Calibrating COCOMO® II for Functional Size Metrics

Anandi Hira
University of Southern California

Joint Software and IT Cost Forum 2020

Motivation

COCOMO® II (Constructive Cost Model)

Objective/Goal:
Allow FPs and CFPs as size inputs

Software size estimate
(in SLOC (Source Lines of Code))

Effort Factors (project, product, environmental)

Software reuse, maintenance, and increment parameters

Software Project data

COCOMO Estimates:

- **Resource**
- **Equivalent Size**
- **Reuse impact**
- **Re-Engineering or conversion**
- **Maintenance**

Software development and maintenance:

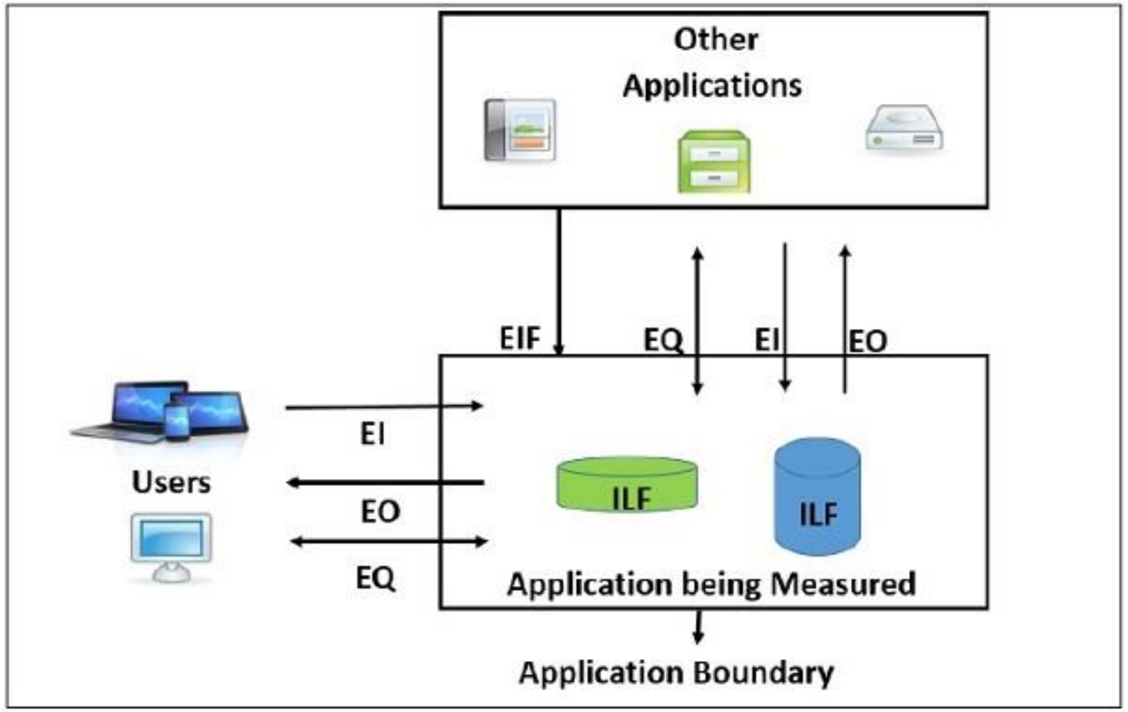
- Costs (effort)
- Schedule estimates
- Distributed by phase, activity, increment

Local calibration to organization's data

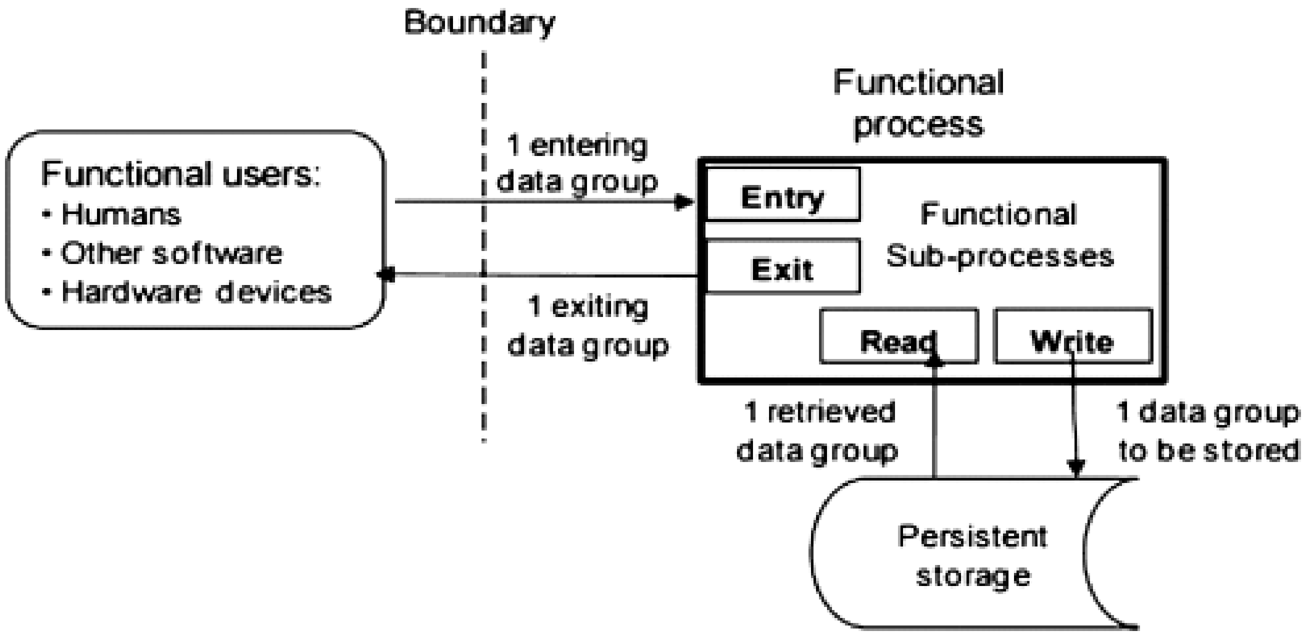
COCOMO® II is an open and free model

2 Prominent Functional Size Methods

IFPUG Software Model for Function Points (FPs)

