Università degli Studi di Udine

Analisi Costi

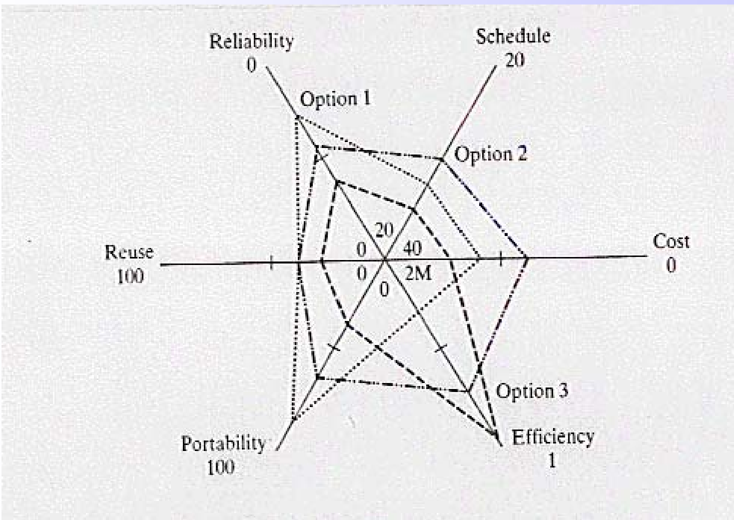
prof. Maurizio Pighin

Dipartimento di Matematica e Informatica

Modelli decisionali

- Il Problema decisionale
  - *Insieme di opzioni e strade diverse*
- Passi del processo decisionale
  - *Definizione target di progetto*
    - ad es. manutenibilità, bassi costi
    - usualmente gli obiettivi sono contraddittori
  - *Applicazione teoria delle decisioni*

Modelli Decisionali: Diagrammi Polari



Modelli Decisionali

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- *Metodo di Bohem (payoff matrix / break-even)*
  - soluzione 1 - MAXMIN - minimizza le perdite - Opzione B
  - soluzione 2 - MAXIMAX - massimizza i ricavi - Opzione A
  - soluzione 3 - punto di payoff - introduzione di probabilità
    - *se in A e B le due colonne sono equiprobabili si sceglie B*
      - » *A=0.5*100+0.5*40 =70*
      - » *B=0.5*70+0.5*55=62.5*
    - *con pesi (A=35,65 B=42-58) si sceglie A*
      - » *A=0.35*100+0.65*40=61*
      - » *B=0.42*70+0.58*55=61.3*

Scelta	Cost MAX	Cost EXP
A	100	40
B	70	55

Modelli Decisionali

- *Massimizzazione rapporto Prestazioni/Costo*
- *Massimizzazione differenza Prestazioni/Costo*
- *Modello del valore atteso*
  - n caratteristiche
  - w_i pesi ($0 < w_i <= 1$) (a seconda dell'impatto)
  - s_i risultati ($0 < s_i <= 10$) (a seconda della strategia)
  - Si cerca il massimo di G

$$G = \sum_{1}^{n} W_i S_i$$

Modelli Decisionali

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- *Modello Congiuntivo*
  - Vengono fissate X_i soglie (una per ogni parametro) e vengono scartate le alternative che non raggiungono la soglia in UN valore.
    Poi si applica il modello del valore atteso
- *Modello Disgiuntivo*
  - Vengono fissate X_i soglie (una per ogni parametro) e vengono scartate le alternative che non raggiungono la soglia in TUTTI i valori.
    Poi si applica il modello del valore atteso

Modelli Decisionali

– *Modello con stima dell'impatto*

Problemi	Soluz. 1	Soluz. 2	Soluz.3	Totale
A	40%	30%	-10%	60%
B	30%	90%	40%	160%
Totale	70%	120%	30%	

