

CS614B

*Advanced Topics in Software Quality*

 **Software Estimation: Methods and Experience**

Danny Ho

Toronto Design Center

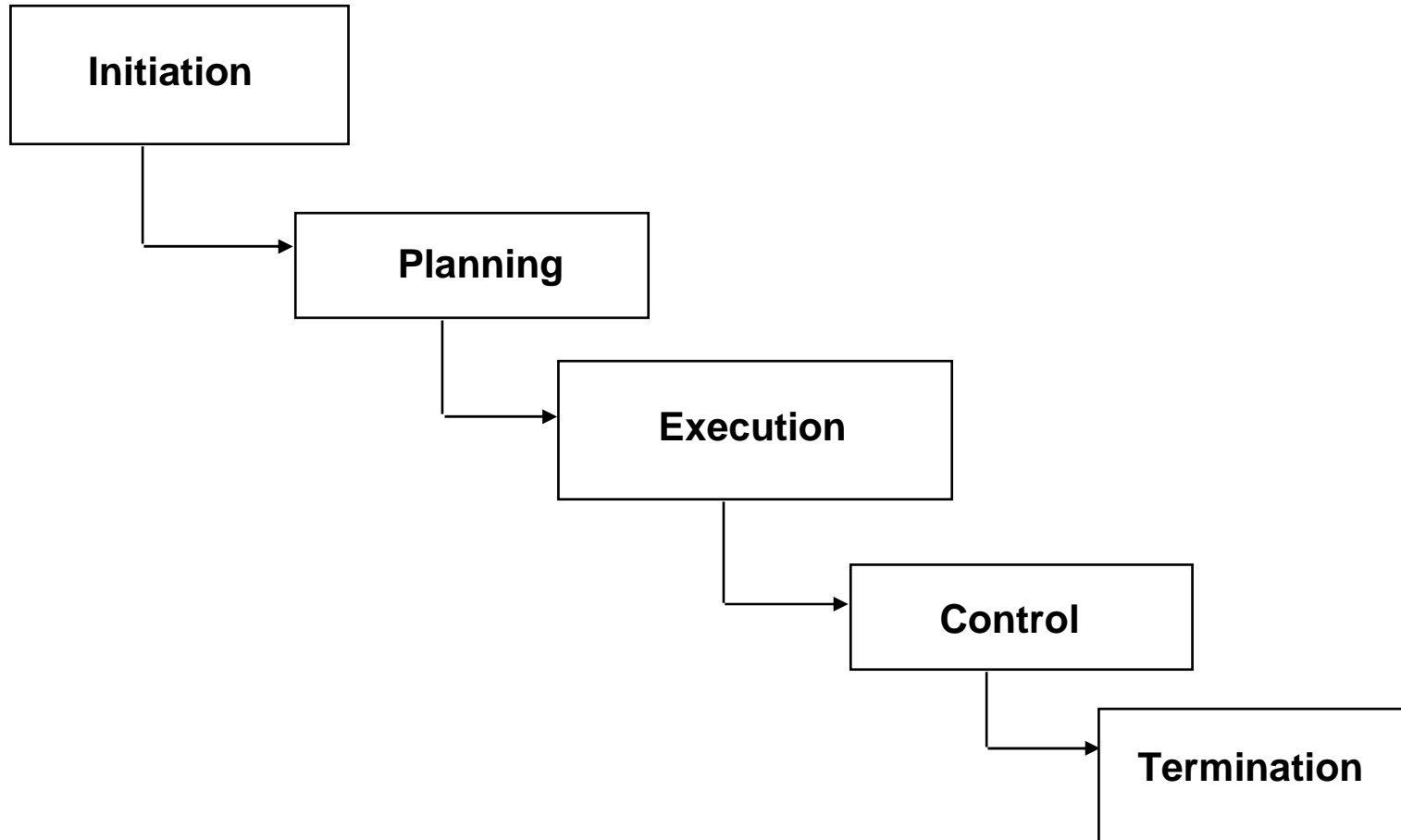
Motorola Canada Ltd

February 2006

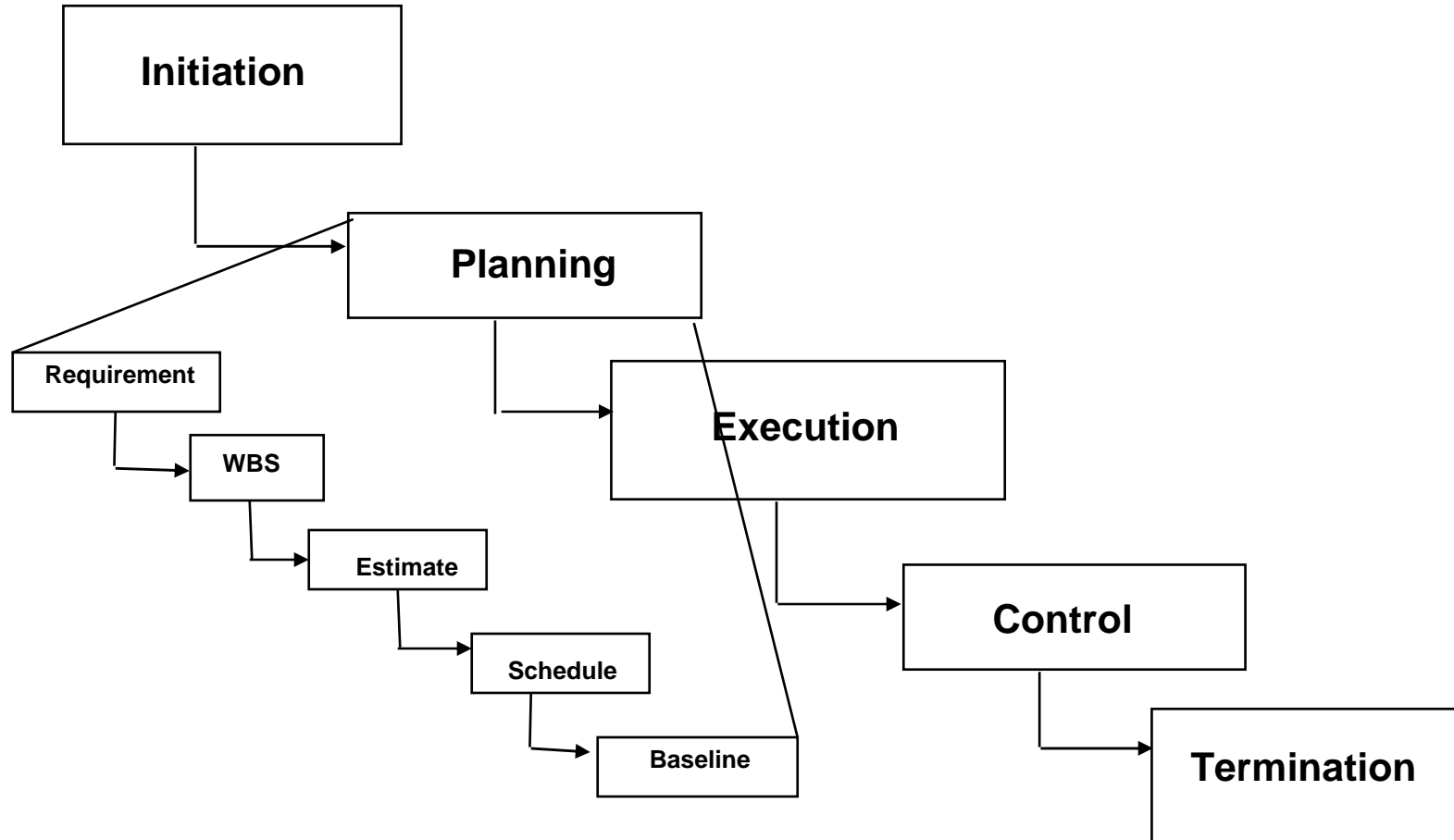
# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Project Lifecycle



# Project Planning



# M-Gates Core Process Redesign

## Business Case Development

M15: idea accepted  
M14: concept accepted  
M13: solution selected

## Portfolio Planning

M12: portfolio accepted  
M11: solution lockdown

## Project Definition

M10: project initiation  
M9: system requirements baselined  
M8: system requirements allocated  
M7: contract book baselined and approved

## Implementation

M6: design readiness  
M5: system test readiness  
M4: ready for field test  
M3: ready for controlled introduction

## Launch and Closeout

M2: volume deployment  
M1: retirement plan approved  
M0: end of life

# M-Gates Project Estimation

## Business Case Development

M15: idea accepted  
M14: concept accepted  
M13: solution selected

M13 Estimate: + 75%, -25%

## Portfolio Planning

M12: portfolio accepted  
M11: solution lockdown

## Project Definition

M10: project initiation  
M9: system requirements baselined  
M8: system requirements allocated  
M7: contract book baselined and approved

M10 Estimate: +25%, -10%

M7 Estimate: +10%, -5%

## Implementation

M6: design readiness  
M5: system test readiness  
M4: ready for field test  
M3: ready for controlled introduction

M3: Capture Actuals

## Launch and Closeout

M2: volume deployment  
M1: retirement plan approved  
M0: end of life

# What is Software Estimation?

## Estimate

- size
- schedule
- cost
- staff
- reliability

# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Why use Estimation Models?

