Market Entry Decisions: Numbers or Politics?

Hans Sassenburg

SE-CURE AG (www.se-cure.ch), CH-3775 Lenk, Switzerland

hsassenburg@se-cure.ch

Abstract:

In unpredictable software manufacturer organizations, it is difficult to determine when a software product will be released, the features the product will have, the associated development costs or the resulting product quality. The NPVI-method is presented, enabling a software manufacturer to compare and evaluate different release or market entry strategies. However, information has its price in time and cost, forcing decision-makers to make a trade-off between search costs and opportunity costs. In addition, decision-makers simplify the real world, as they cannot escape the diverse psychological forces that influence individual behaviour. Combined with the potential presence of sources of conflict, this often leads to the situation where different stakeholders experience difference aspiration levels. As such, satisficing behaviour where decision-makers try to find consensus and choose a satisfactory release alternative is a good characterisation of the software release decision-making process as found in practice. Successful adoption of the NPVI-method requires that software manufacturers reach the zone of cost effectiveness for the perfection of information; a zone where numbers make business sense, and can be convincingly used to support informed decision-making.

Keywords

Market entry strategies, software releasing, decision-making, satisficing behaviour

1 Introduction

A relatively unexplored area in the field of software management is the release or market entry decision, deciding whether or not a software product can be transferred from its development phase to operational use. As many software manufacturers behave in an unpredictable manner [1] [12], they have difficulty in determining the 'right' moment to release their software products. It is a trade-off between an early release, to capture the benefits of an earlier market introduction, and the deferral of product release, to enhance functionality, or improve quality. A release decision is a trade-off where, in theory, the objective is to maximize the economic value. Inputs into the release decision are expected cash inflows and outflows if the product is released. What is the market window? What are the additional pre-release development costs when continuing testing and the expected post-release maintenance costs when releasing now?

2 Maximizing Behaviour

A market entry decision is a trade-off between early release to capture the benefits of an earlier market introduction (a larger installed base), and the deferral of product release to enhance functionality, or improve quality. For many software manufacturers, especially those operating in mass markets, this is the point of no return. At first sight, this trade-off seems not to be of any special nature, from a strictly economic perspective. If a software product is released 'too early', a software product with less functionality and/or significant defects would be released to intended users and the software manufacturer incurs post-release costs of later fixing 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. These two alternatives need to be compared, to determine which alternative maximizes economic value (revenues minus costs). When the perspective of maximizing behaviour is assumed, the primary objective of a software manufacturer is to maximize long-term expected value. In that case, it is needed to be able to evaluate and compare different market entry strategies: which strategy will maximize economic value?

Product life-cycle models, as for instance frequently used in the semiconductor industry, can be used to demonstrate the effects on revenues of a delayed market entry [5] [6] [13]. By extending these models with cost functions for pre-release development costs and post-release operational costs the effects on profits can be calculated as well. Based on these profit models, a method was defined using the NPV capital budgeting method. Different alternatives can be evaluated by comparing their NPV values. Erdogmus introduces 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 [2]. Erdogmus distinguishes comparison metrics for various variables that influence the NPV of a project. This method was used as the basis for the definition of a method to reflect market entry decisions for software-intensive systems. The resulting so-called NPVI-method expresses the difference between two alternatives in a single variable. This variable, called the Net Present Value Incentive, is calculated from various underlying metrics, and measures the economic incentive to favour one alternative over another. The metrics are classified into premium metrics at the lowest level, advantage metrics at the medium level and incentive metrics at the highest level. See Figure 1.