- Nessuna soluzione è in grado da sola di risolvere alcun problema
- La sol.2 in combinazione con anche una sola delle altre è in grado di risolvere il problema 2 (Somma >100)
- Il primo problema non è solubile neanche con le tre soluzioni messe insieme (Somma <100)
- La sol.2 è la migliore (Somma per colonne maggiore)

OSSERVAZIONI
- Stima può essere poco affidabile se la quantificazione è semplificata
- L'effetto complessivo di azioni non sempre è la somma degli effetti

Software cost estimation

- Predicting the resources required for a software development process
- Problems
  - *Local data definition*
  - *Calibration (calibration process)*
  - *Indipendent estimation group*
  - *Reduce input subjectivity*
  - *Preliminary estimation and re-estimation*
    - Wideband Delphi technique (stime anonime non conoscendo e conoscendo stime altrui)
    - Modifiche con scarti estremi e valutazioni medie
  - *Alternative size measure*
  - *Locally developed cost models*

Fundamental estimation questions

- How much effort is required to complete an activity?
- How much calendar time is needed to complete an activity?
- What is the total cost of an activity?
- What is the total project estimation?

Costing and pricing

- Estimates are made to discover the cost, to the developer, of producing a software system
- There is not a simple relationship between the development cost and the price charged to the customer
- Broader organisational, economic, political and business considerations influence the price charged

Software pricing factors

Factor	Description
Market opportunity	A development organisation may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the opportunity of more profit later. The experience gained may allow new products to be developed.
Cost estimate uncertainty	If an organisation is unsure of its cost estimate, it may increase its price by some contingency over and above its normal profit.
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Requirements volatility	If the requirements are likely to change, an organisation may lower its price to win a contract. After the contract is awarded, high prices may be charged for changes to the requirements.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a small profit or break even than to go out of business.

Programmer productivity

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- A measure of the rate at which individual engineers involved in software development produce software and associated documentation
- Essentially, we want to measure useful functionality produced per time unit

Factors affecting productivity

Factor	Description
Application domain experience	Knowledge of the application domain is essential for effective software development. Engineers who already understand a domain are likely to be the most productive.
Process quality	The development process used can have a significant effect on productivity. This is covered in Chapter 31.
Project size	The larger a project, the more time required for team communications. Less time is available for development so individual productivity is reduced.
Technology support	Good support technology such as CASE tools, supportive configuration management systems, etc. can improve productivity.
Working environment	a quiet working environment with private work areas contributes to improved productivity.

Quality and productivity

- All metrics based only on volume/unit time are difficult to apply because they do not take quality into account
- Productivity may generally be increased at the cost of quality
- It is not clear how productivity/quality metrics are related
- If change is constant then an approach based on counting lines of code is not meaningful

Estimation techniques

- There is no simple way to make an accurate estimate of the effort required to develop a software system
  - *Initial estimates are based on inadequate information in a user requirements definition*
  - *The software may run on unfamiliar computers or use new technology*
  - *The people in the project may be unknown*
- Project cost estimates may be self-fulfilling
  - *The estimate defines the budget and the product is adjusted to meet the budget*

Modelli stima Costi - Richiami

- Estimation by analogy
- Expert judgement
- Parkinson's Law
- Pricing to win
- Top-down estimation
- Bottom-up estimation
- Algorithmic cost modelling (COCOMO)

Expert judgement

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached.
- Advantages: Relatively cheap estimation method. Can be accurate if experts have direct experience of similar systems
- Disadvantages: Very inaccurate if there are no experts!

