Estimating for Lifecycle and Product Line Affordability

Barry Boehm, JoAnn Lane,
Supannika Koolmanojwong, USC-CSSE
Ray Madachy, NPS

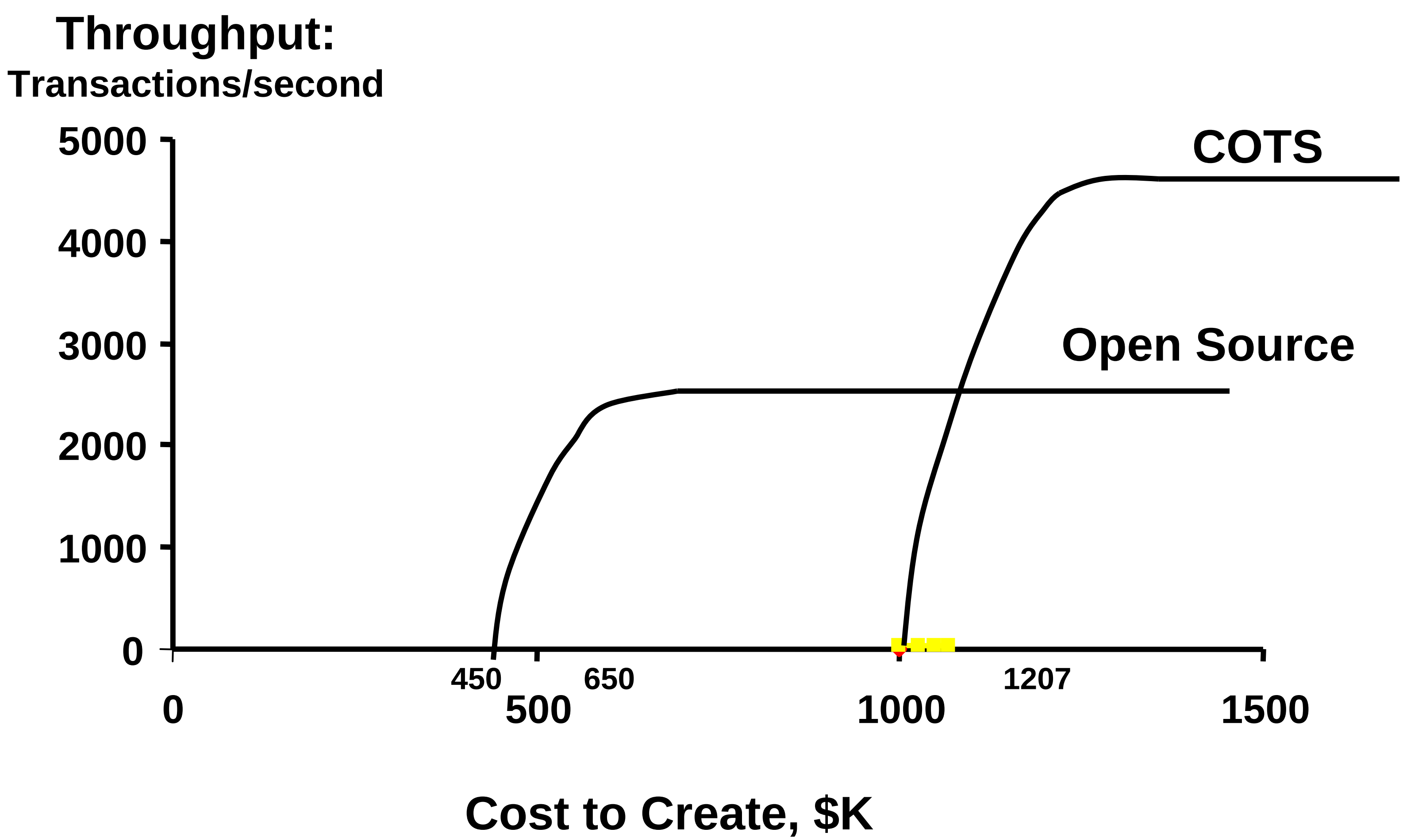
SCEA-ISPA Joint Conference
June 2012

Outline

- Perspectives on Affordability
  - Relations to value propositions and constraints
  - Affordability context considerations
- Utility of Total Ownership Cost Approaches
  - For a single system's life cycle
  - For the life cycles of a family of systems
- Conclusions

