

# Estimating Software Reuse Costs

A. Mili, S. Fowler Chmiel

## Genesis

- An Attempt to Synthesize Current State of Art in Software Reuse Costs
- *Organizations, Components and Metrics: Foundations for Engineering Discipline of Software Reuse*, H.Mili, A. Mili, E. Addy, John Wiley and Sons, 2000/2001
- 30 Cost Models, All Talking about Reuse Costs, Cannot be All Saying Different Things.

## Classification Schem

Cost Models can be Characterized by  
Seven Features

1. Investment Cycle
2. Economic Function
3. Cost Factors Involved
4. Reuse Organization
5. Scope
6. Hypothesis
7. Viewpoint

30 is not too many, but too few.

## Synthetic Model

Model that  
Recognizes these  
Dimensions of  
Variability, and  
Provides for Them,  
rather than  
Unnecessarily  
Restricting Them.



## Model Premises

- Four stakeholders in the software reuse lifecycle
- Each stakeholder is responsible for a key decision in the process
- All decisions are based on the economic rationale
- All decisions can be modeled as investment decisions
- All decisions can be quantified by means of investment analysis functions

## Basis for Making Reuse Happen

Rather than  
Preaching,  
Lecturing,  
Finger-pointing,



Define a reward / incentive structure that integrates individual goals with reuse goals

## Formulate Incentive / Reward Structure

- Four Stakeholders: Corporate, Domain, Application, Component
- For each Stakeholder: Formulate ROI Function
- Identify Controllable Factors in ROI Functions
- Fine Tune Controllable Factors to Make ROI Positive for All Stakeholders
- Measure Stakeholder Performance / Reward Stakeholders by Means of ROI

## Optimal Incentive / Reward Structure

- Optimize Corporate ROI, under the Constraint that all Stakeholder's ROI's are Positive



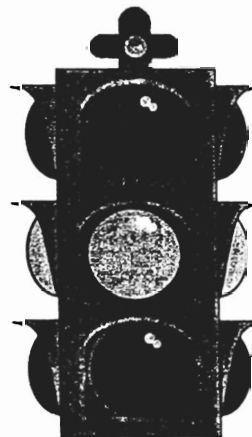
## Main Features of the Model

- Four Investment Cycles: Corporate, Domain, Application, Component
- Six Cost Factors:
  - Start Date, SD
  - Cycle Length, Y
  - Discount Rate, d
  - Investment Costs, IC
  - Episodic Benefits, B(y)
  - Episodic Costs, C(y)



## Main Features of the Model

- Seven Economic Functions
  - NPV
  - ROI
  - PI
  - ARR
  - ARBV
  - IRR
  - PB



## Hypotheses

- Non Linear Cost Effects
- Integration Costs
- Quantifying Gains (ACT differential)
- Code Inflation
- Quantifying Time-to-Market Gains

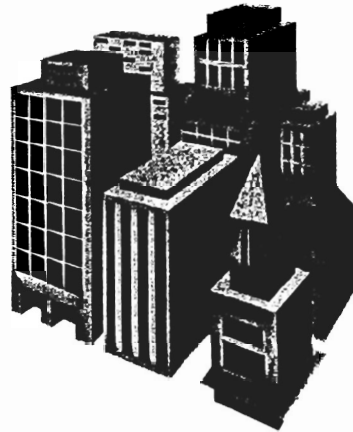


## Classification of Existing Cost Models, Comparison to Proposed Model

- Development of an Automated Cost Estimation Tool, which has an Analytical function and an Archival Function
- Development of Sample Archival Data
- “Work in Progress” Website:  
<http://www.csee.wvu.edu/reuseroi>

# Cost Structure

- Component Balance Sheet
- Domain Balance Sheet
- Application Balance Sheet
- Corporate Balance Sheet



year, $y$	Cost, $C_c(y)$	Benefit, $B_c(y)$
$y = SD$	Development for reuse	
$y > SD$	residence costs + maintenance	Sales to Projects

Component Balance Sheet

year, $y$	Cost, $C_d(y)$	Benefit, $B_d(y)$
$y = SD$	Domain Analysis + Asset develop.	
$y > SD$	asset dev. + conc. domain analysis	Asset Sales to Projects

