



Fundamente de Inginerie Software

Cap. 1

Introducere în Inginerie Software

Conf.Dr.Ing. Dan Pescaru

Textbooks: Sommerville “Software Engineering 7”, 2004, Cap.1

Maciaszek “Practical Software Engineering”, 2005, Cap. 1

Sursă: <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE7/>

<http://www.comp.mq.edu.au/books/pse/>

2009



Introducere

“Software engineering is the field of computer science that deals with the building of software systems that are so large or so complex that they are built by a team or teams of engineers.”

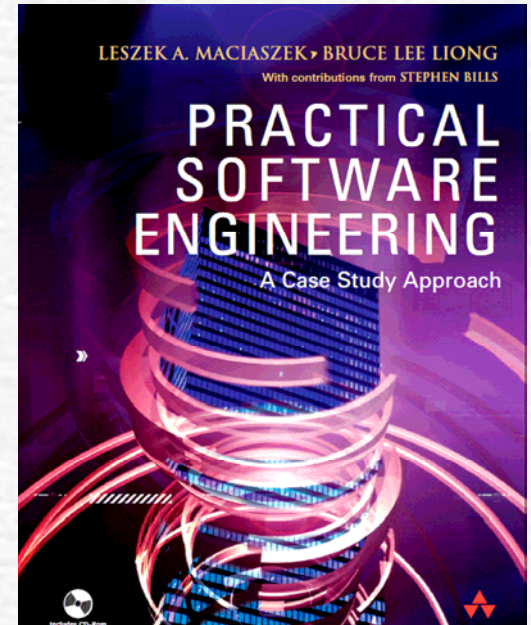
Ghezzi et al. '2003

- ☛ Ingineria Software se ocupă de dezvoltarea (și mentenanța) Sistemelor Software Complexe
- ☛ The Chaos Report 2003: doar **34%** din proiectele software au fost terminate la timp și în bugetul alocat

Inginerie Software

Fațete ale Ingineriei Software (Maciaszek '2005):

- Modele și faze de dezvoltare
- Limbaje de modelare (ex. UML)
- Unelte de inginerie software
- Planificare (ex. timp, buget etc.)
- Management
(ex. personal, risc etc.)



Inginerie Software

• Întrebări fundamentale despre IS (Sommerville '2004):

• Ce este IS?

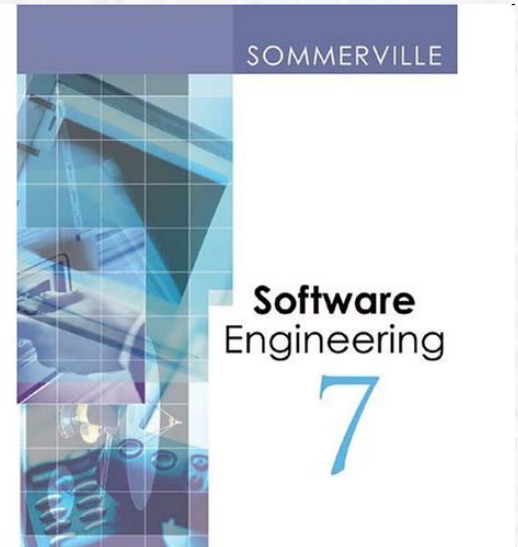
“Ingineria Software este o disciplină inginerească care se ocupă cu toate aspectele producției software”

• Care este deosebirea între IS și CS?

“CS se ocupă cu teoria iar IS se ocupă cu practica dezvoltării aplicațiilor software utile (in industrie)”

• Care sunt problemele presante în IS?

“Stăpânirea diversității, reducerea timpului de livrare și dezvoltarea unor aplicații software de încredere”



Dezvoltare Sistemelor Software

☞ Fazele de dezvoltare ale unui sistem software:

- Analiza cerințelor
- Proiectarea sistemului
- Implementarea
- Instalarea și integrarea
- Operarea și mentenanța

☞ Faze independente, aplicate pe toate nivelele:

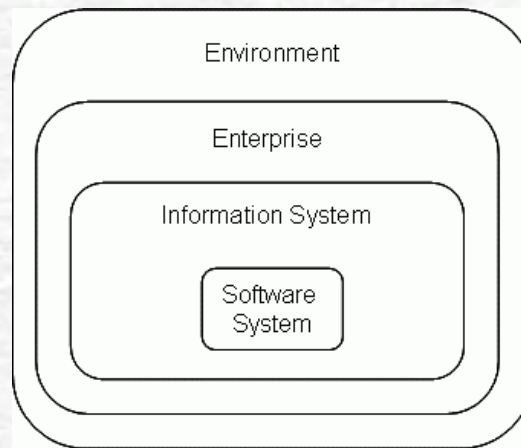
- Testarea și validarea
- Managementul

Observații

Observații esențiale referitoare la ingineria software:

- Sisteme software sunt incluse in sisteme informatice ale companiilor (fără să se substituie acestora)
- Software-ul face parte și este subordonat afacerii
- Ingineria software este diferită de alte ramuri ingineresti
- Ingineria software înseamnă mai mult decât programare
- Ingineria software se sprijină pe modelare
- Sistemele software sunt sisteme complexe

Sisteme software sunt incluse in sisteme informatice ale companiilor (fără să se substituie acestora)

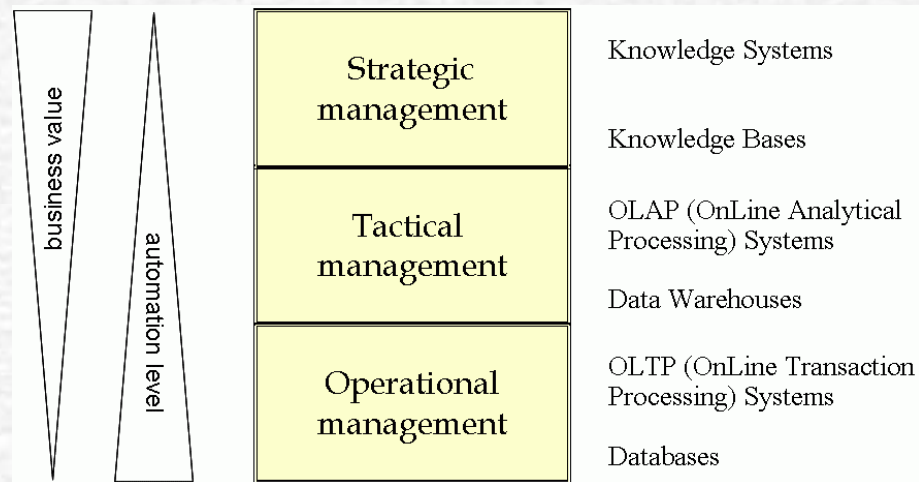


Sistemul Informatic

- Generează și gestionează informații pentru personal
- Conține: personal, informații, proceduri, software, echipamente hardware și căi de comunicații

Software-ul face parte și este subordonat afacerii

- Scopul software-ului este acela de a asigura eficiența afacerii
- Legătura dintre nivelele de management și sistemul software într-o companie:



Ingineria software este diferită de alte ramuri ingineresti tradiționale (1)

- ☞ Software-ul este imaterial
 - Modelele matematice clasice pot fi aplicate doar unor anumite aspecte ale lui
 - Software-ul este definit în termeni fuzzy – “bun”, “rău”, “acceptabil”, “satisfăcător conform cerințelor”, etc. (asemănător cu sectorul serviciilor)
- ☞ Ingineria software, în ciuda terminologiei fuzzy, nu trebuie să fie mai puțin riguroasă sau imposibil de demonstrat
- ☞ Software trebuie să fie ușor de înțeles, ușor de întreținut și scalabil

Ingineria software este diferită de alte ramuri inginerești tradiționale (2)

- Diferența principală față de ramurile tradiționale: cerințele se pot modifica mult mai drastic (ex. nu se cere mutarea unui pod cu 10 m mai încolo după ce a fost deja construit)
- Aplicațiile software nu sunt fabricate ci implementate. Cele destinate afacerilor vor fi mulate pe situația existentă în realitate
- Inginerii software trebuie să cunoască în general destul de bine domeniile pentru care implementează aplicații

Ingineria software înseamnă mai mult decât programare

- ☞ Ingineria software se aplică problemelor complexe ce nu pot fi rezolvate doar prin programare
 - Sistemele complexe trebuie proiectate înainte de a se trece la programare
- ☞ Înainte de proiectare trebuie înțelese clar cerințele
- ☞ Inginerii software trebuie să integreze componentele pe care programatorii le implementează
- ☞ Ingineria software este o disciplină de echipă
 - Echipele trebuie gestionate

Ingineria software se sprijină pe modelare (1)

- ☞ Modelele sunt abstracții ale realității
- ☞ Prin abstractizare se reduce complexitate, permițând concentrarea asupra celor mai importante aspecte
- ☞ Abstractizarea se aplică și asupra produselor și a proceselor software
 - "Software process model" este o reprezentare abstractă a unui proces software
 - "Software product model" este o reprezentare abstractă a unui produs software

Ingineria software se sprijină pe modelare (2)

- Există modelele pe diferite nivele de-a lungul ciclului de viață a unui produs software:
 - Model pentru cerințe – relativ informal
 - Model pentru specificații – de obicei formal (UML)
 - Modelul arhitectural – arhitectura sistemului
 - Model detaliat de proiectare – detaliază soluțiile software/hardware
 - Model de programare – model folosit la implementare
- Modelarea este influențată de paradigmele utilizate:
 - Paradigma funcțională se bazează pe decompoziție funcțională
 - Paradigma orientată pe obiecte împarte sistemul în pachete și componente legate prin diverse relații

