



True Program Success™

Process to evaluate the Costs to Implement a Software Product Line

David P. Seaver

Managing Director

Information Technology Practice

PRICE Systems LLC

david.seaver@pricsystems.com

2007 Joint ISPA SCEA Conference
New Orleans, LA
June 2007

Outline

- **Background**
- **What Is a Software Product Line**
- **What are the benefits**
- **What are the investment costs**
- **Evaluating the Costs**

Background

- **NASA Goddard Space Flight Center Product Line for Space Ground Control Systems**
- **ALCATEL Transport Automation Group 1994 to 1996 Product Line for Automated Transport (automated people movers)**
- **Fidelity Investments (Product Lines)**
 - *Workflow Product Line*
 - *Common customer support and Customer UI*
 - *Net Benefits HR Applications*
 - *Common XML financial transactions*
 - *COTs Broker and Trader*
- **PRICE Systems**
 - **FBCB2 (Blue Force Tracking)**
 - **CAAS (Aircraft Cockpit HW/SW)**
 - **DoD ERP Implementations**

Software Product Line Defined

- **A software product line is a set of software intensive features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.**
- **Core Assets are those assets that form the basis for that software product line.**

Core Assets

Core assets often include but are not limited to

- Architecture
- Reusable software components
- Domain models
- Requirements statements
- Documentation and specifications
- Work plans
- Process description
- Performance models
- Schedules
- Budgets
- Test plans
- Test cases

Product Line Benefits

2007 Joint ISPA SCEA Conference
New Orleans, LA
June 2007

Software Product Line Benefits

- **Substantial production economies can be achieved when the systems in a software product line are developed from a common set of assets in a prescribed way**
- **Each product is formed by taking applicable components from the base of common assets,**
 - Tailoring them as necessary through preplanned variation mechanisms such as parameterization or inheritance
 - Adding any new components that may be necessary
 - Assembling the collection according to the rules of a common product line wide architecture.

Tailoring A Product Line

- **Parameter Specification**
- **Script Writing**
- **GUI Screen Specification**
- **Report Specification and Layout**
- **Security & Access Protocol Setup**
- **Table configuration**

Product Lines Efficiencies

- ***PRODUCT LINES AMORTIZE THE INVESTMENT IN THESE AND OTHER CORE ASSETS:***
 - requirements and requirements analysis
 - domain model
 - software architecture and design
 - performance engineering
 - documentation
 - test plans, test cases, and test data
 - people: their knowledge and skills
 - processes, methods, and tools
 - budgets, schedules, and work plans
 - Components
- ***PRODUCT LINES = STRATEGIC REUSE***

Product Line Quantified Gains

- **IMPROVED PRODUCTIVITY**
 - *by as much as 10x*
- **DECREASED TIME TO MARKET (TO FIELD, TO LAUNCH...)**
 - *by as much as 10x*
- **DECREASED COST**
 - *by as much as 60%*
- **DECREASED LABOR NEEDS**
 - *by as much as 10X fewer software developers*
- **INCREASED QUALITY**
 - *by as much as 10X fewer defects*

Investment Costs

Increased Costs
as opposed to one time project costs

Additional Costs

- **Architecture:** *Must support variations inherent in product line*
- **Software Components:** *Must be designed to be general without loss of performance; must support variation points*
- **Business Case:** *Must address a family of software products, not just one product*
- **Test Plans, Test Cases, Test Data:** *Must consider variation points and multiple instances of the product line*
- **Project Plans:** *Must be generic or extensible to support product line requirements*
- **People, Skills and Training:** *Must involve training and expertise centered around the assets and procedures associated with the product line*
- **Tools and Processes:** *Must be more robust*
- **Software Development:** *Cost to build re-usable software*

Refresh Rates

- **Refresh Rate – This rate reflects the fact that, while assets usually change very little from product to product, over several years the asset base may change completely to reflect new technology or mission demands.**
 - A rate of 10%, for example, indicates that 10% of the asset base will be replaced or undergo major modifications from year to year, while 90% of the asset base will remain static during that time.

Practical Example

2007 Joint ISPA SCEA Conference
New Orleans, LA
June 2007

Product Line Estimation Process

The following 7 questions are used to audit a software estimate.

