Project Estimation

Contd.

COnstructive COst MOdel

- It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.
- The key parameters which define the quality of any software products, which are also an outcome of the CoCoMo are primarily Effort & Schedule:
  - **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
  - **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

In COCOMO, projects are categorized into three types:

Organic: A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

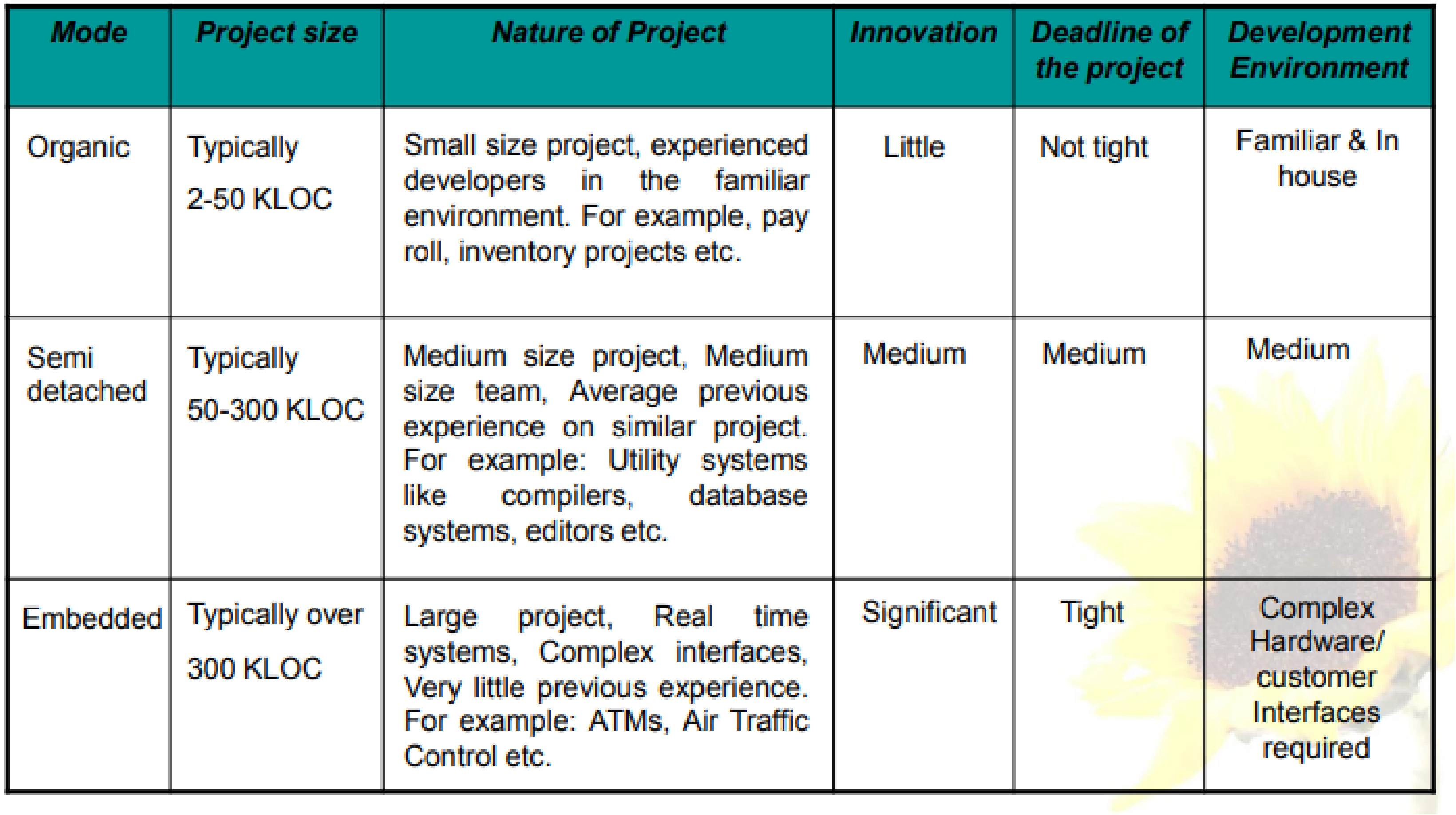
In COCOMO, projects are categorized into three types:

Semidetached: A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.

In COCOMO, projects are categorized into three types:

- **Embedded:** A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. For Example: ATM, Air Traffic control.

