July 8, 2002

Data from Salion's Software Product Line Initiative

Report # 2002-07-08-1

Leverage – ***magnifying the force applied to solving a problem.***
Utilizing BigLever Software GEARS and a reactive model, Salion adopted a software product line approach in a fraction of the time and effort compared to what has been reported previously for software product line initiatives. Salion's ROI occurred at 1.01 products, in contrast with typical results of 2.5 to 3.5 products. This number is 1% over the theoretical minimum of 1.
Early data indicates that Salion will be able to create and maintain the product variants in its software product line with 5% of the conventional time and effort.

Contact Information:

Charles W. Krueger, Ph.D., CEO
ckrueger@biglever.com
www.biglever.com
(512) 426-2227

1.0 Background

Salion adopted a software product line approach using BigLever Software GEARS. A reactive model is used, where each new Salion customer provides unique customization requirements that cannot be predicted in advance.

A fully functional default product was initially implemented and serves as the common product baseline. This baseline product was implemented as a standalone product that is used by customers without customization requirements. The baseline product was implemented without special support for software product line customizations. The GEARS software mass customization infrastructure was added after this product was complete in order to support product line variants.

2.0 Adoption Effort, Steady-State Effort, and ROI

The effectiveness of the Salion software product line initiative is measured in terms of the effort required to deploy the individual products in the product line and comparing this to typical data for proactive and conventional approaches. An overview of the theory, formulas, and graphs used in this section is provided in Appendix A.0 on page 3.

The Salion product line initiative had three phases, consistent with the reactive product line model:

1. develop the baseline product
2. add the GEARS software mass customization infrastructure
3. incrementally add product variants as new product requirements become available

Following is the data collected thus far, using the variables described in Appendix A.3:

- Effort to develop baseline product: $E_{\Delta 1}$ = 190 person*months
- Effort to add GEARS product line infrastructure: E_{IR} = 2 person*months
- Effort to engineer second product: $E_{\Delta 2}$ = 15 person*months
- Effort to engineer third product: $E_{\Delta 3}$ = 12 person*months (projected estimate)
- Effort to engineer fourth product: $E_{\Delta 4}$ = 10 person*months (projected estimate)

Using equation EQ 10, Salion achieved ROI in going from their baseline product to a product line at 1% of the “distance” between product 1 (the baseline product) and product 2 (the first custom product), or 1.01 products. Note that absolute best case is 1.0 products where no effort is required to adopt a product line approach. A typical ROI number reported from industry case studies is 3.0.

$$N_{Rr} = \frac{E_{IR} + E_{\Delta 1} - E_{\Delta 2}}{E_{\Delta 1} - E_{\Delta 2}} = \frac{2 + 190 - 15}{190 - 15} = 1.01$$

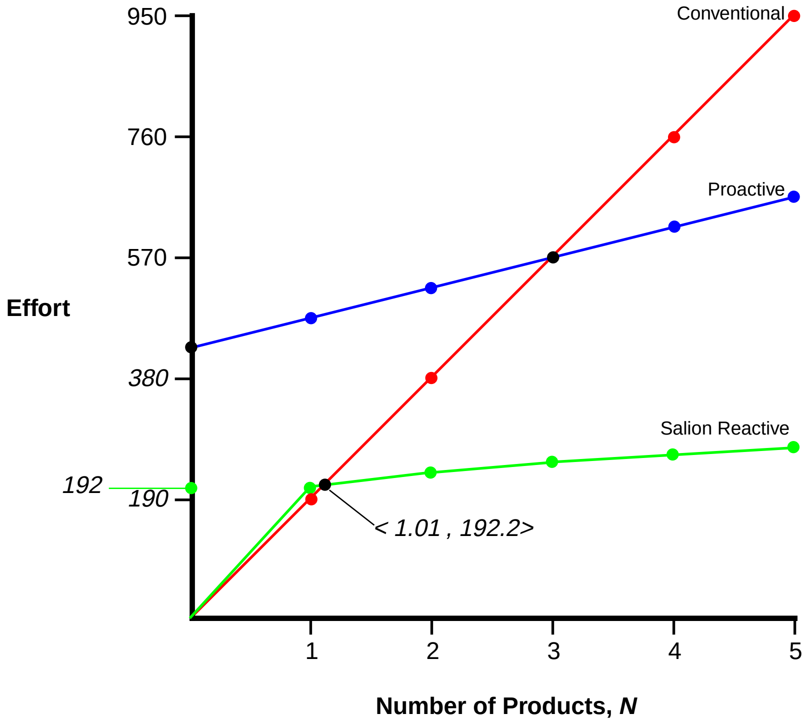
Number of Products for Salion to Achieve ROI **(EQ 1)**

