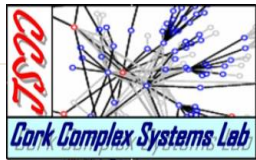
Metrics for Project Planning:
COCOMO Models

CS 6406



Overview

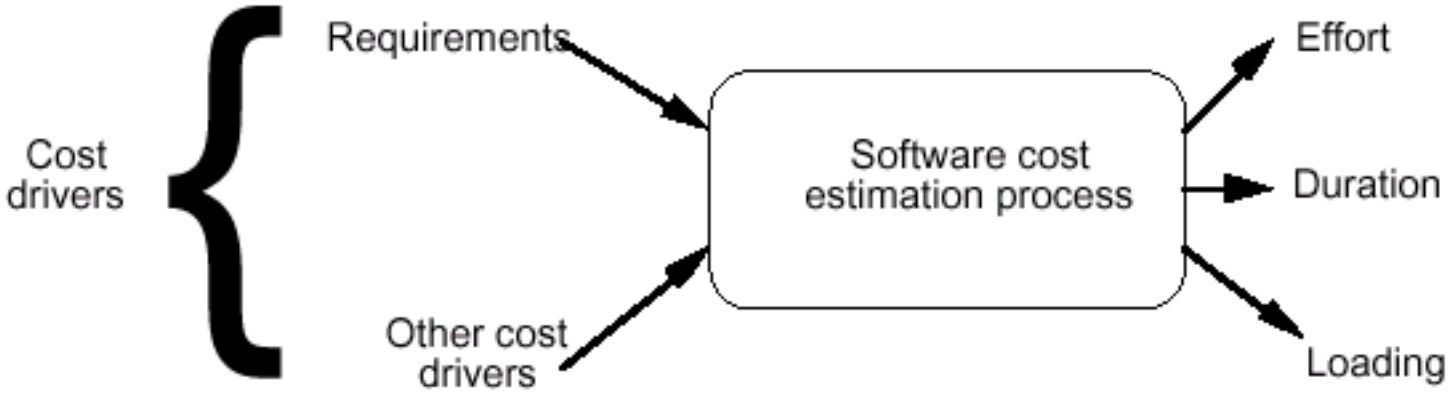
- Software project planning
- Role of metrics and planning models
- Model example
  - COCOMO

Learning Outcomes

- Most widely-used software planning tool: COCOMO
- Gain experience with how to use this tool

- Details
  - You are NOT expected to memorise the model
  - You will need to be able to perform computations given a model
  - Only simple models will be expected in exam situations