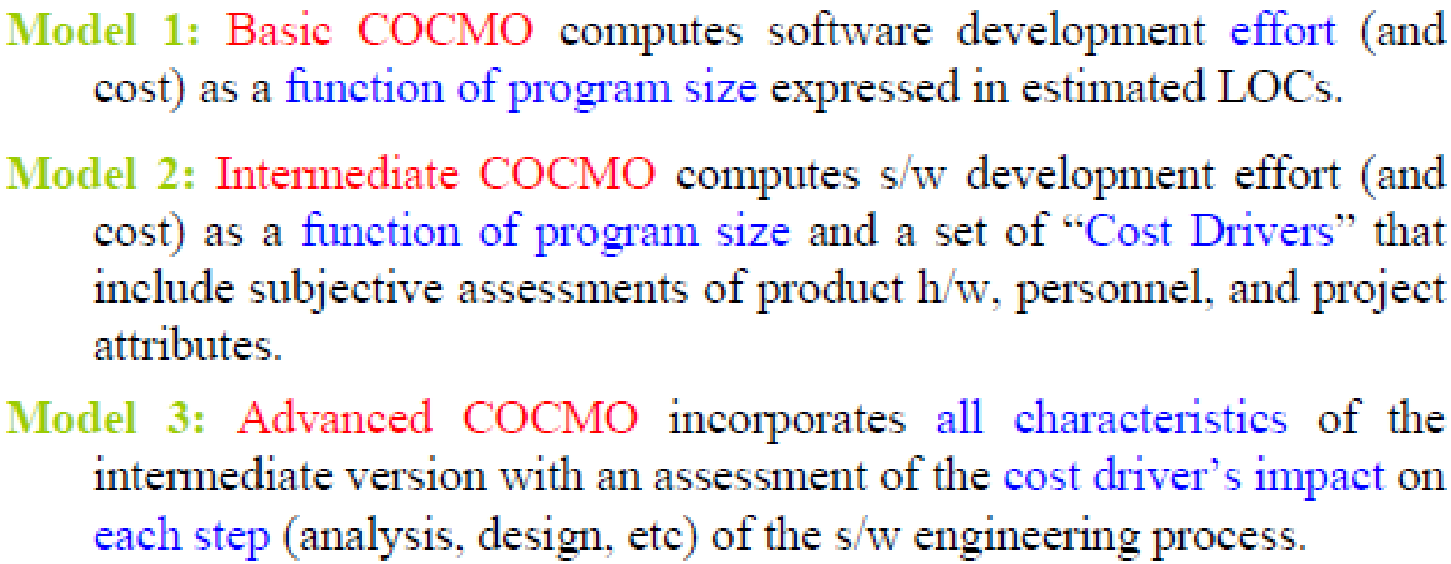
Comparison of three COCOMO modes



Mode	*Project size*	*Nature of Project*	*Innovation*	*Deadline of the project*	*Development Environment*
Organic	Typically 2-50 KLOC	Small size project, experienced developers in the familiar environment. For example, pay roll, inventory projects etc.	Little	Not tight	Familiar & In house
Semi detached	Typically 50-300 KLOC	Medium size project, Medium size team, Average previous experience on similar project. For example: Utility systems like compilers, database systems, editors etc.	Medium	Medium	Medium
Embedded	Typically over 300 KLOC	Large project, Real time systems, Complex interfaces, Very little previous experience. For example: ATMs, Air Traffic Control etc.	Significant	Tight	Complex Hardware/ customer Interfaces required

The COCOMO Model

- COnstructive COst MOdel
- COCOMO



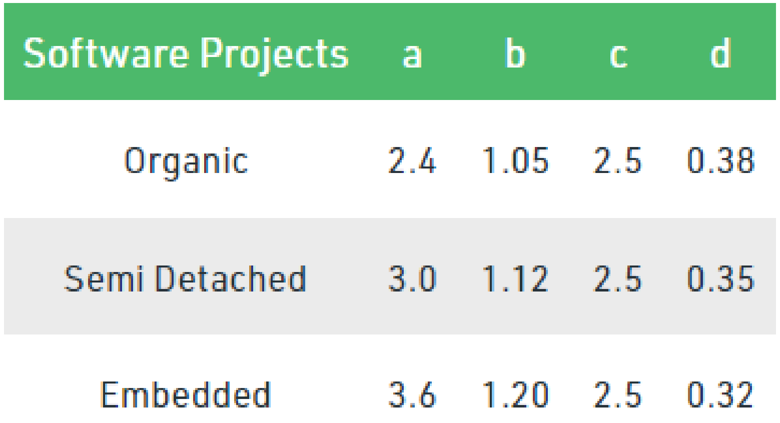
Model 1: Basic COCMO computes software development effort (and cost) as a function of program size expressed in estimated LOCs.

