

The NPVI Method to Support Market Entry Strategies for Software Products

Hans Sassenburg
Software Engineering Institute Europe
Carnegie Mellon University
An der Welle 4
D-60322 Frankfurt
hanss@sei.cmu.edu

Egon Berghout
CITER – Centre of IT Economics Research
University of Groningen
P.O. Box 800
NL-9700 AV Groningen
e.w.berghout@rug.nl

Abstract: Market entry strategies deal with the complex trade-off between economising on time-to-market and economising on product maturity. How much testing is required? What are the expected cash inflows and cash outflows if the product is released? What is the market window? Based on product life-cycle models frequently used in the semiconductor industry, a method was defined using one of the most commonly used capital budgeting methods: NPV. This method takes into account the discounted value of money, and can be used to calculate the difference between two alternatives in one single variable. This variable, the NPV Incentive (NPVI), is calculated from various underlying metrics and measures the economic incentive to favour one alternative over another. Practical use of the method requires the ability to determine the values of the underlying metrics with significant certainty. It should be the objective for decision-makers to search for a zone of cost effectiveness: a bandwidth in which the marginal net asset value of additional information is equal or close to zero.

Keywords: Market Entry Strategy / Optimal Release Time / Uncertainty / Zone of Cost Effectiveness / Decision-making / Information Economics

1 Introduction

Software has emerged as a major worldwide industry, and demand is still increasing exponentially [BS00] [LS00] [Hu02]. Software pervades in a multitude of products in social, business and military human-machine systems. It includes information (technology) systems developed for the gathering, processing, storing, retrieval and manipulation of information to fulfil organizational needs, as well as commercially developed software products to be sold to one or more customers or end-users. The increasing demand for software also results in an increase of our dependence upon it [LS00]. Software-based systems replace older technologies in safety- or mission-critical applications.

- Software moves from an auxiliary to a primary role in providing critical services.
- Software becomes the only way to perform some function which is not perceived as critical but whose failure would deeply affect individuals or groups.
- Software-provided services become increasingly an accepted part of everyday life without any special scrutiny.
- Software-based systems are increasingly integrated and interacting, often without effective human control.

Despite the exponential increase in the demand for software and the increase in our dependence upon it, many software manufacturers behave in an unpredictable manner [BS00] [SG04]. In an unpredictable software manufacturer organization, it is difficult to determine when a software product will be released, the features the product will have, the associated development costs, and the resulting product quality. Without knowledge of when a software product will be released, a software manufacturer will experience difficulty in planning activities such as product promotions, customer training, and maintenance support. Further, resource utilization across projects may become inefficient and difficult to manage when projects fail to meet schedules. Finally, when a scheduled release date is missed, customers will have difficulties in planning for the introduction of the new software product into their organizations. For many software manufacturers, especially those operating in mass markets, the market entry is a point of no return. A software release decision can be seen as a trade-off between early release to capture the benefits of an earlier market introduction (such as a larger installed base), and the deferral of product release to enhance functionality or improve quality. At first sight, taking a strictly economic perspective, this trade-off does not seem to be of a special nature. If a software product is released 'too early', it will have less functionality and/or significant defects, and the software manufacturer will incur the post-release cost of fixing the resulting failures. If a software product is released 'too late', the additional development cost and the opportunity cost of missing a market window could be substantial.

Despite the multitude of stories about post-release problems of prematurely released software, there is little in the literature on how the optimal release time is determined in practice. A study prepared for the National Institute of Standards & Technology, revealed (a combination of) the following non-analytical methods to decide when a software product is ‘good enough’ to release [Rt02]:

- A ‘sufficient’ percentage of test cases run successfully.
- Statistics are gathered about what code has been exercised during the execution of a test suite.
- Defects are classified and numbers and trends are analysed.
- Real users conduct beta testing and report problems, which are then analysed.
- Developers analyse the number of reported problems in a certain period of time. When the number stabilizes or remains below a certain threshold, the software is considered ‘good enough’.

In section 2, the market entry trade-off is discussed from an economic perspective. A simple product life-cycle that is frequently used in the semiconductor industry is used as the basis. In section 3, this model is extended with cost functions. In section 4, a method is presented that will enable a software manufacturer to compare different market entry strategies. Considerations regarding the information perfection of values for the underlying metrics in this method are discussed in section 5. Conclusions are presented in section 6.

2 Market Entry Strategies: An Economic Model

The market entry timing trade-off question is not unique to the software industry in general. For example, Hatch and Macher have studied the trade-off between time-to-market and manufacturing performance, using the semiconductor industry as the empirical setting [HM99]. The trade-off question is identical, being to find the optimal time to commercialise a new technology: economising on time-to-market or economising on technology maturity. Delaying the transfer will increase the opportunity costs of lost revenues and falling prices; transferring the technology too early will create manufacturing problems that cause the product to fail. The software industry is confronted with the same entry timing trade-off. Spending too much on quality issues such as safety or security can result in a manufacturer not making any money on his product, while spending too little can lead to a disaster.

Assuming the concept of maximizing behaviour, or the theory of the firm, the primary objective of an organization will be to maximize its profit. Profit maximization starts from the assumption that marginal revenue (asset value generated from cash inflows) equals marginal costs (cash outflows) and thus, marginal profit equals zero. Total profit is equal to total revenue minus total costs. A break even point is reached when revenues equal costs and thus, profit is zero. In manufacturing environments, costs normally comprise pre-release development costs and post-release operational or maintenance costs.

To demonstrate the effects of a delayed market entry on profit, information should be available about the market life-cycle model, defining the market or demand window. As an example, a simple product life-cycle that is frequently used in the semiconductor industry, as described by Liu [Li95], is presented. The product life-cycle is approximated to a triangle with duration $2W$, partitioning the market into two stages:

- Market growth window with duration W . The product begins to gain market acceptance and sales tend to grow rapidly as the product reaches the mass market.
- Market decline window with duration W . As technology advances and superior products are launched, product sales will begin to decline. This downward trend in sales will continue as the market declines, thereby forcing managers to phase out the product.

