



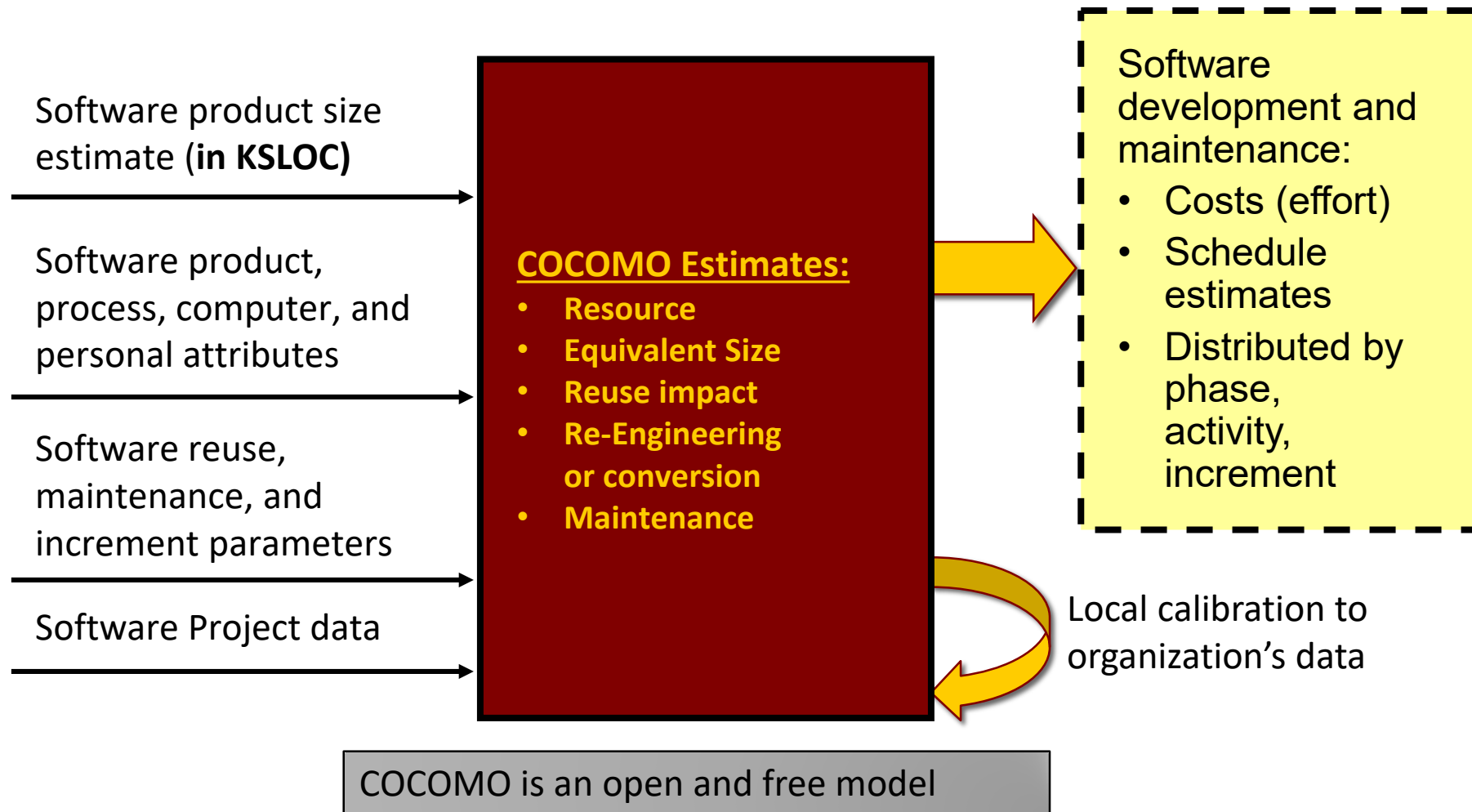
Calibrating COCOMO[®] II for Functional Size Metrics

ANANDI HIRA

BRAD CLARK, BARRY BOEHM

Background and Motivation for Workshop

COCOMO[®] II Model



Size Metrics' Level of Abstraction

Requirement Levels

Size Metrics

Summary Goals

Story Points

User Goals

Use Cases

Use Case Points (UCPs)

