GITFLOW
CARATTERISTICHE E ALTERNATIVE

Bacaro Tech

CODE AND FUN

CIAO
MI PRESENTO

Giorgio Basile
Full stak developer - Almaviva Digital Tech

...VI PRESENTO BACARO TECH

Bacaro tech è un'inizitiva che ha il compito di voler avvicinare le persone che hanno un interesse verso il mondo del tech e della programmazione, ma non hanno trovato ancora un luogo dove potersi informare, oltre al fornire informazioni nuove a coloro che bazzicano in questo settore già da tempo

Bacaro
Tech

CODE AND FUN

CHE COS'È BACARO TECH

PAGINA INSTAGRAM

ORA INIZIAMO CON GITFLOW!

Bacaro
Tech

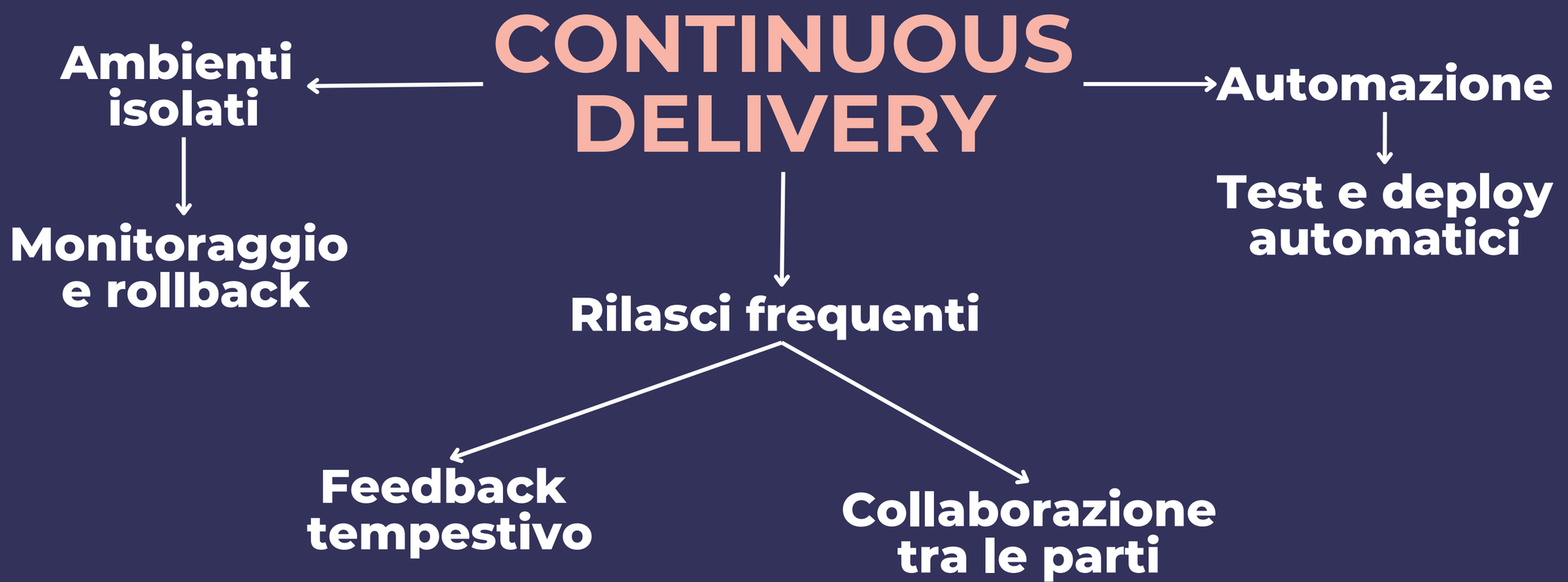
CODE AND FUN

CONTINUOUS DELIVERY
CHE COS'È?

La "**Continuous Delivery**" è un approccio al processo di sviluppo software che mira a ridurre al minimo i tempi tra lo sviluppo di nuove funzionalità e il loro rilascio al pubblico.

L'obbiettivo principale della Continuous Delivery è quello di **automatizzare il processo di rilascio** in modo che le nuove versioni del software possano essere consegnate in modo **rapido e affidabile**

CONTINUOUS DELIVERY
CHE COS'È?

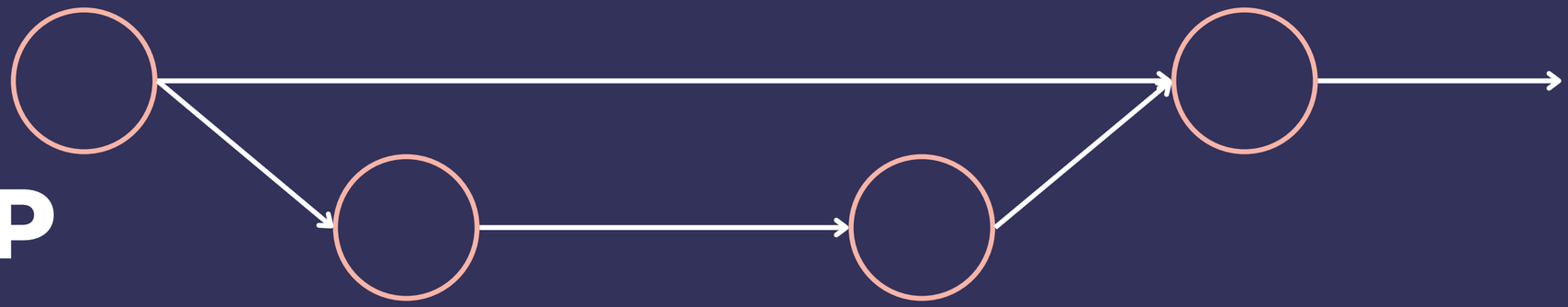


CONTINUOUS DELIVERY
COME OTTERLA?

