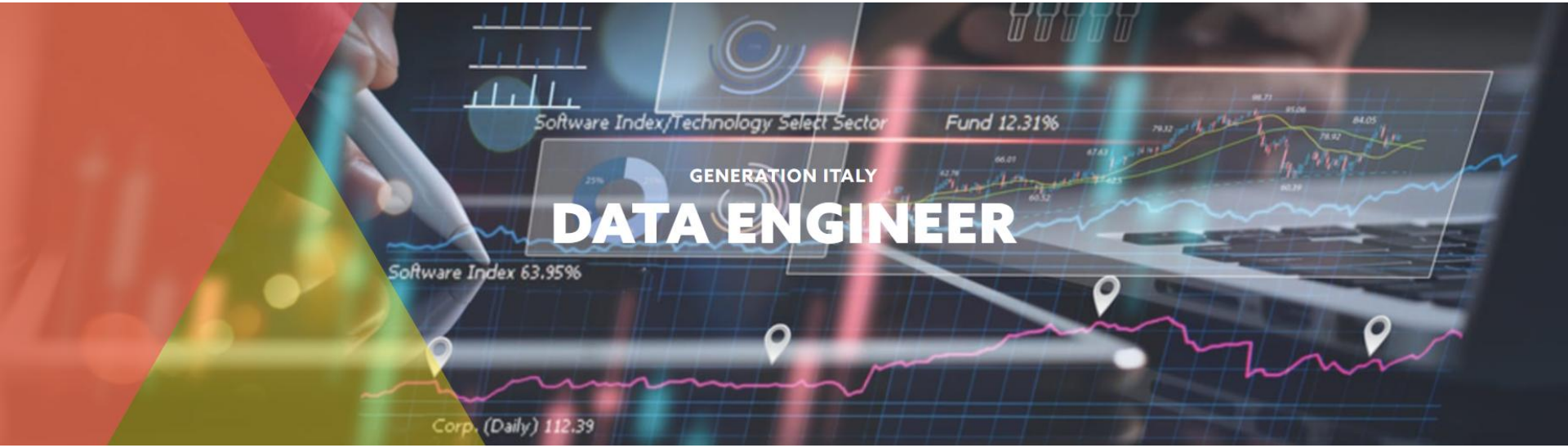
ACCADEMIA DEL LEVANTE
WWW.ACCADEMIADELLEVANTE.ORG

Generation ITALY

IASEM
Istituto Alti Studi Euro Mediterranei

Data Engineer



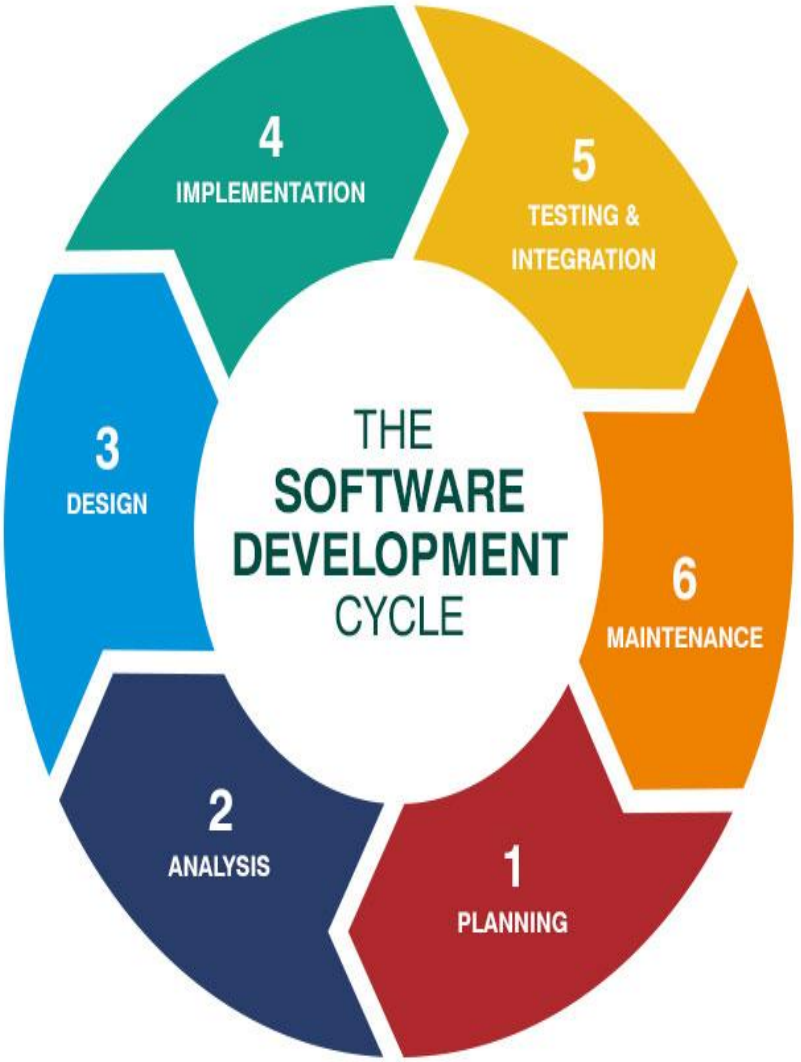
Ciclo di Vita del Software

Franchini Roberto

Fasi del Ciclo di Vita del software: Metodologie di Gestione e Strumenti di Supporto

SDLC

Ciclo di vita dello sviluppo software è l'applicazione di pratiche aziendali standard atte alla creazione di applicazioni software. Solitamente è suddiviso in sei-otto fasi: pianificazione, requisiti, progettazione, creazione, documentazione, test, distribuzione, manutenzione.



Project Workflow

1. Pianificazione
2. Analisi
3. Design
4. Implementazione
5. Testing
6. Integrazione
7. Manutenzione

Student Management System

❖ Roles

1. User(i.e. Studenti)
2. Insegnanti
3. Admin

❖ Admin + Insegnanti

1. **Dashboard**: In questa sezione, l'amministratore può vedere tutti i dettagli in breve come lezioni totali, studenti totali, avvisi di classe totali e avvisi pubblici totali.
2. **Class**: In questa sezione, l'amministratore può gestire la classe (New/Update/Delete).
3. **Students**: In questa sezione, l'amministratore può gestire gli studenti(New/Update/Delete).
4. **Notices:** In this section, the admin can manage notices (New/Update/Delete).
5. **Public Notices:** In questa sezione, l'amministratore può gestire gli avvisi pubblici.
6. **Pages:** In questa sezione, l'amministratore può gestire le pagine "about us" e "contatti"
7. **Search:** In questa sezione, l'amministratorecan ricerca gli studenti tramite il loro id.
8. **Reports:** In questa sezione, l'amministratore può visualizzare quanti studenti si sono registrati e in quale periodo .
9. L'amministratore può aggiornare il suo profilo, cambiare e/o recuperare la password .

❖ User (Studenti):

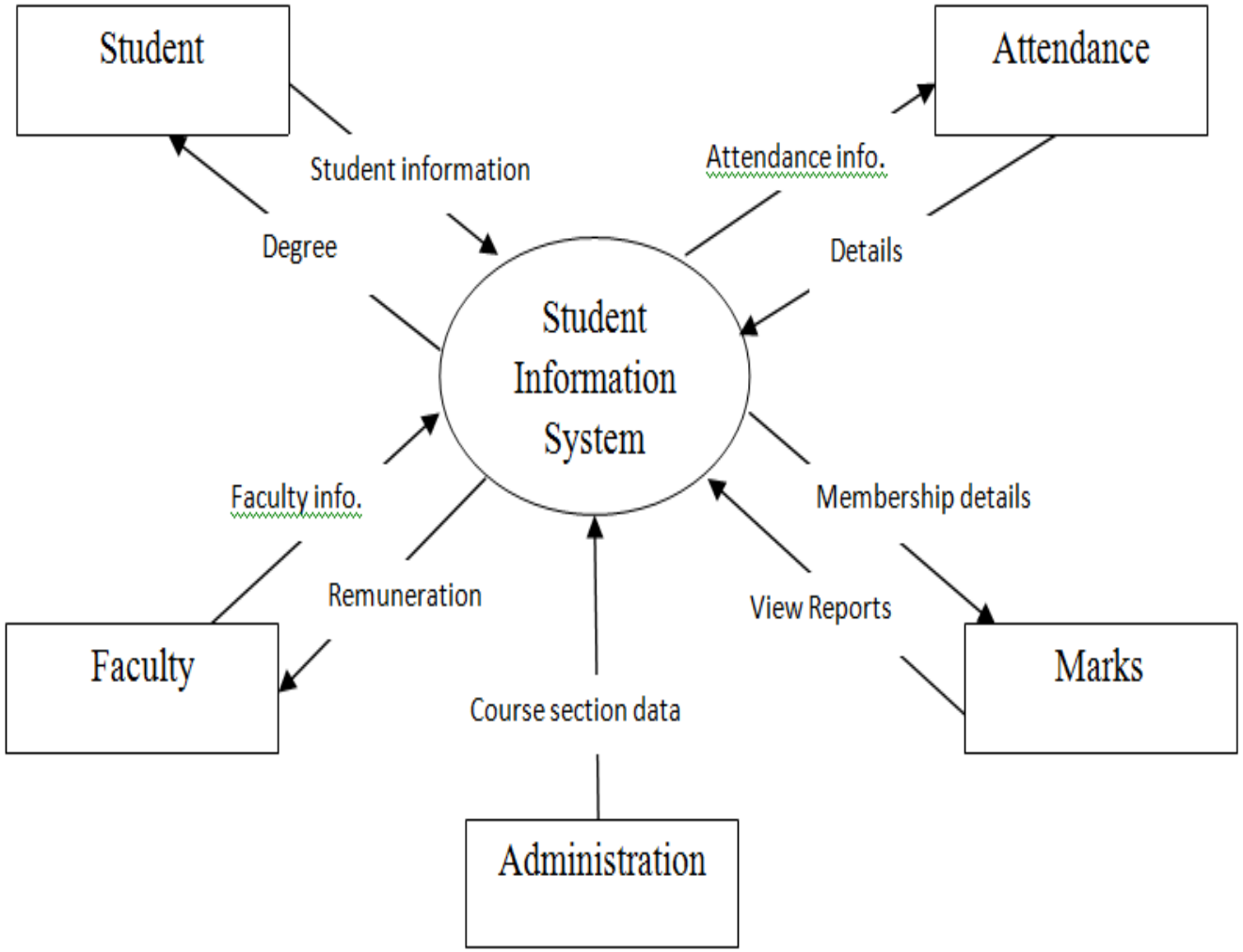
1. **Dashboard**: La pagina di benvenuto di ogni utente.
2. **View Notices**: In questa sezione, l'utente può visualizzare gli avvisi che sono emanati dall'amministratore.
3. Gli studenti possono visualizzar eil loro profilo, modificare e/o recuperare la password..

Sistema di gestione Studenti– Specifiche tecniche

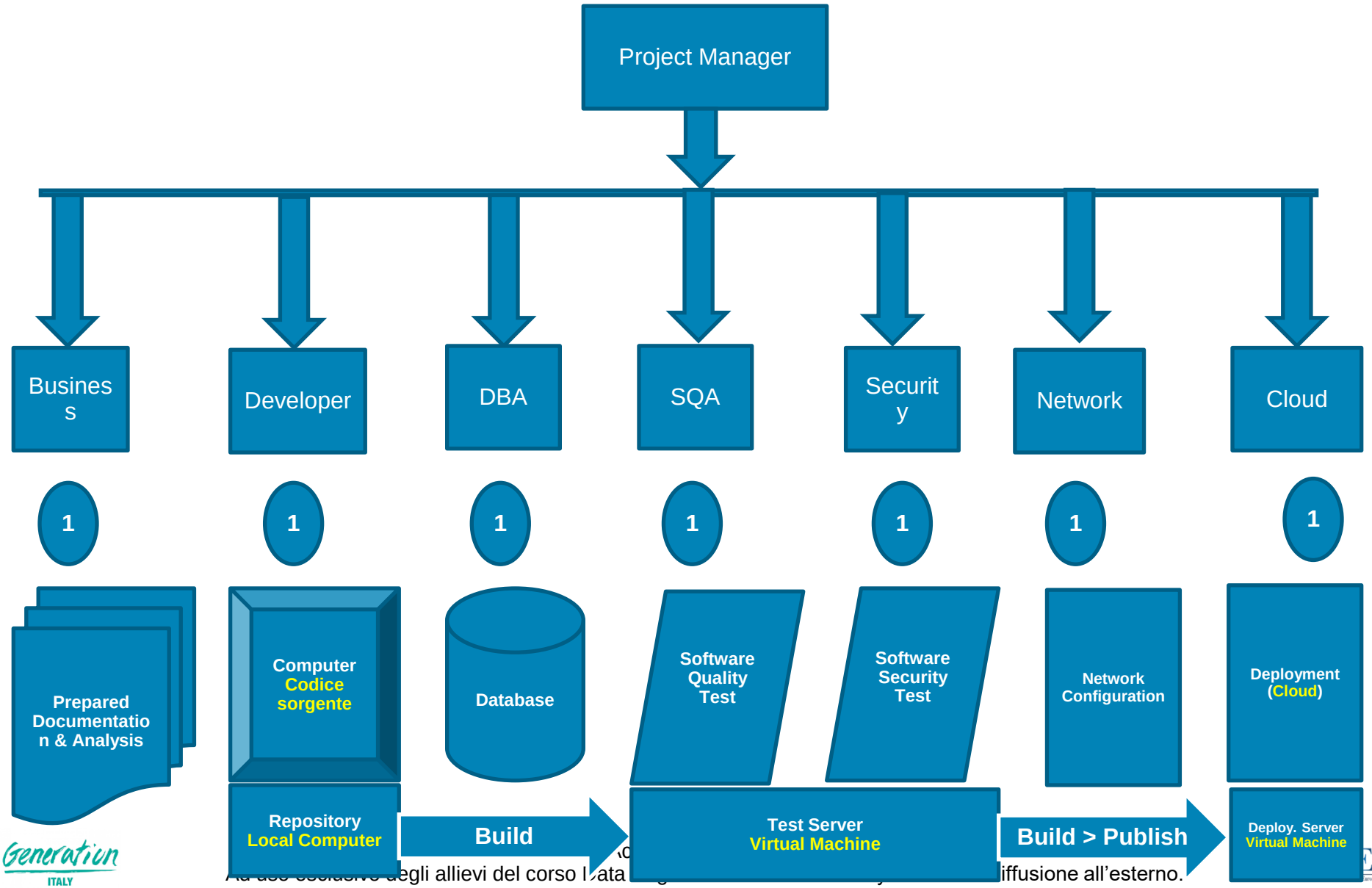
- User Interface(UI): HTML, AJAX,JQUERY,JAVASCRIPT
- Database: MySQL
- Programming Language: PHP
- PHP Framwork: Laravel
- Sever: 8 Core Linux Server