Sub-Functions

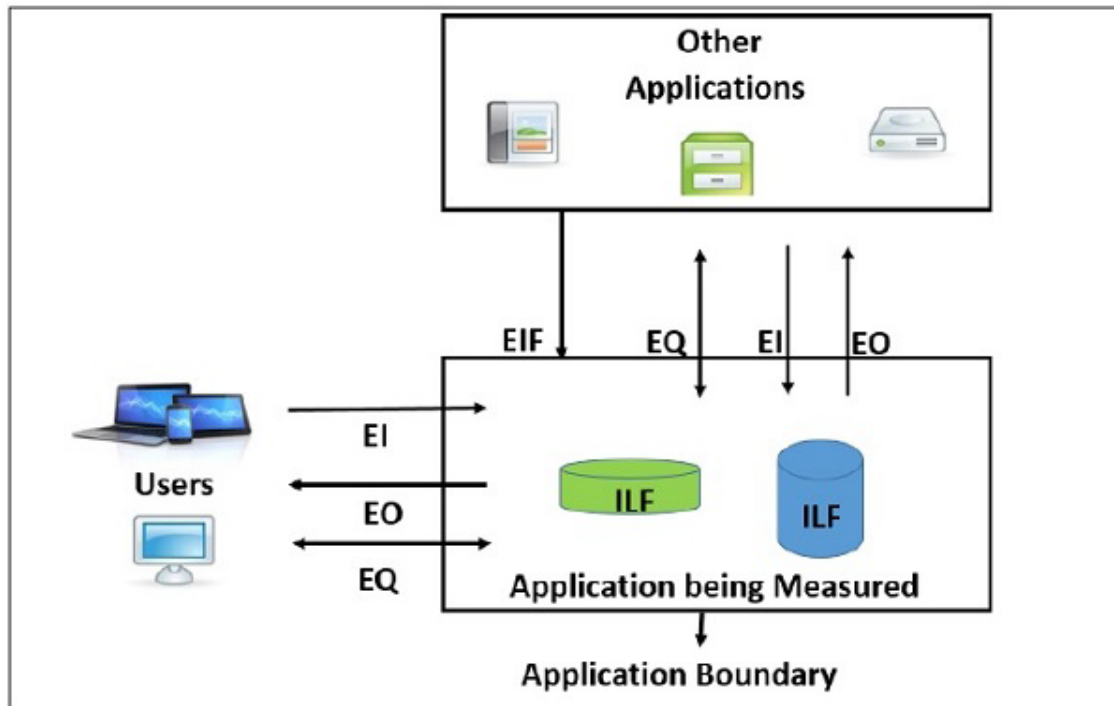
IFPUG Function Points (FPs)

COSMIC Function Points (CFPs)

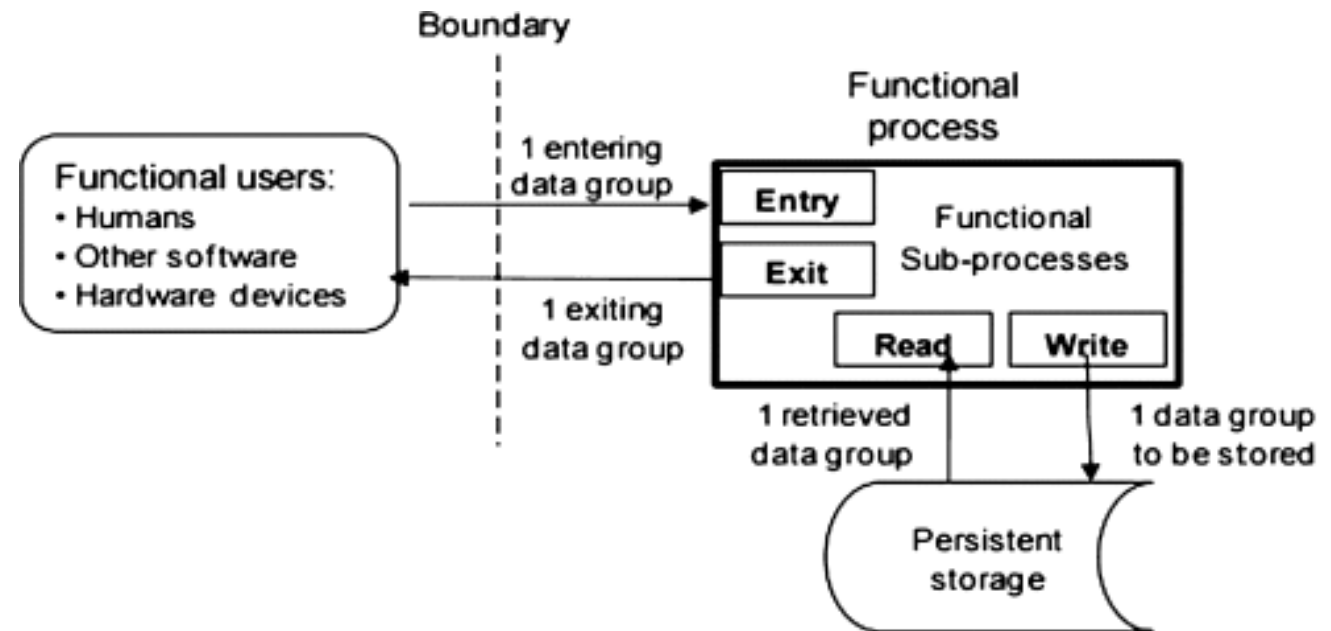
Source Lines of Code (SLOC)