CONTINUOUS DELIVERY → RAPIDITÀ AFFIDABILITÀ AUTOMAZIONE TEST → COME FARE ? → METODOLOGIA

GIT FLOW
CHE COS'È?

Ideato da Vincent Driessen, Git Flow è un **modello di branching e workflow per il sistema di controllo della versione Git** che offre una struttura ben definita per il flusso di lavoro durante lo sviluppo del software.



I "branch“ si riferiscono a percorsi separati di sviluppo di una repository

Bacaro
Tech

CODE AND FUN

GIT FLOW

QUALI SONO I BRANCH?

Branch principale (master): Il branch principale, spesso chiamato "master," rappresenta il codice in produzione. Questo è il ramo stabile e deve essere sempre pronto per il rilascio.

Branch di sviluppo (develop): Il branch di sviluppo è utilizzato per l'integrazione continua delle funzionalità. Tutti i contributi dai vari sviluppatori sono uniti in questo branch.

Bacaro
Tech

CODE AND FUN

GIT FLOW

QUALI SONO I BRANCH?

Feature branches: Per sviluppare nuove funzionalità o correzioni di bug, vengono creati branch specifici chiamati "feature branches." Questi branch si staccano dal branch di sviluppo e, una volta completate le modifiche, vengono uniti nuovamente al branch di sviluppo.

Release branches: Quando è giunto il momento di preparare una nuova versione del software, viene creato un branch di rilascio. Questo branch è utilizzato per effettuare gli ultimi aggiustamenti e le correzioni di bug prima di un rilascio ufficiale.

Bacaro
Tech

CODE AND FUN

GIT FLOW
QUALI SONO I BRANCH?

Hotfix branches: Se si verificano problemi critici nella versione in produzione, vengono creati branch di correzione immediata, noti come "hotfix branches." Questi branch vengono utilizzati per risolvere i problemi di produzione in modo rapido e immediato, e le correzioni vengono quindi integrate sia nel branch principale che nel branch di sviluppo.

Bacaro
Tech

CODE AND FUN

GIT FLOW
WORKFLOW DEVELOP

Il workflow legato allo sviluppo di codice prevede:

1. **Stacco un branch master** che verrà battezzato **develop**.
2. **Stacco un branch da develop**, con la naming convention **“feature/...”**, per lo sviluppo di una nuova feature.
3. **Faccio il merge della feature in develop** dopo averla accuratamente testata.
4. Eseguo punto 2 e 3 per eventuali fix, ma la naming convention questa volta è **“fix/...”.**

Bacaro
Tech

CODE AND FUN

GIT FLOW
WORKFLOW RELESE

Il workflow legato ai rilasci prevede:

1. **Da develop viene staccato un branch relese** per appunto il rilascio in produzione futuro
2. Viene **stabilizzato il codice** che si trova su branch relese e in caso vengano scovati dei bug, questo branch oltre a essere mergiato su master verrà anche mergiato in develop, per evitare rigressioni
3. Viene **rilasciata l'applicazione nell'ambiente di produzione**
4. In caso ci fossero dei bug bloccanti in prod, **allora si aprira' un branch hotfix** per sistemare questo problema. Successivamente le modifiche verranno portate su master e develop(guarda 2)

Bacaro
Tech

CODE AND FUN

GIT
LISTA DI COMANDI?

