COCOMO

Software Cost Estimating Model

Software project estimation

- Predicting the resources required for a software development process
- Process of estimation of various resources required for the completion of a software project.
- Software project estimation is composed of following steps:
    1. Size estimation
    2. Effort estimation
    3. Schedule estimation
    4. Cost estimation

Reasons of Poor / Inaccurate Estimation

- Imprecise requirements estimations
- Unknown / New project type
- Lack of sufficient information about past projects
- Estimation on the basis of available resources only
- Cost and time trade offs

Objectives of Estimation

- To introduce the fundamentals of software costing and pricing
- To describe three metrics for software productivity assessment
- To explain why different techniques should be used for software estimation
- To describe the COCOMO 2 algorithmic cost estimation model

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?
- Project estimation and scheduling are interleaved management activities

Software cost components

- Hardware costs
- Software costs
- Travel and training costs
- Effort costs (the dominant factor in most projects)
- Effort costs must take overheads into account
- Organization typically has a percentage of direct costs that are allocated to overheads

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 organizational, economic, political and business considerations influence the price charged

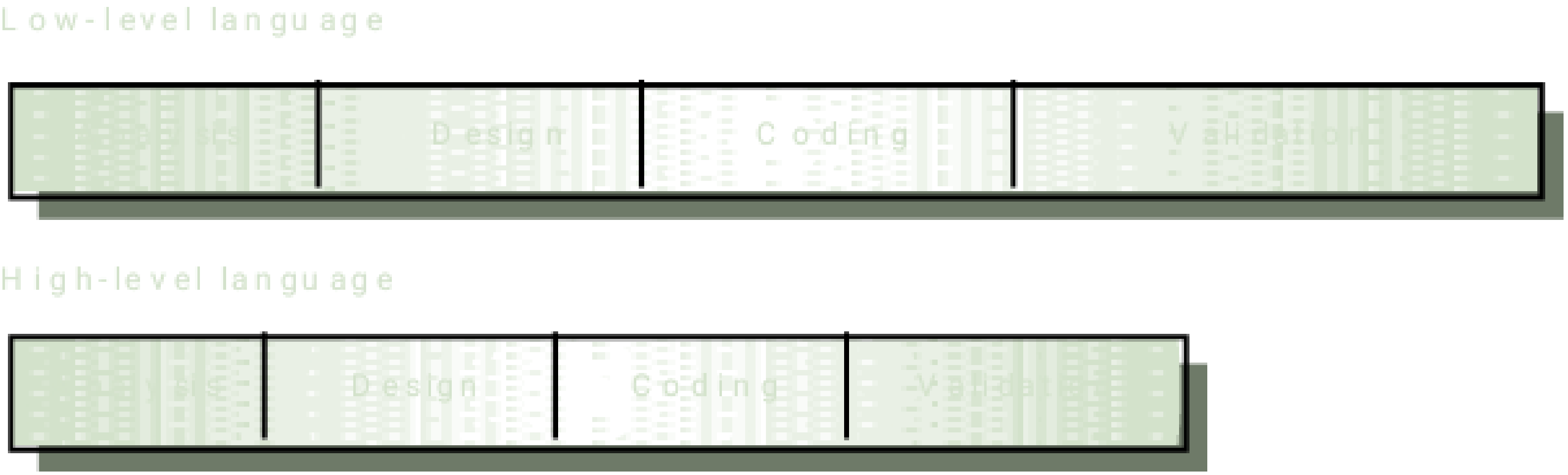
Lines of code

- Number of lines present in the sourcecode of the program
- What programs should be counted as part of the system?
- Assumes linear relationship between system size and volume of documentation

Impact of Languages on Development Times

	Analysis	Design	Coding	Testing	Documentation
Assembly code	3 weeks	5 weeks	8 weeks	10 weeks	2 weeks
High-level language	3 weeks	5 weeks	8 weeks	6 weeks	2 weeks

	Size	Effort	Productivity
Assembly code	5000 lines	28 weeks	714 lines/month
High-level language	1500 lines	20 weeks	300 lines/month



Empirical Estimation Models

- An empirical estimation model is based on project experiences
- Well-documented, ‘independent’ model which is not tied to a specific software vendor
- Predicts effort as a function of LOC or FP
- Major categories of such models are:

1. Static Single Variable Models – (Basic COCOMO – 81)

   Resource = $c_1 \times e \times c_2$

2. Static Multivariable Models – (Intermediate COCOMO – 81, Detailed COCOMO – 81)

   Resource = $c_1e_1 + c_2e_2 + \ldots\ldots c_ne_n$

3. Dynamic Multivariable Models – (Putnam Model)

   Resource = f(e,d)

   Here c = constant, e = estimated characteristic, d = dynamic characteristic

COCOMO Series of Models

COCOMO: The COCOMO (Constructive Cost Estimation Model) is proposed by **DR. Berry Boehm in 1981** and that's why the first COCOMO model is also known as COCOMO'81.

