

# **IT5450- KINH TẾ CÔNG NGHỆ PHẦN MỀM (SOFTWARE ECONOMICS)**

**Năm học 2014-2015**

**Giảng viên: PGS. TS. Huỳnh Quyết Thắng**  
**BM Công nghệ phần mềm**  
**Viện CNTT-TT, ĐHBK HN**  
**[www.soict.hust.edu.vn/~thanghq](http://www.soict.hust.edu.vn/~thanghq)**

---

# **Ước lượng chi phí phần mềm**

# **Software Cost Estimation**

## Ước lượng chi phí phần mềm

---

- Thông thường, không khả thi để ước tính chính xác phần mềm chi phí.
- Tại sao chúng ta phải quan tâm đến ước lượng?
- Ví dụ: Chúng tôi có 6 tháng và 10 chuyên gia phân tích / lập trình, phát triển PM. Mất 6 tháng và 60 người-tháng. Tại sao bận tâm về ước tính chi phí?

## Ước lượng chi phí phần mềm

---

- Một ước lượng tốt là một ước tính cung cấp đầy đủ và rõ ràng cho phép lãnh đạo dự án để đưa ra quyết định tốt làm thế nào để kiểm soát các dự án để đạt mục tiêu của mình  
Steve McConnell. Software Estimation: Demystifying the Black Art
- Theo dõi được: Các điểm chính chúng ta nên tập trung các nỗ lực; Có thể cập nhật: nó phải là dễ dàng để "tinh chỉnh" với mới dữ liệu.
- Tính chính xác
- Tính tin cậy

# Các phương pháp ước tính chi phí phần mềm

---

- Cost estimation: dự đoán của cả số lượng nhân lực và thời gian thực hiện của một dự án
- Methods:
  - Algorithmic
  - Expert judgement
  - Estimation by analogy
  - Parkinsonian
    - Price-to-win
    - Top-down
    - Bottom-up
- Cách tiếp cận tốt nhất là sự kết hợp các phương pháp
- COCOMO là mô hình chi phí được sử dụng rộng rãi nhất, dữ liệu đầy đủ và hiệu chuẩn.