Software Cost Estimation



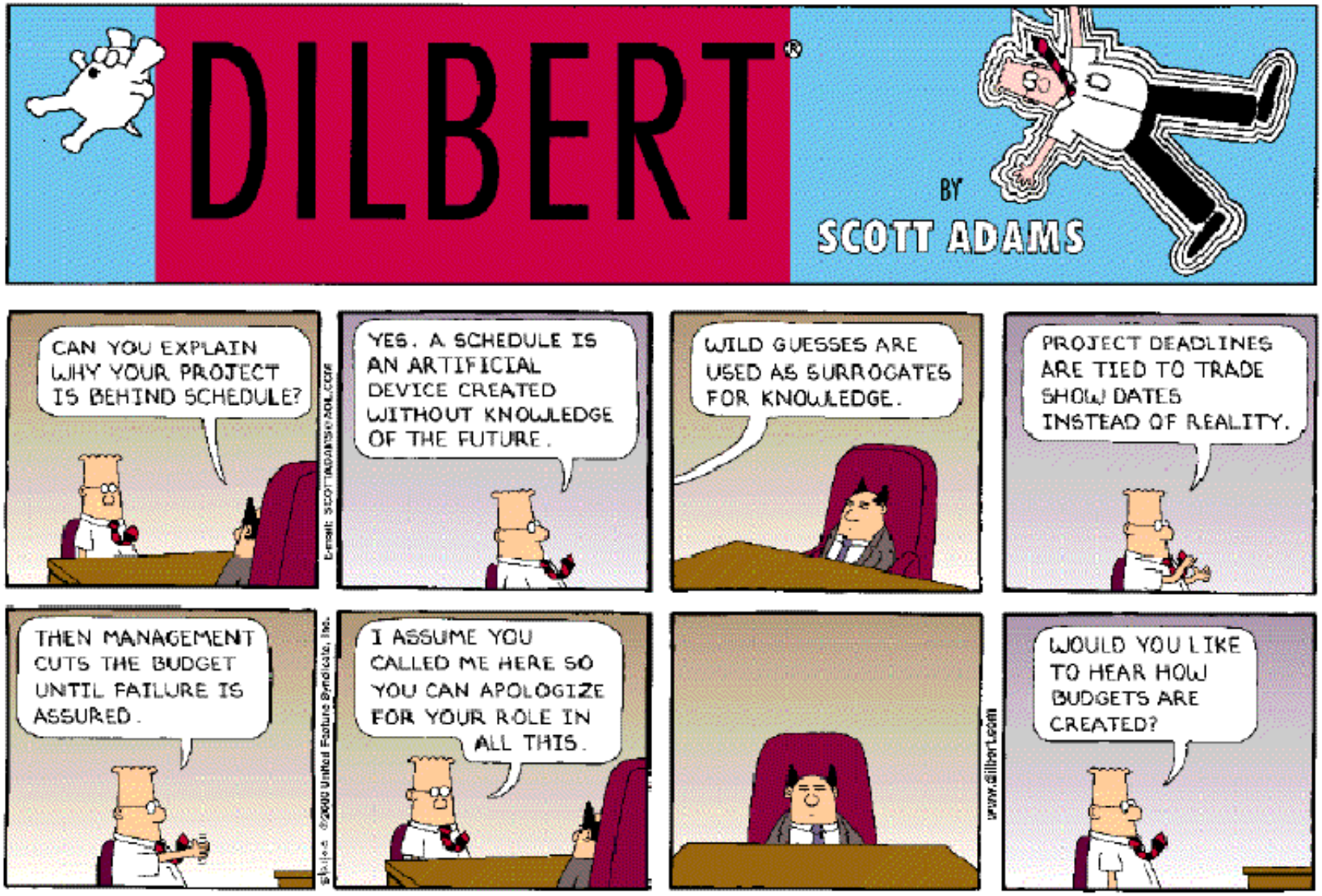
Types of Planning Models

- Size-based models
  - COCOMO: based on lines of code
  - Uses statistical parameter estimation
- Function-point based
  - Use notion of significant software “function”
    - Input, output, function estimation
- Structure-based
  - Examine *structure* of code
  - Must have code already written/planned

Project Management and Mr. Murphy

1. Logic is a systematic method of coming to the wrong conclusion with confidence.
2. Technology is dominated by those who manage what they do not understand.
3. Nothing ever gets built on schedule or within budget.
4. If mathematically you end up with the incorrect answer, try multiplying by the page number.

Cork Complex Systems Lab



Motivation

Software cost estimation provides:

- vital link between the general concepts and techniques of **economic analysis** and the particular world of **software engineering.**
- Essential part of the foundation for **good software management**.

Cost of a project

- The cost in a project is due to:
  - the requirements for software, hardware and human resources
  - the cost of software development is due to the human resources needed
  - most cost estimates are measured in ***person-months (PM)***

Cost of a project (.)

- the cost of the project depends on
  - the nature and characteristics of the project
- the accuracy of the estimate depends on the amount of reliable information we have about the final product
  - Good model (?)
  - Good project plan (?)