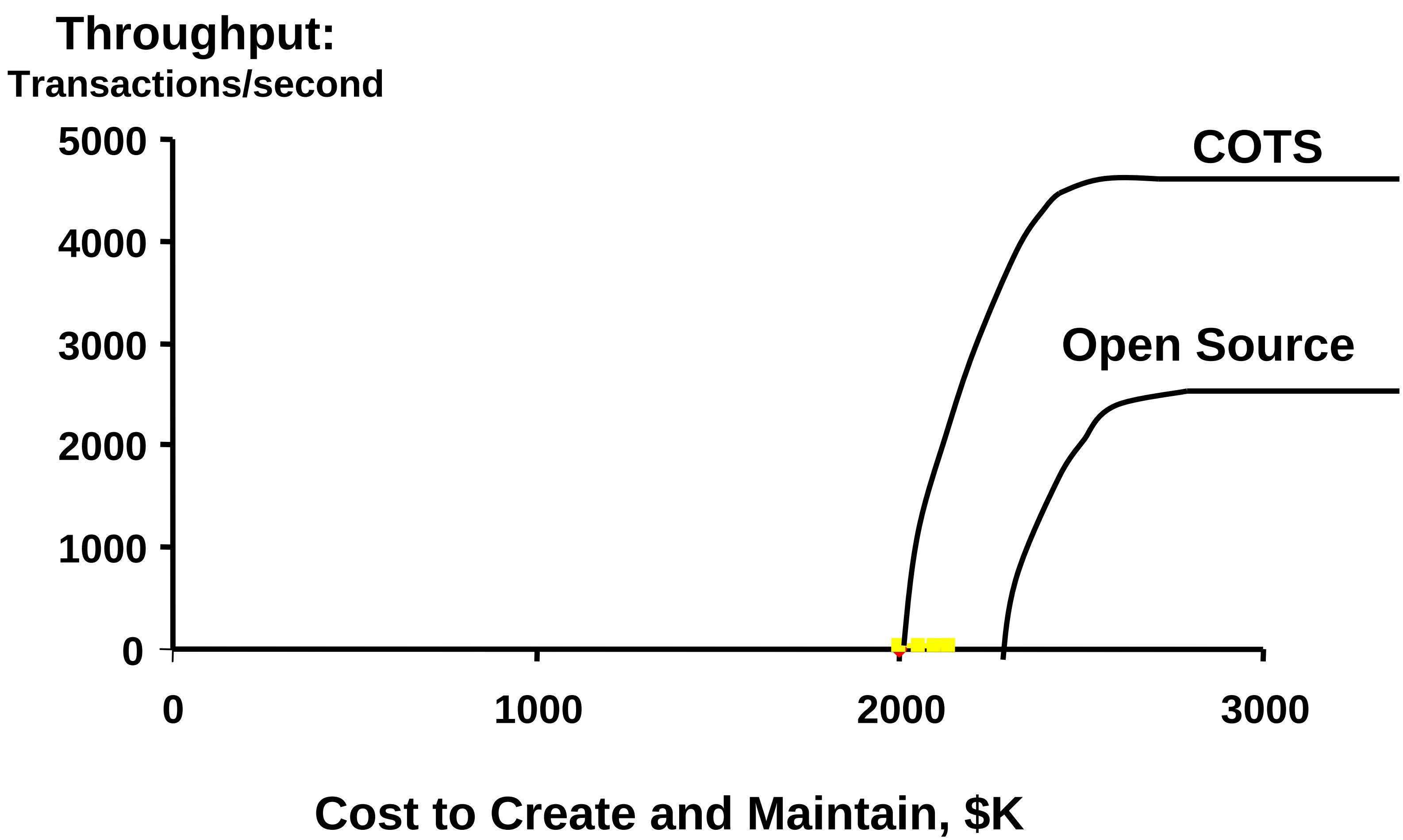
Which Is More Affordable?

