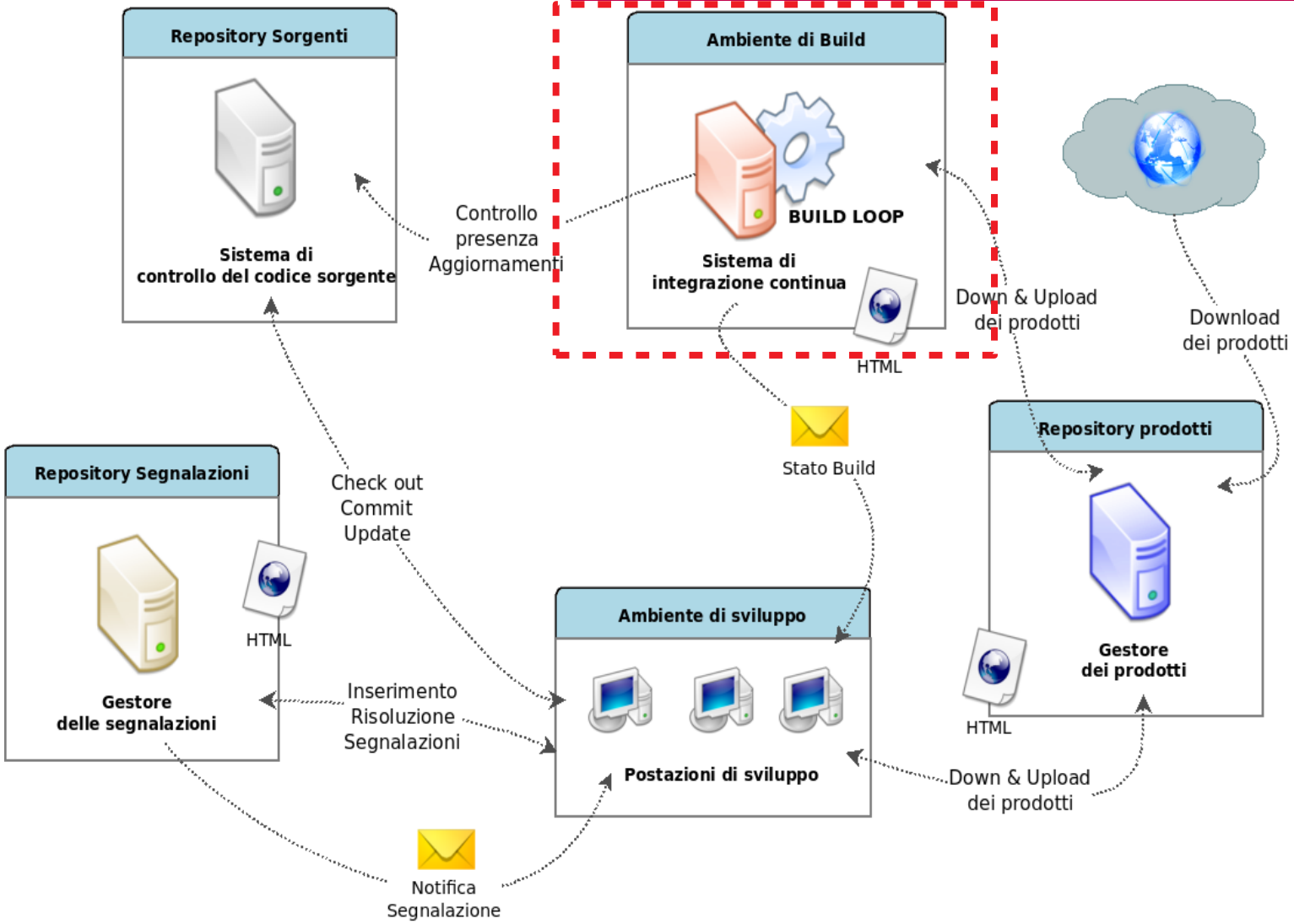
Integrazione Continua

METODI E TECNOLOGIE PER LO SVILUPPO SOFTWARE

Nicola Bertazzo
nicola.bertazzo [at] unipd.it

Università degli Studi di Padova
Dipartimento di Matematica
Corso di Laurea in Informatica, A.A. 2022 – 2023

Visione Generale



Continuous Integration

Definizione

Nell'ingegneria del software, l'integrazione continua o continuous integration è una pratica che si applica in contesti in cui lo sviluppo del software avviene attraverso un sistema di versioning. Consiste **nell'allineamento frequente** (ovvero "molte volte al giorno") **dagli ambienti di lavoro degli sviluppatori verso l'ambiente condiviso** (mainline). Il concetto è stato originariamente proposto nel contesto dell'extreme programming (XP), come contromisura preventiva per il problema dell' "integration hell" (le difficoltà dell'integrazione di porzioni di software sviluppati in modo indipendente su lunghi periodi di tempo e che di conseguenza potrebbero essere significativamente divergenti).