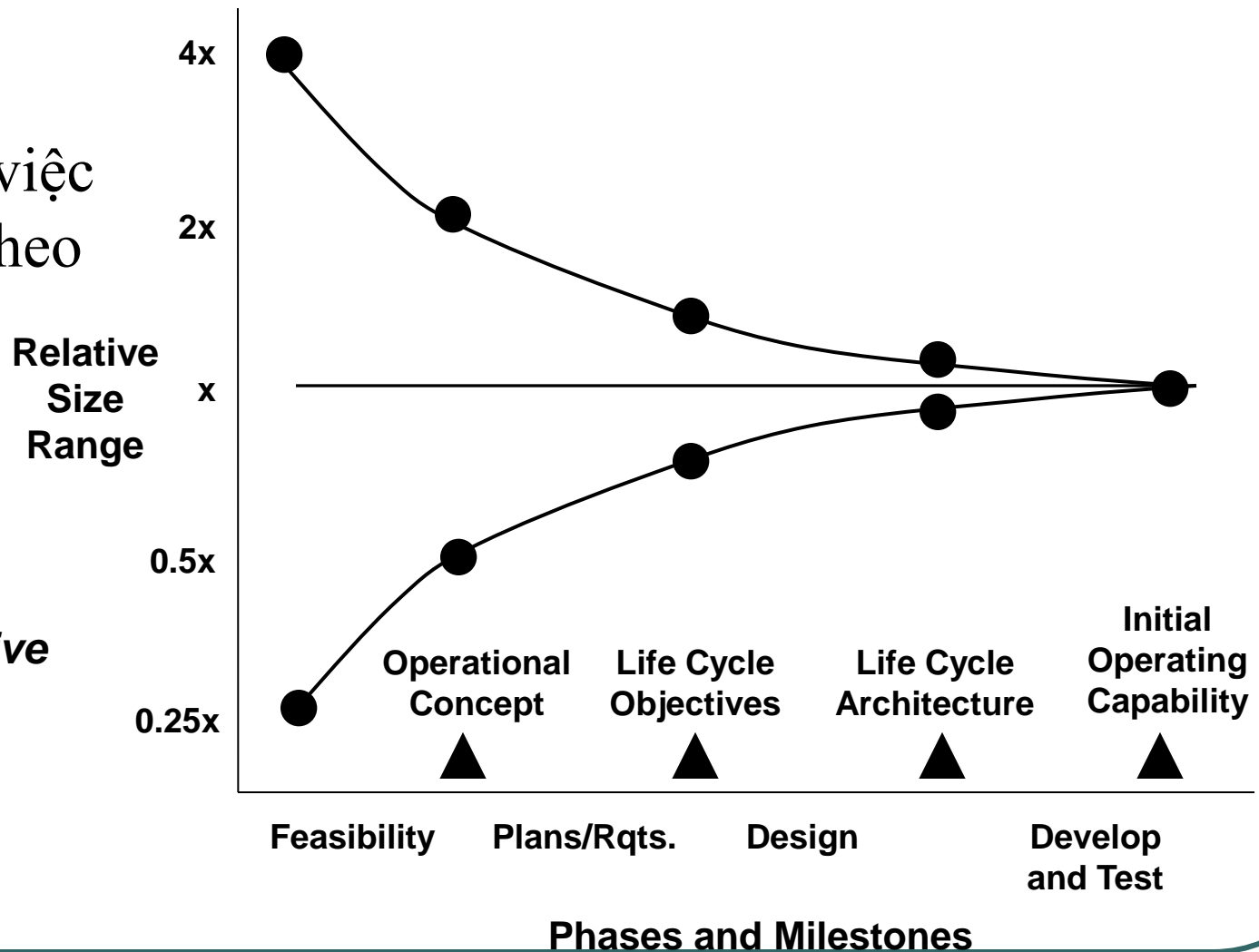
# COCOMO

---

- Xây dựng bởi Dr. Barry Boehm, trong công bố năm 1981: *Software Engineering Economics (sách)*
- COCOMO II được mô tả trong cuốn sách *Software Cost Estimation with COCOMO II*
- COCOMO có thể được sử dụng như công cụ trong xây dựng các dự toán chi phí và các hoạt động liên quan

# Tính chính xác của công việc ước tính

- Mức độ chính xác của công việc ước tính theo thời gian



# Cocomo

---

Là viết tắt của "Chi phí ước tính xây dựng phần mềm Constructive Cost Model"

Được phát triển tại USC (Barry Boehm et al.)  
Dựa trên một cơ sở dữ liệu của 63161 dự án

Phiên bản đầu tiên của COCOMO (nay COCOMO 81)

Phiên bản mới nhất COCOMO II.2000

Trên cơ sở xây dựng mô hình thống kê (phù hợp thực tế dữ liệu vào phương trình)