It is assumed therefore, that the peak of the market is in the middle of the product life-cycle, and that in order to maximize revenues or asset value, the product should be on the market by the start of the market (demand) window.¹ This is illustrated in Figure 1. This model can be used to make a comparison between delivering a software product on time and delivering it with a delay.

In the case of an on-time market entry, the resulting asset value C will be equal to:

$$C = \frac{1}{2} \cdot 2W \cdot C_{max} \quad (1)$$

¹ These models are described in terms of ‘revenues’. For the sake of consistency with the remainder of this paper, the term ‘asset value’ is used instead.

In the case that a product launch is not meeting its initially planned release date T_r (start of market window), but has a delay D , and assuming that the peak of the market is in the middle of the product life-cycle, the asset value C' will now be equal to:

$$C' = \frac{1}{2} \cdot (W-D+W) \cdot ((W-D)/W) \cdot C_{max} \quad (2)$$

The loss in asset value C_{loss} is:

$$\begin{aligned} C_{loss} &= C - C' \\ &= C \cdot D (3W - D) / 2W^2 \end{aligned} \quad (3)$$

See Figure 2.

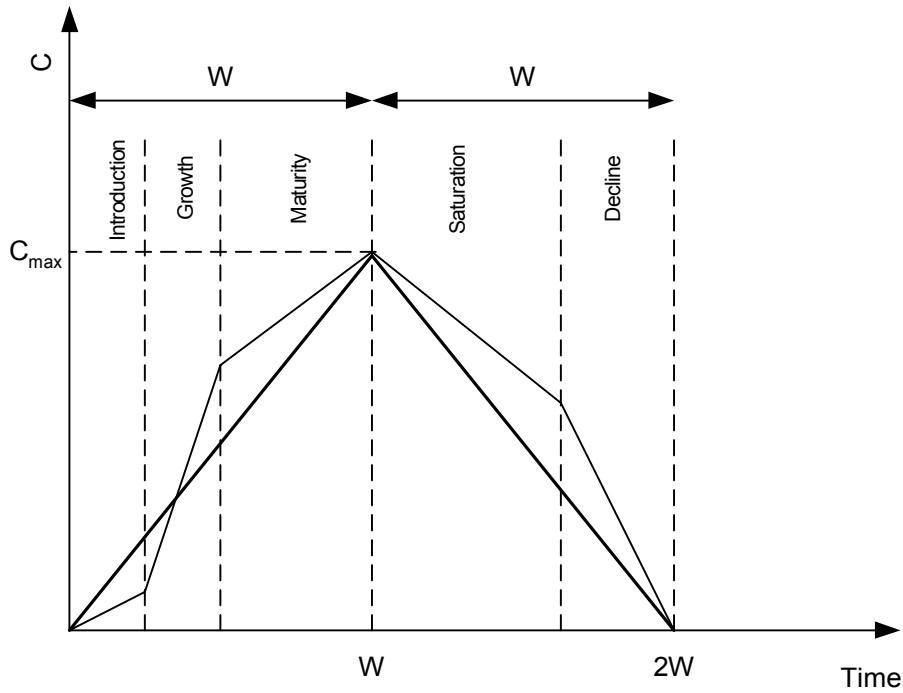


Figure 1: Approximation of product life-cycle to a triangle [Li95]

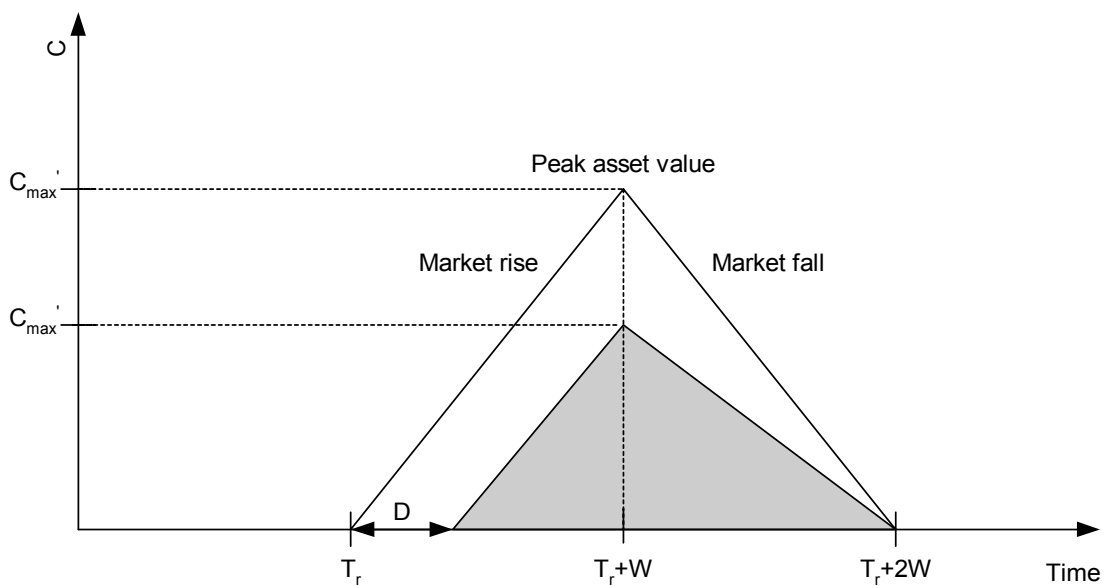


Figure 2: Asset value in the case of a delayed market entry [Li95]

3 Extended Model with Cost Functions

The model presented is a simplification of the real world and has its origin in the semiconductor industry, which is characterized by small market windows and high sales volumes. The principle of demonstrating lost asset value due to delayed market entry can, however, be applied to any industry, including the software industry. The important issues to find out are the overall asset value curve related to the market window and the resulting revenue curve in the case of a delayed market entry. Manufacturer organizations capable of defining these curves obtain valuable input for decision-making, especially when making cost-benefit analyses. These require additional information, namely, the costs. The model will be extended with cost functions, describing the pre-release development costs and the post-release operational costs.