Important to consider value, constraints



Which Is More Affordable?

Important to consider total ownership cost



Outline

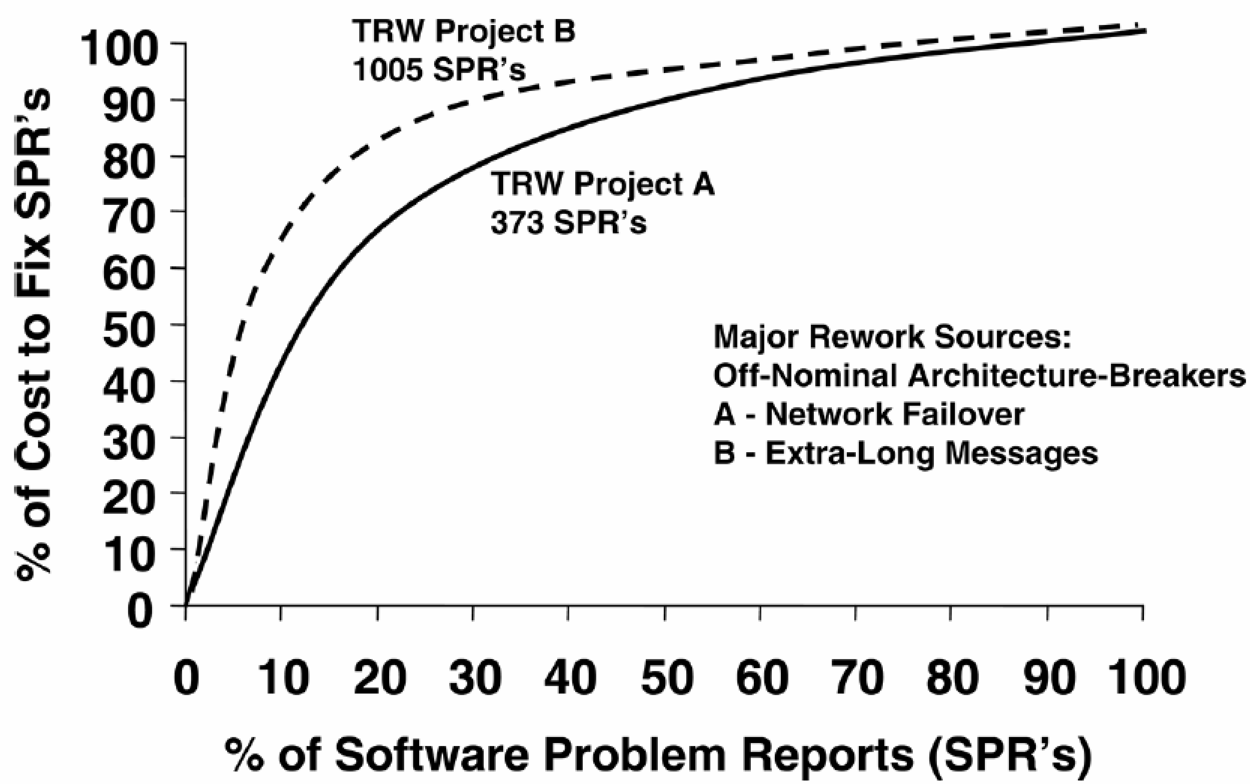
- Perspectives on Agility and Affordability
  - Primary agility failure modes
  - Affordability context considerations

→ Utility of Total Ownership Cost Approaches
  - For a single system's life cycle
  - For the life cycles of a family of systems

- Conclusions

Overfocus on Acquisition Cost

C4ISR Contracts: Nominal-case requirements; 90 days to PDR



Rework Sources Analysis: Projects A and B

- Change processing over 1 person-month = 152 person-hours

Category	Project A	Project B
Extra long messages		3404+626+443+328+244= 5045
Network failover	2050+470+360+160= 3040	
Hardware-software interface	620+200= 820	1629+513+289+232+166= 2832
Encryption algorithms		1247+368= 1615
Subcontractor interface	1100+760+200= 2060	
GUI revision	980+730+420+240+180 =2550	
Data compression algorithm		910
External applications interface	770+330+200+160= 1460	
COTS upgrades	540+380+190= 1110	741+302+221+197= 1461
Database restructure	690+480+310+210+170= 1860	
Routing algorithms		494+198= 692
Diagnostic aids	360	477+318+184= 979
TOTAL:	**13620**	**13531**

