Resource Model Studies

MODELING AND MEASURING RESOURCES

Model Validation Study

Walston and Felix build a model of resource estimation for the set of projects at the IBM Federal Systems Division.

They did this by characterizing the relationship between size and effort for projects in their environment.

They studied the relationship of other variables and how they affected effort.

They also characterized other relationships, e.g., the relationship between size and number of pages of documentation.

Do the W/F equations hold in other environments? Do the same parameters affect effort in other domains.

Are there other measures of size that might be of interest? Can there be further tailoring of their model?

MODELING AND MEASURING RESOURCES

Model Validation Study in the SEL

Object of study:relationships among variables, e.g., size and effort
Purpose: Evaluation (Valid for another environment?)
Improvement (Other variables that do better?)
Focus: Qualtiy of relationship/ fit of the data to a relationship
Point of view: Organization (management)/ researcher
Environment: SEL

Goal: Analyze the relationship between size and effort for the purpose of evaluation (improvement)of the Walston-Felix equations with respect to the quality of the fit of the data to the curve from the point of view of the organization

Object of Study Model: $E = a*Size^b$ specifically $E = 5.2L^{.91}$

Focus Model: Standard Error of Estimate, R^2

Basili/Freburger
University of Maryland, 1979

MODELING AND MEASURING RESOURCES

Model Validation Study in the SEL

Are there other parameters of interest?

e.g., DL = number of developed/delivered source lines of code (new code + 20% reused code)

M = total number of modules = FORTRAN Subroutine I

DM= total number of developed modules (all new or more than 20% new)

P= productivity = L/E

RDTODL =ratio of developed to toatal lines of code (.2 --> all old, I --> all new)

RDTODM= ratio of developed to total modules

Basili/Freburger
University of Maryland, 1979

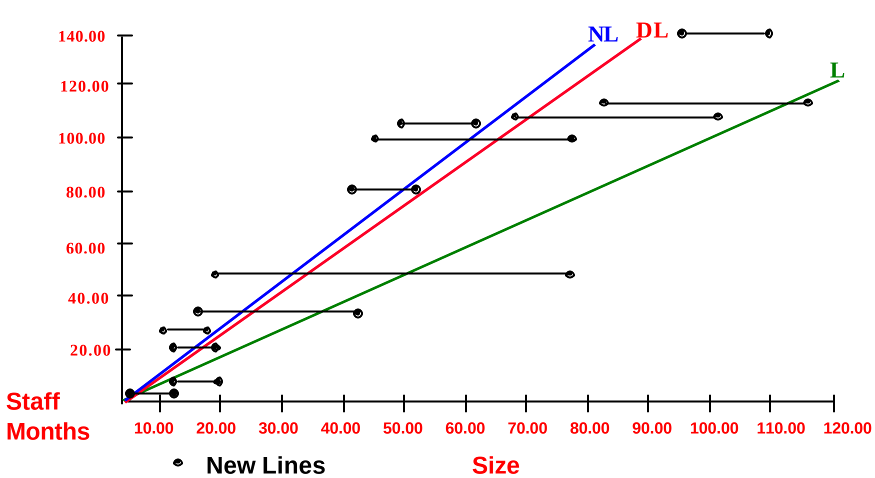
MODELING AND MEASURING RESOURCES
Model Validation Study in the SEL

