A model for predicting effort

COCOMO 2.0 approach

Object Point

- Object points were selected for size input.

Object Points

- To compute object points, an initial size measure is generated by counting number of screens, etc.
- Next each object is classified as simple, medium and difficult as per given guidelines.

Object Point complexity levels
For Screens

Number of views contained	Number and source of data tables		
	Total < 4 (<2 server, <2 client)	Total < 8 (2-3 server, 3-5 client)	Total 8+ (>3 server, >5 client)
< 3	Simple	Simple	Medium
3 – 7	Simple	Medium	Difficult
8+	Medium	Difficult	Difficult

Object Point complexity levels
For Reports

Number of sections contained	Number and source of data tables		
	Total < 4 (<2 server, <2 client)	Total < 8 (2-3 server, 3-5 client)	Total 8+ (>3 server, >5 client)
< 2	Simple	Simple	Medium
2 or 3	Simple	Medium	Difficult
> 3	Medium	Difficult	Difficult

Object Point Analysis – Complexity Weighting

- The number in each cell is weighted according to the given table

	Complexity		
Type of object	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3GL component	N/A	N/A	10

Object Point Analysis

- The weight reflects the relative effort required to implement an instance of that complexity level.
- Then weighted instances are summed to yield a single object point number.
- Then reuse is taken into account.
- Assuming that r% of objects will be reused from previous projects, the number of new object points is calculated to be

new object points = (object points) x (100-r)/100

Object Point Analysis

- To use this number for effort estimation, COCOMO 2.0 determines a productivity rate (i.e. new object points per person month) from a table based on developer experience and capability.

Example

- Suppose 840 object points are computed from a system specification and 20% can be supplied by existing components the

new object points = 840 x (100-20)/100 = 672 object points

Object Points effort Conversion table

Developer's experience and Capability / ICASE maturity and capability	Very low	Low	Nominal	High	Very high
PROD: Productivity Object-point per person-month	4	7	13	25	50

- In our example, productivity is normal, PROD is 13

Effort in p-m = NOP / PROD

- In our example, Effort = 672 / 13 = 52 months

Another Example

- Assessment of a software system shows that the system includes:
  - 6 screens: 2 simple + 3 medium + 1 difficult
  - 3 reports: 2 medium + 1 difficult
  - 2 3GL components
- 30 % of the objects could be supplied from previously developed components
- Productivity is high
- Compute the estimated effort PM 'Person-months' needed to develop the system.

Object Point estimation Solution

- Object Counts

2 simple screens	x1 = 2
3 medium screens	x2 = 6
1 difficult screen	x3 = 3
2 medium reports	x5 = 10
1 difficult report	x8 = 8
2 3 GL components	x10 = 20
Number of object points	**49**

Object Point estimation Solution

Adjusted NOP 'New NOP'

= NOP * (1 -% reuse/ 100)

= 49 * (1-(30/100))

= (34.3)

= 35

For high productivity (metric table): PROD = 25 OP/P-M

Estimated effort Person-Month

= **Adjusted NOP/ PROD**

= 35/ 25

= 1.4P-M

COCOMO II – without object points

Effort = Constant × (Size)$^{\text{scale factor}}$ x Effort Multiplier

- Effort in terms of person-months
- Constant: 2.45 in 1998
- Size: Estimated Size in KLOC
- Scale Factor: combined process factors
- Effort Multiplier (EM): combined effort factors

System to be built

- An airline sales system is to be built in C:
  - Back-end database server has already been built.
- We will use object point estimation technique for high level estimates and FP for detailed estimates.

Object Point Analysis

- Application will have 3 screens and will produce 1 report
  - A booking screen: records a new sale booking
  - A pricing screen: shows the rate for each day and each flight
  - An availability screen: shows available flights
  - A sales report: shows total sale figures for the month and year, and compares figures with previous months and years

Rating of system

- Booking screen:
  - Needs 3 data tables (customer info, customer history table, available seats)
  - Only 1 view of the screen is enough. So, the booking screen is classified as simple.
- Similarly, the levels of difficulty of the pricing screen, the availability screen and the sales report are classified as simple, medium and medium, respectively. There is no 3GL component.