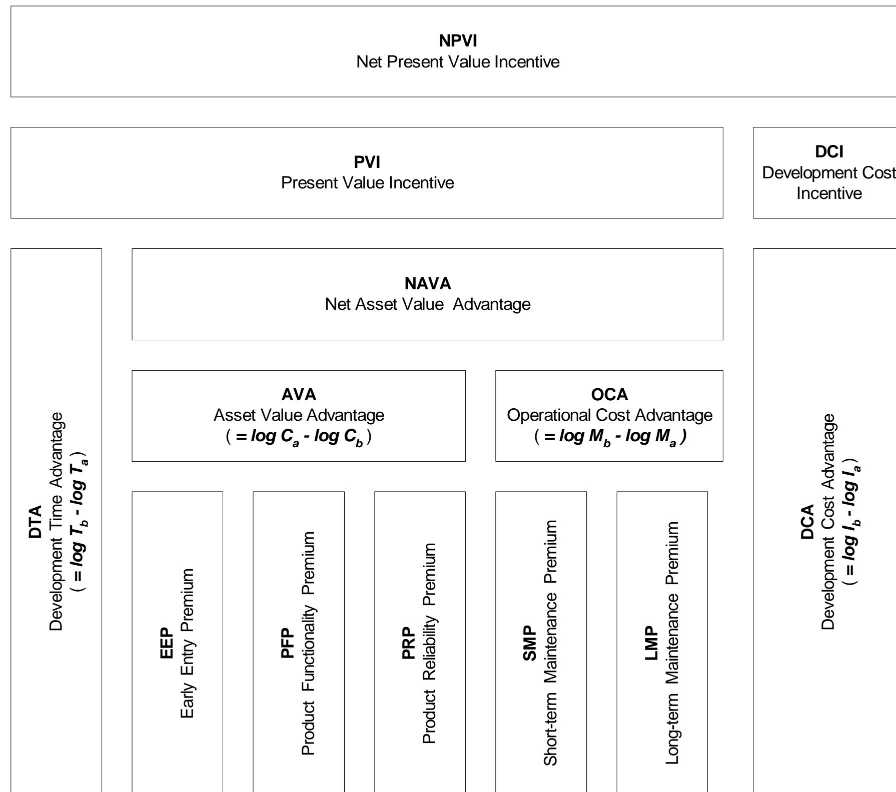

Figure 1. Breakdown of NPV Incentive to Lower-level Metrics.

This method allows the comparison of different alternatives during different project phases, including release alternatives. 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 is the contribution of the Early Entry Premium *EEP*, the Product Functionality Premium *PFP* and the Product Reliability Premium *PRP*:

$$\begin{aligned} AVA &= \log C_a - \log C_b \\ &= \log [\, C_b + C_b \,.\, (EEP + PFP + PRP) \,] - \log C_b \\ &= \log (\, 1 + EEP + PFP + PRP \,) \qquad (1) \end{aligned}$$

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

$$OCA = \log M_b - \log M_a$$
$$= \log M_b - \log [M_b - M_b . (SMP + LMP)]$$
$$= \log [1 / (1 - SMP - LMP)]$$
$$= - \log (1 - SMP - LMP) \quad (2)$$

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*:

$$NAVA = \log NAV_a - \log NAV_b$$
$$= \log (C_a - M_a) + \log (C_b - M_b)$$
$$= \log (e^{AVA} C_b - M_b / e^{OCA}) - \log NAV_b \quad (3)$$

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:

$$PVI = [PV_a - PV_b] / NAV_b$$
$$= [(NAV_a / (1 + r)^{Ta}) - (NAV_b / (1 + r)^{Tb})] / NAV_b$$
$$= [1 / (1 + r)^{Tb}] . [e^{NAVA} / (1 + r)^{Ta - Tb} - 1]$$
$$= [1 / (1 + r)^{Tb}] . [e^{NAVA} / (1 + r)^{\beta} - 1] \quad (4)$$

with:

$$\beta = T_b [(1/e^{DTA}) - 1] \quad (5)$$

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

$$DCI = (I_b - I_a) / I_b$$
$$= 1 - (1 / e^{DCA}) \quad (6)$$

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

$$NPVI = (NPV_a - NPV_b) / (NAV_b + I_b)$$
$$= (PV_a - I_a - PV_b + I_b) / (NAV_b + I_b)$$