Model 2: Intermediate COCMO computes s/w development effort (and cost) as a function of program size and a set of "Cost Drivers" that include subjective assessments of product h/w, personnel, and project attributes.

Model 3: Advanced COCMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc) of the s/w engineering process.

Basic COCOMO

The constant values a,b,c and d for the Basic Model for the different categories of system:



Software Projects	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

$$\mathrm{E}= \mathrm{a}(\mathrm{KLOC})^b$$

$$time = c(Effort)^d$$

$$Personrequired = Effort/time$$

An Example

- We have determined our project fits the characteristics of **Semi-Detached** mode.
- We estimate our project will have **32,000** Delivered Source Instructions. Using the formulas, we can estimate:
- **Effort** = 3.0*(32) $^{\mathbf{1.12}}$ = 146 person-months
- **Schedule** = 2.5*(146) $^{\mathbf{0.35}}$ = 14 months
- **Productivity** = 32,000 DSI / 146
  = 219 DSI/pm
- **Average Staffing** = 146 MM /14 months
  = 10

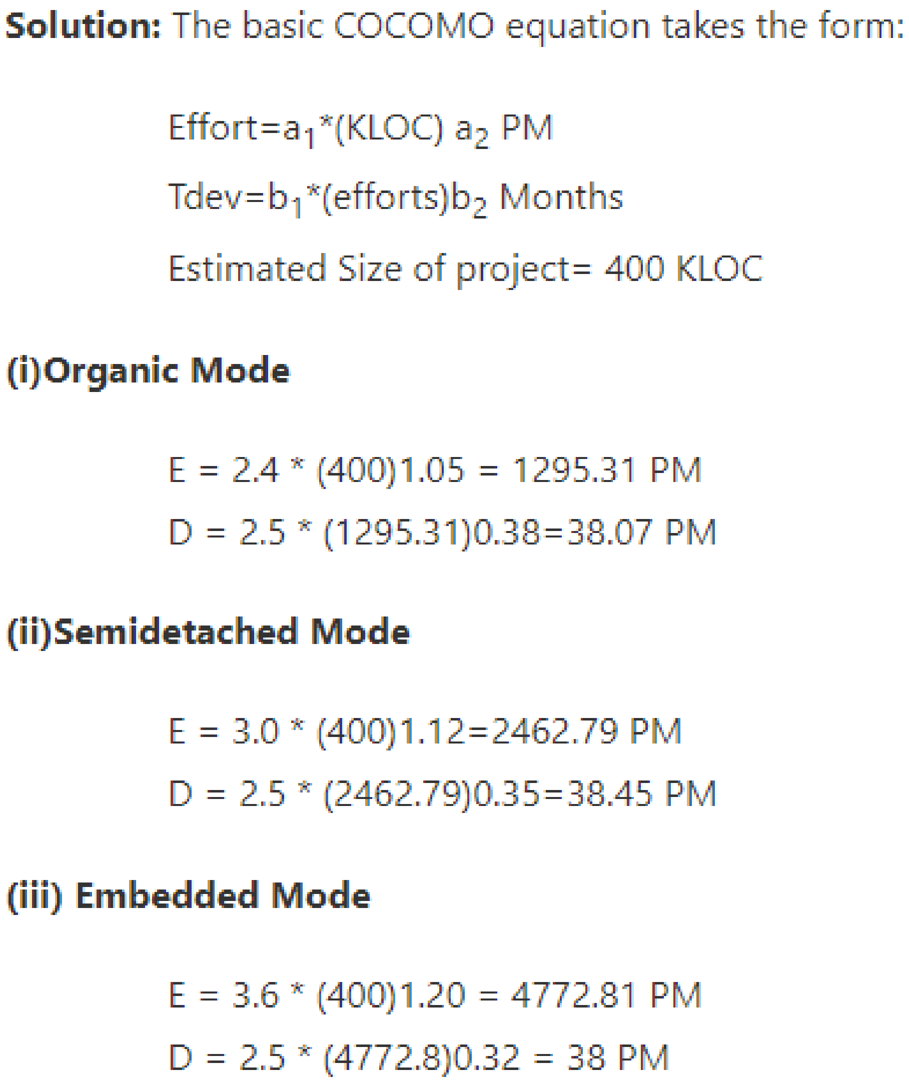
Productivity = DSI / EFFORT

Average staffing = Effort / Schedule

Question

- Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three model i.e., organic, semi-detached & embedded.