Ecc, ecc, ecc…

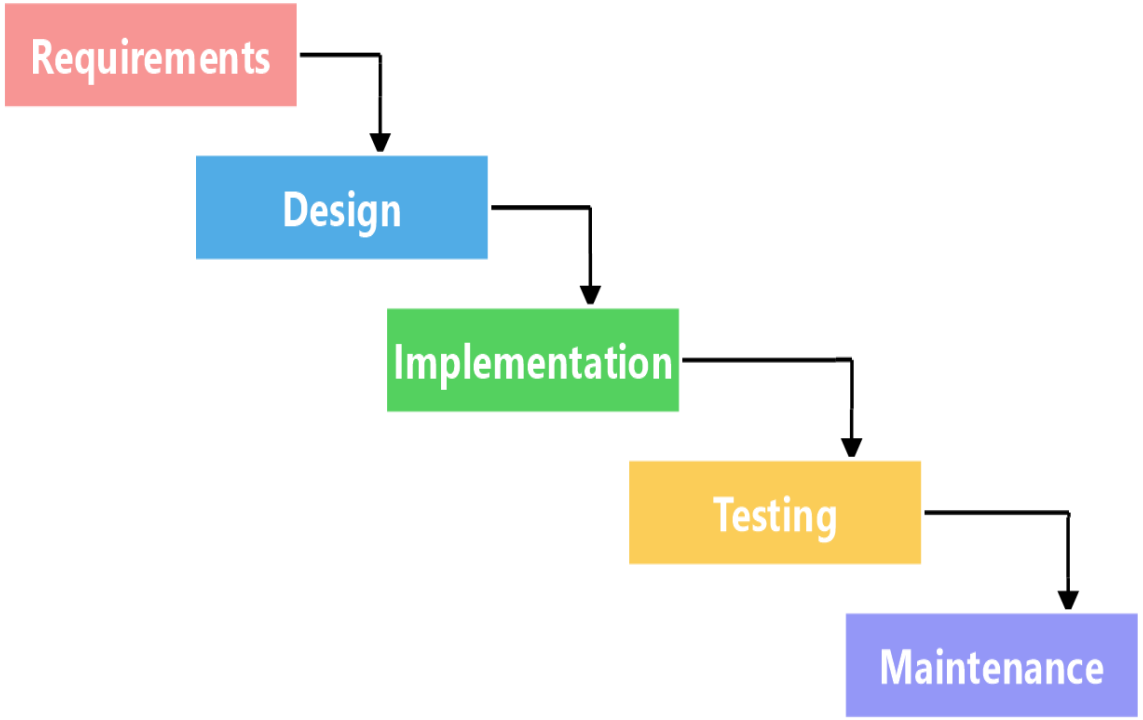
Sistema di gestione Studenti(Design)



Membri del team di progettazione



Metodologia a cascata (Waterfall)



Metodologia del Waterfall Model, nota anche come Liner Sequential Life Cycle Model. Il modello a cascata è seguito in ordine sequenziale, quindi il team di sviluppo del progetto passa alla fase successiva di sviluppo o test solo se il passaggio precedente è stato completato con successo.

Vantaggi del Waterfall Model

- È uno dei modelli più facili da gestire. Per sua natura, ogni fase prevede risultati specifici e un processo di revisione.
- Funziona bene per progetti di piccole dimensioni in cui i requisiti sono facilmente comprensibili.
- Consegna rapida del progetto
- Il processo e i risultati sono ben documentati.
- Metodo facilmente adattabile
- Questa metodologia di gestione del progetto è utile per gestire le dipendenze.

Un grandissimo numero di progetti è stato realizzato tramite il Waterfall Model

Ma cosa accade se…?

- Il cliente cambia alcuni requisiti?
- Si rileva un problema di bug/vulnerabilità?
- Il cliente vuole aggiungere un ulteriore modulo/funzionalità?
- Si vuole altro….

Waterfall Model methodology is unable to solve this types of problems!

Limiti del Waterfall Model

- Non è un modello ideale per un progetto di grandi dimensioni
- Se il requisito non è chiaro all'inizio, il metodo è meno efficace.
- Molto difficile tornare indietro per apportare modifiche alle fasi precedenti.
- Il processo di test inizia una volta terminato lo sviluppo. Pertanto, è molto probabile che i bug vengano rilevati più avanti nello sviluppo e siano costosi da risolvere.

Agile

Agile è un approccio iterativo alla gestione dei progetti e allo sviluppo software che aiuta i team a fornire valore ai propri clienti più velocemente e con meno problemi.

Vantaggi dell' Agile Model

- È focalizzato sul processo del cliente. Quindi, fa in modo che il cliente sia continuamente coinvolto durante ogni fase.
- I team agili sono estremamente motivati e auto-organizzati, quindi è probabile che forniscano un risultato migliore dai progetti di sviluppo.
- Il metodo di sviluppo software agile garantisce il mantenimento della qualità dello sviluppo
- Il processo è completamente basato sul progresso incrementale. Pertanto, il cliente e il team sanno esattamente cosa è completo e cosa no. Ciò riduce il rischio nel processo di sviluppo.

Precedenti problemi? Risolti!

- Cambiamenti di requisiti voluti dal client epossono essere facilmente e velocemente integrati.
- bug/problemi di vulnerabile dell'applicazione sono velocemente risolti.
- È possibile aggiungere ulteriori moduli a progetto in corso modulo

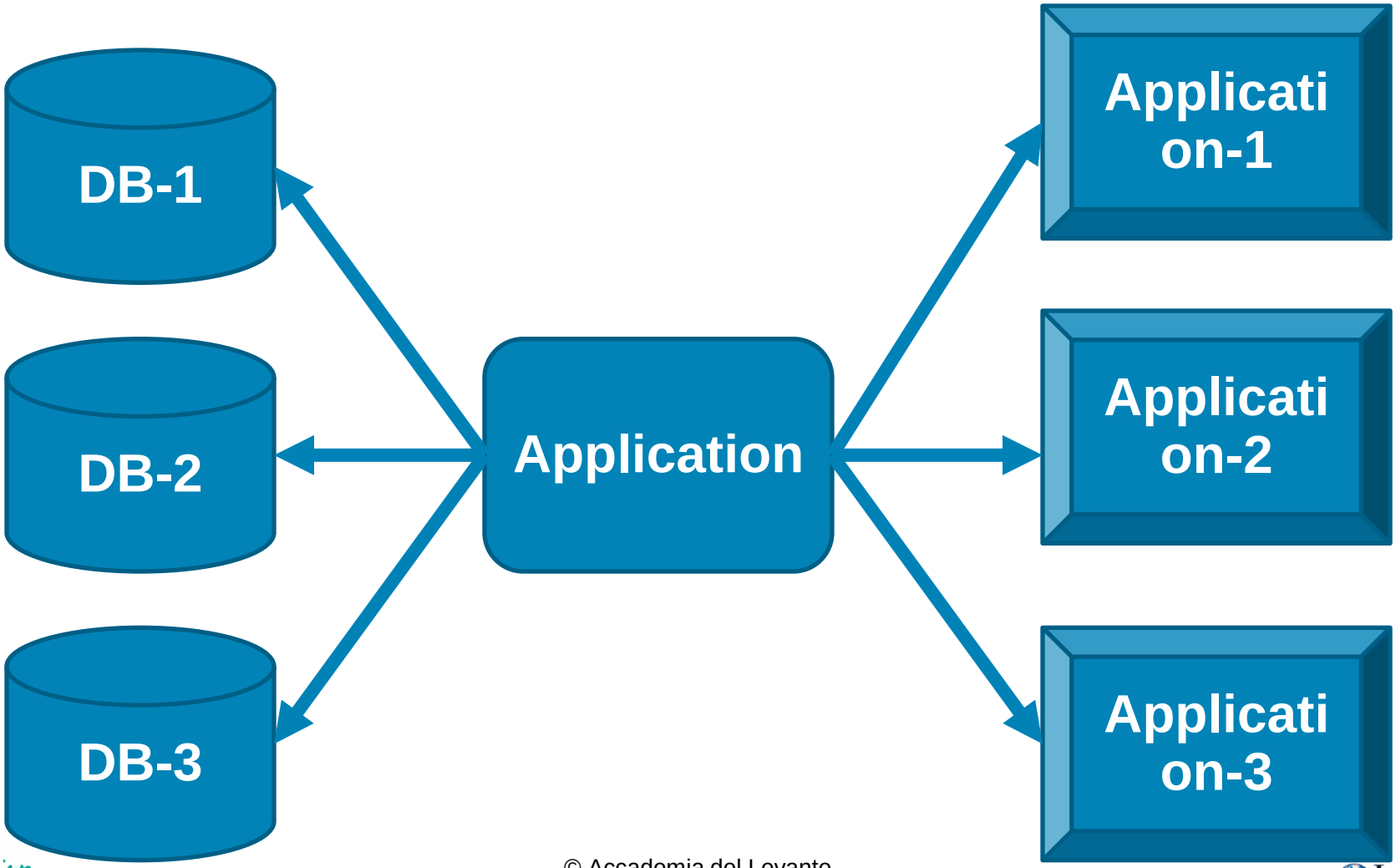
Agile methodology risolve i problemi del waterfall model!

Scenario

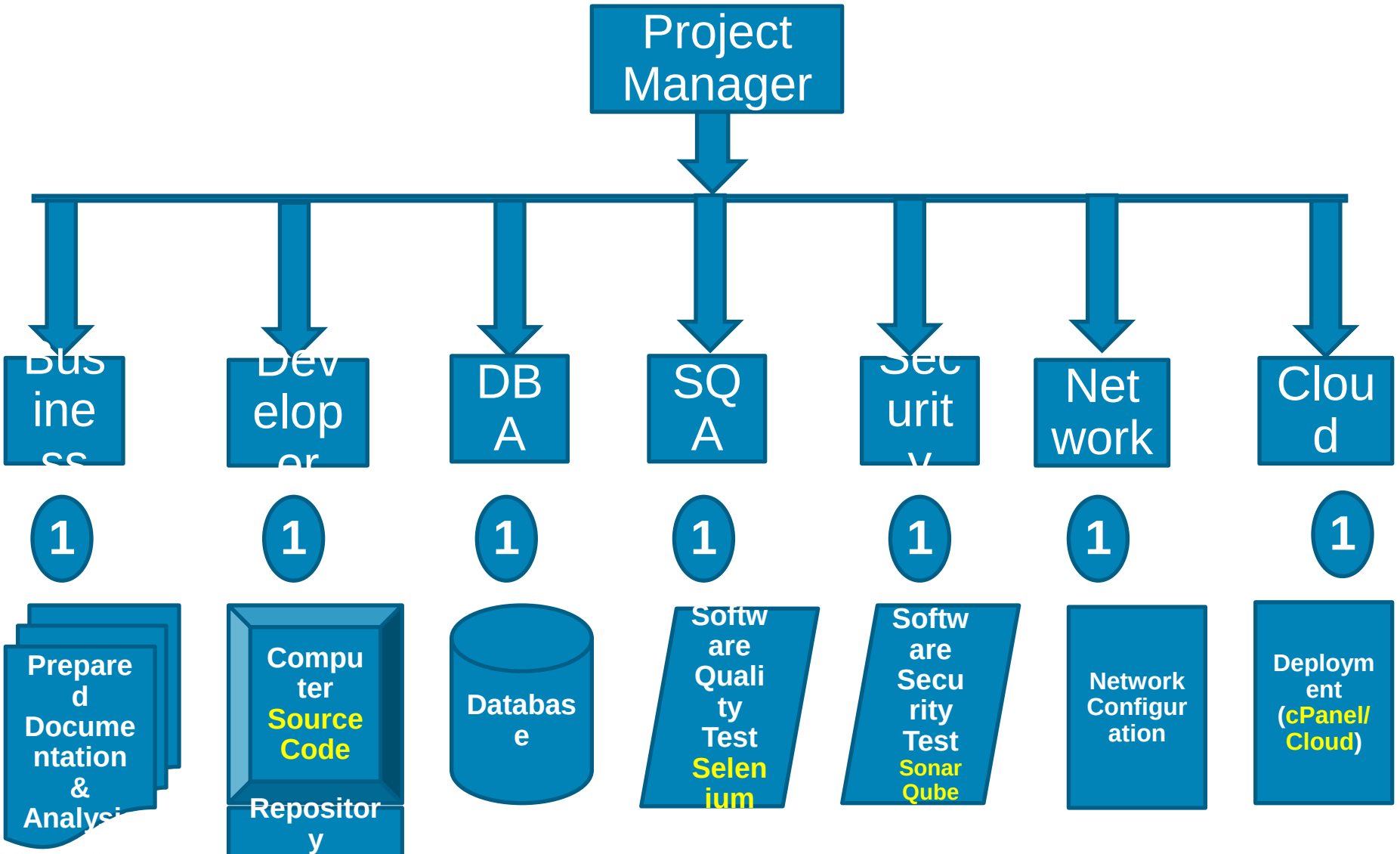
- Molte scuole sono interessate al “**School Management System**”
- La vostra application business e stata alalrgata
- Vi piacerebbe implementare il vostro business come un **SaaS** Model.

Cosa è il SaaS Model?