- provide estimates using analytical tools
- form basis for disciplined planning
- calibrate tools based on experience
- validate bottom-up estimates
- provide insight in productivity and quality improvement

Models are not perfect - use more than one

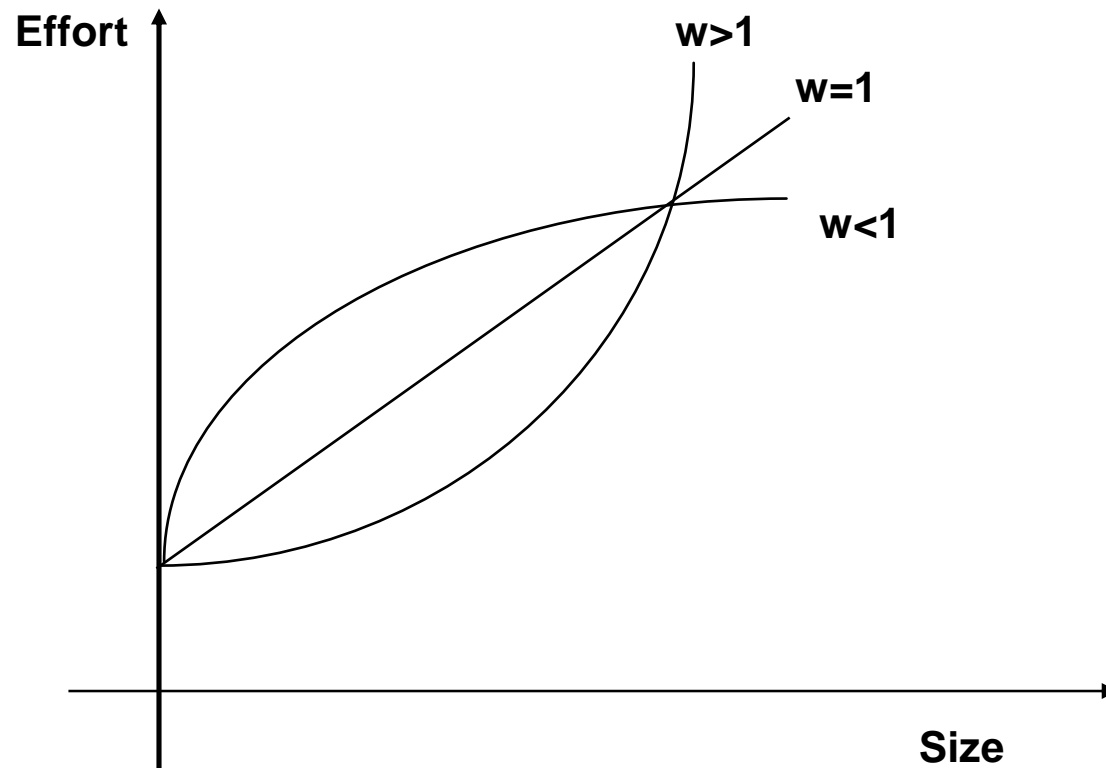
# Categorization of Estimation Models

- Historical (Experiential) Models
- Statistical Models
- Theoretical Models
- Composite Models

# Statistical Model

**Generic Formula:  $\text{effort} = k (\text{size})^w$**

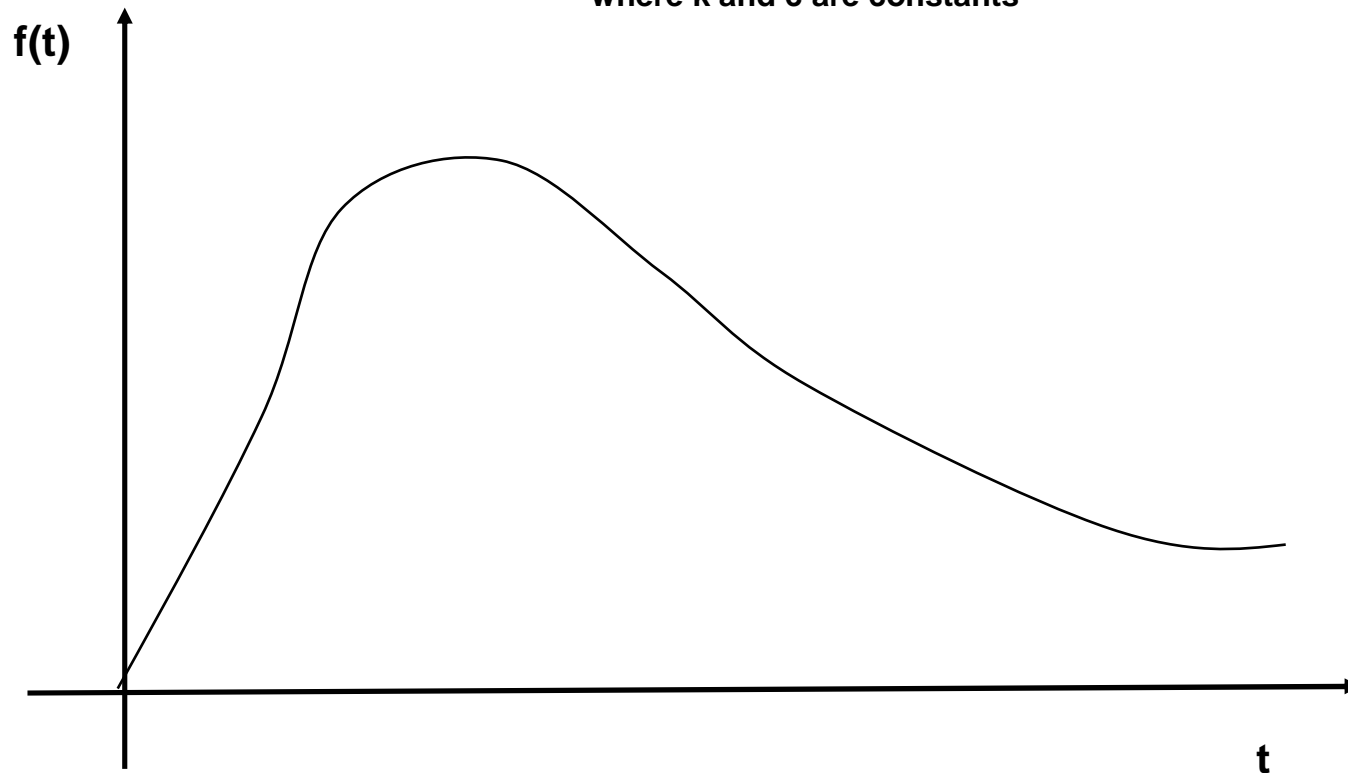
where  $k$  and  $w$  are constants



# Rayleigh Model

**Generic Formula:  $f(t) = (2k / c^2) t \exp(- (t/c)^2 )$**

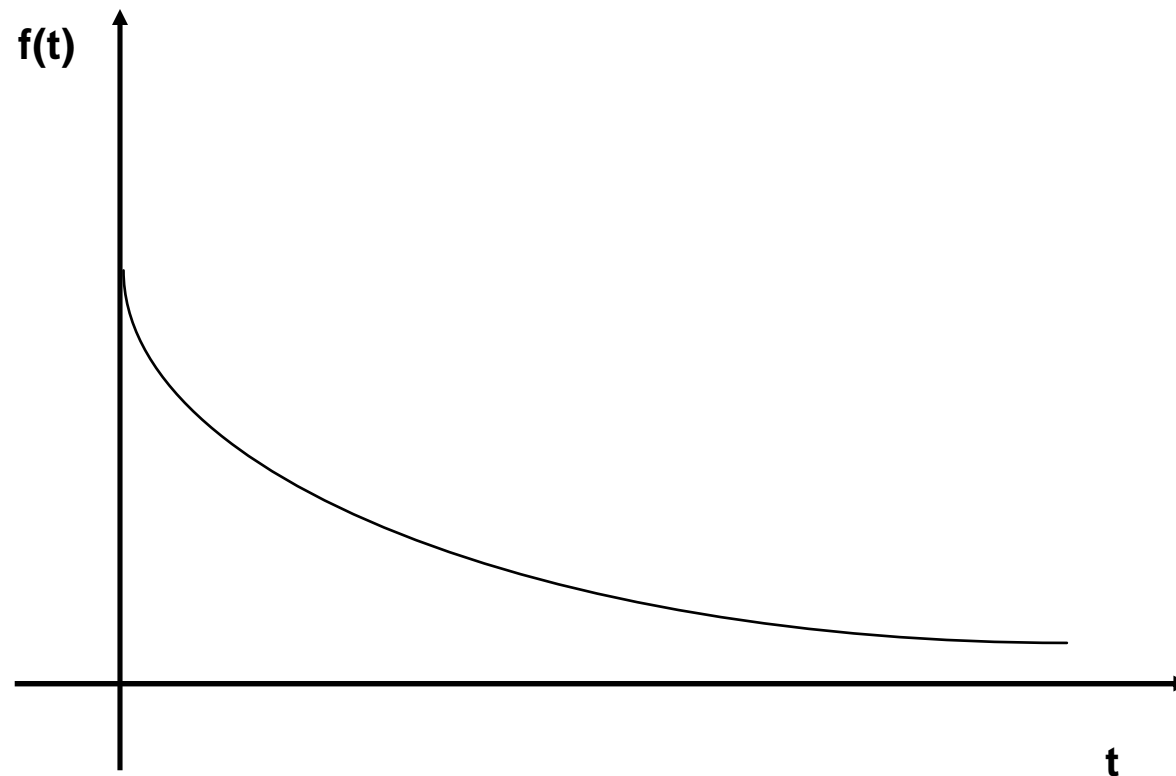
where k and c are constants



# Exponential Model

**Generic Formula:  $f(t) = (k / c) \exp(- (t/c) )$**

where k and c are constants



# Function Point

- invented by Al Albrecht
- populated by Capers Jones
- dimensionless measure using hard factors

