

Lumea Open Source

Descriere - Beneficii - Probleme

Ce inseamna Open Source?

Open Source - ce este?

- Open Source - ul (sau modelul “Sursa deschisa”) reprezinta o ideologie bazata pe un model descentralizat si deschis de creare a resurselor.
 - Deschis -> toate resursele deja create sunt valabile in mod public.
 - Resursa = orice element ce aduce valoare unui proiect, fie el o schita sau o linie de cod.
 - Descentralizat -> deciziile sunt transparente, luate cu ajutorul unei comunitati.
- Modelul de baza se numeste Colaborare Deschisa.
 - Un set de valori si idei ce identifica un anumit mod de a construi un proiect.
- Cand vorbim de Open Source, intelegem o referire directa la resurse deschise, publice, disponibile tuturor.

Open Source - ce este?

- Proiectele dezvoltate pe ideea de Colaborare Deschisa respecta urmatoarele principii:
 - Schimb transparent (public) de idei / munca.
 - Management colaborativ, bazat pe o comunitate.
 - Prototipizare rapida, deschisa.
 - Propunerea de schimbari la indemana tuturor din mediul public asociat.
 - Transparenta (in decizii, logica, procesare etc).
 - Meritocratie.
 - Putere de dezvoltare bazata pe o comunitate.

Open Source - origini?

- Miscare a pornit in momentul in care un grup mare de programatori ce nu erau constransi de drepturi de proprietate sau bani si-au dat seama ca pot produce un produs mult mai bun, feature-rich si cu o mai mica probabilitate de bug-uri daca formeaza o comunitate si colaboreaza la un nivel deschis.
- Ideologia Open Source a luat avant odata cu raspandirea Internet-ului. Odata cu raspandirea lui, au intervenit urmatoorii factori:
 - Crearea a tot mai multor servicii care necesitau un standard.
 - Rezolvarea unor probleme de multiple ori.
 - Monopol de piata si costuri mari in cazul firmelor cu tehnologii noi, closed-source si foarte raspandite.

Open Source Software - ce reprezinta?

- Orice **program / soft** a carui coloana vertebrala (sursa, planuri de proiectare, perspective de viitor, bug-uri etc) sunt disponibile la **nivel public**, sub **o anumita licenta**.
- Pe baza unor restrictii oferite de licenta, oricine poate prelua codul, il poate personaliza (in functie de licenta) si poate distribui (iarasi, in functie de licenta) propria versiune a programului.
- Conceptul se bazeaza pe analiza si imbunatatirea facuta asupra codului de alti programatori, parte a comunitatii. Comunitatea este o entitate publica cu volum volatil.
- In functie de popularitatea si gradul de folosire a unui software Open Source, procesul de eliminare a problemelor si de imbunatatire a programului poate fi mai accelerat fata de programele comerciale. Informatia este distribuita la nivel global in comunitatea open source, iar oricine poate contribui (sau cel putin raporta o problema).

Open Source Software

- Utilizatorii tratati ca si co-dezvoltatori
 - Au access la cod si sunt incurajati sa adauge functionalitati, sa imbunatateasca programul, sa raporteze probleme, sa documenteze.
 - Legea lui Linus: “Given enough eyeballs all bugs are shallow”

Open Source - licente <http://choosealicense.com/>

MIT

Oricine poate face modifica si folosi oricum codul atat timp cat autorul original este mentionat si nu este tras la raspundere.

Ruby on Rails, Node.js, jQuery, AngularJS, Haiku, Putty,

Apache 2.0

Similara cu MIT, dar in plus asigura drepturi de autor pentru colaboratori

Android, Apache, Swift,

GNU GPLv3

Oricine distribuie codul sau o versiune trebuie sa o faca sub aceeasi licenta, si de asemenea asigura drepturi de autor pentru colaboratori

Bash, GIMP, Privacy Badger

Open Source Software - mod de lucru

- **Pasul 1** - Un programator vine cu o idee.
- **Pasul 2** - Programatorul creaza un plan si o serie de specificatii pe care proiectul le va urma.
- **Pasul 3** - Programatorul creaza un mediu public in care toate resursele legate de acest proiect vor fi oferite in mod deschis.
- **Pasul 4** - Programatorul se apuca de treaba si atrage o comunitate in jurul proiectului.
- **Pasul 5** - In timp, se creaza un produs software ce poate fi folosit. Comunitatea creste.
- **Pasul 6** - Se gasesc bug-uri si se concep imbunatatiri.
- **Pasul 7** - Se reia pasul 4.