$$= \quad (PVI \,.\, NAV_b + DCI \,.\, I_b) / (NAV_b + I_b) \qquad (7)$$

3 Optimizing Behaviour

Maximizing behaviour assumes that decision-makers have complete information about costs and benefits associated with each option. They compare the options on a single scale of preference, value or utility. Modern behavioural economics acknowledge however, that the assumption of perfect (complete and reliable) information is implausible. Etzioni and Amitai argue that because, normally, limitations on information will exist, it is impossible to undertake the precise analysis necessary to maximize economic objectives [3]. Many economists put similar, and other arguments, against the case for maximizing behaviour [4] [7]. Rather than assuming decision-makers possess all relevant information for making choices, information is, itself, treated as a commodity, something that has a price in time and/or money. This argument of limitations on information can be used to 'soften' maximizing behaviour to optimizing behaviour, where an individual decision-maker makes a trade-off between information perfection (completeness and reliability) and the cost related to searching for additional information.

This relationship is given in Figure 2. On the horizontal axis, *Information perfection* is measured, which is knowledge about the decision outcome of an alternative. When information perfection equals *100%*, the information is complete and reliable, or, supposedly, perfect. The vertical axis measures the value, cost and yield (marginal value) as a function of information perfection on the horizontal axis. *Value* refers to how desirable a particular decision outcome is considering the value of the alternative, whether in money, satisfaction or other benefit. The value curve $V(i)$ rises steadily. *Cost* is the cost involved in searching for alternatives, for example, extending information perfection. The cost curve $C(i)$ moves in the opposite direction, rising rather slowly at the start because the initial information requires relatively little effort. *Time* is the time involved in searching for alternatives and moves in the same direction as the cost function. Additional information becomes more difficult to obtain and the associated cost and time increase exponentially. *Yield* is the difference between value and cost (net value). The yield curve $Y(i)$, the difference between the value and cost functions, reduces sooner, and more steeply than the value curve. *Yield* represents the net value with the point of diminishing returns, or *point of optimality* Y^*, the point where this curve reaches its maximum with the corresponding values I^*, V^* and C^*. Beyond this point, the cost of acquiring additional information outweighs the value or benefit.

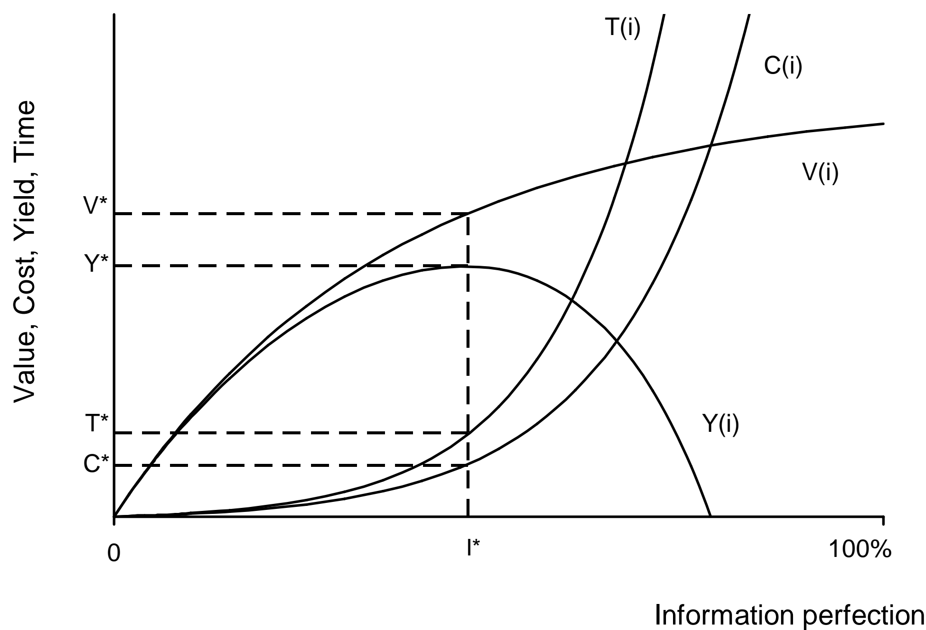

Figure 2: Value (V), Cost (C), Time (T) and Yield (Y) as a Function of Information Perfection [9].