Có thể được hiệu chỉnh dựa trên công ty cụ thể dữ liệu lịch sử



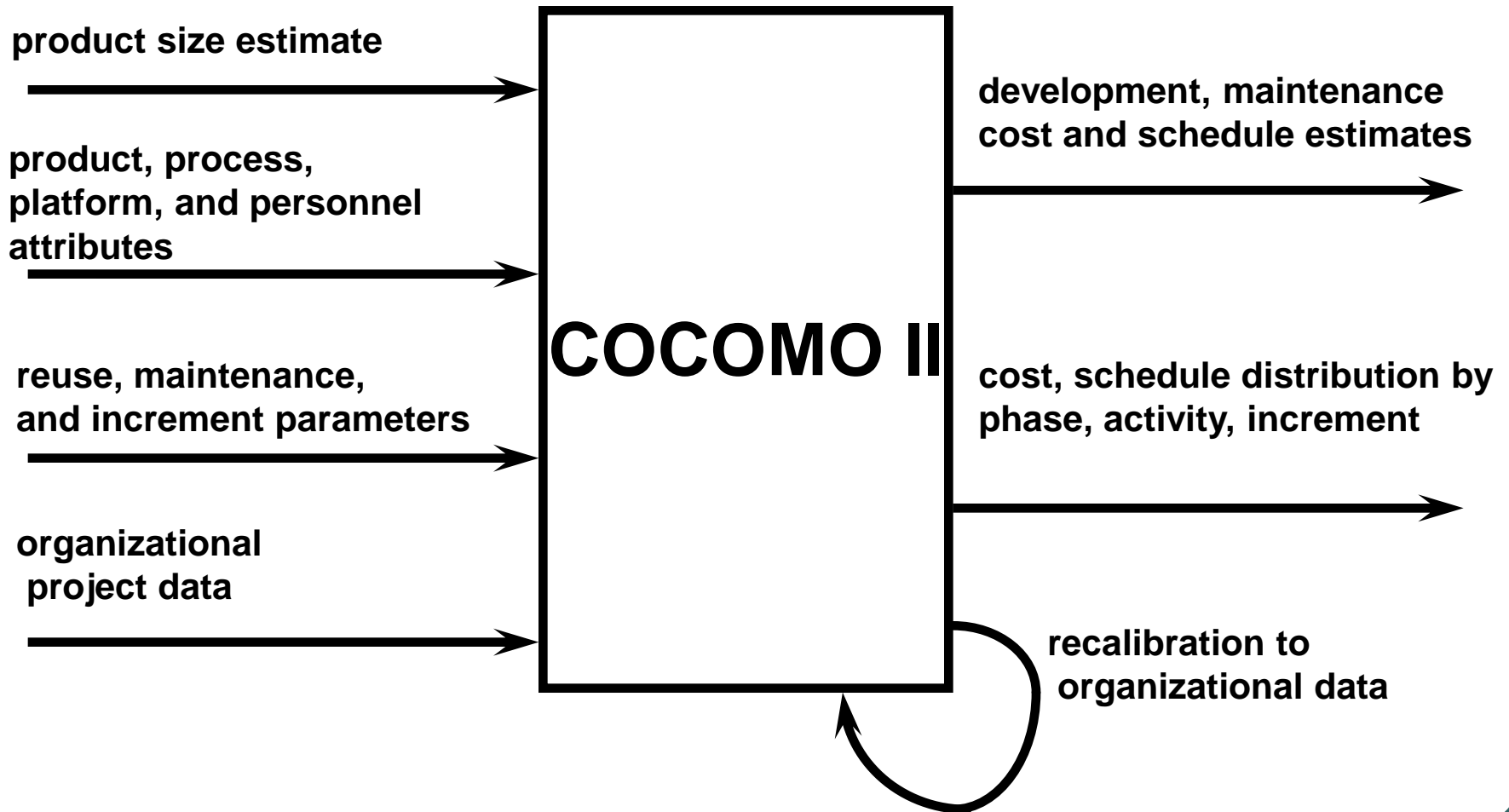
## Basic COCOMO 81

---

- Ứng dụng được hiểu rất rõ, được phát triển bởi nhóm nhỏ, với nhiều kinh nghiệm trong hệ thống liên quan:  $PM = 2.4 (KLOC)^{1.05}$
- Dự án phức tạp, nơi các thành viên có thể có kinh nghiệm hạn chế của hệ thống liên quan.  
 $PM = 3.0 (KLOC)^{1.12}$
- Các dự án phức tạp, nơi các phần mềm là hạn chế bởi hạn chế phần cứng (nhúng), cần phải đáp ứng trong thời gian thực, hoặc là rất quan trọng.  
 $PM = 3.6 (KLOC)^{1.20}$

# Mô hình hộp đen của Cocomo II:

---



## Các đặc điểm chính của COCOMO II

---

- Đảm bảo đa mô hình của các thành phần phát triển khác nhau
- Biến đơn vị cho các chi phí đầu vào của mô hình
- Mềm dẻo trong các thông số đầu vào:
  - SLOCS
  - function points
  - application points
  - other (use cases ...)

# Ứng dụng COCOMO trong Software Decision Making

---

- Quyết định đầu tư và phân tích kinh doanh hợp
- Thiết lập dự toán cho Ngân sách và lịch trình
- Thực hiện phân tích sự cân bằng
- Quản lý rủi ro chi phí
- Quyết định phát triển so với tái sử dụng
- Phần mềm tái sử dụng và quyết định dòng sản phẩm
- Quyết định cải tiến quy trình

# COCOMO Submodels

---

- Applications Composition liên quan đến việc phát triển nhanh chóng hoặc nỗ lực tạo mẫu để giải quyết các vấn đề có nguy cơ cao tiềm năng như giao diện người dùng, phần mềm / hệ thống tương tác, hiệu suất, hoặc trưởng thành công nghệ. Nó có kích thước với điểm ứng dụng (các yếu tố màn hình trọng, báo cáo và 3GL mô-đun).
- The Early Design mô hình liên quan đến việc xem xét, thăm dò phần mềm / hệ thống kiến trúc thay thế và khái niệm về hoạt động sử dụng các điểm chức năng và thiết lập một trình hạt của 7 trình điều khiển chi phí.