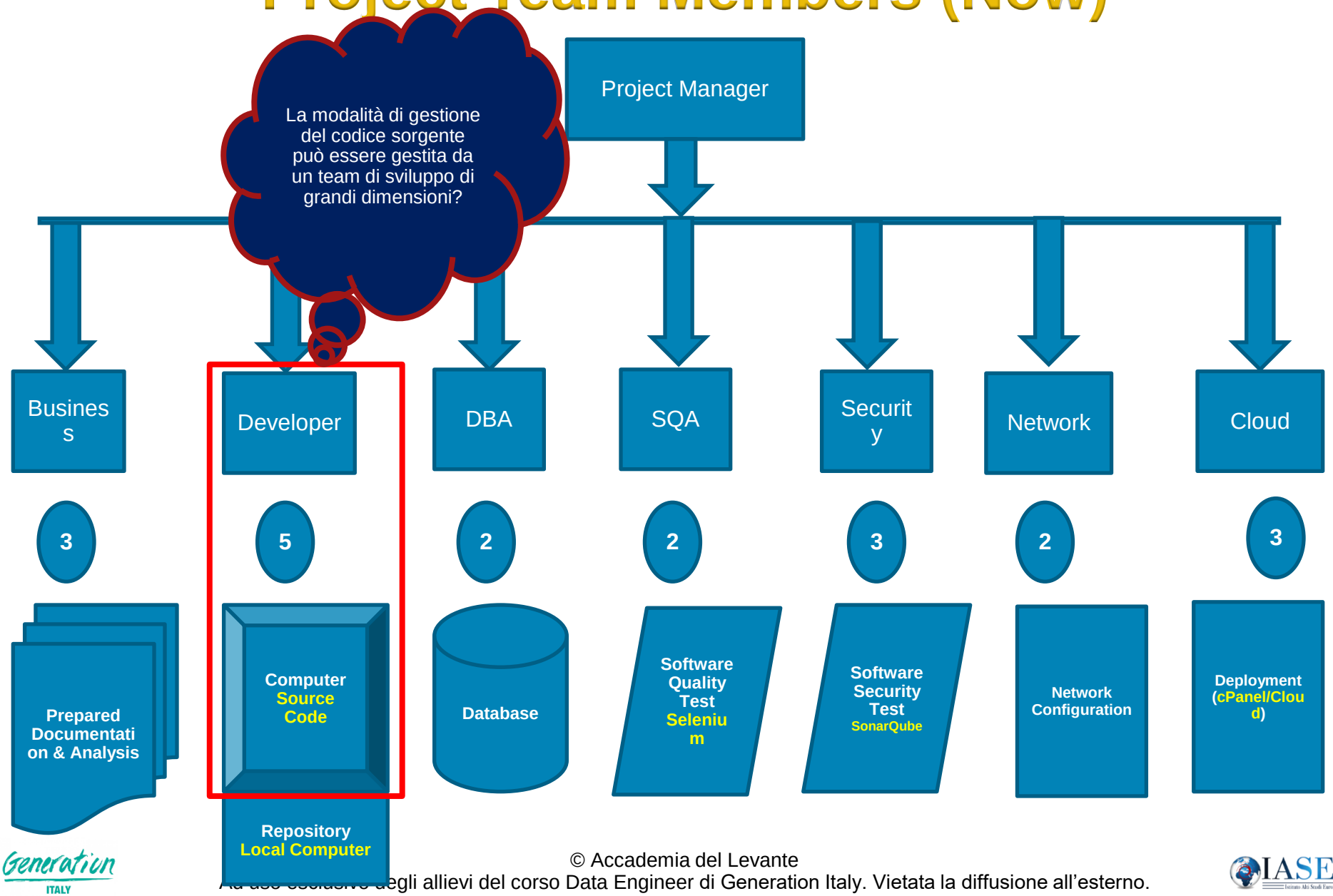
Software as a Service (SaaS) Model
(Software come un Servizio)



Project Team Members (Prev.)



Project Team Members (Now)



Perchè il sistema di controllo versione è important?

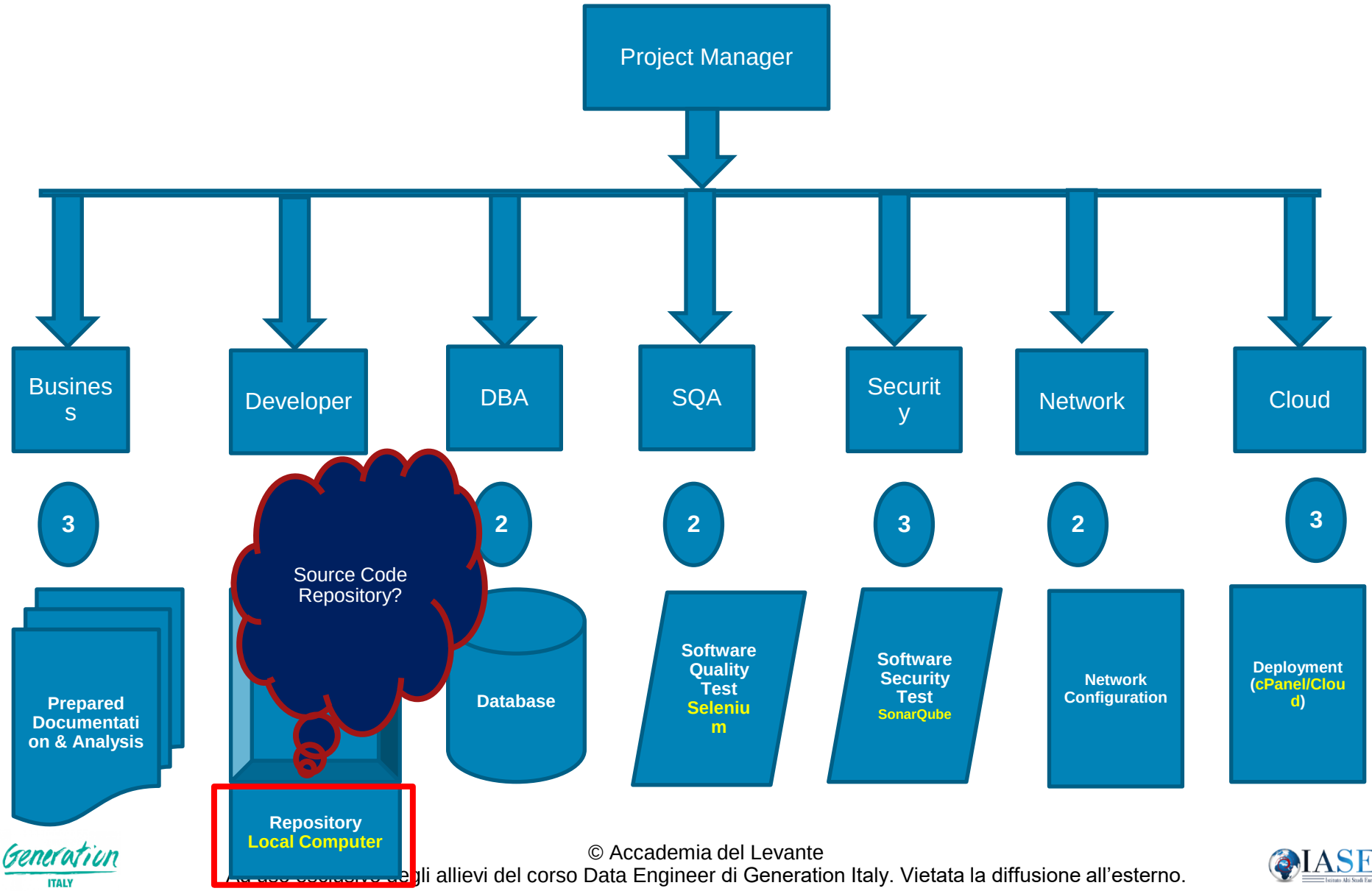
Poiché sappiamo che un prodotto software è sviluppato in collaborazione da un gruppo di sviluppatori, questi potrebbero trovarsi in luoghi diversi e ognuno di essi contribuisce ad alcuni tipi specifici di funzionalità/caratteristiche. Quindi, per contribuire al prodotto, hanno apportato modifiche al codice sorgente (aggiungendo o rimuovendo linee di codice). Un sistema di controllo della versione è un tipo di software che aiuta il team di sviluppatori a comunicare e gestire (tracciare) in modo efficiente tutte le modifiche apportate al codice sorgente insieme alle informazioni su chi ha apportato e quali modifiche sono state apportate

Il sistema di controllo della versione tiene traccia delle modifiche apportate a un particolare software e scatta un'istantanea di ogni modifica.

Sistemi di controllo delle versioni

Il controllo della versione (Version Control), noto anche come controllo del codice sorgente, è la pratica di tenere traccia e gestire le modifiche al codice sorgente. I sistemi di controllo della versione sono strumenti software che aiutano i team software a gestire le modifiche al codice sorgente nel tempo. Con l'accelerazione degli ambienti di sviluppo, i sistemi di controllo delle versioni aiutano i team software a lavorare più velocemente e in modo più intelligente.

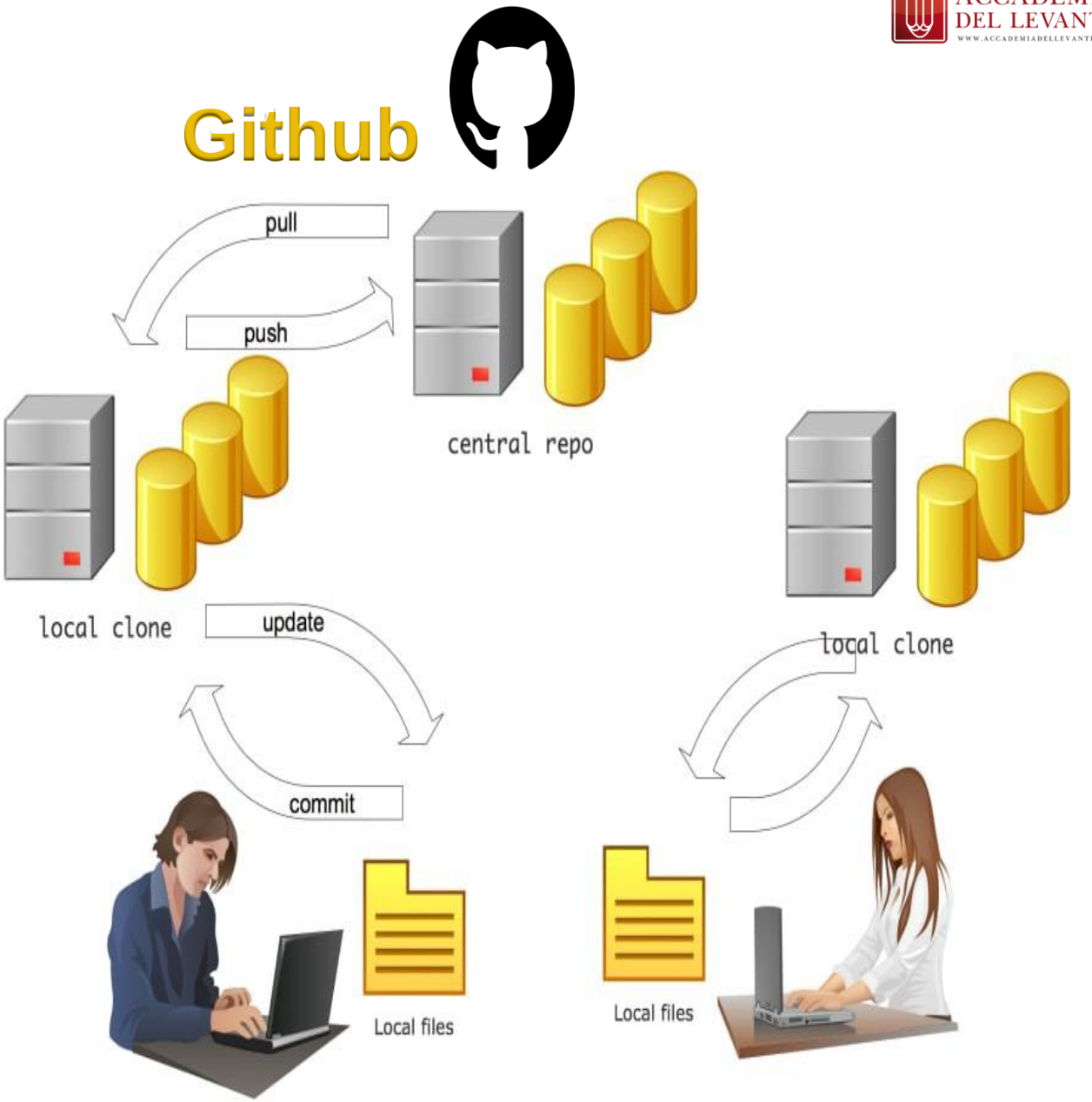
Membri del team di progettazione (ora)



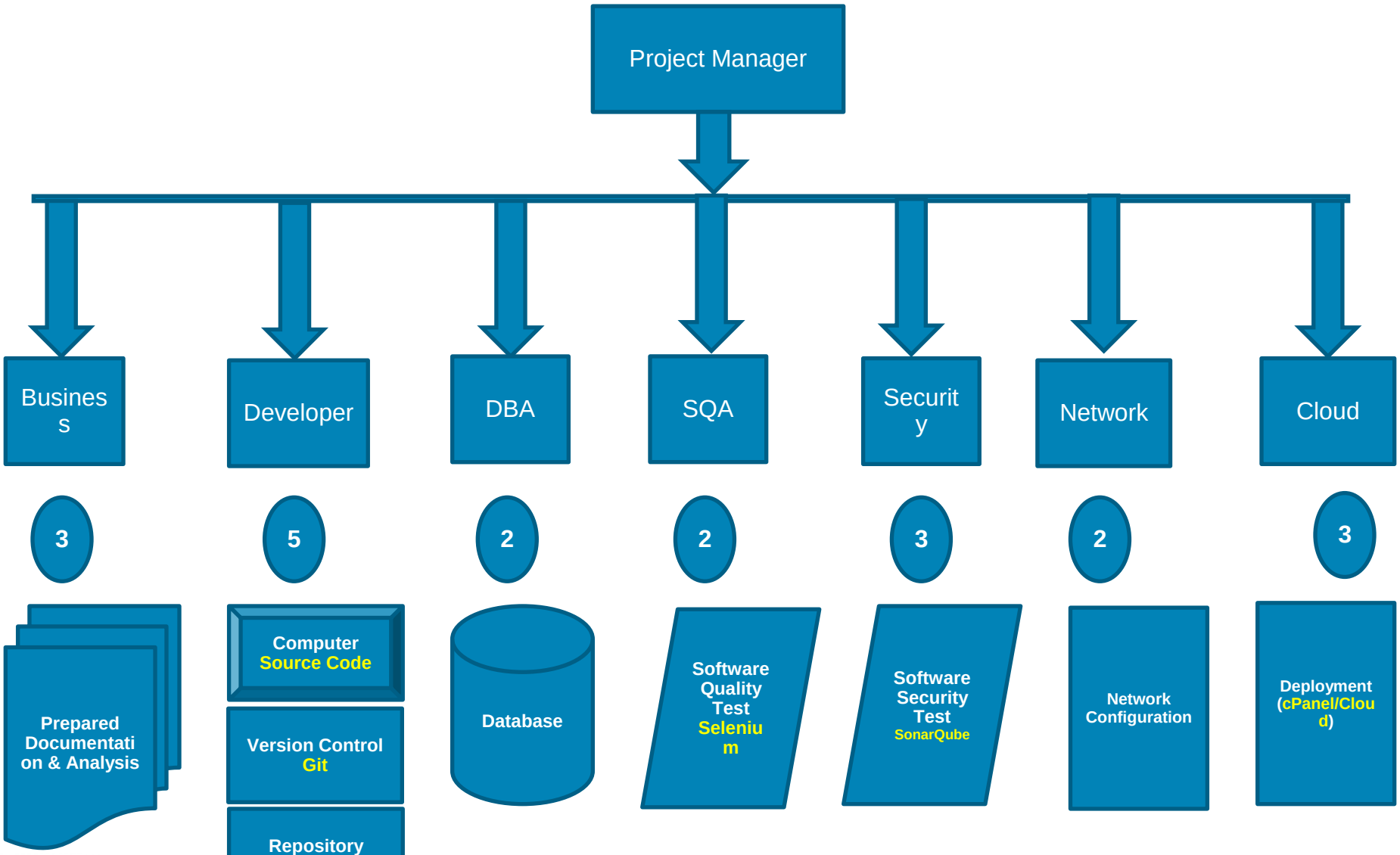
Repository di codice (SCM)

Un repository di codice (Source Code Repository) è un archivio del codice stesso su cui si sta lavorando. Oltre al codice stesso, si può conservare nel repository elementi come documentazione, note, pagine Web e altri elementi. Per qualsiasi progetto di sviluppo software di successo è necessario un repository di codice.