Function Type	Simple	Average	Complex
Input	x3	x4	x6
Output	x4	x5	x7
Logical Internal File	x7	x10	x15
External Interface	x5	x7	x10
External Inquiry	x3	x4	x6

# Feature Point

- superset of function point with soft factor consideration: program/code/data complexity
- additional type: algorithm

Function Type	Weight
Algorithm	x3
Input	x4
Output	x5
Logical Internal File	x7
External Interface	x7
External Inquiry	x4

# Software Sizing Model

## ● Prerequisites

- based on at least two modules of known size
- total modules to estimate  $< 35$

## ● Algorithm

- Pairwise comparison
- PERT
- Sorting
- Ranking

# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Constructive Cost Model (COCOMO)

- estimate time, staffing and cost based on size
- arguably the most frequently used model
- developed in 1976-1980 by Dr Barry Boehm
- users' group since 1985
- developed COCOMO 2 in 1990s
- many commercial versions and military standard

# COCOMO Equation

$$\text{Effort} = A * (\text{Size})^B * (\text{EMs})$$

**where:**

- **effort is in staff-month**
- **A is a constant**
- **Size is in KSLOC**
- **B is a scale factor**
- **EMs are effort multipliers**

# Sizing Approach

- **Application composition / prototyping**
  - **object point**
- **Early design**
  - **unadjusted function point**
- **Post architecture**
  - **KSLOC**

# Scale Factors

$$B = 1.0997 + \text{scale factors}$$

<b>Scale Factors</b>	<b>very low</b>	<b>low</b>	<b>nominal</b>	<b>high</b>	<b>very high</b>	<b>extra high</b>
<b>Precedentedness</b>	0.1207	0.0966	0.0724	0.0483	0.0240	0.00
<b>Development Flexibility</b>	0.0348	0.0279	0.0209	0.0139	0.0070	0.00
<b>Architecture/Risk Resolution</b>	0.0050	0.0040	0.0030	0.0020	0.0010	0.00
<b>Team Cohesion</b>	0.1858	0.1486	0.1115	0.0743	0.0372	0.00
<b>Process Maturity</b>	0.0542	0.0433	0.0325	0.0216	0.0108	0.00

# Early Design Effort Multipliers

Cost Drivers	extra low	very low	low	nominal	high	very high	extra high
RCPX	0.48	0.62	0.80	1.0	1.25	1.75	2.50
RUSE			0.89	1.0	1.16	1.34	1.56
PDIF			0.87	1.0	1.25	1.80	3.00
PERS	2.40	1.66	1.25	1.0	0.80	0.60	0.44
PREX	1.82	1.44	1.20	1.0	0.83	0.68	0.55
FCIL	1.49	1.34	1.17	1.0	0.86	0.71	0.59
SCED		1.23	1.08	1.0	1.04	1.10	

# Post-Architecture Effort Multipliers

## Product

- RELY, DATA, CPLX, RUSE, DOCU

## Platform

- TIME, STOR, PVOL

## Personnel

- ACAP, PCAP, PCON, AEXP, PEXP, LTEX

## Project

- TOOL, SITE, SCED

# The Costar Tool

**Costar - ProjectName**

File View Reports Help

Estimate Name:  ID:

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K\$)	Total Size: 10,000
Requirements	RQ:	2.1	1.6	0.0	
Development	PD+DD+CT+IT:	29.5	8.8	0.0	
<b>Total</b>	<b>RQ+PD+DD+CT+IT:</b>	<b>31.5</b>	<b>10.4</b>	<b>0.0</b>	

**COCOMO Model:**  
  
 Model Type: COCOMO II Model  
 Model ID: 1997

**COCOMO II Scale Factors**

Precedentedness:   
 Development Flexibility:   
 Architecture / Risk Resolution:   
 Team Cohesion:   
 Process Maturity:

Increment:   
 Level:

**Personnel**

ACAP... High  
 AEXP... Nominal  
 PCAP... Low  
 PEXP... Nominal  
 LTEX... Low  
 PCON... Very High

**Platform**

TIME... Nominal  
 STOR... Nominal  
 PVOL... Nominal

**Product**

RELY... Nominal  
 DATA... Low  
 CPLX... High  
 RUSE... Nominal  
 DOCU... Nominal

**Project**

TOOL... High  
 SITE... Low  
 SCED... Nominal

**Size Summary**

Size: 10,000  
 Method: SLOC

**User Defined**

USR1... Undefined  
 USR2... Undefined  
 USR3... Undefined  
 USR4... Undefined

Model / Equations / Increment Phasing / Increment Breakage / Labor Cost / APM / MN APM / Description /

Cost Drivers / Size & Breakage / Adaptation / Function Points / Maint. / Costs / Descr.

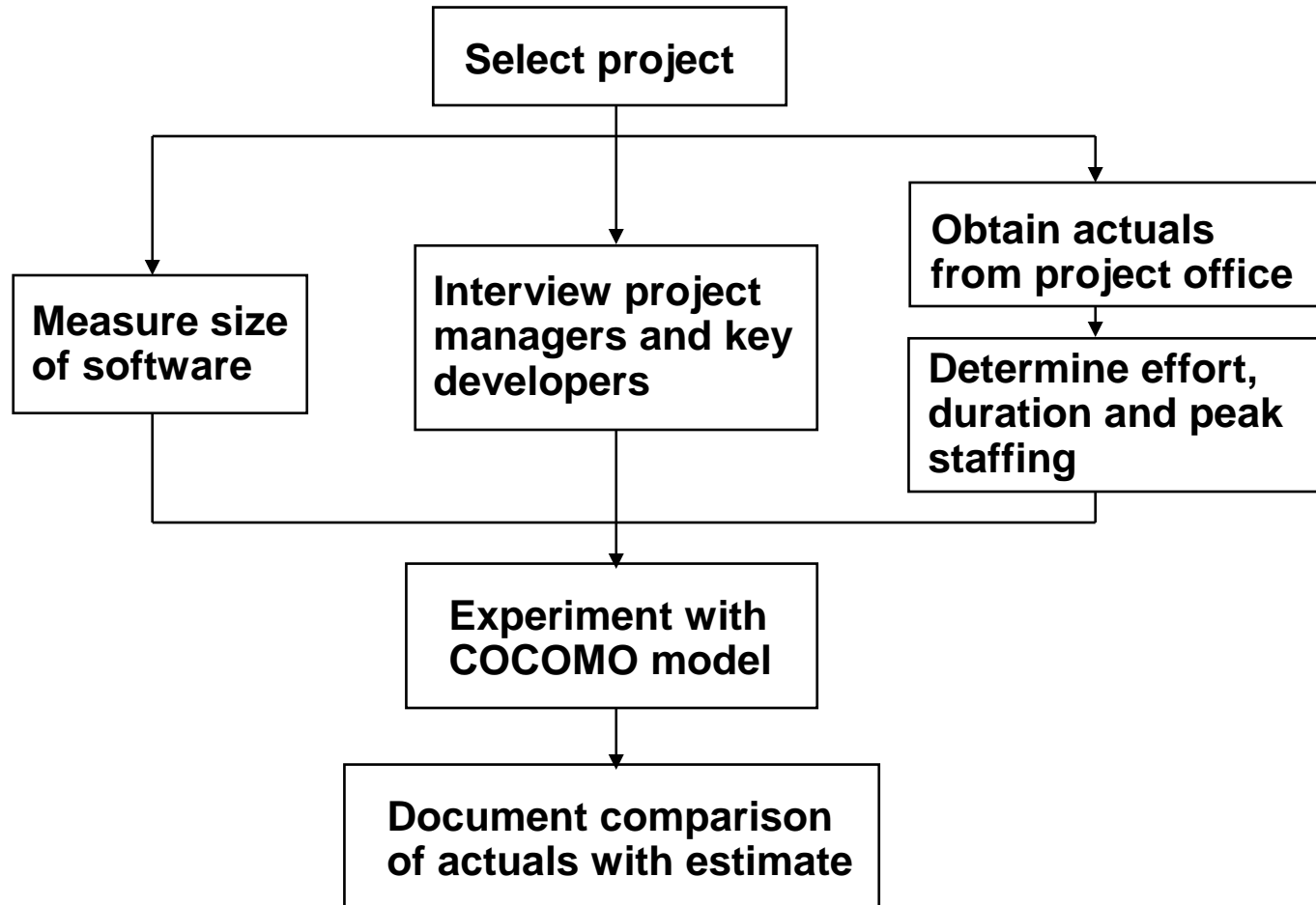
# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Source of Project Data