GIT - LA GUIDA TASCABILE

GIT FLOW
ESEMPIO

HOTFIX

RELESE

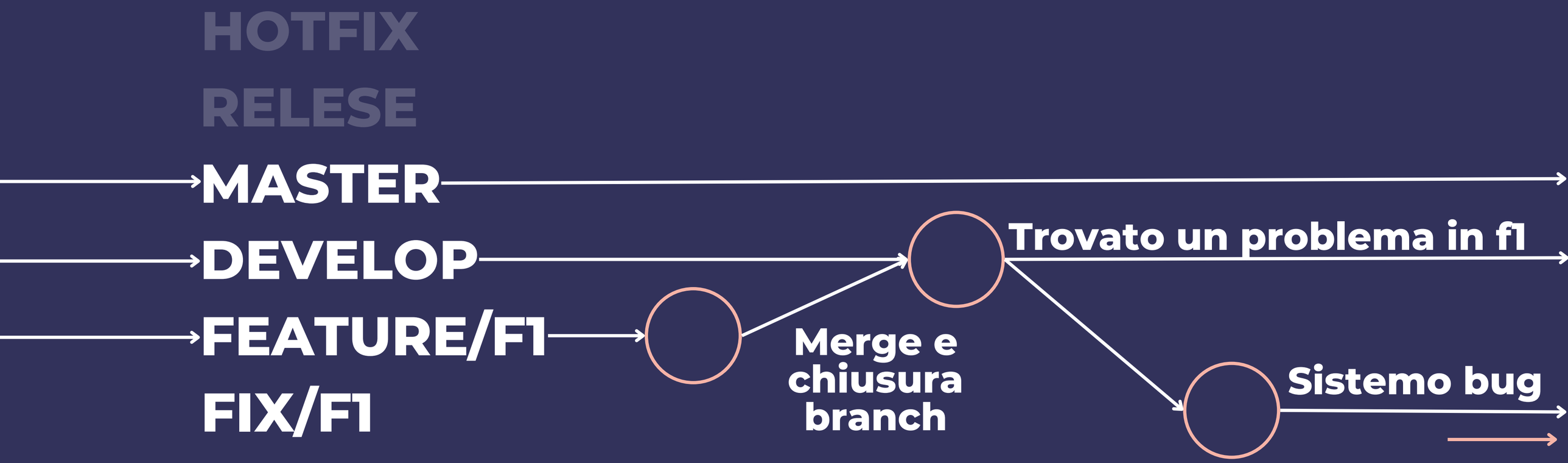
MASTER

DEVELOP

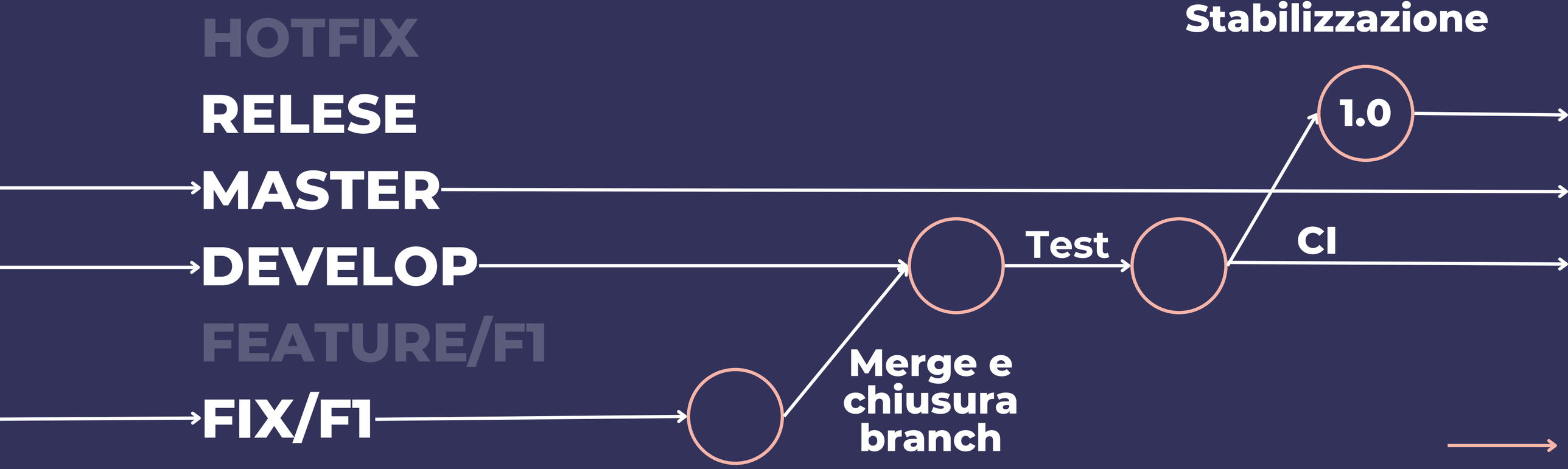
FEATURE/F1

FIX/F1

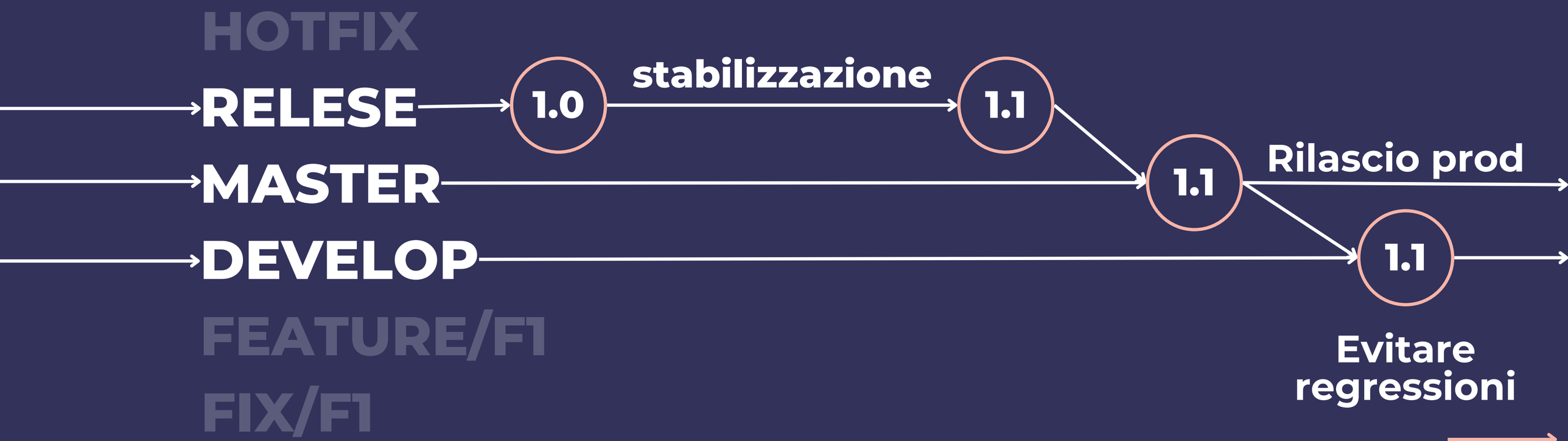
GIT FLOW
ESEMPIO



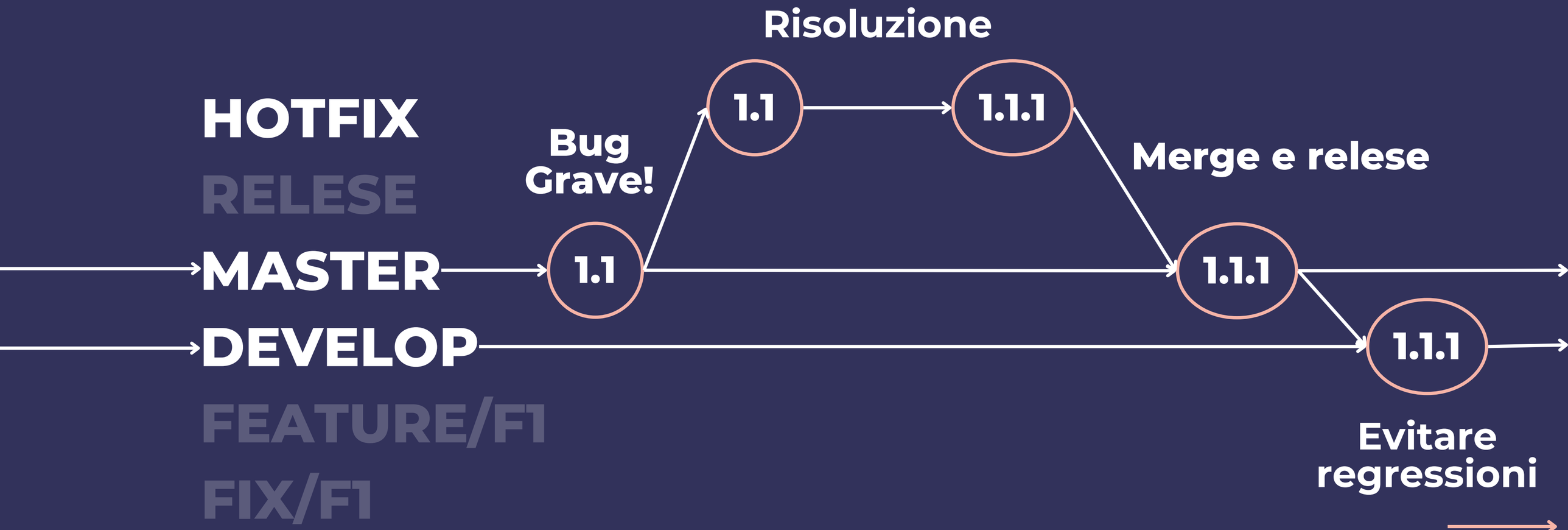
GIT FLOW
ESEMPIO