For the case of on-time delivery of a product, three functions are defined:

- Asset value C (Figure 1):
 - Product lifetime is equal to $2W$ with peak C_{max} at $T_r + W$.
 - Time of market entry defines a triangle, representing market penetration.
 - Triangle area equals total asset value.
- Development cost I (Figure 3):
 - Product development time is equal to T_r with peak I_{max} at $T_r/2$.
 - Start of project at $T=0$ defines a triangle, representing development cost distribution.
 - Triangle area equals total development cost.
- Operational cost M (Figure 3):
 - Peak M_{max} at $T_r + W$.
 - Time of market entry defines a triangle, representing operational cost distribution.
 - Triangle area equals total operational cost.

This leads to the following equations:

$$C = \frac{1}{2} \cdot 2W \cdot C_{max} \quad (4)$$

$$I = \frac{1}{2} \cdot T_r \cdot I_{max} \quad (5)$$

$$M = \frac{1}{2} \cdot 2W \cdot M_{max} \quad (6)$$

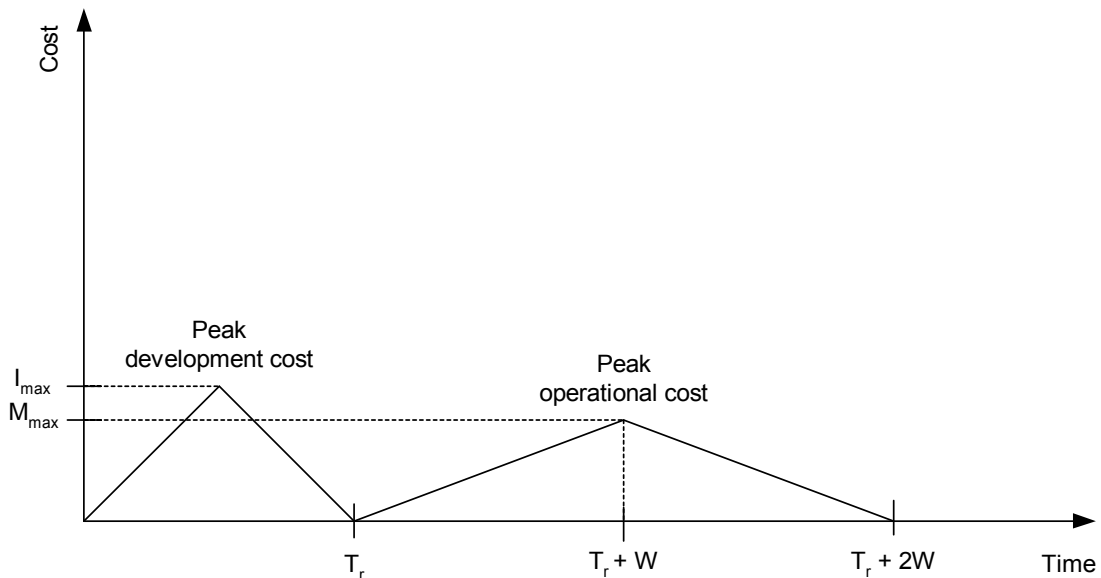


Figure 3: Development cost and operational cost model (on-time entry)

Combining the three models, the resulting profit or net asset value can be calculated:

$$\begin{aligned} NAV &= -I + C - M \\ &= \frac{1}{2} \cdot 2W \cdot C_{max} - \frac{1}{2} \cdot T_r \cdot I_{max} - \frac{1}{2} \cdot 2W \cdot M_{max} \end{aligned} \quad (7)$$

The resulting break even point and net asset value are given in Figure 4.

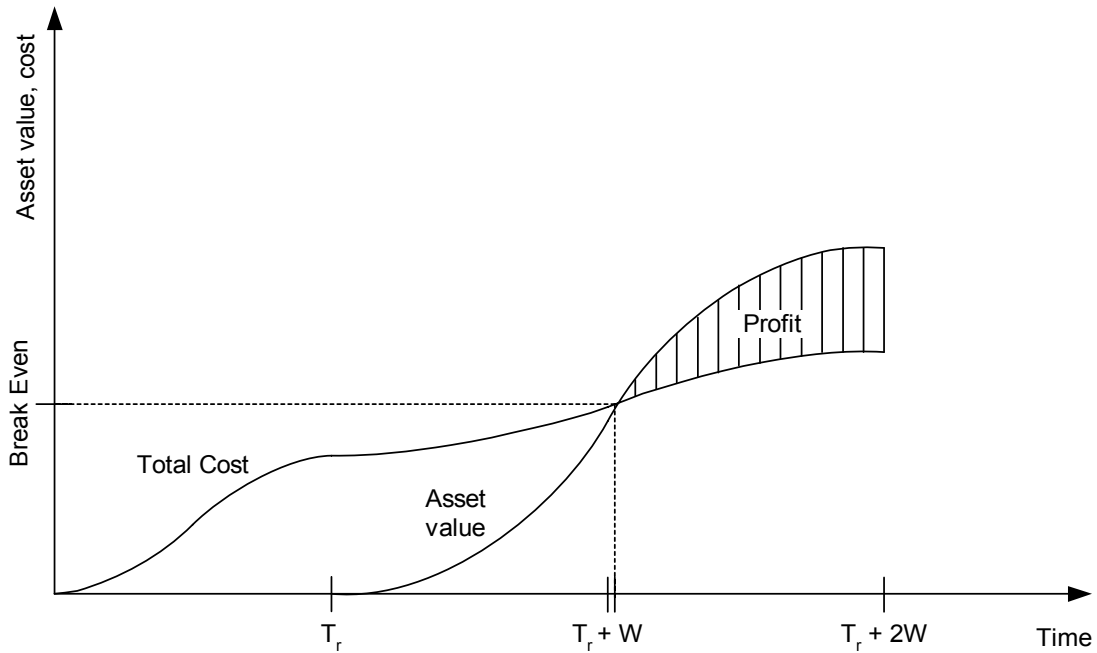


Figure 4: Profit model (on-time entry)

In the case that a product launch is not meeting its initially planned release date T_r (start of market window) but has a delay D , and assuming that the peak of the market is in the middle of the product life-cycle, the three functions become:

- Asset value C (Figure 1):
 - Product life time is equal to $2.W$; product is released at $T_r + D$, with peak C_{max}' at $T_r + W$, with $C_{max}' = ((W-D)/W) \cdot C_{max}$.
 - Time of market entry defines a triangle, representing market penetration.
 - Triangle area equals total asset value.
- Development cost I (Figure 5):
 - Product development time is equal to $T_r + D$ with peak I_{max}' at $(T_r + D)/2$, with $I_{max}' = ((T_r + D)/T_r) \cdot I_{max}$.
 - Start of project at $T=0$ defines a triangle, representing development cost distribution.
 - Triangle area equals total development cost.
- Operational cost M (Figure 5):
 - Peak M_{max}' at $T_r + W$, with $M_{max}' = ((W - D)/W) \cdot M_{max}$.
 - Time of market entry defines a triangle, representing operational cost distribution.
 - Triangle area equals total operational cost.

This leads to the following equations:

$$C' = \frac{1}{2} \cdot (W-D+W) \cdot ((W - D)/W) \cdot C_{max} \quad (8)$$

$$I' = \frac{1}{2} \cdot (T_r + D) \cdot ((T_r + D)/T_r) \cdot I_{max} \quad (9)$$

$$M' = \frac{1}{2} \cdot (W-D+W) \cdot ((W - D)/W) \cdot M_{max} \quad (10)$$

Combining the three functions, the resulting profit or net asset value can be calculated:

$$\begin{aligned} NAV' &= -I' + C' - M' \\ &= \frac{1}{2} \cdot (W-D+W) \cdot ((W-D)/W) \cdot C_{max} \\ &\quad - \frac{1}{2} \cdot (T_r + D) \cdot ((T_r + D)/T_r) \cdot I_{max} \\ &\quad - \frac{1}{2} \cdot (W-D+W) \cdot ((W-D)/W) \cdot M_{max} \end{aligned} \quad (11)$$

The resulting break even point and net asset value are given in Figure 6.

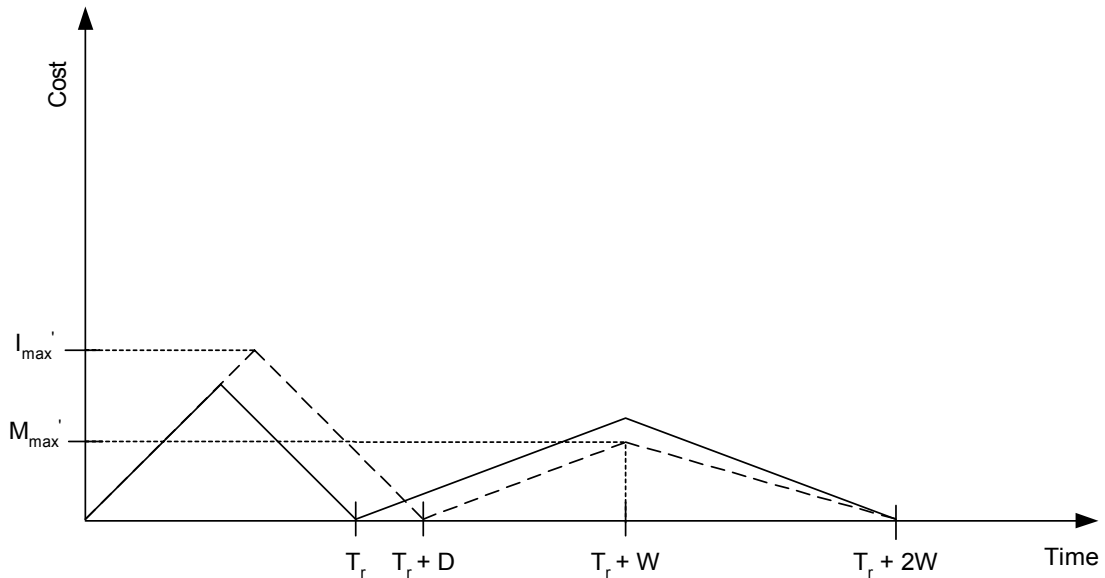


Figure 5: Development cost and operational cost model (delayed market entry)

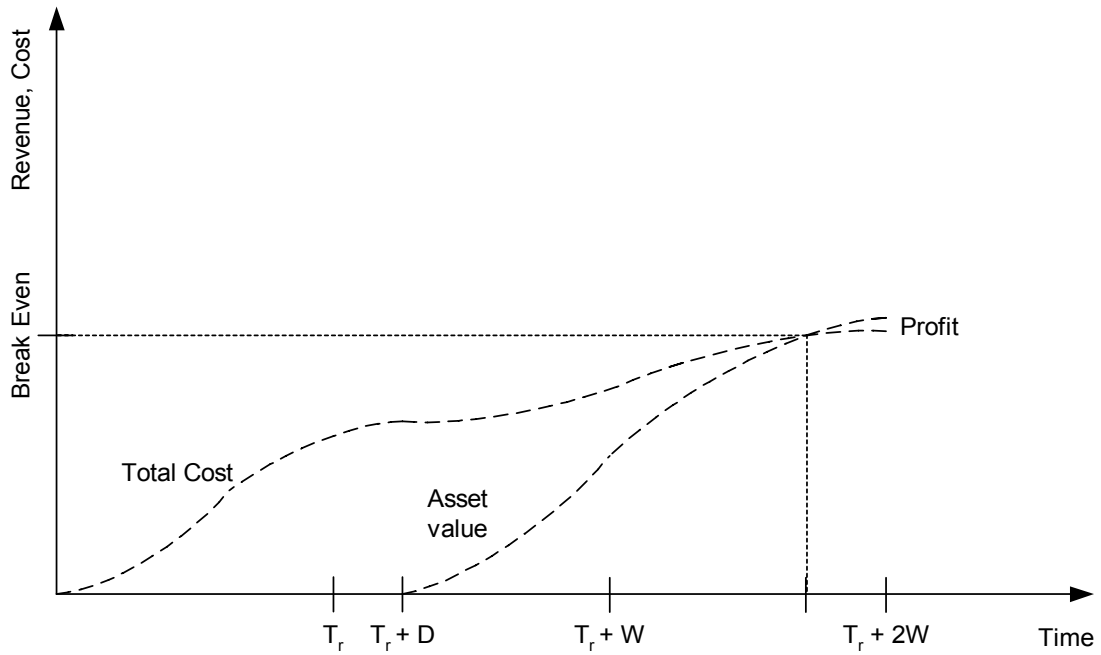


Figure 6: Profit model (delayed market entry)

Figure 7 presents an example of the relative consequences of a delayed market entry on the profit or net asset value.