Open Source Software - mod de lucru

- **In vederea proiectarii:**

- Planurile sunt publice (daca exista).
- Oricine poate contribui la proiect, si o contributie poate insemna devierea planului intr-o anumita directie.
- Oricine poate decide sa duplice proiectul si sa plece pe o alta ramura (sau o alta perspectiva) pentru acel proiect.

- **In vederea contributiilor:**

- Contributiile la un anumit proiect sunt acceptate sau respinse de comunitate.
- Exista o serie de oameni de decizie (core developers) care aproba sugestiile exterioare in ramura publica sau de dezvoltare a proiectului.
- Orice contributie trebuie mai intai evaluata de un core developer (meritocratie).
- Dupa un anumit numar de contributii, poti urca in rank.

Open Source Software - mod de lucru

- **In vederea rezolvarii bug-urilor:**

- Bug-urile sunt raportate intr-un mod public (se numesc “Issue”-uri).
- Bug-urile sunt asignate si rezolvate de catre cineva.
- O raportare poate sa fie sau nu insotita de cod ce rezolva acel bug.

- **In vederea suportului asupra produsului software:**

- Software-ul Open Source, ca si cel comercial, nu acopera mereu toate cazurile.
- Nu orice software va fi Cross Platform.
- Nu orice software va fi stabil.
- Nu orice software va fi gata de folosit in productie.
- Nu orice software va fi pregatit pentru probleme de securitate.
- Nimeni nu isi asuma responsabilitatea pentru folosirea produsului.

Open Source Software - avantaje

- Control
 - Poti examina codul pentru a fi sigur de conformitatea cu specificatiile.
 - Se pot extrage partile redundante sau nefolositoare dintr-un software.
- Training
 - Codul poate fi studiat pentru a invata tehnici de dezvoltare software.
- Securitate
 - Perspectiva generala este ca open source software-ul este mult mai sigur decat cel comercial.
 - Pentru ca foarte multi programatori lucreaza la un loc, bug-urile sunt detectate si rezolvate mult mai repede (in functie si de cat de folosit este software-ul).
- Valabilitate
 - Chiar daca lucrul la produsul respectiv se opreste, codul exista si poate fi folosit.
- Cost