## COCOMO Submodels

---

- The Post-Architecture mô hình liên quan đến thực tế phát triển và duy trì một sản phẩm phần mềm sử dụng hướng dẫn nguồn gốc và / hoặc điểm chức năng cho kích thước, với bộ để tái sử dụng và vỡ phần mềm, một tập hợp của 17 trình điều khiển chi phí nhân giống, và một tập hợp 5 yếu tố quyết định mở rộng quy mô số mũ của dự án.

# COCOMO Effort Formulation

---

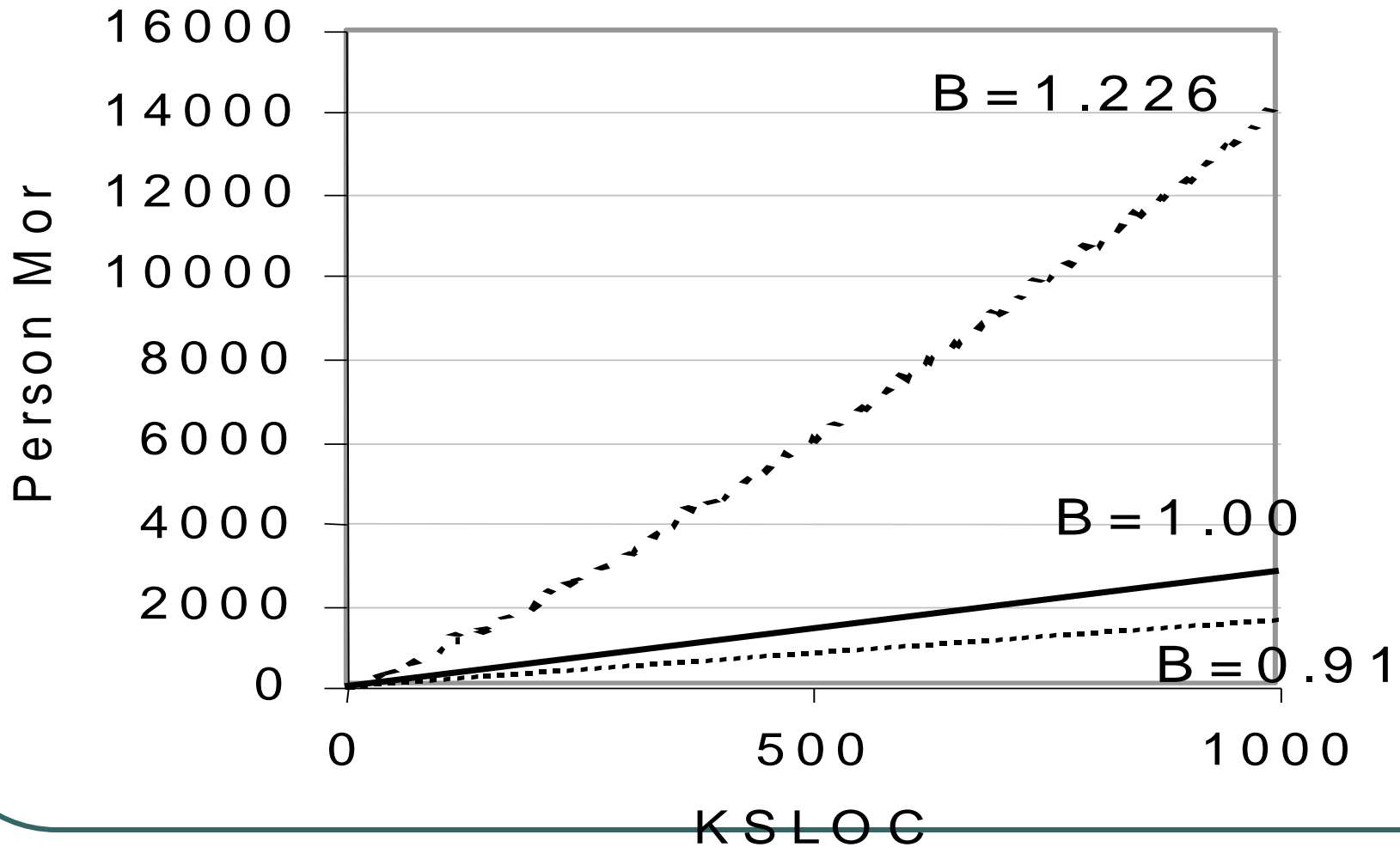
$$\textit{Effort (person-months)} = \mathbf{A} (\textit{Size})^{\mathbf{B}} \prod_{i=1}^{\text{\# of cost drivers}} \mathbf{EM}_i$$

