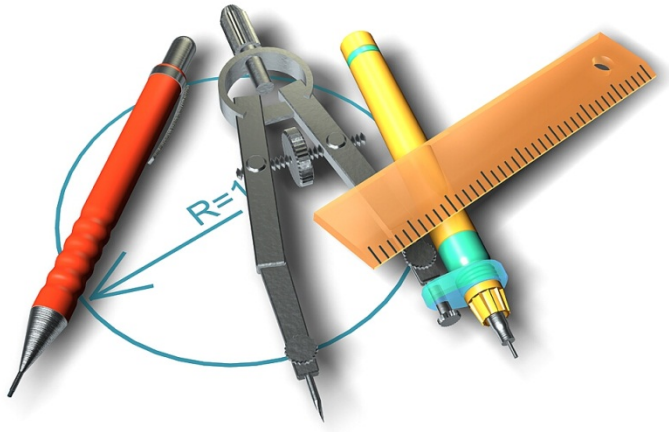
Effort and Schedule Estimation



Nguyễn V. Vũ

nvu@fit.hcmus.edu.vn

Outline

- Effort Estimation
  - Productivity Factor
  - Regression Technique
  - Wideband Delphi
- Schedule Estimation
  - Regression Technique
  - COCOMO II – Like Technique
- COCOMO II

Effort Estimation 1 – Productivity

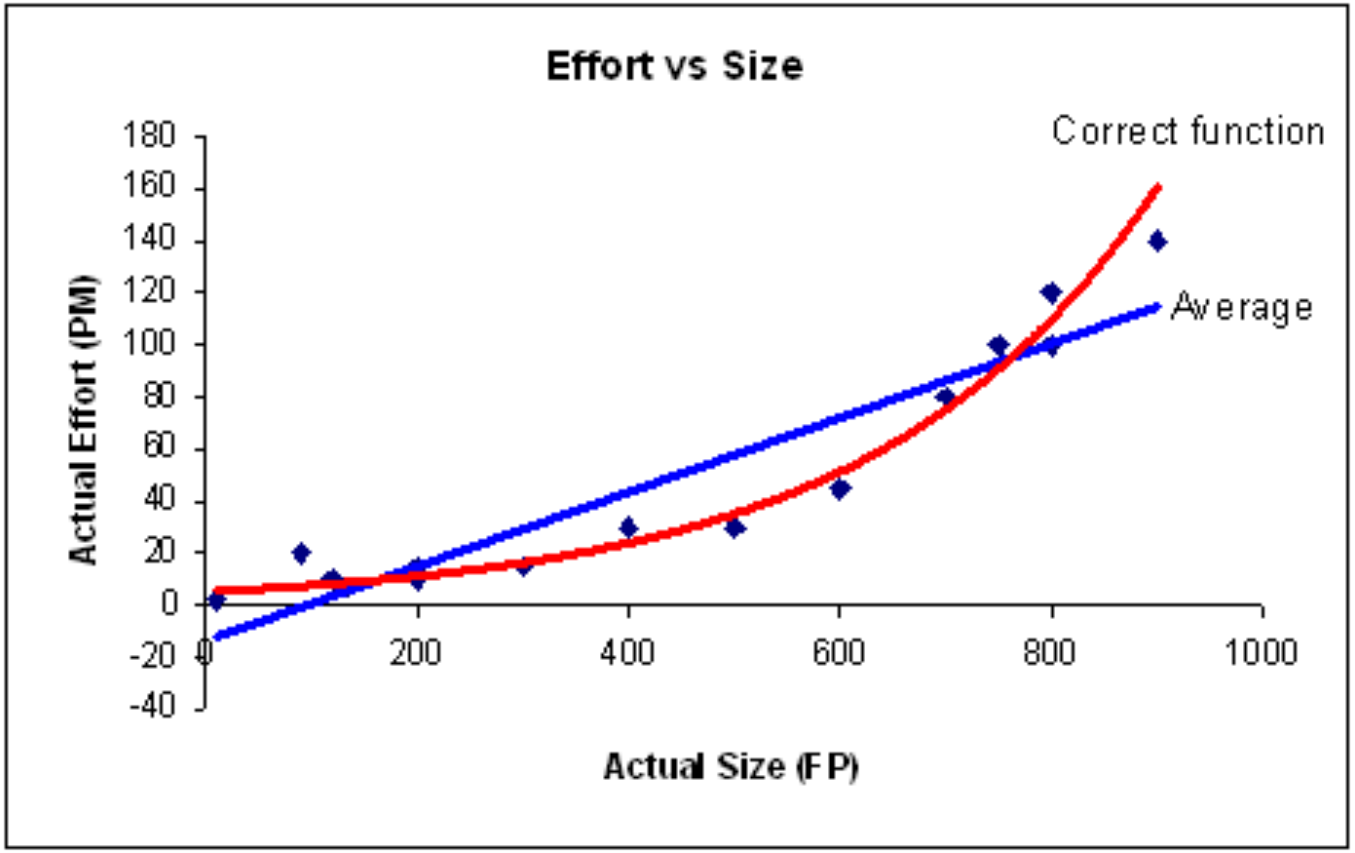
- Productivity is the ratio of actual project size and effort
  - E.g., 9 - 30 FP per person-month
- Productivity of *N* completed projects is averaged
- Productivity of projects of the same type or domain can be collected and used for estimating projects of that domain