Il CI è stato originariamente concepito per essere complementare rispetto ad altre pratiche, in particolare legate al Test Driven Development (sviluppo guidato dai test, TDD). In particolare, si suppone generalmente che siano stati predisposti **test automatici** che gli sviluppatori **possono eseguire immediatamente** prima di rilasciare i loro contributi verso l'ambiente condiviso, in modo da garantire che le modifiche non introducano errori nel software esistente. Per questo motivo, il CI viene spesso applicato in ambienti in cui siano presenti sistemi di **build automatico** e/o esecuzione automatica di test, come Jenkins.

WIKIPEDIA
The Free Encyclopedia

Motivazioni

In molti progetti:

- Per **lunghi periodi di tempo**, durante il processo di sviluppo, **il progetto non è in uno stato funzionante** o in uno stato utilizzabile. Sopratutto in progetti dove si sviluppa in un singolo ramo di sviluppo (centralized work-flow)
- Nessuno è interessato a provare ad eseguire l'intera applicazione fino a quando non è finito il processo di sviluppo
- In questi progetti spesso viene pianificata la fase di integrazione alla fine del processo di sviluppo. In questa fase gli sviluppatori effettuano attività di merge ed effettuano attività di verifica e validazione

PROBLEMA (*integration hell*):

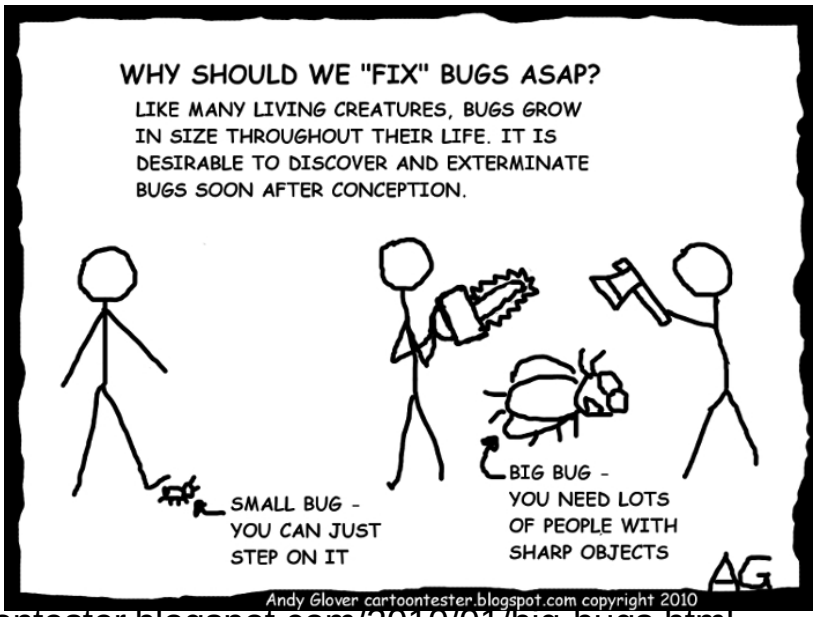
La fase di integrazione può richiedere molto tempo, e nel caso peggiore, **nessuno ha modo di prevedere quando terminerà** questa fase e di conseguenza quando l'applicazione può essere rilasciata.