2 Prominent Functional Size Methods

IFPUG SOFTWARE MODEL



COSMIC SOFTWARE MODEL



COCOMO[®] II Effort Model Format

$$PM = A \times Size^{\underbrace{(B + 0.1 \times (\sum SF))}_{\text{Exponent ranges from 0.9 to 1.2, with 1.0991 as default}}} \times \prod EM$$

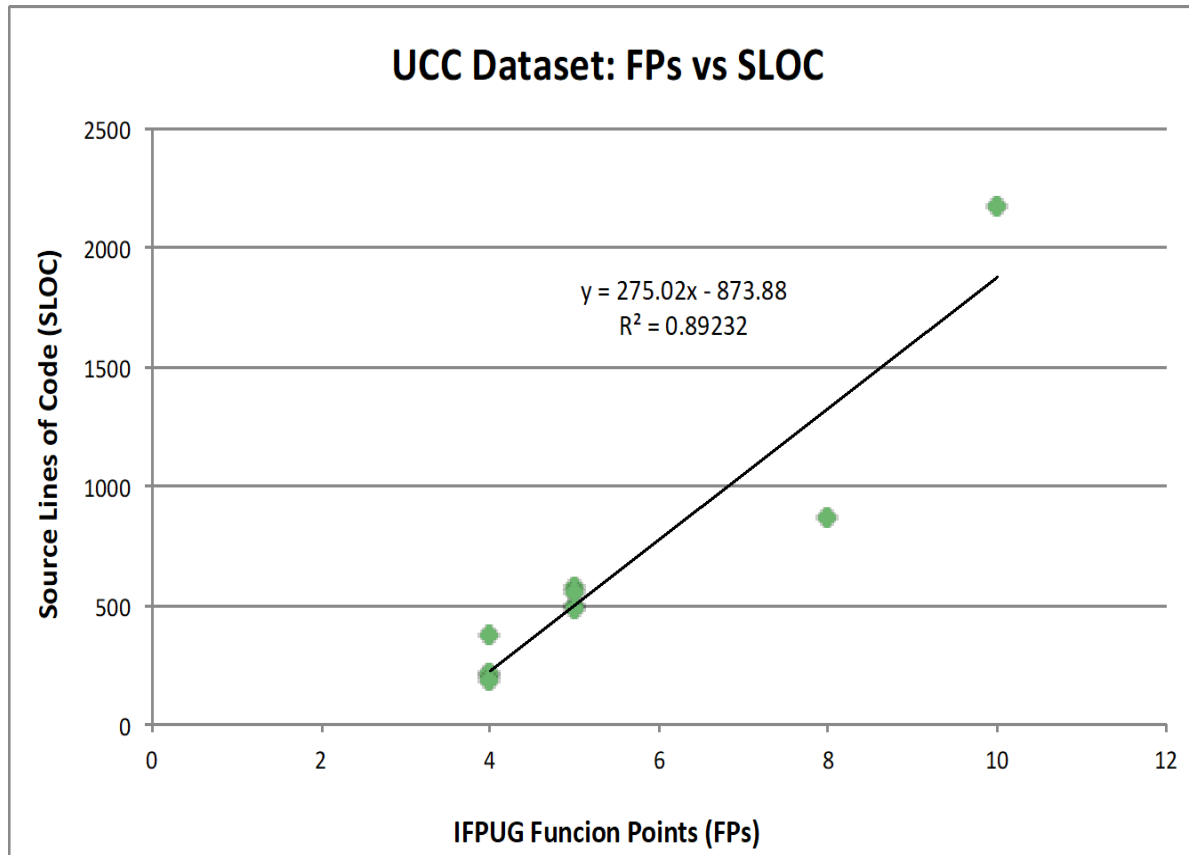
Exponent ranges from 0.9 to 1.2, with 1.0991 as default