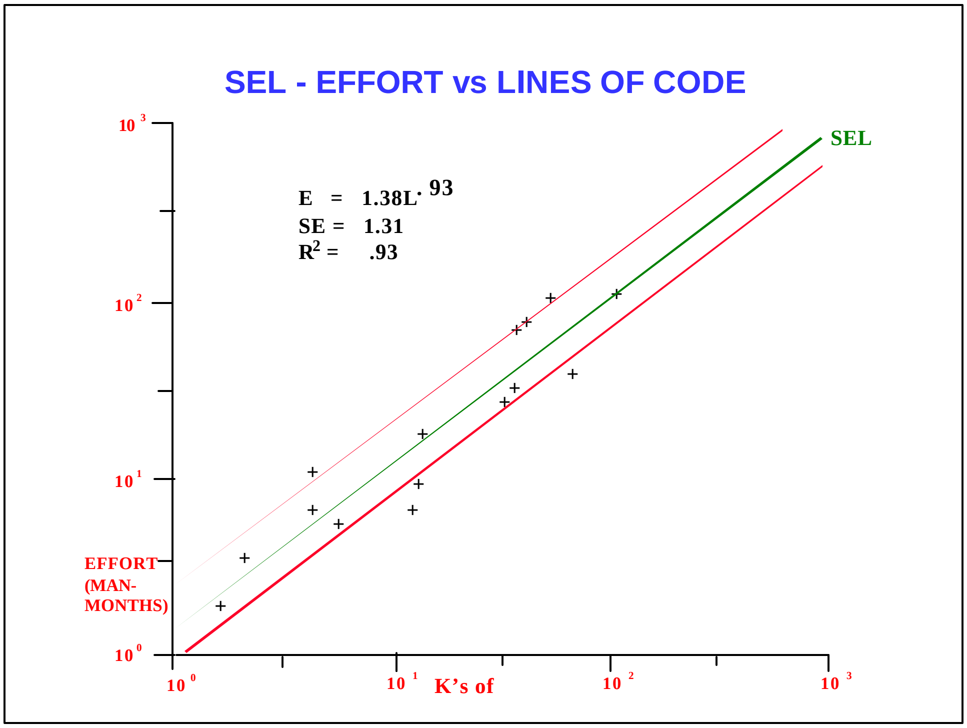
SEL Data

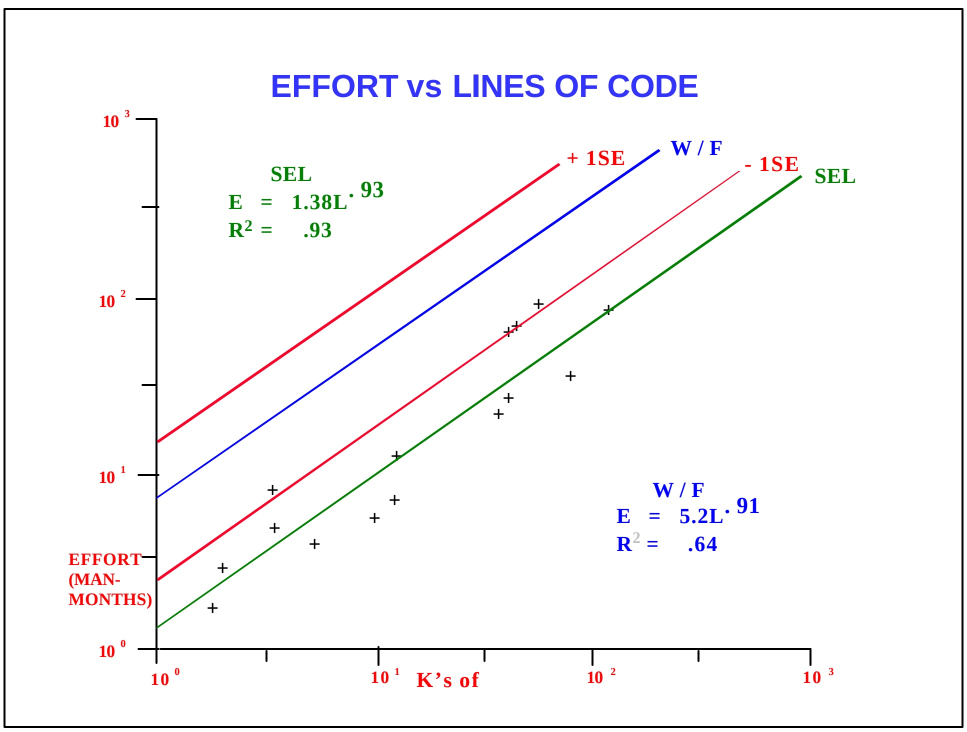
We have examined a set of projects

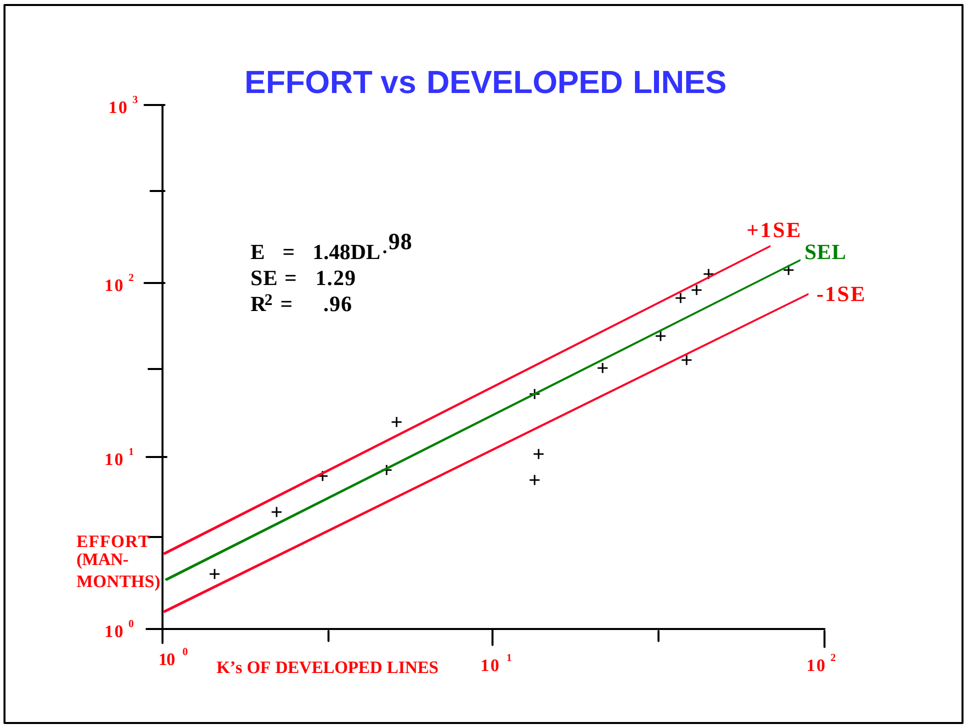
- Dealing with ground support software
- Ranging from 2K to 101K developed source lines
- Duration ranging from 4.6 to 17.4 months
- Effort ranging from 5 to 138 staff months
- Average staff size from 1 to 8 people
- Productivity from 413 to 1068 developed source lines/staff month with an average of 668 developed source lines/staff month
- Data covers design through acceptance test
- Includes manager, programmer and support staff