C4ISR Project C: Architecting for Change
USAF/ESC-TRW CCPDS-R Project*

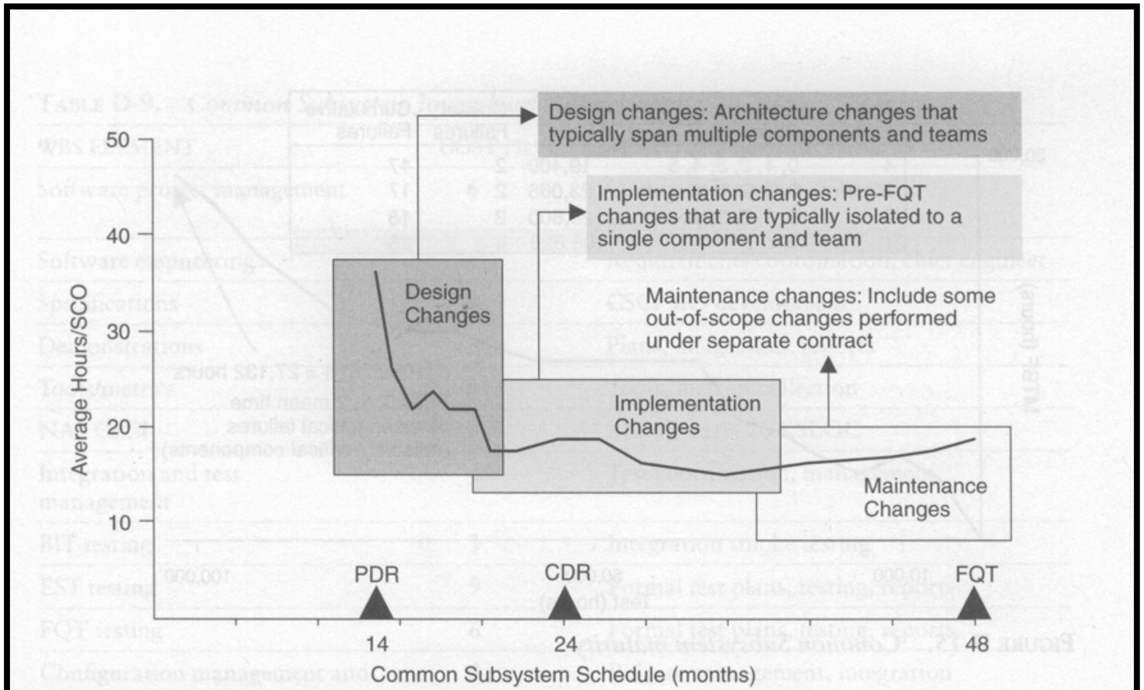


FIGURE D-14. *Common Subsystem adaptability*

When investments made in architecture, average time for change order becomes relatively stable over time…

*** Walker Royce, *Software Project Management: A Unified Framework*. Addison-Wesley, 1998.**

Current TOC-Single System Model

The simple initial TOC-SS model has the following inputs:

%D: The % of development cost invested in Design for Flexibility

System Size: For software, the equivalent KSLOC (thousands of source lines of code)

- **For hardware, the COSYSMO size parameter: complexity-weighted numbers of requirements, interfaces, operational scenarios, and algorithms [Valerdi, 2005].**

#F: The number of years that the system undergoes field changes

%FC: The percentage of the fielded system size undergoing change

The TOC-SS model has the following outputs:

TOC (Devel): The TOC for development

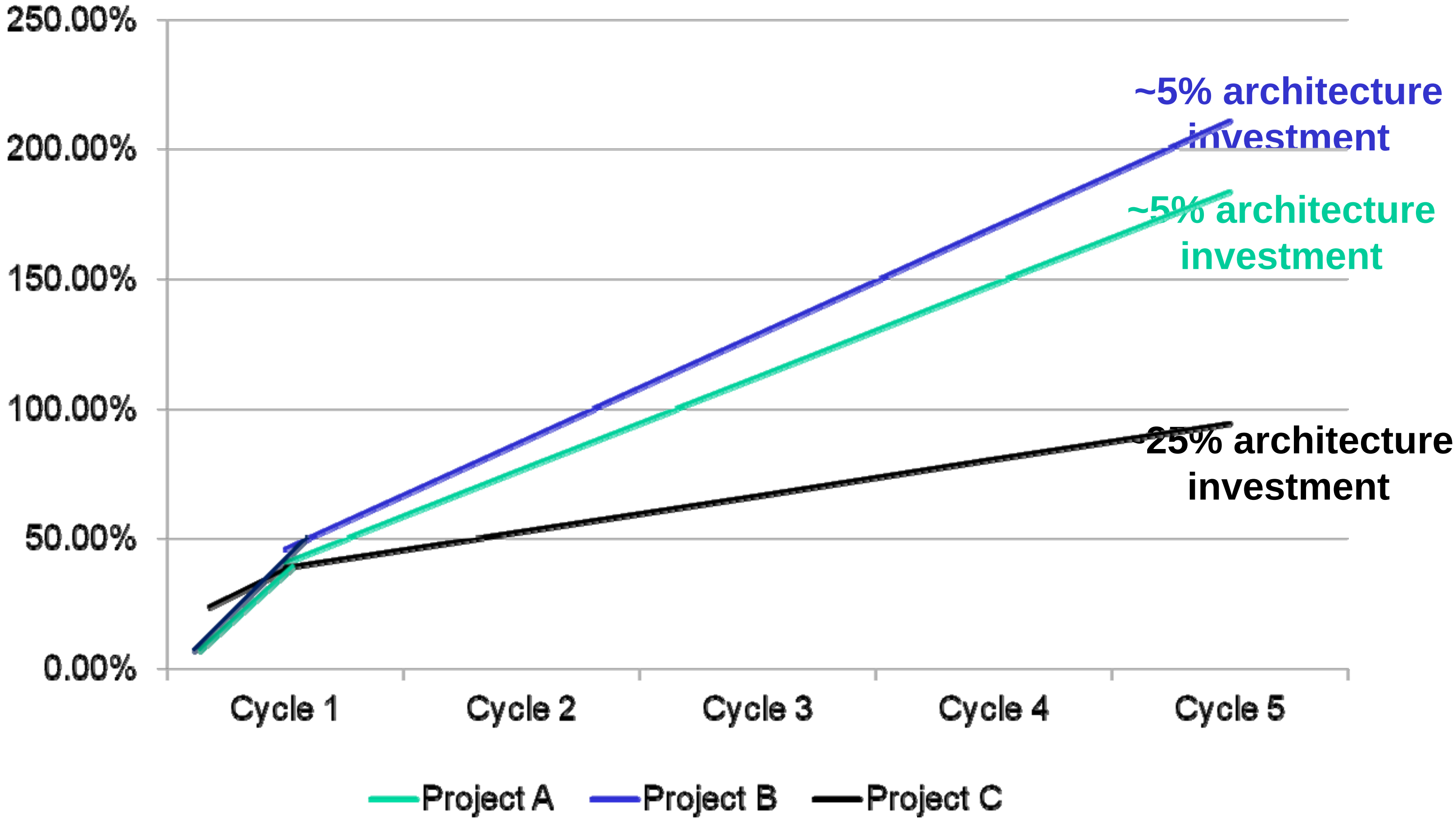
TOC (Devel + K): TOC (Devel) + TOC (K years of fielding), K = 1, …, #F

