

# Utilizzo di Git per sviluppo collaborativo

Nicolò Paraciani – Istituto di Scienze per il Patrimonio Culturale ISPC-CNR

ICTalk@CNR Workshop

13-15 ottobre 2021



# Cos'è Git

Git è un sistema *open-source* di controllo del *versioning*, o *Version Control System (VCS)*, utilizzato tipicamente in contesti di sviluppo software, ma applicabile anche in altri casi.

Git è stato sviluppato inizialmente nel 2005 principalmente da Linus Torvalds e altri collaboratori.

Attualmente, lo sviluppo è portato avanti da una *community*.

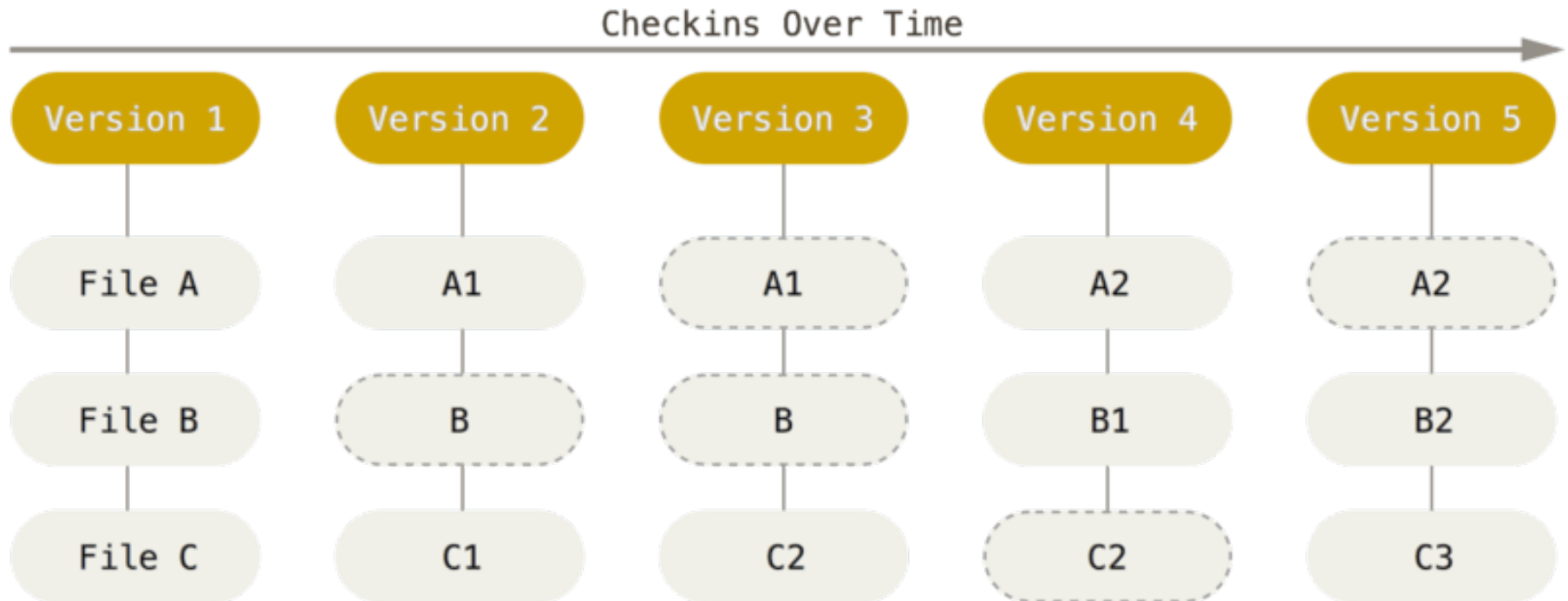
# Piattaforme Web & automazione

- Ampia adozione di Git in progetti di sviluppo open source anche molto vasti.
- Diffusione di piattaforme Web come **GitHub**, **GitLab**, **Gitea** e **BitBucket**: servizi *Git server* per repository remoti, condivisione, collaborazione, *issue tracking* ecc.
- Git utilizzato in contesti di automazione del processo di sviluppo e deployment → CI/CD

# Caratteristiche di Git

- Modello *distribuito* per il *versioning*, non è necessario un server centrale.
- Ogni utente può avere una copia locale esatta (*clone*) del repository con sorgenti e database delle versioni.
- Basato sul concetto di *commit* inteso come *snapshot* o immagine dello stato dei file in un determinato momento.
- Possibilità di lavorare facilmente su più linee di sviluppo parallele (*branches*).