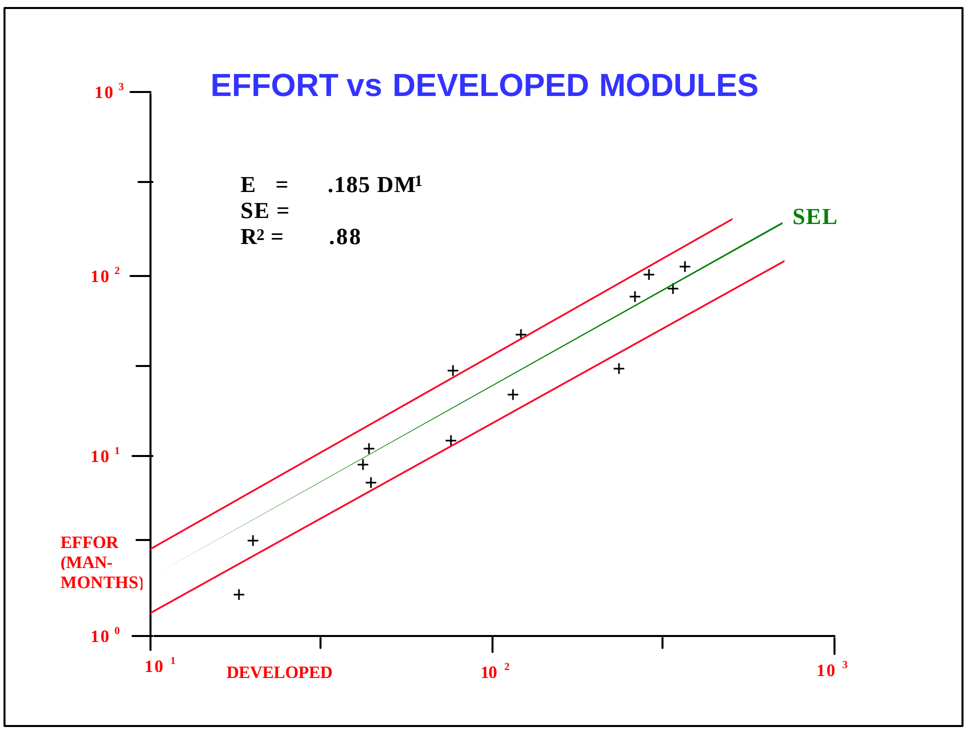
COMPARISON OF NL, DL, L



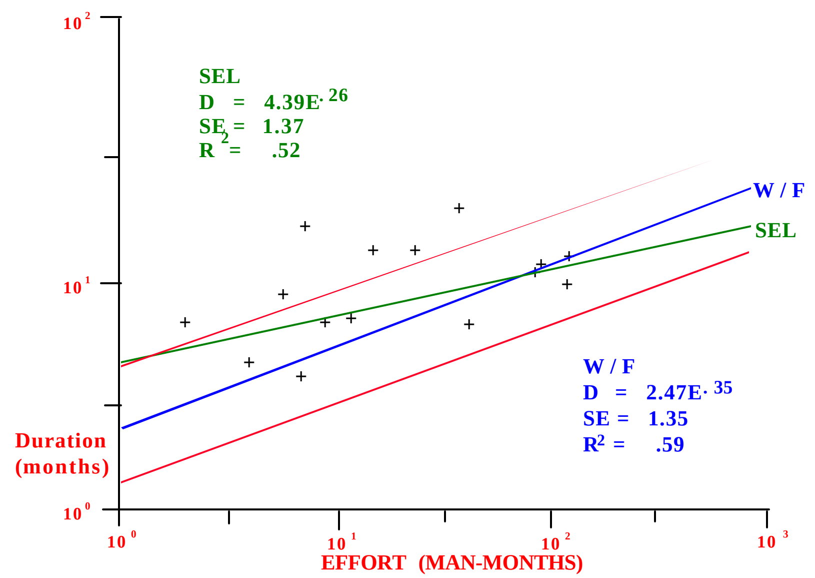

SEL - EFFORT vs LINES OF CODE
E = 1.38L.93
SE = 1.31
R2 = .93
SEL
EFFORT (MAN-MONTHS)
K's of
10 0
10 1
10 2
10 3


EFFORT vs LINES OF CODE
SEL
E = 1.38L.93
R2 = .93
+ 1SE
W / F
- 1SE
SEL
W / F
E = 5.2L.91
R2 = .64
EFFORT (MAN-MONTHS)
K's of
10 0
10 1
10 2
10 3

