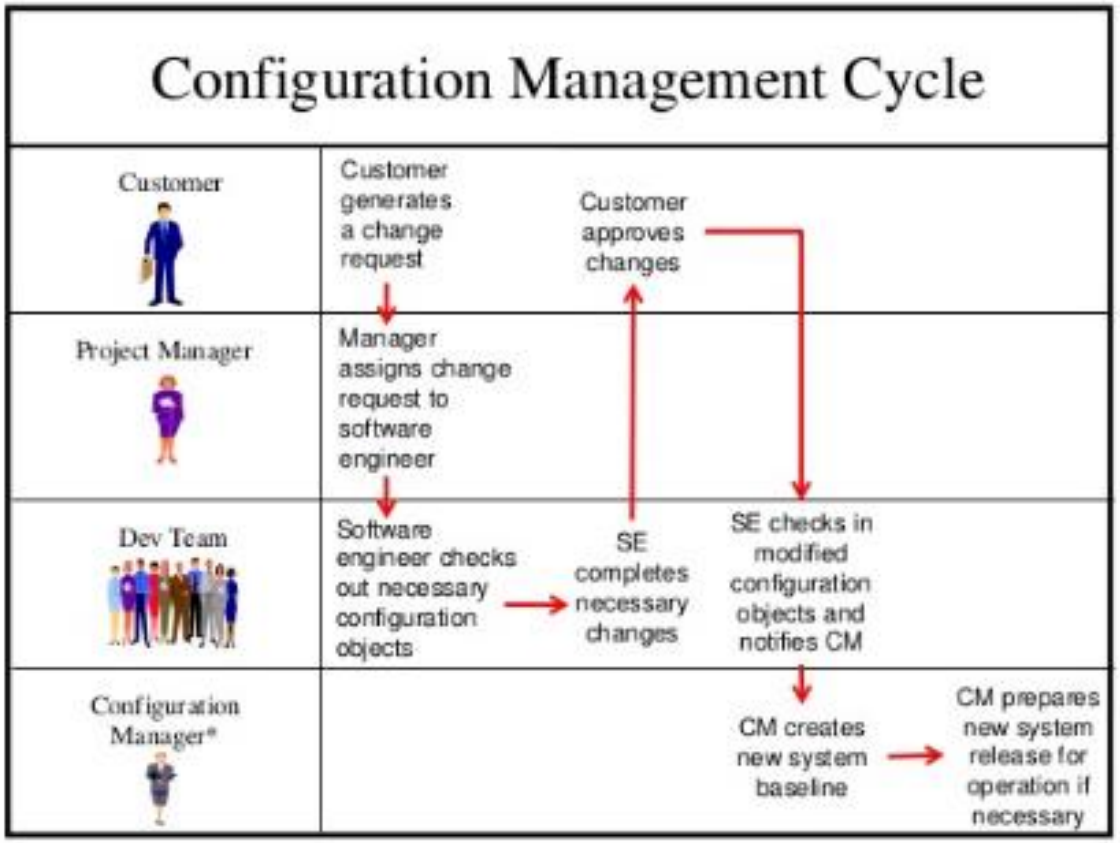
CONFIGURATION MANAGEMENT, SOURCE CONTROL MANAGEMENT

Software Configuration Management

Wikipedia: Software configuration management (SCM or S/W CM) is the task of tracking and controlling changes in the software. SCM practices include revision control and the establishment of baselines.