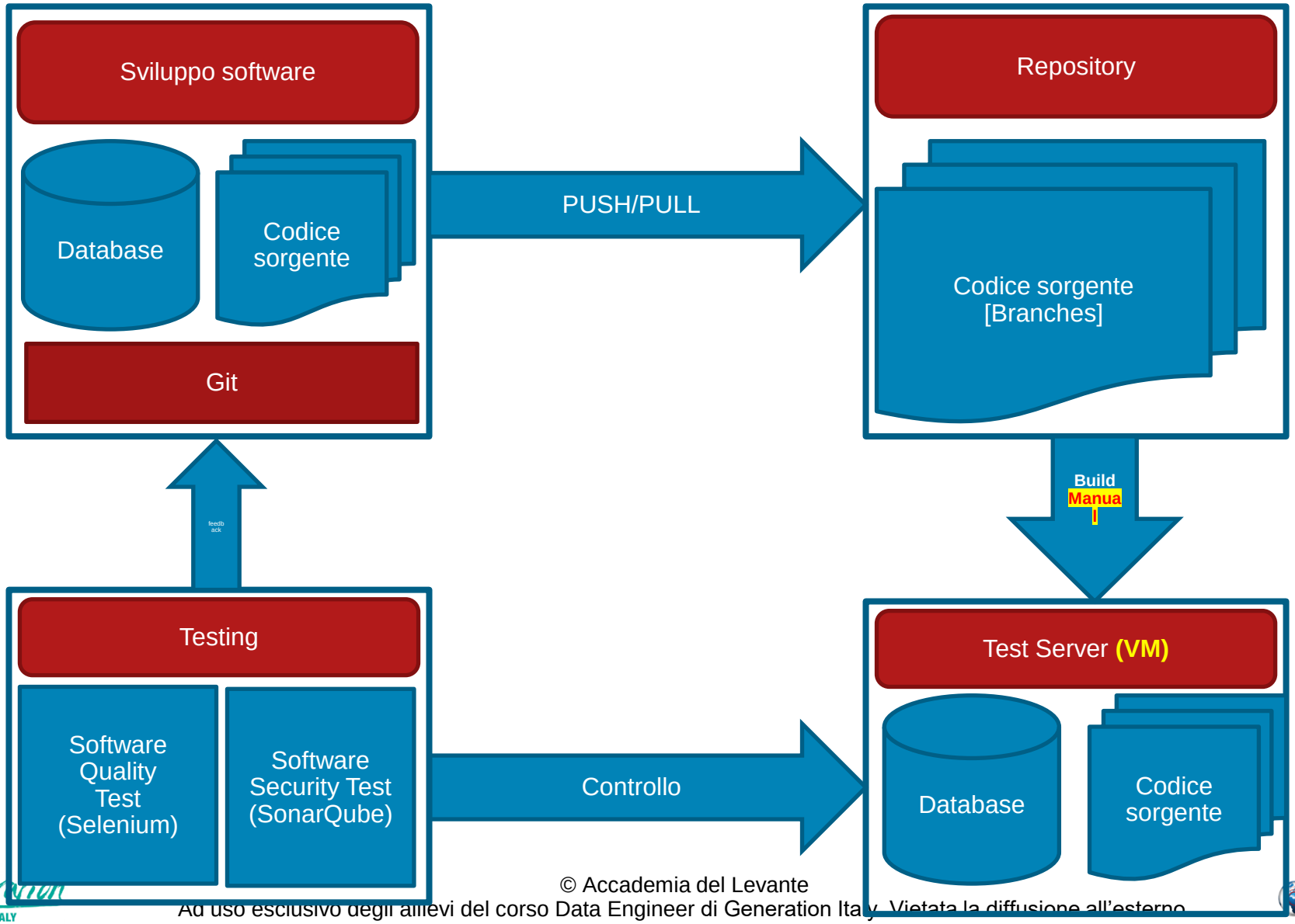
GitHub, Inc. è un fornitore di hosting Internet per lo sviluppo di software e il controllo della versione tramite Git. Offre la funzionalità di controllo della versione distribuita e di gestione del codice sorgente (SCM) di Git, oltre alle proprie funzionalità. Fornisce il controllo degli accessi e diverse funzionalità di collaborazione come il rilevamento dei bug, le richieste di funzionalità, la gestione delle attività, l'integrazione continua e i wiki per ogni progetto.



Membri del team di progettazione



Ciclo di sviluppo



In Real Development Project

- Ogni sprint, Può essere implementato con **migliaia di** Features, Richieste di modifica, Bug, Tasks, Defect-testing e problemi correlati alla vulnerabilità.
- In tal modo ogni giorno possono essere effettuati **centinaia** di commit di codice sorgente.
- Ma risulta difficile dover costruire **manualmente** la **build** dopo ogni commit del codice sorgente

Per risolvere quest'ultimo problema, introduciamo la **CI/CD pipeline automatizzata**

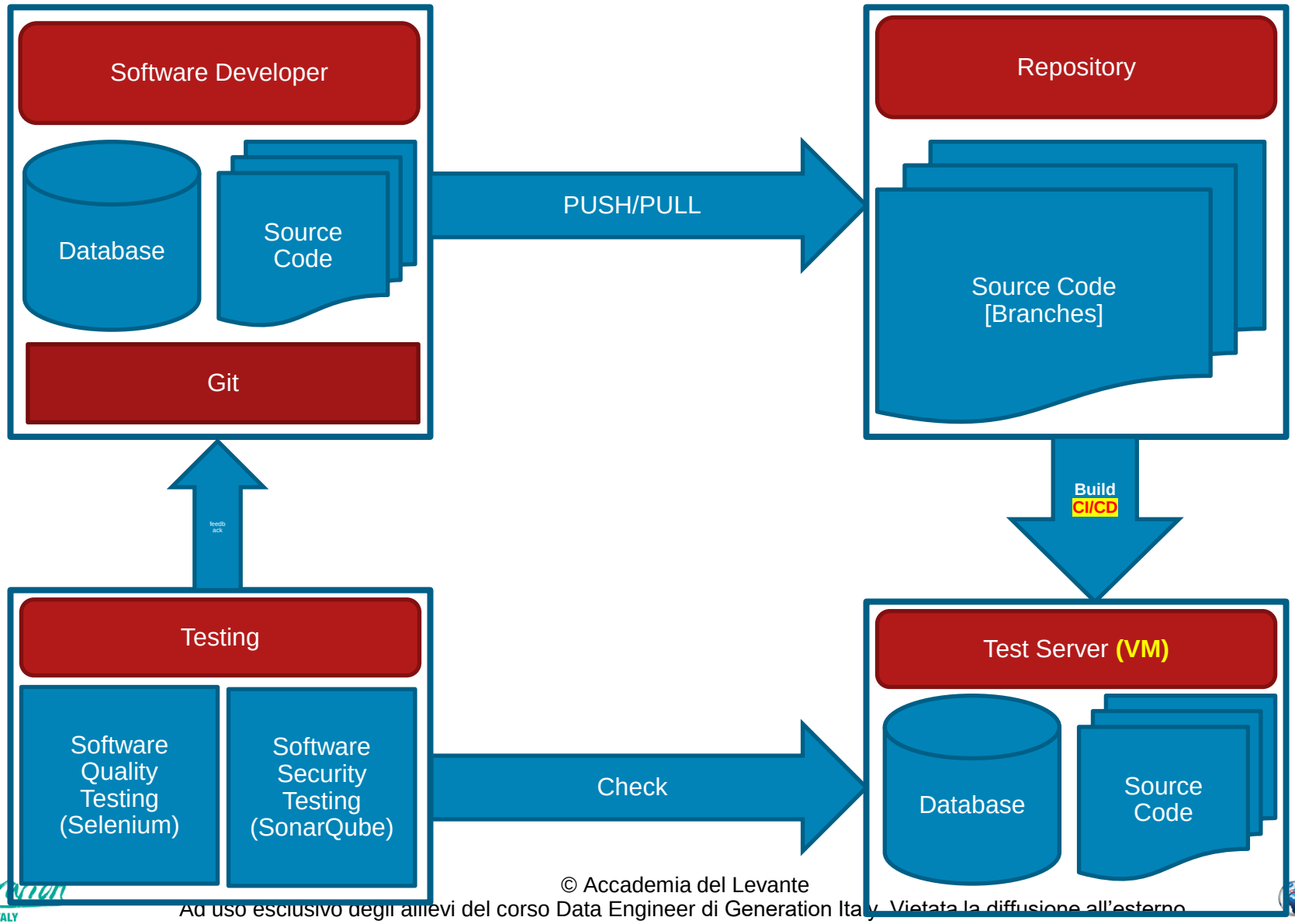
CI/CD Pipeline automatizzata

Un processo CI/CD automatizza la fase di delivery del software. Le pipeline automatizzate rimuovono gli errori manuali e forniscono cicli di feedback standardizzati.

Continuous Integration (CI)
Continuous Delivery/ Deployment(CD)

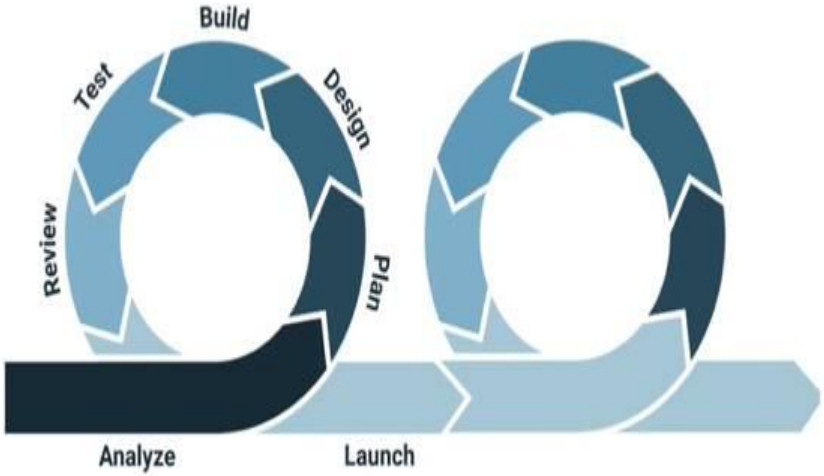
Some of the CI/CD tools

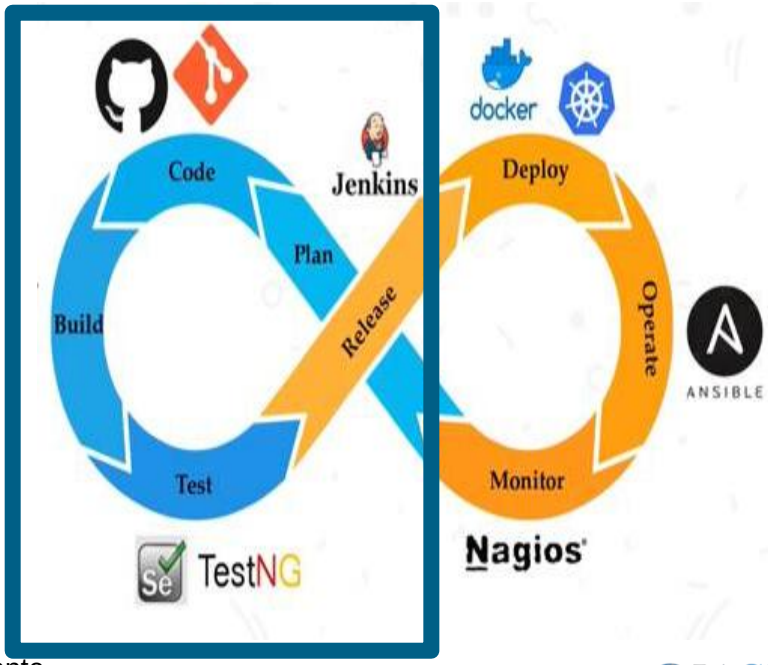
Development Cycle



ACCADEMIA DEL LEVANTE
WWW.ACCADEMIADELLEVANTE.ORG

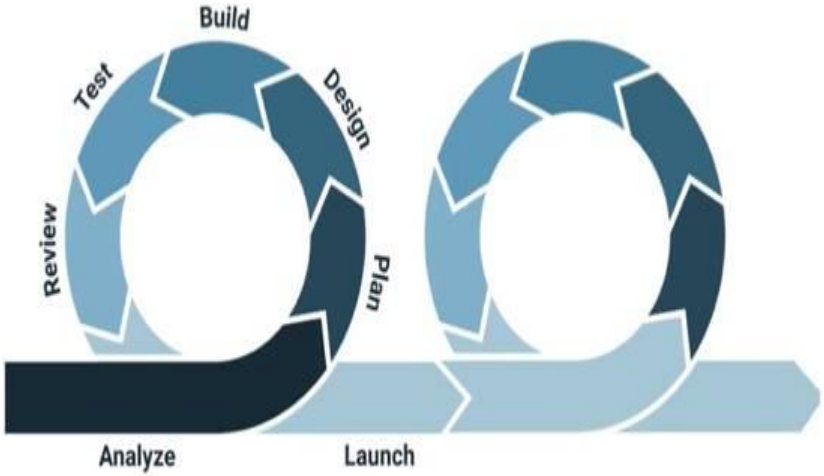
Fino ad ora abbiamo completato... la metà del DevOps



