Inginerie software

Lab. 1

Inginerie software - definitii

- o ramură a informaticii care se ocupă cu proiectarea, implementarea și întreținerea programelor complexe
- un proces de dezvoltare a unui produs software într-o abordare sistematică, bine definită

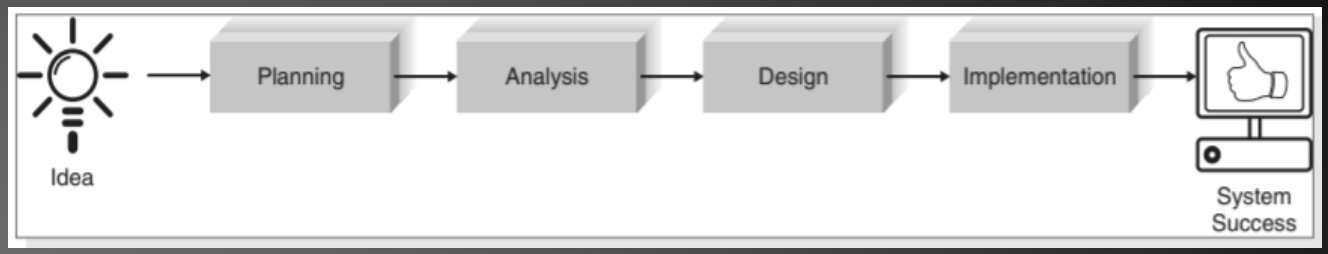
Scopul ingineriei software

- de a rezolva criza de software în care acesta este livrat cu întârziere, cu defecte și peste buget

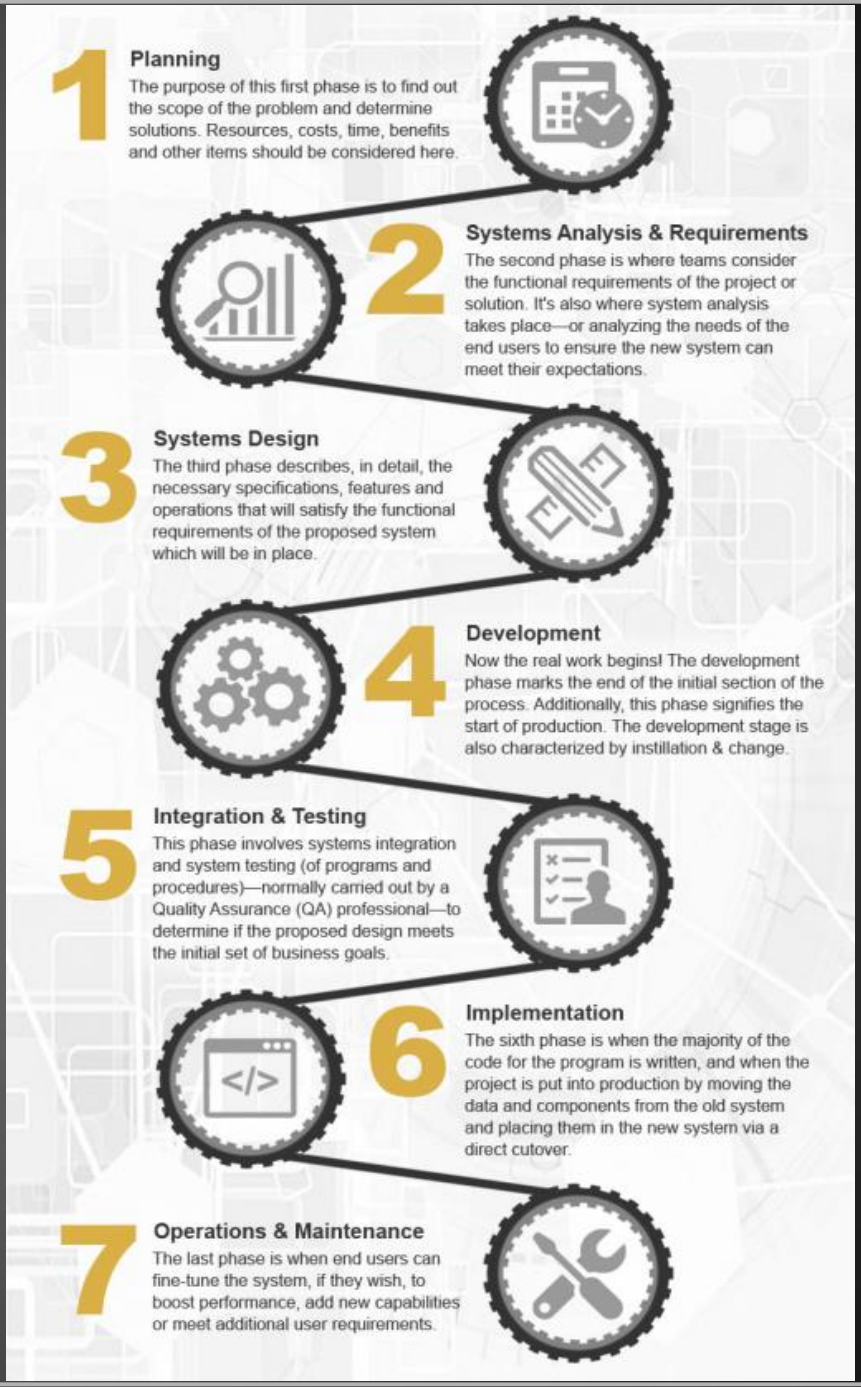
Etapele dezvoltării unei aplicații software

