Reuse Economics: A Novel Cost Metric for Software Reuse

Tullio Vardanega

May 15, 2002

Contents

Definitions

- **systematic reuse** entails planning, executing and monitoring changes in the processes at organisational level
- **opportunistic reuse** is narrow (it arises from the goodwill of individuals) and shallow (does not impact the processes)

Definitions (cont'd)

- only systematic reuse can produce **measurable** benefits
- asset reusability is the delicate balance of (at least) three crucial attributes:
  1. **functional reusability**: the reuse asset must offer the functionality required by the project
  2. **technical reusability**: the reuse asset should integrate easily in the operating environment of the project
  3. **quality**: the quality of the reuse asset must meet the requirements of the project

Morisio, M., Tully, C., and Ezran, M. (2000). Diversity in Reuse Processes. *IEEE Software*, 17(4):56–63.

Motivations

- focus on **third-party unmodified** reuse

  → **consumer** perspective

- **less coarse** parameter set than classical models

  → tighter control of bounds

- **less complex** than probabilistic models

  → lower entry barrier, inherently adaptive

- full recognition of **overhead (indirect) costs** of reuse
- in keeping with most frequently used data items and outputs
- centered on **development** phase

  → needs enhancement to cover **O&M** phase

Foundations

Definition

- $-\infty \leq \varepsilon \leq 1$

  cost impact of reuse (i.e., *savings* or *losses*) **relative** to a functionally-equivalent development of **all-new** code

Assertions

- two distinct factors *jointly* determine ε:
  - $-\infty \leq \pi \leq 1$: the extent of **development reduction** attained by reuse at **product** level
  - $0 \leq \gamma \leq 1$: the effect of **reduction preservation** warranted by the **process**
- reuse can cause **size inflation** by way of verbosity and / or importation of additional needs
- a custom baseline may **pre-exist** the reuse under analysis

Model representation

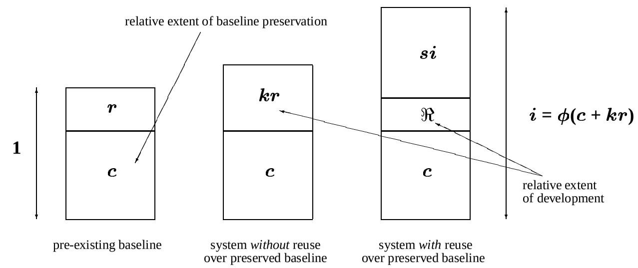

Figure 1: Model notions.

→ **whatever the nature of the reference product** ←

Model definitions

- normalised cost basis: $c + r = 1$
- own baseline may pre-exist reuse: $0 \leq c \leq 1$
- system *without* reuse: $c + kr$
- baseline replacement may need enhancing to meet system requirements: $k \geq 1$
- system *with* reuse: $\phi(c + kr)$
- reuse may cause size inflation: $\phi \geq 1$
- reuse ratio relative to size with reuse: $0 \leq s \leq 1$

Economic model

$$-\infty \geq \varepsilon = \gamma\pi \leq 1$$

- The saving potential of reuse at *product* level, as preserved (or eroded) by the *process*

Economic model (cont'd)

π : development reduction factor (cf. figure 1)

definition : the relative proportion by which reuse decreases (resp. increases) the development effort

$$\pi = 1 - \frac{\Re}{kr} = 1 - \frac{\phi(1-s)(1+(k-1)r) - 1 + r}{kr} \quad (1)$$

γ : reduction preservation factor (from the process)

definition : the extent to which the software development process is able to preserve the proportion of development reduction attainable by reuse

$$\gamma = \sum_i e_i \gamma_i \quad \text{where: } \sum_i e_i = 1 \text{ and } 0 \leq e_i, \gamma_i \leq 1\ \forall i \quad (2)$$

distinct share of development effort per phase (e_i)

distinct extent of reduction preservation per phase (γ_i)

Economic model (cont'd)

productivity increase ratio (inferred)

definition : trivially, how much more (i.e., ϕ, cf. figure 1) in how much less (i.e., $1-\varepsilon$)

$$\vartheta = \frac{\phi}{1-\varepsilon} \quad (3)$$

Observation

The model allows for $\phi > 1$ (i.e., *increase of development size on reuse*) and $r \to 0$ (i.e., *pre-existing baseline covers almost all requirements*) so that π can go **deeply negative** cf. eqn. 1

$$-\infty \leq \pi \leq 1$$

Challenge and Pedigree

Yet another model!

