**.ვერსიის მართვის სისტემა vcs.**

როდესაც ერთ დაწერილ კოდს ცვლი ანუ მის ვერსიას ცვლი, შეიძლება ერთ კოდზე ბევრმა დეველოპერმა იმუშაოს და ყველას ექნება საშუალება თვალყური ადევნოს პროექტის ცვლილებას

Cvs\_ის საჭიროება

1 - ცვლილებების თვალყურის დევნება

2 - გუნდის კოლაბორაცია

3 - ვერსიების ცვლილება

4 - განშტოებები შერწყმა

Branch - განშტოებები

**.GIT.**

გიტ ტექნოლოგია არის განაწილებული ვერსიის კონტროლის სისტემა DVCS, რომელიც გამოიყენება პროგრამის უზრუნველყოფის განვითარების დროს თვალყურის დევნებაში

Repository - დირექტორია რომელიც კონტროლდება გიტჰაბის მიერ რაშიც სხვადასხვა დეველოპერები იმუშავებენ სხვადასხვა ბრენჩებზე

**როგორ მუშაობს გიტი ატანა პლატფორმაზე**

1 - როდესაც რაიმეს დავწერთ ეს იქნება **სამუშაო დირექტორია working directory** თუმცა ეს არ იქნება კონტროლირებადი გიტის მიერ. working directory\_ში ფაილები გვაქვს ან **tracked** ან **untracked** თრექია ისეთი ფაილებსაც გიტი აკონტროლებს ხოლო ანთრექი როდესაც ვერა აკონტროლებს

2 - სამუშაო დირექტორიადან გადაგვაქვს **staging area**\_ში **add** ქივორდით რომ ცვლილება და თვალყურის დევნება იყოს შესაძლებელი,

3 - ხოლო აქედან **commit**\_ის შემთხვევაში ანუ დადასტურების შემთხვევაში ხვდება **ლოკალურ რეპოზიტპრიში local repository**

4 - ხოლო როდესაც ეს ყველაფერი დასრულდება შემდეგ აგვაქვს იმ პლატფორმაზე რაზეც გვინდა ანუ გითჰაბზე ან სხვა პლატფორმაზე ამას ეწოდება **დისტანციური რეპოზიტორია remote repository**

**1 – working**\_იდან add\_ით **2 – staging**\_ში commit\_ით **3 – local**\_ში push\_ით **4 – remote**\_ში

**პლატფორმიდან ჩამოტანა git fetch**

Remote repository\_დან local repository\_ში ჩამოსატანად გვინდა **fetch** ბრძანება მაგრამ არ იქნება ასახული და ასახვისთვის გვჭირდება **merge** ბრძანება და ამის შემდეგ აისახება working დირექტორიაში

თუმცა ამ ორი ბრძანების გაერთიანება შეიძლება **git pull**\_ის საშუალებით რაც ჩამოიტანს კიდეც და ასახავს კიდეც

ჯერ უნდა ჩამოვიტანო ფაილი ფულით, ვემუშაო და მერე ახლიდან ავტვირთო რეპოზიტორიაში და თუ მოხდა რომ ჩამოტვირთული ფაილი სხვამ ჩამოტვირთა ემუშავა და ატვირთა გამოვა რომ სხვა ვერსია მაქვს ფაილის და მოხდება ერორი

**Git pull –rebase** ბრძანება ჯერ commit\_იან ბრძანებას ჩამოიტანს და შემდეგ ატვირთავს ამას

**Git clone <url>** ჩამოიტანს მთლიანად ყველაფერს რაც კონკრეტულ რეპოზიტორიაში იქნება თუმცა ამ რეპოზიტორიაში ვეღარ ავიტანთ რადგან ეს რეპოზიტორია სხვას ეკუთვნის

**Diff** ბრძანებით შეგვიძლია ვნახოთ რა ცვლილება მოვახდინე working\_ში

**Diff –staged** ბრძანებით შეგვიძლია staging არეაში რაც გადატანილია იმის ცვლილებები ვნახოთ

**Diff <commit>^!** როდესაც კომიტი დადასტურება იქმნება სახელის გარდა მას ექმნება რაღაც აიდი და თუ ამ აიდის ჩავუწერთ ამ კომიტის ცვლილებასაც გვაჩვენებს

**ტერმინალით მუშაობა**

ტერმინალი დგას მისამართზე სადაც საქაღალდეა გახსნილი

Git status ვამოწმებთ რა მდგომარეობაშია საიტი

თუ დაწერა fatal ესეიგი არარის ლოკალურად მზადყოფნაში

Git init - ინიციალიზაციას გააუკეთებს და გახდის გიტის მიერ კონტროლირებადს