Rudimentary version control

Rudimentary – distributed - version control

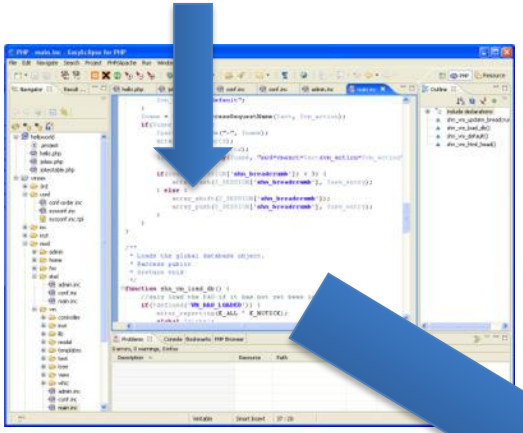
Proyecto_v1.zip

Proyecto_v1_final.zip

Proyecto_este_si_final.zip

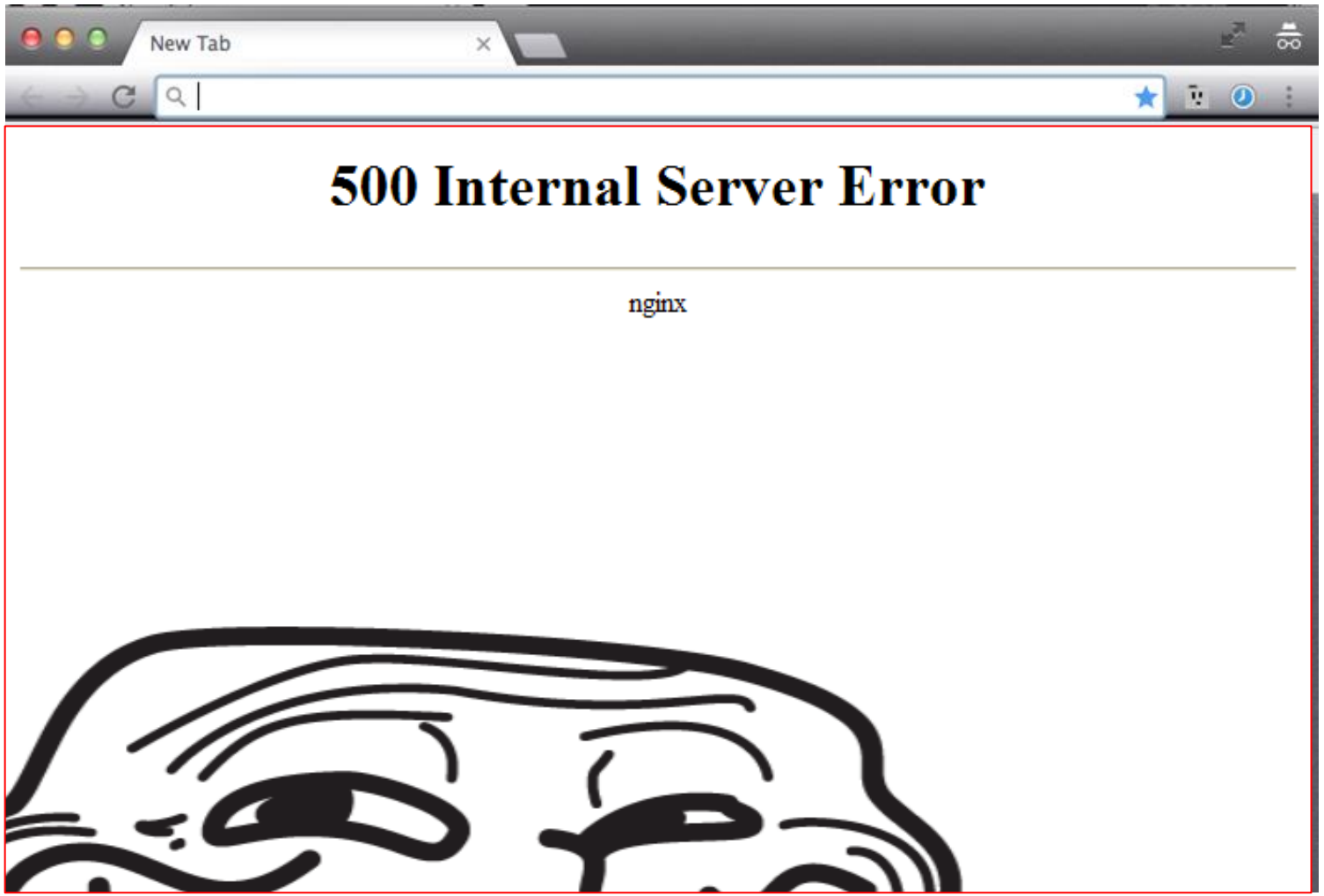