Why would we need it at all?

In what do we differ?

Solid relationship to major models

References

[1] Gaffney, J. and Durek, T. (1989). Software Reuse - Key to Enhanced Productivity: Some Quantitative Models. *Information and Software Technology*, 31(5):258–267.

[2] Lim, W. (1996). Reuse Economics: A Comparison of Seventeen Models and Directions for Future Research. In *Proceedings of the Int'l Conference on Software Reuse*, pages 41–50, Orlando, FL (USA). IEEE.

[3] Poulin, J. (1997). *Measuring Software Reuse: Principles, Practices and Economic Models*. Addison Wesley. ISBN 0201634139.

Most common type of model output

- savings from reuse $[\varepsilon]$

Most common type of data items

Producer perspective

- cost-to-producer to create asset for reuse $\rightarrow C_{p,r}$
- (projected) number of reuses $\rightarrow n$
- cost to create asset without reuse $\rightarrow C_{p,w}$

Consumer perspective

- cost to create system without reuse $\rightarrow C_{c,wp}$
- cost-to-consumer to reuse asset $\rightarrow C_{c,ra}$
- cost-to-consumer to create non-reusable asset $\rightarrow C_{c,wa}$

Producer-consumer break-even condition

$$n_0 = \left(\frac{C_{p,r}}{C_{p,w}}\right)\left(\frac{C_{c,wa}}{C_{c,wa} - C_{c,ra}}\right) \quad (4)$$

- The consumer is *the same* as the producer
- The term $\frac{C_{p,r}}{C_{p,w}}$ denotes the cost increase, at the producer end, arising from setting reuse objectives
  - This value generally is ≥ 1, but should be minimised (ideally to 1)
- The term $\frac{C_{c,wa}}{C_{c,wa}-C_{c,ra}}$ denotes the differential cost increase, at the consumer end, upon a *new* development in the face of a reuse opportunity
  - This value inherently is ≤ 1, and should be maximised (ideally to 1)

The Gaffney-and-Durek economic analysis [1] shows the **impact of the n_{th} reuse relative to** that obtained on a project developed from **all-new code**, where the cost basis equals 1:

$$C = 1 + R(b + \frac{E}{n} - 1) = 1 - R(1 - b - \frac{E}{n}) \quad (5)$$

where:

$R = \frac{\text{reused source statements}}{\text{total delivered source statements}}$ $[s]$

$0 \leq b \leq 1$ **relative cost of integrating reuse** $[\frac{C_{c,ra}}{C_{c,wa}}]$

$E \geq 0$ **relative cost of developing for reuse** $[\frac{C_{p,r}}{C_{p,w}}]$

which, for $r = 1$ (all new) and $E = 0$ (third-party reuse asset), our model expresses as:

$$C = 1 - \varepsilon = 1 - \gamma(1 - \phi(1 - s)) \quad (6)$$

Behaviour at boundary conditions

- **no reuse**
  $C(R = 0) = 1$
  $C(s = 0) = 1 - \gamma(\phi_{|s=0} - 1) = 1 - \gamma(1 - 1) = 1$
- **only reuse**
  $C(R = 1) = b$
  $C(s = 1) = 1 - \gamma$ $\rightarrow b = 1 - \gamma$
- **no cost of reuse integration** (reuse asset and reuse process work extremely well)
  $C(b = 0) = 1 - R = 1 - s$
  $C(\gamma = 1) = \phi(1 - s)$ **→ more conservative!**
- **max cost of reuse integration** (reuse asset and reuse process work extremely bad)
  $C(b = 1) = 1$
  $C(\gamma = 0) = 1$

Poulin [3] uses the previous model to determine the **total project cost with reuse**:

$$TPC_r = c_u \sigma (1 - s(1 - b)) \quad (7)$$

where:

c_u = unitary cost of development (per source statement)