A decision-maker should look for the point of optimality. Below this point, uncertainty is high and might confront a software manufacturer with releasing unexpectedly high post-release maintenance costs. Beyond this point, the extra information leads to additional costs that outweigh the benefits (law of diminishing returns). 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.

4 Language

Simon argues that limited cognitive capabilities in decision-makers lead to simplification [11]. A decision-maker simplifies reality, leaves out information and applies heuristics as a consequence of limited cognitive capabilities. Reasons are, for

example, that the decision-maker has limited, unreliable or even too much information, available, or that the search for acceptable alternatives is felt to be too time, and cost, consuming. This problem of computation is classically illustrated by the travelling salesman problem in which the objective is to minimize the travel costs of a salesperson having to visit 50 cities. The 50! calculation is computable but not within a reasonable time horizon. He suggests that in choice situations, people actually have the goal of satisficing, rather than maximizing, or optimizing, and a decision-maker applies heuristic rules of search in a heuristic frame. The heuristic (or cognitive) frame referring to the representation of the problem and solution space, whereas the heuristic rules of search are the algorithms used to find solutions in this solution space [10]. Following this approach, an alternative is satisfactory if a set of criteria exists that minimally describes satisfactory alternatives, and the alternative in question meets, or exceeds, all these criteria [7]. A general corresponding strategy is [8]:

1. Set an aspiration level such that any option that reaches, or surpasses it, is ‘good enough’. The aspiration level is the smallest outcome deemed satisfactory.
2. Begin to enumerate and evaluate the options on offer.
3. Choose the first option which, given the aspiration level, is ‘good enough’.

How can this approach be integrated into the model describing optimizing behaviour? An example is given in Figure 3, incorporating *satisficing* behaviour at individual level (aspiration level for one stakeholder or decision-maker). The aspiration level is a horizontal line and reflects the boundary at, or above, which the decision-maker is satisfied. The aspiration level is given by the line $V = V^{*\prime}$, which denotes that a decision-maker will choose the first option reaching, or surpassing, $V^{*\prime}$ for the value function $V(i)$. In the example of Figure 3, the resulting point of optimality $(I^{*\prime}, Y^{*\prime})$ does not coincide with the point of optimality (I^{*}, Y^{*}) and lies to the left. This is not necessarily the case in general. Satisficing behaviour might also lead to setting an aspiration level where the resulting level of information exceeds I^{*}. In this case, unnecessary costs are incurred, as the resulting cost value exceeds C^{*}. The aspiration level can also consist of a lower and upper boundary. A decision-maker will accept the first option for which:

$$V_{low} \leq V(i) \leq V_{high}$$

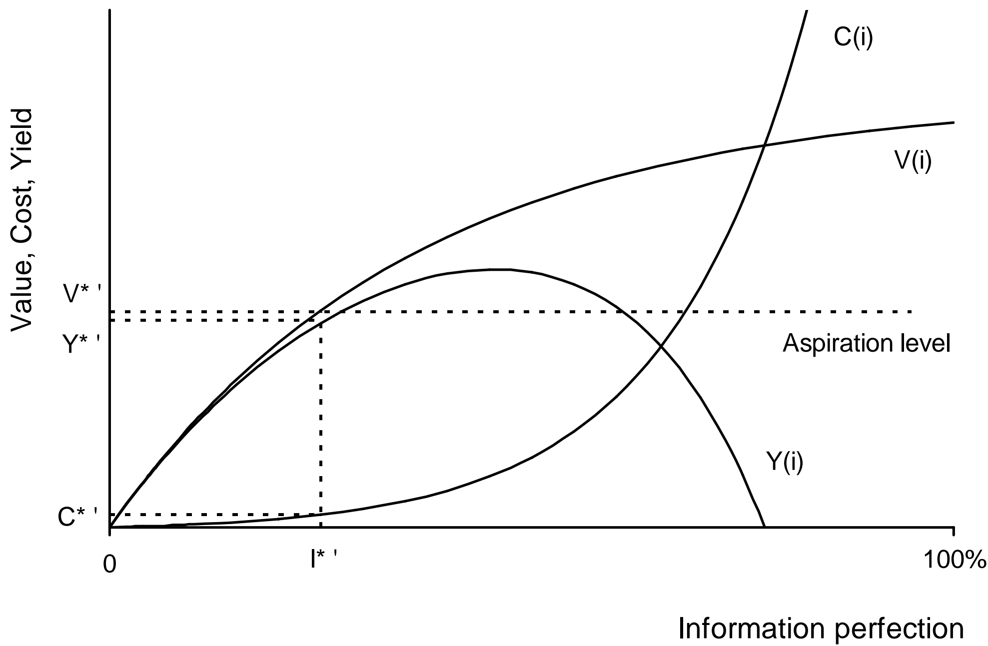

Figure 3. Model to Incorporate Satisficing Behaviour of Single Stakeholder [9].

An aspiration level is not necessarily restricted to the value function *V(i)*. A decision-maker might, for example, set an aspiration level for the information perfection itself, in which case the aspiration level would be a vertical line in Figure 3. There may also be aspiration levels for cost and/or time: an upper boundary constraint C_{high} for the cost function *C(i)* and/or an upper boundary constraint T_{high} for the time function *T(i)*. It is obvious that a solution is only possible if the information level at V_{low} is less than, or equal to, the information level at C_{high} and T_{high}:

$$V^{-1}(V_{low}) < C^{-1}(C_{high}) \quad \text{and:} \quad V^{-1}(V_{low}) < T^{-1}(T_{high})$$

It is concluded here that the notion of optimizing behaviour (imperfect information) as discussed in the previous section, must be extended with the notion of satisficing behaviour. A decision-maker simplifies reality, leaves out information and applies heuristics as a consequence of limited cognitive capabilities. As stakeholders may apply different heuristics and one, or more, determinants of conflict may be present, different stakeholders may arrive at different aspiration levels during the decision-making process. This is illustrated in Figure 4, incorporating satisficing behaviour at group level, and showing the different aspiration levels for three different stakeholders S_a, S_b, and S_c. In the ideal situation, all aspiration levels would be equal and be within the zone of cost effectiveness (or even intersect with the point of optimality). However, in a practical context, with high uncertainty, this is not a likely situation.