Rating results

Name	Object	Complexity	Weight
Booking	Screen	Simple	1
Pricing	Screen	Simple	1
Availability	Screen	Medium	2
Sales	Report	Medium	5
		Total	**9**

Rating results

- Assessment of the developers and the environment shows:
  - The developers' experience is very low (4)
  - The CASE tool is low (7). So, we have a productivity rate of 5.5.
- According to COCOMO II, the project requires approx. **1.64 (= 9/5.5) person-months**

Function point table

Number of FPs	Complexity		
External user type	Low	Average	High
External input type	3	4	5
External output type	4	5	7
Logical internal file type	7	10	15
External interface file type	5	7	10
External inquiry type	3	4	6

Function Point Estimation (FP->KLOC)

Name	External User Types	Complexity	FP
Booking	External output type	Low	4
Pricing	External inquiry type	Low	3
Availability	External inquiry type	Medium	4
Sales	External output type	Medium	5
		Total	16

FP->LOC

- Total function points = 16
- Published figures for C show that:
  - 1 FP = 128 LOC in C
- Estimated Size
  - 16 * 128 = 2048 = 2 KLOC

Scale Factor Estimation

Name	Very low (0.05)	Low (0.04)	Nominal (0.03)	High (0.02)	Very High (0.01)	Extra High (0.00)	Assessment	Value
Precedentedness	Thoroughly unprecedented	Largely unprecedented	Somewhat unprecedented	Generally familiar	Largely familiar	Thoroughly familiar	**Very high**	**0.01**
Flexibility	Rigorous	Occasional relaxation	Some relaxation	General conformity	Some conformity	General goals	**Very high**	**0.01**
Significant risks eliminated	Little (20%)	Some (40%)	Often (60%)	Generally (75%)	Mostly (90%)	Full (100%)	**Nominal**	**0.03**
Team interaction process	Very difficult	Some difficult	Basically cooperative	Largely cooperative	Highly cooperative	Seamless interactions	**High**	**0.02**
Process maturity	Level 1	Level 2	Level 2+	Level 3	Level 4	Level 5	**Low**	**0.04**
							Add	**1.01**
							Total	**1.13**

Effort Adjustment Factors (EAF)

Effort = $2.45 \times (2.048)^{1.13} \times 0.4826 = 2.66$ person-months

Identifier	Name	Ranges (VL – EH)	Assessment VL/L/N/H/VH/EH	Values
RCPX	product Reliability and ComPleXity	0.5 – 1.5	low	0.75
RUSE	required reusability	0.5 – 1.5	nominal	1.0
PDIF	Platform DIFficulty	0.5 – 1.5	high	1.1
PERS	PERSonnel capability	1.5 – 0.5	high	0.75
PREX	PeRsonnel EXperience	1.5 – 0.5	very high	0.65
FCIL	FaCILities available	1.5 – 0.5	nomial	1.0
SCED	SChEDule pressure	1.5 – 0.5	low	1.2
			Product	0.4826

Function Point table

- External (user) inputs: input transactions that update internal files
- External (user) outputs: reports, error messages
- User interactions: inquiries
- Logical internal files used by the system:

Example a purchase order logical file composed of 2 physical files/tables Purchase_Order and Purchase_Order_Item

- External interfaces: files shared with other systems

Effort = Constant × (Size)$^{\text{scale factor}}$ x Effort Multiplier

- 'Constant' is an organization-dependent constant.
- 'Scale factor' reflects the disproportionate effort for large projects.
- Effort multiplier M is a multiplier reflecting product, process and people attributes.

- Scale factor varies from 1.1 to 1.24 depending on novelty of the project, development flexibility, risk management approaches and the process

Exponent scale factors

- Precedentedness
- Development flexibility
- Architecture/risk resolution
- Team cohesion
- Process maturity

The exponent term

- This depends on 5 scale factors.
- Their sum/100 is added to 1.01

Example for exponent term

- **Precedentedness** - new project - 4
- **Development flexibility** - no client involvement - Very high – 1
- **Architecture/risk resolution** - No risk analysis – Very Low – 5
- **Team cohesion** - new team - nominal – 3
- **Process maturity** - some control - nominal – 3

- Scale factor is therefore 1.17

Multipliers

- **Product attributes** - concerned with required characteristics of the software product being developed
- **Computer attributes** - constraints imposed on the software by the hardware platform
- **Personnel attributes** -multipliers that take the experience and capabilities of the people working on the project into account.
- **Project attributes** - concerned with the particular characteristics of the software development project.

Multipliers

- Multipliers reflect the capability of the developers, the non-functional requirements, the familiarity with the development platform, etc.

Multipliers

- RCPX - product reliability and complexity
- RUSE - the reuse required
- PDIF - platform difficulty
- PREX - personnel experience
- PERS - personnel capability
- SCED - required schedule
- FCIL - the team support facilities

$$M = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$$