GIT FLOW
ESEMPIO

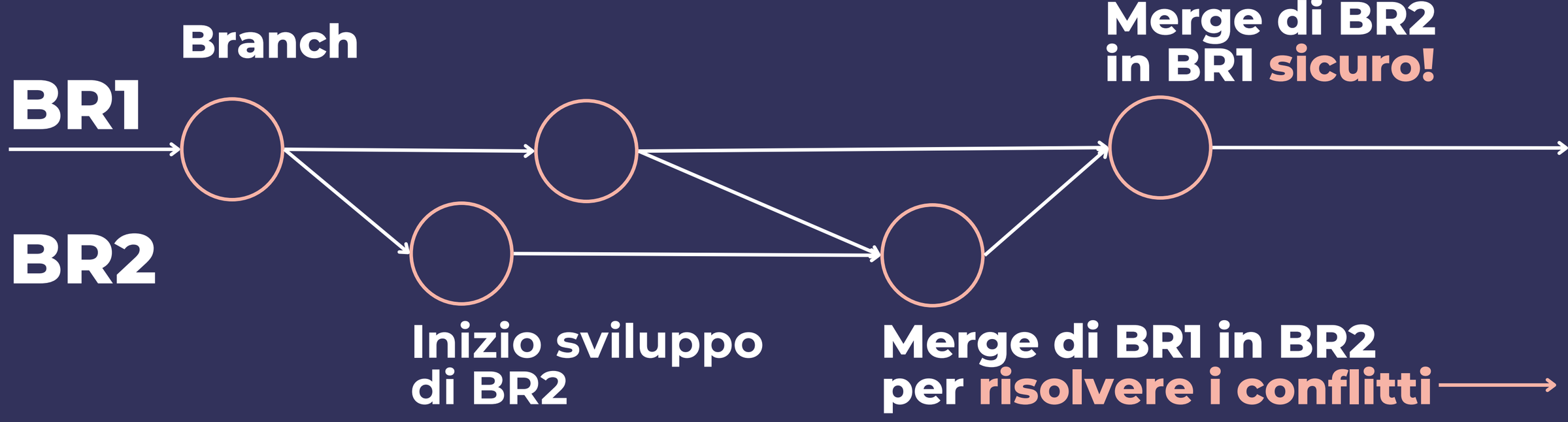


GIT FLOW
ESEMPIO



GIT FLOW
GESTIONE DEI MERGE

Una cosa importante da sottolineare è l'importanza di risolvere i conflitti nel proprio branch feature, e non nel branch che dovrà mergiare il nostro branch



GIT FLOW
VANTAGGI...

