Testarea produsului program. Asigurarea calității produsului program.

Testare:

este procesul de utilizare a unui produs software pentru a verifica dacă acesta respectă cerințele specifice, cu scopul de a detecta erori.

Asigurarea calității:

Calitatea?

- **Calitatea** este un concept complex care variază în funcție de context și înseamnă lucruri diferite pentru persoane diferite.
- Pentru utilizator, calitatea înseamnă satisfacerea nevoilor și cerințelor sale.
- Pentru producător, calitatea înseamnă conformarea cu specificațiile tehnice.
- Cu alte cuvinte, calitatea reprezintă valoarea pe care produsul o aduce pentru cineva.
- Testarea nu garantează automat calitatea! Ea măsoară doar nivelul de calitate al unui produs.

Atributele calității:

???

Atributele calității:

Atributele calității sunt caracteristicile sau trăsăturile care definesc gradul de calitate al unui produs sau serviciu. Acestea pot varia în funcție de contextul în care sunt aplicate, dar în general includ următoarele aspecte:

Fiabilitatea

Fiabilitatea unui obiect (unei componente, unui proces, unui sistem) exprimă siguranța sa în funcționare. Este o funcție de timp F(t)F(t)F(t), definită ca probabilitatea ca, în condiții înconjurătoare specificate, obiectul (componenta, procesul, sistemul) să funcționeze adecvat, menținându-și parametrii prestabiliți în intervalul de timp [0, t). Practic, fiabilitatea este o probabilitate (de bună funcționare), cu o valoare cuprinsă între 0 și 1.

Eficiența

Capacitatea produsului software de a oferi performanțe adecvate în raport cu resursele disponibile și condițiile de utilizare specificate.

Portabilitatea

Ușurința cu care un produs software poate fi transferat de pe un echipament pe altul sau dintr-un mediu în altul.

Folosință

Capacitatea produsului software de a fi înțeles, învățat și utilizat cu ușurință, precum și de a fi atractiv pentru utilizator în condiții specifice.

Mentenabilitate

Ușurința cu care un produs software poate fi modificat pentru a corecta defecte, a adăuga noi cerințe, a facilita mentenanța viitoare sau a se adapta la un mediu schimbat.

Modulul 1:

Bazele testării produselor-program - 10 ore

Modulul 2:

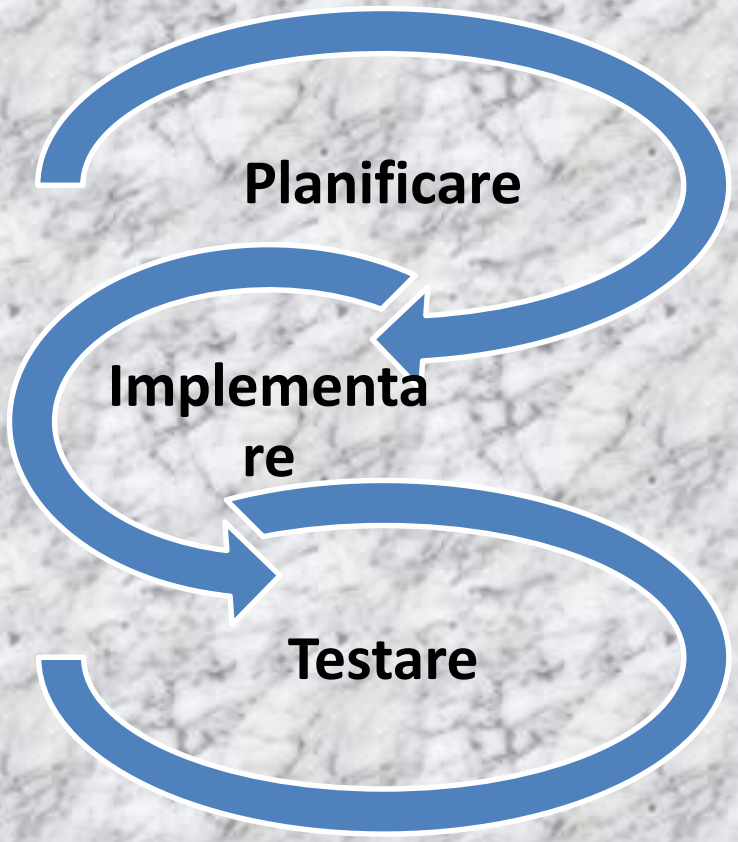
Strategii de testare - 20 ore

Curs 1:

- **Procesul de dezvoltare a produselor program;**
- **Terminologia din domeniul testării produselor program;**
- **Ciclul de viaţă al dezvoltării produselor program;**
- **Modele ale ciclului de dezvoltare a produselor program.**

Procesul de dezvoltare a produsului software:

CLASIC



Procesul de dezvoltare a produsului software:
MODERN

Terminologia din domeniul testării produselor program

- **Bug** - O problemă, eroare sau defect într-un sistem software care împiedică sistemul să funcționeze conform așteptărilor. Un bug reprezintă orice problemă care nu este documentată și afectează funcționalitatea sistemului.
- **Bug report** - Un raport care detaliază un bug într-un program, inclusiv modul în care se manifestă, momentul în care apare și partea de cod responsabilă. De asemenea, poate include o soluție posibilă.
- **Bug tracking** - Procesul organizat de urmărire a bugurilor și a stărilor acestora (deschis, rezolvat, testat, închis). Bug tracking-ul poate varia de la forme neorganizate (întâlniri în echipă) la soluții organizate (liste de discuții, e-mailuri, soluții specializate).
- **Sistem de bug tracking** - O aplicație software destinată să faciliteze bug tracking-ul, oferind o metodă centralizată și ușor de utilizat pentru gestionarea bugurilor. Există soluții gratuite, dar și opțiuni enterprise cu prețuri ridicate. Un sistem de bug tracking este un subtip al sistemelor de issue tracking.
- **Sistem de issue tracking** - O aplicație software destinată centralizării și gestionării tuturor cererilor legate de unul sau mai multe produse, inclusiv probleme, schimbări de design și aplicări de patch-uri. Cele mai multe sisteme de issue tracking includ și funcționalități de bug tracking. În general, acestea sunt mai complexe decât simplele bug trackere.

Ciclul de viaţă al dezvoltării produselor program:

Ciclul de viaţă al unui produs software reprezintă intervalul de timp de la momentul deciziei de realizare şi până la retragerea sau înlocuirea totală a acestuia cu un nou produs. Acesta include toate etapele în care produsul software este activ și evoluează. Conform glosarului de termeni al IEEE (Institute of Electrical and Electronics Engineers), ciclul de viață al software-ului reprezintă o abordare sistemică care acoperă dezvoltarea, utilizarea, mentenanța și retragerea software-ului.

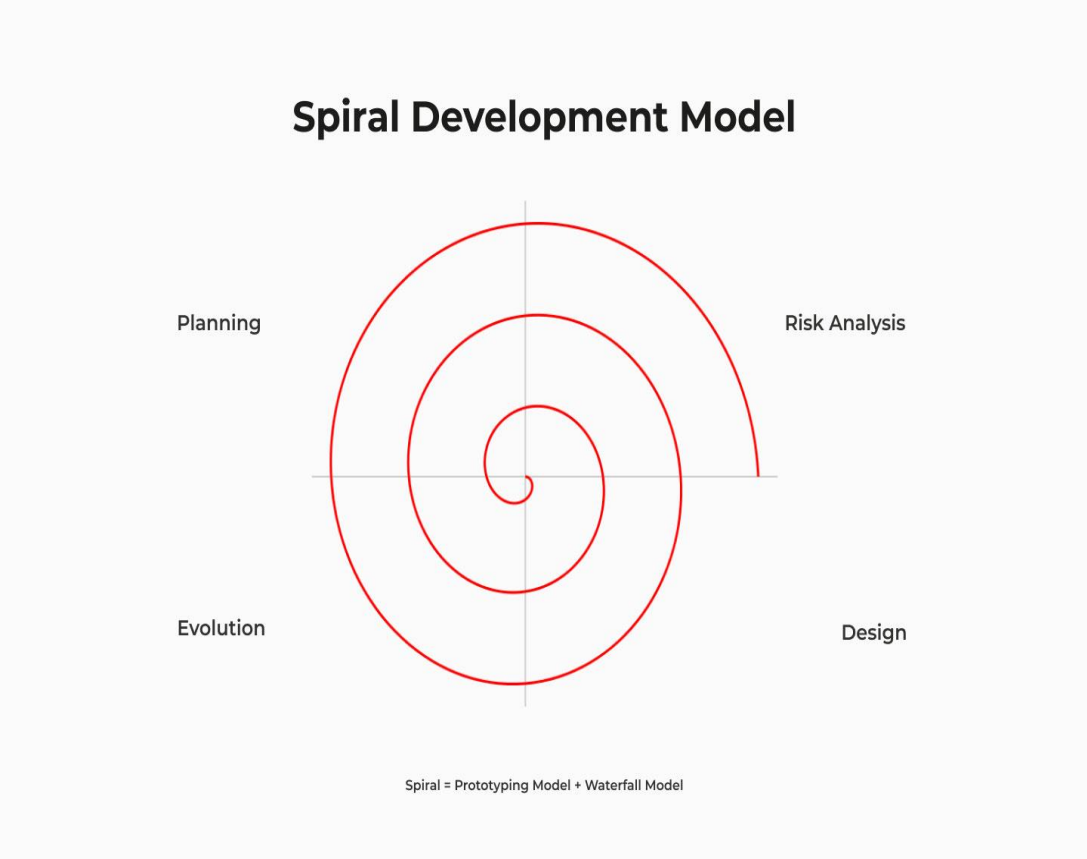
Ciclul de viaţă al dezvoltării produselor program:

Etape	Obiective	Produse finale
Cerințe utilizator	Definirea problemei	Specificații cerințe utilizator
Cerințe software	Analiza problemei	Cerințe și specificații software
Proiectarea arhitecturii	Soluții de ansamblu	Proiect de ansamblu
Producție	Implementare	Proiect de detaliu, Software testat
Transfer	Instalare	Software instalat, training clienți.
Mentenanță	Evoluție produs software	Software întreținut și dezvoltat

Modele ale ciclului de dezvoltare a produselor program:

- **Clasice** – modelul spirală(spiral), modelul cascadă(waterfall), modelul V.
- **Moderne** – agile (sprinten), integrare continuă (continuous integration).

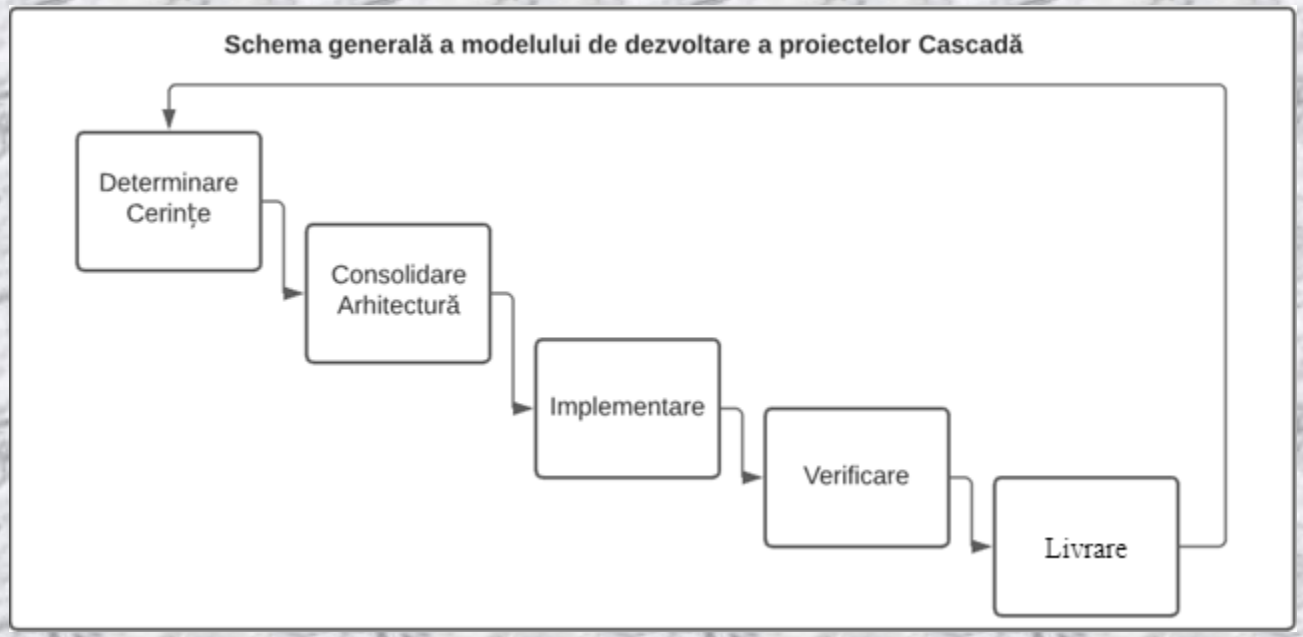
Modelul spirală:



Avantaje:
Începe cu puține specificații descrise în detaliu și adaugă treptat informații și funcționalități.
Costuri mici de descoperire a defectelor.
Răspuns timpuriu din partea clienților.
Testarea nu este presată de timp la finalul procesului.