- messaging applications
- call center applications
- open system platforms
  
- development platforms
  - Windows
  - HP/UX
  - OS/2

# Calibration Process



# Calibration with Projects

<b>Applications</b>	<b>real time messaging</b>	<b>bug fix release</b>	<b>second update</b>
<b>Size</b> (ESLOC x1000) (function points) <b>Adaptation</b> (design/code/integration) <b>Language</b> <b>Platform</b>	596 old, 131 new 6549 old, 1440 new 5%, 5%, 25% Pascal UNIX	197 old, 11.4 new 2165 old, 125 new 10%, 25%, 30% Pascal UNIX	51.7 old, 54.3 new 568 old, 597 new 10%, 20%, 30% Pascal UNIX
<b>High EMs</b> <b>Low EMs</b>	ACAP, PCAP, VEXP, LEXP, CPLX, TIME, STOR, TOOL, RUSE DATA, VMVH, VMVT, SCED	ACAP, PCAP, AEXP, VEXP, LEXP, CPLX, STOR, TOOL, RUSE DATA, VMVH, VMVT, SCED	ACAP, PCAP, AEXP, VEXP, LEXP, CPLX, STOR, TOOL, RUSE DATA, VMVH, VMVT, SCED
<b>Duration</b> (months) <b>Effort</b> (staff-months) <b>Peak staffing</b>	COCOMO vs Actual 21.6 vs 24 811.9 vs 310.1 47.6 vs 48	COCOMO vs Actual 11.9 vs 12 148.9 vs 185 15.2 vs 37	COCOMO vs Actual 17.9 vs 19 275.4 vs 332 19 vs 25

All projects used COCOMO 87 database and semi-detached mode of development

# Calibration with Projects

<b>Applications</b>	<b>open system platform</b>	<b>call center application</b>	<b>messaging application</b>
Size (ESLOC x1000) (function points)	131 new 1016 new	84.4 old, 1.25 new 659 old, 10 new	32.1 old, 13.8 new 251 old, 108 new
Adaptation (design/code/integration)	N/A	2%, 5%, 40%	10%, 20%, 30%
Language	C	C	C
Platform	NT	UNIX	OS/2
High EMs	RELY, STOR, RUSE	VEXP, RELY, CPLX, STOR, RUSE	CPLX, TOOL, RUSE
Low EMs	VEXP, LEXP, DATA, VMVH, VMVT	VMVH, VMVT, MODP	AEXP, VEXP, LEXP, DATA, VMVH, VMVT
Very low EMs	TURN		TURN
Duration (months)	COCOMO vs Actual 24.7 vs 21	COCOMO vs Actual 11.1 vs 10	COCOMO vs Actual 5.7 vs 6
Effort (staff-months)	496.9 vs 421	69.9 vs 64.8	35.8 vs 31.2
Peak staffing	35.8 vs 28	7.5 vs 10	8.4 vs 5.7

# Estimation of a New Project

<b>Application</b>	<b>real time call center</b>	<b>six months later</b>
Size (ESLOC x1000) (function points) Adaptation Language Platform	98 new 875 new N/A C++ NT	98 new 875 new N/A C++ NT
Very high EMs High EMs Low EMs Very low EMs	TOOL RELY, DATA, CPLX, RUSE AEXP, VEXP, LEXP, VMVH, VMVT TURN	TOOL RELY, DATA, CPLX, RUSE AEXP, VEXP, LEXP, VMVH, VMVT, SCED TURN
Duration (months) Effort (staff-months) Peak staffing	COCOMO VS Bottom-up 23.3 vs 25 590.6 vs 484 32 vs 34	COCOMO VS Bottom-up 19.8 vs 20 638.3 vs 699 40.5 vs 46

# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Software Lifecycle Model (SLIM)

- metrics-based estimation model
- developed by Larry Putnam
- captured over 7000 projects from industry
- based on hundreds of consulting engagements
- classified into 9 application categories
- over 50 PY of R&D on the model

# SLIM Equation

**Product = Productivity \* Effort \* Time**

$$\text{ESLOC} = \text{PP} * (\text{PY}/\text{b})^{1/3} * (\text{Y})^{4/3}$$

**where:**

- **ESLOC is thousands of source lines of code**
- **PP is process productivity**
- **PY is effort in person years**
- **b is a special skills factor**
- **Y is duration in years**

# Determination of SLIM Variables

<b>b is a special skills factor and a function of system size</b>						
<b>Size</b>	5-15K	20K	30K	40K	50K	>70K
<b>Skills factor: b</b>	0.16	0.18	0.28	0.34	0.37	0.39

<b>Process Productivity (PP) translates into Productivity Index (PI)</b>								
<b>PP</b>	.....	5186	6765	8362	10946	17711	21892	.....
<b>PI</b>	.....	10	11	12	13	14	15	.....

# Productivity Index (PI)

Macro measure of total development environment

- management practices
- skills and motivation of developers
- tools and methods
- user involvement
- project complexity

# Productivity Index (PI) Baselines

<b>Application Category</b>	<b>PI</b>	<b>Standard Deviation (+/-)</b>
<b>business</b>	16.2	3.8
<b>scientific</b>	12.5	4.0
<b>system software</b>	11.3	3.9
<b>telecom</b>	10.4	3.1
<b>process control</b>	10.3	4.3
<b>command and control</b>	9.9	4.1
<b>real time</b>	6.8	3.1
<b>avionic</b>	6.4	3.3
<b>microcode</b>	1.5	2.7