University of Southern California
Center for Systems and Software Engineering

Single-System TOC Model Example

	A	B	C	D	E
1	Input Parameters	System			
2		A	B	C	
3	Software Size (KSLOC)	100	100	355	
4	# Change Requests/Release	373	1005	1600	
5	# Change Requests (I&T only)				
6	# I&T Change Requests/Release/ > 1 PM	27	22		
7	# Total Change Requests/Release/ > 1 PM			16	
8	Change Request Fix Time (See assumption #2)	261	356	263	
9	Total Effort (Person Months)	731	865	1900	
10	% Arch, RESL	5%	5%	25%	
11	% Rework, RVOL	35.70%	41.16%	13.85%	
12					
13	Cumulative Total Cost of Ownership	Project A	Project B	Project C	
14	Cycle 1	40.70%	46.16%	38.85%	
15	Cycle 2	76.41%	87.31%	52.70%	
16	Cycle 3	112.11%	128.47%	66.55%	
17	Cycle 4	147.82%	169.62%	80.40%	
18	Cycle 5	183.52%	210.78%	94.25%	

Relative* Total Ownership Cost (TOC)



*** Cumulative architecting and rework effort relative to initial development effort**

Use of life cycle cost ratios (%O&M)

- Hardware [Redman 2008]
  - 12% -- Missiles (average)
  - 60% -- Ships (average)
  - 78% -- Aircraft (F-16)
  - 84% -- Ground vehicles (Bradley)
- Software [Koskinen 2010]
  - 75-90% -- Business, Command-Control
  - 50-80% -- Complex platforms as above
  - 10-30% -- Simple embedded software
- Apply lack-of-flexibility factor to O&M component

Outline

- Perspectives on Affordability
  - Relations to value propositions and constraints
  - Affordability context considerations
- Utility of Total Ownership Cost Approaches
  - For a single system's life cycle
  - ➔ For the life cycles of a family of systems
- Conclusions

Systems Product Line TOC Model

For Set of Products:

- Average Product Cost
- Annual Change Cost
- Ownership Time
- Percent Mission-Unique, Adapted, Reused
- Relative Cost of Developing for PL Flexibility via Reuse
- Relative Costs of Reuse

→ **Systems ProductLine TOC Model** →

As Functions of # Products, # Years in Life Cycle:

- PL Total Ownership Costs
- PL Flexibility Investment
- PL Savings (ROI)

University of Southern California
Center for Systems and Software Engineering

Product Line Engineering and Management

SYSTEMS ENGINEERING Research Center

Preferences

Systems Product Line Flexibility Value Model

Welcome SERC Collaborator

Open | Save | Save As

System Costs

Average Product Development Cost (Burdened $M) 5 | Ownership Time (Years) 3

Annual Change Cost (% of Development Cost) 10 | Interest Rate (Annual %) 7

Product Line Percentages

Unique % 40

Adapted % 30

Reused % 30

Relative Costs of Reuse (%)