Dezavantaje:
Nu este aplicabil pentru proiecte mici.
Succesul produsului depinde de analiza riscurilor.
Poate fi un model costisitor.

Modelul cascadă:

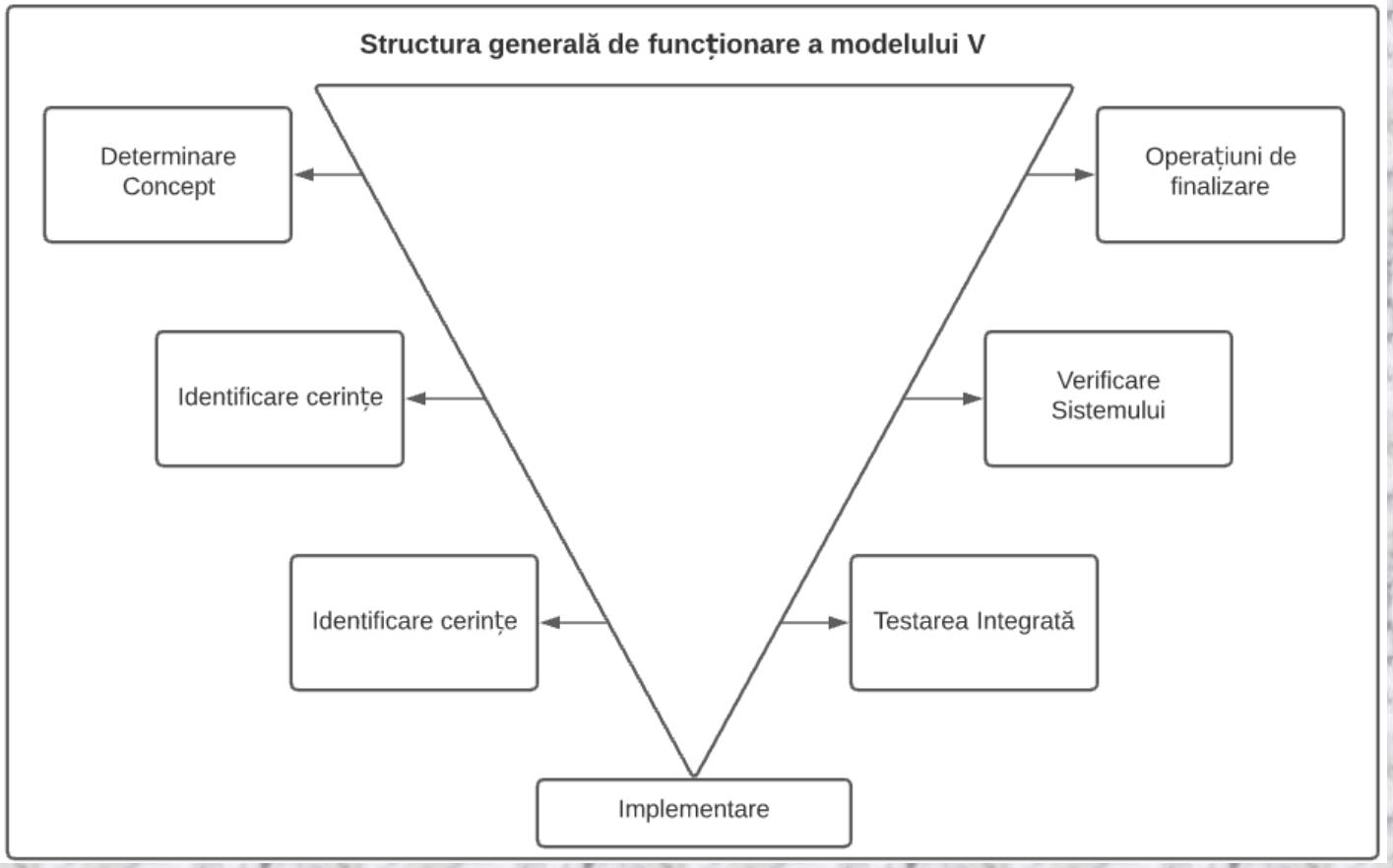


Valoarea modelului Cascadă nu poate fi neglijată, având în vedere istoria sa lungă și proiectele de succes realizate în domenii unde riscurile sunt minimizate, cum ar fi medicina și domeniul militar.