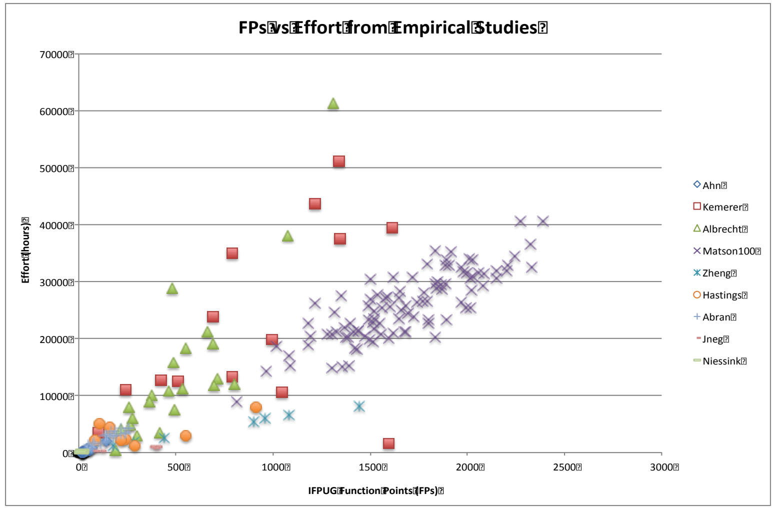


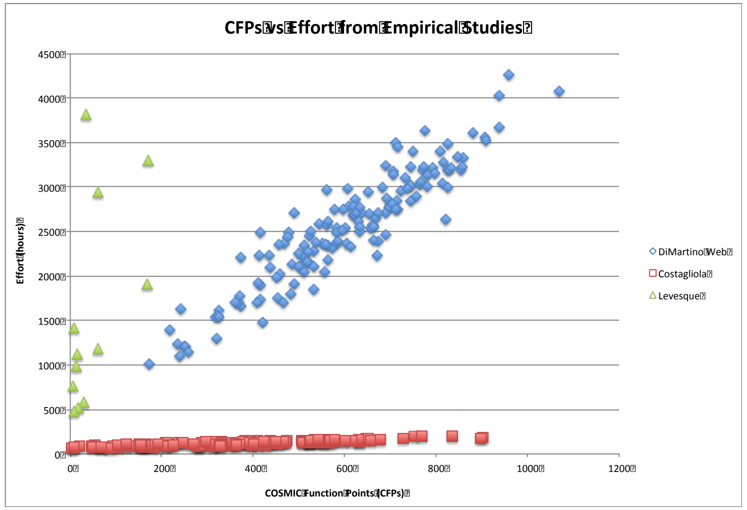
COSMIC Software Model for Function Points (CFPs)

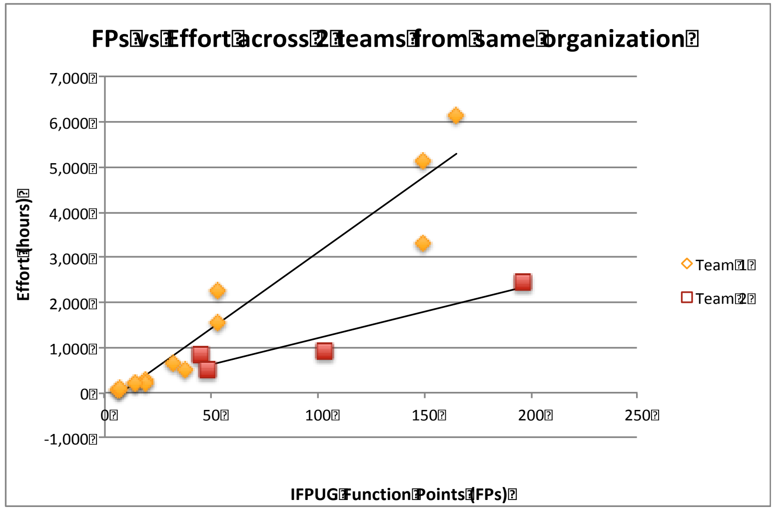


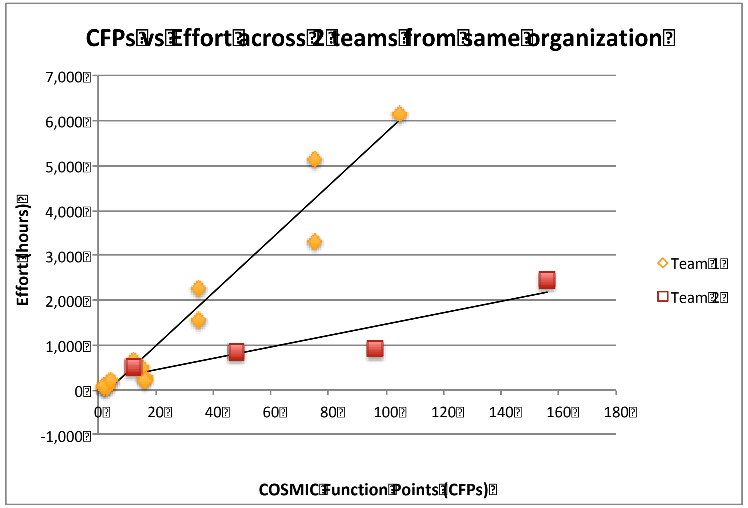
Challenges for Organizations/Teams

- Research papers and organizations typically do not account for effort factors.
- Teams will not have much data initially.
- Sharing data across teams cause high variance.
- Empirical research doesn't propose generalized model.
- Public datasets varied with high variance.