Continuous Integration

Definizione

“Continuous Integration doesn’t get rid of bugs, but it does make them dramatically easier to find and remove.” (Martin Fowler)

Consente a un Team di intensificare l’attività di sviluppo e test, **integrando gli sviluppi** (nel VCS) **il più spesso possibile**

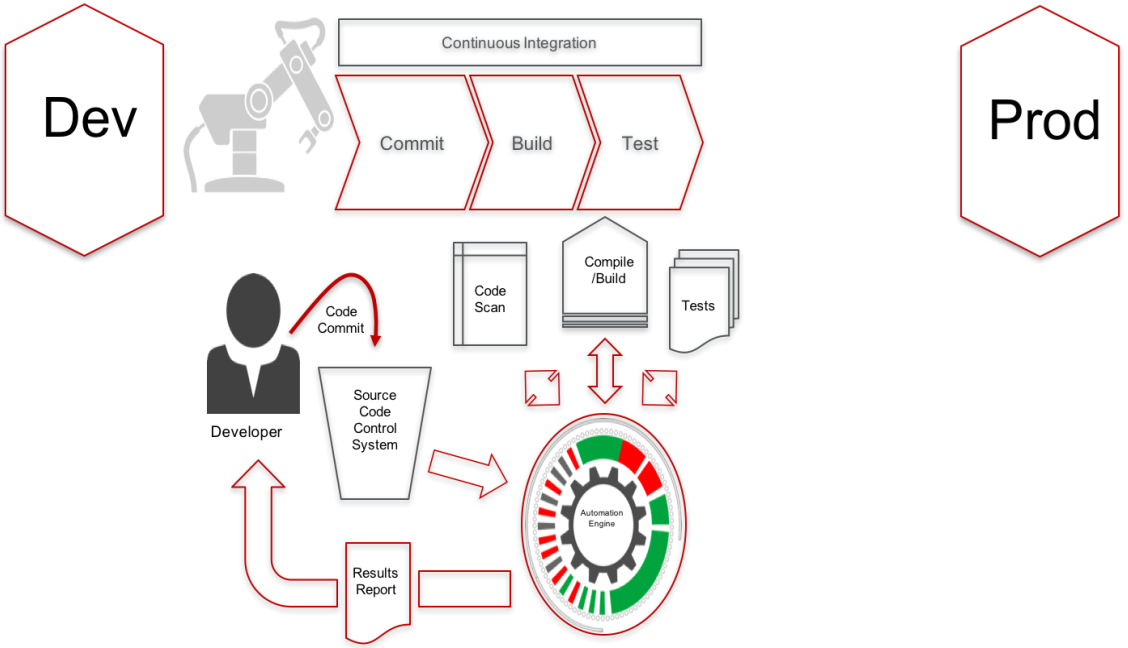


http://cartoontester.blogspot.com/2010/01/big-bugs.html

Continuous Integration

Definizione

*"Continuous Integration is a software development **practice** where members of a team integrate their work **frequently**, usually each person integrates **at least daily** - leading to **multiple** integrations per day." (Martin Fowler)*



Processo - visione generale

- Al **completamento** di un'attività viene **costruito il prodotto**:
- ogni volta che uno sviluppatore invia un commit al VCS viene eseguito il processo di build (compilazione e test)

- Se il processo di costruzione **fallisce l'attività non continua** fino a che il prodotto non viene riparato.

- Se **non è possibile riparare** il prodotto immediatamente (in pochi minuti) si **ritorna all'ultima versione funzionante**

- In questo modo si **assicura la presenza di un prodotto consistente** potenzialmente pronto per essere validato e rilasciato

Prerequisiti

Per implementare la pratica di CI è necessario che:

1)Il codice del progetto venga gestito in un **VCS**
2)Il **processo di build** del progetto sia **automatico**
3)Il processo di build esegua delle **verifiche automatiche** (test di unità, test di integrazione, analisi statica del codice ...)
4)il team di sviluppo adotti correttamente questa pratica
5)Un sistema automatico dove eseguire il processo di build ad ogni integrazione [Opzionale]

Processo in dettaglio

1. Controllo se il processo di build è in esecuzione nel sistema di CI. Se è in esecuzione aspetto che finisca, se fallisce lavoro con il team in modo da sistemare il problema.

2. Quando il processo di build ha terminato con successo, aggiorno il codice nel mio workspace con il codice del VCS ed effettuo l'integrazione in locale.