Solution



Solution: The basic COCOMO equation takes the form:

Effort=a_1*(KLOC) a_2 PM

Tdev=b_1*(efforts)b_2 Months

Estimated Size of project= 400 KLOC

(i)Organic Mode

E = 2.4 * (400)1.05 = 1295.31 PM

D = 2.5 * (1295.31)0.38=38.07 PM

(ii)Semidetached Mode

E = 3.0 * (400)1.12=2462.79 PM

D = 2.5 * (2462.79)0.35=38.45 PM

(iii) Embedded Mode

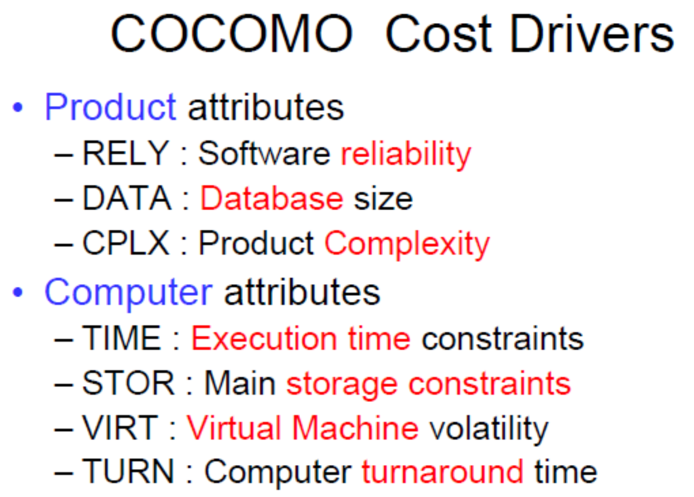
E = 3.6 * (400)1.20 = 4772.81 PM

D = 2.5 * (4772.8)0.32 = 38 PM

Intermediate COCOMO

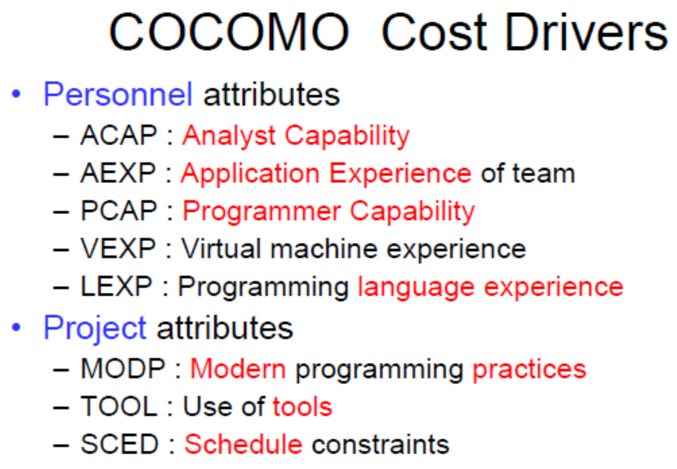
- The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software system. However, in reality, no system's effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation.