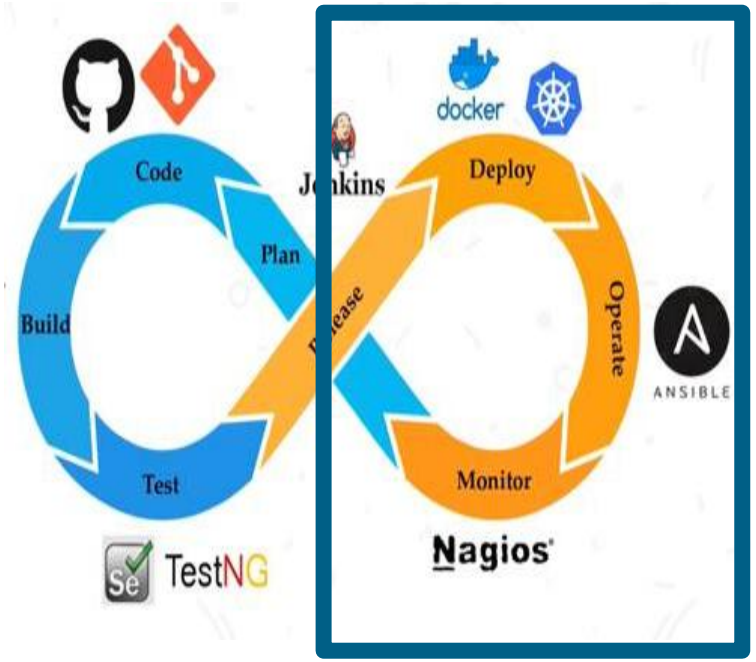


Questo è solo la Dev – Part del DevOps
Dev = Development

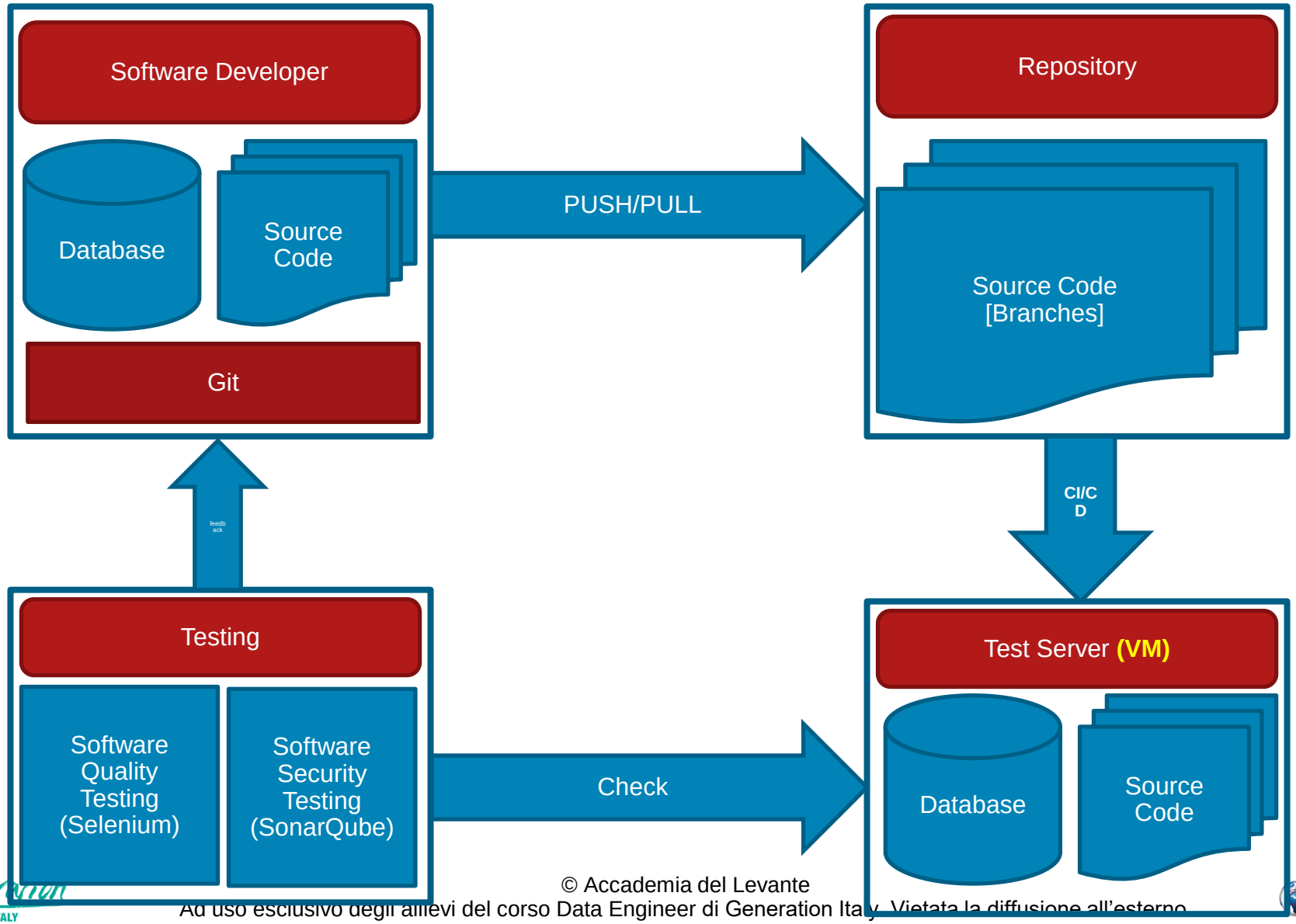
Spostiamoci alla – Ops (Operations) part del DevOps



Passiamo alla Ops – Part del DevOps
Ops = Operations

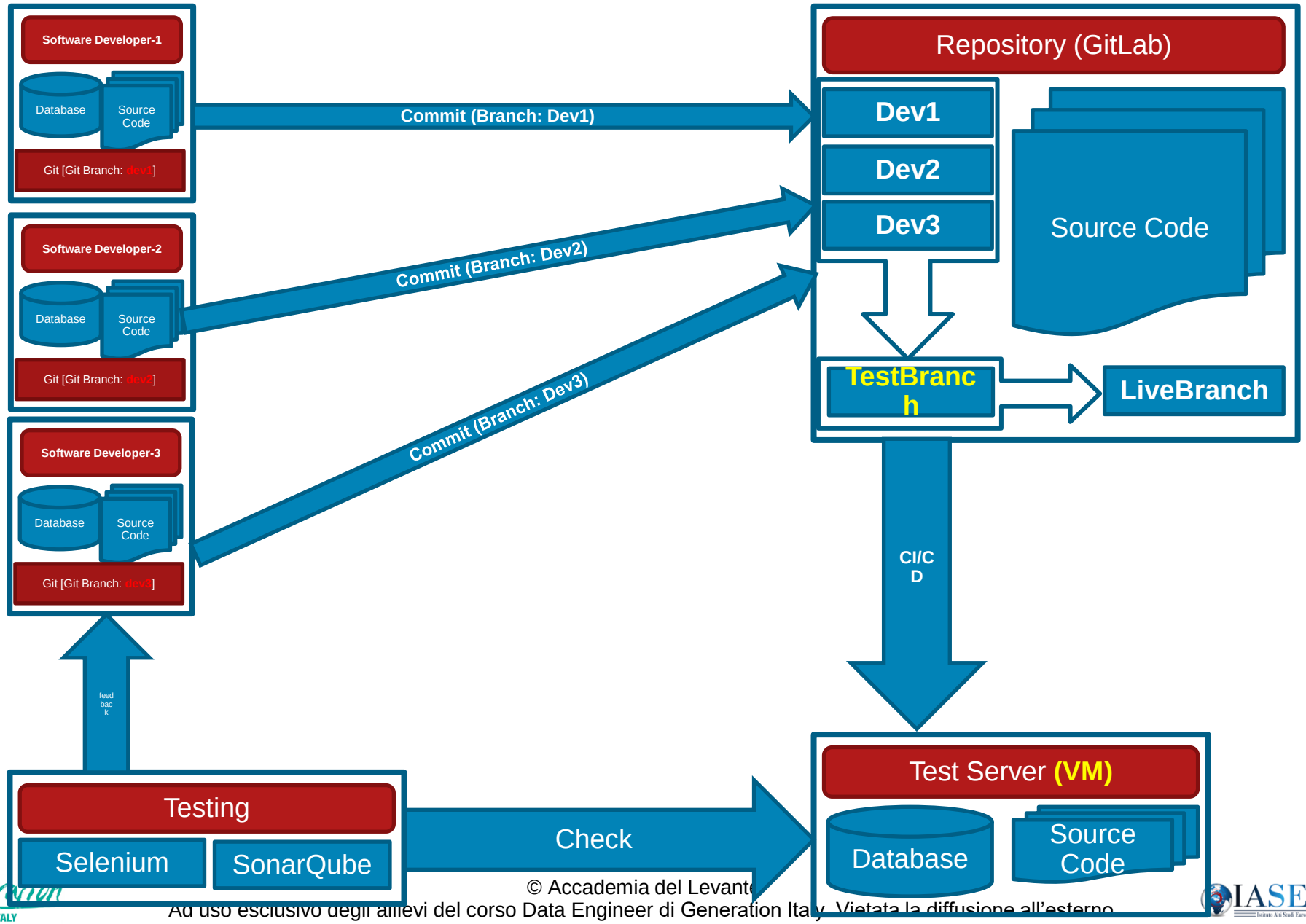


Development Cycle

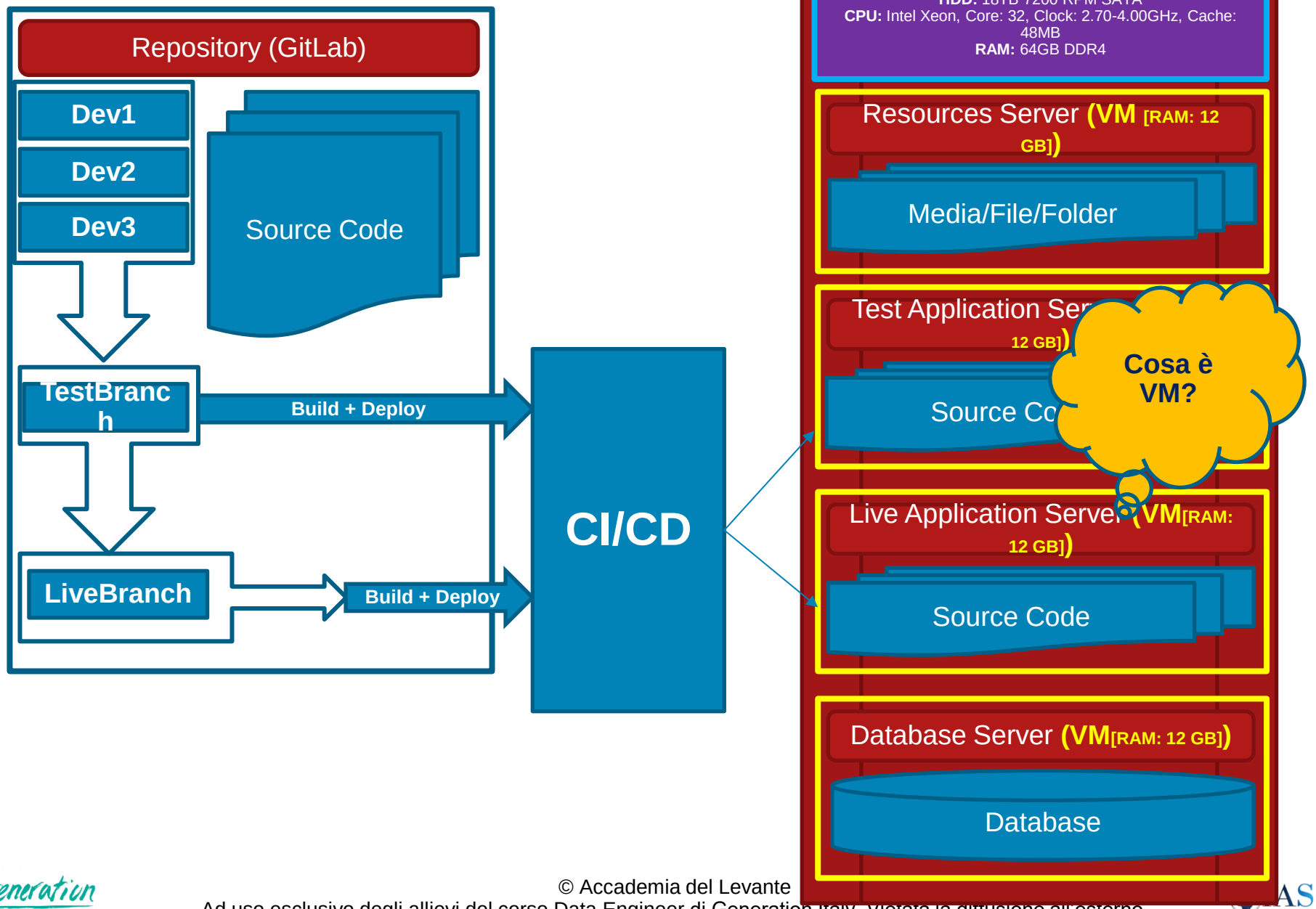


ACCADEMIA DEL LEVANTE
WWW.ACCADEMIADELLEVANTE.ORG

Diamo un'occhiata da vicino…Development Cycle!