3. Eseguo il processo di build in locale in modo da verificare che tutto funzioni correttamente

4. Se il processo di build termina con successo invio le modifiche al VCS

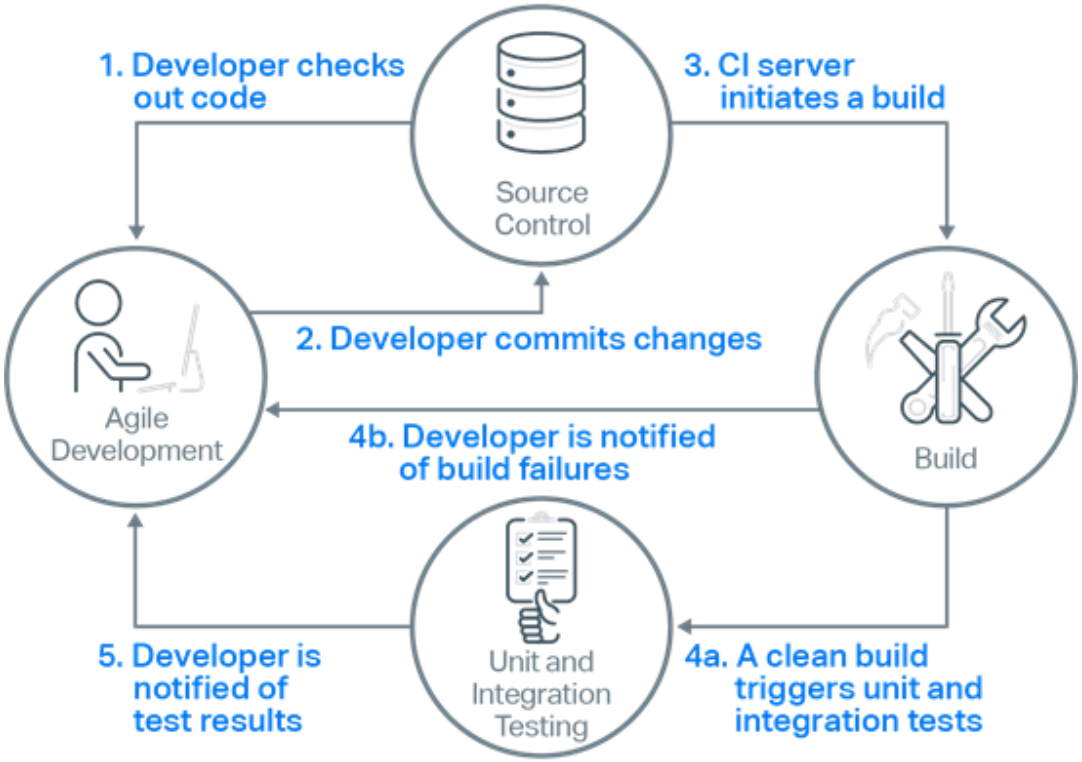
5. Attendo che il sistema di CI esegua il processo di build con i miei cambiamenti.

6. Se il processo di buil fallisce mi fermo con le attività di sviluppo, e lavoro per sistemare il problema in locale e riprendo dal passo 3.

7. Se il processo di build termina con successo passo allo sviluppo dell'attività successiva

Continuous Integration

Processo in dettaglio



https://www.microfocus.com/documentation/visual-cobol/VC40/EclWin/GUID-70375FE3-745F-4FA5-B28A-6F6595

Integrare frequentemente gli sviluppi

Integrare frequentemente gli sviluppi, più di una volta al giorno.

In questo modo:

- le modifiche da integrare saranno poche e più facile da gestire
- Se viene segnalato un errore dall'esecuzione del processo di build sarà più semplice identificare il codice che ha introdotto l'errore (che sarà presente nei commit che hanno fatto scatenare il processo di build)

Per poter integrare frequentemente gli sviluppi è richiesto che il progetto venga scomposto in tante attività brevi ("DIVIDE ET IMPERA")

Create a Comprehensive Automated Test Suite

Se non vengono eseguiti dei test automatici per verificare e validare il progetto, il processo di build può solo verificare se il codice integrato compila correttamente.