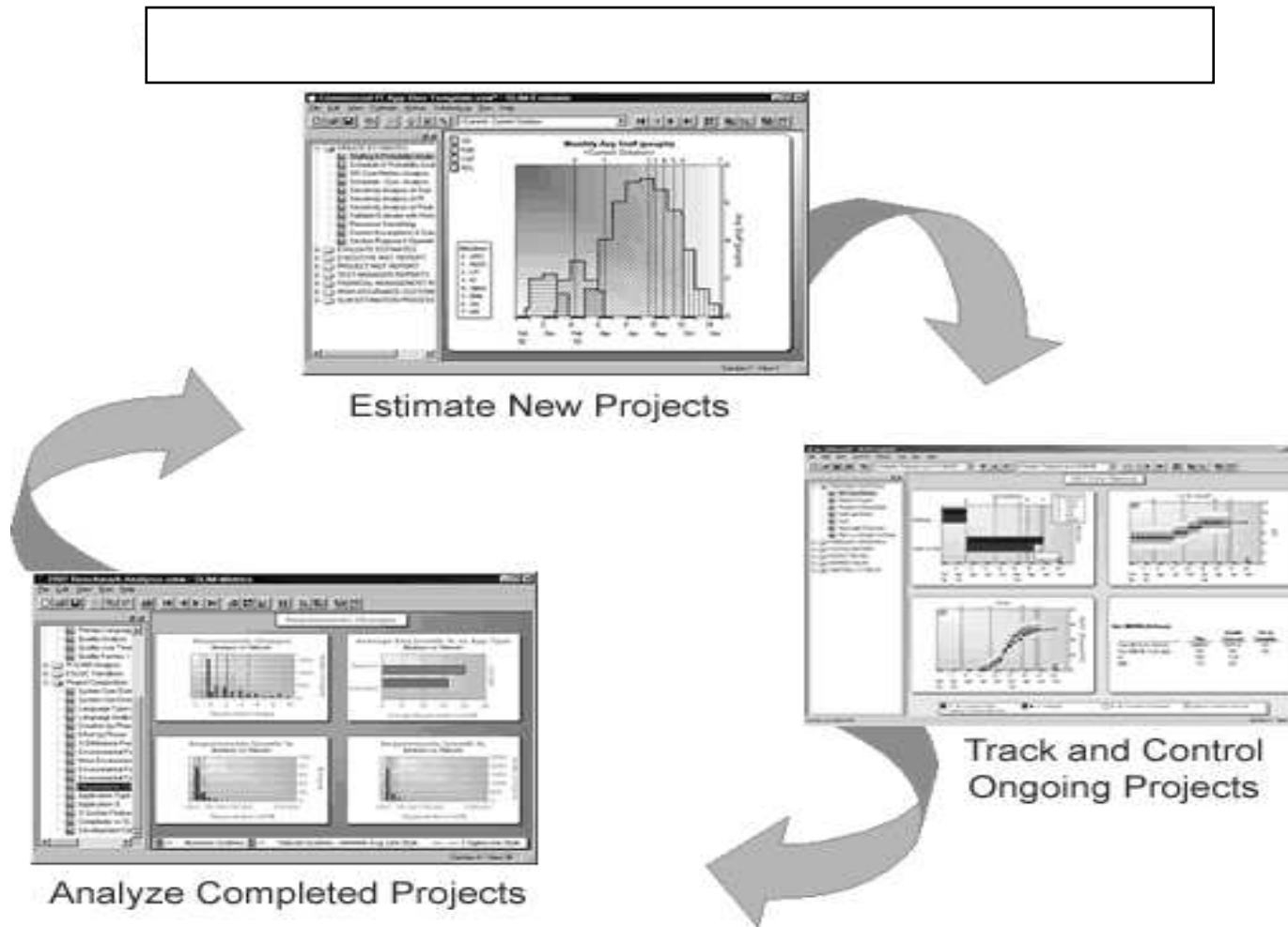
# Rule of Thumb for PI

- 1 PI improvement is worth roughly:
  - 10% schedule reduction
  - 25% effort reduction
  - 25% reliability improvement
- 1 PI improvement every 2 - 4 years

# Breakpoints for Engineering Systems

<b>SEI CMM Level</b>	<b>Description</b>	<b>PI</b>
I	<b>initial</b>	N/A
II	<b>repeatable</b>	11.0
III	<b>defined</b>	15.0
IV	<b>managed</b>	17.5
V	<b>optimized</b>	20.5

# The SLIM Tool



# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Source of Project Data

- compiler applications
- application development platforms
- distributed computing environment
- multimedia applications
- database applications
  
- development platforms
  - mainframe
  - OS/2
  - AIX
  - OS/400

# Calibration/Estimation with Projects

<b>Applications</b>	<b>compiler</b>	<b>compiler</b>	<b>multimedia</b>	<b>database</b>	<b>database</b>	<b>compiler</b>
Size( $ESLOC \times 1000$ )	20-40	39-41	40-60	70-90	85-87	90-110
Duration (months)	16.5	10.8	8.7	19.4	23	20.3
Effort (staff-months)	70.3	207	110	158	277	367
Peak staffing	5.7	28	19	12.5	18	28
Productivity index (PI)	10.3	13.8	16.3	12.8	11.3	12.3

# Calibration/Estimation with Projects

<b>Applications</b>	<b>distributed computing environment</b>	<b>application development platform</b>	<b>database</b>	<b>multimedia</b>	<b>distributed computing environment</b>
Size( $ESLOC \times 1000$ )	120-142	143-155	160-200	160-195	222-243
Duration (months)	25.3	22.9	23.8	14.2	35.4
Effort (staff-months)	709	526	641	498	973
Peak staffing	43	35	36	54	42
Productivity index (PI)	11.3	12.8	13.1	16.3	11.3

# COCOMO versus SLIM

	<b>COCOMO</b>	<b>SLIM</b>
Size (ESLOC x1000) (function points)	100 old, 4 new 781 old, 31 new	100 old, 4 new 781 old, 31 new
Adaptation	5% design, 10% code, 15% integration	1% lowest, 6% most likely, 12% highest
Language	C	C
COCOMO EMs versus SLIM PI	HIGH: ACAP, PCAP, VEXP, LEXP, MODP  LOW: DATA, VIRT  VERY LOW: RELY, TURN	PI = 13.5 tooling/method: 5 technical difficulty: 4 personnel profile: 7
Duration (months)	7	8
Effort (staff-months)	15.1	11.9
Peak staffing	2.4	2.1

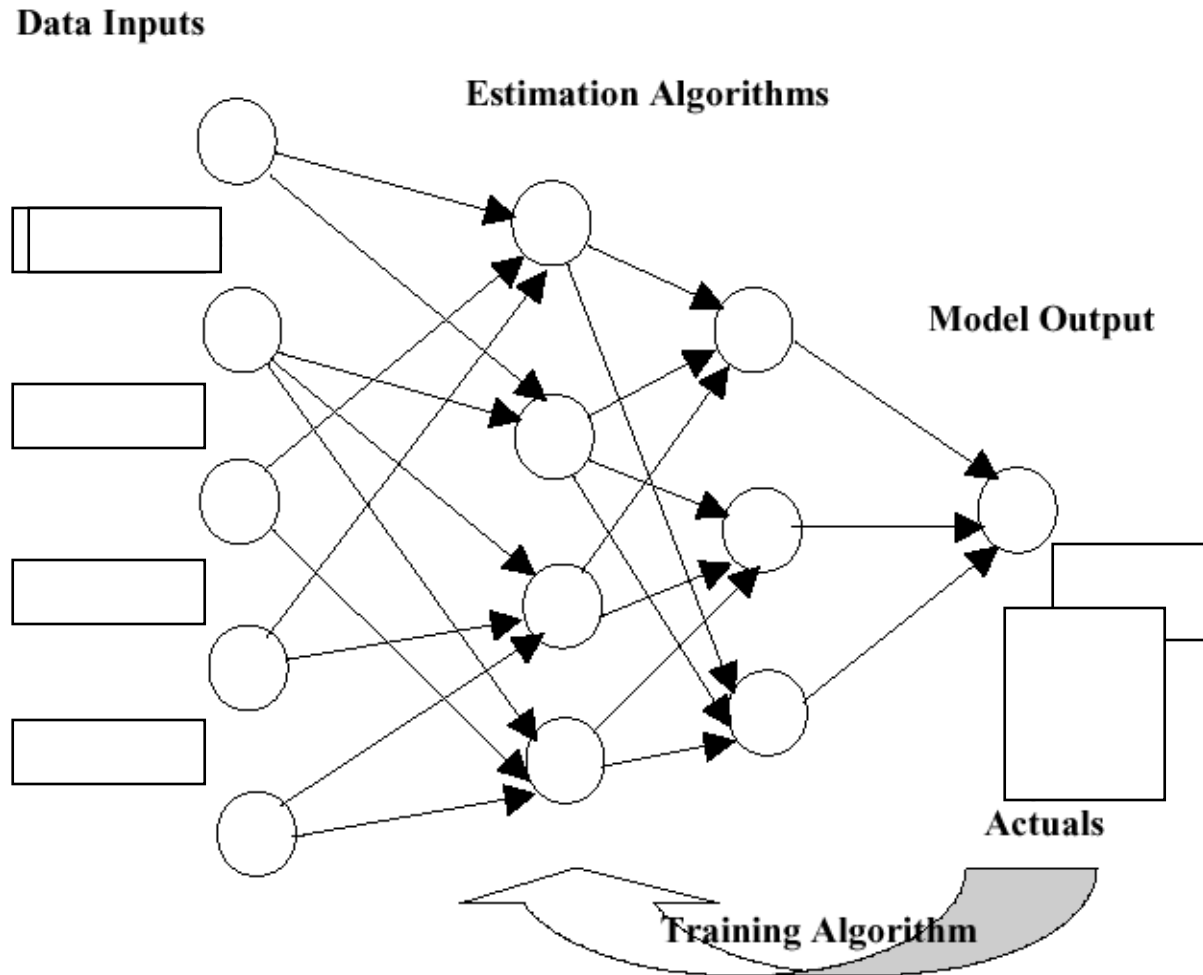
# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Neural Network

- Based on the principle of learning from example
- Advantage: model complex input-output relationship
- Disadvantage: lack of explanation