Using equation EQ 11, the effort for Salion to reach ROI was 192.2 person*months, or 2.2 over the effort to build the first product. Note that the absolute best case is 190 and that typical industry numbers for the effort of 3 convention product variants would be 570 person months.

$$E_{Rr} = E_{IR} + E_{\Delta 1} + (1.01 - 1) \times E_{\Delta 2} = 2 + 190 + 0.01 \times 15 = 192.2$$

Effort for Salion to Achieve ROI **(EQ 2)**

Following is the graph of this data, shown relative to conventional and proactive graphs.



Appendix A.0 Background

This section outlines the theory behind the software product line effort, comparative analysis, and ROI descriptions.

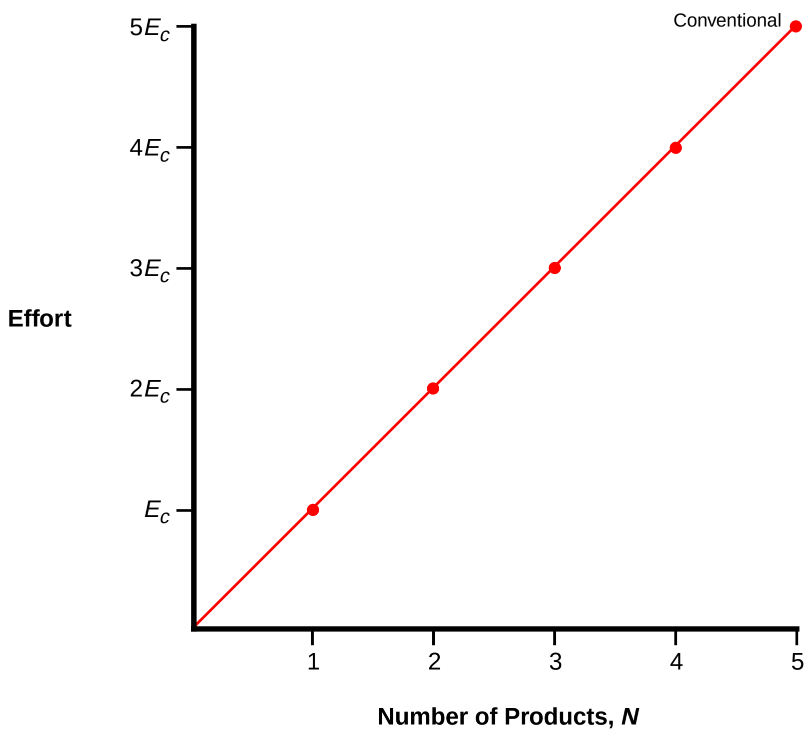
A.1 Conventional Effort Formula

The conventional approach to building software products in a product line typically results in products evolving on independent maintenance paths. There may be some initial ad hoc reuse as new products are created (such as clone-and-own), but subsequent development and maintenance is independent.