Slide 17

Estimation by analogy

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- The cost of a project is computed by comparing the project to a similar project in the same application domain
- Advantages: Accurate if project data available
- Disadvantages: Impossible if no comparable project has been tackled. Needs systematically maintained cost database

Slide 18

Parkinson's Law

- The project costs whatever resources are available
- Advantages: No overspend
- Disadvantages: System is usually unfinished

Pricing to win

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- The project costs whatever the customer has to spend on it
- Advantages: You get the contract
- Disadvantages: The probability that the customer gets the system he or she wants is small. Costs do not accurately reflect the work required

Top-down and bottom-up estimation

- Any of these approaches may be used top-down or bottom-up
- Top-down
  - *Start at the system level and assess the overall system functionality and how this is delivered through sub-systems*
- Bottom-up
  - *Start at the component level and estimate the effort required for each component. Add these efforts to reach a final estimate*

Top-down estimation

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- Usable without knowledge of the system architecture and the components that might be part of the system
- Takes into account costs such as integration, configuration management and documentation
- Can underestimate the cost of solving difficult low-level technical problems

Bottom-up estimation

- Usable when the architecture of the system is known and components identified
- Accurate method if the system has been designed in detail
- May underestimate costs of system level activities such as integration and documentation

Estimation methods

- Each method has strengths and weaknesses
- Estimation should be based on several methods
- If these do not return approximately the same result, there is insufficient information available
- Some action should be taken to find out more in order to make more accurate estimates
- Pricing to win is sometimes the only applicable method

Experience-based estimates

- Estimating is primarily experience-based
- However, new methods and technologies may make estimating based on experience inaccurate
  - *Object oriented rather than function-oriented development*
  - *Client-server systems rather than mainframe systems*
  - *On the shelf components (COTS)*
  - *Component-based software engineering*
  - *CASE tools and program generators*

Pricing to win

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- This approach may seem unethical and unbusinesslike
- However, when detailed information is lacking it may be the only appropriate strategy
- The project cost is agreed on the basis of an outline proposal and the development is constrained by that cost
- A detailed specification may be negotiated or an evolutionary approach used for system development

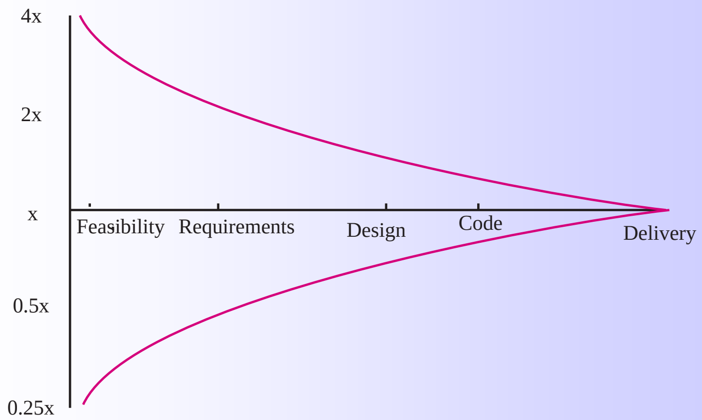
Algorithmic cost modelling

- Cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers
  - *Effort = A * Size^B * M*
    - A is an organisation-dependent constant,
    - Size in KDSI
    - B reflects the disproportionate effort for large projects
    - M is a multiplier reflecting product, process and people attributes
    - Effort is usually expressed in person-months
- Most commonly used product attribute for cost estimation is code size
- Most models are basically similar but with different values for A, B and M

Estimation accuracy

- The size of a software system can only be known accurately when it is finished
- Several factors influence the final size
  - *Use of COTS (components on the shelf)*
  - *Programming language*
  - *Distribution of system*
- As the development process progresses then the size estimate becomes more accurate

Estimate uncertainty



The COCOMO model

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- An empirical model based on project experience
- Well-documented, ‘independent’ model which is not tied to a specific software vendor
- Long history from initial version published in 1981 (COCOMO-81) through various instantiations to COCOMO 2
- COCOMO 2 takes into account different approaches to software development, reuse, etc.

COCOMO 81

Project complexity	Formula	Description
Simple	PM = 2.4 $(KDSI)^{1.05}$ M	Well-understood applications developed by small teams.
Moderate	PM = 3.0 $(KDSI)^{1.12}$ M	More complex projects where team members may have limited experience of related systems.
Embedded	PM = 3.6 $(KDSI)^{1.20}$ M	Complex projects where the software is part of a strongly coupled complex of hardware, software, regulations and operational procedures.