# Snapshot versioni

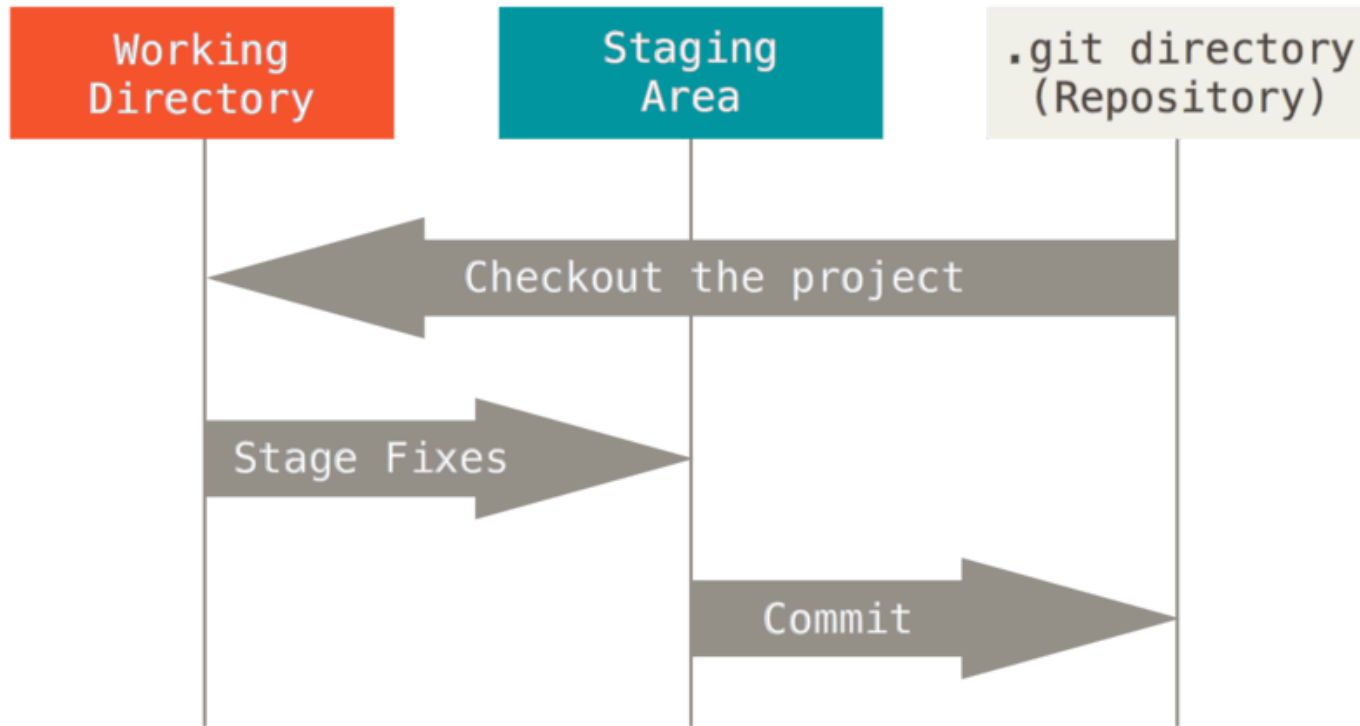


(Fonte: S. Chacon, B. Straub (2014) "Pro Git", <https://git-scm.com/book/en/v2>)

# Stati del flusso di lavoro

- **Modifiche** (*working directory*): stato dei cambiamenti dei vari file contenuti nella dir principale del progetto. Le modifiche possono essere tracciate o non tracciate.
- **Indice** (*staging area*): file `index` in `.git/` contenente le modifiche aggiunte alla *staging area*, che saranno incluse nel prossimo *commit*.
- **Committed**: modifiche incluse nella *staging area* salvate nel database interno del repository come snapshot (hash).