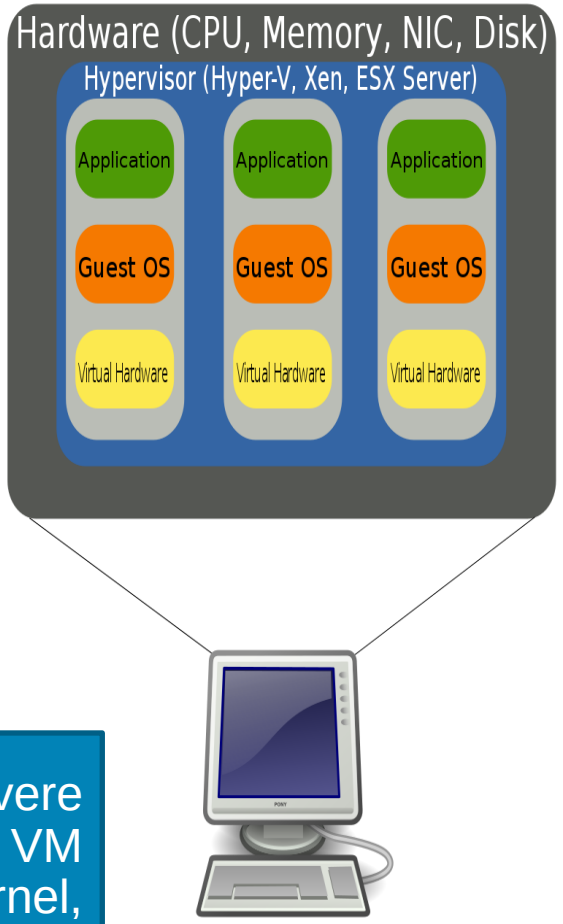


Development Cycle



Cosa è una VM (Virtual Machine)?

Una **macchina virtuale (VM)** è una risorsa di calcolo che utilizza software anziché un computer fisico per eseguire programmi e distribuire app. Ogni macchina virtuale esegue la propria CPU, memoria, interfaccia di rete e spazio di archiviazione, creati su un sistema hardware fisico (localizzato o fuori sede). Le macchine virtuali funzionano sul software hypervisor, che imita l'infrastruttura fisica e divide le risorse in più macchine virtuali. L'hypervisor viene anche definito macchina host o monitor della macchina virtuale.



Le VM tendono ad essere ingombranti e ad avere dimensioni di molti gigabyte perché ciascuna VM contiene il proprio sistema operativo guest, kernel, file binari, librerie e la relativa applicazione.

Introduciamo il concetto di container invece della VM

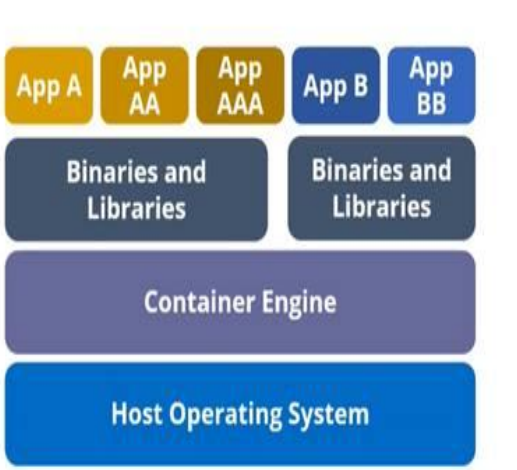
- I contenitori creano ambienti isolati in un server fisico virtualizzando il sistema operativo host ed eseguendo su di esso applicazioni pacchettizzate.

- Invece di virtualizzare l'hardware come le macchine virtuali, i container virtualizzano il sistema operativo. È costruito su un kernel del sistema operativo host e di solito condivide le sue librerie e i suoi file binari..

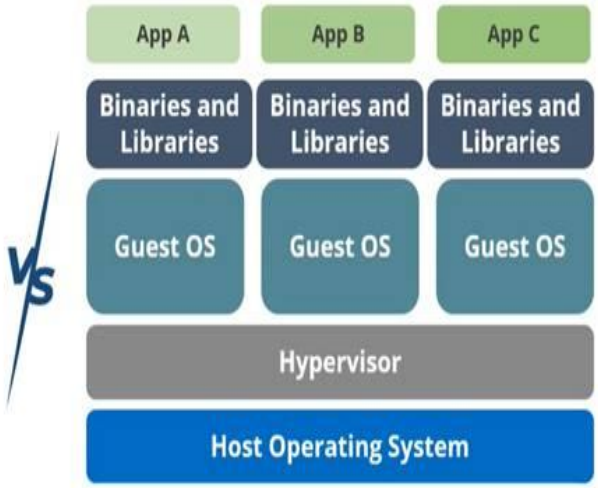
- Poiché condivide la maggior parte delle sue necessità, i contenitori impacchettano solo l'applicazione e le sue dipendenze. Sono molto più leggeri delle VM e hanno una dimensione di soli megabyte.

ACCADEMIA DEL LEVANTE
WWW.ACCADEMIADELLEVANTE.ORG

Container vs. VM



Containerization



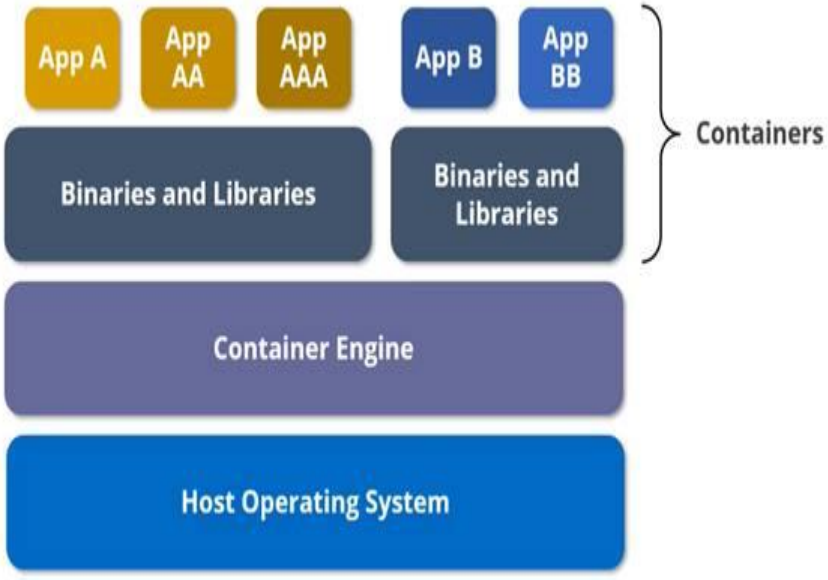
Virtualization

Virtualization	Containerization
Virtual Machines	Containers
Multiple OS on a single server	Single OS on a single server
Replicates real hardware	OS-level virtualization
Heavyweight	Lightweight
Complete isolation from host OS	Less isolation from host OS
VMs have their own OS	Containers share the same OS that is the host OS
Comparatively more start time	Very less start time
Not portable	Portable

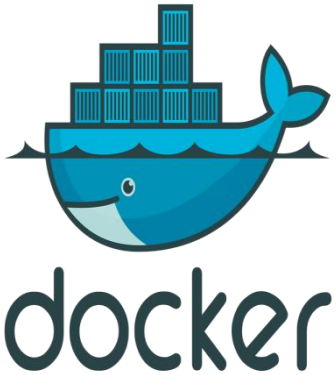
Container

Un container è un pezzo di software che impacchetta il codice e tutte le sue dipendenze.

La containerizzazione è il processo di confezionamento del codice software insieme a tutti i suoi componenti essenziali.



Popular container providers



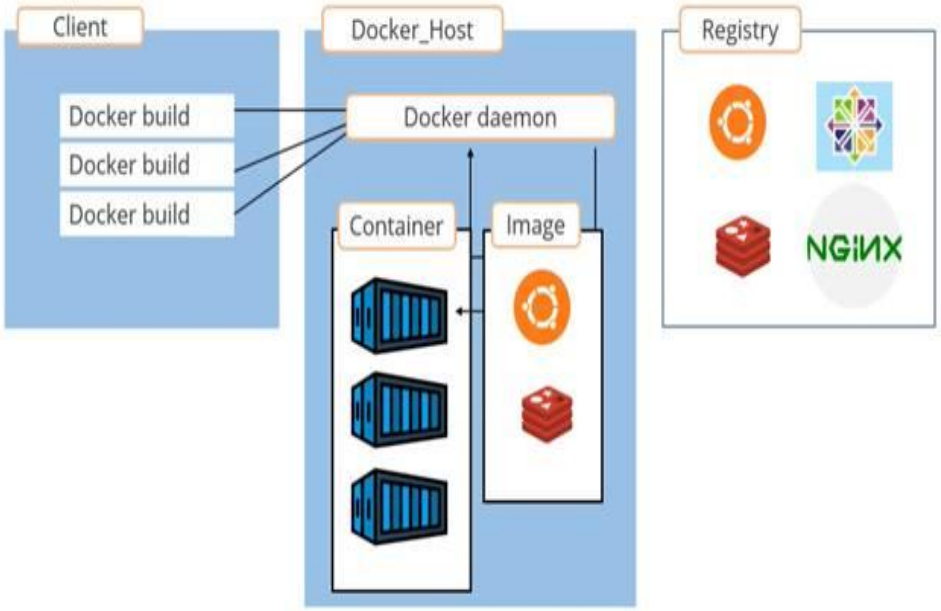
Docker

Docker è una piattaforma di containerizzazione per creare pacchetti della tua applicazione insieme a tutte le sue dipendenze.

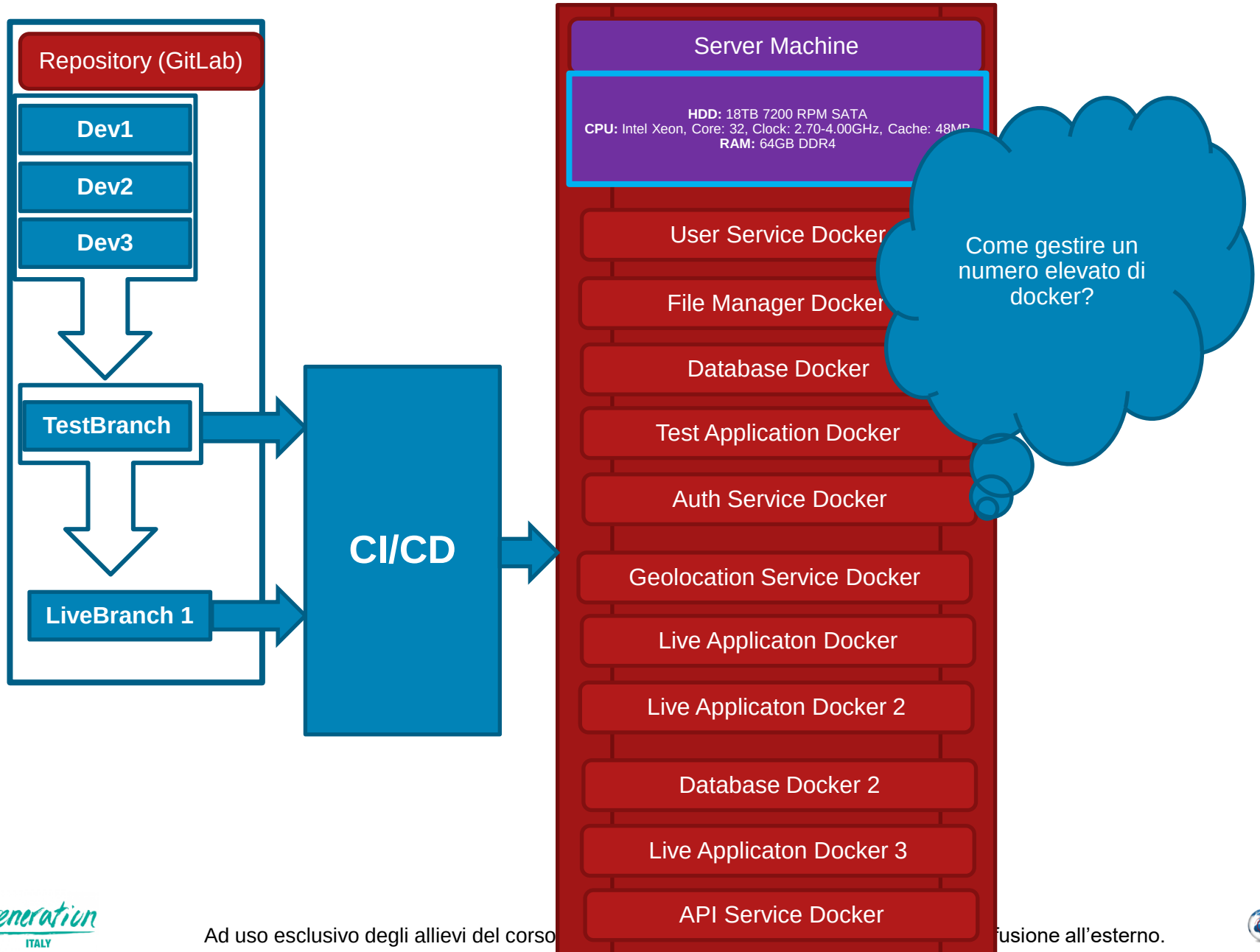
Docker è una piattaforma gratuita e aperta per lo sviluppo, la distribuzione e l'esecuzione di software.