Estimated effort = estimated size * productivity

Effort Estimation 1 – Productivity (cont)

- Advantages
  - Simple and easy to understand and use
  - Straightforward
  - Most popular
  - Does not require many completed projects and effort to build the model
- Disadvantages
  - High variance (low accuracy)
  - Does not account for the overhead of big projects
    - Big projects should have lower productivity than small projects (e.g., high communication overhead)
  - Difficult to calculate confidence interval

Effort and Size Correlation



Effort Estimation 2
Linear Regression Technique

- Linear regression is the most common technique to build estimation models
  - Used in parts of the COCOMO, SEER-SEM, and SLIM model building process
- Is a simple technique to calibrate models using local data
- Enables estimators to learn from completed projects
- Is more reliable and interpretable than using average productivity to estimate effort

Linear Regression Technique (cont)

- Requires at least 3 projects to run the regression
- Might result in wrong models as *coefficients* might be negative
  - For example, *Effort = 1.2 – 0.5 * Size*
- Like the productivity technique, it does not account for the overhead of big projects

Linear Regression Technique (cont)

- Simple effort regression
  - Given *N* completed projects which have *Effort* and *Size* as actual size and actual effort, respectively
  - Actual effort is expressed as follows

$$Effort = A + B * Size$$

  - *A* and *B* are estimated using historical data
  - *A* and *B* become constants of the resulted model after regression
  - For example, *Effort = 1.2 + 0.08 * Size*

Linear Regression Technique (cont)

- **Evaluation of Model Consistency**
  - **Evaluated using *correlation coefficient, R^2***