Project Delivery

Proyecto_este_si_final.zip

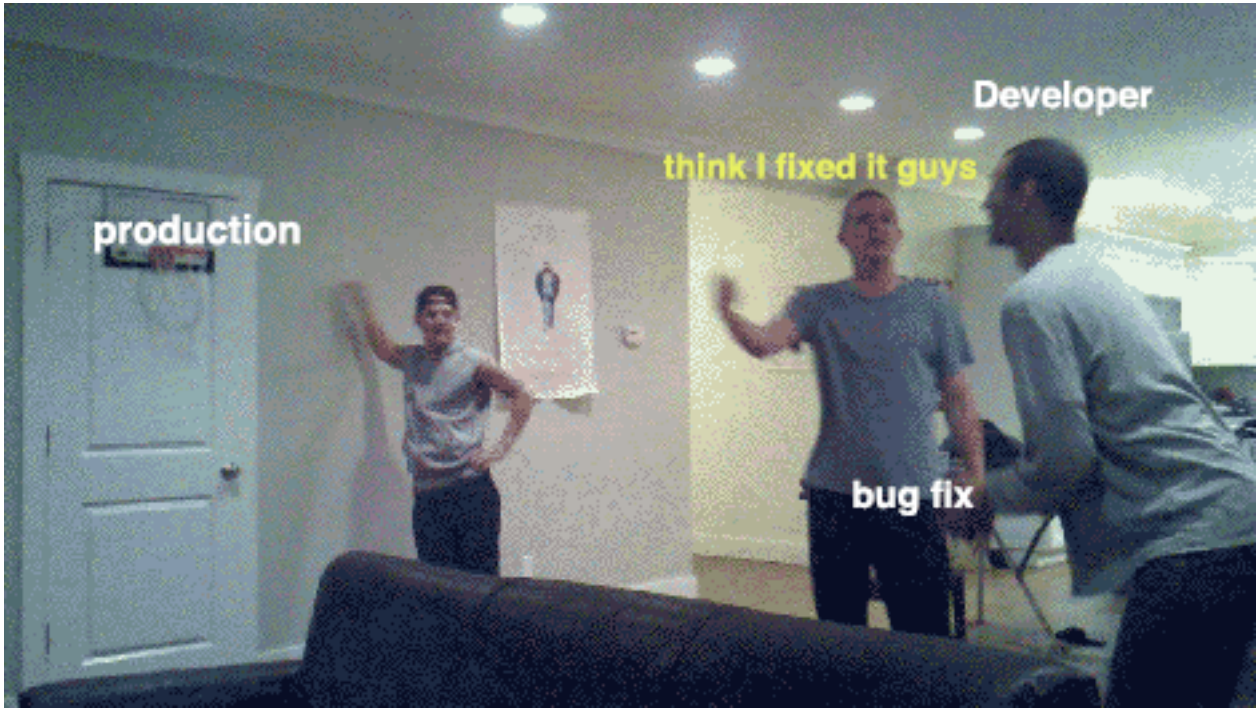


Proyecto1.war

Project Delivery



Project Delivery



¯_(ツ)_/¯

IT WORKS

on my machine

??????