Cost drivers



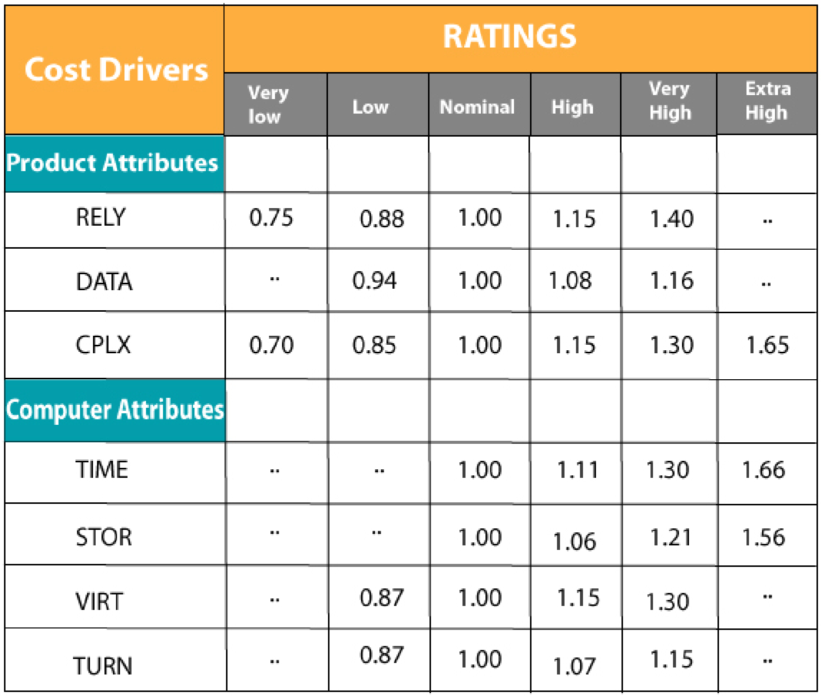
COCOMO Cost Drivers

- Product attributes
  - RELY : Software reliability
  - DATA : Database size
  - CPLX : Product Complexity
- Computer attributes
  - TIME : Execution time constraints
  - STOR : Main storage constraints
  - VIRT : Virtual Machine volatility
  - TURN : Computer turnaround time

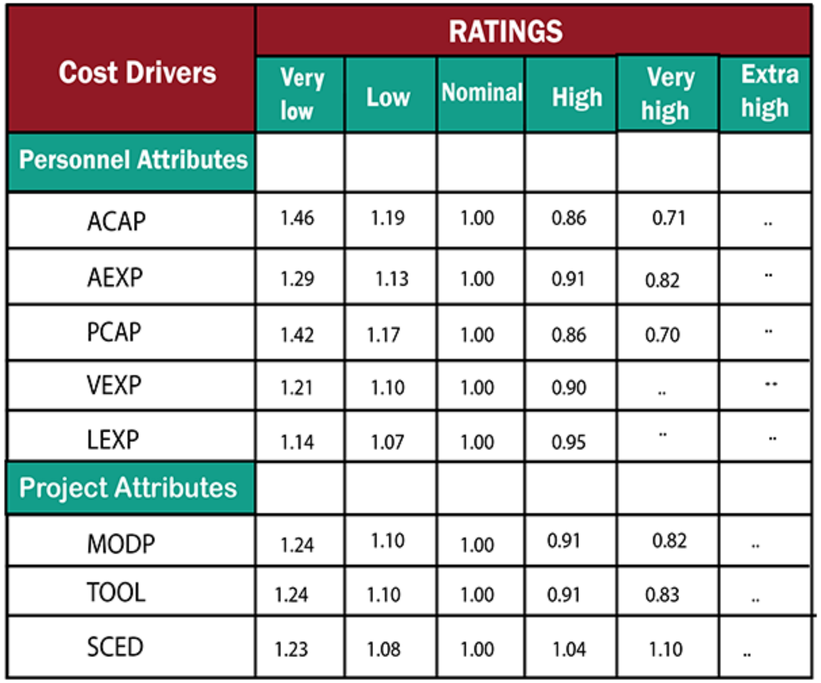


COCOMO Cost Drivers

- Personnel attributes
  - ACAP : Analyst Capability
  - AEXP : Application Experience of team
  - PCAP : Programmer Capability
  - VEXP : Virtual machine experience
  - LEXP : Programming language experience
- Project attributes
  - MODP : Modern programming practices
  - TOOL : Use of tools
  - SCED : Schedule constraints