Where

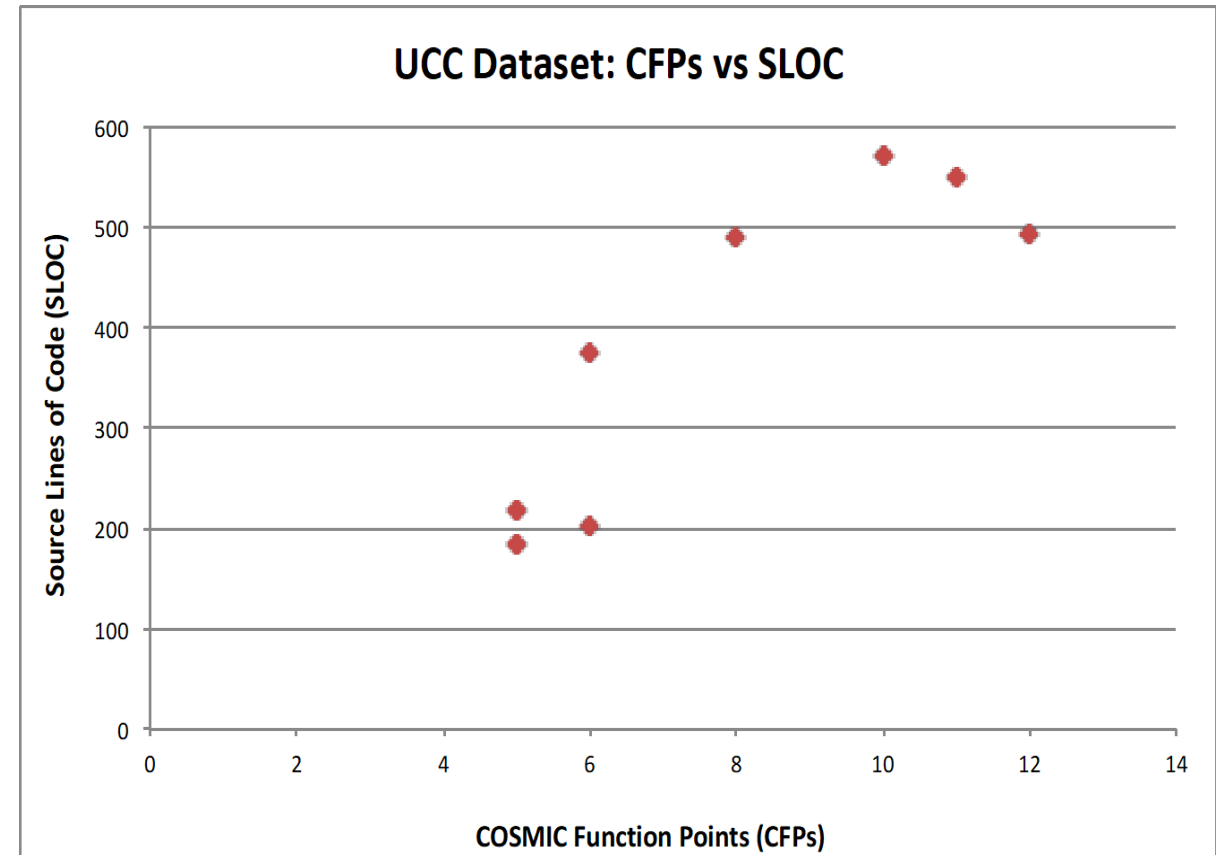
PM	=	Software development effort (in Person-months)
Size	=	Size in Thousand Equivalent Source Lines of Code (KESLOC)
A	=	Calibrated Productivity constant (ESLOC/PM)
B	=	Calibrated Exponent constant
SF	=	Scale Factors – have exponential effect
EM	=	Effort Multipliers – have multiplicative effect

Example FP and CFP vs SLOC (UCC Dataset)

IFPUG FUNCTION POINTS (FPS)



COSMIC FUNCTION POINTS (CFPS)



Workshop Summary

Attendees

1. David Seaver
2. John Kiser
3. Kammy Mann
4. Mike Konrad
5. Brad Clark
6. Anandi Hira

Objective/Goal

Get expert opinions on changes for:

- Scale Factors – how quickly effort grows with respect to size
 1. Precedentedness (PRED)
 2. Development Flexibility (FLEX)
 3. Team Cohesion (TEAM)
 4. Risk and Architecture Resolution (RESL)
 5. Process Maturity (PMAT)
- Effort Multipliers – if necessary
 - Perhaps Product drivers, such as Product Complexity (CPLX)?