Proyecto_v1_final.zip

Proyecto_v1.zip

Proyecto1.war

Proyecto_estesi.war

Proyecto_este_si_final.zip

Proyecto_mar23.war

Proyecto_funciona.war

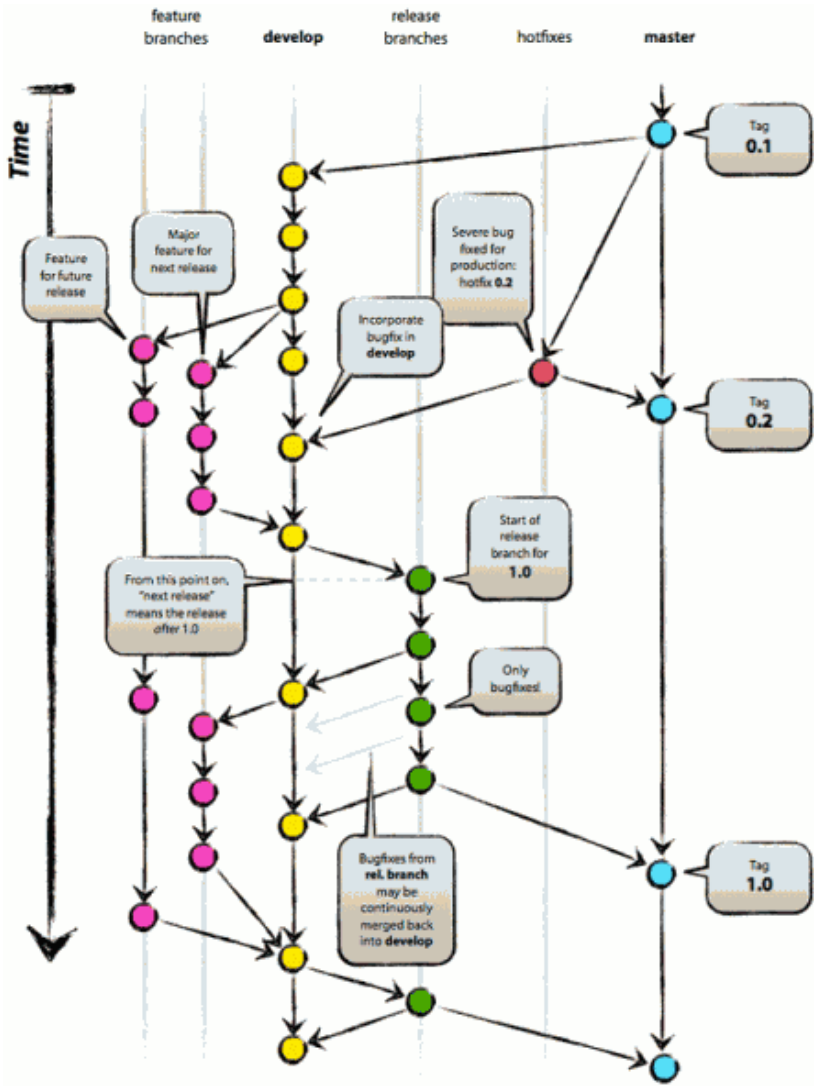
Continuous Integration

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.
Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.

