



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Managementul Proiectelor Software

Metodologii de estimare

Metodologii de estimare

- De sus in jos (Top-down)
- De jos in sus (Bottom-up)
- Analogie
- Judecata expertului
- Pretul castigator
- Metode parametrice sau algoritmice
 - Formule si ecuatii matematice

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



Estimari “top-down”



- Caracteristicile principale ale proiectului
- Avantaje
 - Usor de calculat
 - Foarte eficiente la inceput (ex. Estimarea initiala a costurilor)
- Dezavantaje
 - Modele discutabile
 - Risc de potrivire redusă
 - Precizie redusă – nu iau in calcul detaliile proiectului
- Analogia, Judecata Expertului si Metodele algoritmice pot fi de tipul top-down



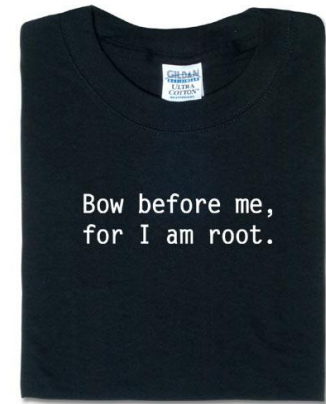
Estimari “bottom-up”



- Genereaza WBSs (Work-Breakdown Structures)
- Avantaje
 - Foarte eficiente in cazul activitatilor care sunt foarte bine intelese
- Dezavantaje
 - Anumite activitati nu sunt cunoscute intotdeauna
 - Consumatoare de timp



Judecata expertului



- Parerea unei persoane cu experienta recenta cu un proiect similar
- Estimare subiectiva
 - Depinde de expertiza reala a persoanei in cauza
- Expertul trebuie sa fi lucrat la un proiect similar
- Alternativ, medie ponderata a mai multor pareri



Analogii

- Informatiile referitoare la un proiect anterior
 - Proiectul suficient de similar din punct de vedere al tehnologiei, tipului si organizarii
 - Atribute ce pot fi comparate (ex. numar de intrari/iesiri)
 - Functii matematice de comparatie
- Avantaje
 - Se bazeaza pe date istorice
- Dezavantaje
 - Dificil de găsit proiecte similare
 - Datele privind proiectele anterioare pot fi eronate
 - Complicat de măsurat diferentele dintre doua proiecte



Pretul castigator

- Se urmaresc alte estimari
- Metoda mult mai rapida decat o estimare completa
- Necesita informatii referitoare la alte estimari
- Cumparatorul trebuie sa analizeze cu atentie trade-off-urile



Masuratori algoritmice

- Linii de cod (Lines Of Code)
- Puncte de functie (Function Points)
- Puncte de caracteristici (Feature Points)
- Altele
 - Numarul de baloane dintr-un DFD
 - Numarul de entitati ERD
 - Numarul de procese de pe o diagrama de structura
- Folosite rar



Linii de cod (KLoC)

Every program attempts to expand until it can read [mail](#). Those programs which cannot so expand are replaced by ones which can.

- **Avantaje**
 - O metrica comuna si foarte usor de inteles
 - Permite comparatii specifice
 - Usor de masurat
- **Dezavantaje**
 - Greu de estimat la inceputul proiectului
 - Variație in functie de limbajul de programare folosit
 - Nu ia in considerare multe costuri (ex: analiza cerintelor)
 - Generare automata de cod - cod in exces



Function Points

- Tehnica mai metodică decât Linii de Cod
- Analogia cu o casă
 - Aria Casei \sim Linii de cod
 - Ia în considerare dimensiunea
 - Numarul de dormitoare și bai \sim Function points
 - Ia în considerare atât dimensiunea, cât și funcția



Function Points (2)

- 1. Se numara toate functiile business din fiecare categorie
 - Categori: intrari, iesiri, cereri in baza de date, fisiere sau structuri de date, interfete
- 2. Se stabileste un factor de complexitate pentru fiecare
 - Simplu, Mediu, Complex
- 3. Se calculeaza si se aplica un “multiplicator de influeta”
 - Valoare intre 0.65 si 1.35
 - Se calculeaza in functie de 14 factori diferiti
- 4. Rezultatul este un “total de puncte de functie”
 - Folosit pentru estimari comparative



Metoda “Wideband Delphi”

- Consensul de grup
- Problema prezentată unui grup de experți, împreună cu un formular de răspuns
- Discuții în grup, colectare de opinii anonime, feedback
- Continuarea discuțiilor până la consens
- Avantaje
 - Usor, ieftin, se folosește expertiza mai multor persoane
 - Nu necesită date sau informații istorice
- Dezavantaje
 - Greu de repetat
 - Uneori nu se poate ajunge la un consens, sau se ajunge la un rezultat complet eronat



Estimarile si reutilizarea codului

- Tipuri de cod
 - Nou – modificare mai mult de 50%
 - Modificat
 - Reutilizat
- Factorii de reutilizare a codului
 - Durată reutilizare / durată scriere cod nou \approx 30%
 - Durată modificare / durată scriere cod nou \approx 60%
- Integrare cod refolosit / integrare cod nou \approx 100%
- Anti-patterns
 - Not Invented Here
 - Reinventing the Wheel



Estimarea efortului

- Modele:
 - Empirice
 - Matematice
 - Subiective
- Se exprima in unitati de durata
 - Oameni-luna (sau “personal-luna”)



Estimarea efortului (2)

- Tabele de orar
 - Convertirea estimarilor de dimensiune in estimari de effort
 - Folosesc date istorice
- Combinare între estimarile de dimensiune și de effort
- Programarea bazată pe angajament
 - Un dezvoltator isi ia un angajament pe baza unei estimari proprii