$$R = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

x_i – estimates

y_i – actuals

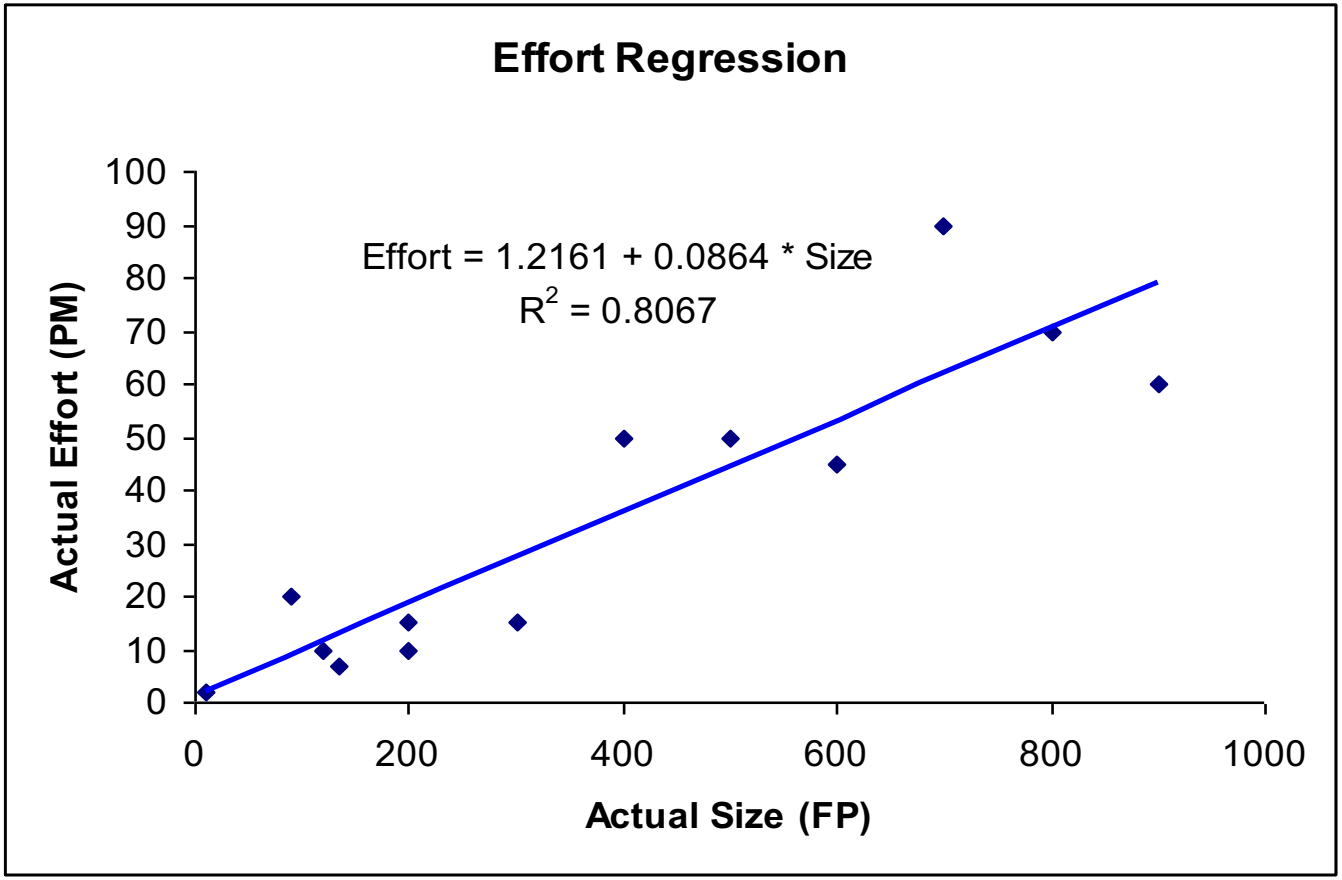
S_x – standard deviation of estimates

S_y – standard deviation of actuals

$0.9 \leq R^2 \leq 1.0$	Strong correlation. The model is consistent.
$0.7 \leq R^2 < 0.9$	High correlation. The model is consistent but needs to be reviewed and/or refined.
$0.0 \leq R^2 < 0.7$	The model should be reviewed and refined. It is not recommended using this model.

Linear Regression Technique (cont)

- Example



Effort Estimation 3 – Wideband Delphi

- Principles
  - Obtain estimates using expert-consensus
  - When the estimate is independently done by many experts, it is likely correct
  - Experts work on the same assumptions
- Tasks
  - Planning
  - Kick-off meeting
  - Individual preparation
  - Estimation meeting
  - Review results

Wideband Delphi – Procedure

1. Form estimation team consisting of experts, system author, and coordinator (4 – 6 people)
2. Author explains the system
3. Author and experts identify tasks and assumptions
4. Author and experts decide level of acceptance (e.g., variance of 20%)
5. Experts independently estimate tasks
6. Coordinator prepares estimation summary based on inputs from experts
7. Discuss the differences in estimates where variance is higher than the acceptance level
8. Continue steps 5 – 7 until high differences are resolved

Wideband Delphi (cont)

- Advantages
  - Suitable for providing rough budget estimates
  - Produce high reliable estimates if experts have experience in the domain of the project being estimated
  - No historical data is needed
- Disadvantages
  - Requires much time from participants. The process may take long time to complete
  - High variance in new domains (in which experts do not have experience)
  - Difficult to interpret the result

Planning Poker

- Game: deck of cards
- Moderator provides the estimation-item
- Participants privately choose appropriate card from deck
- Iterative
  - repeat if having divergence
  - stop when converged
- Ensures everyone participates
- Useful for estimation in agile projects

Planning Poker (cont'd)

- Các bước thực hiện
  1. Chọn 1 chức năng đại diện, gán 1 kích thước cho chức năng đó (2)
  2. Chọn chức năng, thảo luận, mỗi người ước lượng độc lập (chọn 1 con số trong Fibonacci)
  3. Tất cả mọi người cho biết ước lượng của mình
  4. Điều phối viên, tổng hợp kết quả, tính trung bình, xem các ước lượng có độ hội tụ hay không -> nếu có thì có kết quả ước lượng -> ước lượng chức năng kế tiếp (bước 2)
  5. Nếu không, thảo luận lại, người ước lượng cao hay thấp nhiều so với trung bình của nhóm cần giải thích
  6. Quay lại bước 2.

Exercise

- Form groups of five to practice Planning poker
- Estimating a mobile app for taking notes
  - Register an account
  - Create a note (3): ideal hour, story point
  - Update a note
  - Delete a note
  - Send a note via SMS
  - Send a note via an account
  - Synch notes with other devices

Outline

- Effort Estimation
  - Productivity Factor
  - Regression Technique
  - Wideband Delphi
- Schedule Estimation
  - Regression Technique
  - COCOMO II – Like Technique
- COCOMO II

Schedule Estimation

- Schedule estimation is the least investigated area in software cost estimation
  - It is partly due to high variance in actual schedule implemented by projects
  - Clients tend to compromise schedule with effort to shorten the time-to-market
  - Schedule cannot be always traded off with effort
    - You cannot ask a women to deliver a baby in less than 9 months
- There must be shortest possible schedule for a specific project

Schedule Estimation

- Methods
  - Method 1: $$Schedule = 5 * \sqrt[3]{KSLOC}$$
    - E.g., KSLOC = 10k, schedule = 5 * 2.15 ~ 10.8 mo
    - “5” can be set by organizations
  - Method 2: regression method

$$Schedule = A + B * Effort$$

    - Use the regression form introduced for effort estimation
    - Schedule as the response and effort as the predictor variable

Schedule Estimation (cont)

