

## Laborator 3 - Mijloace de organizare a proiectului. Controlul versiunii

---

Cum să ne organizăm pentru a atinge obiectivele definite în SRS conform planurilor din SDD?

---

În managementul de proiect, există o serie de utilitare de organizare a lucrului în proiecte, precum:

- **WBS**
- **Grafic Gantt**

### WBS (Work Breakdown Structure)

---

- structură arborescentă cu rolul de a determina și de a grupa **sarcinile proiectului** în funcție de categoria din care fac parte
- ajută la organizarea proiectului prin clarificarea obiectivelor, a **elementelor de lucru** mai generale (*work packages*) și mai specifice (*work elements*)
- pe baza WBS, se pot face estimări de durată, de resurse umane și de buget → *suport pentru construirea diagramei Gantt*
- ajută la identificarea relațiilor de precedență dintre elementele de lucru (care elemente de lucru sunt independente, care sunt restricționate să aibă loc înainte de altele, etc.) - elemente vizibile în diagrama Gantt.
- WBS reprezintă graful acțiunilor necesare unui proiect grupând acțiunile în categorii de interes
- oferă suport pentru identificarea tuturor acțiunilor necesare și prioritizarea lor în funcție de efort, cost, durată etc.

### Modul de construire a WBS

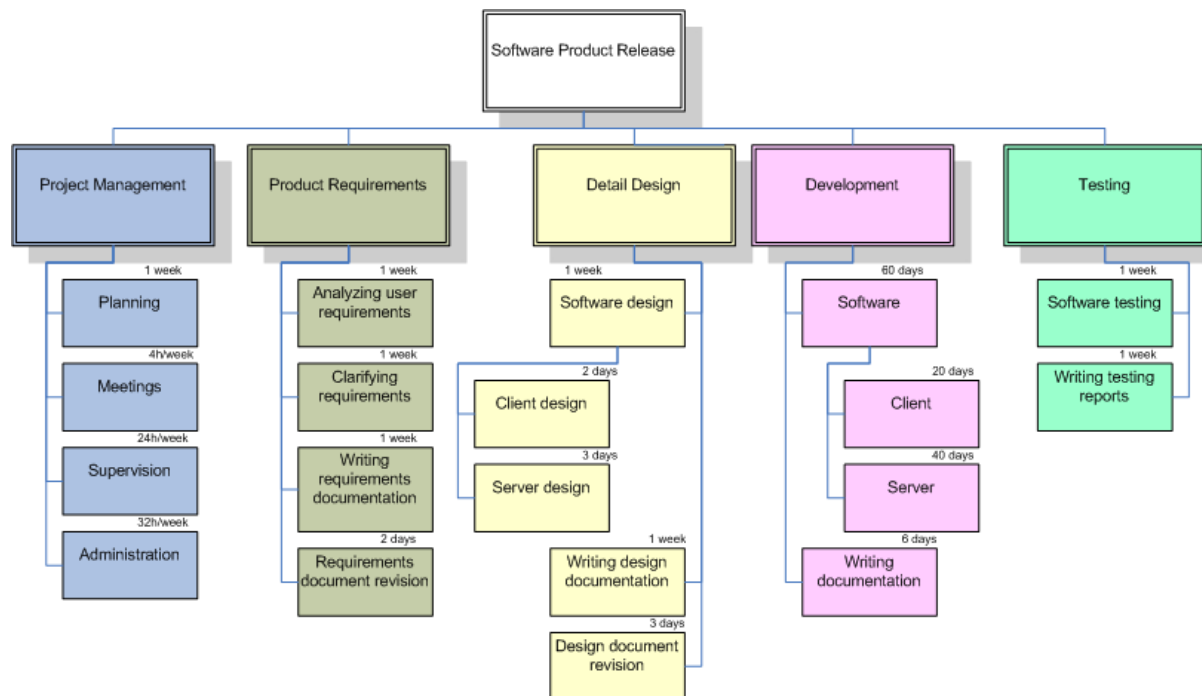
- nodurile arborelui conțin elementele de lucru, care pot fi produse, date, servicii
- rădăcina arborelui coincide cu obiectivul proiectului: sistemul software de dezvoltat
- nivelul al doilea al arborelui: subsistemele componente ale sistemului
- tot așa în adâncimea arborelui, în sensul că orice nod părinte înseamnă un element de lucru general, iar copiii reprezintă elemente de lucru componente ale lui

- deci, se detaliază efortul de dezvoltare a proiectului, de la obiectivul general la elemente de lucru și sarcini practice.
- nivelul de detaliere al unui WBS nu este foarte ridicat; astfel, adâncimea arborelui nu depășește, adesea, mai mult de 4-5 niveluri.