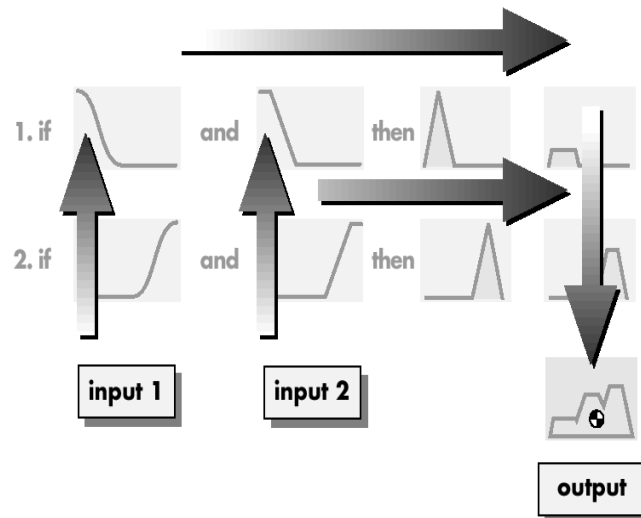
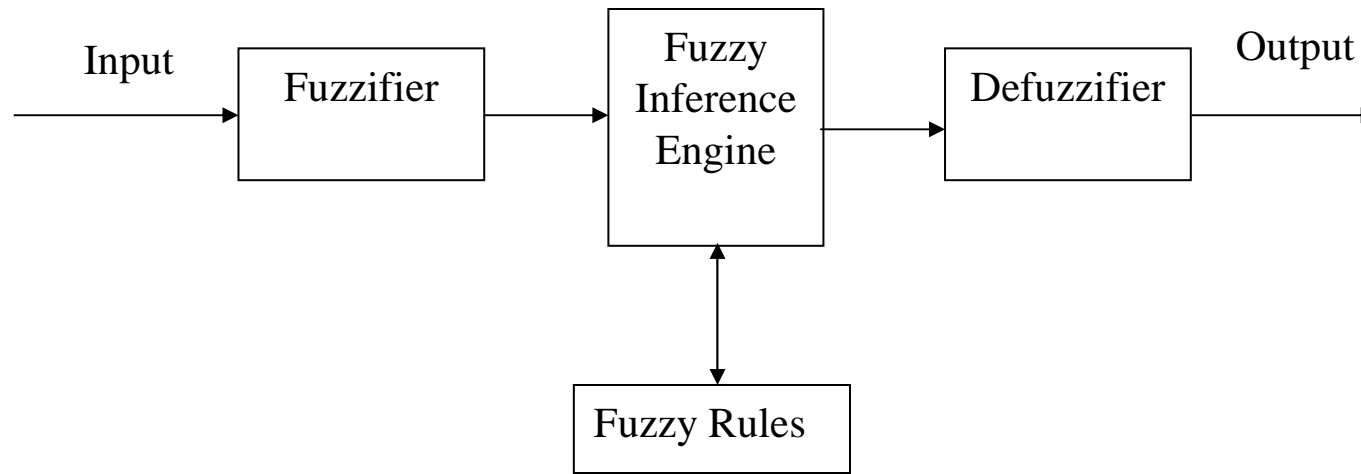
# Neural Network Model



# Fuzzy Logic

- Resolve uncertainty by mapping linguistic terms to numeric variables
- Capture human knowledge accurately
- Advantage: good interpretability
- Disadvantage: hard to define fuzzy rules

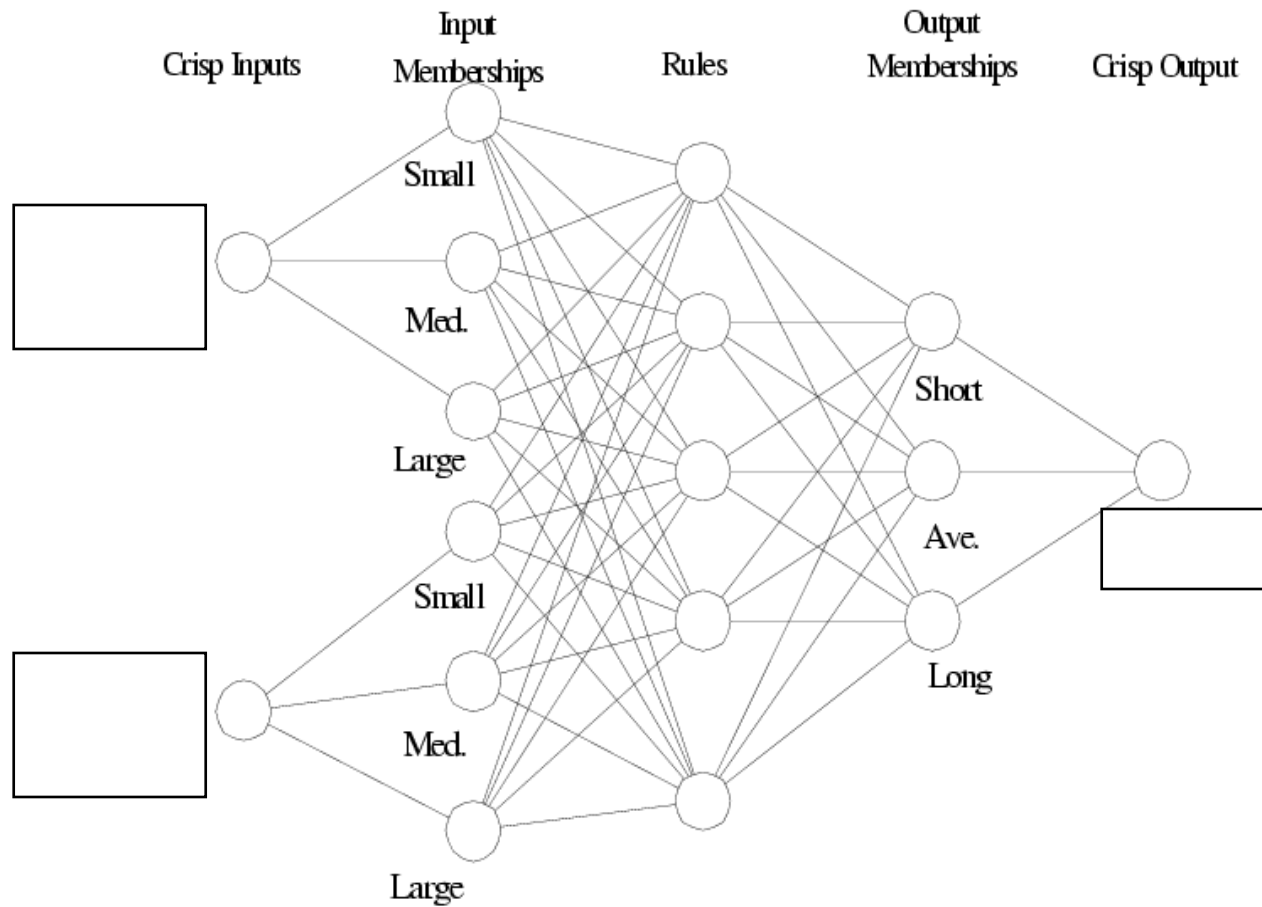
# Fuzzy Logic System



# Neuro-Fuzzy

- Combine the advantages of neural network and fuzzy logic
- Learning ability
- Knowledge integration
- Good Interpretability
- Good generalization