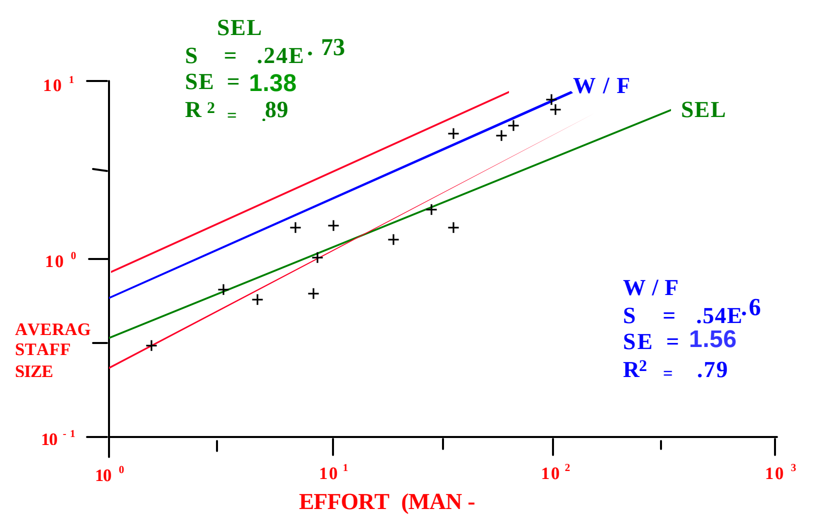

EFFORT vs DEVELOPED LINES
E = 1.48DL.98
SE = 1.29
R2 = .96
+1SE
SEL
-1SE
10 3
10 2
10 1
10 0
EFFORT (MAN-MONTHS)
10 0
K's OF DEVELOPED LINES
10 1
10 2

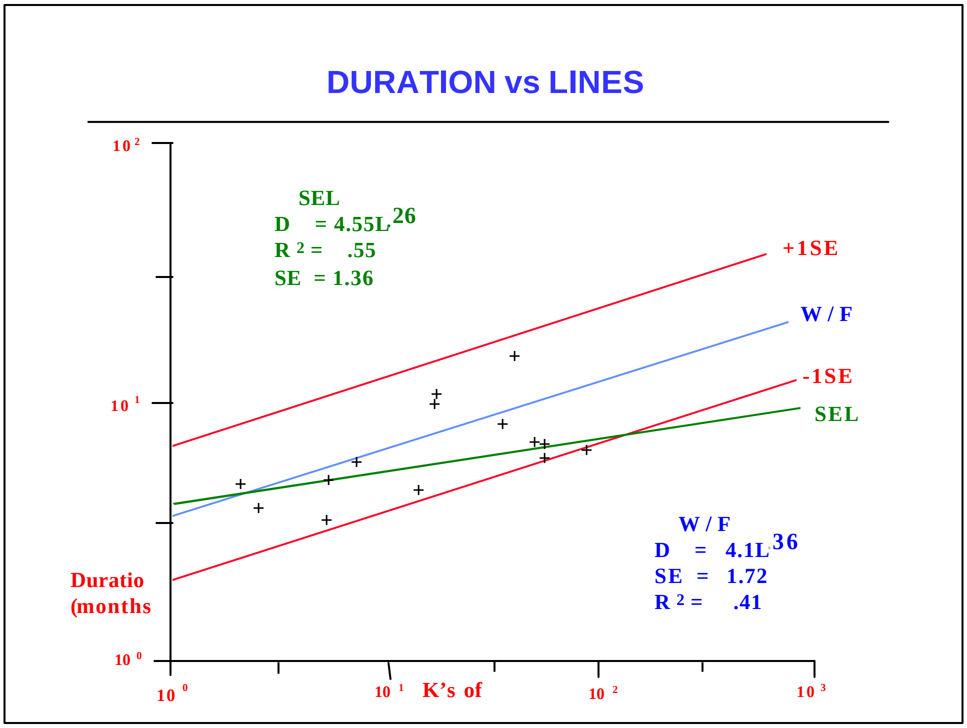

EFFORT vs DEVELOPED MODULES
E = .185 DM1
SE =
R2 = .88
SEL
10 3
10 2
10 1
10 0
EFFOR (MAN-MONTHS)
10 1
DEVELOPED
10 2
10 3