It is a method for evaluating the cost of a software package.

DEVELOPMENT MODES of COCOMO Models:

There are three modes of development:

Organic Mode:

1. Relatively Small, Simple Software projects.
2. Small teams with good application experience work to a set of less than rigid requirements.
3. Similar to previously developed projects.
4. Relatively small and require little innovation.

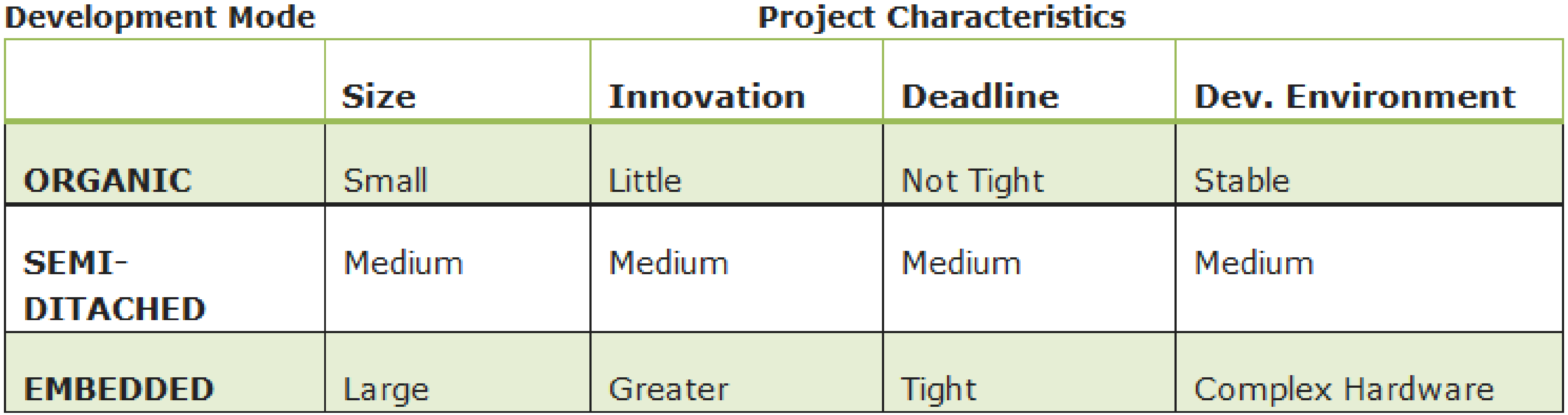
Semidetached Mode:

- Intermediate (in size and complexity) software projects in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

Embedded Mode:

1. Software projects that must be developed within set of tight hardware, software and operational Constraints

DEVELOPMENT MODES of COCOMO Models:



Development Mode	Project Characteristics			
	Size	**Innovation**	**Deadline**	**Dev. Environment**
ORGANIC	Small	Little	Not Tight	Stable
SEMI-DITACHED	Medium	Medium	Medium	Medium
EMBEDDED	Large	Greater	Tight	Complex Hardware

COCOMO Series of Models

CO (Constructive) CO (Cost) MO (Model)

COCOMO – I (COCOMO - 81)

Basic COCOMO – 81

Intermediate COCOMO – 81

Detailed COCOMO – 81

COCOMO - II

Application Composition Model

Early Design Model

Post Architecture Model

Basic COCOMO 81

Basic COCMO Model is good for quick, early, rough order of magnitude estimate of software cost.

It does not account for differences in hardware constraints, personal Quality and experience, use of modern tools and techniques, and other project attribute known to have a significant influence on software cost, which limits its accuracy.

It gives an approximate estimate of the project parameters.

The basic COCOMO estimation model is given by the following expressions:

$$\textbf{Effort (E)} = a \times (\text{KLOC})^{b}$$
$$\textbf{DevT (D)} = c \times (\text{Effort})^{d}$$

- KLOC is the estimated size of the software product expressed in Kilo Lines of Code,
- a, b, c, d are constants for each category of software products,
- Tdev is the estimated time to develop the software, expressed in months,
- Effort is the total effort required to develop the software product, expressed in person months (PMs).