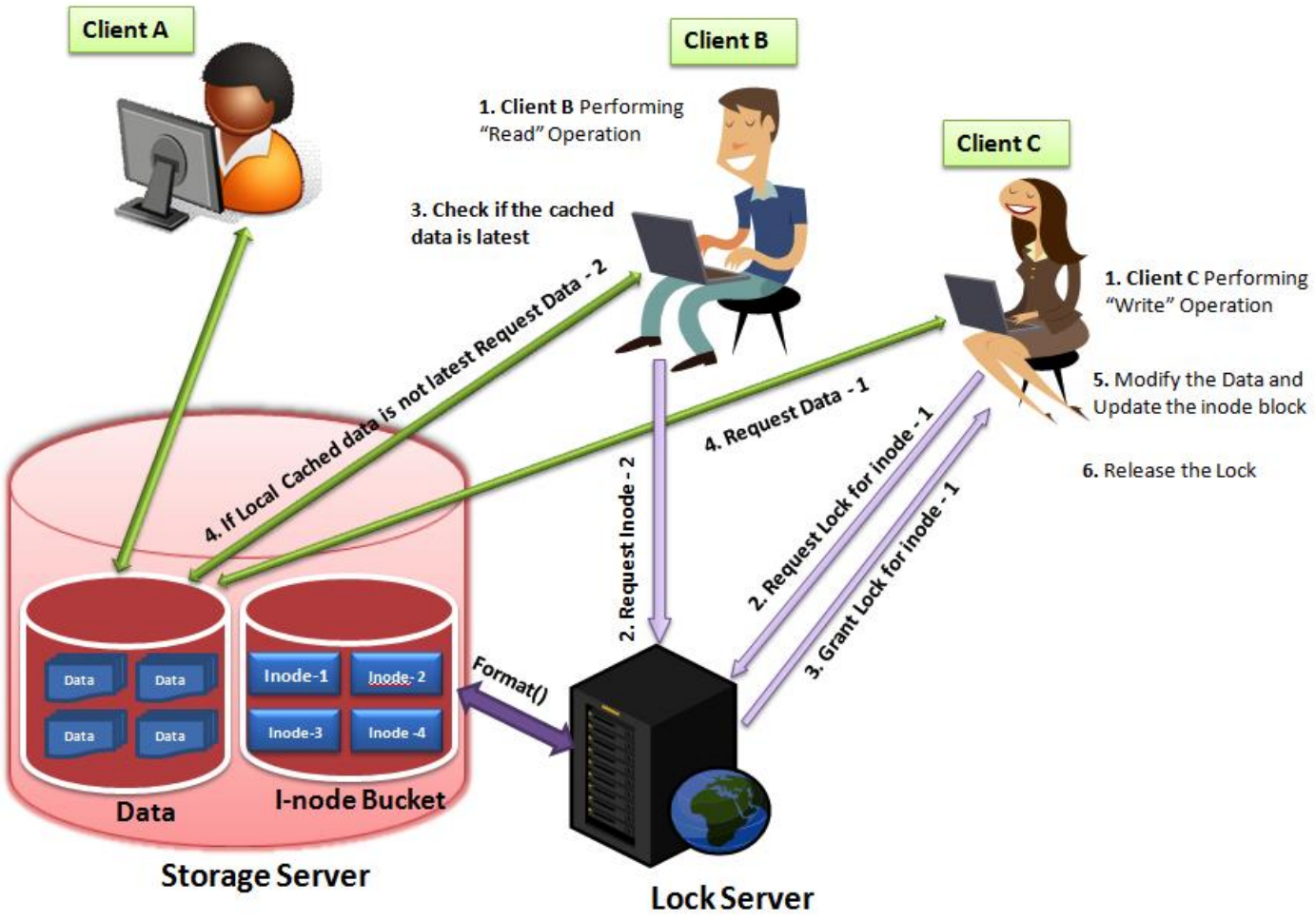
Continuous Integration - practices

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits every day
- Every commit (to mainline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