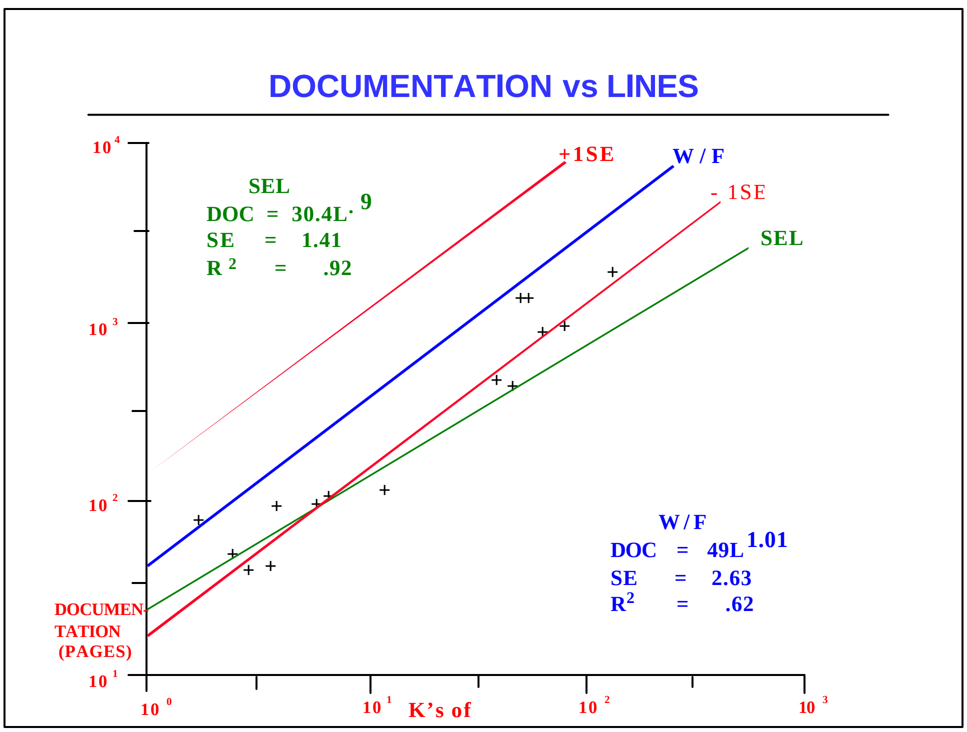
DURATION vs EFFORT



STAFF SIZE vs EFFORT




DURATION vs LINES
SEL
D = 4.55L.26
R 2 = .55
SE = 1.36
+1SE
W / F
-1SE
SEL
W / F
D = 4.1L.36
SE = 1.72
R 2 = .41
Duratio
(months
10 2
10 1
10 0
10 0
10 1
K's of
10 2
10 3


DOCUMENTATION vs LINES
SEL
DOC = 30.4L.9
SE = 1.41
R 2 = .92
+1SE
W / F
- 1SE
SEL
W / F
DOC = 49L1.01
SE = 2.63
R2 = .62
DOCUMEN-
TATION
(PAGES)
10 4
10 3
10 2
10 1
10 0
10 1
K's of
10 2
10 3

MODELING AND MEASURING RESOURCES

Model Validation Study in the SEL

Do the W/F equations hold in other environments?

Results show that:
SEL data fits within the broad context of the Walston and Felix equations
But it would be better for the SEL to build its own base model

Are there other measures of size that might be of interest?

Yes, there are other measures of size that might be more efective in the context of the environment

Do the same parameters affect effort in other domains?

Can there be further tailoring of their model?

MODELING AND MEASURING RESOURCES

Model Creation Study in the SEL

Build a mechanism for creating local models of effort estimation based upon size and other parameters, using variable tailored to the local environment

Goal: Analyze the new resource modeling for the purpose of evaluation with respect to the quality of the fit of the data to the curve from the point of view of the SEL

MODELING AND MEASURING RESOURCES
Building your Own Model

A Resource Meta-Model

Meta-Model Assumptions:

Each software development environment is different

There are factors that reflect the organizational environment rather than the project

There are other factors that reflect the differences in the projects

Approach:

Compute the background equation

Analyze the factors available to explain the difference between actual effort and effort as predicted by the background

Use the model to predict the effort for new project

Bailey/Basili
Univ. of Maryland

MODELING AND MEASURING RESOURCES
Building your Own Model

A Resource Meta-Model

Step 1

Compute the background equation

A) Pick size measure (S)
- lines (comments, executable stmts, new, total,developed)
- modules (new, total, with data blocks)
- function points

B) Choose background relationship for effort (E) and size (S)

$E = aS + b$

$E = aSb$

$E = aSb + c$

C) Calculate best fit based on
- minimizing the absolute error (least squares)
- minimizing the percent error

MODELING AND MEASURING RESOURCES
Building your Own Model

A Resource Meta-Model

Based upon 18 points

Linear Fit
$E = 1.37\ DL + 3.48$ — see = 29.3%
(minimizing absolute error) — see = 9.54mm

Log Transformation Fit
$E = 1.991\ DL^{.914}$ — see = 27.8%
(minimizing absolute error) — see = 9.11mm

Direct Fit
$E = 1.545\ DL^{.967}$ — see = 24.7%
(minimizing percent error) — see = 9.44mm

Direct Fit (with constant)
$E = .5766\ DL^{1.2} + 3.75$ — see = 20.8%
(minimizing percent error) — see = 11.68mm

MODELING AND MEASURING RESOURCES
Building your Own Model

A Resource Meta-Model

Step 2

Analyze the factors available to explain the difference between actual effort and effort as predicted by the background equation for the available data