1. Are the objectives of the estimate clear and correct?
2. Has the task been appropriately sized?
3. Are the estimated cost and schedule consistent with demonstrated accomplishments on other projects?
4. Have the factors that affect the estimate been identified and explained?
5. Have steps been taken to ensure the integrity of the estimation process?
6. Is the organizations historical evidence capable of supporting a reliable estimate?
7. Has the situation changed since the estimate was prepared?

1 Objectives

- **Does this save me time and money?**
- **Have I evaluated all the costs**
- **Which issues might have variable inputs?**

2. Appropriate Size

- **Size is a mixture of**
 - **Developed Software**
 - New Software
 - Adapted Software
 - Reused Software
 - Auto-generated
 - Auto-translated
 - **Commercial off the shelf software**
 - **Tailoring Costs**
 - **Customization Costs**
 - **Data Conversion Costs**

Example Software Size

Core Product Line		
Class of Software	Size Measure	
	Function Points	SLOC
New	5000	
Adapted	200	
Reused	100	
Auto Translated	3000	
Auto Generated	250	
COTS	2000	
% COTS Modified	15	
COTS Glue Code		2000

- **Core product usually a mix of new, reused and modified software**
- **Can also have COTS as part of the core product**
 - Expect COTs upgrade costs downstream
 - Changes to Glue Code

3. Are cost and schedule consistent with accomplishments on other projects?

- **Do we have historical data?**
 - Cost Data
 - Size Data
 - For multiple releases
 - For different domains
- **Can we normalize prior data if we do not have product line specific data**
- **If we already have stable mature processes how does that impact cost projections?**

4. Have the factors that affect the estimate been identified and explained?

- **What operations specification has the product been developed for?**
- **Is this different from target operations specification?**
 - Proprietary Normal Reliability
 - Proprietary High Reliability
 - Commercial Normal Reliability
 - Commercial High Reliability

	Normal SW Activity	Product Line Activity	Extra Effort
Manage Project	Yes	Yes	
Perform Configuration Management	Yes	Yes	Yes
Perform Joint Technical Reviews	No	Yes	
Perform Quality Assurance	Yes	Yes	Yes
Plan and Oversee	Yes	Yes	
Plan Software Development	Yes	Yes	Yes
Write Documentation	Yes	Yes	
Analyze System Requirements	Sometimes	Yes	Yes
Design System	Sometimes	Yes	Yes
Perform Assembly Integration and Test	Yes	Yes	
Software Requirements Analysis	Yes	Yes	
Software Design	Yes	Yes	Yes
Code and Unit Test	Yes	Yes	Yes
Software Integration and Test	Yes	Yes	Yes
Software Qualification Test	No	Yes	
Perform HW/SW Integration and Test	Sometimes	Yes	
Perform Software Product Evaluations	Sometimes	Yes	
Perform System Qualification Test	No	Yes	

4. Have the factors that affect the estimate been identified and explained?

- **What is the present stage of the complete product line implementation?**
- **How well is the product line implementation tracking against the business case estimates?**
- **Which product line practices are in place which are not?**
- **Are there any license fees for COTs software**
- **Are there any upgrades for COTs software**
- **Will COTs software require any modifications**

5. Have steps been taken to ensure the integrity of the estimation process?

- **Are the estimates reasonable?**
- **Are dates and schedule forced by business requirements?**
- **Was the process followed?**
- **Can we recreate the estimate with the same data if we follow the process?**

6. Is the organizations historical evidence capable of supporting a reliable estimate?

- **Is there experience with product line development?**
- **Is data available?**
- **If not are there public or 3rd party sources available?**
 - COPLIMO USC CSSE
 - True Planning PRICE Systems
- **Does the evidence validate the business case?**

Product Line Quantified Gains

- IMPROVED PRODUCTIVITY
 - *by as much as 10x*
- DECREASED TIME TO MARKET
(TO FIELD, TO LAUNCH...)
 - *by as much as 10x*
- DECREASED COST
 - *by as much as 60%*
- DECREASED LABOR NEEDS
 - *by as much as 10X fewer software developers*
- INCREASED QUALITY
 - *by as much as 10X fewer defects*