- Methods (cont)
  - Method 3: COCOMO-derived method

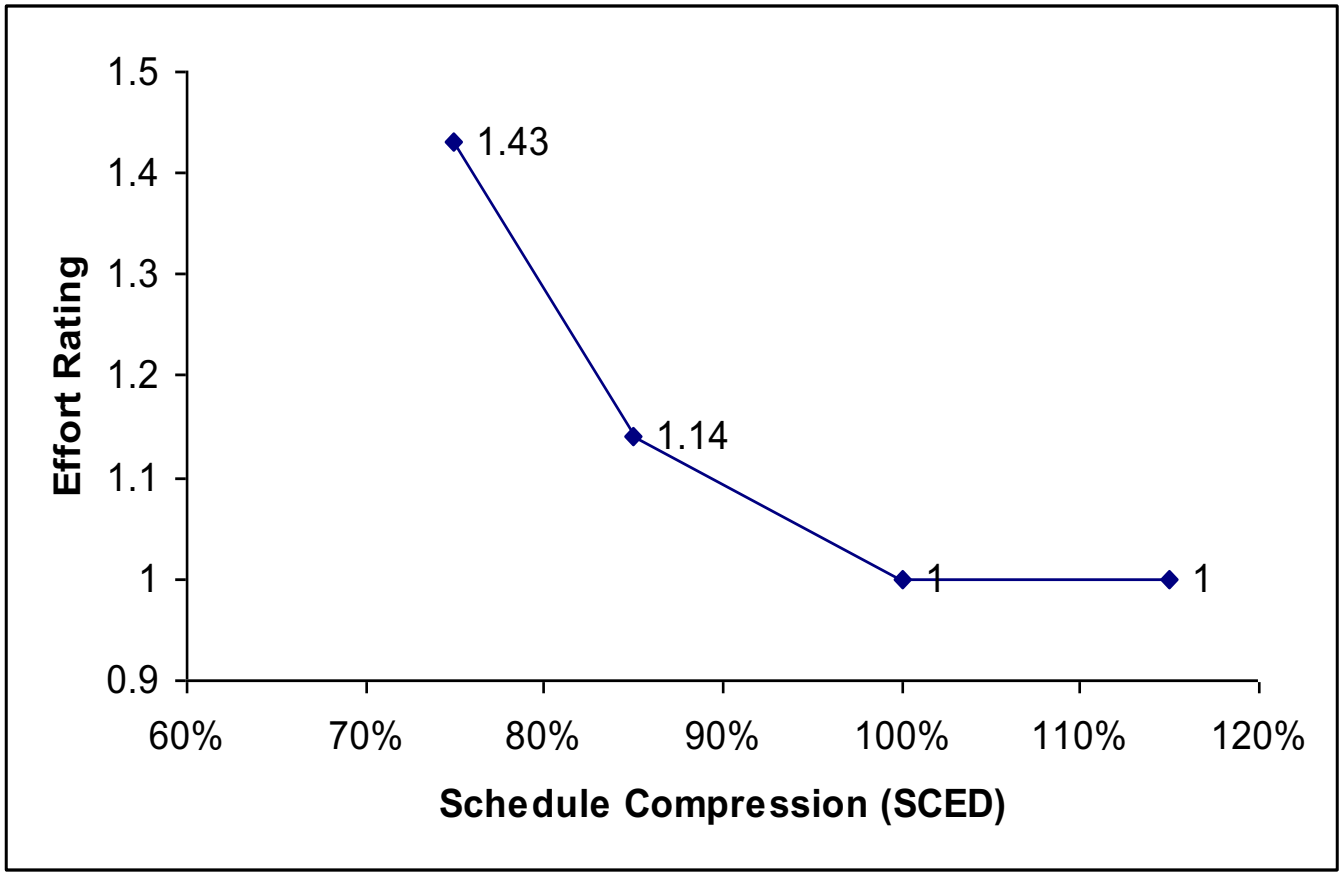
$$Schedule = P * \sqrt[3]{PM}$$

Where,

- P is the constant calculated from historical data
- P = Average of ($P_1 + P_2 + \ldots + P_n$)
- $P_i = Schedule_i / \sqrt[3]{PM_i}$

Schedule Compression

- COCOMO II's schedule compression



Schedule Compression (cont)

- COCOMO II's schedule compression
  - Effort increases 143% if schedule is compressed 75%
  - Effort increases 114% if schedule is compressed 85%
  - Stretching schedule does not affect effort
  - Use this pattern to convert between effort and schedule when schedule is compressed