Vantaggi di Docker: alcuni dei vantaggi offerti da Docker nelle varie fasi del ciclo di vita dello sviluppo del software (SDLC) sono i seguenti:

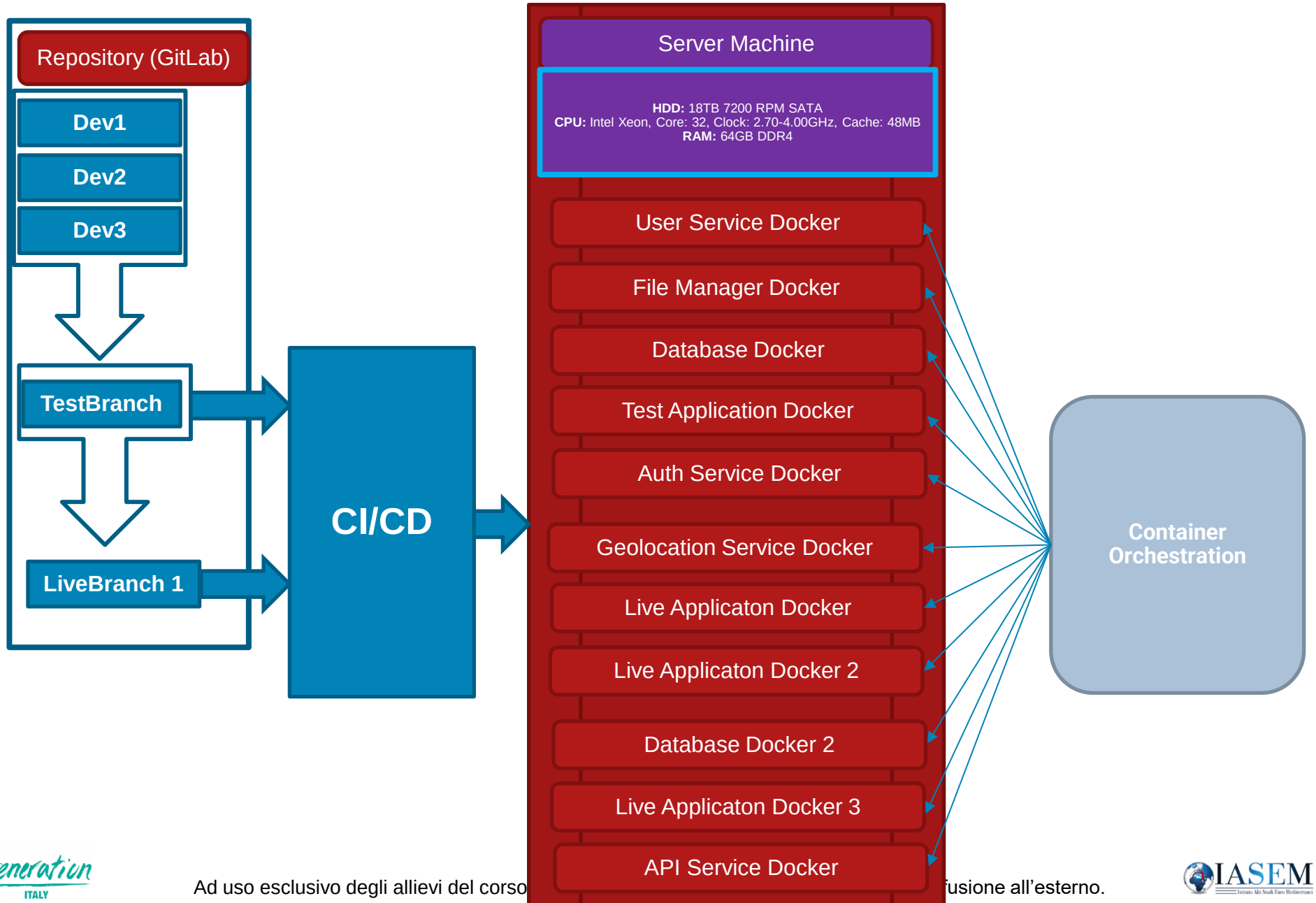
1. **Build**
2. **Test**
3. **Deploy**
4. **Maintain**



Development Cycle (Container)



Development Cycle (Con't)



Container Orchestration

Container Orchestration automatizza la distribuzione, la gestione, il dimensionamento e il networking dei containers. È utile per le aziende distribuire e gestire più containers e host.

Scopo dell'uso degli Orchestratori di Containes:

Un orchestratore di containersdistribuisce e gestisce automaticamente le app inserite in contenitori.

1. Risponde dinamicamente ai cambiamenti.
2. Garantisce che tutte le istanze di container distribuite vengano aggiornate se viene rilasciata una nuova versione di un servizio.
    I. Risponde dinamicamente ai cambiamenti
    II. Distribuisce la stessa applicazione in ambienti diversi.

Container Orchestration Tools

Gli strumenti di orchestrazione dei containers forniscono un framework per la gestione dei containers e dell'architettura dei microservizi su larga scala. Semplificano la gestione dei containers e forniscono un quadro per la gestione di più containers come un'unica entità. Alcuni strumenti popolari utilizzati per [illegible] sono:

Kubernetes

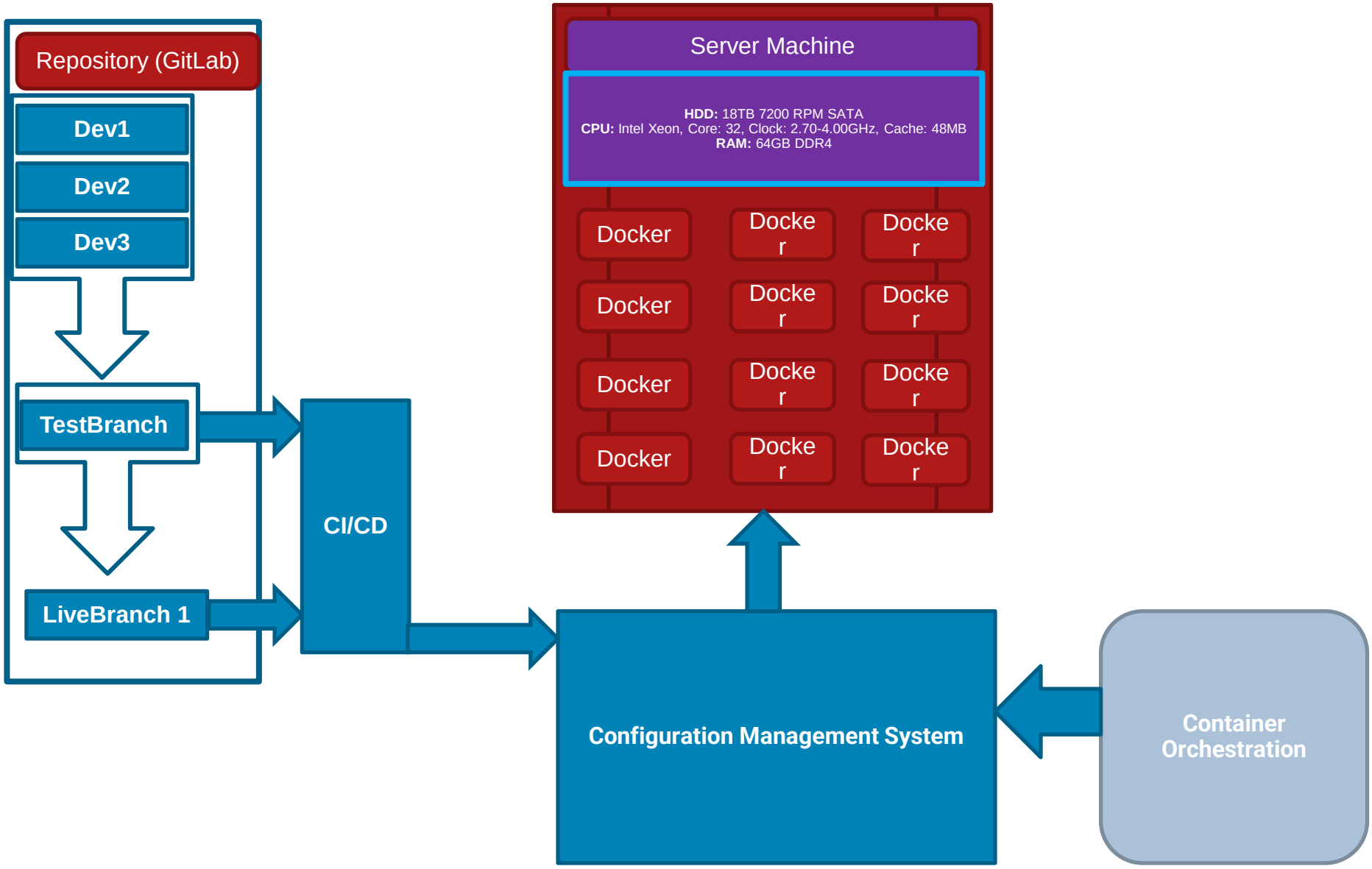
Docker Swarm

Apache Mesos

OPENSHIFT

HashiCorp Nomad

Development Cycle (Con't)



Configuration Management

Gestione della configurazione: è un metodo di ingegneria del sistema che garantisce che le caratteristiche di un prodotto rimangano coerenti durante il suo ciclo di vita. Può coprire risorse non IT e prodotti lavorativi utilizzati per sviluppare servizi. Fornisce un modello di configurazione dei servizi, degli asset e delle infrastrutture registrando la relazione tra-

- Service assets
- Elementi di configurazione
- Controlled environments
- Operational use

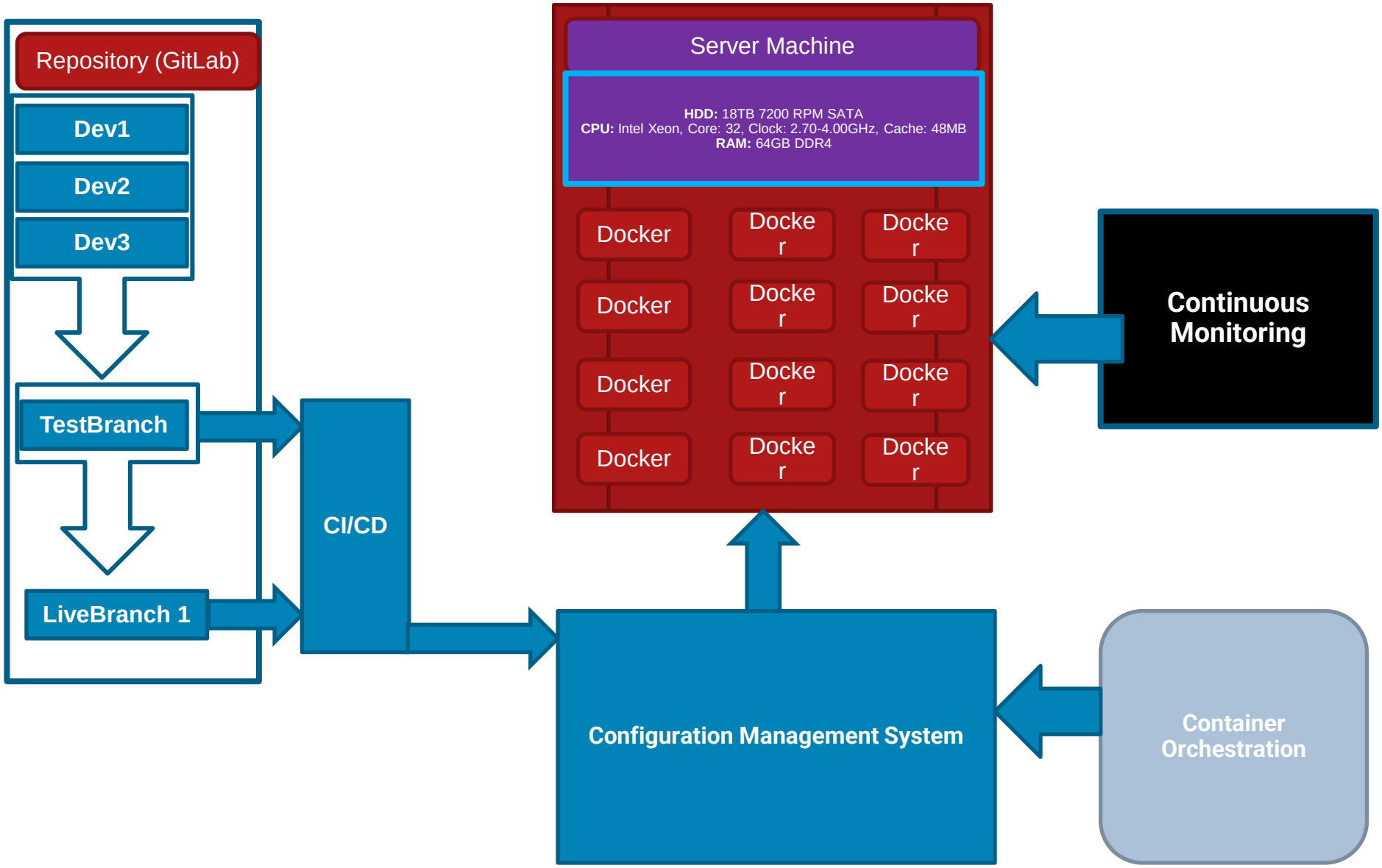
Qualsiasi cambiamento nella configurazione può impattare in modo importante su:

(1) Performance (2) Security (3) Functionality

Vantaggi della configuration management

- Aumentare l'efficienza con un processo di configurazione ben definito che migliora la visibilità.
- Ottimizzare i costi avendo una conoscenza dettagliata di tutti gli elementi IT.
- Tenere traccia dei requisiti, dalle specifiche ai test.
- Identificare e controllare le versioni del software.
- Migliorare l'affidabilità del sistema e del processo grazie agli effetti di rilevamento.
- Gestire le informazioni sull'elemento di configurazione.
- Forniree un ripristino più rapido del servizio in caso di errore del processo.
- Facilitare la conduzione degli audit di configurazione funzionale.

Development Cycle



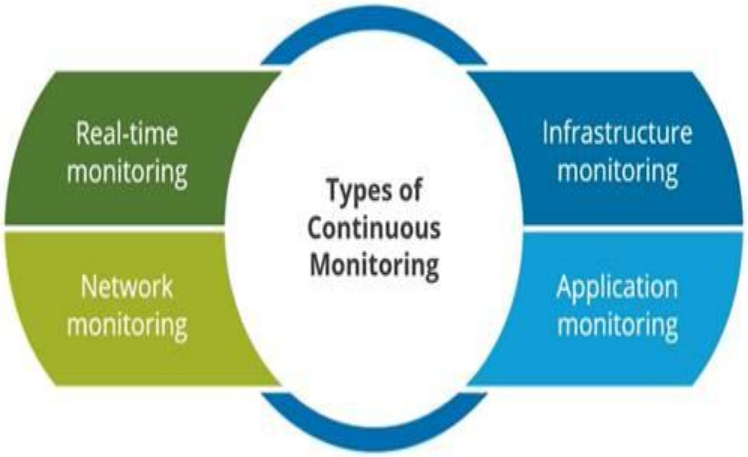
Continuous Monitoring

Il Continuous Monitoring è la capacità di rilevare rischi, conformità e problemi di sicurezza in un ambiente operativo. Funziona come uno strumento di controllo, in cui può navigare tra i vecchi dati di monitoraggio per analizzare e migliorare le prestazioni del sistema.

Le regole del Continuous Monitoring sono:

- Aiuta nella progettazione di un sistema affidabile.
- Visualizza il comportamento dell'applicazione durante le ore lavorative di punta.
- Riduce i costi acquisendo una conoscenza precisa della duplicazione delle risorse software.
- Riduce la possibilità che un'applicazione venga interrotta.
- Avvisa se c'è un problema con il servizio dell'applicazione.
- Recupera e analizza i dati storici.