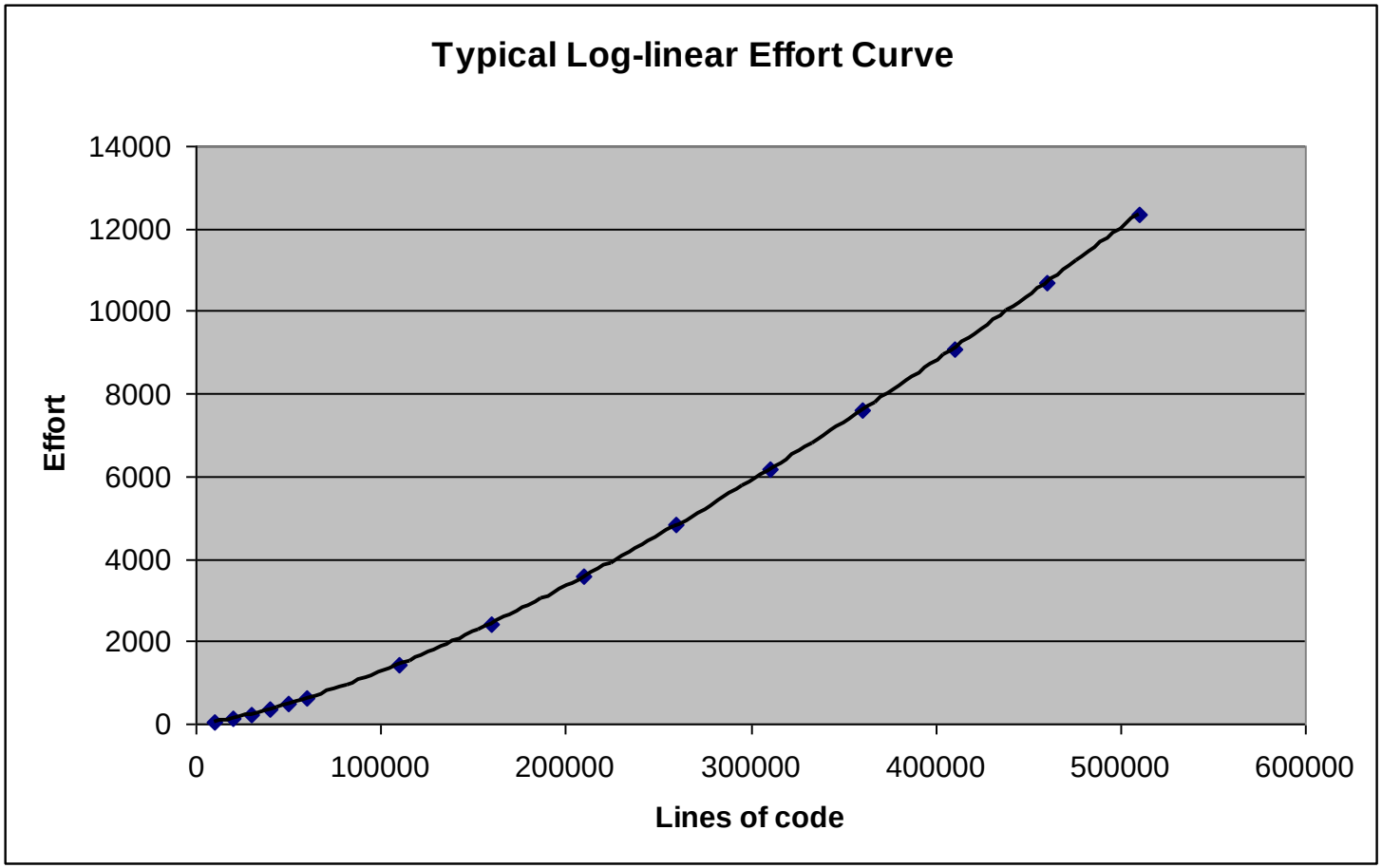
Introduction to COCOMO models

- The **COstructive COst Model** (COCOMO) is the most widely used software estimation model in the world
- The COCOMO model predicts the **effort** and **duration** of a project based on
  - inputs relating to the size of the resulting systems
  - a number of "**cost drivers**" that affect productivity.

Overview of COCOMO

- Predictive model of man-hours for a project
- Statistical model
  - Based on statistical analysis of prior projects