I test che possono essere eseguiti nel processo di CI sono:

- I test di unità
- I test di integrazione di subsystem interni
- Analisi statica del codice

Keep the Build and Test Process Short

Nella CI il processo di Build viene eseguito molto frequentemente (ad ogni integrazione)

Se il processo di Build è lento:

- Gli sviluppatori smetteranno di eseguire il processo di build e i test prima di inviare le modifiche al VCS → Inizieranno a generare più build in errore
- Il processo di integrazione continua richiederà così tanto tempo che si verificheranno più commit nel momento in cui è possibile eseguire nuovamente la build → sarà più difficile identificare cosa ha fatto fallire la build
- Si disincentiva l'invio frequente delle modifiche al VCS

Managing Your Development Workspace

Ogni sviluppatore deve essere in grado di:

- Eseguire localmente il processo di build e test e deploy → per questo nel VCS devono essere gestiti:
  - Codice di produzione
  - Codice di test
  - Script richiesti per configurare l'ambiente dove eseguire il progetto
- **Always Be Prepared to Revert to the Previous Revision**: Scaricare le modifiche dal VCS e essere in grado di ripristinare il progetto ad uno stato consistente.
- **Never Go Home on a Broken Build:** Verificare l'esito della compilazione nel CI server → Se il processo di CI fallisce lo sviluppatore deve essere in grado o di risolvere il problema o di ripristinare la versione del VCS all'ultimo stato consistente

Fix Broken Builds Immediately

"nobody has a higher priority task than fixing the build" [Kent Beck]

Il tempo per ripristinare lo stato del progetto deve essere limitato. Se non è possibile correggere l'errore che ha fatto fallire la build velocemente, ripristinare lo stato del VCS all'ultima versione funzionante.

Non è ammesso correggere il problema commentano le verifiche che hanno fatto fallire la build.