Tipologia di Continuous Monitoring



ACCADEMIA DEL LEVANTE
WWW.ACCADEMIADELLEVANTE.ORG

Monitoring

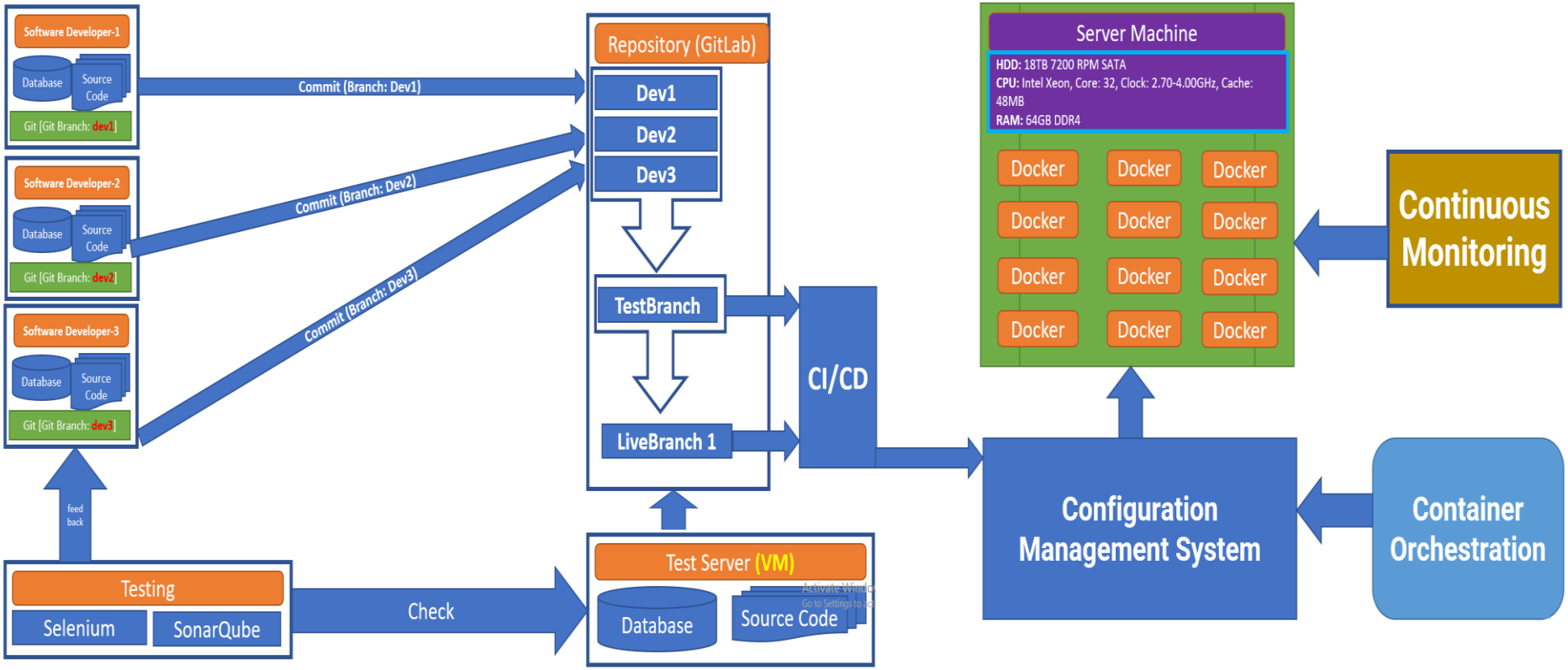
API success or failure

API HTTP error codes

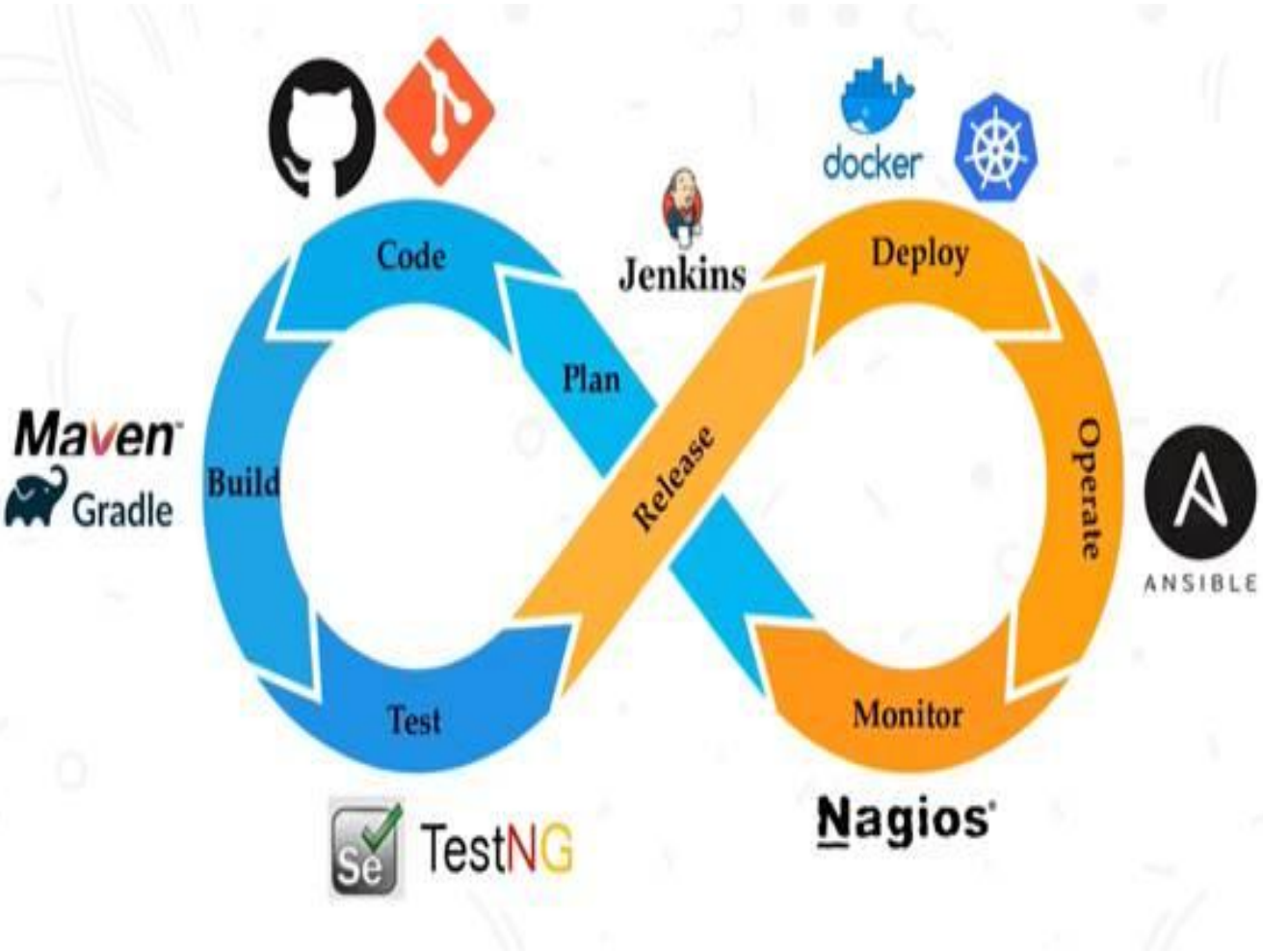
Performance metrics count

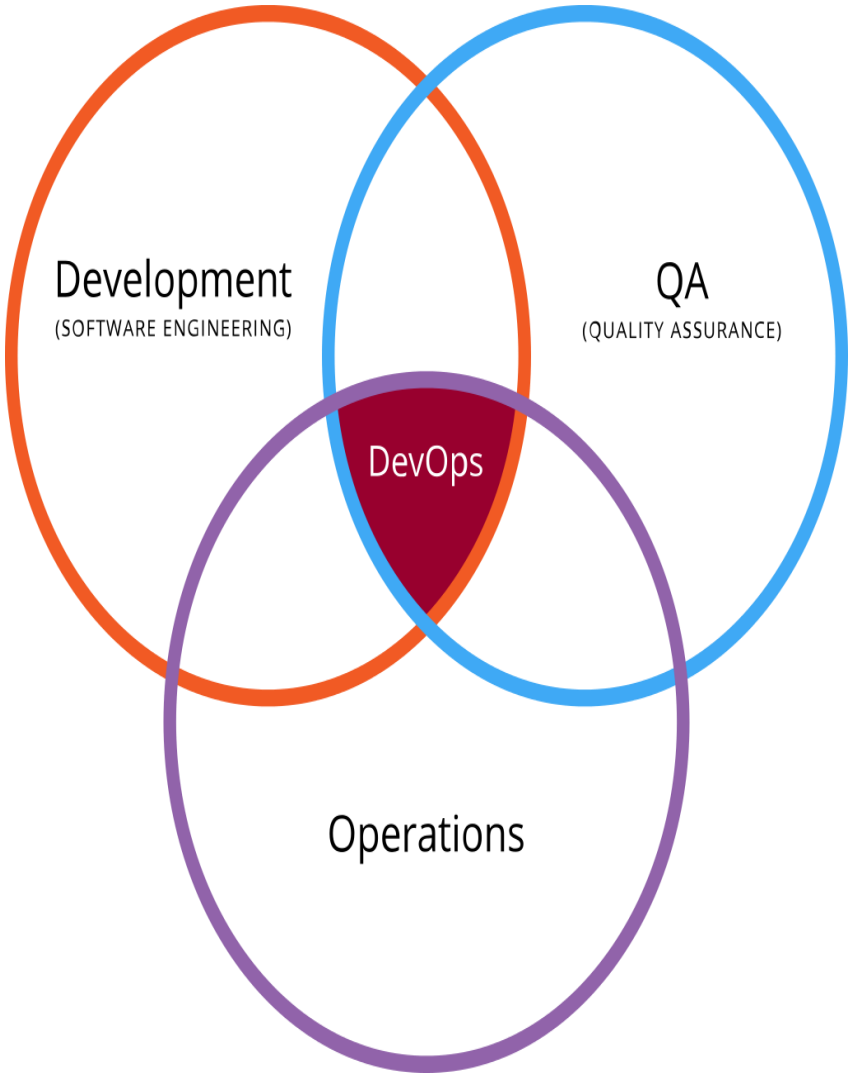
API accessibility

DevOps Cycle



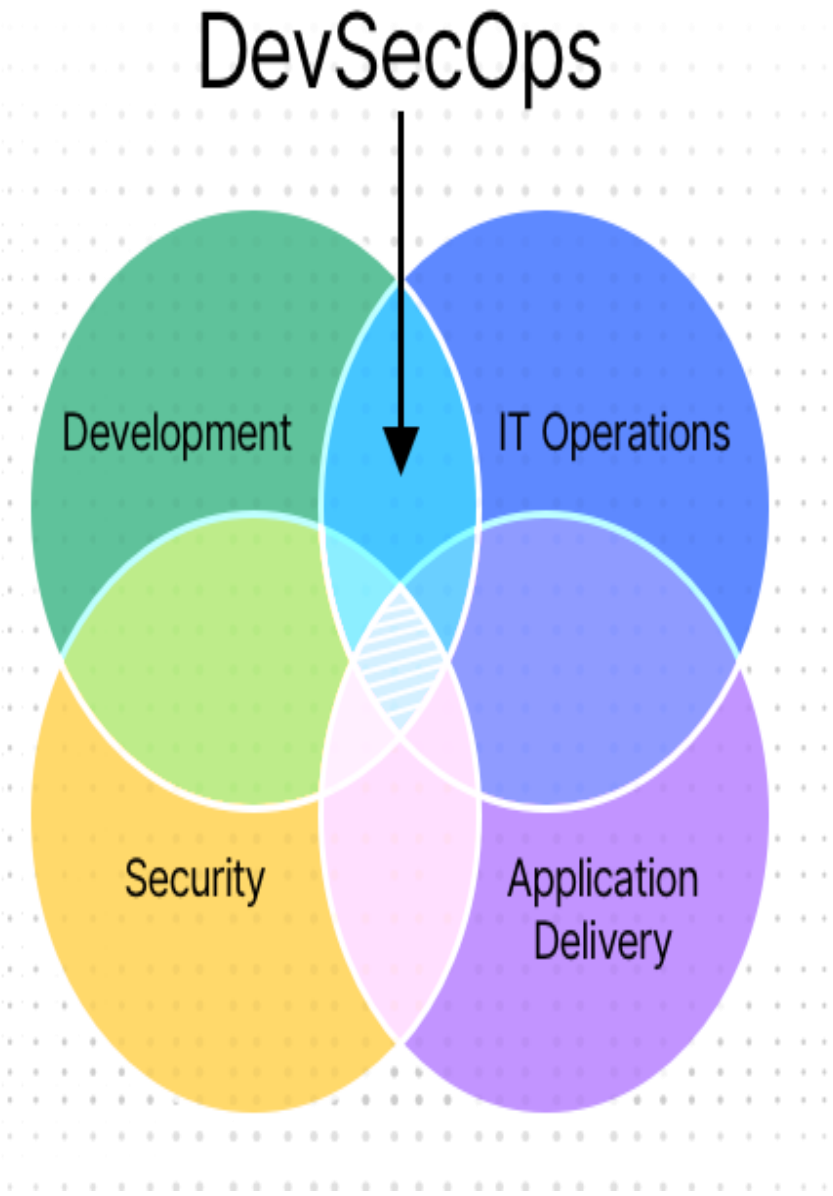
DevOps Tools





DevOps

DevOps è un insieme di pratiche che mira a fornire rapidamente software di qualità superiore integrando i processi tra i team di sviluppo e operativi.



DevSecOps

- DevSecOps (abbreviazione di sviluppo, sicurezza e operazioni) è una pratica di sviluppo che integra iniziative di sicurezza in ogni fase del ciclo di vita dello sviluppo del software per fornire applicazioni robuste e sicure.

More Resources

http://technttuts.com/story/DevOps-Tutorial

DevOps Tutorial Outline:-

1. DevOps-1: Introduction to DevOps
2. DevOps-2: Learn about Linux
3. DevOps-3: Version Control System [Git]
4. DevOps-4: Source Code Management (SCM) [Github]
5. DevOps-5: CI/CD [Jenkins]
6. DevOps-6: Software and Automation Testing Framework [Selenium]
7. DevOps-7: Configuration Management [Ansible]
8. DevOps-8: Containerization [Docker]
9. DevOps-9: Continuous Monitoring [Nagios]
10. DevOps-10: Continuous Orchestration [Kubernetes]