Cu toate acestea, în contextul accelerării livrării soluțiilor software și al revoluției digitalizării, modelele tradiționale se dovesc a fi mai puțin eficiente și prezintă următoarele provocări:

- **Perioade lungi de timp** necesare pentru livrarea unui produs care să aducă valoare reală beneficiarului.
- **Riscul de a produce un software** care nu mai corespunde cerințelor actuale ale pieței, având în vedere dinamica rapidă a acesteia.
- **Consum excesiv de resurse financiare**, adesea depășind oportunitățile economice, reflectate prin coeficientul pragului de rentabilitate.
- **Riscuri crescute de identificare a erorilor** cu prioritate și severitate ridicate în ultima fază a ciclului de viață, ceea ce poate duce la retragerea întregii echipe la fazele anterioare și la reînceperea lucrului de la zero.

Modelul V :

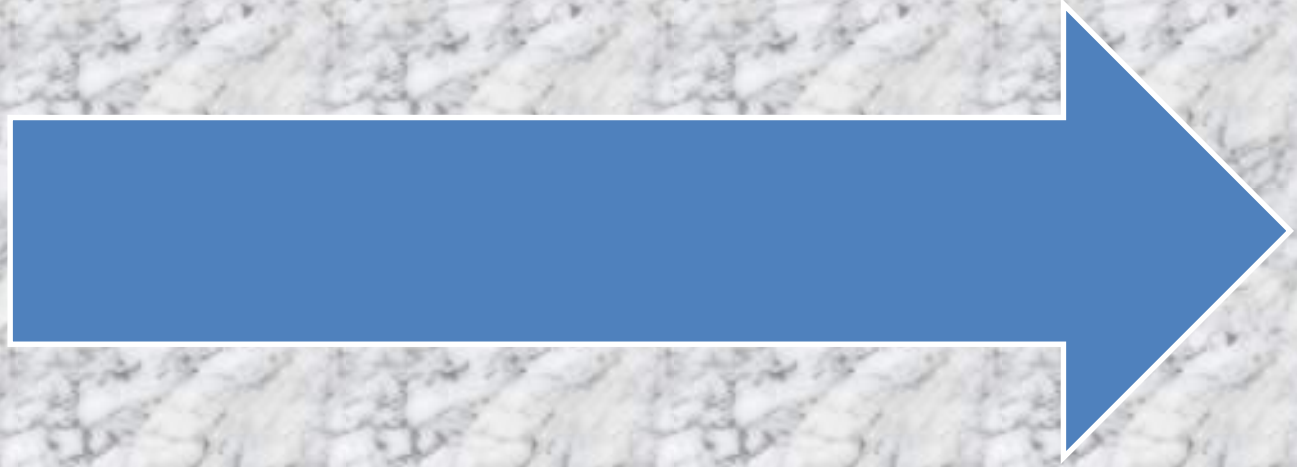


Modelul V a moștenit structura pas cu pas de la modelul cascadă. Modelul în formă de V este aplicabil sistemelor pentru care funcționarea lină este deosebit de importantă.

De exemplu, aplicații clinice pentru monitorizarea pacientului, software integrat pentru mecanisme de control al accidentelor vehiculelor și așa mai departe. O caracteristică a modelului poate fi considerată faptul că vizează verificarea și testarea amănunțită a unui produs care se află deja în etapele inițiale de proiectare. Nivelul înalt de precizie cu care se întreprind acțiunile din cadrul ciclului de viață permit modelului V să se prezinte ca fiind unul foarte avantajos pentru organizațiile ce nu-și asumă riscuri înalte, ceea ce nu este corespunzător pentru subiecții cercetării în cauză – organizații care pun la dispoziție soluții IT în timp restrâns și în puternică concordanță cu schimbările dinamice ale pieței. Timpul reprezintă sursa majoră care împiedică organizațiile să-și asume responsabilitatea adaptării modelului V. Posibilitatea de a răspunde în timp util la schimbările pieței, presupun identificarea unui model care este suficient de adaptiv și rămâne a păstra standardele de calitate indiferent de timpul restrâns de livrare.