$T_r = 50$ weeks, $W = 50$ weeks	$D = 0$ wk	$D = 2.5$ wk	$D = 5$ wk	$D = 7.5$ wk	$D = 10$ wk
Asset value (C)	-	-7%	-14%	-21%	-28%
Development cost (I)	-	10%	21%	32%	44%
Operational cost (M)	-	-7%	-14%	-21%	-28%
Net asset value (NAV) ($C_{max} = 10, I_{max} = 5, M_{max} = 5$)	-	-25%	-50%	-75%	-100%

Figure 7: Example of consequences for delayed market entry

In the next section, the theory regarding capital budgeting methods and market entry trade-off models will be applied to software release decisions.

4 A Trade-off Model for Market Entry Decisions

In theory, the model presented in the previous section enables a software manufacturer to evaluate different software release alternatives. Furthermore, the model might also be used when evaluating investment decisions or different product design alternatives. In practice, further refinements are necessary to make it more usable when comparing software release alternatives.

The value of expected cash inflows and cash outflows should be discounted back to the present by taking into account the time value of money. Therefore, alternatives should be expressed in their net present value to enable comparison. The determinants of economic value of a software product, whereby a software project is separated into a development and an operations phase, are (see Figure 8):

- T is the development time or time-to-market, defined as the elapsed time between the commitment to invest in the project and the time at which the product is released (start of first major cash inflow from revenues or cost savings).
- I is the total present value at time 0 of all cash outflows from the time the decision to invest is made to the product release date.
- C is the total present value at time T of the cash inflows that the product is expected to generate during its lifetime (revenues, direct cost savings), also known as the asset value or revenue.
- M is the total present value at time T of all cash outflows of the operations phase (corrective and adaptive/perfective maintenance), also known as operational costs.
- r is the discount rate representing the systematic risk in the software product.

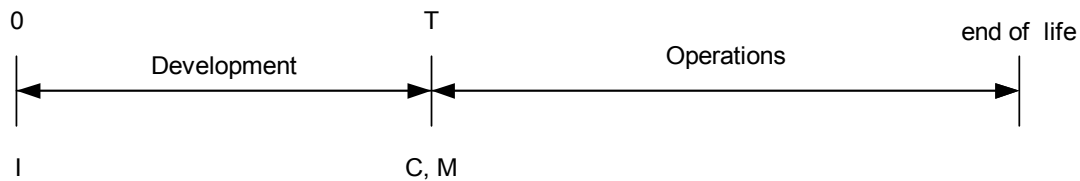


Figure 8: Determinants of economic value for a (software) product

NPV is taken as the capital budgeting method, being the discounted present value of the difference between total cash inflows and total cash outflows. NPV is the most commonly employed method for investment evaluations. It can be calculated as the net asset value, equal to $C - M$, from which the cost of development I is deducted, with all cash inflows and outflows expressed in their present value. Equation:

$$NPV = -I + (C - M) / (1 + r)^T \quad (12)$$

Different alternatives can be evaluated by comparing their NPV values. Erdogmus introduced a method for comparative evaluation of software development strategies based on NPV-calculations, used to compare custom-built systems and systems based on Commercial Off-The-Shelf (COTS) software [Er99]. Erdogmus distinguishes comparison metrics for various variables that influence the NPV of a project. This method is used as the basis for the definition of a model to reflect software release decisions.²

Let V be a variable and let V_a and V_b denote the value of variable V for alternative A and B, respectively. A comparison metric is a function of V_a and V_b and for a specific value of a comparison metric, alternative A is said to be favourable to B if for the value of that metric the project NPV for alternative A is superior to the project NPV for alternative B when everything else is equal. Metrics distinguished are:

- *Premium*: the relative difference between two quantities (if the value of alternative A is 20% more than the value of alternative B, the premium equals 0.2). A negative premium is a penalty.
- *Advantage*: the natural logarithm of the ratio of two quantities (for mathematical convenience and ease of interpretation³). A negative advantage is a disadvantage.

² Some definitions have been adapted, with some metrics added, changed, or disregarded. Further, after communication with the author, some formulas have been corrected. See [Er99] to make a detailed comparison.

³ Some basic properties are: $e^0 = 1$; $\log 1 = 0$; $e^{\log a} = a$; $\log a + \log b = \log (a * b)$, $\log a - \log b = \log (a/b)$, $d e^x / dx = e^x$. Note that when 'log' is used here, the natural logarithm 'log_e' or 'ln' is meant.

- *Incentive*: normalized difference of two quantities to allow comparison of alternatives of variable scale. A negative incentive is a disincentive.