COCOMO® II Effort Estimation Model

$$PM = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

Exponent ranges from 0.9 to 1.2, with 1.0997 as default

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

Datasets

Unified Code Count (UCC)

- Maintenance projects
  - Add new features (10)
  - Modify existing features (23)
- Code metrics tool
- Command line program
- Implemented in C++, Java
- Each project by new team
- 32 data points

Confidential Industry

- New development, with some reuse from previous work
- Firmware and software interacts with hardware
- Command line program
- Implemented in C, Verilog, VHDL
- Data from 2 teams
- 18 data points

Methodology

Research Question and Hypothesis

1. Can calibrated COCOMO® II for FPs and CFPs perform better than options suggested in research?

- Null Hypothesis (H_0): Calibrated COCOMO® II will not perform better than the currently available options.

2. Do functional size metrics, using the calibrated COCOMO® II model, perform better on some types of projects compared to others?

- Null Hypothesis (H_0): Functional size metrics perform equally well on all types of projects.

Calibrating COCOMO® II

$$PM = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

1. Productivity Rates
   - New Development
   - Enhancement – New Features
   - Enhancement – Modify Existing Features
2. Adjust factors that may have relationship with size – Complexity (CPLX)
3. Adjust Exponent constants to adjust the rate at which effort grows with respect to size

Regressions to Compare (Question 1)

Compare Improvement

1. Linear Regression
2. Nonlinear Regression (log transform)
3. Convert FPs to SLOC using ratios published by Capers Jones
4. Convert FPs (and CFPs) to SLOC with custom conversion ratio
5. Convert CFPs to FPs and use existing model (linear and nonlinear)

Calibration Comparisons

Step-wise to determine significance

1. Productivity Factor
2. New Dev/Enh
3. New Dev/Add/Mod
4. Prod Factor & Complexity
5. New Dev/Enh & Complexity
6. New Dev/Add/Mod & Complexity

Step 1

2-Step Calibration

$$P\text{–}hrs = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

$$\log(P\text{–}hrs) = \log(A) + \left(B + C \times (\sum SF)\right) \times \log(Size) + \log\left(\prod EM\text{–}CPLX\right) + \log(CPLX)$$

Calibrate for Productivity Rates and Complexity (CPLX) factor.

Compare R^2 for best fit before moving to Step 2.

$$\log(P\text{–}hrs) - \log\left(\prod EM\text{–}CPLX\right) = \log(A) + \underbrace{\left(B + C \times (\sum SF)\right)} \times \log(Size) + \log(CPLX)$$

A number for now – throw away. Will calibrate in Step 2

2-Step Calibration

Step 2

Calibrate for Exponent constants B & C.

$$log(P\!-\!hrs) - log\left(\prod EM\!-\!CPLX\right) = log(A) + \left(B + C \times \left(\sum SF\right)\right) \times log(Size) + log(CPLX)$$

$$log(P\!-\!hrs) - log\left(\prod EM\!-\!CPLX\right) - log(A) - log(CPLX) = \left(B + C \times \left(\sum SF\right)\right) \times log(Size)$$

$$\frac{log(P\!-\!hrs) - log\left(\prod EM - CPLX\right) - log(A) - log(CPLX)}{log(Size)}$$

$$= B + C \times \left(\sum SF\right)$$

Normalize Effort

With Respect to Effort Factors defined by COCOMO® II

$$P\text{–}hrs = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

$$\frac{P\text{–}hrs}{\prod EM} = A \times Size^{(B + C \times (\sum SF))}$$

$$\frac{P\text{–}hrs}{\prod EM \times Size^{(C \times (\sum SF))}}$$

$$= A \times Size^{(B + C \times (\sum SF)) - (C \times (\sum SF))}$$

$$\frac{P\text{–}hrs}{\prod EM \times Size^{(C \times (\sum SF))}} = A \times Size^{B}$$

Software Estimation Prediction Accuracy Statistics

R^2: how closely the regression curve fits the data points

MMRE: Mean Magnitude of Relative Error. Ideally ≤ 25%

PRED(25): Percentage of estimates within 25% of actuals. Ideally ≥ 75%

Types of Projects (Question 2)

1. Low parsing projects – UCC: control and data management operations, 1-3 computational operations
2. High parsing projects – UCC: control and data management operations, 3-5 computational operations
3. Data transfers, interact with Hardware – Industry: control, data management, device-dependent, and simple computations
4. Record, Encrypt, Decrypt – Industry: control, data management, device-dependent, and complex computations
5. Input and Outputs – Industry, UCC: control and data management operations, 0 computational operations. Industry also includes some device-dependent operations.