A) Choose a set of factors (we used ~ 100)
B) Group the factors based upon relevance and experience
C) Choose a few factor groups based upon intuition and correlations with productivity and the differencebetween actual and predicted effort
D) Run a forward multiple regression (it will add a factor at a time)
- there is no rule to judge how many factors to use
- a rule of thumb is to limit the number of factors to either
  - 10% of the number of data points or
  - stop adding factors when the new factor accounts for minimal improvement in the explanation of the difference R^2)

E) Iterate A, B, C, D to try to improve the explanation of the difference

MODELING AND MEASURING RESOURCES
Building your Own Model

Factors Grouped

Total Methodology

Tree charts, Top down design, Design formalisms, Code reading
Formal documentation, Chief programmer teams, Formal test plans
Unit development folders, Formal training

Cumulative Complexity

Customer interface complexity, Application process complexity,
Customer-initiated program design changes, Program flow complexity
Internal communication complexity, External communication complexity
Data base complexity

Cumulative Experience

Programmer qualifications, Programmer experience with machine
Programmer experience with language,
Programmer experience with application
Team previously worked together on same type problem

MODELING AND MEASURING RESOURCES
Building your Own Model

Step 3

Use the model to predict the effort for the new project

A. Estimate size

B. Use background equation to predict standard effort

C. Estimate values of the factors used in the regression analysis

D. Compute the difference this project should exhibit based upon the values from C used in the multiple regression equation

E. Apply that difference to the standard effort to compute the improved effort estimate

MODELING AND MEASURING RESOURCES
Application of the Model in the SEL

We Used 17 points to predict the 18th

For our model we chose

$S = DL$ (developed lines of code)

The background relationship

$E = aS^b + C$

We chose to minimize the percent error

This yields a background equation (based upon the first 17 data points) of

(1) $E_s = .72DL^{1.17} + 3.4$

with a standard error of estimate - 1.254

For project 18 we estimated 101,000 lines yielding

$E_s = 163$ staff-months with interval (130,204)

We grouped factors to form new factors and chose two:

Meth: Total methodology (11 factors)

Complx: Complexity (5 factors)

MODELING AND MEASURING RESOURCES
Application of the Model in the SEL

We applied the multiple regression routine to compute the effort ratio

Using meth

$ER = -.036meth + 1.0 = -.224$

then the improved estimated $E_I + E_S / 1.224 = 133$ staff-months

with interval (115,154)

Using meth and cmplx

$E_R = -.036meth + .006cmplx + .86 = -.166$

then the improved estimate $E_I = E_S / 1.166 = 140$ staff-months

with interval (121,162)

The actual effort was 138 staff-months

MODELING AND MEASURING RESOURCES
Building Your Own Model

Summary

Background equation tries to express the relationship between size and effort for the average project

It should reflect all those properties that are constant across the environment

Factors reflect local differences among projects within the environment

It is hard to get good factors and good data for the factors

We could only explain half the variation with factors we used

It is a viable approach to estimating software development resource expenditures

Problems

- Need historical data
- Background equation changes with improvement

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on Productivity

We examined the relationship between productivity and various factors

Found no significant relationship between productivity and size

A large set of methodology factors showed varying degrees of positive correlation with productivity

A combined methodology factor showed a significant positive correlation with productivity

Projects with high methodology rating came from a different population than those with a low methodology rating

No other factors showed a significant positive correlation with productivity

Methodology is correlated with productivity

Bailey/Basili
5th NASA SEL Workshop

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on Productivity

Projects vary with respect to the set of software development techniques used and the extent to which they were used

There was formal training for some projects

Each project was rated with respect to

- A large set of factors
- Covering environment, methodology, experience, performance, etc.
- Values were given on a six-point scale
- Ratings were subjective
- Relative to the local environment
- Done near end of project without knowledge of the productivity results
- By NASA (McGarry), CSC (Page), and University of Maryland (Basili)

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on Productivity

Based upon a similar study by Doug Brooks (IBM/FSD)

Tried to see if methodology had a significant effect on productivity

Used a statistical test to see if the projects with high methodology use came from a different environment (with respect to productivity) than the projects with a low methodology use

The data used was based upon a relative ranking rather than an absolute rating

- The approach was
  - Divide the ratings for each technique into 3 categories: low (-1), medium (0), high (1) (Done to offset differences in scales)
  - Add the ratings to get a cumulative methodology rating
  - Divide projects into groups based upon their rating and analyze using the Mann-Whittney-U test (nonparametric statistics)

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on Productivity

No significant relationship between productivity and size (no point categorizing by size)

Methodology factors for all that showed a difference among projects,the correlations between methodology and productivity:

PDL	.26
Formal design review*	.62
Design formalism	.38
Design decision notes*	.62
Design walk-through	.28
Code walk-through	.19
Code reading*	.58
Top down design	-.19
Structured code	.02
Librarian use*	.52
Chief programmer team*	.62
Formal test plans**	.51
Heavy management involvement	-.09
Formal training*	.58
Top down code	.29

*sig. < .01 **sig. < .05

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on Productivity

Group:	Low	Medium	High
Ratings:	(-11,-9,-9,-9)	(2,2,2,1,0,-1,-3-3)	(12,11,8,5,5,3)
Productivity:	535 DL/SM	660 DL/SM	768 DL/SM