## Reguli

- un WBS trebuie să cuprindă prin elementele sale **tot** efortul necesar dezvoltării proiectului (100% din muncă)
- elementele de muncă de pe același nivel din arbore nu se acoperă total sau parțial (nu sunt *overlapping*)
- frunzele din arbore au adesea asociate resurse, durate și costuri.

**Tool-uri** de creare a WBS-urilor: [WBS Schedule Pro](#), [Online Visual Paradigm](#), [Planhammer.io](#)  
**Exemplu** de diagramă WBS a unui proiect software:



### Resources:

**Project Manager:**  
PM\_Name

**Requirements team:**  
Leads\_name,  
Members\_name

**Design team:**  
Leads\_name,  
Members\_name

**Development team:**  
Leads\_name,  
Members\_name

**Testing team:**  
Leads\_name,  
Members\_name

## Grafic Gantt

- grafic pentru **planificarea temporală** a sarcinilor proiectului

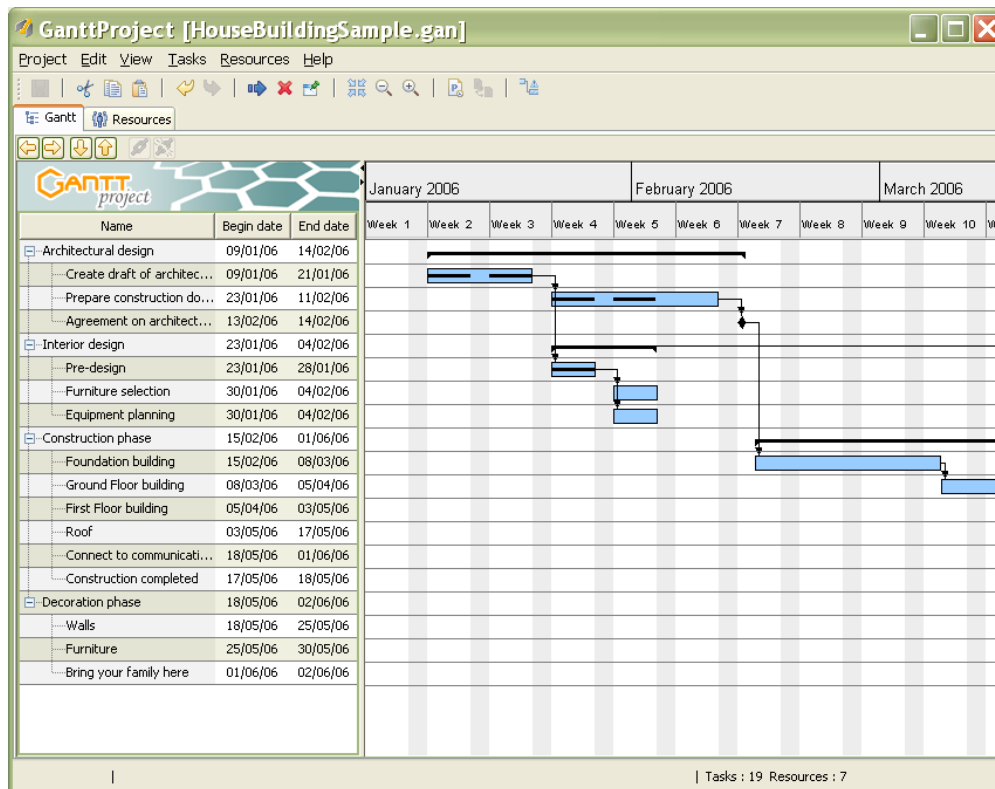
- construit pe baza activităților identificate în **WBS**
- indică data de început și de sfârșit a activităților din WBS și dependențele dintre activități:
  - start-to-start
  - finish-to-start
  - finish-to-finish
- întotdeauna se va ține cont de potențialele întârzieri și vor incluse în planificare sub o formă sau alta (review, delay, etc.)
- graficul Gantt este unul orientativ și starea sa se poate altera în timpul proiectului în funcție de situație
- acțiunile în grafic sunt trecute ca începând cel mai curând posibil, urmând să se asigure o marjă de eroare până la momentul cel mai târziu posibil
- este un tool bun pentru monitorizarea stării proiectului în timp, în funcție de gradul de realizare a acțiunilor până la un moment dat
- adesea, conține și următoarele elemente:
  - o linie verticală ce marchează timpul prezent
  - alte linii verticale cu milestone-urile proiectului
  - starea activităților exprimată prin procente (100% înseamnă că activitatea a fost realizată complet).