Since maintenance typically consumes 80% of the effort in a software product, the initial ad hoc reuse is mostly insignificant in reducing the effort required to build a software product line. Therefore, the effort of building a software product line using the conventional ad hoc approach is directly proportional to the number of products built, *N*, times the average effort of building a conventional product, E_c:

$$\sum_{1}^{N} E_i = N \times E_c$$

Cost of Conventional Product Line **(EQ 3)**

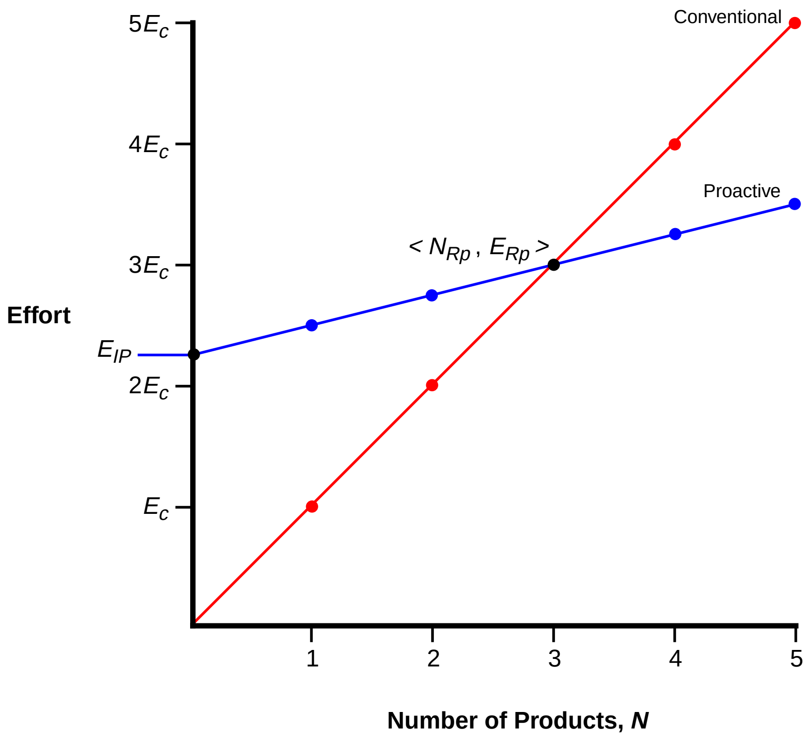


A.2 Proactive Effort Formula

The proactive approach to building software products in a product line requires an up-front effort to create the initial core assets plus the infrastructure for maintaining and instantiating products. After the initial investment, the cost of building individual products is simply the delta cost of building what hasn't been built before. Thus the effort of build a software product line using the proactive approach is the initial proactive effort, E_{IP}, plus the number of products built, N, times the average delta effort of building each new product instance, E_{Δ}.

$$\sum_{1}^{N} E_i = E_{IP} + N \times E_{\Delta} \qquad \text{(EQ 4)}$$

Cost of Proactive Product Line



The ROI crossover of the proactive approach relative to the conventional approach (i.e., the number of products where the total proactive effort becomes equal to the conventional effort) can be derived from the equality of equation EQ 3 and equation EQ 4. This is shown in the graph under *equation EQ 4* as, where N_{Rp} is the number of products for proactive ROI and E_{Rp} is the effort for proactive ROI.

$$N_{Rp} = \frac{E_{IP}}{E_c - E_\Delta}$$

Number Products for Proactive Product Line ROI **(EQ 5)**

Substituting the ROI expression from equation EQ 5 into the effort expression in equation EQ 4 gives the effort for proactive ROI.