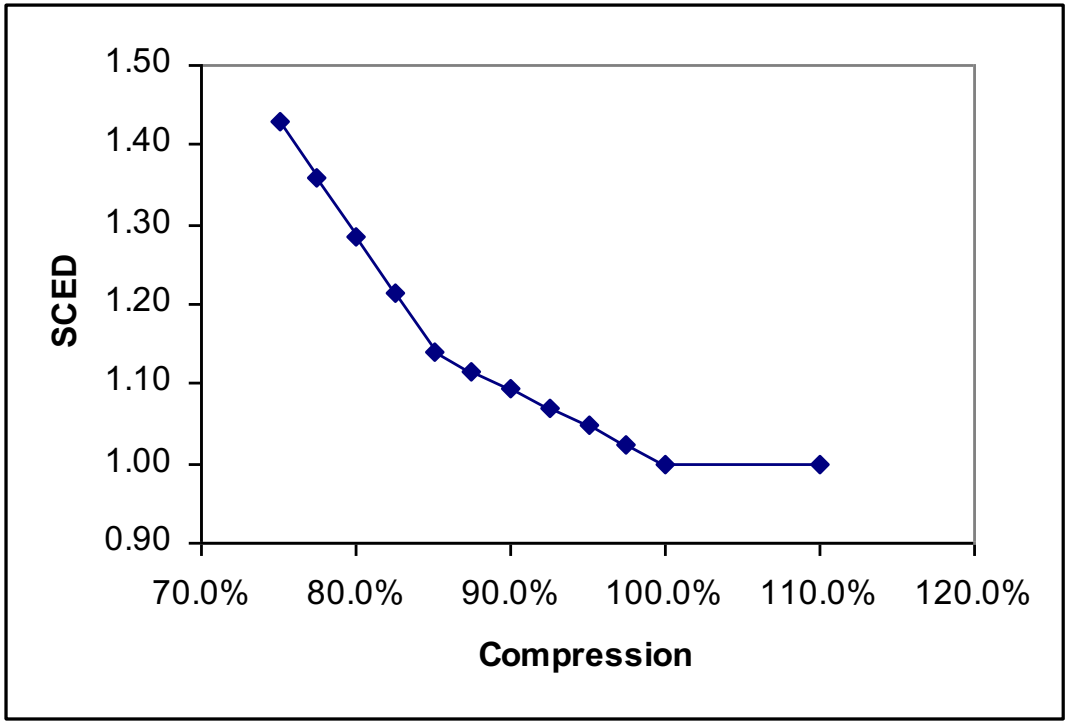
$$PM_c = SCED * PM$$

  - Where,
    - PM_c is effort after compressing normal schedule
    - PM is effort with normal schedule

Schedule Compression (cont)

$$PM_c = SCED * PM$$

Compression	SCED
75.0%	**1.43**
77.5%	**1.36**
80.0%	**1.29**
82.5%	**1.21**
85.0%	**1.14**
87.5%	**1.12**
90.0%	**1.09**
92.5%	**1.07**
95.0%	**1.05**
97.5%	**1.02**
100%	**1.00**
120.0%	**1.00**