- Where:

- $\mathbf{A}$  is a constant derived from historical project data (currently  $\mathbf{A} = 2.94$  in COCOMOII.2000)
- $\textit{Size}$  is in KSLOC (thousand source lines of code), or converted from function points or object points
- $\mathbf{B}$  is an exponent for the diseconomy of scale dependent on five additive scale drivers according to  $\mathbf{b} = .91 + .01 * \sum \mathbf{SF}_i$ , where  $\mathbf{SF}_i$  is a weighting factor for  $i^{\text{th}}$  scale driver
- $\mathbf{EM}_i$  is the effort multiplier for the  $i^{\text{th}}$  cost driver. The geometric product results in an overall effort adjustment factor to the nominal effort.
- Automated translation effects are not included

# Diseconomy of Scale

- Nonlinear relationship when exponent  $> 1$





# COCOMO Schedule Formulation

---

$$\text{Schedule (months)} = C (\text{Effort})^{(.33+0.2(B-1.01))} \times \text{SCED\%/100}$$

- Where:
  - **Schedule** is the calendar time in months from the requirements baseline to acceptance
  - **C** is a constant derived from historical project data (currently **C** = 3.67 in COCOMOII.2000)
  - **Effort** is the estimated person-months excluding the SCED effort multiplier
  - **B** is the sum of project scale factors
  - **SCED%** is the compression / expansion percentage in the SCED cost driver
- This is the COCOMOII.2000 calibration
- Công thức có thể thay đổi để phản ánh mô hình quá trình cho phần mềm tái sử dụng và COTS, và ảnh hưởng của khả năng thành phần ứng dụng.

## Coverage of Different Processes

---

- COCOMO II cung cấp một khuôn khổ để áp dụng cho các mô hình cho bất kỳ quá trình mong muốn
- Ban đầu COCOMO đã được xác định trên quá trình thác nước: (single-pass, sequential progression of requirements, design, code, test)
- Quá trình hiện đại đồng thời, lặp đi lặp lại, gia tăng, và theo chu kỳ (Rational Unified Process (RUP), the USC Model-Based Architecting and Software Engineering (MBASE) process)
- Nỗ lực và tiến độ được phân bố trong các giai đoạn khác nhau và các hoạt động trong cấu trúc chi tiết công việc của quá trình lựa chọn

## Cost Factors / Các yếu tố chi phí

---

- Yếu tố quan trọng của chi phí phát triển:
  - *scale drivers* (quy mô) là nguồn biến nỗ lực theo cấp số nhân
  - *cost drivers* là nguồn tuyến tính biến nỗ lực
    - sản phẩm, nền tảng, nhân viên và các thuộc tính dự án
    - nỗ lực liên quan đến xếp hạng lái xe chi phí
  - Xác định là khách quan như có thể
- Mỗi yếu tố được đánh giá giữa rất thấp và rất cao cho mỗi hướng dẫn
  - Đánh giá nhân nỗ lực liên quan điều chỉnh chi phí lên hoặc xuống

# Scale Drivers

---

- **Precedentedness (PREC)**
  - Mức độ mà hệ thống là kinh nghiệm mới và áp dụng trong quá khứ
- **Development Flexibility (FLEX)**
  - Mức độ cần phải phù hợp với yêu cầu quy định
- **Architecture/Risk Resolution (RESL)**
  - Mức độ thiết kế tỉ mỉ và loại bỏ nguy cơ rủi ro
- **Team Cohesion (TEAM)**
  - Cần phải đồng bộ hóa các bên liên quan trong Dự án và giảm thiểu xung đột
- **Process Maturity (PMAT)**
  - SEI CMM process maturity rating