Details of the calibrated COCOMO® II model

COCOMO® II for Functional Size Metrics

Calibration Step 1 Results

Step-wise process to determine significance

	IFPUG FPs (FPs)	COSMIC FPs (CFPs)
Prod Factor	0.404	0.631
New/Enh	0.655	0.748
New/Add/Mod	0.804	0.777
Prod Factor & CPLX	0.921	0.954
New/Enh & CPLX	0.922	0.973
New/Add/Mod & CPLX	0.957	0.975

Perform Step 2 of Calibration (exponent) on last model.

Original vs Calibrated COCOMO® II Model

$$\text{P−hrs} = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

	A	B	C
COCOMO® II (SLOC)	446.88	0.91	0.01
FP, New Development	52.602	0.833	0.011
FP, Add Features	100.51		
FP, Modify Features	43.84		
CFP, New Development	166.94	0.629	0.014
CFP, Add Features	95.04		
CFP, Modify Feature	76.32		

Original vs Calibrated Exponent

$$P - hrs = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

Exponent Range:	Low	Default	High
COCOMO® II (SLOC)	0.91	1.0997	1.2262
COCOMO® II (FPs)	0.833	1.0511	1.1963
COCOMO® II (CFPs)	0.629	0.9015	1.0829

Original vs Calibrated Complexity Factor

$$PM = A \times Size^{(B + C \times (\sum SF))} \times \prod EM$$

Complexity Range:	**Very Low**	**Nominal**	**Extra High**
COCOMO® II (SLOC)	0.73	1	1.74
COCOMO® II (FPs)	0.57	1	2.298
COCOMO® II (CFPs)	0.53	1	2.57

Calibrated COCOMO® II vs Options in Research

Does Calibrated COCOMO® II perform better than options suggested or provided in research papers?

IFPUG FPs

Statistic	Linear	Nonlinear	Jones Conversion to SLOC	Local Conversion to SLOC	Calibrated COCOMO® II
MMRE	89.87%	72.86%	47.92%	72.07%	**31.14%**
PRED(25)	20%	36%	14%	2%	**68%**

Conclusion: Calibrated COCOMO® II performed better than other options

COSMIC FPs

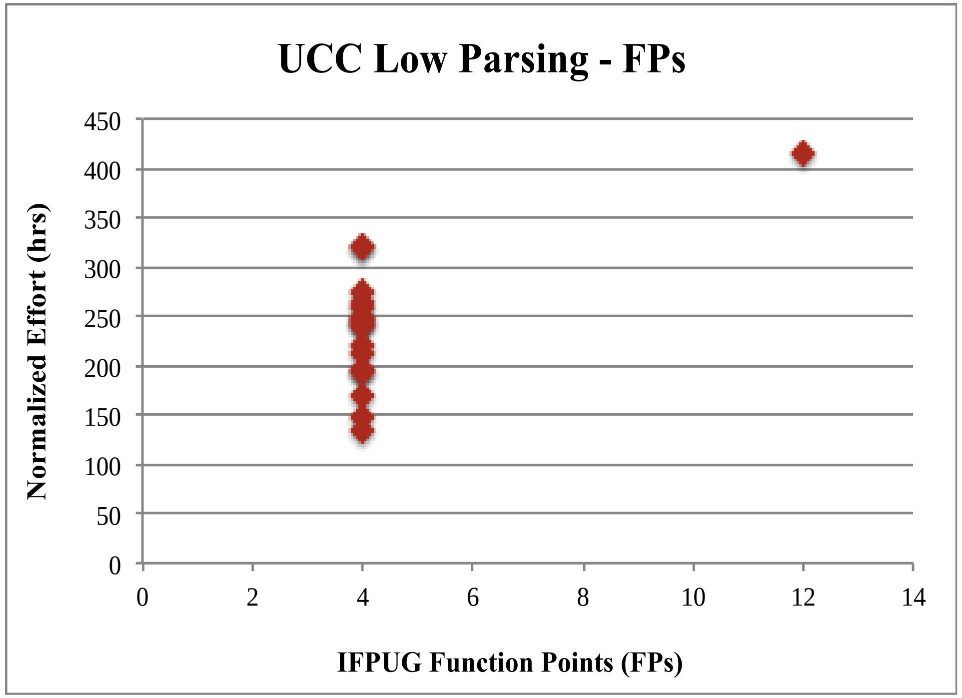
Statistic	Linear	Nonlinear	Local Conversion to SLOC	Conversion to FPs - Linear	Conversion to FPs - Nonlinear	Calibrated COCOMO® II
MMRE	96.2%	56.02%	80.05%	89.38%	70.39%	**20.94%**
PRED(25)	18%	38%	4%	22%	20%	**70%**