Scale Factor Ratings

Each Scale Factor rating has an underlying value that is applied to COCOMO II's Size input

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF_j:	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF_j:	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
SF_j:	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF_j:	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
SF_j:	7.80	6.24	4.68	3.12	1.56	0.00

Project Examples for Discussion

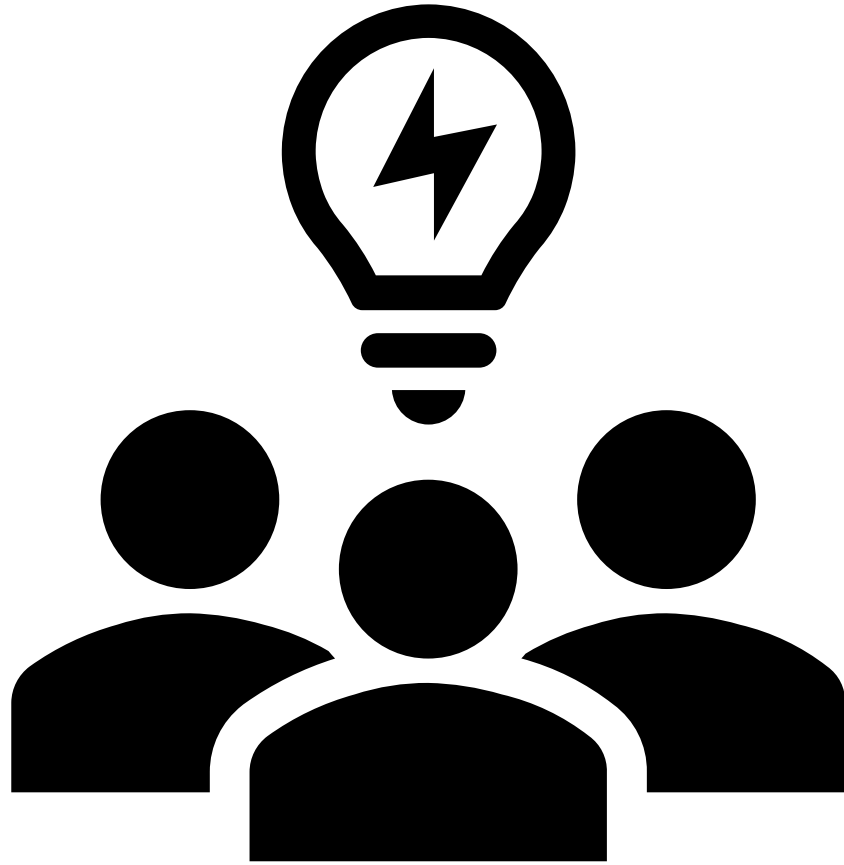
Small project

- 170 Function Points
- Approximately 9,010 lines of code for Java
- COCOMO II estimated 5,013 hours for 11.2 months for this development
- Each Scale Factors was set to the "Nominal" rating

Large project example

- 1,000 Function Points
- Approximately 53,000 lines of code for Java
- COCOMO II estimated 35,187 hours for 20.6 months for this development using lines of code
- Each Scale Factors was set to the "Nominal" rating

Workshop Question: *Do the underlying values for the Scale Factors change if Function Points are used as the Size measure instead of lines of code?*



Summary

Great discussion about:

- Differences between SLOC and functional size metrics
- Typical sizes of projects
- People/effort differences for scale factors

Conclusions

Asked for effort difference across total range of ratings

- Very Low to Extra High

Found that easier for people to think about difference between 1 level

- Example, Nominal to High

Good test run!



Next Steps

- Get in touch with me to participate:
 - anandihi@usc.edu
- Come to USC CSSE's COCOMO Forum
 - October 28-29
 - Topics: Costing Security Development, Software Size, Software Quality, etc.
 - <https://csse.usc.edu/new/event/2019-international-forum-on-cocomo-and-systems-software-cost-modeling>
- Data!