# Cost Drivers

---

- **Product Factors**

- Reliability (RELY)
- Data (DATA)
- Complexity (CPLX)
- Reusability (RUSE)
- Documentation (DOCU)

- **Platform Factors**

- Time constraint (TIME)
- Storage constraint (STOR)
- Platform volatility (PVOL)

- **Personnel factors**

- Analyst capability (ACAP)
- Program capability (PCAP)
- Applications experience (APEX)
- Platform experience (PLEX)
- Language and tool experience (LTEX)
- Personnel continuity (PCON)

- **Project Factors**

- Software tools (TOOL)
- Multisite development (SITE)
- Required schedule (SCED)

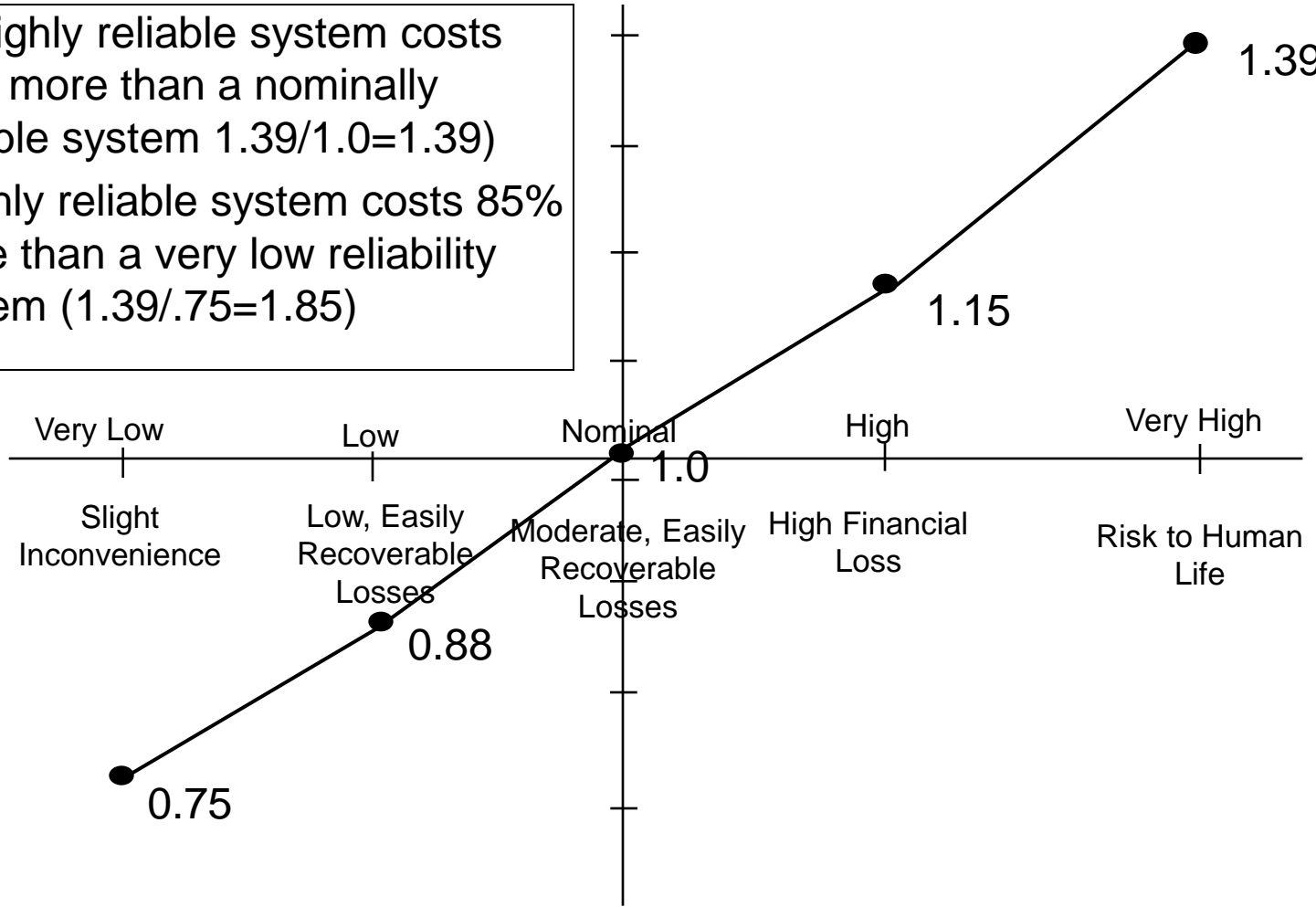
## Example Cost Driver - Required Software Reliability (RELY)