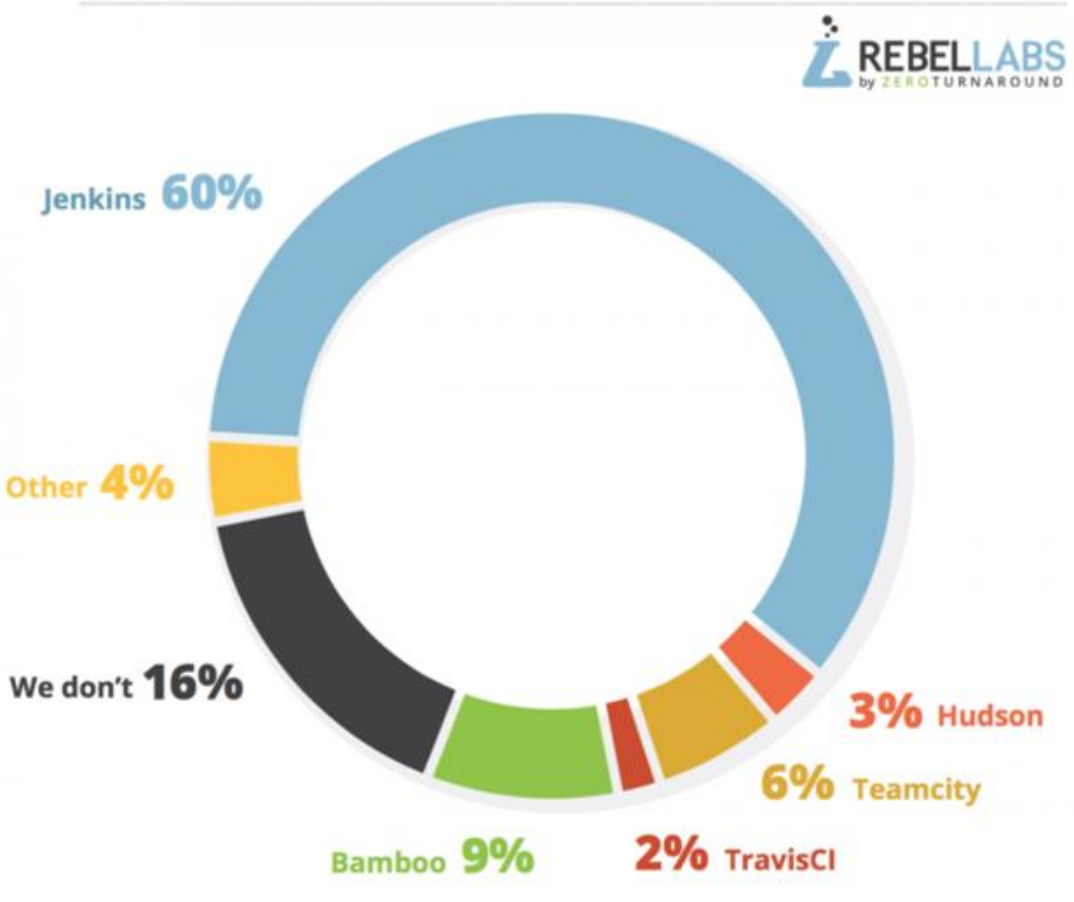
Everyone can see what's happening

Lo stato della build deve essere pubblicato in un servizio visibile a tutto il team del progetto.
Ogni componente del team deve essere in grado di capire lo stato del progetto e capire qual'è l'ultima versione in cui è stato eseguito il processo di build con successo.
Se il processo di build fallisce deve essere possibile:

- identificare chi ha introdotto l'errore
- Avere a disposizione un log per identificare quale parte del processo di build è fallita
- Avere a disposizione la lista dei commit che hanno introdotto l'errore

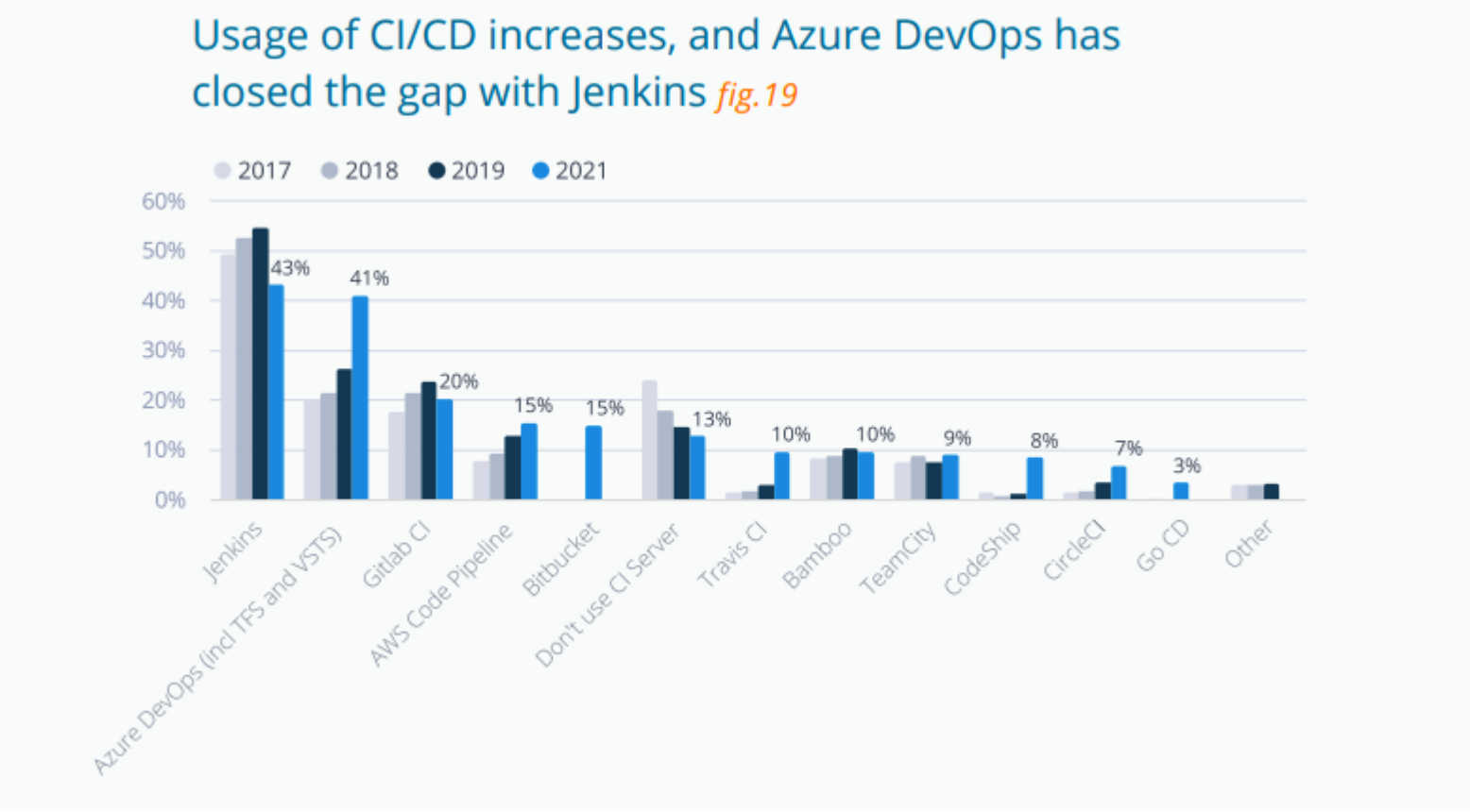
Il sistema di continuous integreation deve poter avvisare (con delle notifiche) i componenti del team ad ogni cambio di stato del processo. (da successo ad fallimento, da fallimento a successo)