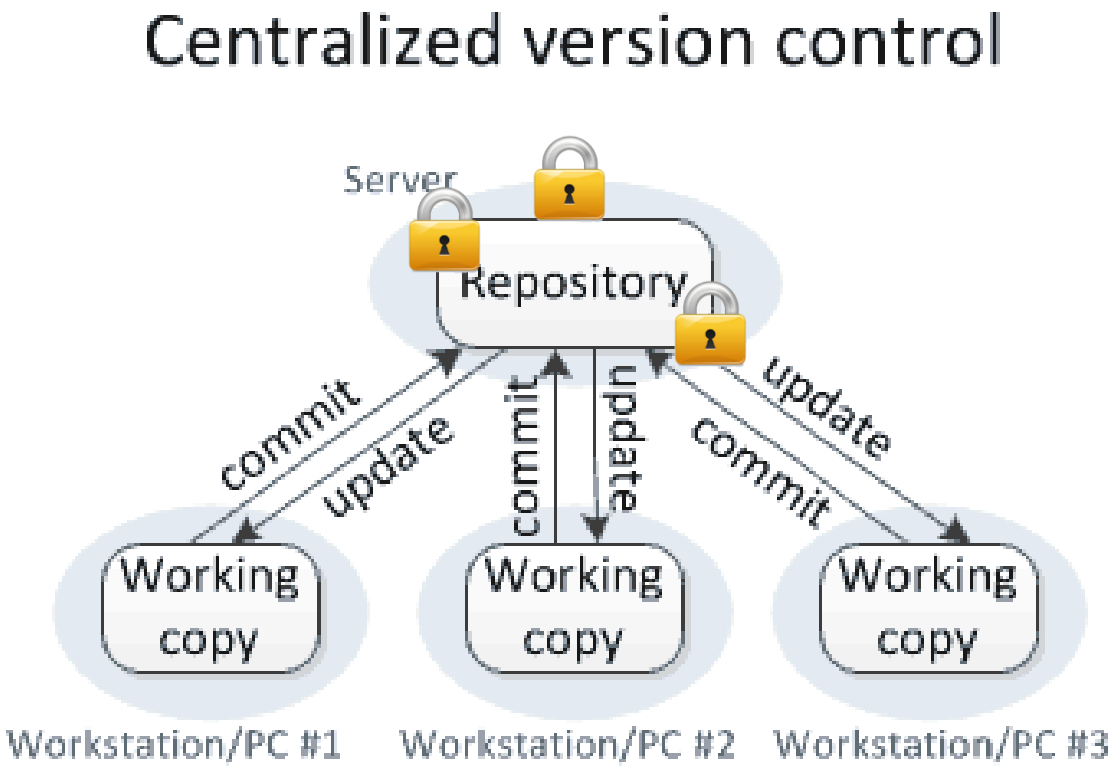
Source control



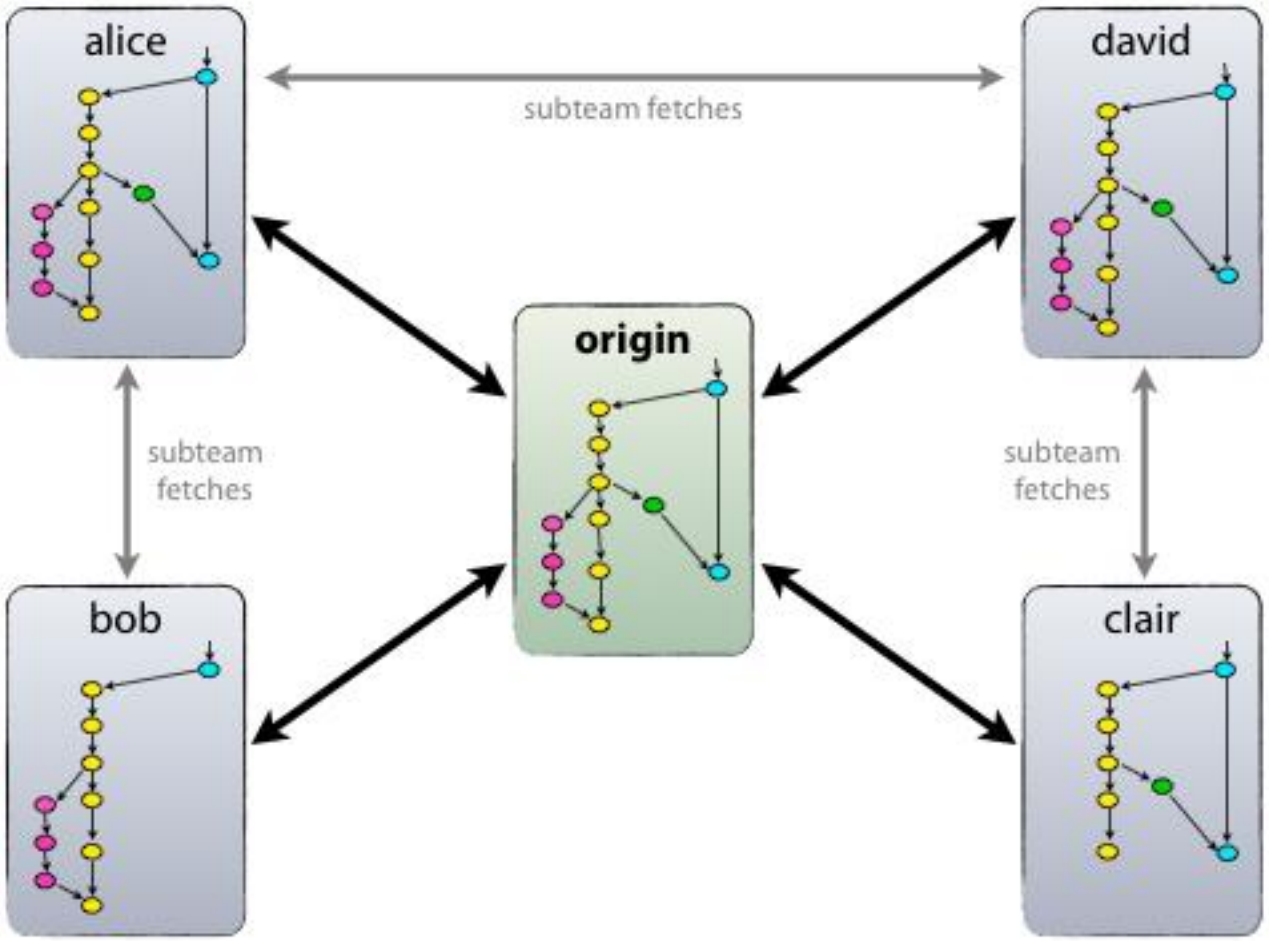
Concurrent (centralized) version control?