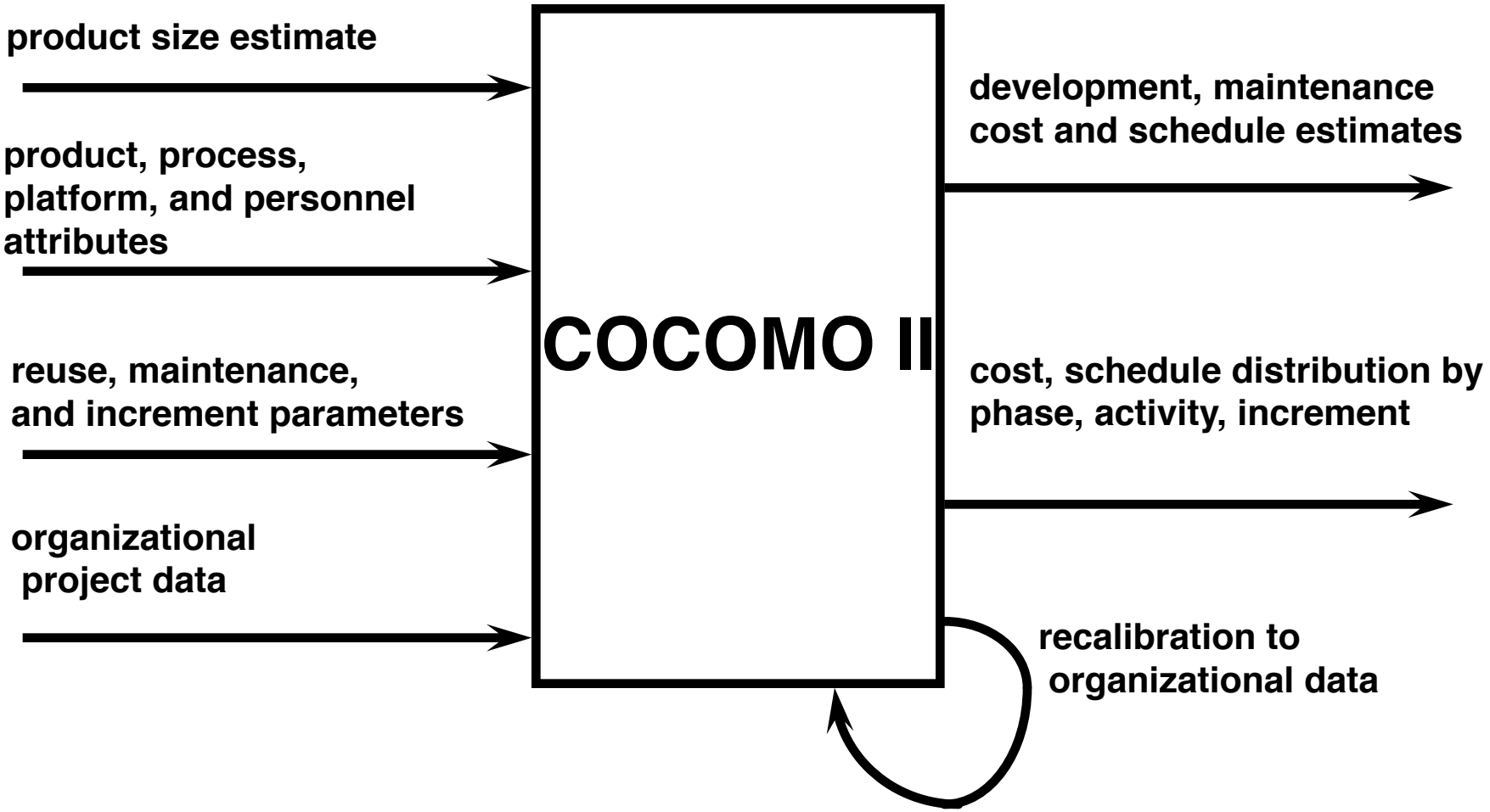
Outline

- Effort Estimation
  - Productivity Factor
  - Regression Technique
  - Wideband Delphi
- Schedule Estimation
  - Regression Technique
  - COCOMO II – Like Technique
- COCOMO II

COCOMO II Overview

- COnstructive COst MOdel (COCOMO) was invented in 1981 by Dr. Barry Boehm at University of Southern California
- COCOMO II was developed in mid-1990s to adapt to new changes in current and future practices
- COCOMO II model is one of the most widely-used cost models in the industry, especially by US DoD contractors (Boeing, Northrop Grumman, Aerospace, TRW, etc.)
- COCOMO is actively supported by Dr. Boehm's teams at Center for Software Engineering, USC:
  - Annual international conference in COCOMO
  - Emerging extensions are published regularly

COCOMO Black Box Model



Major COCOMO II Features

- Multi-model coverage of different development sectors
- Variable-granularity cost model inputs
- Flexibility in size inputs
  - SLOCS
  - function points
  - application points
  - other (use cases ...)
- Range vs. point estimates per funnel chart

COCOMO II Effort Formula

$$PM = A \times Size^{E} \times \prod_{i=1}^{N} EM_i$$

Where $E = B + 0.01 \times \sum_{j=1}^{5} SF_j$

PM – Effort in person-month

Size – Size of projects or modules

A – Constant, currently *A* = 2.94

B – Constant, currently *B* = .91

SF_j – Five scale factors, determined by estimators

EM_i – Effort multipliers, determined by estimators

**Note: A and B can be calibrated*

COCOMO II Schedule Formula

$$TDEV = C \times PM^{F} \times SCED\%$$

Where $F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$

$= D + 0.2 \times (E - B)$

TDEV – Time to develop, calendar month

C – Constant, currently *C* = 3.67

D – Constant, currently *D* = .28

SCED% – Percentage of schedule compression

**Note: C and D can be calibrated*

COCOMO II Cost Drivers

- Cost drivers are variables considered to be source of cost variations
- Classified into two categories:
  - Scale factors: 5
  - Effort multipliers: 17 for Post-Architecture Model, 7 for Early-Design Model
- Each cost driver is rated into six levels (some cost drivers have 4 or 5)
  - Very Low (VLO)
  - Low (LO)
  - Nominal (NOM)
  - High (HI)
  - Very High (VHI)
  - Extra High (XHI)

Scale Factors

- Precedentedness (PREC)
  - Degree to which system is new and past experience applies
- Development Flexibility (FLEX)
  - Need to conform with specified requirements
- Architecture/Risk Resolution (RESL)
  - Degree of design thoroughness and risk elimination
- Team Cohesion (TEAM)
  - Need to synchronize stakeholders and minimize conflict
- Process Maturity (PMAT)
  - SEI CMM process maturity rating

Precedentedness (PREC)

- This scale factor represents whether the project team has developed similar projects.
- PREC rating is based on the following features
  - A - Organizational understanding of product objectives
  - B - Experience in working with related software systems
  - C - Concurrent development of associated new hardware and operational procedures
  - D - Need for innovative data processing architectures, algorithms