- Đo lường mức độ mà các phần mềm phải thực hiện chức năng dự định của nó trong một khoảng thời gian.
- Câu hỏi: Vai trò của một lỗi phần mềm là gì?

	Very Low	Low	Nominal	High	Very High	Extra High
RELY	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	

## Example Effort Multiplier Values for RELY

E.g. a highly reliable system costs 39% more than a nominally reliable system  $1.39/1.0=1.39$ )  
or a highly reliable system costs 85% more than a very low reliability system ( $1.39/.75=1.85$ )



# Scale Factors

- Sum scale factors  $W_i$  across all of the factors to determine a scale exponent,  $B$ , using  $B = .91 + .01 \sum W_i$

Scale Factors ( $W_i$ )	Very Low	Low	Nominal	High	Very High	Extra High
Precedentedness (PREC)	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
Development Flexibility (FLEX)	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
Architecture/Risk Resolution (RESL)*	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
Team Cohesion (TEAM)	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
Process Maturity (PMAT)	Weighted average of "Yes" answers to CMM Maturity Questionnaire					

\* % significant module interfaces specified, % significant risks eliminated



## Process Maturity (PMAT)

---

- Two methods based on the Software Engineering Institute's Capability Maturity Model (CMM)
- Method 1:  
Overall Maturity Level  
(CMM Level 1 through 5)
- Method 2:  
Key Process Areas  
(see next slide)

# Key Process Areas

- Decide the percentage of compliance for each of the KPAs as determined by a judgement-based averaging across the goals for all 18 Key Process Areas.

$$PMAT = 5 - \left[ \sum_{i=1}^{18} \left( \frac{KPA\%_i}{100} \times \frac{5}{18} \right) \right]$$

Key Process Areas	Almost Always (>90%)	Frequently (60-90%)	About Half (40-60%)	Occasionally (10-40%)	Rarely If Ever (<10%)	Does Not Apply	Don't Know
1 Requirements Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 Software Project Planning	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 Software Project Tracking and Oversight	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Software Subcontract Management	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(See COCOMO II Model Definition Manual for remaining details)

# Cost Drivers

---

- Product Factors
- Platform Factors
- Personnel Factors
- Project Factors

# Platform Factors

---

- Platform

- Đề cập đến mục tiêu độ phức tạp của phần cứng và phần mềm cơ sở hạ tầng (trước đây được gọi là máy ảo).

- Execution Time Constraint (TIME)

- Đo lường các ràng buộc về thời gian áp đặt lên một hệ thống về tỷ lệ phần trăm của thời gian thực hiện sẵn sàng sẽ được sử dụng bởi hệ thống.

---

	Very Low	Low	Nominal		High	Very High	Extra High
TIME			≤ 50% use of available execution time		70%	85%	95%

---

# Platform Factors cont'd

- Main Storage Constraint (STOR)

- Measures the degree of main storage constraint imposed on a software system or subsystem.

	Very Low	Low	Nominal	High	Very High	Extra High
STOR		≤ 50% use of available storage		70%	85%	95%

- Platform Volatility (PVOL)

- Assesses the volatility of the platform (the complex of hardware and software the software product calls on to perform its tasks).

	Very Low	Low	Nominal	High	Very High	Extra High
PVOL		major change every 12 mo.; minor change every 1 mo.	major: 6 mo.; minor: 2 wk.	major: 2 mo.; minor: 1 wk.	major: 2 wk.; minor: 2 days	

# Personnel Factors

## ● Analyst Capability (ACAP)

- Analysts work on requirements, high level design and detailed design. Consider analysis and design ability, efficiency and thoroughness, and the ability to communicate and cooperate.

---

	Very Low	Low	Nominal	High	Very High	Extra High
ACAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	

---

## ● Programmer Capability (PCAP)

- Evaluate the capability of the programmers as a team rather than as individuals. Consider ability, efficiency and thoroughness, and the ability to communicate and cooperate.