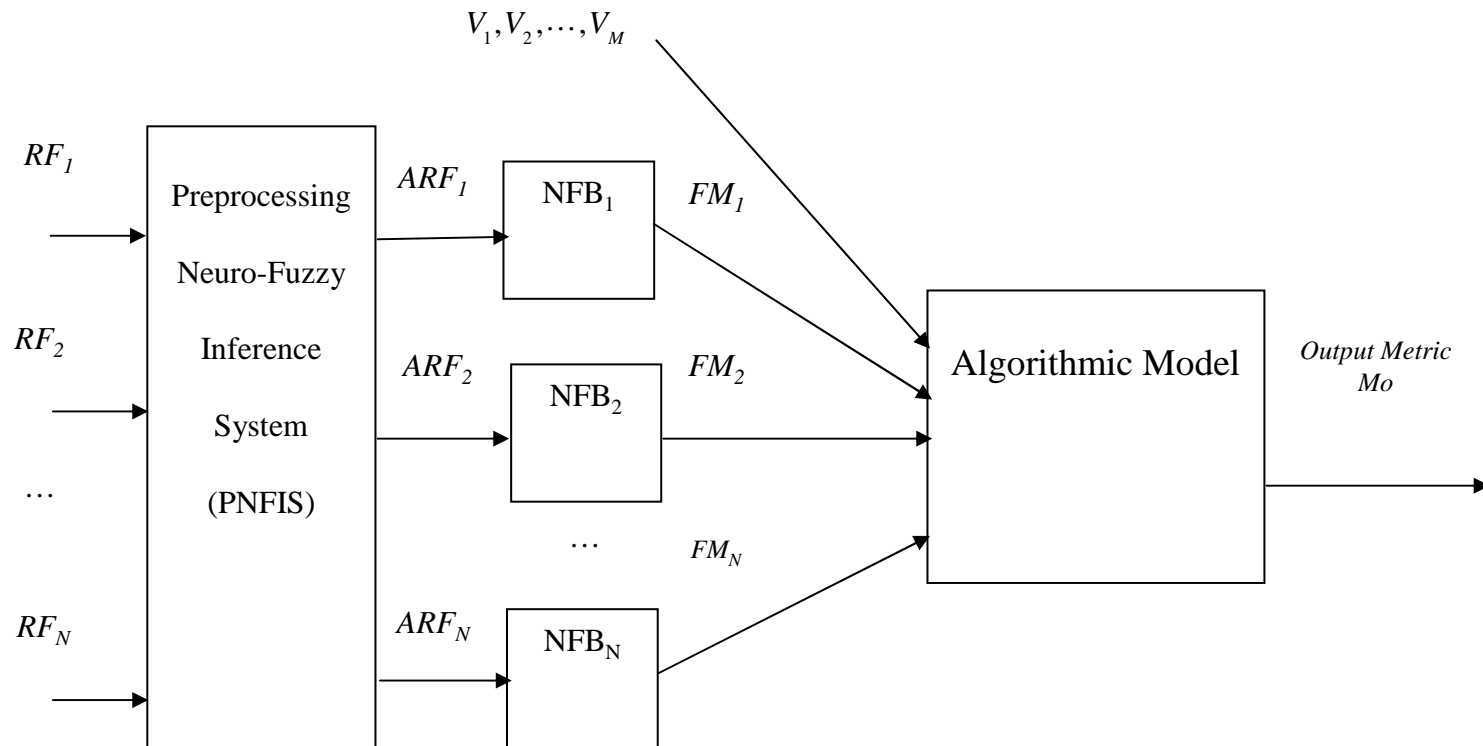
# Neuro-Fuzzy Architecture



# Agenda

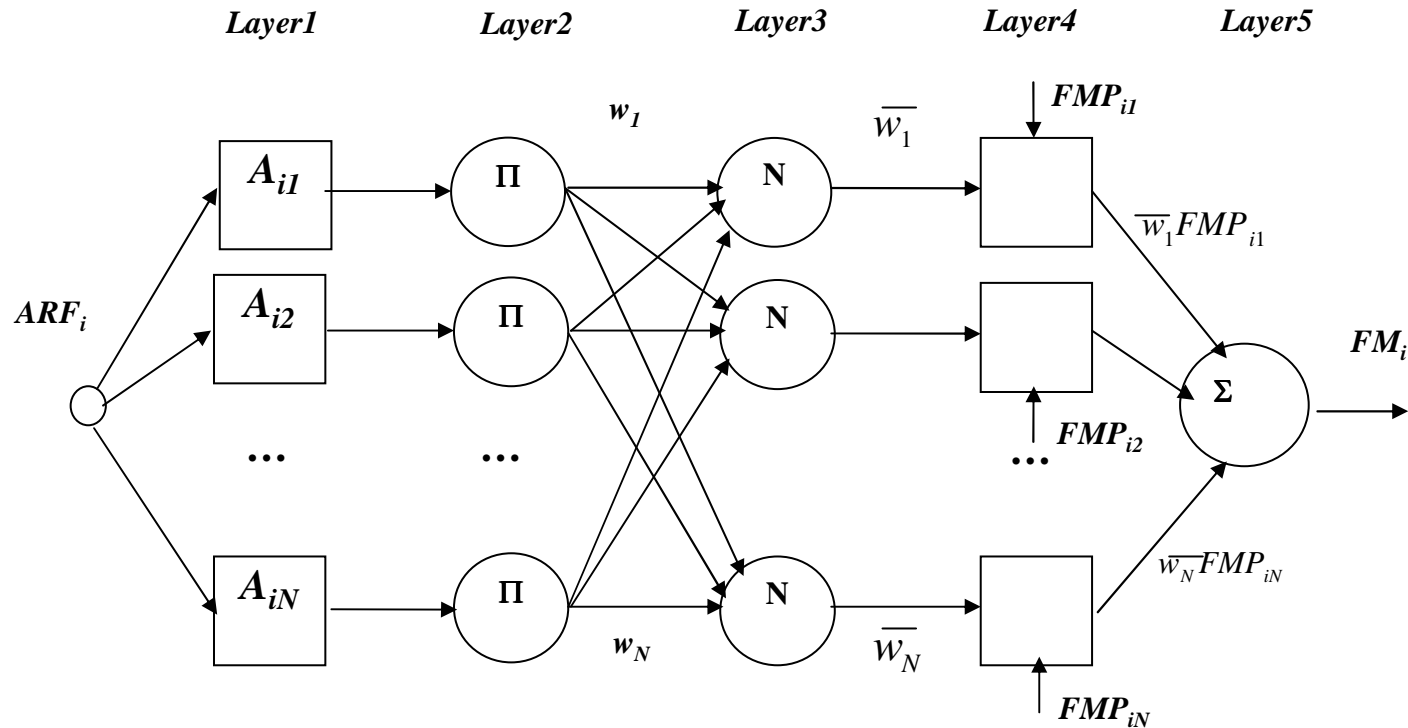
1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Neuro-Fuzzy Algorithmic Model



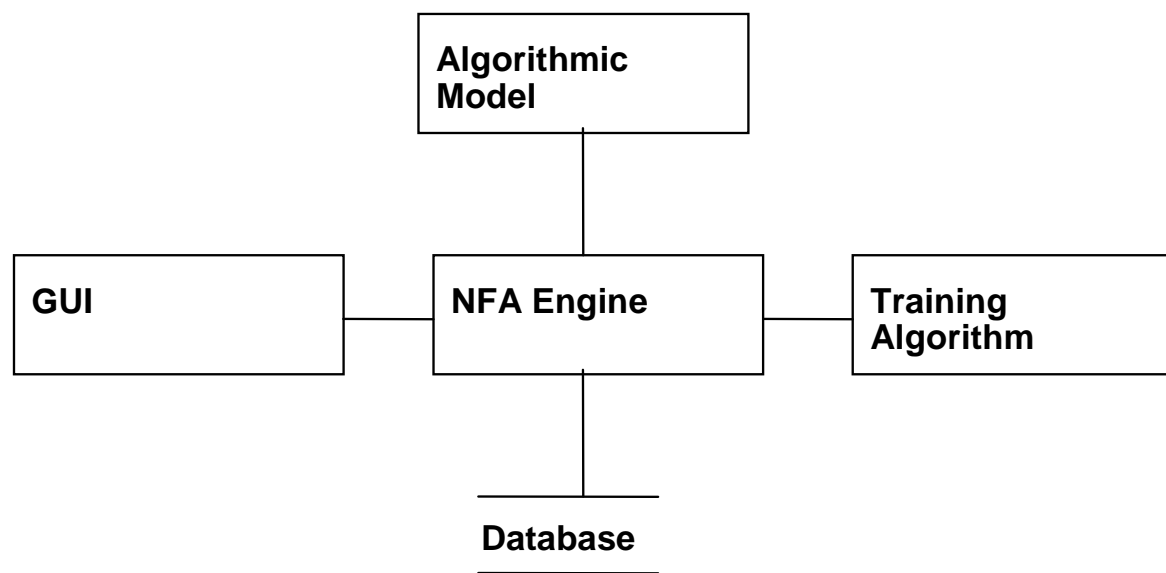
where  $N$  is the number of contributing factors,  
 $M$  is the number of other variables in the Algorithmic Model,  
 $RF$  is Factor Rating,  
 $ARF$  is Adjusted Factor Rating,  
 $NFB$  is the Neuro-Fuzzy Bank,  
 $FM$  is Numerical Factor/Multiplier for input to the Algorithmic Model,  
 $V$  is input to the Algorithmic Model,  
and  $Mo$  is Output Metric.

# Neuro-Fuzzy Bank (NFB)

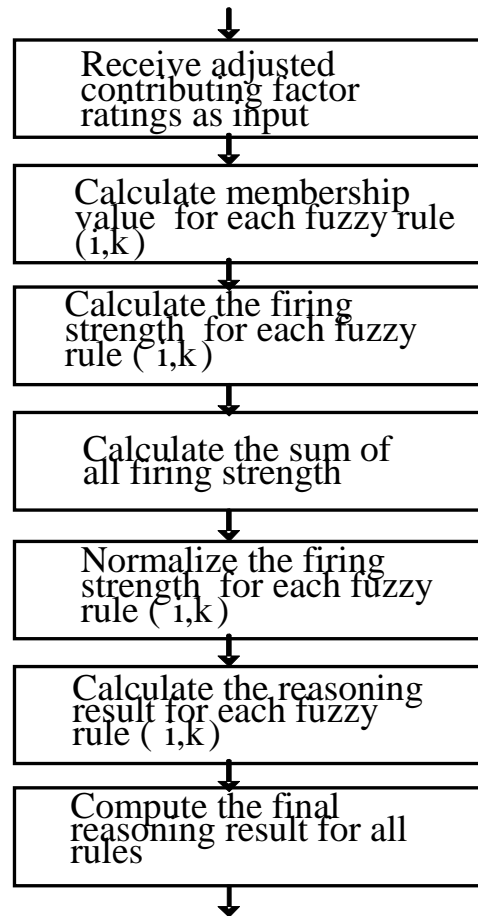


where  $ARF_i$  is Adjusted Factor Rating for contributing factor  $i$ ,  
 $A_{ik}$  is fuzzy set for the  $k$ -th rating level of contributing factor  $i$ ,  
 $w_k$  is firing strength of fuzzy rule  $k$ ,  
 $\bar{w}_k$  is normalized firing strength of fuzzy rule  $k$ ,  
 $FMP_{ik}$  is parameter value for the  $k$ -th rating level of contributing factor  $i$ ,  
 and  $FM_i$  is numerical value for contributing factor  $i$ .