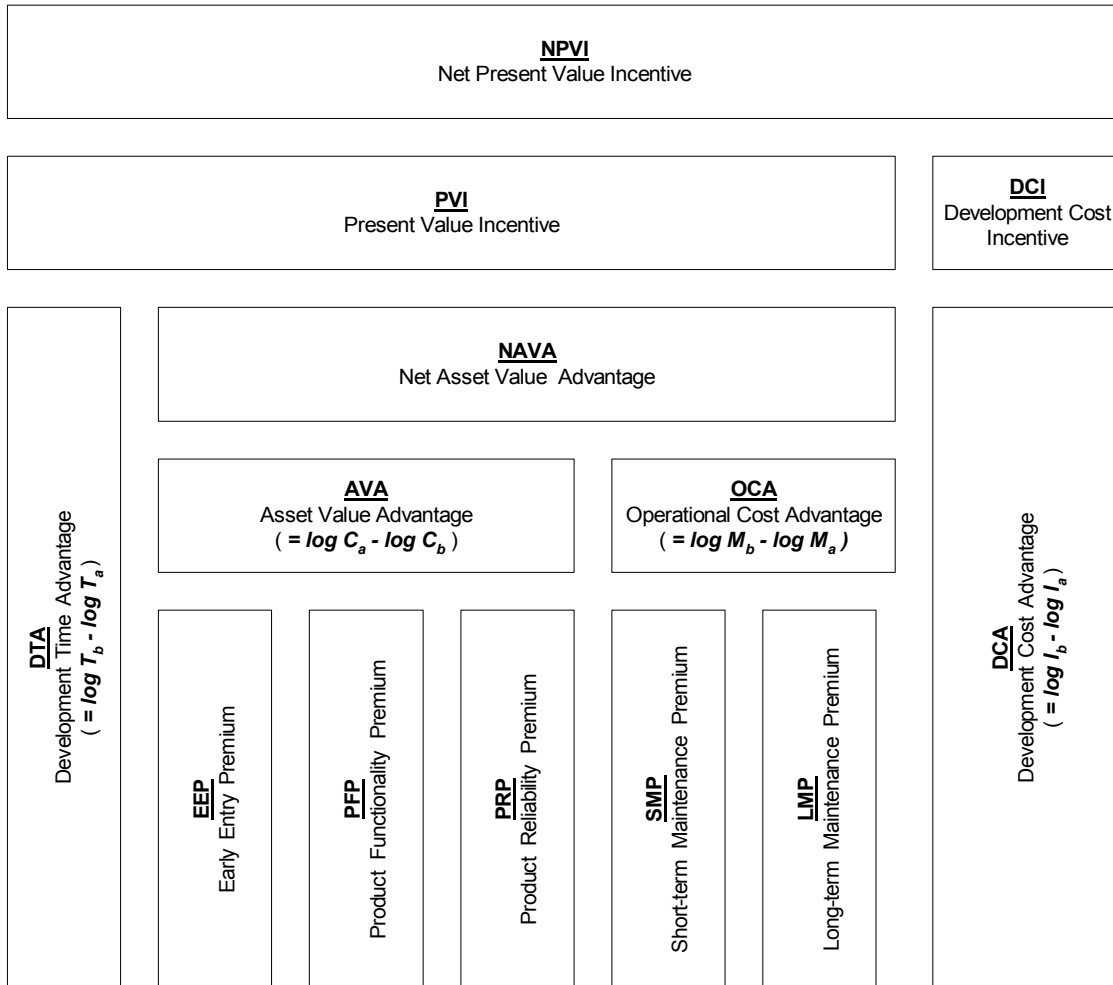


Figure 9: Breakdown of NPV Incentive to lower-level metrics to compare two release alternatives

The structure of the model, with the breakdown into incentives, advantages and premiums, is illustrated in Figure 9 and will be further explained.

At the lowest level, two categories of premium metrics are distinguished:

- *Asset value premiums*. Three variables influencing the asset value are considered, namely, early market entry (*EEP*), product functionality (*PFP*) and product reliability (*PRP*).
- *Operational cost premiums*. Two variables influencing the operational cost are considered, namely, the short-term costs for corrective maintenance (*SMP*) and the long-term costs for adaptive/perfective maintenance (*LMP*).

The Asset Value Advantage *AVA* is equal to the expected increase in future cash inflows (difference between the two alternatives C_a and C_b) and comprises the contribution of the Early Entry Premium *EEP*, the Product Functionality Premium *PFP*, and the Product Reliability Premium *PRP*:

$$\begin{aligned}
 AVA &= \log C_a - \log C_b \\
 &= \log (C_b + C_b \cdot (EEP + PFP + PRP)) - \log C_b \\
 &= \log (1 + EEP + PFP + PRP)
 \end{aligned} \tag{13}$$

The Operational Cost Advantage *OCA* is equal to the future cash outflows savings (difference between the two alternatives M_b and M_a) when the product is transferred to the operational phase, and comprises the contribution

of the Short-term Maintenance Premium *SMP* (corrective maintenance) and the Long-term Maintenance Premium *LMP* (adaptive/perfective maintenance):

$$\begin{aligned}
 OCA &= \log M_b - \log M_a \\
 &= \log M_b - \log (M_b - M_b \cdot (SMP + LMP)) \\
 &= \log (1 / (1 - SMP - LMP)) \\
 &= -\log (1 - SMP - LMP)
 \end{aligned} \tag{14}$$

The Asset Value Advantage *AVA* (expected future cash inflows) and the Operational Cost Advantage *OCA* (expected future cash outflows) are combined in the Net Asset Value Advantage *NAVA*:

$$\begin{aligned}
 NAVA &= \log NAV_a - \log NAV_b \\
 &= \log (C_a - M_a) + \log (C_b - M_b) \\
 &= \log (e^{AVA} C_b - M_b / e^{OCA}) - \log NAV_b
 \end{aligned} \tag{15}$$

The Present Value Incentive *PVI* is derived from the Net Asset Value Advantage *NAVA*, taking into account the discount rate *r* and normalizing it to the base alternative *NAV_b*:

$$\begin{aligned}
 PVI &= [PV_a - PV_b] / NAV_b \\
 &= [(NAV_a / (1 + r)^{T_a}) - (NAV_b / (1 + r)^{T_b})] / NAV_b \\
 &= (1 / (1 + r)^{T_b}) \cdot (e^{NAVA} / (1 + r)^{T_a - T_b} - 1) \\
 &= (1 / (1 + r)^{T_b}) \cdot (e^{NAVA} / (1 + r)^{\beta} - 1)
 \end{aligned} \tag{16}$$

with:

$$\beta = T_b ((1/e^{DTA}) - 1) \tag{17}$$

The Development Cost Incentive *DCI* is the normalized difference of the development cost between the two alternatives considered, *I_b* and *I_a*:

$$\begin{aligned}
 DCI &= (I_b - I_a) / I_b \\
 &= 1 - 1 / e^{DCA}
 \end{aligned} \tag{18}$$