Typical Effort Vs Project Size Curve



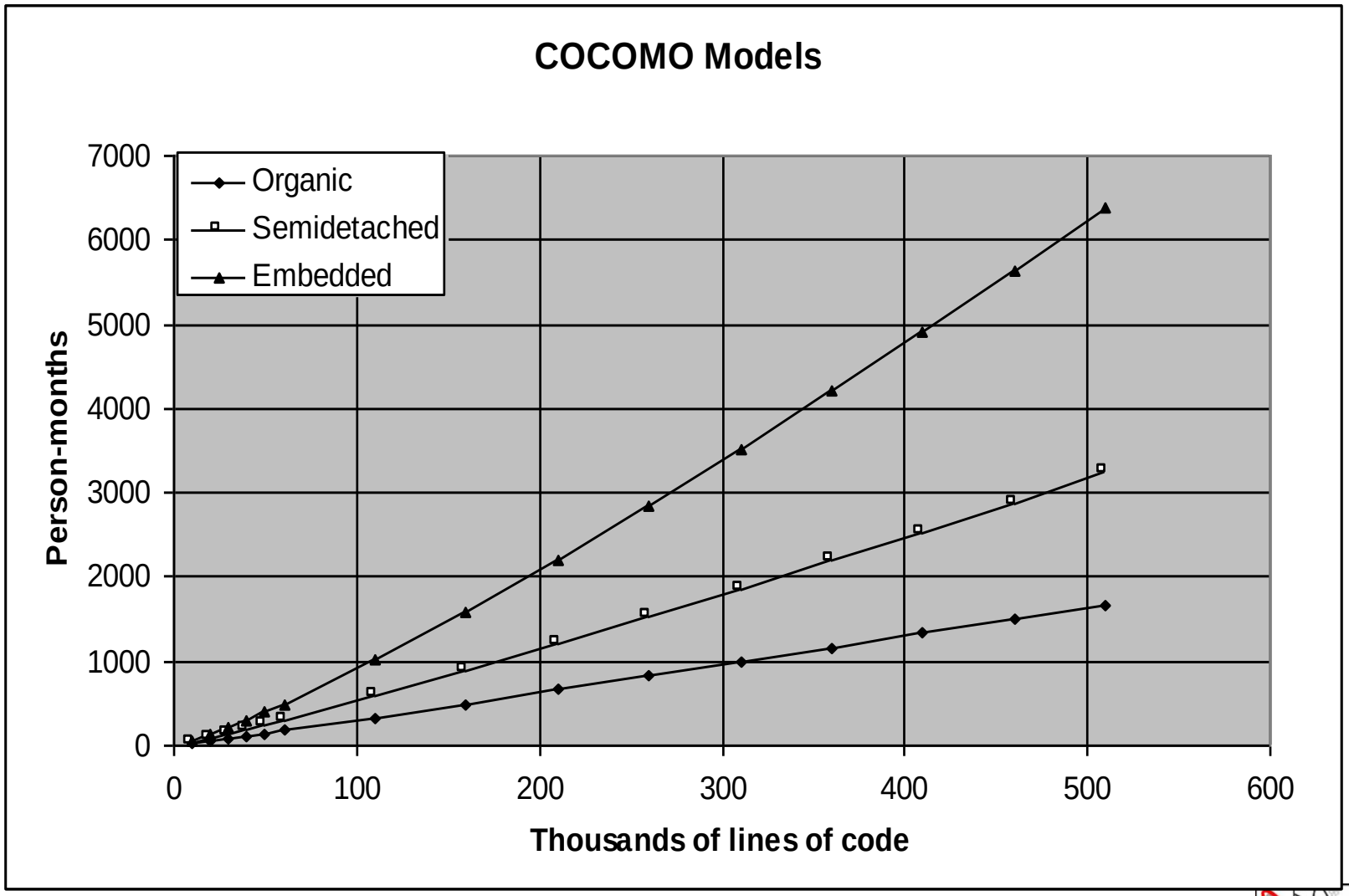
Constructive Cost Model (COCOMO)

- Developed by Barry Boehm
- Statistical model of software development effort and time.
- Base on results from 63 projects completed at TRW.
- Basic model is a log-linear regression model that fits the 63 projects
- Productivity ranges:
  - 20 - 1250 LOC/PM

Basic COCOMO

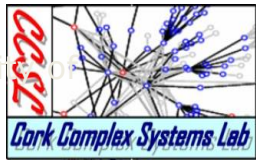
- Organic - small to medium size, familiar projects
  - Person-months=2.4(KLOC)$^{1.05}$
  - Development-time = 2.5(PM)$^{.38}$
- Semidetached - intermediate
  - Person-months=3.0(KLOC)$^{1.12}$
  - Development-time = 2.5(PM)$^{.35}$
- Embedded - ambitious, tightly constrained
  - Person-months=3.6(KLOC)$^{1.20}$
  - Development-time = 2.5(PM)$^{.32}$

COCOMO Models



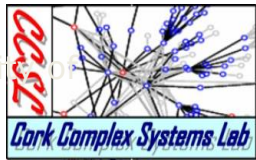
Cost Drivers

- Product Attributes
  - Required Reliability
  - Database Size
  - Product Complexity
- Computer Attributes
  - Execution Time Constraints
  - Main storage constraints
  - Virtual Machine Volatility
  - Computer turnaround time