Struttura chiara

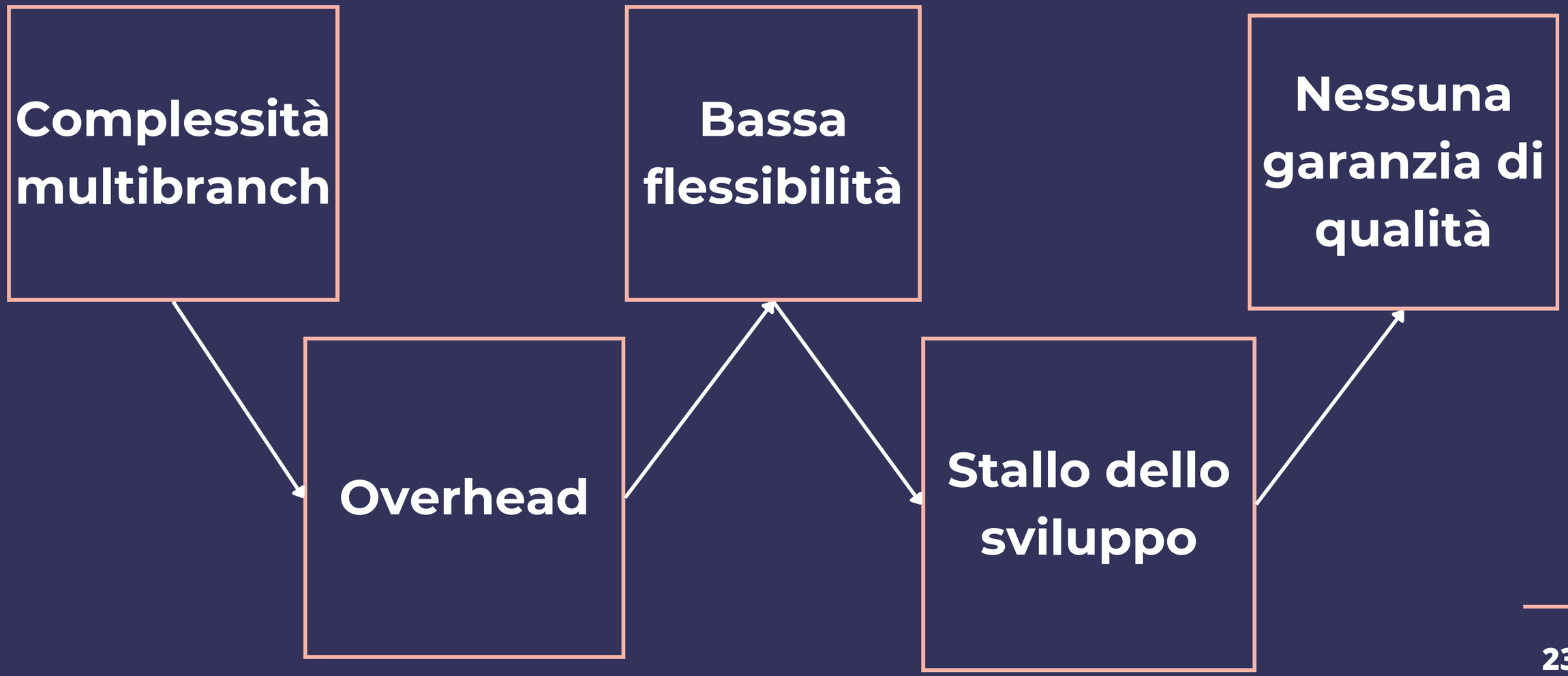
Isolamento delle funzionalità

Stabilità del master

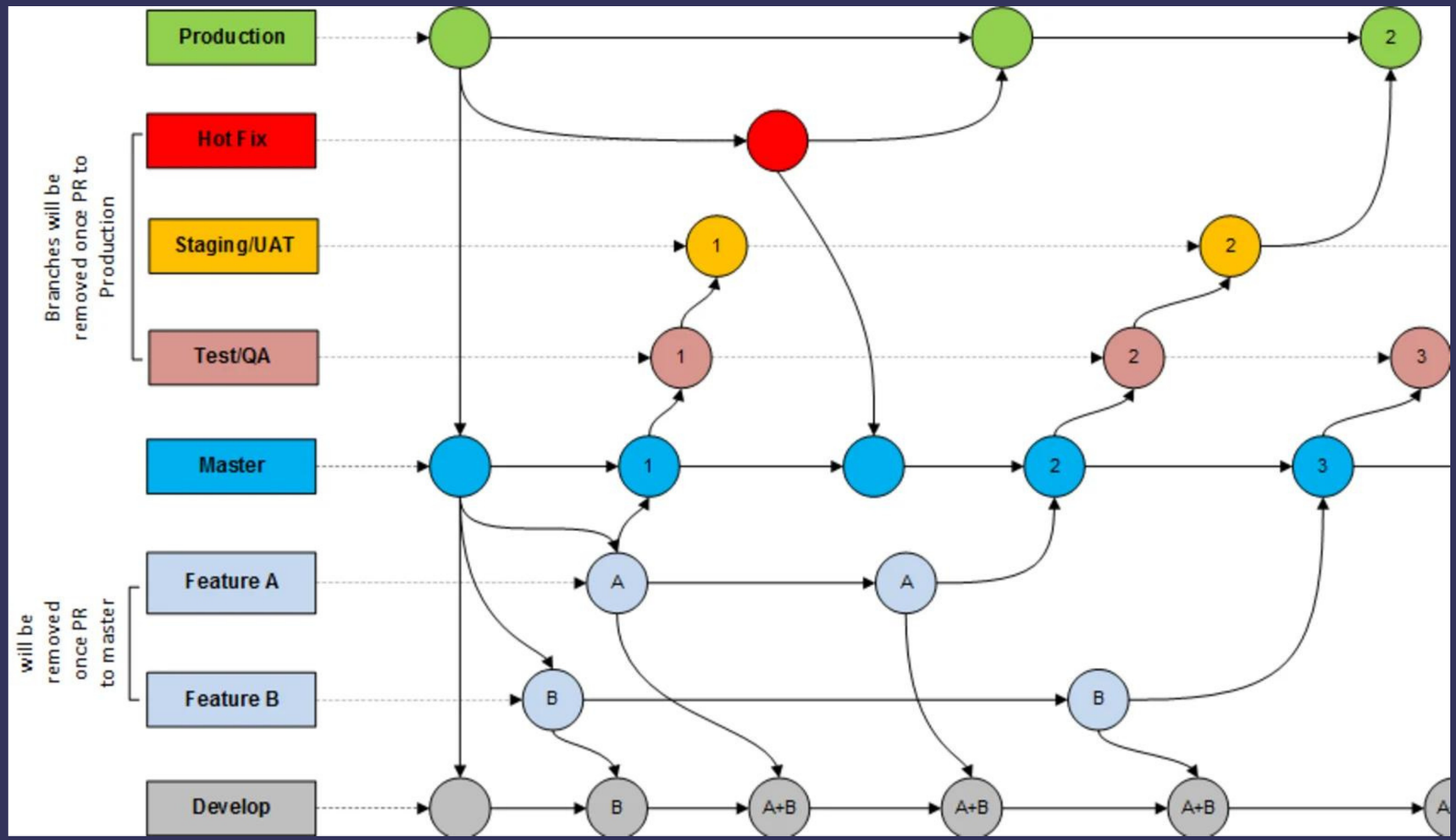
Correzione rapida dei bug

Gestione versioni

GIT FLOW
...SVANTAGGI



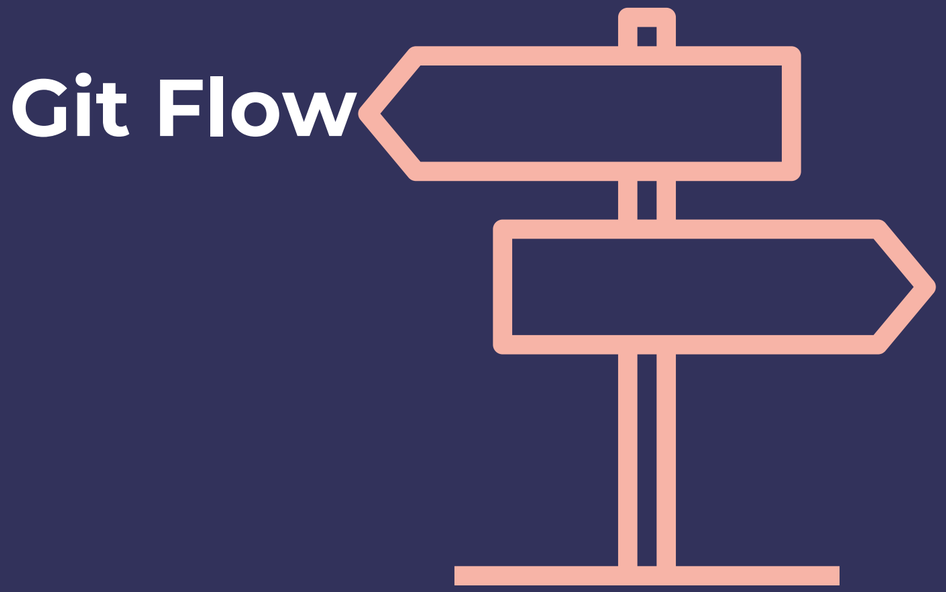
GIT FLOW
ALL'AUMENTARE DEI BRANCH



GIT FLOW
ALL'AUMENTARE DEI BRANCH

ALTRO ESEMPIO DI GIT FLOW

GIT FLOW ALTERNATIVE



Trunk-Based Development

+

Feature Toggles

Bacaro
Tech

CODE AND FUN

TRUNK-BASED DEVELOPMENT
CHE COS'È?