Conclusion: Calibrated COCOMO® II performed better than other options

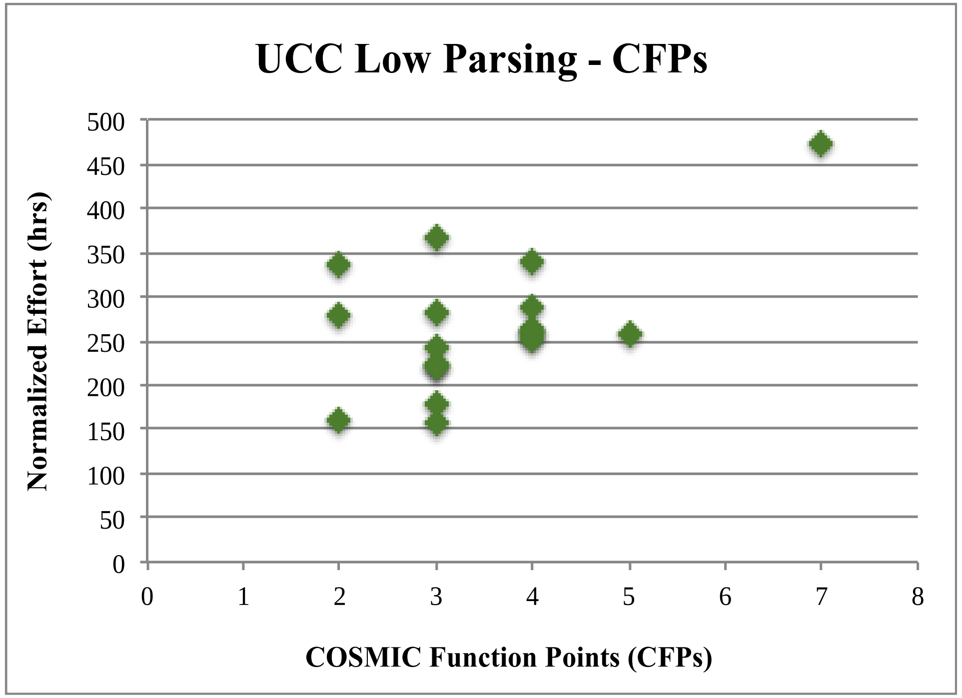
Do functional size metrics, with the calibrated COCOMO® II model, perform better on some types of projects compared to others?