Result:

Low different from Medium U High (sig. at 0.5)

High different from Medium U Low (sig. at .03)

Group:	Low	High
Ratings:	(-11,-9,-9,-9,-3,-3,-1)	(0,1,2,2,2,3,5,5,8,11,12)
Productivity:	602 DL/SM	710 DL/SM

Result:

Low different from High (sig. at .05)

USING THE SAME DATA FOR OTHER GOALS
Relation of Other Factors with Productivity

Tried the following:

Customer interface complexity
Customer originated program design changes
Complexity of: application processing, program flow, internal communication, external communication, data base complexity, Jerry's general complexity rating.
Constraints: I/O capability, timing, main store
Programming group experience: machine familiarity, language familiarity, application experience, same type before
Hardware changes during development
Percent real time or interactive*
Percent programmer involved in specifications

All but one showed no significant correlation with productivity

*Showed significant difference at .05 level in wrong direction (i.e., higher % real time --> higher productivity)

USING THE SAME DATA FOR OTHER GOALS
The Effect of Various Factors on QUALITY

We compressed three sets of metrics into three factors:
quality, methodology, and complexity

Methodology and complexity were not significantly correlated

Quality was significantly correlated with
methodology (R = .67) and complexity (R = .64)
at less than .001 significance level

Using methodology alone to predict quality, $R^2 = .65$

There is evidence we can predict quality from methodology and complexity

Methodology is correlated with quality

Bailey/Basili

MODELING AND MEASURING RESOURCES
Static Adjusted Baseline

Static single variable effort equation acts as a baseline equation,
e.g., effort = A * $size^b$

This provides a basic estimate of effort

The initial estimate is adjusted by a set of multipliers that attempts to incorporate the effect of important product and process attributes

E.g, if the initial estimate is E = 100 staff months and the complexity of the job is rated higher than normal, a multiplier 1.1 is associated with it, yielding an adjusted estimate of 110 staff months

MODELING AND MEASURING RESOURCES
Static Adjusted Baseline

Effort = A * $size^b$

A + B determined by regression analysis ($E = 2.3\ L^{1.27}$)

Product:
Fault freedom, data base size, product complexity, adaptation from existing software

Computer:
Execution time constraints, machine storage constraints, virtual machine volatility, computer response time

Personnel:
Analyst capability, applications experience, programmer capability, virtual machine experience, programming language experience

Project:
Modern programming practices, use of software tools, required development schedule

Boehm/TRW

MODELING AND MEASURING RESOURCES
Levels of the COCOMO Model

The COCOMO model levels are:

- **Basic**, which is used for quick, early approximate estimates of software cost and schedule, but its accuracy is limited due to not using detailed data.

- **Intermediate**, which is used for better estimates of cost and schedule, because it considers software project environment factors in terms of their aggregate impact on the project parameters.

- **Detailed**, which is used for even better estimates, because it accounts for the influence of the software project environment factors on individual project phases.

We will concentrate our discussions on the basic and intermediate levels.

MODELING AND MEASURING RESOURCES
Modes of the COCOMO Model

The COCOMO model modes are:

- **Organic mode**, which is appropriate for small, stable projects

- **Embedded mode**, which is appropriate for large projects with tight constraints, that require some degree of innovation and have a complex software interface

- **Semi-detached mode**, which is appropriate for projects that fall in between the above two categories

MODELING AND MEASURING RESOURCES
Formulas the Basic COCOMO Model

The required effort (**E**) to develop the software system as a function of source size (S) (where E is expressed in Person-Months and S is expressed in KLOC):

BASIC COCOMO MODEL

MODE	**EFFORT**
Organic	$E = 2.4 * (S^{1.05})$
Semi-detached	$E = 3.0 * (S^{1.12})$
Embedded	$E = 3.6 * (S^{1.20})$

MODELING AND MEASURING RESOURCES
Formulas the Basic COCOMO Model

The project duration (**TDEV**) as a function of effort (**E**) (where TDEV is expressed in calendar months, and E in Person-Months):

BASIC COCOMO MODEL

MODE	**SCHEDULE**
Organic	$TDEV = 2.5 * (E^{0.38})$
Semi-detached	$TDEV = 2.5 * (E^{0.35})$
Embedded	$TDEV = 2.5 * (E^{0.32})$

MODELING AND MEASURING RESOURCES
The Intermediate COCOMO Model

The intermediate COCOMO is an extension of the basic COCOMO model, which used only one predictor variable, the KLOC variable

The intermediate COCOMO uses 15 more predictor variables called "**cost drivers**." The manager assigns a value to each cost driver from the range:

Very low
Low
Nominal
High
Very high
Extra high

For each of the above values, a numerical value corresponds, which varies with the cost drivers

INTERMEDIATE COCOMO MODEL
How to Use the Cost Drivers

The manager assigns a value to each cost driver according to the characteristics of the specific software project

The numerical values that correspond to the manager assigned values for the 15 cost drivers are multiplied

The resulting value **I** is the multiplier that we use in the intermediate COCOMO formulas for obtaining the effort estimates