Open Source - dezavantaje

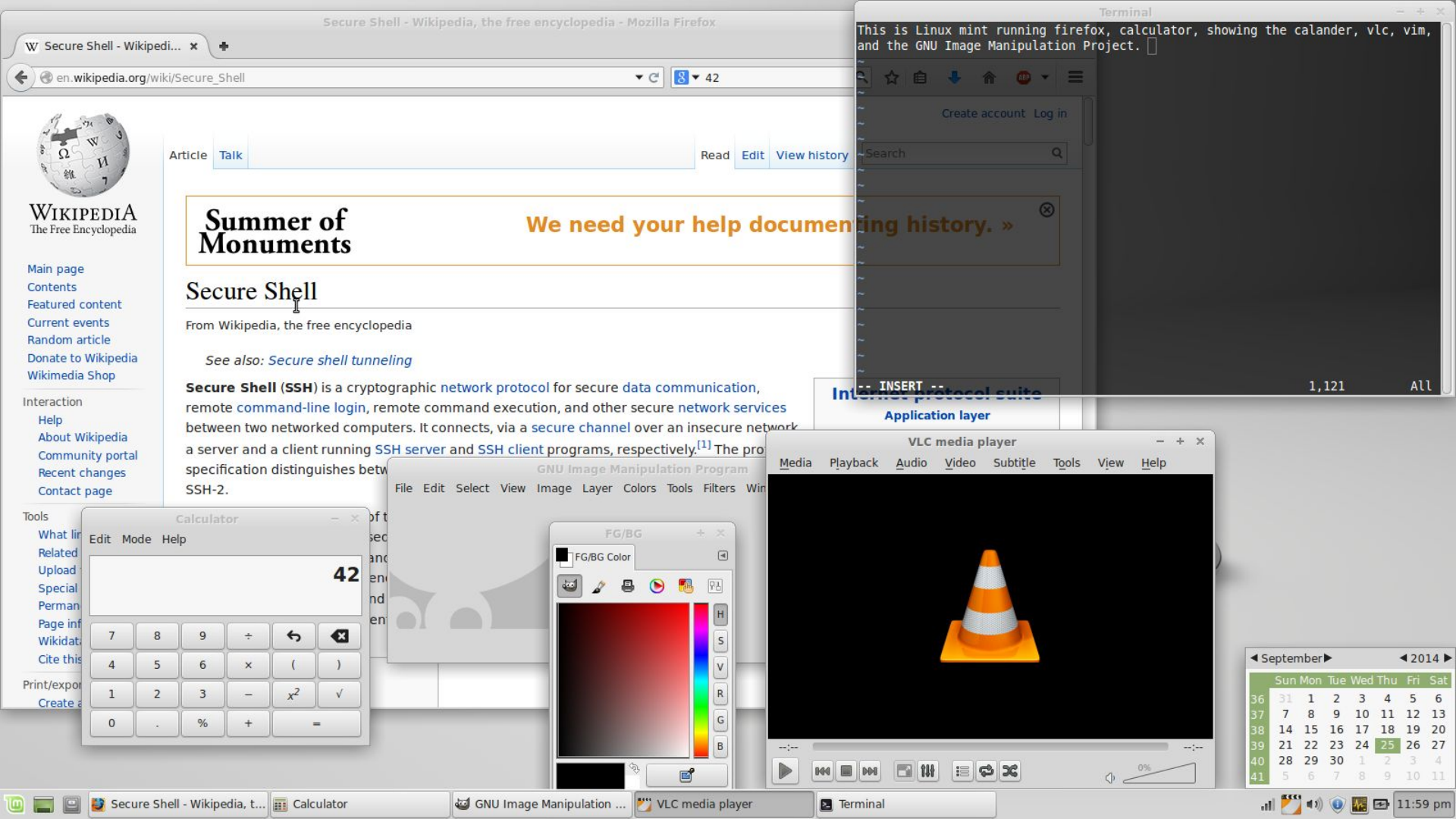
- Unele produse pot duce lipsa de documentatie.
- Unele produse pot fi incomplete / incomod de folosit.
- Incompatibilitate cu alte bucati de software sau hardware. Suportul redus in unele cazuri.
- Bad Code, si contributiile unor oameni necalificati - poate fi rezolvata cu o comunitate indeajuns de mare.
- Calitatea nu este asigurata de nimeni (nu are nimeni o raspundere directa asupra software-ului).
- Probleme de stabilitate. Nu toate produsele sunt testate indeajuns de riguros.

Open source - criterii

- Cod sursa deschis
- Redistribuire libera (in functie de licenta)
- Distribuirea licentei
- Meritocratie - asocierea codului sursa cu autorul
- Licenta nu trebuie sa aiba restrictii in privinta altor programe
- Fara discriminari impotriva unor persoane sau grupuri

Open Source - mituri

- Open Source Software-ul este tot timpul gratis.
- Open Source Software-ul nu este sigur sau nu ofera suport.
- Daca un software devine open source oricine il poate modifica.
- Licenta open source este prea liberala.
- Daca iti faci produsul open source, voluntari au sa-ti rezolve toate problemele gratis.
- Cineva imi va fura codul si se va imbogati.
- Nu poti face bani din Open Source. Nu poti trai din asta.