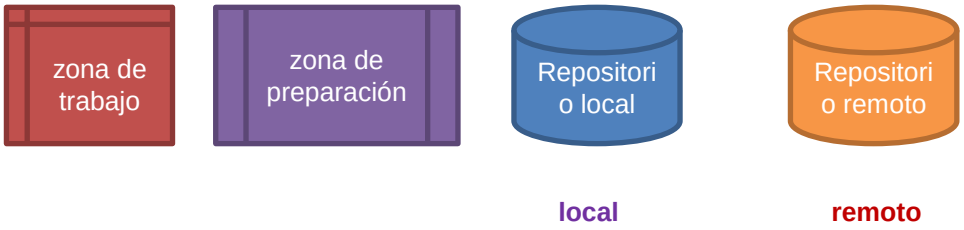
Centralized version control: RCS (Revision Control System)



Distributed Versioning System.



Arquitectura de GIT



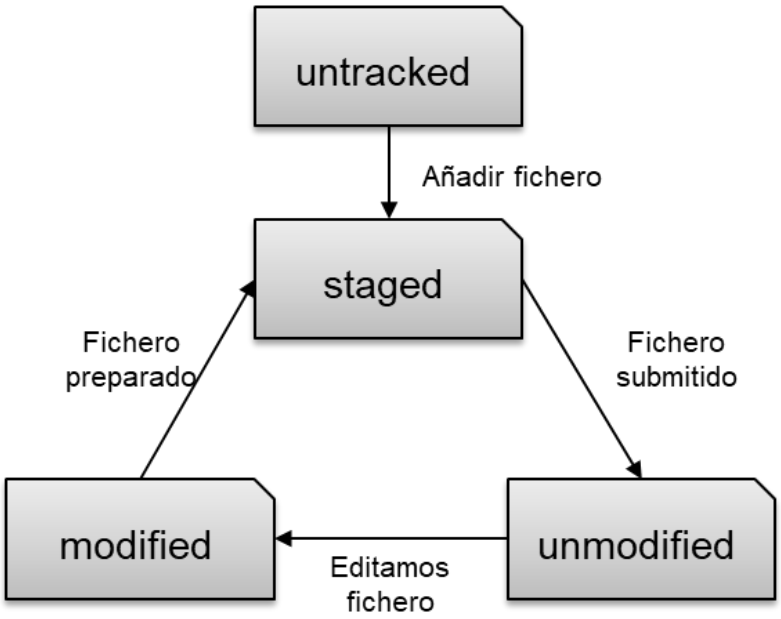
El sistema está compuesto por:

- Repositorio remoto: Versión del proyecto alojados en internet o en algún punto de la red.
- Workspace: Zona de trabajo local que contiene una réplica del código fuente
- Stage: Actúa como una zona de preparación de los ficheros, previo a realizar un commit a nuestro repositorio local.
- Repositorio local: Almacena versiones o commits, apunta al último commit realizado en nuestro repositorio local.

Ciclo de vida de los ficheros

Los estados principales en los que se pueden encontrar los archivos:

- Modificado (modified)
- Preparado (staged)
- Confirmado (commited)



Ignorar archivos

Fichero .gitignore

Reglas sobre los patrones que puedes incluir en el archivo .gitignore:

- Ignorar las líneas en blanco y aquellas que comiencen con #.
- Aceptar patrones glob estándar.
- Los patrones pueden terminar en barra (/) para especificar un directorio.
- Los patrones pueden negarse si se añade al principio el signo de exclamación (!).

¿Qué necesitamos para empezar a trabajar con Git?