- La crearea unui produs software există mai multe faze importante, de exemplu:
  - analiza (analiza cerinţelor clientului);
  - proiectarea (descompunerea în module cât mai mici a produsului pentru uşurinţa implementării);
  - implementarea (scrierea efectivă a codului);
  - testarea:
  - întreţinerea.

Etapele dezvoltării unei aplicații software



Etapele dezvoltării unei aplicații software



Etapele dezvoltării unei aplicații software

- În cadrul lucrului la un proiect software sunt implicate mai multe persoane, fiecare are atribuții complementare celorlalți:
  - **utilizatori** - exploatatorii softului;
  - **șefi de proiect** - întotdeauna trebuie să existe;
  - **analiști de sistem** - execută modelul informatic al aplicației;
  - **proiectanți de sistem** - execută arhitectura programului aplicației;
  - **programatori** - execută modulele;
  - **ingineri de test** - verifică ceea ce s-a creat.

Unified Modeling Language

(UML)

Unified Modeling Language (UML)

- **UML** acronim pentru **Unified Modeling Language**
- este un limbaj vizual de modelare utilizat pentru specificarea, construirea şi documentarea sistemelor de aplicaţii orientate obiect şi nu numai.
- surprinde întreg ciclul de viaţă al dezvoltării software-ului
- datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat şi dincolo de domeniul IT
  - există aplicații ale UML-ului pentru management de proiecte, pentru *business Process Design* etc.

Unified Modeling Language (UML)

- Limbajul UML este un limbaj grafic creat pentru descrierea, specificarea şi documentarea funcţiilor unei aplicaţii în timpul fazei de proiectare (design).
- ajută la prezentarea posibilelor erori din structurile aplicațiilor, comportamentele sistemelor și alte procese de afaceri.