Application software :

- 7-Zip
- Eclipse
- GIMP
- Chromium
- Blender
- Mozilla Firefox
- Open Office

Operating Systems:

- Android
- Linux
- FreeBSD
- ReactOS
- Haiku
- FreeDOS

Programming Language :

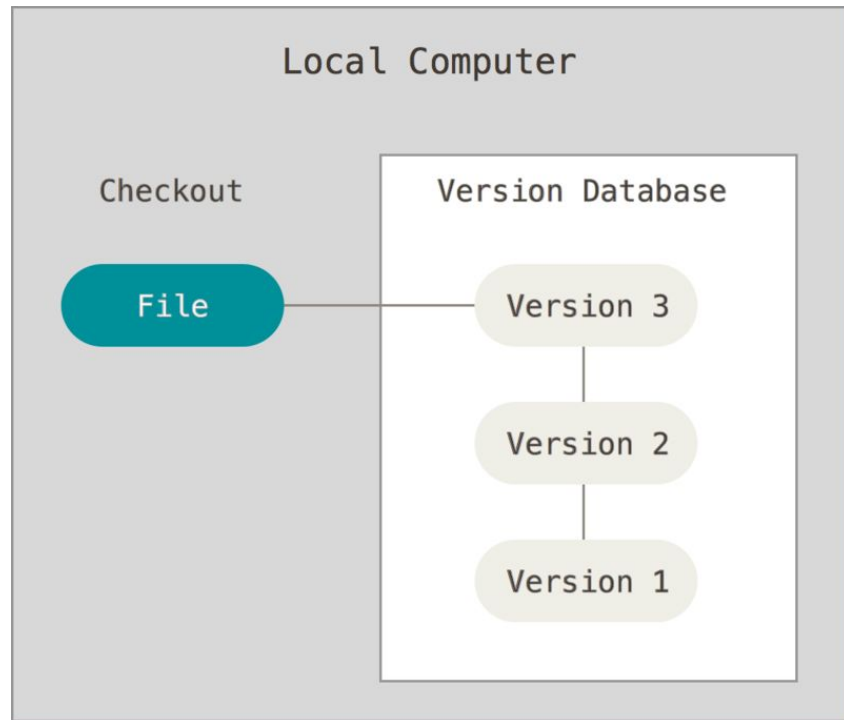
- Perl
- PHP
- Python
- Ruby
- PHDL
- Prolog

Sistem de control al versiunilor

- Un sistem de control a versiunilor este o colectie de programe ce ajuta o echipa de programatori sa gestioneze modificarile asupra codului unui proiect de-a lungul timpului
- Controlul versiunilor tine minte orice modificare asupra codului in baze de date specializate
- Atunci cand se fac greseli, programatorii pot “da timpul inapoi” si pot compara codul curent cu cel din trecut, eventual scriind din nou unele parti. In tot acest timp perturbarea celorlalti coechipieri este minima.