REMOTO

Servidor de repositorios remotos para alojar nuestro proyecto, por ejemplo, los mas extendidos en el mercado actualmente:

GitHub

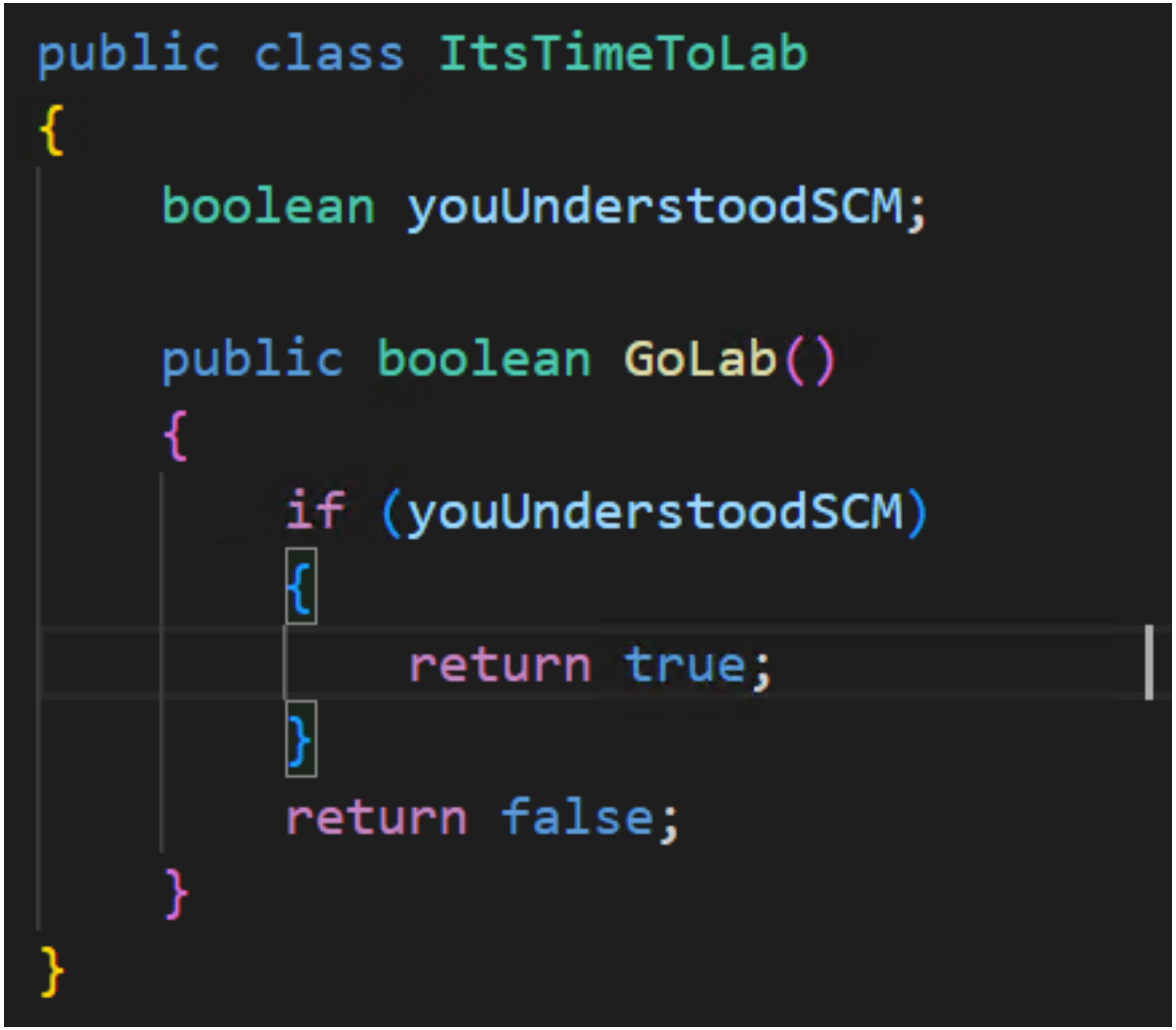
GitLab

ATLASSIAN Bitbucket

LOCAL

El motor de Git, el cual proporciona tanto la **versión de línea de comandos** como la interfaz gráfica de usuario estándar. Hay otros clientes que ofrecen mayor visibilidad:

Sourcetree


public class ItsTimeToLab
{
boolean youUnderstoodSCM;
public boolean GoLab()
{
if (youUnderstoodSCM)
{
return true;
}
return false;
}
}