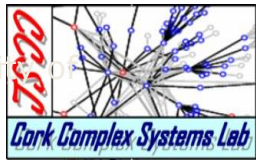
More Cost Drivers

- Personnel Attributes
  - Analyst Capability
  - Application Experience
  - Programmer Capability
  - Virtual Machine Experience
  - Programming Language Experience
- Project Attributes
  - Modern Programming Practices
  - Use of Software Tools
  - Required Development Schedule



Example

- Need to produce 10,000 LOC (10 KLOC).
- Small project, familiar development
- Use organic model:
  - Person-months=$2.4(10)^{1.05}$ =26.9 Person-months
  - Development-time = $2.5(26.9)^{.38}$ =8.7 Months
  - Average People = 26.9 PM/8.7 Months = 3 People
- *Linear model*
  - *3 people would take 16.5 months, at 50 person-months*

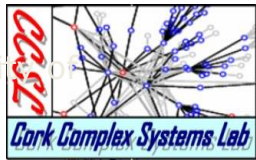


Example

- We also know that the design experience is low
  - Analyst, - 1.19
  - application, - 1.13
  - programmer experience is low. - 1.17
- Yet the programming experience is high - .95
- Adjustment factor 1.19*1.13*1.17*.95 = 1.49
- PM = 26.9*1.49 = 40 Person-months
- Development time = 10.2 Months
- People = 3.9 People

Drawbacks

- COCOMO has to be calibrated to your environment.
- Very sensitive to change.
  - Over a person-year difference in a 10 KLOC project with minor adjustments
- Broad brush model that can generate significant errors

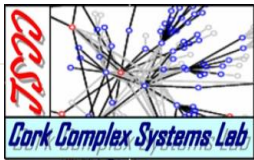


COCOMO 2.0

- Includes
  - COTS and reusable software
  - Degree of understanding of requirements and architectures
  - Schedule constraints
  - Project size
  - Required reliability
- Three Types of models
  - Application Composition – Prototyping/Bidding
  - Early Design - Alternative evaluation
  - Post-architecture - Detailed estimates

COCOMO 2 Models

- COCOMO is defined in terms of three different models:
  - the **Basic model**,
  - the **Intermediate model**, and
  - the **Detailed model**.
- The more complex models account for more factors that influence software projects, and make more accurate estimates.

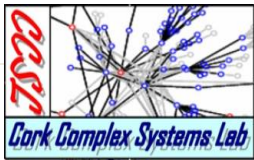


Effort

- Effort Equation
  - **PM = C * (KDSI)^n** (person-months)
    - where **PM** = number of person-month (=152 working hours),
    - **C** = a constant,
    - **KDSI** = thousands of "delivered source instructions" (DSI) and
    - **n** = a constant.

Productivity

- Productivity equation
  - **(DSI) / (PM)**
    - where **PM** = number of person-month (=152 working hours),
    - **DSI** = "delivered source instructions"



Schedule

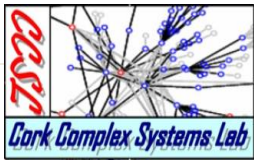
- Schedule equation
  - **TDEV = C * (PM)^{n} (**months)
    - where TDEV = number of months estimated for software development.

Average Staffing

- Average Staffing Equation
  - **(PM) / (TDEV)** (FSP)
    - where FSP means Full-time-equivalent Software Personnel.

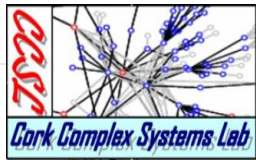
The Development mode

- the most important factors contributing to a project's duration and cost is the Development Mode
  - **Organic Mode**: The project is developed in a familiar, stable environment, and the product is similar to previously developed products. The product is relatively small, and requires little innovation.
  - **Semidetached Mode**: The project's characteristics are intermediate between Organic and Embedded.



The Development mode

- the most important factors contributing to a project's duration and cost is the Development Mode:
    - **Embedded Mode**: The project is characterized by tight, inflexible constraints and interface requirements. An embedded mode project will require a great deal of innovation.



Modes

Feature	Organic	Semidetached	Embedded
Organizational understanding of product and objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-established requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications	Basic	Considerable	Full

Modes (.)

Feature	Organic	Semidetached	Embedded
Concurrent development of associated new hardware and operational procedures	Some	Moderate	Extensive
Need for innovative data processing architectures, algorithms	Minimal	Some	Considerable
Premium on early completion	Low	Medium	High
Product size range	<50 KDSI	<300KDSI	All

Relation between LOC and FP

- Relationship:
  - ***LOC = Language Factor * FP***
  - where
    - **LOC** (Lines of Code)
    - **FP** (Function Points)

Relation between LOC and FPs

Language	LOC/FP
assembly	320
C	128
Cobol	105
Fortan	105
Pascal	90
Ada	70
OO languages	30
4GL languages	20

Relation between LOC and FP(.)