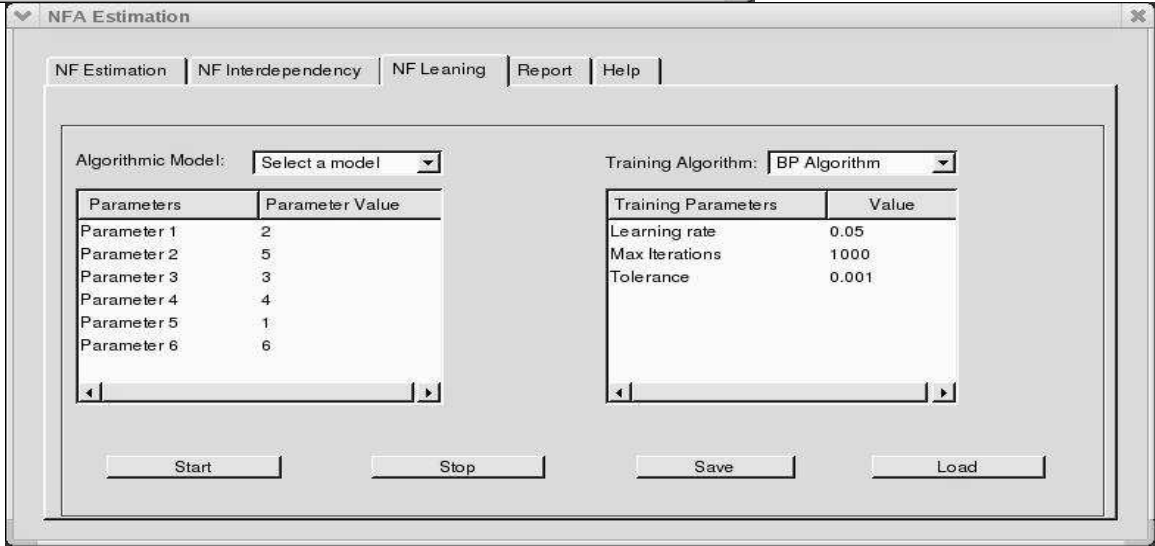
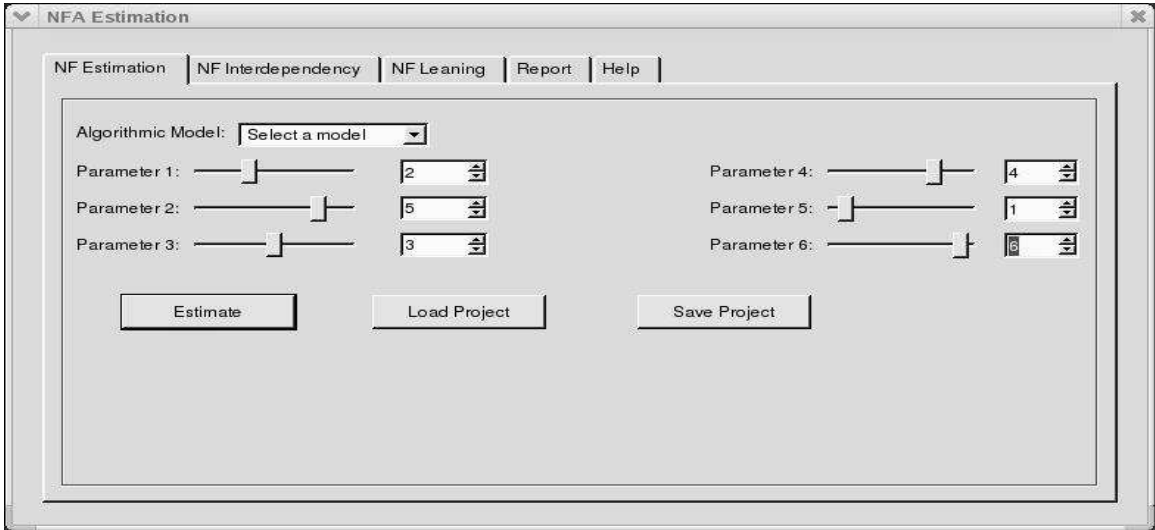
# NFA Tool Architecture



# NFA Engine



# NFA Tool



# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Validation of NFA Model

- Standard COCOMO Model using 69 project data points
- Stepwise ANOVA Model using 63 project data points
- Function Point Analysis using 184 project data points

# NFA vs COCOMO

Estimate within Actual (%)	COCOMO Model		NFA Model		Improvement
	# Projects	Accuracy	# Projects	Accuracy	
20%	49	71%	62	89%	18%
30%	56	81%	64	92%	11%
50%	65	94%	67	97%	3%
100%	69	100%	69	100%	0%

# NFA vs ANOVA

Estimate within Actual (%)	ANOVA		NFA		Improvement
	# Projects	Accuracy	# Projects	Accuracy	
20%	21	33%	28	44%	11%
25%	25	39%	33	52%	13%
30%	33	52%	37	58%	6%
50%	50	79%	52	82%	3%
100%	63	100%	63	100%	0%

# NFA vs Function Point

	<b>Exp.1</b>	<b>Exp.2</b>	<b>Exp.3</b>	<b>Exp.4</b>	<b>Exp.5</b>
<b>MMRE Function Point</b>	1.38	1.58	1.57	1.39	1.42
<b>MMRE NFA</b>	1.10	1.28	1.17	1.03	1.11
<b>Improvement (%)</b>	20%	19%	25%	26%	22%

# Agenda

1. Introduction
2. Overview of Software Estimation Models
3. The COCOMO Model
4. Experience with COCOMO
5. The SLIM Model
6. Experience with SLIM
7. Neural Network and Fuzzy Logic
8. The NFA Model
9. Experience with NFA
10. Conclusion and Future Directions

# Conclusion

- software estimation complements bottom-up estimate and project management
- deployment of estimation models take time
- management commitment is very important
- need to capture historical data
- need to faithfully input to model
- treat NFA as superset of all models

# Future Directions

- Short-term

  - validate accuracy of NFA over well-known algorithmic models

- Long-term

  - apply NFA to other aspects of estimation

- Series of potential new products

  - NF COCOMO, NF Function Point, NF SLIM, NF Size, NF Defect, NF SoS, NF SysInteg, NF Stock, NF Medical, and so forth

# References

- B. Boehm, E. Horowitz, R. Madachy, D. Reifer, B. Clark, B. Steece, A. Brown, S. Chulani, C. Abts, Software Cost Estimation with COCOMO II, Prentice Hall, July 2000.
- L. Putnam, W. Myers, Measures for Excellence, Yourdon Press Computing Series, 1992.
- S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.
- L.A. Zadeh, Fuzzy Logic, Computer, Vol 21, 1988.
- D. Ho, L.F. Capretz, X. Huang, J. Ren, Neuro-Fuzzy Algorithmic (NFA) Models and Tools for Estimation, Twentieth International Forum on COCOMO and Software Cost Modeling, University of Southern California, Los Angeles, October 2005.
- X. Huang, D. Ho, J. Ren, L.F. Capretz, Improving the COCOMO Model Using a Neuro-Fuzzy Approach, Applied Soft Computing, Elsevier Science, October, 2005.
- X. Huang, D. Ho, J. Ren, L.F. Capretz, A Soft Computing Framework for Software Effort Estimation, Soft Computing, Springer-Verlag, April 2005.
- X. Huang, L.F. Capretz, J. Ren, D. Ho, Neuro-Fuzzy Model for Software Cost Estimation, ISBN 0-7695-2015-4, Third Quality Software International Conference, Dallas, November 2003.
- X. Huang, D. Ho, L.F. Capretz, J. Ren, An Intelligent Approach to Software Cost Prediction, Eighteenth International Forum on COCOMO and Software Cost Modeling, University of Southern California, Los Angeles, October 2003.

# References

- M. Gittens, J.M. Bennett, D.Ho, Empirical Defect Modeling to Extend the Constructive Cost Model, Eighth European Software Control and Metrics Conference, Berlin, Germany, June 1997.
- D. Ho, Software Estimation Using COCOMO - the Toronto Laboratory Experience, Nortel Design Forum, December 1996.
- D.Ho, Experience Report on COCOMO and the Costar Tool from Nortel's Toronto Laboratory, Eleventh International Forum on COCOMO and Software Cost Modeling, University of Southern California, Los Angeles, October 1996.
- J.M. Bennett, D. Ho, Towards Semi-Automation of the Software Development Process, ISBN 1-56252-337-6, Third International Conference on Software Quality Management, Seville, Spain, April 1995.
- N. Panlilio-Yap, D. Ho, Lessons Learned in Deploying Software Estimation Technology and Tools, Nineteenth Annual Software Engineering Workshop, NASA Goddard Space Flight Center, Greenbelt, Maryland, December 1994.
- N. Panlilio-Yap, D. Ho, Deploying Software Estimation Technology and Tools: the IBM SWS Toronto Lab Experience, Ninth International Forum on COCOMO and Software Cost Modeling, University of Southern California, Los Angeles, October 1994.