare le problematiche in base a categorie specifiche, priorità o altri criteri rilevanti per l'organizzazione.
- Conservare un registro delle modifiche apportate alle problematiche nel tempo, consentendo la tracciabilità delle attività e delle decisioni.

- Generare report e dashboard per visualizzare statistiche, tendenze e metriche sulle problematiche, facilitando la valutazione delle prestazioni del progetto.

16 Mappa dell'empatia

La "mappa dell'empatia" è uno strumento di visualizzazione utilizzato nel design thinking e nel processo di sviluppo di prodotti o servizi. Solitamente è divisa in sezioni che rappresentano differenti aspetti dell'esperienza dell'utente. La mappa dell'empatia è spesso realizzata durante sessioni di lavoro collaborative, coinvolgendo membri del team di progettazione, responsabili del prodotto e, se possibile, direttamente gli utenti.

17 Middleware

Il "middleware" è uno strato di software che agisce come intermediario tra il sistema operativo e le applicazioni, facilitando la comunicazione e lo scambio di dati tra diverse componenti di un sistema distribuito o di una rete. Essenzialmente, il middleware svolge un ruolo chiave nell'orchestrare e facilitare l'integrazione di diverse applicazioni e servizi. Le funzioni principali del middleware includono la gestione della comunicazione tra sistemi software, la gestione della distribuzione delle risorse, la sicurezza, la gestione degli errori e la fornitura di servizi di supporto per lo sviluppo di applicazioni distribuite.

18 Modello Agile

Il modello Agile nell'ingegneria del software è un approccio metodologico che promuove lo sviluppo incrementale e iterativo del software, focalizzandosi sulla collaborazione, la flessibilità e la risposta rapida ai cambiamenti nei requisiti del cliente. Questa metodologia si basa su un insieme di principi e valori definiti nel Manifesto Agile, che sottolinea l'importanza di individui e interazioni, software funzionante, collaborazione con il cliente e risposta positiva al cambiamento. Le caratteristiche chiave del modello Agile includono la pianificazione a breve termine attraverso iterazioni chiamate "sprint", la comunicazione continua tra team di sviluppo e stakeholder, il coinvolgimento attivo del cliente durante tutto il processo di sviluppo e la capacità di adattarsi rapidamente alle modifiche dei requisiti. Questo approccio mira a fornire prodotti software di alta qualità in modo efficiente, garantendo al contempo la soddisfazione del cliente e la consegna tempestiva di valore.

19 Performance

In ambito software, il termine "performance" si riferisce all'efficienza e alle prestazioni generali di un'applicazione o di un sistema informatico. Gli aspetti chiave delle performance nel contesto software includono:

- Velocità di esecuzione: Misura quanto rapidamente un'applicazione può completare le sue operazioni, ad esempio, il tempo di risposta per eseguire una funzione o elaborare una richiesta.
- Utilizzo delle risorse: Valuta quanto efficientemente l'applicazione utilizza le risorse di sistema, come la memoria e l'archiviazione, evitando sprechi e garantendo un utilizzo ottimale delle risorse disponibili.
- Scalabilità: Indica la capacità di un'applicazione di gestire un aumento del carico di lavoro o delle richieste senza degradare significativamente le prestazioni.
- Stabilità: Si riferisce alla capacità di un'applicazione di mantenere un funzionamento affidabile e privo di errori sotto varie condizioni, senza crash o comportamenti imprevisti.
- Efficienza energetica: Misura il consumo energetico dell'applicazione o del sistema, essenziale soprattutto in ambienti in cui l'efficienza energetica è una priorità.

20 Plugin

Un "plugin" è un componente software aggiuntivo progettato per estendere le funzionalità di un'applicazione o di un programma principale. Si integra con il software esistente per fornire nuove caratteristiche, miglioramenti o capacità specializzate, senza richiedere modifiche sostanziali al nucleo dell'applicazione.

21 Project Manager

Il Project Manager, o responsabile del progetto, è una figura chiave all'interno di un'organizzazione incaricata di pianificare, eseguire e concludere progetti. Il suo ruolo è multifunzionale e coinvolge diverse responsabilità per garantire che un progetto venga completato in modo efficace, efficiente e in linea con gli obiettivi prefissati.

22 Proof of Concept

Una proof of concept in ingegneria del software è una realizzazione pratica e limitata di un sistema o di una sua parte che mira a dimostrare la fattibilità tecnica o

la validità di un concetto specifico. Questo concetto può riguardare l'architettura del software, l'integrazione di tecnologie, l'efficacia di un algoritmo, o qualsiasi altro aspetto chiave del progetto.