Thus

I = RELY * DATA * CPLX * TIME * STOR * VIRT * TURN * ACAP * AEXP * PCAP * VEXP * LEXP * MODP * TOOL * SCED

Note that although the effort estimation formulas for the intermediate model are different from those used for the basic model, the schedule estimation formulas are the same

MODELING AND MEASURING RESOURCES
Formulas of the Intermediate COCOMO Model

The required effort to develop the software system (**E**) as a function of the nominal effort (**Enom**), (where E and Enom are expressed in Person-Months, and S in KLOC) is

$$E = Enom * I,$$

where:

INTERMEDIATE COCOMO MODEL

MODE	EFFORT
Organic	$Enom = 3.2 * (S^{1.05})$
Semi-detached	$Enom = 3.8 * (S^{1.12})$
Embedded	$Enom = 2.8 * (S^{1.20})$

MODELING AND MEASURING RESOURCES
Formulas of the Intermediate COCOMO Model

The number of months estimated for software development (**TDEV**) (where TDEV is expressed in calendar months, and E in Person-Months):

INTERMEDIATE COCOMO MODEL

MODE	SCHEDULE
Organic	$TDEV = 2.5 * (E^{0.38})$
Semi-detached	$TDEV = 2.5 * (E^{0.35})$
Embedded	$TDEV = 2.5 * (E^{0.32})$

MODELING AND MEASURING RESOURCES
Source Code Size Used in the COCOMO Model

The source size (**S**) is expressed in KLOC, i.e. thousands of delivered lines of code, i.e. , the source size of the delivered software (which does ***not*** include the size of test drivers or other temporary code)

If code is reused, then the following formula should be used for determining the "equivalent" software source size S_e, for use in the COCOMO model:

$$Se = S_n + (a/100) * S_u$$

where S_n is the source size of the new code, S_u is the source size of the reused code, and **a** is determined by the formula:

$$a = 0.4 * D + 0.3 * C + 0.3 * I$$

based on the percentage of effort required to adapt the reused design (D) and code (C), as well as the percentage of effort required to integrate the modified code (I)

SOFTWARE DEVELOPMENT EFFORT MULTIPLIERS

	Ratings					
Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY Required software reliability	.75	.88	1.00	1.15	1.40	
DATA Data base size		.94	1.00	1.08	1.16	
CPLX Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME Execution time constraint			1.00	1.11	1.30	1.66
STOR Main storage constraint			1.00	1.06	1.21	1.56
VIRT Virtual machine volatility [a]		.87	1.00	1.15	1.30	
TURN Computer turnaround time		.87	1.00	1.07	1.15	

[a] For a given software product, the underlying virtual machine is the complex of hardware and software (OS, DBMS, etc.) it calls on to accomplish its tasks.

(Con't)

SOFTWARE DEVELOPMENT EFFORT MULTIPLIERS

Cost Drivers	Ratings Very Low	Low	Nominal	High	Very High	Extra High
Personnel Attributes						
ACAP Analyst capability	1.46	1.19	1.00	.86	.71	
AEXP Applications exp.	1.29	1.13	1.00	.91	.82	
PCAP Program. capability	1.42	1.17	1.00	.86	.70	
VEXP Virtual machine exp [a]	1.21	1.10	1.00	.90		
LEXP Programming language experience	1.14	1.07	1.00	.95		
Project Attributes						
MODP Use of modern programming practices	1.24	1.10	1.00	.91	.82	
TOOL Use of softw. tools	1.24	1.10	1.00	.91	.83	
SCED Required development schedule	1.23	1.08	1.08	1.04	1.10	

[a] For a given software product, the underlying virtual machine is the complex of hardware and software (OS, DBMS, etc.) it calls on to accomplish its tasks.

MODELING AND MEASURING RESOURCES
Other Parameters Used in the COCOMO Model

TDEV starts when the project enters the product design phase (successful completion of a software requirements review) and ends at the end of software testing (successful completion of a software acceptance review)

E covers management and documentation efforts, but not activities such as training, installation planning, etc.

COCOMO assumes that the requirements specification is not substantially changed after the end of the requirements phase

Person-months can be transformed to person-days by multiplying by 19, and to person-hours by multiplying by 152

MODELING AND MEASURING RESOURCES COCOMO II

There is a new modeling capability, COCOMO II

It has the equation form $\mathbf{E = A(Size)}^{B}$

where A is calibrated for the local environment
and B is based upon a smaller set of variables

It is done during post architecture and early dsingn

Also, size may be measured in various ways, including function points.

HOW TO USE THE MODELS

The models should be an aid to software development management and engineering- not be taken as the sole source

An Approach

- First do a prediction
- Apply one or more models
- Examine the range of prediction offered by the model
- Compare the results

If they agree

- I can be more secure about the estimate

If they don't agree

- Examine why not
- What model assumptions did we not satisfy
- What makes this project different
- Am I comfortable with my explanation of the difference

Barry Boehm, Software Engineering Economics, Prentice Hall
Tom DeMarco, Controlling Software Projects, Yourdon Press