Assuming LOC's per FP for:

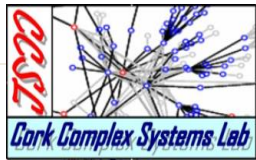
Java = 53,

C++ = 64

aKLOC = FP * LOC_per_FP / 1000

It means for the SpellChekcer Example: (Java)

LOC=52.25*53=2769.25 LOC or 2.76 KLOC



Effort Computation

- The **Basic COCOMO model** computes effort as a function of program size. The Basic COCOMO equation is:
  - ***E = aKLOC^b***
- Effort for three modes of Basic COCOMO.

Mode	**a**	**b**
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.6	1.20

Example

Mode	Effort Formula
Organic	$E = 2.4 * (S^{1.05})$
Semidetached	$E = 3.0 * (S^{1.12})$
Embedded	$E = 3.6 * (S^{1.20})$

Size = 200 KLOC

Effort = $a * \text{Size}^b$

Organic — $E = 2.4 * (200^{1.05}) = 626$ staff-months

Semidetached — $E = 3.0 * (200^{1.12}) = 1133$ staff-months

Embedded — $E = 3.6 * (200^{1.20}) = 2077$ staff-months

Effort Computation

- The **intermediate COCOMO model** computes effort as a function of program size and a set of cost drivers. The Intermediate COCOMO equation is:
  - ***E = aKLOC^b*EAF***
- Effort for three modes of intermediate COCOMO.

Mode	**a**	**b**
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

Effort computation(.)

- **Effort Adjustment Factor**

Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
Required Reliability	.75	.88	1.00	1.15	1.40	1.40
Database Size	.94	.94	1.00	1.08	1.16	1.16
Product Complexity	.70	.85	1.00	1.15	1.30	1.65
Execution Time Constraint	1.00	1.00	1.00	1.11	1.30	1.66
Main Storage Constraint	1.00	1.00	1.00	1.06	1.21	1.56
Virtual Machine Volatility	.87	.87	1.00	1.15	1.30	1.30
Comp Turn Around Time	.87	.87	1.00	1.07	1.15	1.15
Analyst Capability	1.46	1.19	1.00	.86	.71	.71
Application Experience	1.29	1.13	1.00	.91	.82	.82
Programmers Capability	1.42	1.17	1.00	.86	.70	.70
Virtual machine Experience	1.21	1.10	1.00	.90	.90	.90
Language Experience	1.14	1.07	1.00	.95	.95	.95
Modern Prog Practices	1.24	1.10	1.00	.91	.82	.82
SW Tools	1.24	1.10	1.00	.91	.83	.83
Required Dev Schedule	1.23	1.08	1.00	1.04	1.10	1.10

Effort Computation (..)

Total EAF = Product of the selected factors

Adjusted value of Effort: Adjusted Person Months:

APM = (Total EAF) * PM

Example

	Organic	Semidetached	Embedded
a	3.2	3.0	2.8
b	1.05	1.12	1.20

Mode	Effort Formula
Organic	$E = 3.2 * (S^{1.05}) * C$
Semidetached	$E = 3.0 * (S^{1.12}) * C$
Embedded	$E = 2.8 * (S^{1.20}) * C$

---

e.g. Size = 200 KLOC

Effort = $a * Size^{b} * C$

Cost drivers:

- Low reliability .88
- High product complexity 1.15
- Low application experience 1.13
- High programming language experience .95

$C = .88 * 1.15 * 1.13 * .95 = 1.086$

Organic — $E = 3.2 * (200^{1.05}) * 1.086 = 906$ staff-months

Semidetached — $E = 3.0 * (200^{1.12}) * 1.086 = 1231$ staff-months

Embedded — $E = 2.8 * (200^{1.20}) * 1.086 = 1755$ staff-months

Software Development Time

- **Development Time Equation Parameter Table:**

Parameter	Organic	Semi-detached	Embedded
C	2.5	2.5	2.5
D	0.38	0.35	0.32

Development Time, **TDEV = C * (APM **D)**

Number of Personnel, **NP = APM / TDEV**