Basic COCOMO 81

DEVELOPMENT MODES Values of Basic COCOMO-81 (for example):

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Basic COCOMO 81

Estimation of development effort (E):

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic: **Effort = 2.4(KLOC)$^{1.05}$ PM**
Semi-Detached: **Effort = 3.0(KLOC)$^{1.12}$ PM**
Embedded: **Effort = 3.6(KLOC)$^{1.20}$ PM**

Estimation of development time:

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic: **Tdev = 2.5(Effort)$^{0.38}$ Months**
Semi-detached: **Tdev = 2.5(Effort)$^{0.35}$ Months**
Embedded: **Tdev = 2.5(Effort)$^{0.32}$ Months**

Intermediate COCOMO 81

1.The intermediate COCOMO model takes the basic COCOMO model as its starting point which means that it is slightly different coefficients for the effort equation than the Basic model.

2.After that it multiplies the basic estimate by an Effort Adjustment Factor which is calculated as the product of 15 multipliers (cost drivers) which take into account factors such as required product reliability, database size, execution and storage constraints, personnel aptitude, and the use of software tools.

3.It produces better results than the Basic model because the user supplies settings for cost drivers that determine the effort and duration of the software projects.

4.The Intermediate model also allows the system to be divided and estimated in components. DSI values and cost drivers can be chosen for individual components instead of for the system as a whole.

Intermediate COCOMO 81

Effort Adjustment factor (EAF)

The effort adjustment factor use 15 cost drivers (EAF = C_1 x C_2 x C_3 x . . . x C_{15}) that are grouped into four categories:

1.Product
2.Computer
3.Personnel
4.Project

Note 1: Each cost driver has been rated on a 6-point ordinal scale ranging from low to high importance. Product of all effort multipliers leads to EAF.

Note 2: For nominal ratings, the value of cost driver will be = 1

Later this EAF is multiplied with initial effort ($E_{i,}$ derived from Basic COCOMO-81) to compute refined effort and development time.

Intermediate COCOMO 81

Steps

1. Compute EAF
2. Compute initial effort (E_i) = a x $(KLOC)^b$ PM
3. Compute Final effort and Development time estimates by formula

$$\textbf{Effort (E)} = \textbf{EAF} \times E_i$$
$$\textbf{DevT (D)} = c \times (\textbf{Effort})^d$$

Revised DEVELOPMENT MODES Values of Intermediate COCOMO-81:

Software Project	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

Intermediate COCOMO 81

Estimation of development effort:

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic: **Effort = 3.2(KLOC)$^{1.05}$ PM**
Semi-Detached: **Effort = 3.0(KLOC)$^{1.12}$ PM**
Embedded: **Effort = 2.8(KLOC)$^{1.20}$ PM**

Estimation of development time:

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic: **Tdev = 2.5(Effort)$^{0.38}$ Months**
Semi-detached: **Tdev = 2.5(Effort)$^{0.35}$ Months**

Detailed COCOMO 81

A major shortcoming of both the basic and intermediate COCOMO models is that they consider a software product as a single homogeneous entity. However, most large systems are made up several smaller sub-systems. These sub-systems may have widely different characteristics.

The Detailed COCOMO Model differs from the Intermediate COCOMO model in that it uses effort multipliers for each phase of the project.

These phase dependent effort multipliers yield better estimates because the cost driver ratings may be different during each phase.

In Detailed COCOMO-81 Model, the cost of each subsystem is estimated separately. This approach reduces the margin of error in the final estimate.

Detailed COCOMO-81 is the most refined and complex estimation model.

Based on the fact that different cost drivers have different impacts during various phases of a project.

Detailed COCOMO 81

Detailed COCOMO-81 includes all 15 cost driver attributes under 3 modes (organic, semi-detached, embedded) at 3 different levels of development:

1. Module level
2. Sub-system level
3. System level

Example: A distributed Management Information System (MIS) product for an organization having offices at several places across the country can have the following sub-components: Database part

- Graphical User Interface (GUI) part
- Communication part

Of these, the communication part can be considered as **Embedded software**. The database part could be **Semi-detached software**, and the GUI part **Organic software**. The costs for these three components can be estimated separately, and summed up to give the overall cost of the system.

Advantages and Limitations of COCOMO 81

Advantages of COCOMO'81 Model:

1.COCOMO is transparent, one can see how it works unlike other models such as SLIM
2.Drivers are particularly helpful to the estimator to understand the impact of different factors that affect project costs.

Limitations of COCOMO'81 Model:

1.It is hard to accurately estimate KLOC early on in the project, when most effort estimates are required.
2.KLOC, actually, is not a size measure it is a length measure.
3.Extremely vulnerable to misclassification of the development mode.
4.Success depends largely on tuning the model to the needs of the organization, using historical data which is not always available.

Thank You