#### **Organizarea unui proiect înseamnă parcurgerea etapelor:**

1. Determinarea activităților proiectului (*WBS*)
2. Estimarea duratelor activităților și a consumului de resurse (astfel încât costul total să fie mai mic decât bugetul alocat)
3. Planificarea activităților în funcție de dependențe și de eventualele restricții legate de resurse (*Gantt*)
4. Stabilirea de *milestone-uri*.

**Tool-uri** de creare a diagramelor Gantt: [Gantt Project](#), [Team Gantt](#), [Ganttter](#), [Gantt Excel](#), [Microsoft Project](#).

**Exemplu** de diagramă Gantt a unui proiect de construcție de casă:



## Milestone

- deadline de finalizare a unei etape importante și decisive a proiectului
- adesea presupune predarea unor livrabile către beneficiar
- în vederea respectării milestone-urilor fixate, adesea se impune luarea unor decizii cu impact major asupra evoluției proiectului.
- deși reprezintă puncte cheie în cadrul unui timeline, din varii motive acestea pot fi decalate dacă acest lucru ajută și nu au un impact negativ asupra proiectului.

## Controlul versiunii (Git)

Sisteme pentru controlul versiunii (*Version Control Systems* - **VCS** - sau *Source Code Management* - **SCM**) sunt aplicații care permit lucrul colaborativ pe diverse fișiere, în special fișiere cod sursă. Sistemele pentru controlul versiunii sunt practic obligatorii în cadrul unui proiect cu dezvoltatori multipli. Astfel de sisteme rețin istoricul modificărilor efectuate de fiecare dezvoltator și folosesc comenzi specializate care să faciliteze transmiterea acestor modificări între dezvoltatori.

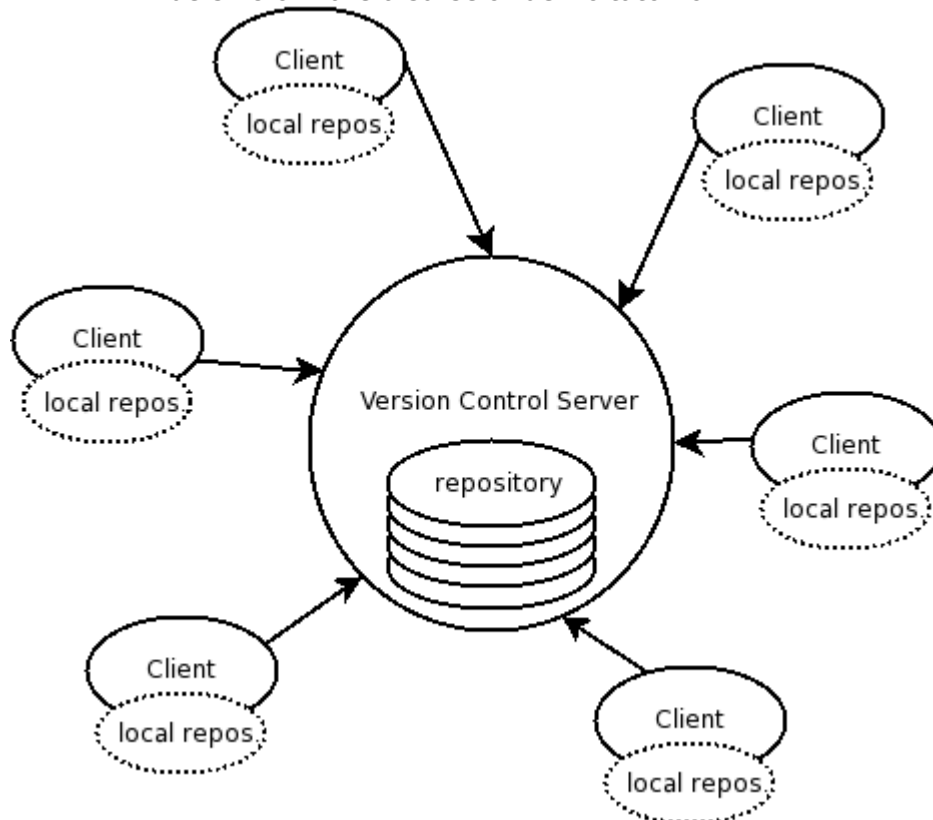
Exceptând sistemele de gestiune a surselor, prezentate mai detaliat în continuare, și alte aplicații folosesc versiuni:

- Wiki Engines - pentru fiecare modificare se rețin doar schimbările de la versiunea anterioară; orice modificare poate fi anulată;

- Google Docs - versiuni pentru fiecare modificare;
- MS Office, OpenOffice - permit atașarea unor numere de identificare pe documentele editate.