σ = total size of development *with* reuse → size *without* reuse = $\frac{\sigma}{\phi}$

which our model expresses as:

$$TPC_r = c_u \frac{\sigma}{\phi}(1 - \varepsilon) = c_u \sigma \frac{1 - \varepsilon}{\phi} \quad (8)$$

from which, still for $r = 1$, we obtain a refined definition of b:

$$b = (1 - \gamma)(1 - \frac{\phi - 1}{s\phi}) \quad (9)$$

which is definitely more sophisticated than the equivalence we inferred from the Gaffney and Durek's model (which assumes $\phi = 1$)

Model equivalence criteria

- **no size inflation on reuse**

  $\phi = 1 \rightarrow b = 1 - \gamma$ **canonical form**

- **only reuse**

  $s = 1 \rightarrow b = (1 - \gamma)\frac{1}{\phi}$ **cost decreases as size increases**

- **reuse exceeds inflation**

  $\phi > 1$ *and* $s > \frac{\phi - 1}{\phi}$ → $0 < b < 1 - \gamma$ **general case**

  (our model is more sensitive to the attrition of reuse on the process!)

- **reuse equals inflation**

  $\phi > 1$ *and* $s = \frac{\phi - 1}{\phi}$ → $b = 0$

- **inflation exceeds reuse**

  $\phi > 1$ *and* $s < \frac{\phi - 1}{\phi}$ → $b < 0$ **productivity increase!**

Reference values

Other models generally regard b as a constant in the range $[0.2, 0.4]$

→ we treat it as a **derived** parameter

with values over a potentially wider range

The problem with equations 5 and 7 is that b is too coarse and hard to determine

The hourly cost of labour (c_h) and the typical productivity (p) determine c_u:

$\rightarrow c_u = \frac{c_h}{p} = \frac{75 \text{ Eur/hr}}{1.5 \text{ SLOC/hr}} = 50 \text{ Eur/SLOC}$

Key parameters

Table 1: Influence of s and ϕ on π.

case s	case ϕ: $\phi = 1$	case ϕ: $\phi > 1$	rationale
$s < \frac{\phi-1}{\phi}$	N/A	$\pi < 0$	inflation exceeds reuse
$s = \frac{\phi-1}{\phi}$	no reuse	$\pi = 0$	reuse equals inflation
$s > \frac{\phi-1}{\phi}$	$\pi > 0$	$\pi > 0$	reuse exceeds inflation

Example 1 ($k = 1$)

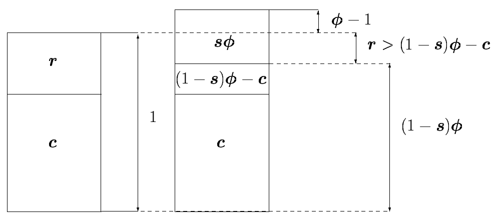

Figure 2: Reuse exceeds inflation ($s > \frac{\phi-1}{\phi}$).

Consistency criteria

- π has upperbound at 1 (for $s = 1$ and $r = 1$) and lowerbound at $-\infty$ (for $r = 0$ and $\phi > 1$)
- Imposing a **notional** lowerbound for π at -1 for $s \leq \frac{\phi-1}{\phi}$ determines a lowerbound for r on the *negative* plane of π
- Imposing that the reuse component (si) **should not overlap** the pre-existing baseline (c) for $s \geq \frac{\phi-1}{\phi}$ determines the corresponding lowerbound for r on the *positive* plane of π

Table 2: Lowerbounds for r.

case s	r range
$0 \leq s \leq \frac{\phi-1}{\phi}$	$r \geq \frac{\phi(1-s)-1}{(k-1)(1-\phi(1-s))+k} \geq 0$
$\frac{\phi-1}{\phi} \leq s \leq 1 - \frac{1-r}{\phi(1+(k-1)r)}$	$r \geq \frac{1-\phi(1-s)}{1+\phi(1-s)(k-1)} \geq 0$

Example 2 ($\pi < 0$)

$k = 1, \phi = \frac{9}{8}, s = \frac{1}{10} < \frac{\phi-1}{\phi} = \frac{1}{9}, r_{min} = \phi(1-s) - 1 = \frac{1}{80}$

Development with reuse for $r = r_{min}$ (from fig. 2):