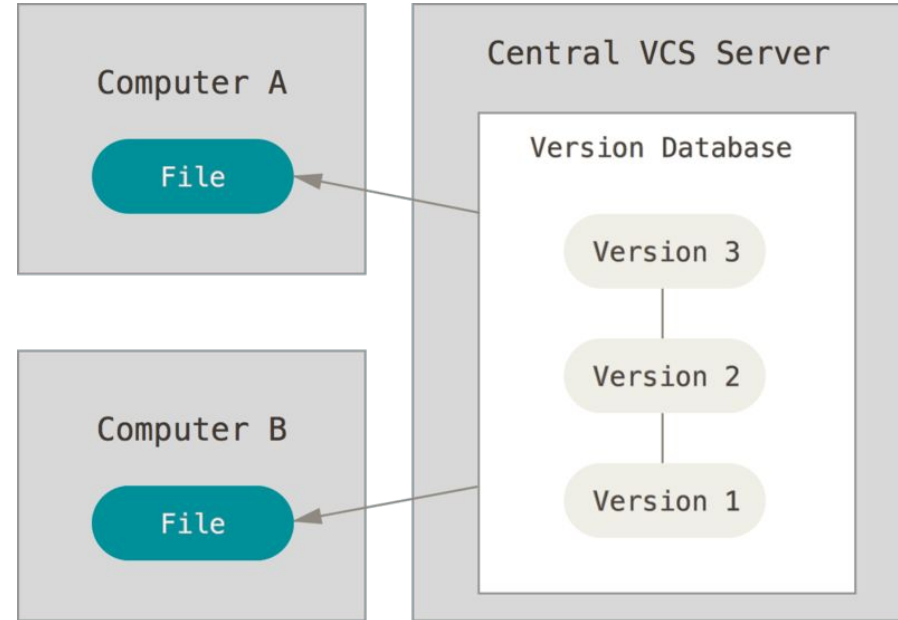
Sistem de control al versiunilor

- Initial - salvezi versiuni ale proiectului in dosare separate
- Problema - modfici in dosarul gresit, suprascrii un dosar
- Solutie - sistem de versionare local
- baza de date locala (rcs)



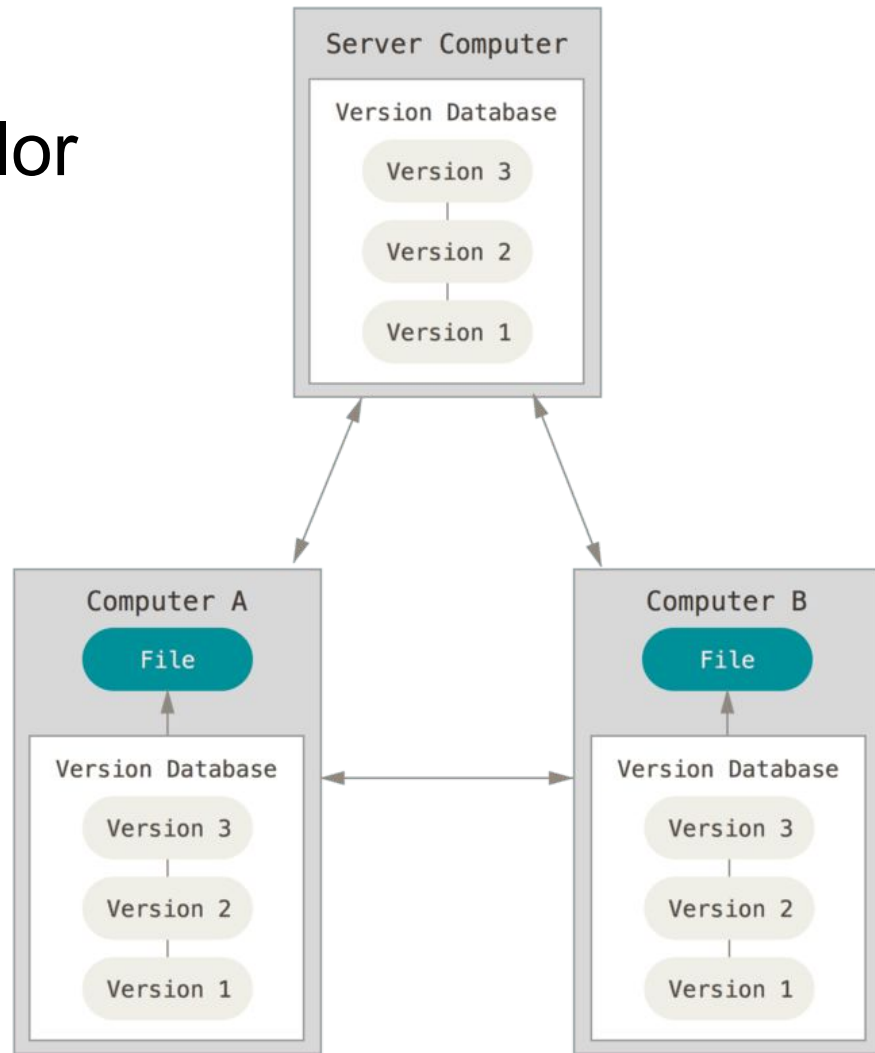
Sistem de control al versiunilor

- Problema - colaborezi cu alte persoane pe un proiect pe care doresti versionare (pe toate proiectele doresti versionare)
- Solutie - sistem centralizat de versionare
 - a. Mentii proiectul (impreuna cu versiunile acestuia) pe o masina accesibila de catre toti colaboratorii proiectului
 - b. CVS, Subversion, Perforce - a fost standardul pentru multi ani