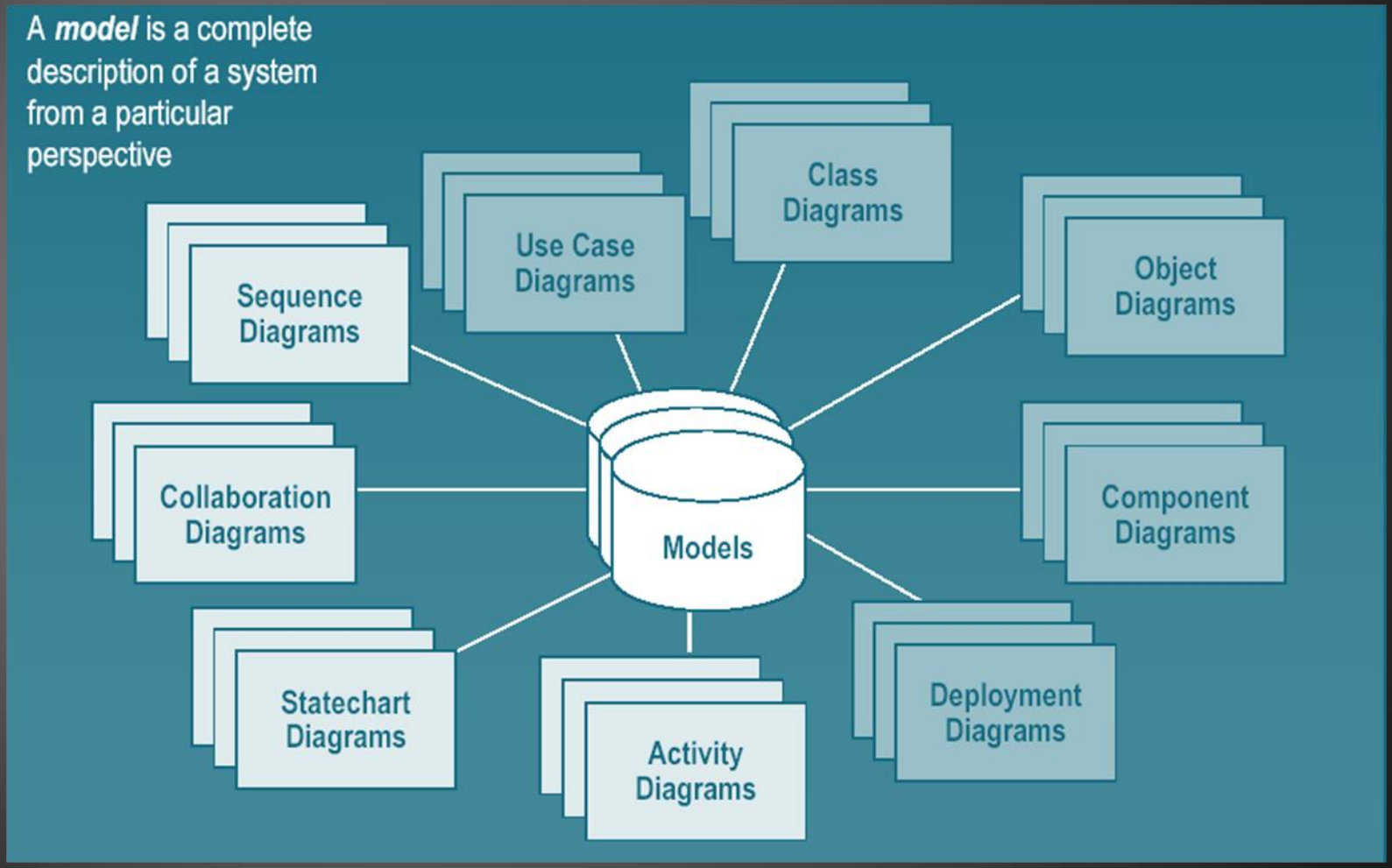
Unified Modeling Language (UML)

- **Model:** descrierea sau specificarea unui sistem.
  - Un model este de obicei prezentat ca o combinaţie de desene şi text.
- **Utilizarea modelului:** cei responsabili pentru succesul sau eşecul unui proiect se pot asigura încă dinainte de începerea scrierii codului că:
  - funcţionalitatea proiectului va fi corectă şi completă
  - toate necesităţile utilizatorului final vor fi satisfăcute
  - designul programului va satisface cerinţele de scalabilitate, robusteţe, extendabilitate etc.

Beneficiile UML

- Simplifică complexitățile
- Menține deschise canalele de comunicare
- Automatizează producția de software și procesele
- Contribuie la rezolvarea problemelor arhitecturale persistente
- Sporește calitatea muncii
- Reduce costurile și timpul de punere în vînzare

Diagramele UML



Tipuri de diagrame UML

- **Analiză**
  - Diagrama cazurilor de utilizare (Use Case Diagram)
  - Diagrama de activităţi (Activity Diagram)
- **Proiectare**
  - Structura: Diagrama de clase (Class Diagram)
  - Comportamentul:
    - Diagrama de stări (Statechart Diagram)
    - Diagrama de obiecte (Object Diagram)
    - Diagrama de secvenţă (Sequence Diagram)
    - Diagrama de colaborare (Collaboration Diagram)
- **Implementare**
    - Diagrama de componente (Component Diagram)
    - Diagrama de desfăşurare (Deployment Diagram)

Diagramele UML

- **Diagrama cazurilor de utilizare:** un caz de utilizare este o descriere a unei funcţionalităţi (o utilizare specifică a sistemului) pe care o oferă sistemul. Diagrama prezintă actorii externi şi cazurile de utilizare identificate, precum şi conexiunile identificate între actori şi cazurile de utilizare.
- **Diagrama de activităţi:** prezintă algoritmi sau fluxuri de date.
- **Diagrama de clase:** descrie clasele, pachetele, proprietăţile lor şi diferitele relaţii care pot apărea între acestea.
- **Diagrama de stări:** prezintă stările pe care le pot avea obiectele în cadrul unui sistem.
- **Diagrama de obiecte:** descrie un anumit scenariu;

Diagramele UML

- **Diagrama de secvenţă şi colaborare:** descrie secvenţa de mesaje schimbate de obiecte în timpul execuţiei.
- **Diagrama de componente:** descrie componentele implementate (Ex: fişierele sursă şi fişierele obiect).
- **Diagrama de desfăşurare:** descrie felul cum componentele sunt repartizate pe diferitele computere.