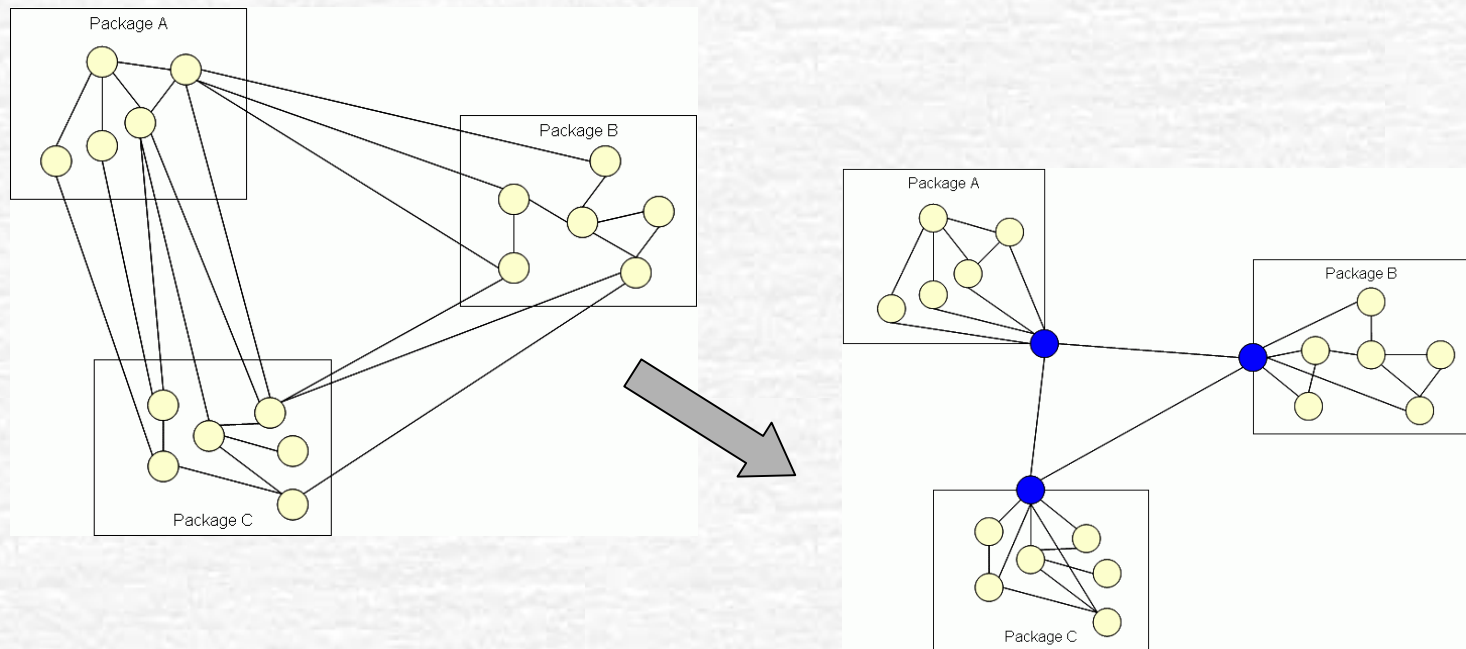
Sistemele software sunt sisteme complexe (1)

- ☞ Pentru software monolitic: complexitate = dimensiune
- ☞ La sistemele din componente (distribuite) complexitatea stă în legăturile și comunicațiile dintre componente
 - "Cost of glue code is three times cost of application code" (Endres, Rombach, 2003)
- ☞ Măsurare: Cumulative Class Dependency (CCD):

$$CCD = \frac{n!}{2!(n-2)!}$$

Sistemele software sunt sisteme complexe (2)

- Structurile ierarhice reduc complexitatea (interfețe între pachete – DP Facade)



Sistemele software sunt sisteme complexe. Observații.

- ☞ O structură este stabilă dacă coeziunea este puternică și cuplajul slab (Larry Constantine)
 - Coeziunea – comunicarea în interiorul modului
 - Cuplajul – interacțiunea între module
 - ☞ Doar ce este ascuns poate fi schimbat fără riscuri (David Parnas)
 - ☞ Separarea preocupărilor conduce la arhitecturi standard (Ernst Denert)
 - ☞ Un sistem care evoluează își mărește complexitatea dacă nu se depune efort pentru menținerea ei sub control (Meir Lehman)
-

Concluzii

- ❧ Domeniul Ingineriei Software este dezvoltarea sistemelor software de mari dimensiuni
- ❧ Sistemul Software este parte a afacerii
- ❧ Natura imaterială și schimbările frecvente sunt factorii care diferențiază Ingineria Software de ramurile ingineresti tradiționale
- ❧ Ingineria Software înseamnă mai mult decât programare.
- ❧ Ingineria Software este în mare parte legată de modelare