PREC Rating

Rating Level	Value	Feature Level
Very Low	Thoroughly unprecedented (6.20)	General *A* Moderate *B* Extensive *C* Considerable *D*
Low	Largely unprecedented (4.96)	
Nominal	Somewhat unprecedented (3.72)	Considerable *A* Considerable *B* Moderate *C* Some *D*
High	Generally familiar (2.48)	
Very High	Largely familiar (1.24)	Thorough *A* Extensive *B* Some *C* Minimal *D*
Extra High	Thoroughly familiar (0.00)	

Development Flexibility (FLEX)

- FLEX specifies the degree of flexibility of the development (establishing requirements, architectures, design, implementation, testing) to which the team should conform.
- The more rigorous conformance requirements are, the less flexibility the development can have.
- FLEX rating is based on the following features
  - A – Need for software conformance with pre-established requirements
  - B – Need for software conformance with external interface specifications
  - C – Combination of features above with premium on early completion

FLEX Rating Levels

Rating Level	Value	Feature Level
Very Low (5. 07)	Rigorous adherence to specifications	Full *A* Full *B* High *C*
Low (4.05)	Occasional relaxation	
Nominal (3.04)	Some relaxation	Considerable *A* Considerable *B* Medium *C*
High (2.03)	General conformity to specifications	
Very High (1.01)	Some conformity to specifications	Basic *A* Basic *B* Low *C*
Extra High (0.00)	Conformity to general goals	

Risk Resolution (RESL)

- RESL scale factor measures the level of risk eliminated at the time of estimation.

Rating Level	Descriptor
Very Low (7.07)	20% of significant risks eliminated
Low (5.65)	40% of significant risks eliminated
Nominal (4.24)	60% of significant risks eliminated
High (2.83)	75% of significant risks eliminated
Very High (1.41)	90% of significant risks eliminated
Extra High (0.00)	100% of significant risks eliminated

Team Cohesion (TEAM)

- TEAM scale factor accounts for the sources of project turbulence and entropy due to difficulties in interaction among the project's stakeholders: onshore team, offshore team, clients, others
- TEAM is the subjective weighted average of
  - A – Consistency of stakeholder objectives and cultures
  - B – Ability, willingness of stakeholders to accommodate other stakeholders' objectives
  - C – Experience of stakeholders in operating as a team
  - D – Stakeholder teambuilding to achieve shared vision and commitments.

TEAM Rating Levels

Rating Level	Value	Characteristic Level
Very Low (5.48)	Very difficult interactions	Little *A, B* None *C, D*
Low (4.38)	Some difficult interactions	Some *A, B* Little C, D
Nominal (3.29)	Basically cooperative interactions	Basic *A, B* Little *C, D*
High (2.19)	Largely cooperative	Considerable *A, B* Basic *C, D*
Very High (1.10)	Highly cooperative	Strong *A, B* Considerable *C, D*
Extra High (0.00)	Seamless interactions	Full *A, B* Extensive *C, D*

Process Maturity (PMAT)

- PMAT scale factor measures the process maturity of the project.
- PMAT rating is based on SEI's CMM/CMMI maturity levels.
- PMAT should be rated on process followed in the project, the not company

Rating Level	Maturity Level/Value
Very Low (7.80)	CMM/CMMI Level 1 (lower half)
Low (6.24)	CMM/CMMI Level 1 (upper half)
Nominal (4.68)	CMM/CMMI Level 2
High (3.12)	CMM/CMMI Level 3
Very High (1.56)	CMM/CMMI Level 4
Extra High (0.00)	CMM/CMMI Level 5

Cost Drivers (EM)

- 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)
  - Programmer 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)

Required Software Reliability (RELY)

- Accounts for the reliability requirements that the software is required to satisfy
- This reflects the type of systems to be developed (finance, defense, mission, flight control, etc.)

Rating Level	Value
Very Low (0.82)	Slight inconvenience
Low (0.92)	Low, easily recoverable losses
Nominal (1.00)	Moderate, easily recoverable losses
High (1.10)	High financial loss
Very High (1.26)	Risk human life
Extra High	N/A

Product Factors (Cont)

- Data Base Size (DATA)
  - Captures the effect large data requirements have on development to generate test data that will be used to exercise the program.
  - Calculate the data/program size ratio (D/P):

$$\frac{D}{P} = \frac{DataBaseSize(Bytes)}{ProgramSize(SLOC)}$$

	Very Low	Low	Nominal	High	Very High	Extra High
DATA		DB bytes/ Pgm SLOC < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P > 1000	
Values		0.9	1.0	1.14	1.28	

Product Factors (Cont)

- Product Complexity (CPLX)
  - Complexity is divided into five areas:
    - control operations,
    - computational operations,
    - device-dependent operations,
    - data management operations, and
    - user interface management operations.
  - Combine these areas to determine ratings for CPLX

	Very Low	Low	Nominal	High	Very High	Extra High
Description	Very simple module	Simple module	Average complexity	Moderately complex	Complex	Highly complex
Values	0.73	0.87	1.0	1.17	1.34	1.74

Product Factors (Cont)

- Required Reusability (RUSE)
  - Accounts for the additional effort needed to construct components intended for reuse

	Very Low	Low	Nominal	High	Very High	Extra High
RUSE		none	across project	across program	across product line	across multiple product lines
Values		0.95	1.0	1.07	1.15	1.24

- Documentation match to life-cycle needs (DOCU)
  - What is the suitability of the project's documentation to its life-cycle needs.