Il **Trunk-Based Development** è una metodologia di sviluppo software che si concentra sull'uso di **un unico "trunk"** (ramo principale) del sistema di controllo delle versioni come punto principale di sviluppo.

Tutti gli sviluppatori lavorano sullo stesso ramo principale del repository del codice, a differenza delle metodologie di sviluppo branch-based.

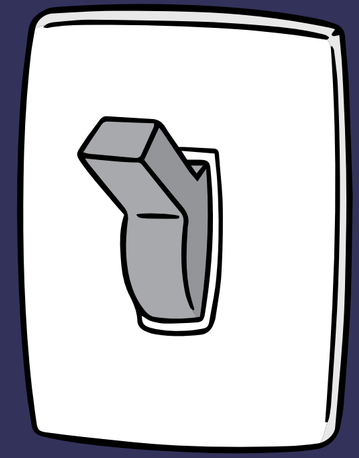
Sviluppo continuo sul trunk

Integrazione continua (CI) e rilasci continui

Feature toggles

FEATURE TOGGLES
CHE COS'È?

I **Feature Toggles** sono un meccanismo di sviluppo software che permette di **attivare o disattivare dinamicamente specifiche funzionalità o porzioni di codice** in un'applicazione senza dover rilasciare una nuova versione del software.

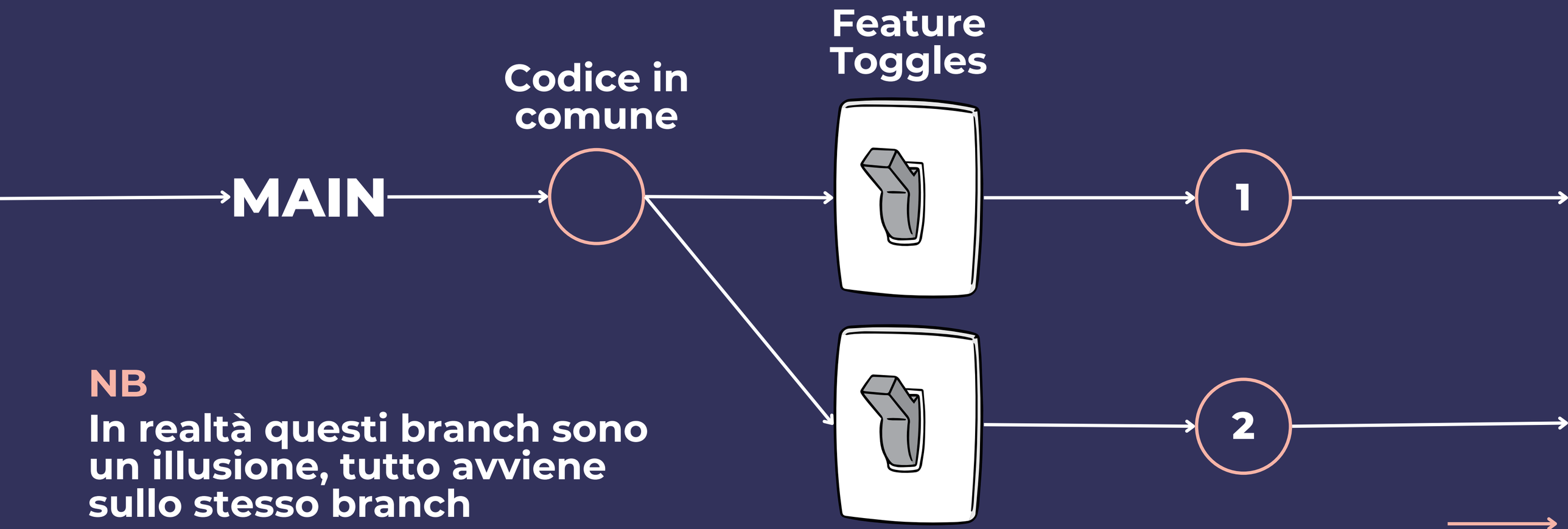


Integrazione nel codice

Controllo dinamico

Branching condizionale

FEATURE TOGGLES
ESEMPIO PRATICO



NB

In realtà questi branch sono un illusione, tutto avviene sullo stesso branch

Bacaro
Tech

CODE AND FUN

FEATURE TOGGLES
ESEMPIO PRATICO

NB

In realtà questi branch sono un illusione, tutto avviene sullo stesso branch

TRUNK-BASED DEVELOPMENT
SITO UFFICIALE

SITO UFFICALE

ORA PARLIAMO DI VERSIONI!