Agile:

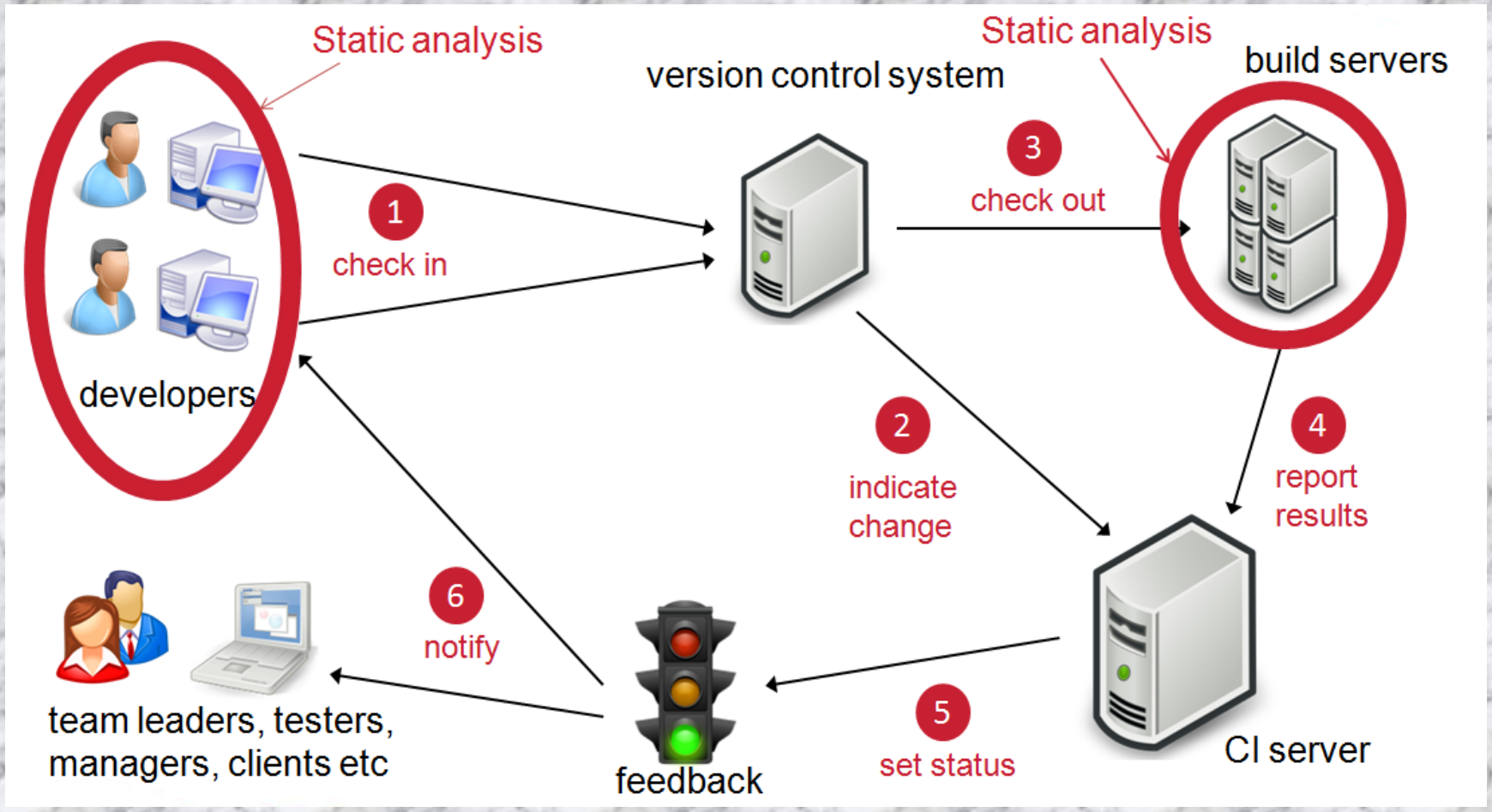
Agile (capabil să se adapteze schimbărilor) este o metodologie de dezvoltare software care implică multiple iterații. În cadrul unei abordări „agile”, după fiecare iterație, clientul poate observa rezultatul și poate evalua dacă acesta îndeplinește așteptările. Acesta este unul dintre beneficiile unui model flexibil, care permite ajustări rapide în funcție de feedback-ul primit.



