

University of Southern California
Center for Software Engineering

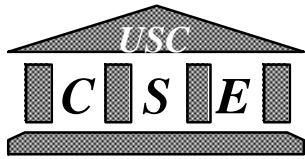
Software Reuse Economics

Barry Boehm

DARPA Workshop

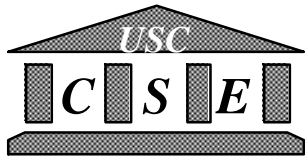
January 14, 1997

<http://sunset.usc.edu>



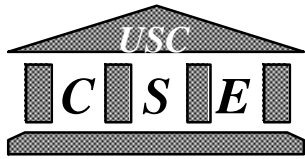
Outline

- Motivation
- Software Reuse Economics Framework
 - Return on Investment Elements
 - Investments and Payoff Elements
 - Some Useful Proxies: RCWR and RCR
- Bottom Line: Lines of Code In Service
- Conclusions and References



Outline

- Motivation
- ➔ • Software Reuse Economics Framework
 - Return on Investment Elements
 - Investments and Payoff Elements
 - Some Useful Proxies: RCWR and RCR
- Bottom Line: Lines of Code In Service
- Conclusions and References

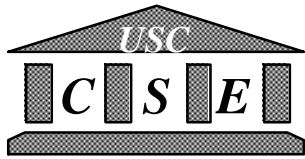


Software Reuse Economics Framework

- Investment: Cost of developing for reuse
- Payoff: Net value added via reuse
- Return on Investment

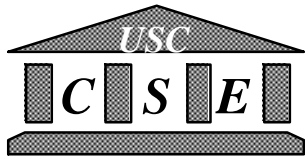
$$\text{ROI} = \frac{\text{NPV}(\text{Payoff}) - \text{NPV}(\text{Investment})}{\text{NPV}(\text{Investment})}$$

- NPV: Net Present Value
 - discounting future cash flows



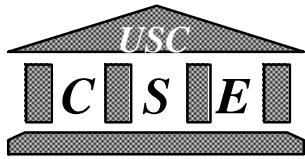
Command Center Functionality Percent of System

	<u>Common to All</u>	<u>Common to Some</u>	<u>Unique</u>	<u>Total</u>
External Interfaces	1	1		2
Message Processing		2		2
Mission Application		10	5	15
Database Management	20			20
System Management	12			12
Exercise Control	6			6
Geographic Information	25			25
Table Generation	4			4
Alarm Generation	4			4
Briefing System	2			2
Other		2	3	5
Total	77	15	8	100



Actual ROI on PRISM Programs

<u>Program</u>	<u>Savings</u>
Tact Forecast	\$11 M
Comm Weather	\$7 M
QRC Terminal	\$1 M
USTRANSCOM	\$3.8M
Security Guards	<u>\$9 M</u>
Total Savings	\$26.4 M
Actual ROI Based on \$7M CRTT investment-->	3.77



Investment and Payoff Elements

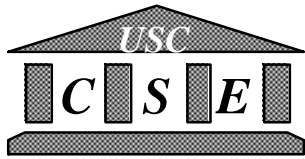
$$\text{NVP}(\text{Investment}) = \sum_i \text{NPV} [I(t_i)]$$

$$\text{NPV}(\text{Payoff}) = \sum_j \text{NPV} [\text{RB}(t_j)] - \sum_k \text{NPV} [\text{RC}(t_k)]$$

$I(t_i)$: time series of reusable asset investments

$\text{RC}(t_k)$: time series of asset reuse costs

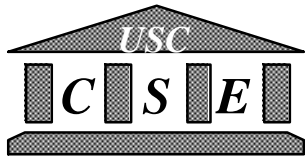
$\text{RB}(t_j)$: time series of asset reuse benefits



Sources of Reusable Asset Investments

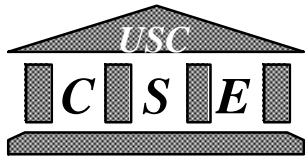
Cost Factors	Major Cost Drivers
<ul style="list-style-type: none">• Domain engineering• Architecture determination• Reusable components development• Legacy SW reengineering• Process redefinition• Training, teambuilding• Repository development, operations• Component certification• Component maintenance• Marketed product costs	<ul style="list-style-type: none">• Breadth, Maturity of domain• Architecture maturity; diversity of legacy, COTS software• Breadth of reuse; other cost drivers• Legacy SW structure, understandability• Process maturity• Novelty of PLM approach; no. of trainees• Number of components; user• Number of components; degree of certification• Stability of application, technology, environment, infrastructure• Type of product (package, service)

1/6/97



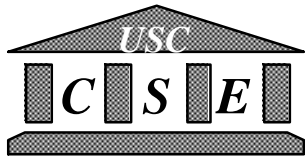
Sources of Asset Reuse Costs

Cost factors	Major Cost Drivers
<ul style="list-style-type: none">• Locating and assessing assets• Modifying assets• Integrating and testing assets and product• Product/asset maintenance• Asset acquisition and maintenance fees	<ul style="list-style-type: none">• Product line maturity; fit of project to product line assets• Asset structure, application clarity, self-descriptiveness• Product line maturity and fit to project; number of asset interfaces• Stability, breadth of product line; stability of product; asset controllability• Fee structures