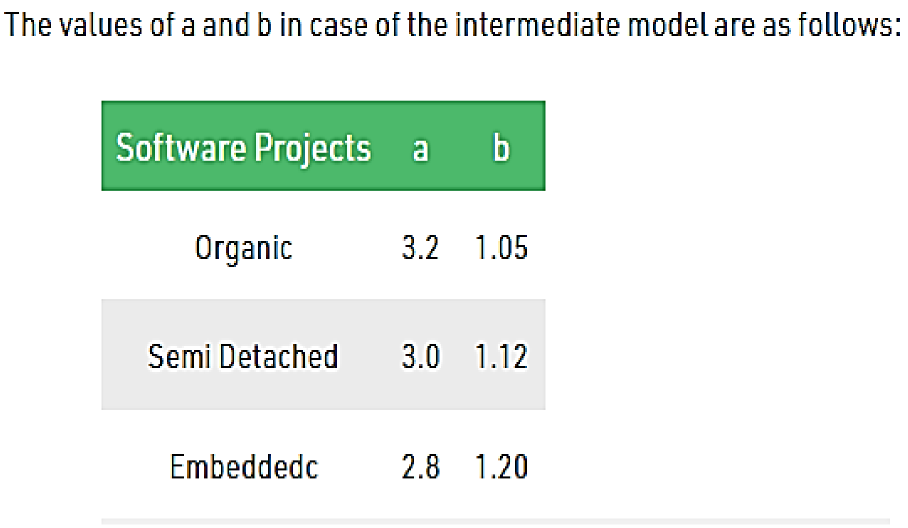
Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	..	..	1.00	1.11	1.30	1.66
STOR	..	..	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..



Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90	..	..
LEXP	1.14	1.07	1.00	0.95	..	..
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

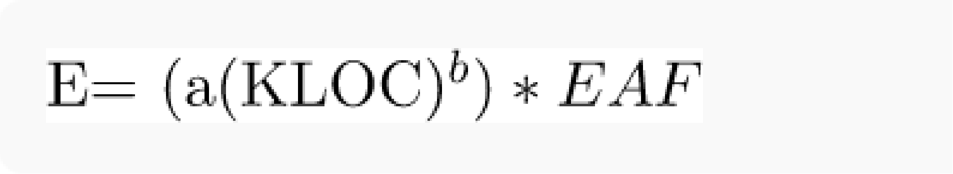
Contd.

- The project manager is to rate these 15 different parameters for a particular project on a scale of one to three. Then, depending on these ratings, appropriate cost driver values are taken from the above table. These 15 values are then multiplied to calculate the EAF (Effort Adjustment Factor).
- **EAF =** It is an Effort Adjustment Factor, which is calculated by multiplying the parameter values of different cost driver parameters.



The values of a and b in case of the intermediate model are as follows:

Software Projects	a	b
Organic	3.2	1.05
Semi Detached	3.0	1.12
Embeddedc	2.8	1.20



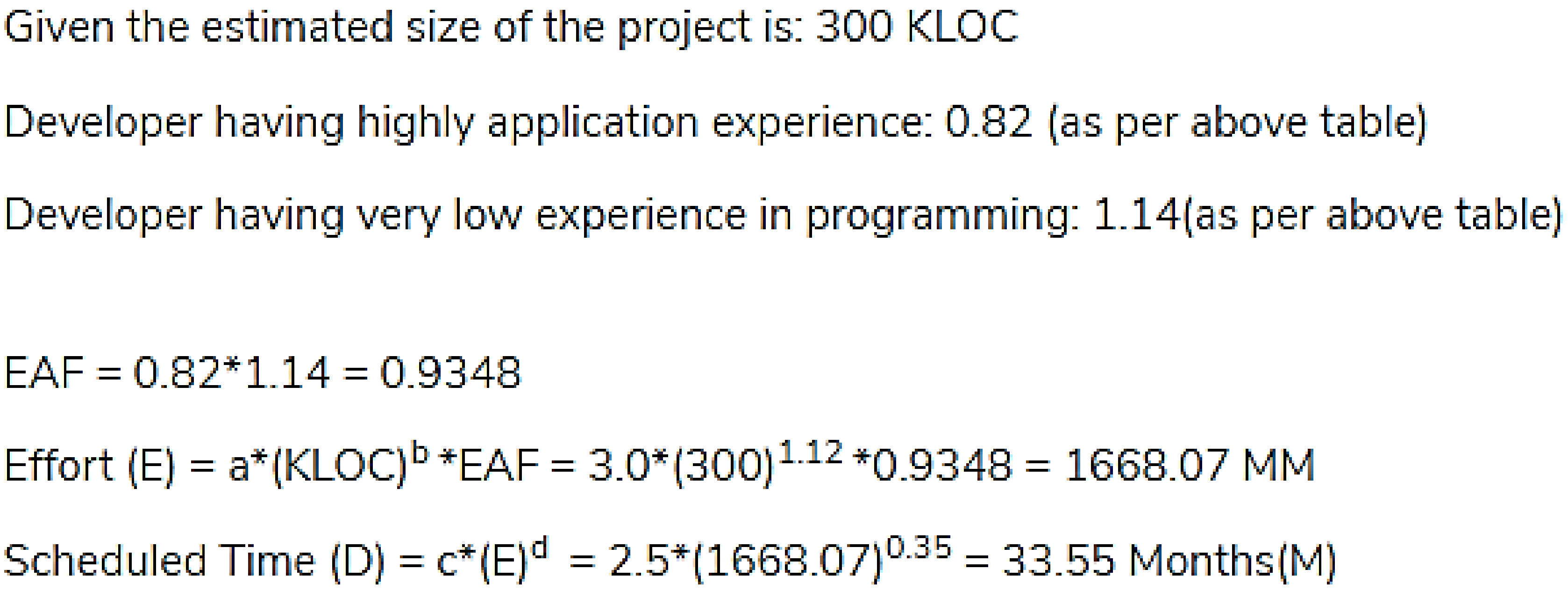
$$\text{E}= (\text{a}(\text{KLOC})^{b}) * EAF$$