Analysis by Project Type

1. Low parsing projects - UCC

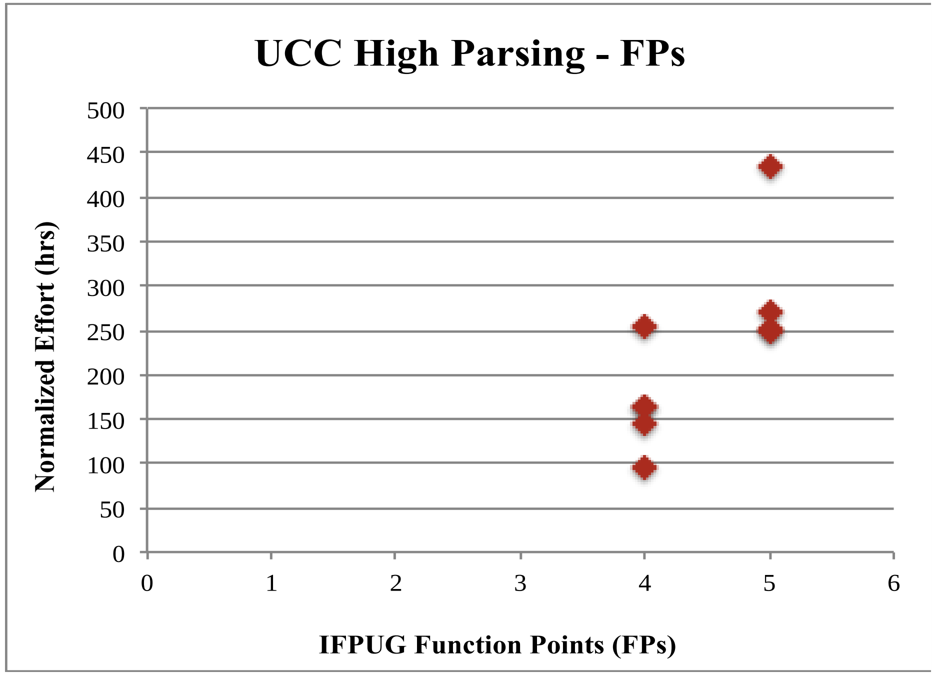


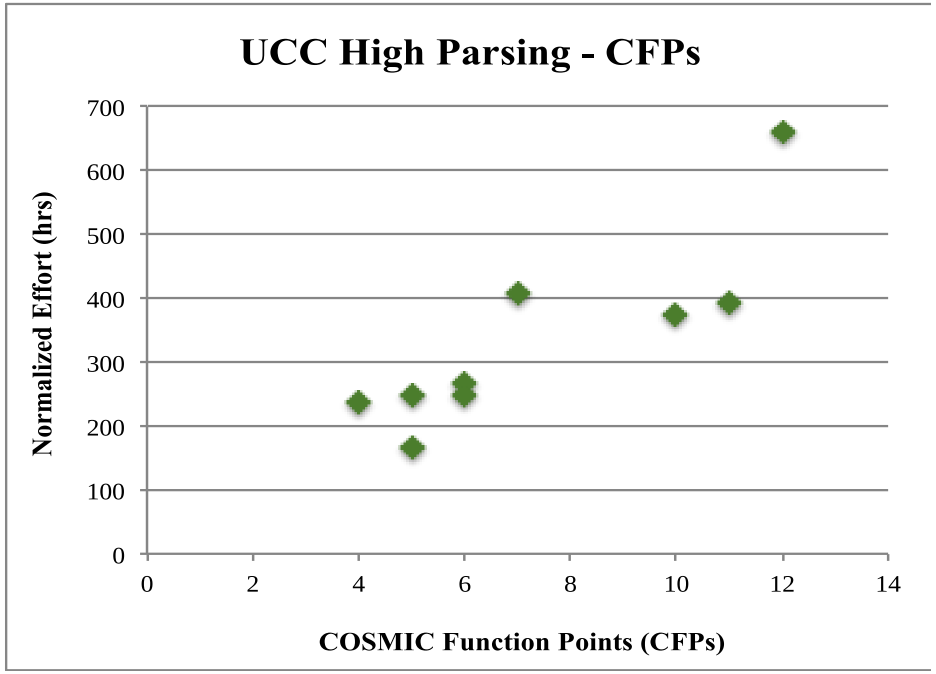
PRED(25)	80
MMRE	16.21
Corr Coeff	0.378



PRED(25)	75
MMRE	19.27
Corr Coeff	0.328

2. High parsing projects - UCC

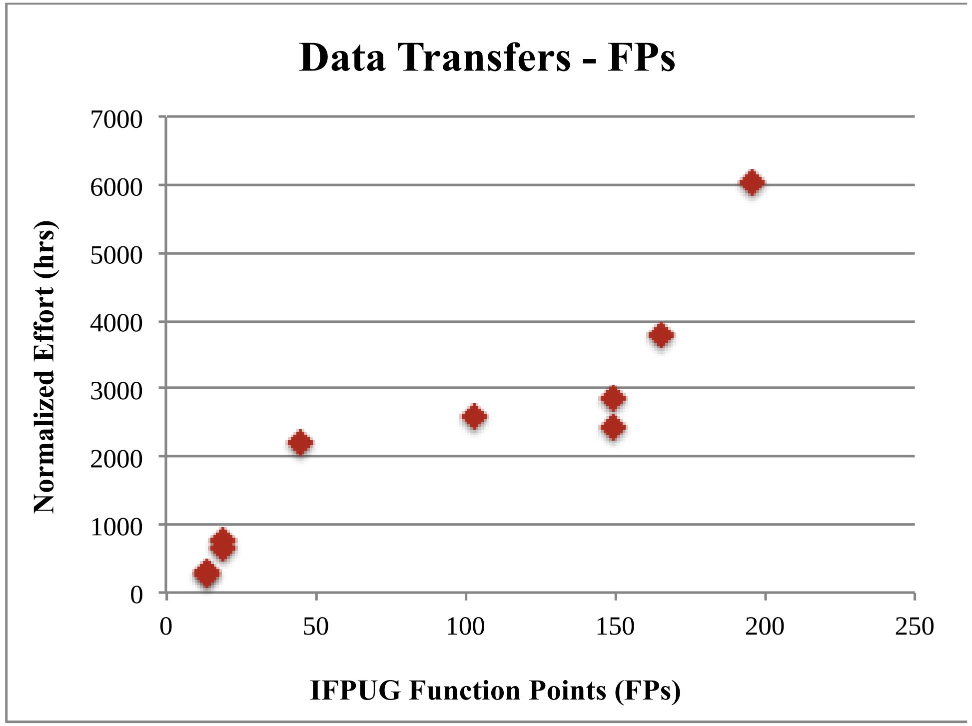




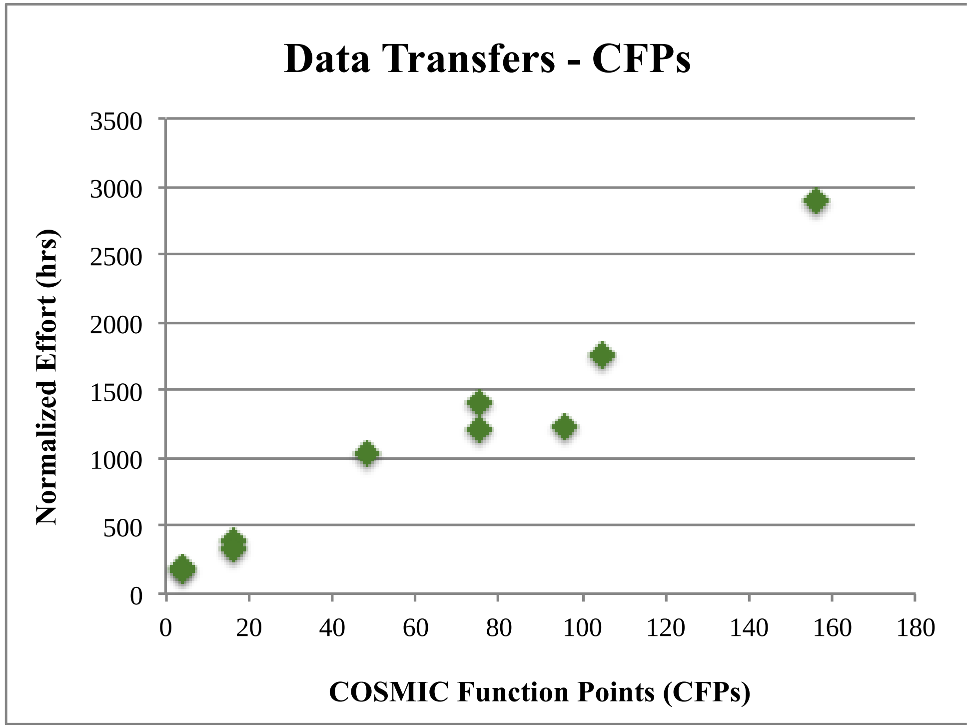
PRED(25)	66.67
MMRE	33.51
Corr Coeff	0.693