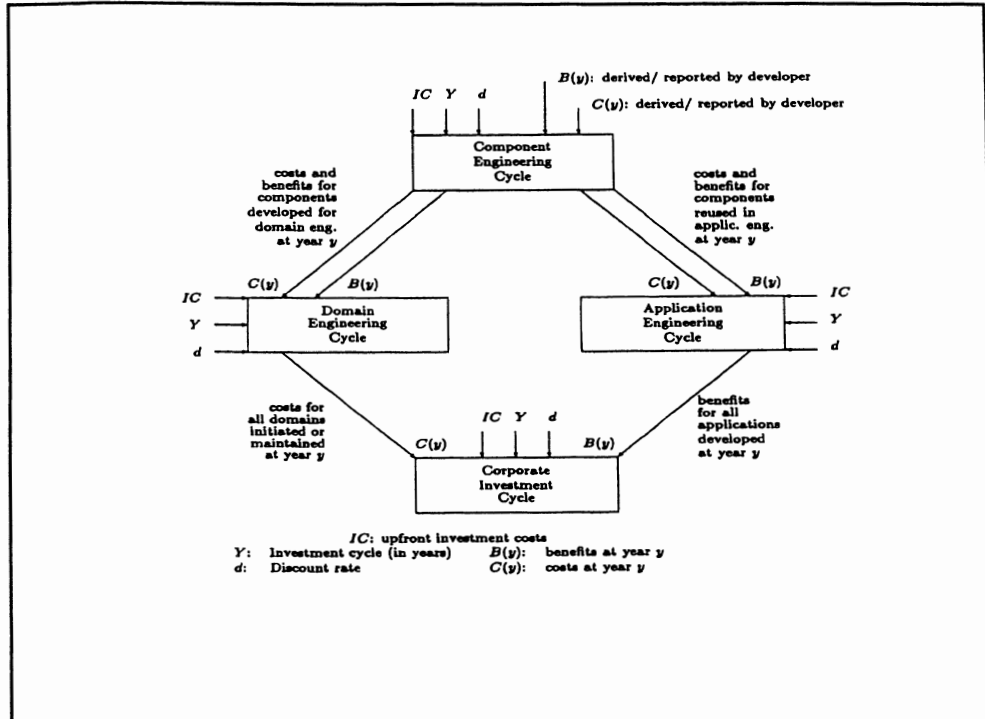
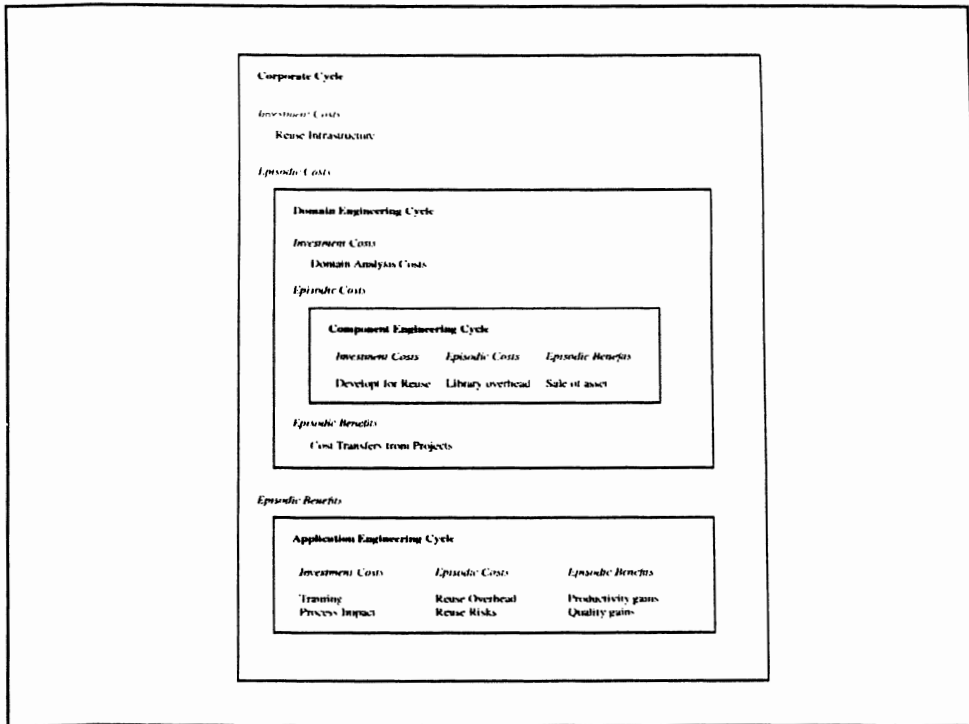
Domain Balance Sheet

year, $y$	Cost, $C_a(y)$	Benefit, $B_a(y)$
$y = SD$	Purchase of reusable assets	savings of dev. costs
$y > SD$		Quality gains

