Software project management

Cost and Effort Estimation

Dr. Juma Lungo

University of Dar-es-Salaam

Cost and Effort Estimation

Cost estimation objectives

Budget

To know what you will spend

Controls

A lever to control the project

Differential analysis

Monitor progress by comparing planned with estimated costs

Cost database

Make future estimation better

Marry costing to management

Cost estimation and planning/scheduling are closely related activities

Software cost components

Hardware and software costs

Travel and training costs

Effort costs (the dominant factor in most projects)

- **salaries of engineers involved in the project**
- **costs of building, heating, lighting**
- **costs of networking and communications**
- **costs of shared facilities (e.g library, staff restaurant, etc.)**
- **costs of pensions, health insurance, etc.**

Costing and pricing

Estimating Cost

- Costs for developer, not buyer
- We need our costs to manage and assess

Estimating Price

- 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.

Productivity Measures

Size-related measures

- **Must be based on some output from the software process**
- **Delivered source code**
- **Object code instructions**

Function-related measures

- **Based on an estimate of the functionality of the delivered software.**
- **Function-points are the best known of this type of measure**

Lines of Codes

LOC = NCLOC + CLOC

LOC: lines of code

NCLOC: non-commented line of code

CLOC: commented line of code

KLOC = one thousand of line of code

Function points

Based on a combination of program characteristics

- external inputs and outputs
- user interactions
- external interfaces
- files used by the system

A weight is associated with each of these

The function point count is computed by multiplying each raw count by the weight and summing all values

Function points

- Function point count modified by complexity of the project
- FPs can be used to estimate LOC depending on the average number of LOC per FP for a given language
- FPs are very subjective
  - Depend on the estimator
  - FP cannot generally be counted automatically

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.

Estimation techniques

Estimation techniques

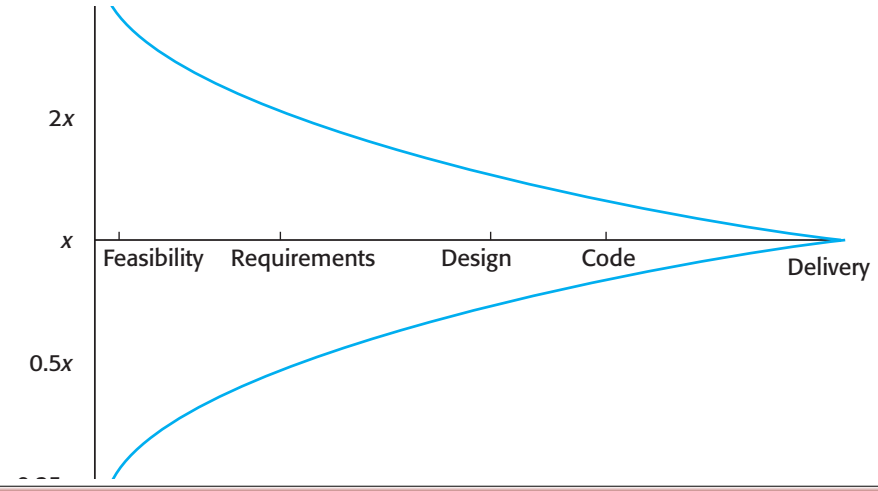
Organizations need to make software effort and cost estimates. There are two types of technique that can be used to do this:

Experience-based techniques The estimate of future effort requirements is based on the manager's experience of past projects and the application domain. Essentially, the manager makes an informed judgment of what the effort requirements are likely to be.

Estimation techniques

Algorithmic cost modeling In this approach, a formulaic approach is used to compute the project effort based on estimates of product attributes, such as size, and process characteristics, such as experience of staff involved.

Estimate uncertainty



Experience-based approaches

Experience-based techniques rely on judgments based on experience of past projects and the effort expended in these projects on software development activities.

Typically, you identify the deliverables to be produced in a project and the different software components or systems that are to be developed.

Experience-based approaches

You document these in a spreadsheet, estimate them individually and compute the total effort required.

It usually helps to get a group of people involved in the effort estimation and to ask each member of the group to explain their estimate.

Problem with experience-based approaches

The difficulty with experience-based techniques is that a new software project may not have much in common with previous projects.

Software development changes very quickly and a project will often use unfamiliar techniques such as web services, application system configuration

Problem with experience-based approaches

If you have not worked with these techniques, your previous experience may not help you to estimate the effort required, making it more difficult to produce accurate costs and schedule estimates.

Estimation techniques

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

Expert judgement

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: May be very costly

Estimation by analogy

The cost of a project is computed by comparing the project to a similar project inthe same application domain

Advantages: Accurate if project data available

Disadvantages: Impossible if no comparable project has been tackled. Needs systematically maintained cost database

Parkinson's Law

The project costs whatever resources are available

Advantages: No overspending

Disadvantages: System is usually unfinished

Pricing to win

- 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 estimation

- Approaches may be applied using a top-down approach. Start at system level andwork out how the system functionality is provided
- 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

- Start at the lowest system level. The cost of each component is estimated individually.These costs are summed to give final cost estimate
- 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

Algorithmic cost modelling

- Cost is estimated as a mathematical function of product, project and processattributes whose values are estimated by project managers
- The function is derived from a study of historical costing data
- Most commonly used product attribute for cost estimation is LOC (code size)
- Most models are basically similar but withdifferent attribute values

Examples of cost models

General form: $E = A + B \times S^C$

E: Effort cost; S: Size; A, B, C: constants

Examples:

$E = 5.2 \times (KLOC)^{0.91}$ Walston-Felix Model

$E = 5.5 + 0.73 \times (KLOC)^{1.16}$ Bailey-Basili Model

$E = 3.2 \times (KLOC)^{1.05}$ COCOMO Basic Model

$E = 5.288 \times (KLOC)^{1.047}$ Doty Model for KLOC > 9

Examples of cost models

Cost models using FP as a primary input include (Pressman, 1997):

$E = -12.39 + 0.0545\ FP$

Albrecht and Gaffney Model

$E = 60.62 \times 7.728 \times 10^{-8}\ FP^3$

Kemerer Model

$E = 585.7 + 15.12\ FP$

Matson, Barnett, and Mellichamp Model

Algorithmic cost modelling

Cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers:

$\text{Effort} = A \times \text{Size}^B \times M$

A is an organisation-dependent constant, B reflects the disproportionate effort for large projects and M is a multiplier reflecting product, process and people attributes.

The most commonly used product attribute for cost estimation is code size.

Most models are similar but they use 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 reused systems and components;
- Programming language;
- Distribution of system.

Estimation accuracy

As the development process progresses then the size estimate becomes more accurate.

The estimates of the factors contributing to B and M are subjective and vary according to the judgment of the estimator.

Effectiveness of algorithmic models

Algorithmic cost models are a systematic way to estimate the effort required to develop a system. However, these models are complex and difficult to use.

There are many attributes and considerable scope for uncertainty in estimating their values.

Effectiveness of algorithmic models

This complexity means that the practical application of algorithmic cost modeling has been limited to a relatively small number of large companies, mostly working in defense and aerospace systems engineering.

The end!

Q & A