This leads to the final Net Present Value Incentive *NPVI*, normalized to the project scale:

$$\begin{aligned}
 NPVI &= (NPV_a - NPV_b) / (NAV_b + I_b) \\
 &= (PV_a - I_a - PV_b + I_b) / (NAV_b + I_b) \\
 &= (PVI \cdot NAV_b + DCI \cdot I_b) / (NAV_b + I_b)
 \end{aligned} \tag{19}$$

The original method was developed to compare different product development strategies to make investment appraisals. The adjusted method can be used in a similar fashion, but reflects more accurately specific criteria related to a software release decision: reliability and expected short-term and long-term maintenance costs. Usage of both models is not restricted to appraisal of initial product development strategies. They may also be used during product development for instance, to evaluate different design alternatives or, important for this study, to evaluate different release alternatives.⁴ Differences with respect to investment appraisals are:

- The development costs spent already should be disregarded, as they are past and irreversible outflows (*sunk costs*).
- Instead of the total development time, only the time left until the release date is to be taken.

In Figure 10, an example comparison of two release alternatives is illustrated: a test alternative A against a base alternative B. It is assumed that the test alternative A has the following differences with respect to the base alternative:⁵

- The product is released earlier in order to capture an early market-entry reward (*DTA* = 0.29).
- It is assumed that this has a positive effect of 50% on the cash inflows (*EEP* = 0.50), but this effect is largely cancelled because customers are presented with less functionality (20% less cash inflows, *PFP*

⁴ Sassenburg states that the appraisal of a software release decision can be seen as the re-appraisal of an earlier investment decision, now probably with less uncertainty. Conversely, the appraisal of an investment decision is the pre-appraisal of a software release decision, probably with high uncertainty. In other words, available methods or models can be applied to both decision types [Sa03].

⁵ Arbitrary values have been taken for the parameters *T*, *I*, *C*, *M*, and *r* merely to demonstrate the method.

= -0.20) and a lower reliability level (25% less cash inflows, $PRP = -0.25$). The overall asset value advantage is, however, still positive (5%, or $AVA = 0.05$).

- It is assumed that the early release with a lower reliability level will influence the cash outflows M during the operations phase in a negative way: an increase of 10% of the short-term maintenance costs ($SMP = -0.10$) and an increase of 40% of the long-term maintenance costs ($LMP = -0.40$). This leads to a negative overall operational cost advantage ($OCA = -0.41$).

From these assumptions, the net asset value advantage $NAVA$ and the resulting present value incentive PVI can be calculated. The result is that the test alternative is less attractive than the base alternative ($PVI = -0.06$). However, assuming that releasing the product earlier will reduce the initial cash outflows I for product development ($DCA = 0.22$), an incentive for the test strategy is revealed, as the net present value incentive $NPVI$ is positive ($NPVI = 0.02$).

Parameters	Base alternative	Test alternative	Name	
T: Development time	4	3	T_b T_a	
I: Initial cash outflows	100	80	I_b I_a	
C: Expected cash inflows	300	-	C_b	
M: Expected cash outflows	100	-	M_b	
R: Discount rate	0.1	0.1	r	
Premiums	Base alternative	Test alternative	Name	Value
Early Market Entry	-	50%	EEP	0.50
Product Functionality	-	-20%	PFP	-0.20
Product Reliability	-	-25%	PRP	-0.25
Short-term Maintenance	-	-10%	SMP	-0.10
Long-term Maintenance	-	-40%	LMP	-0.40
Advantages	Base alternative	Test alternative	Name	Value
Development Time	T_b	T_a	DTA	0.29
Development Cost	I_b	I_a	DCA	0.22
Asset Value	C_b	315	AVA	0.05
Operational Cost	M_b	150	OCA	-0.41
Net Asset Value	200	165	$NAVA$	-0.19
Incentives			Name	Value
Present Value	137	124	PVI	-0.06
Development Cost	I_b	I_a	DCI	0.20
Net Present Value	37	44	$NPVI$	0.02

Figure 10: Example comparison of two alternatives

The presented method also offers the possibility to perform sensitivity analysis when one or more variables are changed. Further, the method can be used in cases of constrained decision-making, for instance when there is a budget constraint (development cost) or a time constraint (small market window). In such cases, calculations may be combined with, for example, the Lagrangian Multiplier Method. This method imposes a penalty upon any proposed solution that is proportional to the extent to which the constraint is violated. By choosing a large enough constant of proportionality, the solution can be forced into compliance with the constraint. However, this solution requires knowledge about the relationships between the different parameters (premium metrics).⁶

5 Optimal Information Level

Using the model described in the previous section, the main sources of information as input to the decision-making process can be derived (see also Figure 11):

1. First, external market-related information about demand functions (market window and expected revenues or cash inflows) is needed to estimate the shape of the asset value function $C(t)$.
2. Regarding product-related information, two different sources can be distinguished:
 - Information regarding the quality of the product must be obtained through verification activities. These verification activities should supply information regarding the quality of the product in terms of its implemented functional and non-functional requirements as well as

⁶ However, making a strictly computational release decision must be considered a utopia: not all parameters can be quantified without uncertainties and in a practical context the exact relationships between all parameters will be unknown.

- information regarding the effort needed to further improve the quality of the product. This information determines the shape of the development cost function $I(t)$, not only during the testing phase, but also during the development phase prior to the testing phase.
- Information regarding the maintainability of the product is needed to estimate the shape of the post-release maintenance function $M(t)$. This information should cover both the expected short-term corrective maintenance cost and the expected long-term adaptive/perfective maintenance cost.
3. Finally, information regarding the relationship between external market-related and internal product-related information is determined by the way the quality of the product, related to $I(t)$, and the maintainability of the product, related to $M(t)$, affect the asset value function $C(t)$. In other words, how do changes in the product quality and product maintainability affect the demand function?