Application Balance Sheet

year, $y$	Cost, $C_p(y)$	Benefit, $B_p(y)$
$y = SD$	Infrastructure	
$y > SD$	Domain costs	Application benefits

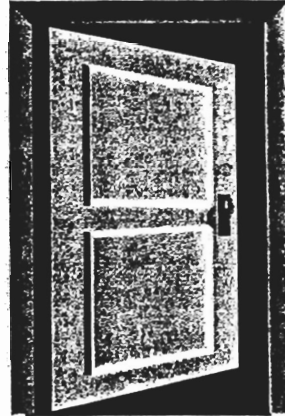
Corporate Balance Sheet



## Corporate Cycle

Initiative Started in  
1997,  
SD=1997

IC = Infrastructure  
Costs + Reuse  
Institutionalization  
Costs  
= 50 PM



## Domain Cycle

Domain Initiative Started in 1997, SD=1997

5 Reusable Assets Developed in 1997

IC=Domain Engineering Costs + Asset Development  
Costs = DA + 5 x ER=175.66 PM

5 Reusable Assets Developed in 1998

=Asset Development Costs=5\*ER=97.60 PM

Benefits: Will be Inferred from Application  
Engineering Activity

## Application Engineering Cycle

- CS98: Application Developed in 1998
  - 10 Black Box Assets
  - 20 White Box Assets
  - 100 KLOC Custom Code
- CS99: Application Developed in 1999
  - 20 Black Box Assets
  - 10 White Box assets
  - 50 KLOC Custom Code

## Component Engineering Cycle

- Reusable Asset size of 5 KLOC
  - Residence Costs for Asset of .5 PM/yr
  - Maintenance Costs of 0.1xE/yr
  - $BP(y) = 0.6 \times E$
  - $WP(y) = 0.2 \times E$
  - Frequency
- | Year      | 1997 | 1998 | 1999 | 2000 |
|-----------|------|------|------|------|
| Black Box | 0    | 1    | 2    | 0    |
| White Box | 0    | 2    | 1    | 0    |

## Estimating Component Engineering ROI

SD = 1997

IC = 29.12 PM

<b>Year</b>	<b>1997</b>	<b>1998</b>	<b>1999</b>	<b>2000</b>
C(y)	29.12	2.32	2.32	2.32
B(y)	0	12.74	17.29	0

## Estimating Domain Engineering ROI

SD = 1997

IC = 109.2 PM

<b>Year</b>	<b>1997</b>	<b>1998</b>	<b>1999</b>	<b>2000</b>
C(y)	245.70	136.50	0	0
B(y)	0	127.40	172.90	0

Estimating Application  
Engineering ROI  
CS98

SD = 1998

IC = 127.40 PM

Year	1997	1998	1999	2000
C(y)	127.40	0	0	0
B(y)	378.23	87.29	87.29	87.29

Estimating Application  
Engineering ROI  
CS99

SD = 1999

IC = 172.90 PM

Year	1997	1998	1999	2000
C(y)	172.90	0	0	0
B(y)	423.53	67.99	67.99	67.99

## Estimating Corporate Engineering ROI

SD = 1997

IC = 50 PM

Year	1997	1998	1999	2000
C(y)	295.70	136.50	0	18.00
B(y)	0	378.23	510.82	155.28

## Results

Cycle	NPV	PI
Component Engineering	-11.06 PM	.70
Application Engineering CS98	450.12 PM	4.53
Application Engineering CS99	405.86 PM	3.35
Domain Engineering	-122.87 PM	.66
Corporate Engineering	612.13 PM	1.97



OOPS:

Domain Engineering and Component Engineering NPV is Negative.

DE and CE will not Cooperate to Make this Work

## Summary Results, Revisited

We Change the Controllable Factors in Favor  
of CE and DE:

$$BP = 0.80 \times E$$

$$WP = 0.25 \times E$$

Cycle	NPV	PI
Component Engineering	10.83 PM	1.34
Application Engineering CS98	422.82 PM	3.73
Application Engineering CS99	242.06 PM	1.72
Domain Engineering	95.93 PM	1.26
Corporate Engineering	612.13 PM	1.97

## Conclusion

- Classification Scheme for Existing Software Reuse Cost Models
- Generic Cost Model that Encompasses Many Features of Existing Models
- Used as a Basis for Controlling Reuse Deployment
- Illustrated by an Example, which Shows What it Means to Make Reuse Happen.