	Very Low	Low	Nominal	High	Very High	Extra High
DOCU	Many life-cycle needs uncovered	Some life-cycle needs uncovered	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
Values	0.81	0.91	1.0	1.11	1.23	

Platform Factors

- Platform
  - Refers to the target-machine complex of hardware and infrastructure software
- Execution Time Constraint (TIME)
  - Measures the constraint imposed upon a system in terms of the percentage of available execution time (CPU) expected to be used by the system

	Very Low	Low	Nominal	High	Very High	Extra High
TIME			≤ 50% use of available execution time	70%	85%	95%
Values			1.0	1.11	1.29	1.63

Platform Factors (Cont)

- 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%
Values			1.0	1.05	1.17	1.46

- Platform Volatility (PVOL)
  - Assesses the volatility of the platform (frequency of changes in the platform used).

	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	
Values		0.87	1.0	1.15	1.30	

Personnel Factors

- Analyst Capability (ACAP)
  - Analysts work on requirements, high level design and detailed design. Consider analysis and design ability, efficiency and thoroughness.

	Very Low	Low	Nominal	High	Very High	Extra High
ACAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Values	1.42	1.19	1.0	0.85	0.71	

- Programmer Capability (PCAP)
  - Evaluate the capability of the programmers as a team rather than as individuals

	Very Low	Low	Nominal	High	Very High	Extra High
PCAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Values	1.34	1.15	1.0	0.88	0.76	

Personnel Factors (Cont)

- Applications Experience (AEXP)
  - Assess the project team's equivalent level of experience with this type of application

	Very Low	Low	Nominal	High	Very High	Extra High
AEXP	≤ 2 months	6 months	1 year	3 years	6 years	
Values	1.22	1.10	1.0	0.88	0.81	

- Platform Experience (PEXP)
  - Assess the project team's equivalent level of experience with this platform including the OS, database, networking, etc.

	Very Low	Low	Nominal	High	Very High	Extra High
PEXP	≤ 2 months	6 months	1 year	3 years	6 year	
Values	1.19	1.09	1.0	0.91	0.85	

Personnel Factors (Cont)

- Language and Tool Experience (LTEX)
  - Measures the level of programming language and software tool experience of the project team.

	Very Low	Low	Nominal	High	Very High	Extra High
LTEX	≤ 2 months	6 months	1 year	3 years	6 years	
Values	1.20	1.09	1.0	0.91	0.84	

- Personnel Continuity (PCON)
  - The scale for PCON is in terms of the project's annual personnel turnover.

	Very Low	Low	Nominal	High	Very High	Extra High
PCON	48% / year	24% / year	12% / year	6% / year	3% / year	
Values	1.29	1.12	1.0	0.90	0.81	

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	
1.17	1.09	1.0	0.90	0.78	

Project Factors (Cont)

- Multisite Development (SITE)
  - Assess and average two factors: site collocation and communication support.

	Very Low	Low	Nominal	High	Very High	Extra High
SITE: Collocation	International	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro. area	Same building or complex	Fully collocated
SITE: Communications	Some phone, mail	Individual phone, FAX	Narrowband email	Wideband electronic communication	Wideband elect. comm, occasional video conf.	Interactive multimedia
Values —	1.22	1.09	1.0	0.93	0.86	

- Required Development Schedule (SCED)
  - Measure the imposed schedule constraint: % of schedule stretch-out or acceleration with respect to a nominal schedule

	Very Low	Low	Nominal	High	Very High	Extra High
SCED	75% of nominal	85%	100%	130%	160%	
Values	1.43	1.14	1.0	1.0	1.0	

Summary

- Effort Estimation Methods
  - Productivity Factor

$$\textbf{Estimated effort = estimated size * productivity}$$

  - Linear Regression

$$\textit{Effort} = A + B\ \textit{Size}$$

  - Wideband Delphi
  - COCOMO II

References

- COCOMO Model Manual and other resources http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
- Model manual and rating values: http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf
- B. Boehm, et al., “Software Cost Estimation with COCOMO II,” Prentice Hall, 2000.

Summary (cont)

- Schedule Estimation Methods
  - Method 1 – using general formula

$$Schedule = 5 * \sqrt[3]{KSLOC}$$

  - Linear Regression

$$Schedule = A + B\ Effort$$

  - COCOMO-derived method

$$Schedule = P * \sqrt[3]{PM}$$

Summary (cont)

- Schedule Compression

$$PM_c = SCED * PM$$

- Recommendations
  - Companies should decide the most suitable and convenient methods to be applied
  - Selected methods should allow companies to calibrate using historical data
  - Estimation processes and methods should be continuously evaluated and refined