---

	Very Low	Low	Nominal	High	Very High	Extra High
PCAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	

---

# Project Factors

---

## ● Use of Software Tools (TOOL)

- Assess the usage of software tools used to develop the product in terms of their capabilities and maturity.

Very Low	Low	Nominal	High	Very High	Extra High
edit, code, debug	simple, frontend, backend CASE, little integration	basic lifecycle tools, moderately integrated	strong, mature lifecycle tools, moderately integrated	strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	

## Reused and Modified Software

---

- Nỗ lực cho phần mềm phù hợp (tái sử dụng hoặc sửa đổi) là không giống như các phần mềm mới.
- Cách tiếp cận: chuyển đổi phần mềm chuyển thể thành kích thước tương đương với các phần mềm mới.



# USC COCOMO Demo

D:\cocomo\book\ars detailed.est - COCOMOII.1998.0

File

View

Edit

Parameters

Calibrate

Phase

Maintenance

Help

Project Name:

APS detailed

Scale Factor

Schedule

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	NOM PM DEV	EST PM DEV	PROD	COST	INST COST	FSWP	RISK
	ruc new	S:3215	0.00	1.08	14.0	15.1	213.5	0.00	0.0	0.9	5.1
	ruc mod	S:351	0.00	1.08	1.5	1.6	213.5	0.00	0.0	0.1	5.1
	ruc reused	S:1721	0.00	1.08	7.5	8.1	213.5	0.00	0.0	0.5	5.1
	tp new	S:13715	0.00	1.26	59.6	75.0	182.8	0.00	0.0	4.3	5.5
	tp mod	S:5392	0.00	1.26	23.4	29.5	182.8	0.00	0.0	1.7	5.5
	rd new	S:12200	0.00	1.50	53.0	79.5	153.5	0.00	0.0	4.6	5.1
	rd reused	S:2848	0.00	1.50	12.4	18.5	153.5	0.00	0.0	1.1	5.1
	dm new	S:15400	0.00	1.08	66.9	72.1	213.5	0.00	0.0	4.1	5.1
	dm reused	S:9432	0.00	1.08	41.0	44.2	213.5	0.00	0.0	2.5	5.1
	dm cots	S:7370	0.00	1.08	32.0	34.5	213.5	0.00	0.0	2.0	5.1
	dc new	S:13200	0.00	1.33	57.4	76.2	173.3	0.00	0.0	4.4	4.3
	dc mod	S:1018	0.00	1.33	4.4	5.9	173.3	0.00	0.0	0.3	4.3
	dc cots	S:863	0.00	1.33	3.8	5.0	173.3	0.00	0.0	0.3	4.3
	bit new	S:16000	0.00	1.81	69.5	126.1	126.9	0.00	0.0	7.2	9.3
	bit reused	S:1260	0.00	1.81	5.5	9.9	126.9	0.00	0.0	0.6	9.3

EST

Sched

PROD

COST

INST

FSWP

RISK

Total SLOC:

103985

Optimistic

481.0

16.1

216.2

0.00

0.0

29.9

Effort (PM):

451.9

Most Likely

601.2

17.4

173.0

0.00

0.0

34.5

83.4

Productivity:

230.1

Pessimistic

751.5

18.8

138.4

0.00

0.0

39.9

Project File : D:\cocomo\book\ars detailed.est Is Loaded

## Recommended Project Cost Data

---

- Đối với mỗi dự án, báo cáo sau đây vào cuối mỗi tháng và cho mỗi bản phát hành:
  - **SIZE:** Cung cấp tổng dung lượng hệ thống phát triển cho đến nay, và báo cáo kích thước mã mới và tái sử dụng / sửa đổi kích thước mã riêng. Điều này có thể ở mức độ dự án hoặc cấp thấp hơn như các dữ liệu hỗ trợ và hợp lý. Cho các ngôn ngữ không được hỗ trợ bởi các công cụ như lắp ráp mã, báo cáo số lượng các dòng vật lý riêng cho từng ngôn ngữ
  - **EFFORT:** Cung cấp tích lũy nhân viên giờ đã phát triển phần mềm cho mỗi dự án tại granularity tương tự như các thành phần kích thước.