Continuous Integration Server Usage - Java 2016



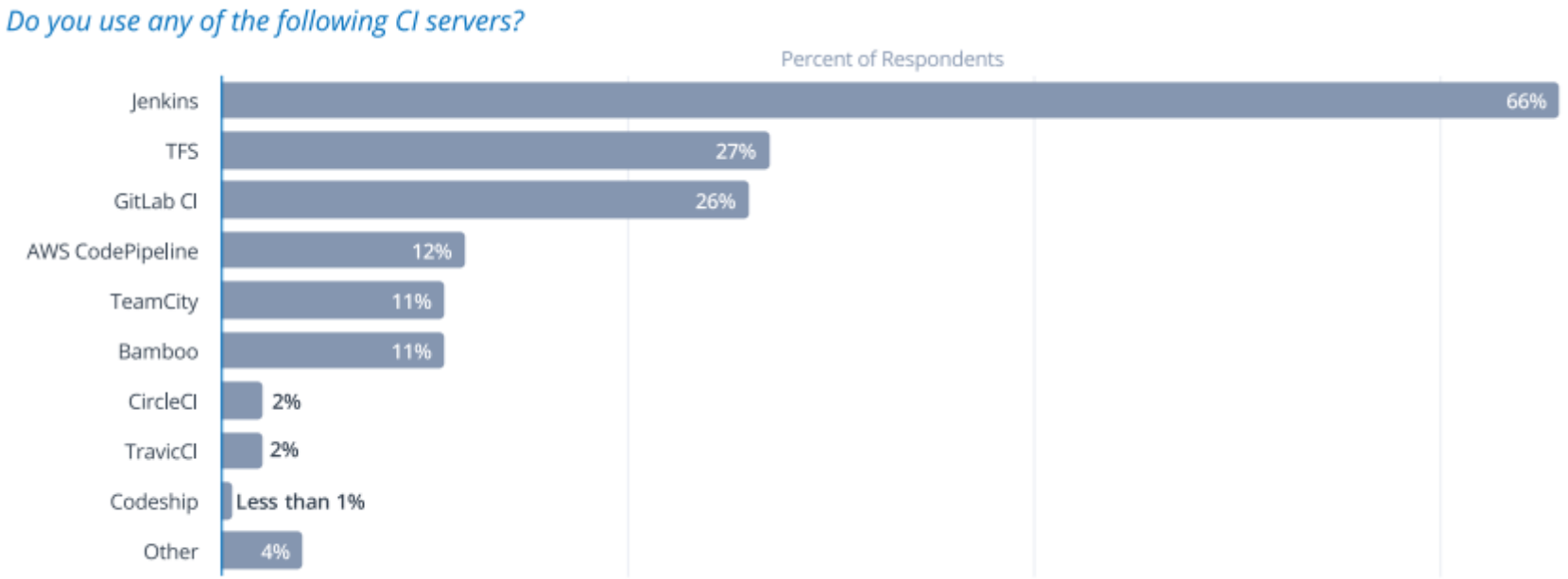
https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/

Continuous Integration



https://static1.smartbear.co/smartbearbrand/media/pdf/state-of-quality-testing-2021.pdf

Continuous Integration



https://smartbear.com/SmartBear/media/ebooks/The-2018-State-of-Testing-Report_1.pdf

Riferimenti

https://it.wikipedia.org/wiki/Integrazione_continua
https://martinfowler.com/articles/continuousIntegration.html
https://smartbear.com/SmartBear/media/ebooks/The-2018-State-of-Testing-Report_1.pdf
https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/
https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912
https://static1.smartbear.co/smartbearbrand/media/pdf/state-of-quality-testing-2021.pdf