Example

- A project was estimated with a size of 300 KLOC. Calculate the Effort, Scheduled time for development by considering developer having very high application experience and very low experience in programming.

- Scheduled Time (D) = $c*(E)^d$ Months(M)

Solution



Given the estimated size of the project is: 300 KLOC

Developer having highly application experience: 0.82 (as per above table)

Developer having very low experience in programming: 1.14(as per above table)

EAF = 0.82*1.14 = 0.9348

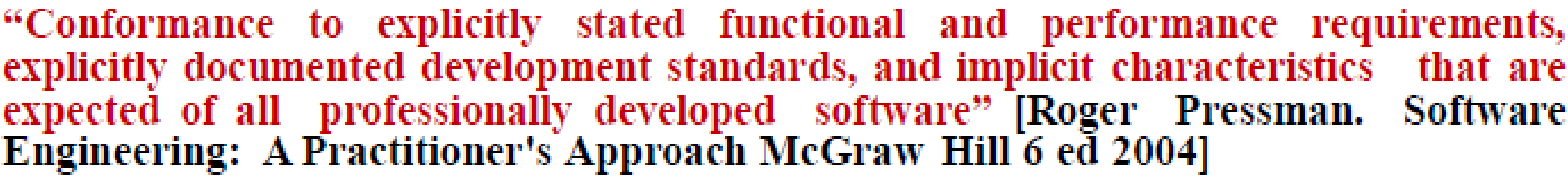
Effort (E) = $a*(KLOC)^{b}$ *EAF = $3.0*(300)^{1.12}$ *0.9348 = 1668.07 MM

Scheduled Time (D) = $c*(E)^{d}$ = $2.5*(1668.07)^{0.35}$ = 33.55 Months(M)

Project Quality Management

Project Quality management:

- International Standard of Organization ISO defines quality as:
  - “The totality of characteristics of an entity that bear on its ability to satisfy stated or implied needs”
  - **“Conformance to requirements** and **fitness for use**.
  - **Conformance to requirement means**: Project processes and products meet written specifications.
  - **fitness for use**: product can be used as intended.



“Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software” [Roger Pressman. Software Engineering: A Practitioner's Approach McGraw Hill 6 ed 2004]

Main Processes in Project Quality Management

The main purpose of quality management is to ensure that the project will satisfy the needs for which it was undertaken.

- ➢Quality management involves establishing effective relationships with the stakeholders to meet their stated and implied needs.
- ➢MAIN Processes involved in Quality management are:
    - Quality Planning
    - Quality Assurance
    - Quality Control

Quality Planning:

- Includes which quality standards are relevant to the project and how to satisfy them.
- Incorporating quality standards into project design is a key part.
- E.g., how long it would take to get reply from helpdesk or how long it should take to ship a replacement part of h/w under warranty? or response time of a system or consistent or accurate information is produced.

Quality Assurance:

- Involves periodically evaluating project performance to ensure the project will satisfy the relevant quality standards.
- It involves taking responsibility of quality during and at the end of the project.