$$E_{Rp} = E_{IP} + \left(\frac{E_{IP}}{E_c - E_\Delta}\right) \times E_\Delta$$

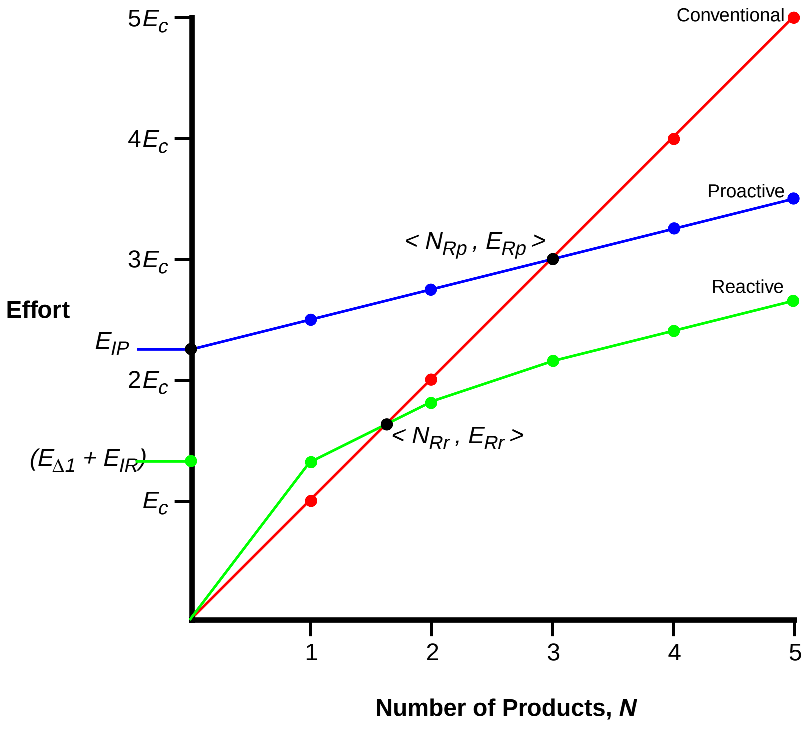
Effort for Proactive Product Line ROI **(EQ 6)**

A.3 Reactive Effort Formula

The reactive approach to building software products in a product line requires an upfront effort to create the initial product and the infrastructure for maintaining and instantiating products. After the initial product, the cost of building individual products is simply the delta cost of building what hasn't been built before. With the reactive approach, latter deltas are likely to be smaller than earlier deltas since "discovery" of commonality and variation is still occurring in the early deltas. Thus the effort of building a software product line using the reactive approach is the initial product effort, E_1, plus the initial infrastructure effort, E_{IR}, plus the sum of the deltas for all subsequent products, $E_{\Delta i}$. For uniformity, E_1 can be treated as a delta starting from scratch and expressed as $E_{\Delta 1}$.

$$\sum_{1}^{N} E_i = E_{IR} + \sum_{1}^{N} E_{\Delta i}$$

Cost of Reactive Product Line **(EQ 7)**



The ROI crossover of the reactive approach relative to the conventional approach (i.e., the number of products where the total reactive effort becomes equal to the conventional effort) can be derived from the equality of equation EQ 3 and equation EQ 7. This is shown in the graph under *equation EQ 7* as, where N_{Rr} is the number of products for reactive ROI and E_{Rr} is the effort for reactive ROI.

The follow equation expresses the number of products, N_{Rr}, for ROI between products *(N-1)* and *N*.

$$N_{Rr} = \frac{E_{IR} + \sum_{1}^{N-1} E_{\Delta i} - (N-1) \times E_{\Delta N}}{E_{\Delta 1} - E_{\Delta N}}$$

Number Products for Reactive Product Line ROI **(EQ 8)**

The follow equation expresses the effort, E_{Rr}, for ROI between products *(N-1)* and *N*.

$$E_{Rr} = E_{IR} + \sum_{1}^{N-1} E_{\Delta i} + \left(\left(\frac{E_{IR} + \sum_{1}^{N-1} E_{\Delta i} - (N-1) \times E_{\Delta N}}{E_{\Delta 1} - E_{\Delta N}} \right) - (N-1) \right) \times E_{\Delta N}$$

Effort for Reactive Product Line ROI **(EQ 9)**

The typical case is for ROI between the first and second product, in which case equation EQ 8 and equation EQ 9 can be simplified as follows:

$$N_{Rr} = \frac{E_{IR} + E_{\Delta 1} - E_{\Delta 2}}{E_{\Delta 1} - E_{\Delta 2}}$$

Reactive Product Line ROI Between Product 1 and 2 **(EQ 10)**

The follow equation expresses the effort, E_{Rr}, for ROI between products *(N-1)* and *N*.

$$E_{Rr} = E_{IR} + E_{\Delta 1} + \left(\left(\frac{E_{IR} + E_{\Delta 1} - E_{\Delta 2}}{E_{\Delta 1} - E_{\Delta 2}} \right) - 1 \right) \times E_{\Delta 2}$$

Effort for Reactive Product Line ROI Between Product 1 and 2 **(EQ 11)**