# Le 3 “aree” di lavoro



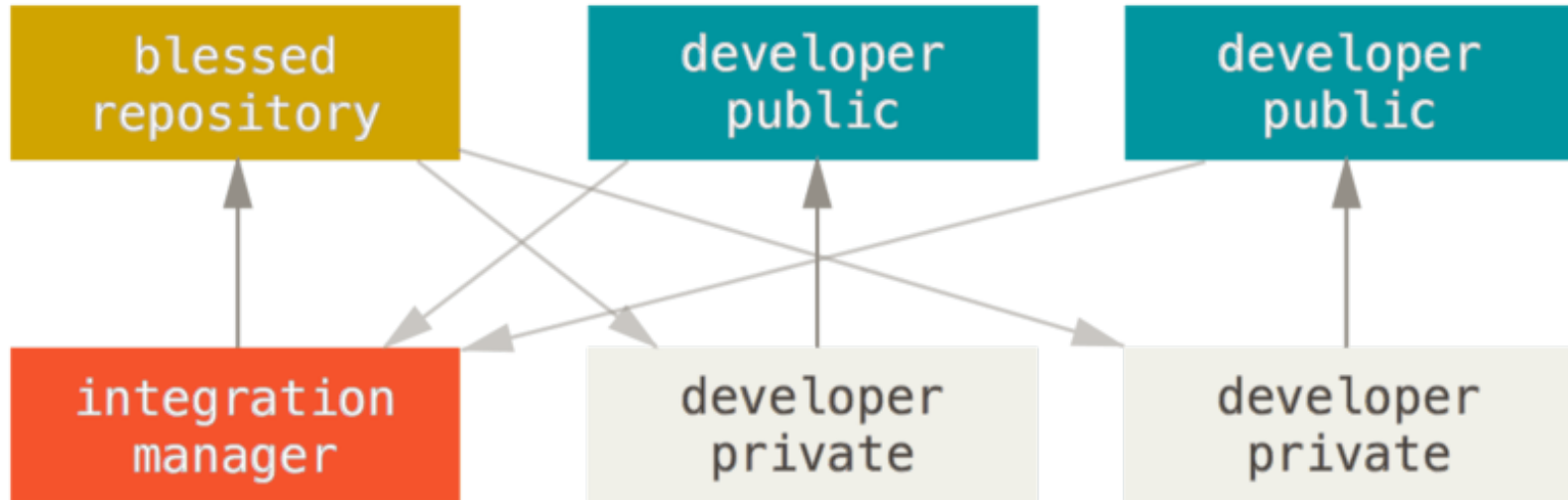
(Fonte: S. Chacon, B. Straub (2014) “Pro Git”, <https://git-scm.com/book/en/v2>)

# Workflow Git di base

1. Si effettuano modifiche ai file nella *working directory*.
2. Le modifiche vengono aggiunte – in parte o del tutto – alla *staging area* (`git add`).
3. Si esegue un *commit* delle modifiche presenti nella *staging area*, in modo da salvarle permanentemente nel repository – ovvero nella Git directory (`git commit`).



# Workflow collaborativo (esempio)



(Fonte: S. Chacon, B. Straub (2014) “Pro Git”, <https://git-scm.com/book/en/v2>)

# Workflow collaborativo (esempio)

1. Repository remoto principale gestito da una o più persone (*maintainers*)
2. *Fork* (cloni pubblici) del repository principale gestiti da collaboratori (1 *fork* per ogni collaboratore)
3. Collaboratore lavora in locale su *branch* separata per introdurre modifiche (aggiornando la copia *master* se necessario tramite *pull*)
4. Collaboratore invia modifiche al proprio *fork* remoto pubblico (*push*) e apre una *pull request* (o *merge request*)
5. Il *maintainer*:
  1. esegue *pull* in locale della branch remota tramite la *pull request*
  2. verifica le modifiche introdotte
  3. esegue *merge* nella branch *master* del repository principale, **oppure** richiede modifiche al collaboratore, che ripeterà il processo.

# Istanza Gitlab CNR (baltig.cnr.it)

Il Centro Servizi dell'Ufficio ICT del CNR ha predisposto un'istanza di GitLab disponibile all'indirizzo <https://baltig.cnr.it> ed accessibile tramite credenziali CNR (LDAP).

Baltig@CNR



Consiglio Nazionale delle Ricerche

Questa è un'istanza di GitLab CE ospitata sui server dell'ufficio **ICT - SAC** e gestita dal **Centro Servizi**.

Per maggiori informazioni: [centroservizi@cnr.it](mailto:centroservizi@cnr.it)

LDAP	Standard
LDAP Username	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	
<input type="button" value="Sign in"/>	

# Possibilità di automazione

1. Utilizzo dei *Git hooks* per eseguire automaticamente script quando si verifica un evento associato al repository (*commit, push, merge, ecc.*).
2. Utilizzo dei *Webhooks* associati a un repository remoto (GitHub, GitLab ecc.), per esecuzione script / app su base Web.
3. Sistemi di tipo *Continuous Integration/Continuous Delivery (CI/CD)* associati a uno o più repository remoti per automatizzare sia sviluppo che deployment.

**Grazie**