$\Re = (1-s)\phi - c = (1-s)\phi - (1 - r_{min}) = \frac{1}{40}$

Twice as much as without reuse! $\rightarrow$ Hence: $\pi = 1 - \frac{\Re}{r_{min}} = -1$

Example 3 ($\pi > 0$)

$k = 1, \phi = \frac{9}{8}, s = \frac{1}{5} > \frac{\phi-1}{\phi} = \frac{1}{9}, r_{min} = 1 - \phi(1-s) = \frac{1}{10}$

Development with reuse for $r = r_{min}$ (from fig. 2):

$\Re = (1-s)\phi - c = (1-s)\phi - (1 - r_{min}) = 0$

No development at all! $\rightarrow$ Hence: $\pi = 1 - \frac{\Re}{r_{min}} = 1$

The notion of r_{min} implies that r cannot possibly get any smaller as long as s and ϕ (and k) stay the same

Effect of r on π (1 of 2)

For $s < \frac{\phi-1}{\phi}$, where *inflation exceeds reuse*, we have:

$$\pi_{|r=r_{min}} = -1 \leq \pi_{|r=1} = 1 - \phi(1-s) < 0 \quad (10)$$

where π **increases** with r but it always lays within the **negative** plane ($\pi_{|r=1}$ denotes the case with no baseline preservation)

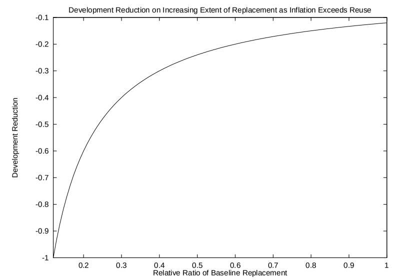

Figure 3: Case $s < \frac{\phi-1}{\phi}$ with $r_{min} = 0.12$ and $\pi_{|r=1} = -0.12$

Note that $\pi_{|r=1} = -r_{min}$ for $k = 1$ (negative plane)

Effect of r on π (2 of 2)

For $s > \frac{\phi - 1}{\phi}$ where *reuse exceeds inflation*, we have:

$$0 < \pi|_{r=1} = 1 - \phi(1 - s) \leq \pi|_{r=r_{\min}} = 1 \qquad (11)$$

where π **decreases** as r increases but it always lays within the **positive** plane

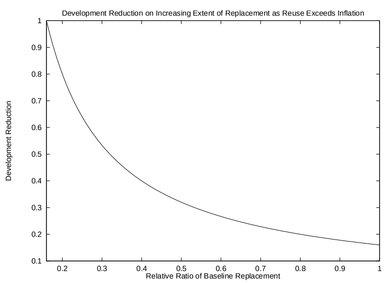

Figure 4: Case $s > \frac{\phi - 1}{\phi}$ with $r_{\min} = 0.16$ and $\pi|_{r=1} = 0.16$

Note that $\pi|_{r=1} = r_{\min}$ for $k = 1$ (positive plane)

Effect of k on π (1 of 2)

For $s < \frac{\phi - 1}{\phi}$, where *inflation exceeds reuse*, we have:

$$\pi|_{k=1} = \frac{1 - \phi(1 - s)}{r} \leq \pi|_{k \to +\infty} = 1 - \phi(1 - s) < 0 \qquad (12)$$

where π **increases** with k but it always lays within the **negative** plane. Note that the case $\pi|_{k \to +\infty}$ corresponds to $r = 1$, i.e., no baseline preservation

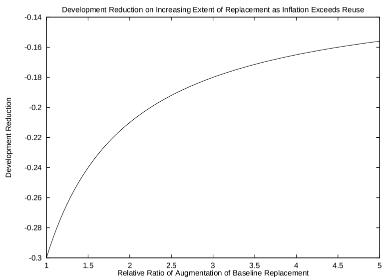

Figure 5: Case $s < \frac{\phi - 1}{\phi}$ with $\pi|_{k=1} = -0.3$ and $\pi|_{k \to +\infty} = -0.12$

Effect of k on π (2 of 2)