PRED(25)	77.78
MMRE	21.76
Corr Coeff	0.882

3. Data Transfer, HW – Industry

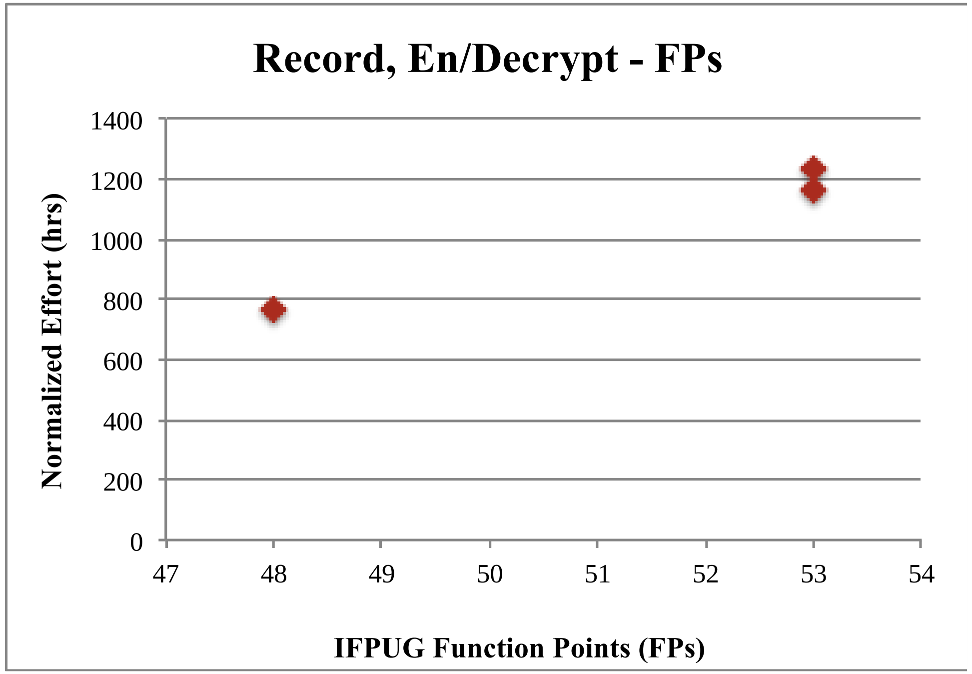


PRED(25)	50
MMRE	31.85
Corr Coeff	0.973

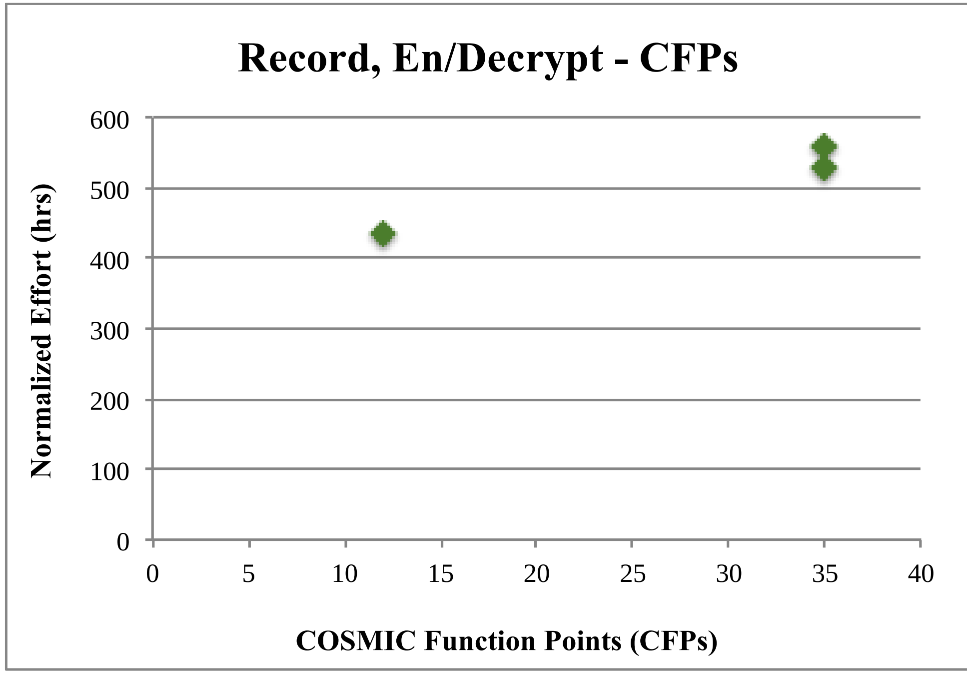


PRED(25)	80
MMRE	19.36
Corr Coeff	0.973

4. Record, Encrypt, Decrypt – Industry

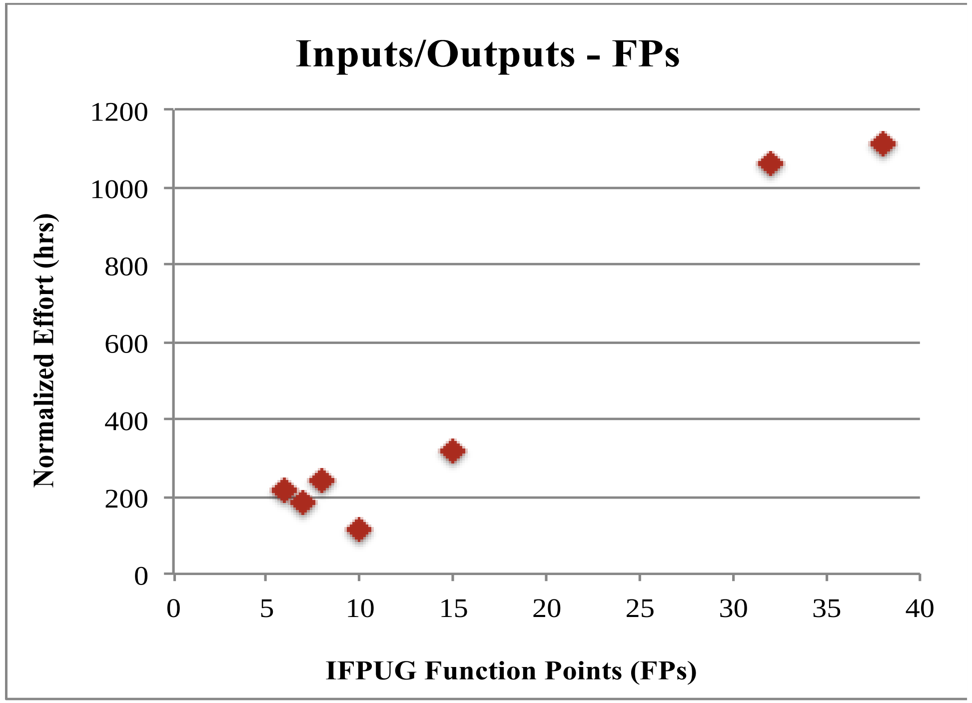


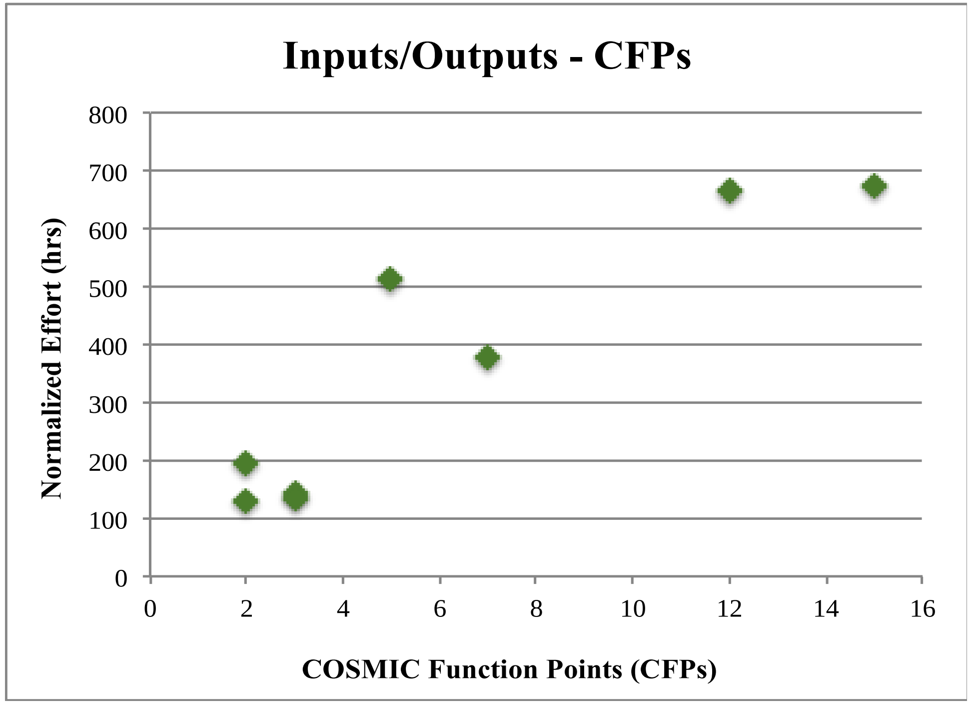
PRED(25)	100
MMRE	21.53
Corr Coeff	0.866



PRED(25)	66.67
MMRE	15.45
Corr Coeff	0.866

5. Inputs/Outputs – Industry, UCC





PRED(25)	50
MMRE	68.49
Corr Coeff	0.683

PRED(25)	37.5
MMRE	28.22
Corr Coeff	0.868

Conclusions

Conclusions (1/2)

Hypothesis 1

- Calibrated COCOMO® II will not improve effort estimation accuracy compared to other suggested methods.
  - False: Calibrated COCOMO® II performed better for FPs and CFPs by a minimum
    - PRED(25): 32%
    - MMRE: 21.44%

COCOMO® II Calibration

1. Different productivity rates for New Development, Add Modules, and Modifying Existing Modules
2. Complexity has stronger impact (in both low and high directions) on effort compared to SLOC.
3. Effort grows at different rate with respect to functional size metrics compared to SLOC

Conclusions (2/2)

Hypothesis (2)

- Functional Size Metrics perform similarly well on all types of projects
  - Performance varies
  - See table to right where
    - Red X means correlation coefficient and/or accuracy low
    - Green check means correlation and accuracy within acceptable ranges

	FPs	CFPs
Low Parse – UCC	✗	✗
High Parse – UCC	✗	✓
Data Transfer – Industry	✗	✓
Record, Encrypt, Decrypt – Industry	✓	✗
Inputs/Outputs – UCC, Industry	✗	✗

IFPUG vs COSMIC Function Points

IFPUG Function Points

- New development tasks
- Large number of data transferred
- Components with encryption and decryption functionality

COSMIC Function Points

- Generally across multiple datasets
- Smaller/various changes in data transferred
- Maintenance tasks
- Object-Oriented Design

Future Work

- Reuse factors in functional size (equivalent size like reuse model in COCOMO® II for SLOC)
- Separate the 5 types of Complexity as separate factors:
  - Control, Computational, Device-Dependent, Data Management, and User Interface Operations
  - Each may have different impacts
- Size metric that accounts for changes in algorithms