Bacaro
Tech

CODE AND FUN

NUMERO DI VERSIONE
CHE COS'È?

I **"numeri di versione"** si riferiscono a una notazione utilizzata per identificare le diverse iterazioni o revisioni di un software, un'applicazione o un prodotto.

Questi numeri vengono assegnati per tenere traccia delle modifiche apportate al prodotto nel corso del tempo e per consentire agli utenti di sapere quale versione stanno utilizzando.

2.4.1

Bacaro
Tech

CODE AND FUN

NUMERO DI VERSIONE
COME LEGGERLO?

2.4.1

Major Version

Questa è la **parte principale del numero di versione**, un cambiamento in questo numero indica solitamente un'**importante revisione del software**, con nuove funzionalità, modifiche significative o aggiunte importanti.

Bacaro
Tech

CODE AND FUN

NUMERO DI VERSIONE
COME LEGGERLO?

2.4.1

Minor Version

Questa è la **seconda parte del numero di versione**, il cui aumento indica una r**evisione più piccola rispetto alla versione principale**, ma comunque significativa che può includere:

1. correzioni di bug
2. miglioramenti delle prestazioni
3. l'aggiunta di nuove funzionalità minori.

Bacaro
Tech

CODE AND FUN

NUMERO DI VERSIONE
COME LEGGERLO?

2.4.1

Patch

Questa è la terza parte del numero di versione. Un aumento in questo numero è **spesso utilizzato per indicare correzioni di bug o aggiornamenti di manutenzione** senza aggiungere nuove funzionalità significative.

TAG
CHE COS'È?

I **"tag"** in Git sono **punti di riferimento statici** che vengono utilizzati per contrassegnare specifici punti nella cronologia di un repository Git. I tag sono spesso utilizzati per segnare versioni stabili del software in modo che possano essere facilmente recuperate in futuro.

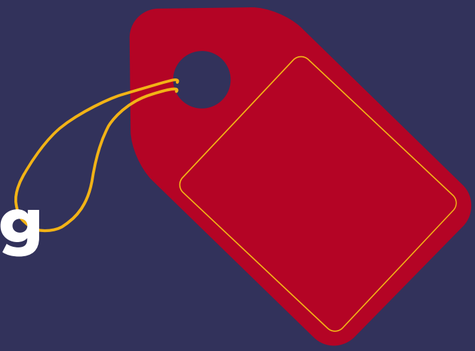
Tag Leggeri/Lightweight Tags

Tag Annotati/Annotated Tags

TAG LEGGERO
CHE COS'È?

Un **tag leggero** è semplicemente un **puntatore a un commit specifico nella cronologia di Git**, in modo tale da marcare un punto specifico senza dover aggiungere ulteriori informazioni
I tag leggeri sono facili da creare e leggeri dal punto di vista dei metadati.

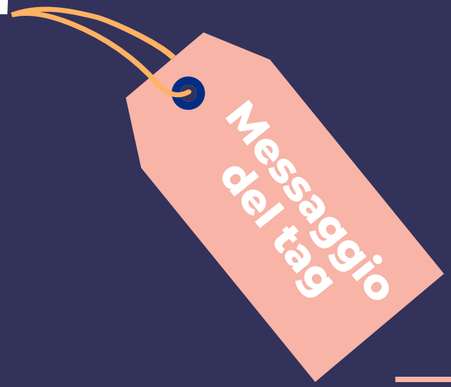
git tag nome_del_tag



TAG ANNOTATI
CHE COS'È?

Un **tag annotato** è un **oggetto Git completo che contiene informazioni aggiuntive** come il nome dell'autore, la data e un messaggio descrittivo.

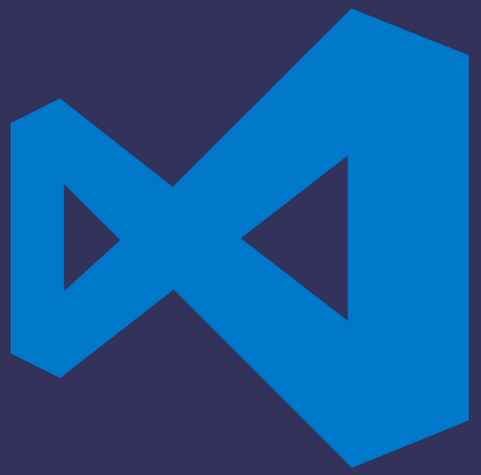
git tag -a nome_del_tag -m "Messaggio del tag"



Bacaro
Tech

CODE AND FUN

NUMERO DI VERSIONE
ESEMPI



MINECRAFT
1.19.83

WHATSAPP
2.23.9

VSC
1.83.1

Bacaro
Tech

CODE AND FUN

VI RINGRAZIA TUTTI PER AVER PARTECIPATO!

Bacaro
Tech

CODE AND FUN

BACARO TECH
SEGUITECI!

PAGINA INSTAGRAM