Integrarea continua:

- **Integrarea continuă (Continuous Integration - CI)** este o practică din ingineria software care implică unificarea frecventă a codului din spațiile de lucru ale dezvoltatorilor într-un depozit comun, de mai multe ori pe zi.
- Scopul principal al CI este de a preveni problemele de integrare, cunoscute sub denumirea de „integration hell”. CI poate fi considerată o extensie a practicilor de integrare periodică promovate de metodele de dezvoltare software bazate pe procese incremental-iterative.

Integrarea continua:



Controlul versiunilor:

- **Controlul versiunilor** (în engleză: *version control* sau *revision control*) este un domeniu software care se ocupă cu gestionarea diferitelor versiuni (sau revizii) ale fișierelor. Este utilizat în mod frecvent în programare pentru a păstra versiuni succesive ale codului sursă al unui program. Deși arhivarea separată și completă a fiecărei versiuni într-o bază de date sau pe un suport de stocare extern ar fi o soluție posibilă, aceasta ar necesita, în general, prea mult spațiu de memorie. În schimb, se utilizează tehnici speciale care reduc spațiul necesar și permit reconstrucția „în zbor” a oricărei versiuni din istoria programului, la cerere.

Controlul versiunilor - Terminologie**:**

- ***repository*„depozitul“** în care sunt păstrate fișierele curente și versiunile anterioare. Deseori acest depozit este o bază de date găzduită pe un server.
- ***working copy*** (copie de lucru)copie a fișierelor din *repository* pe calculatorul de lucru al unui dezvoltator (de unde și numele). Acestea sînt fișierele pe care lucrează un dezvoltator în mod obișnuit.
- ***check-out*** operația de creare a unei copii de lucru luate din *repository*
- ***commit*** sau ***check-in*** operația de introducere în *repository* a schimbărilor din copia de lucru
- ***update*** (actualizare)introducerea în copia de lucru a schimbărilor făcute de alte persoane (colegi la același proiect)
  la *repositorybranch* (ramificare)bifurcarea unui set de fișiere în două căi de dezvoltare distincte
- ***merge*** (integrare, împletire)unirea a două versiuni diferite ale unui aceluiași fișier într-o singură versiune
- ***Tag*** o „etichetă“ aplicată fișierelor din *repository* la un anumit moment important din "viața" programului, de exemplu la lansarea unui produs

Controlul versiunilor – aplicații**:**

- **CVS (Concurrent Versions System)**: Un sistem centralizat de control al versiunilor care a fost popular în anii 1990 și începutul anilor 2000. CVS păstrează un istoric al modificărilor pentru un set de fișiere, de obicei codul sursă al software-ului, și facilitează colaborarea unui grup de persoane la același proiect.
- **Subversion**: Un sistem open source care a apărut în anii 2000, creat pentru a înlocui CVS. Subversion remediază erorile și limitările întâmpinate de CVS, oferind îmbunătățiri în gestionarea versiunilor.
- **Git**: Un sistem de control al versiunilor dezvoltat pentru a gestiona eficient codul sursă și alte fișiere în diferite scenarii. Git permite crearea de soluții personalizate pentru controlul versiunilor și interfețe utilizator. De exemplu, Cogito este un pachet care facilitează utilizarea Git, iar StGit utilizează Git pentru a gestiona colecții de patch-uri.
- **GitHub**: Cel mai mare serviciu web pentru găzduirea și dezvoltarea proiectelor IT, bazat pe sistemul de control al versiunilor Git. GitHub, dezvoltat de GitHub, Inc (anterior Logical Awesome), este construit folosind Ruby on Rails și Erlang. Serviciul este gratuit pentru proiectele open source și, începând cu 2019, pentru proiectele private mici, oferind toate caracteristicile necesare, inclusiv SSL. De asemenea, GitHub oferă opțiuni de plată pentru proiectele corporative mari.

Curs 1:

- **Procesul de dezvoltare a produselor program;**
- **Terminologia din domeniul testării produselor program;**
- **Ciclul de viaţă al dezvoltării produselor program;**
- **Modele ale ciclului de dezvoltare a produselor program.**