Relative Cost of Reuse for Adapted 40

Relative Cost of Reuse for Reused 5

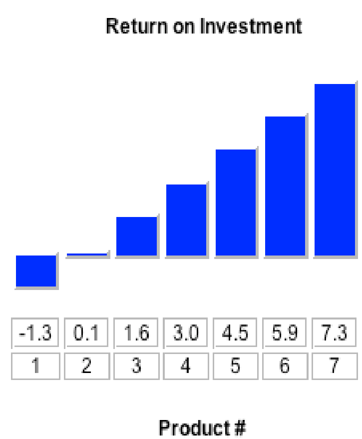
Investment Cost

Relative Cost of Developing for PL Flexibility via Reuse 1.7

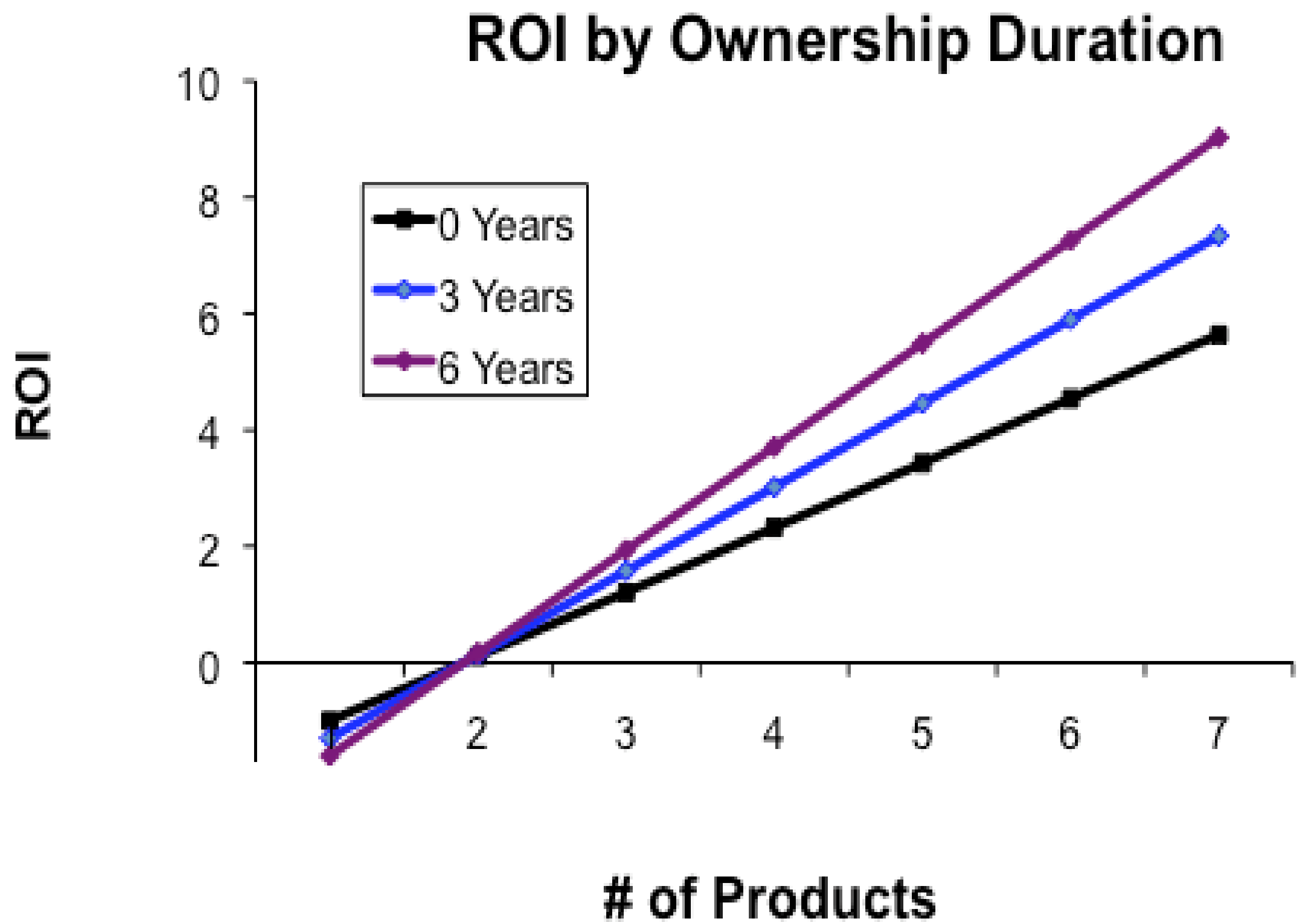
Calculate

Results

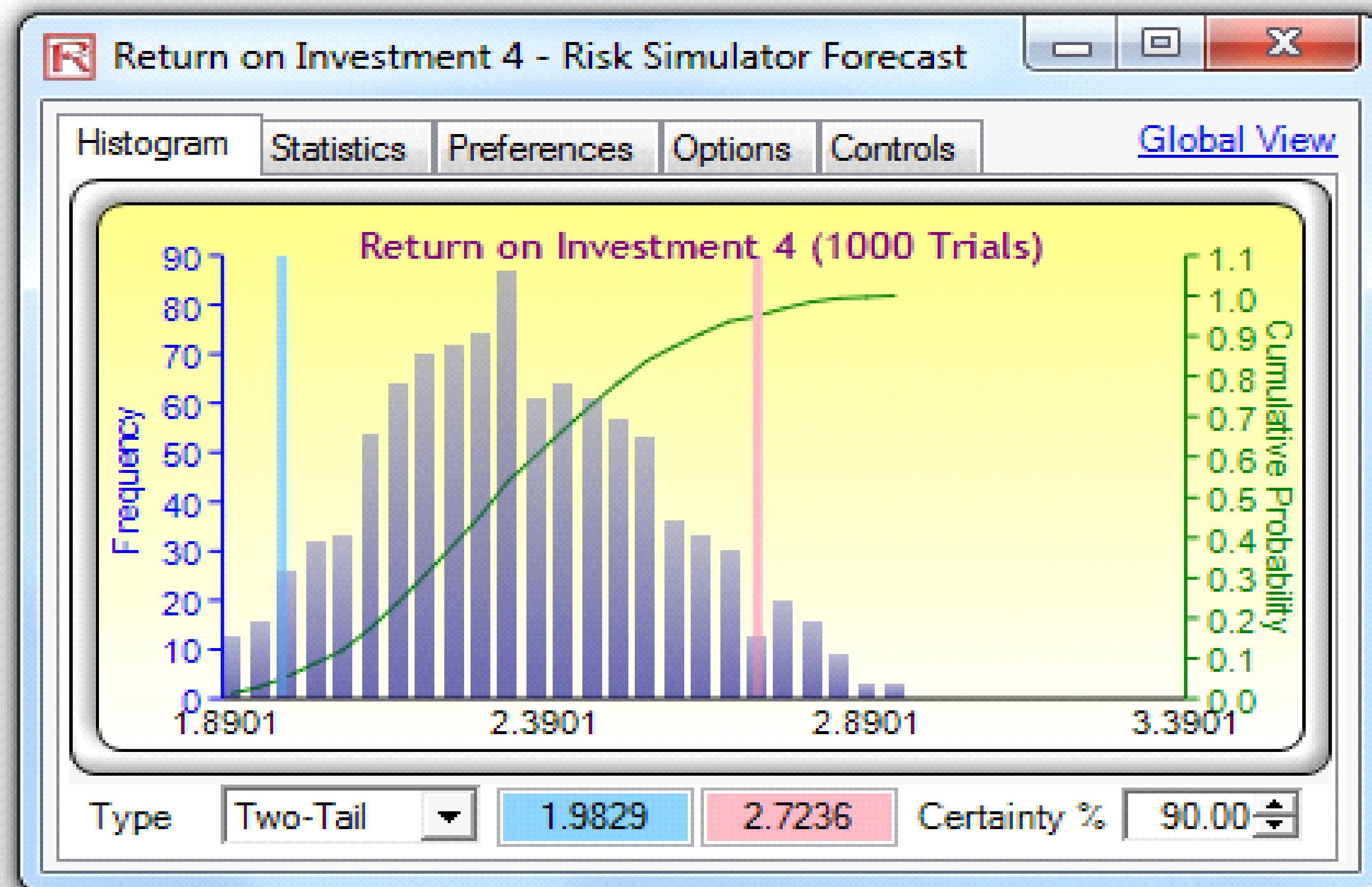
# of Products	1	2	3	4	5	6	7
Development Cost ($M)	$7.1	$2.7	$2.7	$2.7	$2.7	$2.7	$2.7
Ownership Cost ($M)	$2.1	$0.8	$0.8	$0.8	$0.8	$0.8	$0.8
Cum. PL Cost ($M)	$9.2	$12.7	$16.2	$19.7	$23.1	$26.6	$30.1
PL Flexibility Investment ($M)	$2.1	$0	$0	$0	$0	$0	$0
PL Effort Savings	($2.7)	$0.3	$3.3	$6.3	$9.4	$12.4	$15.4
Return on Investment	-1.30	0.14	1.58	3.02	4.46	5.90	7.34



Product Line Payoff Increases with Lifetime



Extension with NPS KVA Capabilities



Conclusions

- Affordability increasingly competition-critical
  - Need to balance cost, schedule, performance, functionality
- Some improvement avenues available
  - Total Ownership Cost Analysis of Alternatives
    - Identify and architect to encapsulate sources of change
  - Product Line Engineering and Management
  - Concurrent vs. Sequential Engineering
    - Using cost-effectiveness, evidence-based decision points
  - Value-Based Engineering
    - Vs. assuming equal-value requirements, tests, defects
- No one-size-fits-all solution