7. Has the situation changed since the estimate was prepared

- **Marketplace**
- **Technology**
- **Funding**
- **Customers**

Backup Slides

2007 Joint ISPA SCEA Conference
New Orleans, LA
June 2007

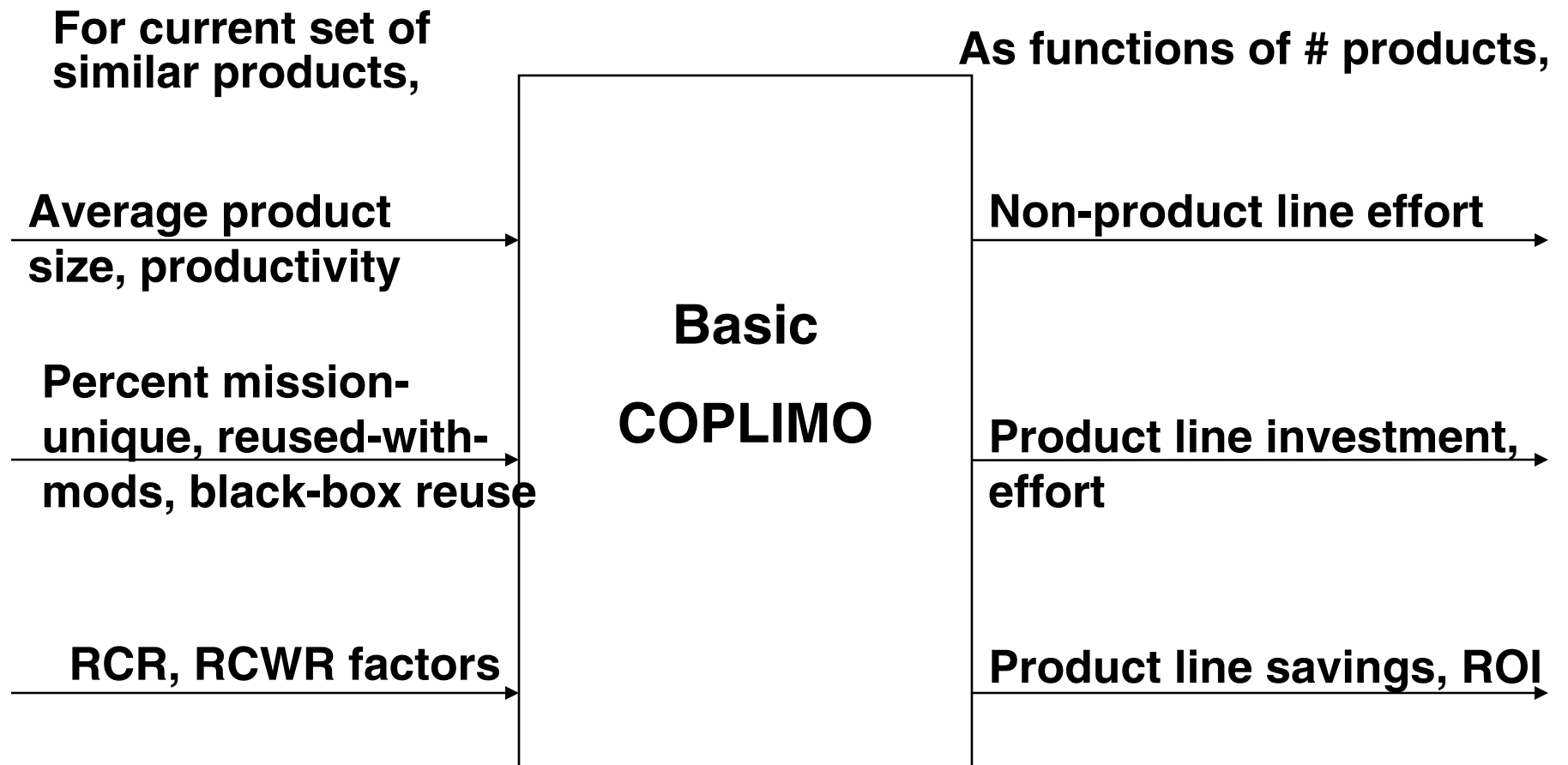
COPLIMO: The Constructive Product Line Investment Model


Ye Yang, Barry Boehm, Ray Madachy, USC

USC-CSE ARR'04

Mar. 16th, 2004

Basic COPLIMO Inputs and Outputs



 Center For Software Engineering	Basic COPLIMO	
	Project Name: A_PLI Preparer: Ye Yang Product Line Domain: Aircraft Date: 03/05/2004 Comments:	