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

COCOMO 81

- Development Time
  - *Simple: TDEV = 2.5 (PM)* $^{0.38}$
  - *Moderate: TDEV = 2.5 (PM)* $^{0.35}$
  - *Embedded : TDEV = 2.5 (PM)* $^{0.32}$
- Personnel requirement: N = PM/TDEV
- Attributi correttivi

COCOMO 2 levels

- COCOMO 2 is a 3 level model that allows increasingly detailed estimates to be prepared as development progresses
- Early prototyping level
  - *Estimates based on object points and a simple formula is used for effort estimation*
- Early design level
  - *Estimates based on function points that are then translated to LOC*
- Post-architecture level
  - *Estimates based on lines of source code*

Early prototyping level

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- Supports prototyping projects and projects where there is extensive reuse
- Based on standard estimates of developer productivity in object points/month
- Takes CASE tool use into account
- Formula is
  - *PM = (NOP * (1 - %reuse/100)) / PROD*
    - PM is the effort in person-months,
    - NOP is the number of object points
    - PROD is the productivity

Early prototyping level

- Object Point Productivity

Developer's experience and capability	Very low	Low	Nominal	High	Very high
CASE maturity and capability	Very low	Low	Nominal	High	Very high
PROD (NOP/month)	4	7	13	25	50

Early design level

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- Estimates can be made after the requirements have been agreed
- Based on standard formula for algorithmic models
  - $PM = A \times Size^{B} \times M + PM_{m}$ *where*
    - A = 2.5 in initial calibration
    - Size in KLOC,
    - B varies from 1.1 to 1.24 depending on novelty of the project, development flexibility, risk management approaches and the process maturity
    - M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED

Multipliers

- Multipliers reflect the capability of the developers, the non-functional requirements, the familiarity with the development platform, etc.
  - *RCPX - product reliability and complexity*
  - *RUSE - the reuse required*
  - *PDIF - platform difficulty*
  - *PREX - personnel experience*
  - *PERS - personnel capability*
  - *SCED - required schedule*
  - *FCIL - the team support facilities*

Multipliers

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

- PM_m reflects the amount of automatically generated code and is expressed by the formula
  $PM_m = (ASLOC \times (AT/100)) / ATPROD$
  - *ASLOC = Automatic generated lines of source code*
  - *ATPROD = Productivity level for this type of code*
  - *AT = Percentage of total system automatically generated*

Post-architecture level

- Uses same formula as early design estimates
- Estimate of size is adjusted to take into account
  - *Requirements volatility. Rework required to support change*
  - *Extent of possible reuse. Reuse is non-linear and has associated costs so this is not a simple reduction in LOC*
  - *ESLOC = ASLOC $\times$ (AA + SU +0.4DM + 0.3CM +0.3IM)/100*
    - ESLOC is Equivalent number of lines of new code.
    - ASLOC is the number of lines of reusable code which must be modified
    - AA is a factor which reflects the initial assessment costs of deciding if software may be reused.
    - SU is a factor based on the cost of software understanding
    - DM is the percentage of design modified
    - CM is the percentage of the code that is modified
    - IM is the percentage of the original integration effort required for integrating the reused software.

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

The exponent term

- This depends on 5 scale factors (see next slide). Their sum/100 is added to 1.01
- Example
  - *Precedenteness - new project - 4*
  - *Development flexibility - no client involvement - Very high - 1*
  - *Architecture/risk resolution - No risk analysis - V. Low - 5*
  - *Team cohesion - new team - nominal - 3*
  - *Process maturity - some control - nominal - 3*
- Scale factor is therefore 1.17

Exponent scale factors

Precedentedness	Reflects the previous experience of the organisation with this type of project. Very low means no previous experience, Extra high means that the organisation is completely familiar with this application domain.
Development flexibility	Reflects the degree of flexibility in the development process. Very low means a prescribed process is used; Extra high means that the client only sets general goals
Architecture/risk resolution	Reflects the extent of risk analysis carried out. Very low means little analysis, Extra high means a complete a thorough risk analysis.
Team cohesion	Reflects how well the development team know each other and work together. Very low means very difficult interactions, Extra high means an integrated and effective team with no communication problems.
Process maturity	Reflects the process maturity of the organisation. The computation of this value depends on the CMM Maturity Questionnaire but an estimate can be achieved by subtracting the CMM process maturity level from 5.