Sistem de control al versiunilor

- Problema - “single point of failure”
- Solutie - sistem centralizat **distribuit** de versionare
 - a. Proiectul este mentinut pe mai multe masini in paralel
 - b. Fiecare mentine o copie actualizata a proiectului
 - c. Fiecare masina mentine o copie de rezerva
 - d. Git, Mercurial, Bazaar, Darcs

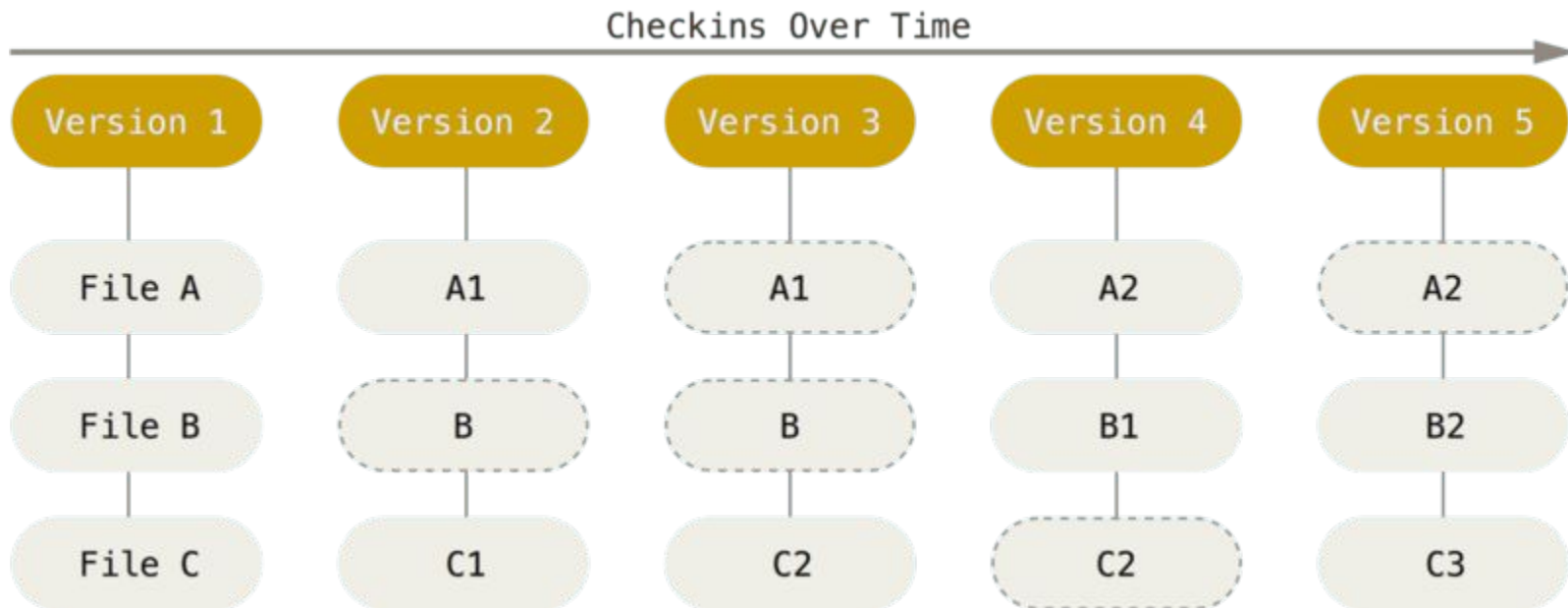


Sistem de control al versiunilor

- Integritatea datelor
 - a. Orice modificare a oricarui fisier este sesizata de git
 - b. SHA-1 hash
 - c. 24b9da6552252987aa493b52f8696cd6d3b00373
- Aproape toate operatiile sunt locale
 - Adaugare, stergere, modificare
 - Ocazional (preferabil dupa 2-3 commituri) - sincronizare a depozitului (repository) local cu cel centralizat