For $s > \frac{\phi - 1}{\phi}$, where *reuse exceeds inflation*, we have:

$$0 < \pi \leq \pi|_{k=1} = \frac{1 - \phi(1 - s)}{r} \qquad (13)$$

where π **decreases** as k increases but it always lays within the **positive** plane. Note that $\pi|_{k=1}$ is a genuine upperbound to π

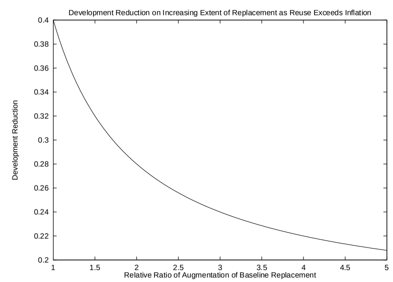

Figure 6: Case $s > \frac{\phi - 1}{\phi}$ with $\pi|_{k=1} = 0.4$ and $\pi|_{k \to +\infty} = 0.16$

Effect of ϕ on ϑ (1 of 2)

For $r < 1$, in the event of *baseline preservation*, we have:

$$\vartheta|_{\phi=1} = \frac{1}{1 - \frac{\gamma s}{r}} \quad \text{and} \quad \vartheta|_{\phi \to +\infty} = \frac{r}{\gamma(1 - s)} \qquad (14)$$

where ϑ **decreases** as ϕ increases, as long as $r < \gamma$.

For $0 \leq r = \gamma \leq 1$, we have $\vartheta = \frac{1}{1-s}$ for any value of ϕ, while for $1 \geq r > \gamma \geq 0$, ϑ **increases** with ϕ. Note that ϑ always lays **above 1** as long as $1 - s < \frac{r}{\gamma}$

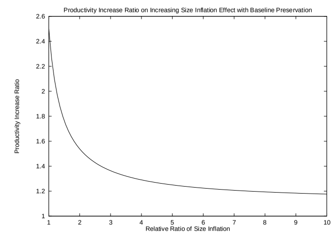

Figure 7: Case $r < \gamma = 0.6$ with $k = 1$ and $s = 0.4$, where $\vartheta|_{\phi=1} = 2.5$ and $\vartheta|_{\phi \to +\infty} = 1.1$

Effect of ϕ on ϑ (2 of 2)

For $r = 1$, in the event of *no* baseline preservation, we have:

$$\vartheta|_{\phi=1} = \frac{1}{1-\gamma s} \quad \text{and} \quad \vartheta|_{\phi\to+\infty} = \frac{1}{\gamma(1-s)} \tag{15}$$

where ϑ **increases** with ϕ and **never** falls below 1 (which is quite obvious, for this represents the productivity on an all-new development)

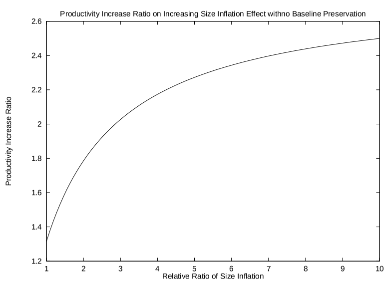

Figure 8: Case $r = 1$ with $s = 0.4$ and $\gamma = 0.6$, where $\vartheta|_{\phi=1} = 1.3$ and $\vartheta|_{\phi\to+\infty} = 2.8$

Sensitivity analysis

Table 3: Sensitivity of cost model to parameter variations.

variation	increment effect
$\Delta\pi\|_{\Delta\phi} = -\Delta\phi\frac{(1-s)(1+(k-1)r)}{kr}$	**decreases value**
$\Delta\pi\|_{\Delta s} = \Delta s\frac{\phi(1+(k-1)r)}{kr}$	**raises value**
$\Delta\pi\|_{\Delta r} = \Delta r\frac{\phi(1-s)-1}{kr_1r_2}$	**decreases loss** for $s<\frac{\phi-1}{\phi}$ **decreases value** for $s>\frac{\phi-1}{\phi}$
$\Delta\pi\|_{\Delta k} = \Delta k\frac{(\phi(1-s)-1)(1-r)}{k_1k_2r}$	**decreases loss** for $s<\frac{\phi-1}{\phi}$ **decreases value** for $s>\frac{\phi-1}{\phi}$
$\Delta\varepsilon\|_{\Delta\gamma} = \Delta\gamma\pi$	**raises value**

Scenario 1 ($r < 1$)

Table 4 lists the results of the sensitivity analysis computed allowing **single errors** in the determination of parameter values, in the event of **baseline preservation**. (The calculation could easily be extended to allow the occurrence of multiple errors.)