Average SW productivity (AVPROD):	300	(SLOC/PM)
Average product size (AVSIZE):	50000	(SLOC)

Expected reuse category percentages (adding to 100%):

Percent of software unique to each application (PFRAC%):	40	(%)
Percent of software adapted from product line (AFRAC%):	30	(%)
Percent of software reused from product line (RFRAC%):	30	(%)

Expected Relative Costs of Reuse (RCR):

For unique s/w (RCR-PFRAC):	100	(%)
For adapted s/w (RCR-AFRAC):	40	(%)
For reuse s/w (RCR-RFRAC):	5	(%)

Expected Relative Costs of Writing for Reuse (RCWR):

RCWR:	1.7
-------	-----

Basic COPLIMO Output Summary

Summary of Inputs:

AVPROD	300	
AVSIZE	50000	(SLOC)
PFRAC%	40	(%)
AFRAC%	30	(%)
RFRAC%	30	(%)
RCR-PFRAC	100	(%)
RCR-AFRAC	40	(%)
RCR-RFRAC	5	(%)
RCWR	1.7	

7 year Product Line Effort Savings:

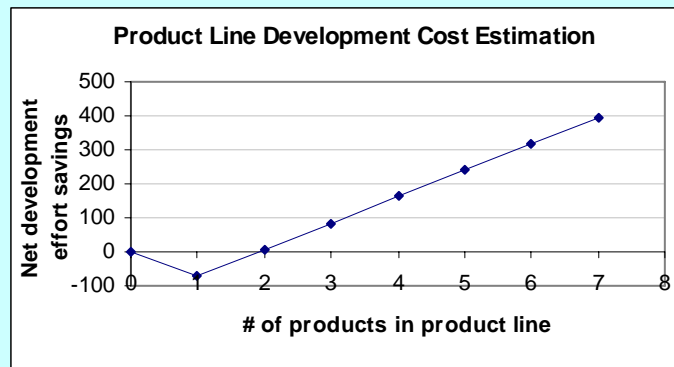


Table of Results:

# of Products	0	1	2	3	4	5	6	7
Unique SLOC	0	20000	40000	60000	80000	100000	120000	140000
Adapted SLOC	0	15000	30000	45000	60000	75000	90000	105000
Reused SLOC	0	15000	30000	45000	60000	75000	90000	105000
Total Non-PL SLOC	0	50000	100000	150000	200000	250000	300000	350000
Non-PL Effort (PM)	0	166.667	333.333	500	666.667	833.333	1000	1166.667
1-Product Equiv. SLOC	0	71000	26750	26750	26750	26750	26750	26750
1-Product Equiv. Effort	0	236.667	89.1667	89.1667	89.1667	89.1667	89.1667	89.1667
Cum. Equiv. PL SLOC	0	71000	97750	124500	151250	178000	204750	231500
Cum. PL Effort	0	236.667	325.833	415	504.167	593.333	682.5	771.6667
PL Effort Savings	0	-70	7.5	85	162.5	240	317.5	395
PL Reuse Investment	0	70						
Return on Investment	N/A	-1	0.10714	1.21429	2.32143	3.42857	4.53571	5.642857

PRICE TruePlanner - [SW Product Line]

File Edit View Tools Window Help

PBS

Product Breakdown Structure [PBS]

- 1 SW Product Line
- 2 Investment Costs
- 3 Software Component
- 4 Product Sales
- 5 Sale 1
- 6 Software COTS
- 7 Sale 2
- 8 Software COTS
- 9 Sale 3
- 10 Software COTS
- 11 Product Refresh
- 12 Upgrade 1
- 13 Upgrade 2
- 14 Upgrade 3
- 15 Oracle DBMS upgrade

Simple Estimate: Software Component

Cost Objects Input Sheet Results Chart

Name: Software Component

		Value	Unit	Rate
1	Start Date	12/18/2006		
2	-			
3	Application Type	Financial Operation System		
4	Functional Complexity	3.33		
5	Organizational Productivity	1.540		
6	Development Team Complexity	2.03		
7	-			
8	Size Units	Function Points		
9	New Code Size	5,000		
10	Language	Java		
11	External Integration Complexity	2.00		
12	-			
13	Activity Set	<Inherited>		
14	Resource Set	<Inherited>		

David Seaver
david.seaver@pricesystems.com