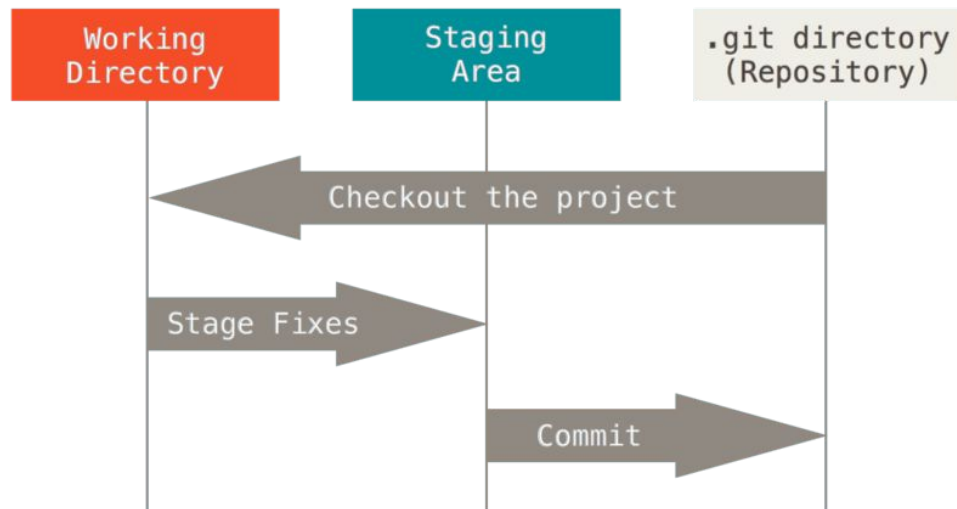
Sistem de control al versiunilor

- Instantaneu (nu modificari)



Sistem de control al versiunilor

- Cele trei stari
 - a. Modifici fisier local
 - b. Adaugi un instantaneu in zona de asteptare (staging)
 - c. Commit - salvezi un instantaneu permanent cu toate informatiile din zona de asteptare in copia locala

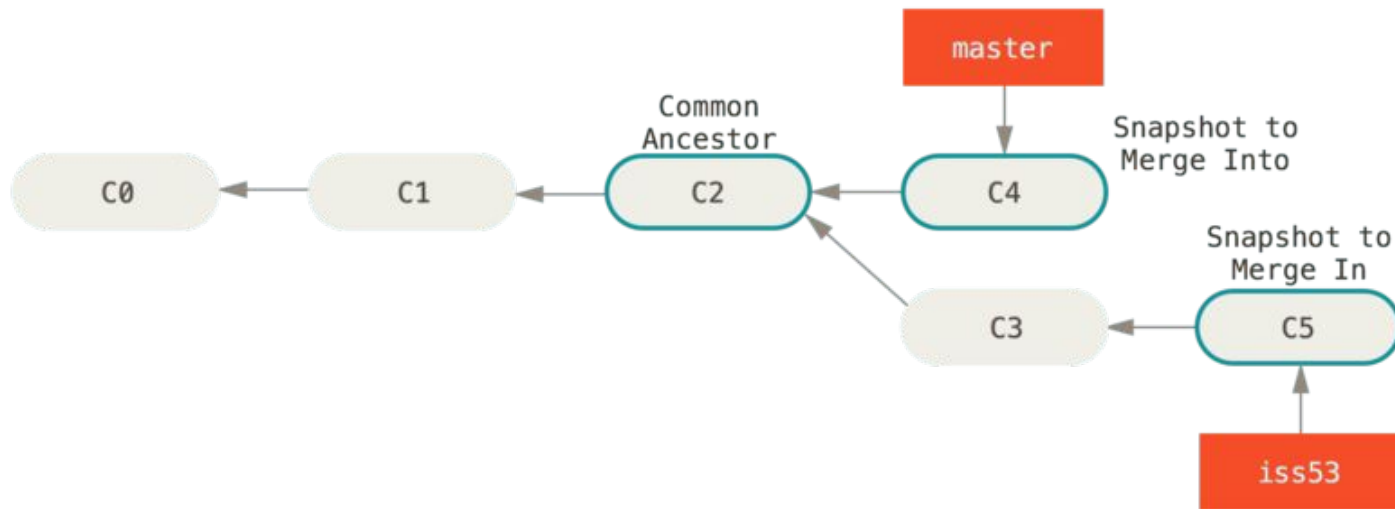


Ce este Github?