Principiul de funcționare a sistemelor de gestiune a codului este comun:

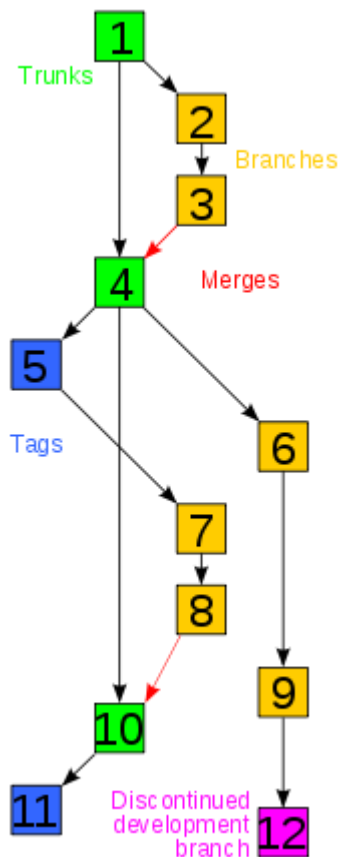
- Sursele sunt păstrate, de obicei, într-un **repository (depozit)** aflat pe un server accesibil tuturor dezvoltatorilor. Acest repository constituie mecanismul principal de sincronizare a surselor dezvoltatorilor.



- Fiecare dezvoltator obține o copie a repository-ului, denumită **copie locală**. Copia rezidă pe sistemul dezvoltatorului. Dezvoltarea se va realiza în acest director.
- Există două operații care permit comunicația cu repository-ul:
  - **update** sau **pull** înseamnă actualizarea copiei locale cu informațiile din repository; este posibil ca alți dezvoltatori să fi comis schimbări în repository;
  - **commit** sau **push** înseamnă transmiterea/comiterea modificărilor locale în repository; ceilalți dezvoltatori vor folosi comanda **update** pentru actualizarea copiei locale cu acele informații.

Câteva cuvinte cheie utile în lucrul cu sisteme de control al versiunii sunt (denumirile sunt în engleză pentru că aceasta este forma uzuală de utilizare):

- **repository** locul (centralizat) în care se găsesc sursele și informațiile de modificare;
- **working copy** copia locală a repository-ului folosită de un dezvoltator;
- **check-out** operația de creare a unei copii locale pornind de la repository (clonarea repository-ului);
- **commit** (vezi mai sus)
- **update** (vezi mai sus)
- **branch** o ramură de dezvoltare (un fork) care poate fi dezvoltată în paralel față de ramura principală de dezvoltare (vezi [imaginea de mai jos](#));
- **trunk** ramura principală de dezvoltare (vezi [imaginea de mai jos](#));
- **tag** o referință la un snapshot al surselor la un moment dat în timp;
- **merge** o operație de unificare a două seturi de schimbări petrecute distinct;
- **conflict** situație în care două modificări sunt efectuate din surse diferite și sistemul nu poate găsi o soluție de unificare (merge) a acestor schimbări; dezvoltatorul va trebui să rezolve conflictul fie prin unificarea schimbărilor, fie prin selectarea unei singure variante



Există două tipuri de sisteme pentru gestiunea codului sursă:

- sisteme centralizate ([Subversion](#), [CVS](#), [perforce](#));
- sisteme distribuite ([Git](#), [Darcs](#), [Mercurial](#), [Bazaar](#)).

Detalii despre diferențele dintre acestea (mai degrabă între doi dintre cei mai cunoscuți reprezentanți, Subversion și Git) găsiți

[aici](#).

## Git

Git este unul dintre cele mai folosite sisteme de versionare distribuite. Cele mai importante concepte din Git sunt următoarele:

- repository - există un repository remote și oricâte repository-uri locale
- commit - conținut modificat
- branch - lanț de commit-uri