Table 4: Sensitivity of ε to relative errors in parameter values.

parameter	relative error	Δ value	$\Delta\varepsilon$
$r = 0.20$	$\pm5\%$	±0.010	∓0.007
$k = 2.50$	$\pm5\%$	±0.125	∓0.006
$s = 0.40$	$\pm5\%$	±0.020	±0.033
$\phi = 1.40$	$\pm5\%$	±0.070	∓0.050
$\gamma = 0.46$	$\pm5\%$	±0.023	±0.001

ϕ appears to be the parameter with the **most impact** on ε.

A relative error of $+5\%$ in ϕ in the scenario results in an absolute variation of about -5% in ε.

Scenario 2 ($r = 1$)

Table 5 lists the results of the sensitivity analysis computed under the same hypotheses as used for scenario 1 in the event of **no** baseline preservation.

Table 5: Sensitivity of ε to relative errors in parameter values.

parameter	relative error	Δ value	$\Delta\varepsilon$
$s = 0.40$	$\pm5\%$	±0.020	±0.016
$\phi = 1.40$	$\pm5\%$	±0.070	∓0.024
$\gamma = 0.58$	$\pm5\%$	±0.029	±0.005

ϕ confirms as the model parameter with the **most impact** on ε.

A relative error of $+5\%$ in ϕ in scenario 2 results in an absolute variation of about -2.4% in ε.

Table 5 also suggests that the model tolerates well a fair amount of approximation in the evaluation of γ.

Conclusions

With respect to other models:

- we have a method for determining b
  $\rightarrow$ *not an independent constant!*
- we have a method for determining s
  $\rightarrow$ *reuse ratio grows with system size*
- we have a method for determining ϕ
  $\rightarrow$ *shows inherent overhead of reuse product*

How can we best increase ε?

- by decreasing ϕ *further use decreaes ϕ''!*
- by increasing γ *via process-supportive reuse assets*
- by increasing s *enlargement or incorporation of fit baseline*

In essence:

by reducing b at **product** and **process** level!

ROI

3-component project-level ROI (**simplified**):

$$(C_{c,wp} - C_{c,rp}) + (C_{c,mw} - C_{c,mr}) - (C_{p,r} - C_{p,w})$$

consumer $\neq$ producer $\rightarrow (C_{p,r} - C_{p,w}) = 0$

O&M cost impact of reuse remains to be determined $\rightarrow \delta$

$$\text{ROI} = C_{c,wp}\,\varepsilon + \delta = c_u \frac{\sigma}{\phi}\varepsilon + \delta$$

Example

$\varepsilon = 0.3, \phi = 1.125, \sigma = 30$ kSLOC, $c_u = 50$ Eur/SLOC

$$\rightarrow \text{ROI} = c_u \frac{\sigma}{\phi}\varepsilon + \delta = 400 \text{ kEur} + \delta$$

Table 6: Model notations.

symbol	denotes	ranges	
r	baseline replacement ratio	$0 \leq r \leq 1$	
c	baseline preservation ratio	$c + r = 1$	
k	functional growth factor	$k \geq 1$	
s	unmodified reuse ratio	$0 \leq s \leq 1$	
ϕ	cumulative expansion factor	$\phi = \frac{1+\phi'}{1-\phi''} \geq 1$	$(0 \leq \phi', \phi'')$
i	normalised system size	$i = \phi(c + kr)$	
π	development reduction factor	$\pi \leq 1$	
γ	reduction preservation factor	$0 \leq \gamma \leq 1$	
ε	effort saving	$\varepsilon = \gamma\pi$	
ϑ	productivity increase ratio	$\vartheta = \frac{1+\phi'}{1-\varepsilon} \geq 0$	
b	relative cost of reuse	$b = 1 - \frac{\varepsilon+\phi'}{s(1+\phi')}$	
σ	system size (source statements)		

where ϕ': base size inflation (generality/specificity) and ϕ'': reuse overhead (COCOMO's AA and SU)