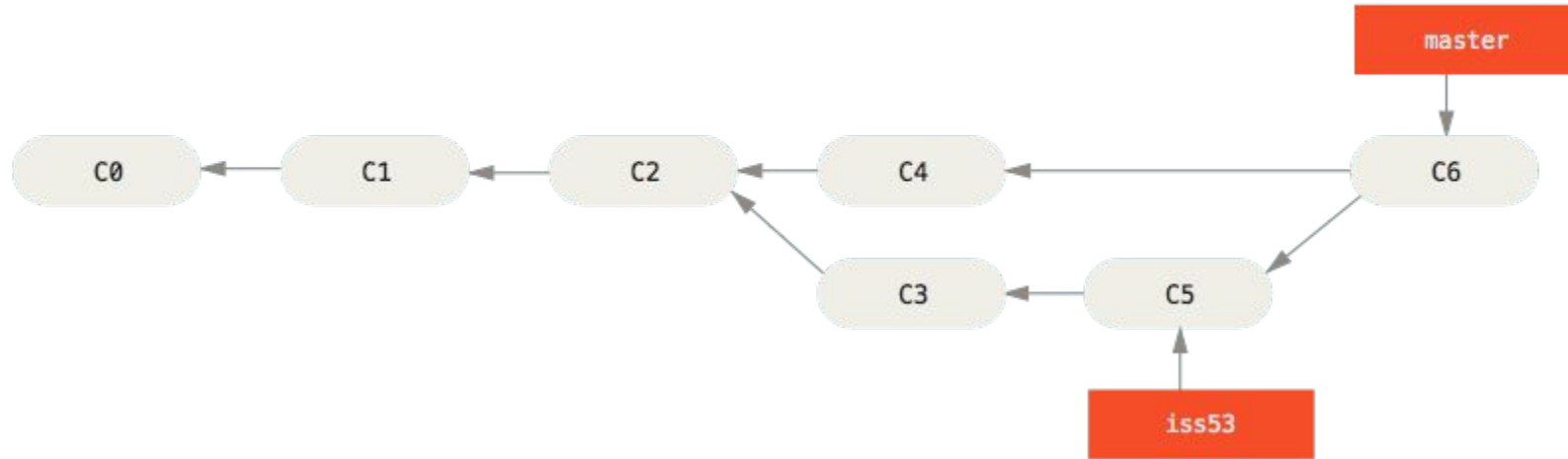
- Platforma de stocarea a codului sursa folosind Git
 - Stocheaza un istoric complet al tuturor modificarilor asupra unui proiect intr-un mod vizual.
 - Suporta deschiderea mai multor ramuri ale unui proiect (ramura de dezvoltare, ramura de productie, ramura de experimentare) pe baza feature-urilor de Git.
 - Suporta conceptul de organizatie (a.k.a. Comunitate de decizie)
 - Este gratuita, open source.

Ramuri - branches

- Lucrez in paralel la doua implementari plecand de la aceeasi stare
 - Implementez o functionalitate in doua moduri diferite - nu vreau sa le am pe amandoua in acelasi timp in cod (conflicte)
 - Adaug o functionalitate noua - nu vreau sa modific codul pe care lucreaza echipa - nu vreau sa risc buguri si/sau sa tin in loc colegii cu starea dezvoltarii featureului meu



Ramuri - branches



Bibliografie

- <https://git-scm.com/doc>
- <https://guides.github.com/activities/hello-world/>
- <https://try.github.io/levels/1/challenges/1>