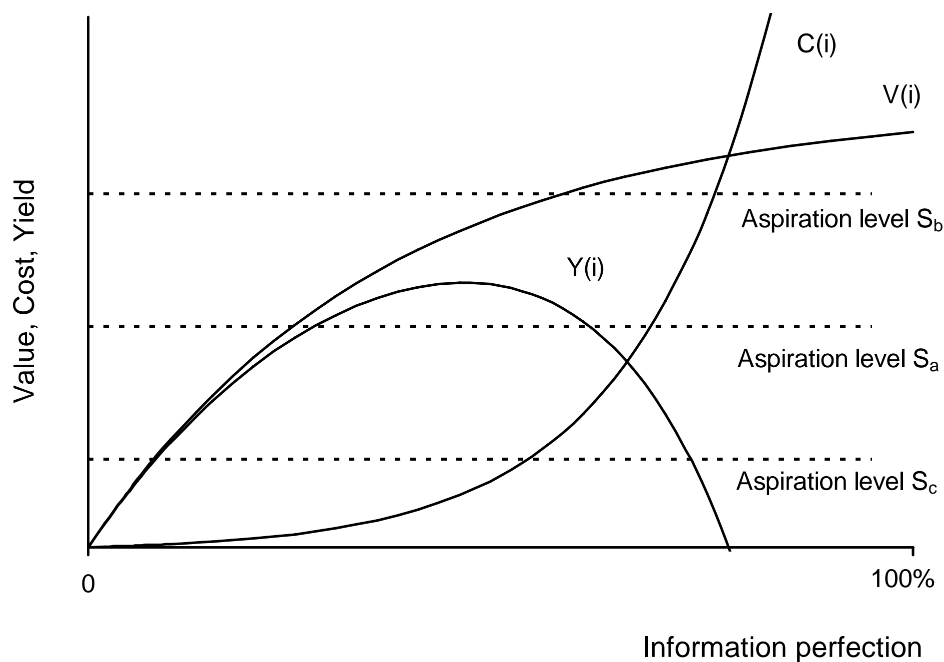

Figure 4. Adjusted Model to Incorporate Satisficing Behaviour of Multiple Stakeholders [9].

5 Design of the document

NPVI-method offers the possibility to evaluate and compare different market entry strategies. Due to its general nature, the method can even be used to compare different product development strategies, architecture or design alternatives, and technology adoption strategies. However, in a practical context, the determination of the optimal release time from a quantitative, financial perspective is difficult, if not almost impossible, due to the presence of uncertainty. Sources of this uncertainty are:

- The state of the art in software engineering technology is such that building software components and products in a predictable way with predictable behaviour is uncommon. Although new innovations may be, or become, available, their application in software industry is severely limited at this stage.
- Information has its price in time and cost, forcing decision-makers to make a trade-off between search costs and opportunity costs.
- Decision-makers simplify the real world, as they cannot escape the diverse psychological forces that influence individual behaviour. Combined with the potential presence of sources of conflict, this may lead to the situation where different stakeholders experience difference aspiration levels.

Increased attention to numbers, by gathering valid information (including historical data) to compare, and evaluate, different release alternatives using the presented NPVI-method and sharing the results among decision-makers is important to reduce uncertainty levels to a more acceptable level, so differences in aspiration levels of stakeholders involved in the decision-making process, are reduced, or eliminated, through convincing information. This is an important contribution to reducing uncertainty, and thus minimizing situations where people lives are put at risk, especially for software products where reliability, safety and security are important non-functional requirements

References

1. Boehm, B.W., Sullivan, K.J., *Software Economics: A Roadmap.* ACM Press, 2000.
2. Erdogmus, H., Comparative evaluation of software development strategies based on Net Present Value*. Proceedings of the First International Workshop on Economics-driven Software Engineering Research*, Toronto (Canada), 1999.
3. Etzioni, Amitai, *Humble Decision Making.* Harvard Business Review (July-August), 1989, 112-126.
4. Jensen, M.C., Meckling, W.H., Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure. *Journal of Financial Economics, 3, 4,* 1976, 305-360.
5. Levitt, M., Economic and Productivity Considerations in ASIC Test and Design-for-Test. *Digest for Papers: Compcon*, 1992, 440-445.
6. Liu, J., Detailed Model shows FPGAs' True Costs. *EDN,* 1995, 153-158.
7. March, J.G., Simon, H.A., *Organizations*. New York: John Wiley, 1958.
8. Pettit, P., Satisficing Consequentialism. Proceedings of the Aristotelian Society, suppl. 58, 1984, 165-176.
9. Sassenburg, H., Design of a Methodology to Support Software Release Decisions: Do the Numbers really Matter? PhD thesis, University of Groningen (The Netherlands), 2006.
10. Simon, H.A., *The Sciences of the Artificial.* Cambridge: MIT Press, 1969.
11. Simon, H.A., *Models of Bounded Rationality.* Cambridge, MIT Press, 1982.
12. Standish Group, *CHAOS: A Recipe for Success.* Standish Group International, Inc., 2004.
13. Wu, C.W., Wei, C.C., The economic models for a VLSI test strategy planning system. *Proceedings SEMICON Taiwan 97*, Test Seminar, 1997, 87-92.