Ingegneria del software
Progettazione e Laboratorio
Costi
Maurizio Pighin

Exponent scale factors

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	throughly familiar
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
RESL[a]	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
PMAT	Weighted average of "Yes" answers to CMM Maturity Questionnaire					

Multipliers

- Product attributes
  - *Concerned with required characteristics of the software product being developed*
- Computer attributes
  - *Constraints imposed on the software by the hardware platform*
- Personnel attributes
  - *Multipliers that take the experience and capabilities of the people working on the project into account.*
- Project attributes
  - *Concerned with the particular characteristics of the software development project*

Project cost drivers

Product attributes			
RELY	**Required system reliability**	**DATA**	**Size of database used**
CPLX	**Complexity of system modules**	**RUSE**	**Required percentage of reusable components**
DOCU	**Extent of documentation required**		
Computer attributes			
TIME	**Execution time constraints**	**STOR**	**Memory constraints**
PVOL	**Volatility of development platform**		
Personnel attributes			
ACAP	**Capability of project analysts**	**PCAP**	**Programmer capability**
PCON	**Personnel continuity**	**AEXP**	**Analyst experience in project domain**
PEXP	**Programmer experience in project domain**	**LTEX**	**Language and tool experience**
Project attributes			
TOOL	**Use of software tools**	**SITE**	**Extent of multi-site working and quality of site communications**
SCED	**Development schedule compression**		

Effect of Cost Drivers

Exponent value	1.17
System size (including factors for reuse and requirements volatility)	128, 000 DSI
Initial COCOMO estimate without cost drivers	**730 person-months**
Reliability	Very high, multiplier = 1.39
Complexity	Very high, multiplier = 1.3
Memory constraint	High, multiplier = 1.21
Tool use	Low, multiplier = 1.12
Schedule	Accelerated, multiplier = 1.29
Adjusted COCOMO estimate	**2306 person-months**
Reliability	Very low, multiplier = 0.75
Complexity	Very low, multiplier = 0.75
Memory constraint	None, multiplier = 1
Tool use	Very high, multiplier = 0.72
Schedule	Normal, multiplier = 1
Adjusted COCOMO estimate	**295 person-months**

Sforzo, produttività, costi, tempi

- La stima è basata su parametri di produttività
  - *Esempio: P= 1000 LOC x Persona x Mese*
- Le attività non sono necessariamente produttive in KLOC
  - *Analisi, test, documentazione, coordinamento...*
- La stima si esprime in impegno per persona
  - *Esempio: MesiPersona*
- La stima si riferisce al solo lavoro implicato direttamente dal progetto
  - *Costo ≠ Stima x CostoMedioPersonaMese*
  - *Tempo ≠ Stima / NumeroPersoneDisponibili*

Dalla Stima al Costo

- Il costo delle persone impegnate in un progetto è dato da diversi fattori:
  - *Costi diretti del personale, comprensivi di oneri fiscali e previdenziali*
    - Varia in relazione ai profili coinvolti
    - Può essere determinato un costo medio delle persone impiegate direttamente nei progetti
  - *Produttività delle singole persone (60-80% del tempo)*
  - *Costi indiretti*
    - Struttura (affitti, macchinari, cespiti in generale....)
    - Servizi fruiti (manutenzioni, consumi, consulenze, ...)
    - Servizi erogati (segreteria, help desk, servizi amministrativi, ...)