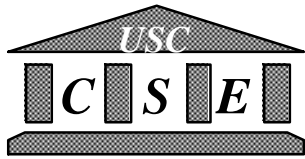
Sources of Asset Reuse Benefits

Benefit Factors	Major Drivers
<ul style="list-style-type: none">• Cost avoidance• Cycle time reduction benefits• Quality benefits<ul style="list-style-type: none">– error reduction– user interface• Risk reduction benefits• Marketed product income	<ul style="list-style-type: none">• No. of products in product line; degree of component reuse• Completeness of PLM solution; training and experience<ul style="list-style-type: none">– Certification, process, training– Completeness of PLM solution user interface area• Completeness of PLM solution; process used• Market size and share; price structure



Some Useful Proxies: RCWR and RCR

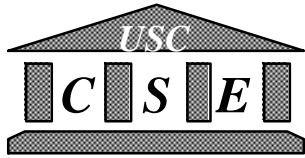
- RCWR: Relative Cost of Writing for Reuse
- RCR: Relative Cost of Reuse
- Typical RCWR, RCR Values [Poulin, 1997]
- Effects of Cost Drivers
 - European Consultants Network
 - COCOMO 2.0



RCWR: Relative Cost of Writing for Reuse

$$\text{RCWR} = \frac{\text{Cost of developing reusable asset}}{\text{Cost of developing single-use asset}}$$

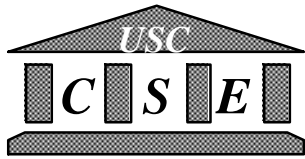
- Typical RCWR values [Poulin, 1997]
 - Range of 12 studies: 1.2 - 2.0
 - Median of 12 studies: 1.5



RCR: Relative Cost of Reuse

$$\text{RCR} = \frac{\text{Cost to reuse asset}}{\text{Cost to develop asset from scratch}}$$

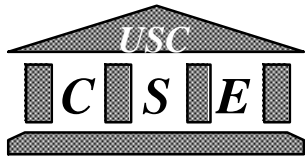
- Typical RCR values [Poulin, 1997]
 - No modifications (9 studies): 0.03 - 0.40
Median: 0.20
 - Small (<25%) mods (2 studies): 0.40 - 0.55



Effect of Complexity on RCWR, RCR

– European Consultants' Network [Favaro, 1991]

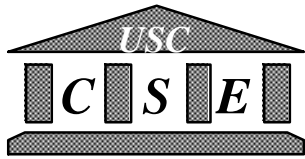
- Monolithic: simple structures (stacks, queues)
- Polyolithic: recursive structures (trees, lists)
- Graph: several types of traversals
- Menu: application-oriented data input
- Mask: application-oriented data output



RCWR, RCR Values vs. Complexity

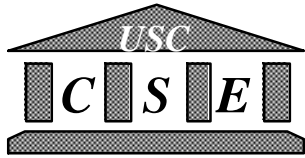
Complexity	RWCR	RCR
Monolithic	1.0	0.10
Polythetic	1.2	0.15
Graph	1.6	0.25
Menu	1.9	0.30
Mask	2.2	0.40

- Menus, masks cheaper now with GUI builders



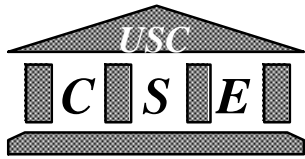
Relative Cost of Reuse: COCOMO 2.0

- Add Assessment & Assimilation increment (AA)
 - Similar to conversion planning increment
- Add software understanding increment (SU)
 - To cover nonlinear software understanding effects
 - Apply only if reused software is modified
- Results in revised Adaptation Adjustment Factor (AAF)
 - Equivalent DSI = (Adapted DSI)*(AAF/100)
 - $AAF = AA + SU + 0.4(DM) + 0.3(CM) + 0.3(IM)$



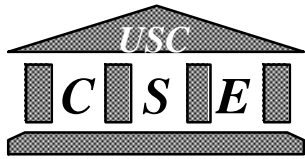
Conclusions

- Software reuse payoffs are significant
 - Vastly reduces hardware-software productivity gap
- Magnitude of reuse payoffs is situation-dependent
 - Most powerful heuristic: Keep It Simple
- Cost models and business case analysis enable assessment of situation-dependencies
 - Lead to better reuse decisions



References

- Jeffery Poulin, Measuring Software Reuse, Addison Wesley, 1997.
- John Favaro, “What Price Reusability: A Case Study,” Ada Letters, Spring 1991, pp. 115-124.
- Barry Boehm et al., “Cost Models for Future Software Processes: COCOMO 2.0,” Annals of Software Engineering, 1995 (also at <http://sunset.usc.edu>).
- David Garlan et al., “Architectural Mismatch: Why Reuse is So Hard,” IEEE Software, November 1995, pp. 17-26.
- DARPA-EDCS Architecture Program (PM: Dr. John Salasin, ITO).



Outline

- Motivation
- Software Reuse Economics Framework
 - Return on Investment Elements
 - Investments and Pay off Elements
 - Some Useful Proxies: RCWR and RCR
- ➔ • Bottom Line: Lines of Code In Service
- Conclusions and References