23 Pull request

La pull request, all'interno dell'issue tracking system, contiene le modifiche apportate al progetto, fornendo una panoramica delle modifiche e facilitando la revisione da parte di altri sviluppatori o responsabili del progetto.

24 Repository

Spazio di archiviazione digitale che contiene e gestisce un insieme organizzato di file, documenti o dati. Nei contesti di sviluppo software, il termine si riferisce comunemente a un "repository di codice sorgente", che è una struttura in cui viene conservato e versionato il codice di un progetto software. Un repository di codice sorgente consente agli sviluppatori di collaborare, tenere traccia delle modifiche nel tempo e gestire lo sviluppo del software in modo coordinato. Ogni modifica al codice viene registrata come una nuova "versione", consentendo agli sviluppatori di monitorare il progresso, risalire alle versioni precedenti e gestire eventuali conflitti.

25 Requisiti di Business

I requisiti di business sono le specifiche delle esigenze aziendali che guidano il processo di progettazione e sviluppo di un sistema, prodotto o servizio.

26 Stakeholder

Un individuo, un gruppo o un'organizzazione che ha un interesse diretto o indiretto nel successo di un progetto software. Gli stakeholder includono, ma non sono limitati a, clienti, utenti finali, responsabili del progetto, sviluppatori, tester, dirigenti, responsabili delle risorse umane e qualsiasi altra parte coinvolta nel ciclo di vita del software.

27 Task

In informatica, un "task" è un'unità di lavoro specifica o un compito assegnato che deve essere completato all'interno di un sistema o di un'applicazione software.

28 Tag

Un "tag" è un'etichetta associata a un elemento di dati, spesso utilizzata per organizzare e categorizzare informazioni. I tag sono utilizzati in diversi contesti, come la gestione di file, la catalogazione di contenuti web, eccetera. Possono essere parole chiave o identificatori brevi che forniscono informazioni specifiche sulla natura o il contenuto di un elemento. L'uso di tag facilita la ricerca, il filtraggio e l'organizzazione dei dati, migliorando l'efficienza nella gestione delle informazioni digitali.

29 Template

Modello o uno stampo predefinito che fornisce una struttura di base per la creazione di documenti, pagine web, codice o altri elementi digitali.

30 Testing

Si riferisce al processo sistematico di valutazione e verifica di un software o di un sistema informatico per identificare errori, problemi o discrepanze rispetto ai requisiti specificati. L'obiettivo del testing è garantire che il software sia robusto, affidabile e in grado di soddisfare le aspettative degli utenti e gli obiettivi del progetto.

31 Trade Off

Il termine "trade-off" si riferisce a una situazione in cui si verificano vantaggi e svantaggi o compromessi tra due o più opzioni o obiettivi. In altre parole, implicano la scelta di rinunciare a qualcosa in favore di qualcos'altro.

32 User Stories

Una User Story rappresenta un requisito o una funzionalità dal punto di vista dell'utente. È una breve descrizione di una caratteristica desiderata, scritta in modo da essere comprensibile agli stakeholder e orientata al valore per l'utente finale.

33 Versione

Si riferisce a una specifica iterazione o release del programma che ha subito modifiche rispetto alla sua precedente.

34 Way of working

In un contesto di ingegneria del software, il termine "way of working" si riferisce al metodo e alle pratiche che guidano l'approccio complessivo adottato per lo sviluppo del software all'interno di un progetto. Questo comprende le metodologie, i processi, gli strumenti e le interazioni che definiscono come il team di sviluppo organizza e gestisce il lavoro. La "way of working" in ingegneria del software può includere aspetti come:

- Specifica l'approccio metodologico adottato per pianificare, progettare, implementare e testare il software.
- Descrive le fasi e le attività coinvolte nello sviluppo del software, inclusi processi di codifica, testing, revisione del codice, gestione delle modifiche e rilascio del prodotto.
- Indica gli strumenti, le piattaforme e le tecnologie utilizzate per supportare il processo di sviluppo, inclusi ambienti di sviluppo, sistemi di gestione di versione (come Git), sistemi di build e strumenti di testing.
- Riguarda le modalità di comunicazione e collaborazione all'interno del team di sviluppo, inclusi incontri regolari, strumenti di comunicazione, e pratiche di condivisione delle informazioni.
- Include l'approccio alla pianificazione del progetto, la gestione delle risorse, il monitoraggio e il controllo delle attività, e la gestione dei rischi.

35 Web App

La web app è un'applicazione software che viene eseguita su un browser web e che interagisce con gli utenti attraverso l'interfaccia utente fornita dal browser stesso.