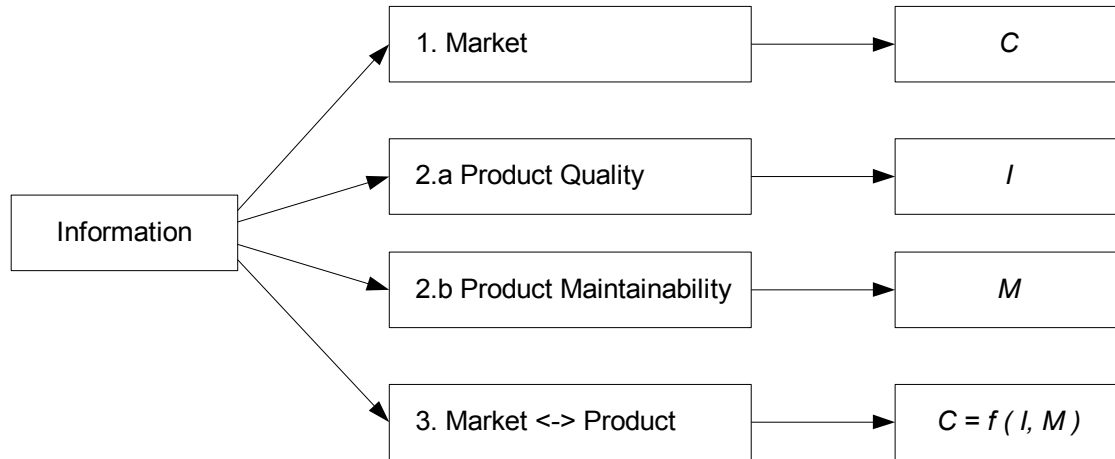


Figure 11: Main sources of information as input to the decision-making process

Ideally, decision-makers would know the exact shapes of these functions and the relationships between the different functions. However, this is considered to be difficult if not impossible. Research indicated that software manufacturers face serious problems in estimating the exact shape of the development cost function $I(t)$, especially during the testing phase, and efforts to determine the shape of the post-release maintenance cost function $M(t)$ prior to a release decision were not found at all [SB04]. In order to improve a software manufacturer's decision-making process, the objective should be to search for sufficient information relating to the three sources specified above to enable decision-making without 'too much' uncertainty. Put differently, a decision-maker should look for the point of optimality. Below this point, uncertainty is high and might confront a software manufacturer with missing a market window or with unexpectedly high post-release maintenance costs. Beyond this point, the extra information leads to additional costs that outweigh the benefits (law of diminishing returns). Using the argumentation of Gigerenzer [Gi04], it is assumed that this point of optimality can probably not be determined precisely, neither ex ante nor post ante. Therefore, instead of finding the point of optimality, software manufacturers will in a practical setting be forced to search for a zone of cost effectiveness: a bandwidth in which the marginal net asset value is equal or close to zero. The information level is considered to be cost effective compared to higher or lower levels of information if it is:

- (1) Less costly and at least as effective;
- (2) More costly and more effective with an added efficacy that is worth paying the additional price for;
- (3) Less effective and less costly, where the additional cost of additional information is too high for the additional benefits provided.

It is assumed that reducing the total uncertainty space to within this zone of cost effectiveness will have two positive effects: the number of scenarios to be considered might be reduced, and the chance of occurrence of each scenario might be better quantified with probability or possibility values. This will make the NPVI method a better candidate for evaluating different release alternatives.

6 Conclusions

It is concluded that the NPVI method presented here can be used to illustrate the market entry trade-off for a software product. This method allows the comparison of different alternatives during different project phases, including release alternatives. The presented method with the hierarchical metrics framework offers the possibility to analyse the sensitivity of the higher-level incentives to variations in medium-level advantages and

lower-level premiums. Use of the method is not restricted to specific manufacturer types. Environmental and internal conditions under which a software manufacturer is operating will determine the market or demand window and thereby affect the expected cash inflows and cash outflows. The underlying method is not influenced. One final remark is felt necessary. Applying the presented method means that all costs and benefits are quantified in monetary terms, including the costs of possible fatalities and injuries in the case of, for instance, safety-critical systems. It is however difficult if not impossible, and it may even be unethical, to place any sort of value on human life.

Bibliography

- [BS00] Boehm, B.W., Sullivan, K.J., *"Software Economics: A Roadmap"*, ACM Press, 2000.
- [Er99] Erdogmus, H., *"Comparative evaluation of software development strategies based on Net Present Value"*, First International Workshop on Economics-driven Software Engineering Research, Toronto (Canada), 1999.
- [Gi04] Gigerenzer, G., *"Striking a Blow for Sanity in Theories of Rationality"*, in *Essays in honor of Herbert Simon*, Augier, B., March, J.D. (Eds), Cambridge, MA: MIT Press, 2004.
- [HM99] Hatch, N.W., Macher, J.T., *"Mitigating the Time to Market and Performance Tradeoff: Innovation and Knowledge Management in New Technology Introductions"*, Strategic Management Society 19th Annual International Conference, Berlin, Germany, 1999.
- [Hu02] Humphrey, W.S., *"Winning with Software: An Executive Strategy"*, Addison-Wesley, 2002.
- [Li95] Liu, J., *"Detailed Model shows FPGAs' True Costs"*, EDN, pp. 153-158, 1995.
- [LS00] Littlewood, B., Strigini, L., *"Software Reliability and Dependability: a Roadmap"*, Proceedings of the 22nd International Conference on Software Engineering, 2000.
- [Rt02] RTI, *"The Economic Impacts of Inadequate Infrastructure for Software Testing"*, Planning Report 02-3, Prepared by RTI for National Institute of Standards and Technology, U.S. Department of Commerce, 2002.
- [Sa03] Sassenburg, H., *"Vrijgavecriteria moeten rol spelen bij investeringsbesluit"*, *Automatisering Gids*, April 11th, p. 15 (in Dutch), 2003.
- [SB04] Sassenburg, H., Berghout, E., *"A managerial release-decision model for IT-applications"*, 11th European Conference on the Evaluation of IT, November 11-12, Amsterdam (The Netherlands), 2004.
- [SG04] Standish Group, *"CHAOS: A Recipe for Success"*, Standish Group International, Inc., 2004.