Quality Control:

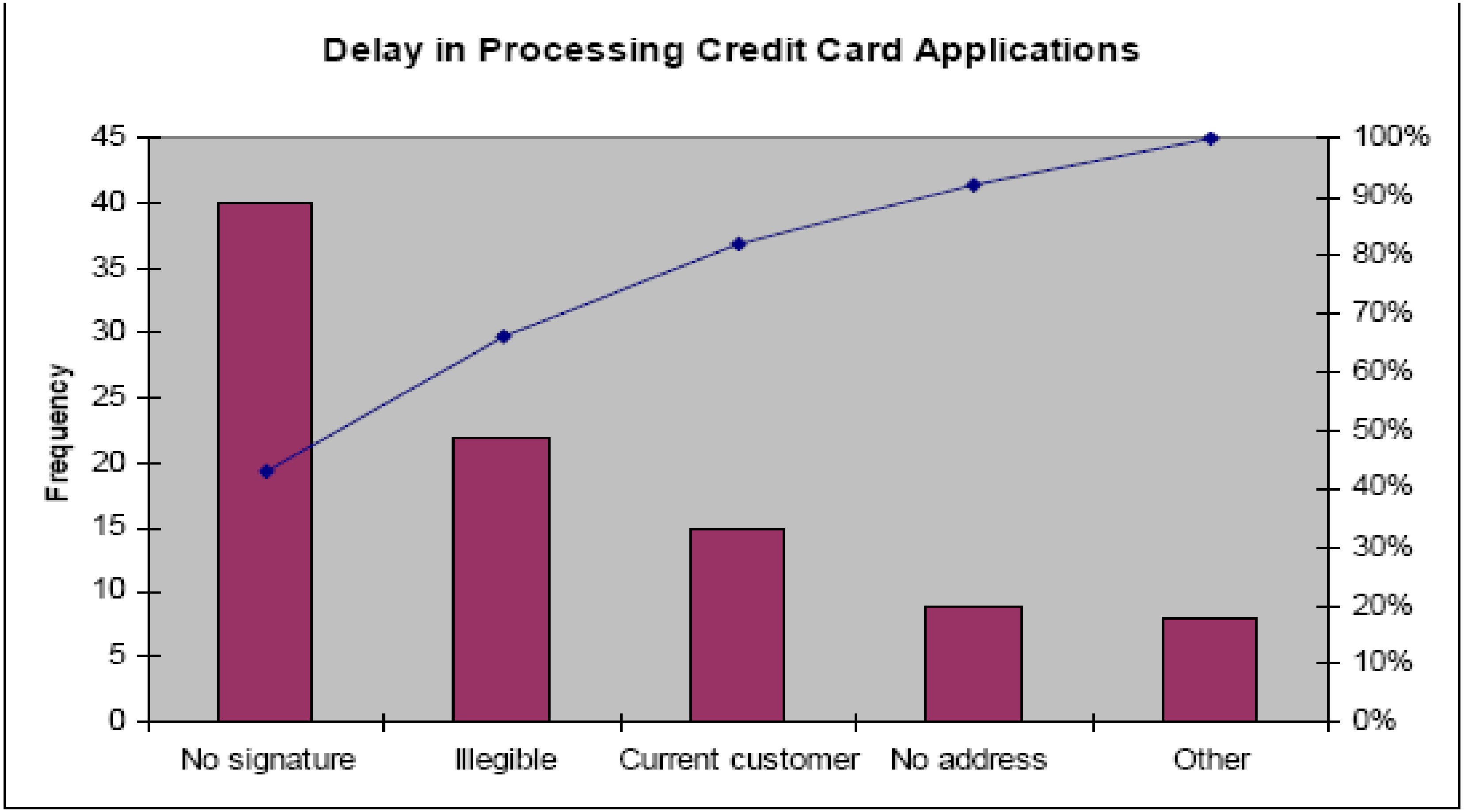
- Comply with the relevant quality standards while identifying ways to improve over all quality.
- This method is associated with tools and techniques such as Pareto charts, quality control charts and statistical sampling.

Quality assurance vs quality control

- One of the major points of quality control vs QA is that assurance of quality is done before starting a project whereas the quality control begins once the product has been manufactured.
- Assurance of quality is a proactive or preventive process to avoid defects whereas quality control is a corrective process to identify the defects in order to correct them.

Quality control tool: Pareto Chart

Involves identifying the vital few contributors that account for most quality problems. It is sometimes referred as 80-20 rule (80 % of problems due to 20% of causes)


Delay in Processing Credit Card Applications
Frequency
No signature
Illegible
Current customer
No address
Other