How Much Should You Invest In Software Security?

Kelce S. Wilson, PhD, MBA, J.D.
Research In Motion
122 West John Carpenter Freeway, Suite 430
Irving, TX 75039 USA

***Abstract* - A Protection Valuation Tool (PVT) provides a new capability to software development project managers, by enabling determination of an optimum software protection budget. The PVT allows analysis of the effects of non-optimal funding, justifies when it is sensible to forego protection, and also facilitates analysis of desirable budget adjustments in response to software protection and attack technology developments.**

I. INTRODUCTION

Few project managers enjoy the idea of spending part of their development budget on software security. Unfortunately, the reality is that protecting software is often necessary, in order to reduce the risk of a catastrophic event, such as sabotage that can cripple infrastructure or theft of valuable intellectual property (IP). The analysis presented here tracks a method for determining legal IP protection budgets [1] and addresses the lack of software protection budgeting tools that was noted on page 39 of [2].

Determining an optimal funding level enables efficient use of resources, while simultaneously managing risks intelligently. Budgeting can now be accomplished using the proposed economic theory, which has additional uses, including explaining the consequences of over-funding or under-funding. These topics will be addressed in more detail in a future article. Additionally, the theory provides a framework for sensibly adjusting budgets in response to cost variations and developments that impact expected protection effectiveness, such as changes in software attack and protection technologies. See [2] for the author's definition of software attack and [3] for the author's definition of security value. However, perhaps one of the most important secondary uses is identifying situations in which it is sensible to forego spending on software protection.

II. PROTECTION VALUATION TOOL

The theory introduced here provides a protection valuation tool ("PVT"), illustrated as Fig. 1, which is somewhat similar in concept to the well-known supply and demand graph. In the common supply and demand graph, a market price is explained by the intersection of a supply curve with a demand curve. The PVT enables determination of an optimum protection budget using Value curves and Effectiveness curves. Similar to the supply and demand graph, the PVT provides explanations for consumer choices, and allows for predicting sensible responses to changes in available options. However, the PVT has the additional benefit of functioning as a budget-setting tool.

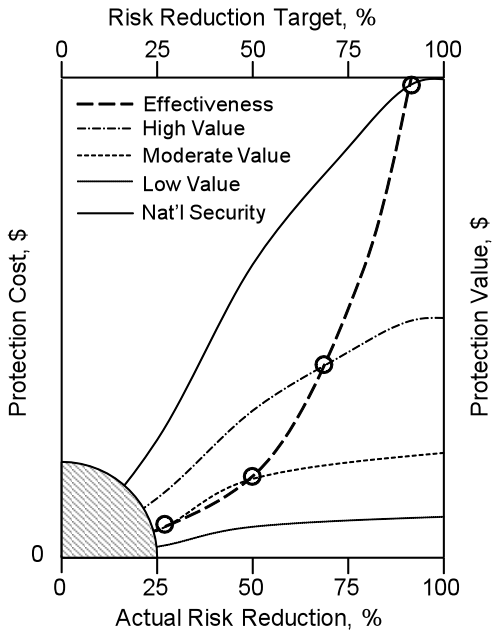

Fig. 1. PVT

Value curves are based upon the owner's perceived value of reducing risk for multiple differing target reduction levels. That is, a Value curve is a collection of hypothetical trade-offs for what a software owner would be willing to pay for each of multiple risk reduction options. In contrast, Effectiveness curves are maps of solution sets that are available in the marketplace. At the discretion of the software owner, a different PVT may be constructed at a narrow scope, one for each software package, or else an aggregate PVT may be constructed at a wide scope, to set a larger-scale budget covering multiple software titles.

A primary concept here is that an owner should identify the value of risk reductions at various target levels, as a separate process than the process of searching for available risk reduction (protection) options. Preferably, a Value curve should be created prior to searching for available solutions, in order to minimize the chance that the Value curve construction is biased by preference for a particular available option. The process of assigning value to various levels of risk reduction, independently from examining available options, is a significant aspect of using the PVT. This first part of the process produces a Value curve.

To create an Effectiveness curve, multiple protection options, available in the marketplace, are ranked by cost and evaluated for actual effectiveness. A wide range of options

should be included, ranging from inexpensive and likely ineffective, up through prohibitively expensive but comprehensive. Historical cost and effectiveness data should be used whenever available, reliable and relevant. The Effectiveness curve, resulting from this part of the process, is overlaid with the Value curve to produce Fig. 1.

If one or more intersections exist in the overlay, which are indicated with circles in Fig. 1, these intersections form a set of potential budget operating points, because the PVT has identified at least one situation in which an available risk reduction solution matches the owner's predetermined valuation criteria. If no intersection points exist, then the most sensible solution is to forego substantial expenditures for protecting the software. Fig. 1 illustrates curve sets having a single intersection point, multiple intersection points, and no intersection points.

To more efficiently describe the interpretation of these intersection points, we now take a detour from describing Fig. 1. We will examine a more detailed comparison with supply and demand graphs, as well as work through an explanation of the construction of the PVT.

The following comparison in Table I highlights similarities and differences between the well-known supply and demand curves, and the newly-introduced Value and Effectiveness curves.

TABLE I
COMPARISON OF CURVE SETS

	Supply and Demand Graph	Protection Valuation Tool ("PVT")
Curves	Supply has a positive slope, and is monotonically non-decreasing. Demand has a negative slope, and is monotonically non-increasing.	Value has a positive slope, and is monotonically non-decreasing. Effectiveness has a positive slope, and is monotonically non-decreasing.
Intersection Points	One point is certain to exist. Only one point exists in a typical market.	One trivial point will exist at zero. No non-zero points are certain to exist. Multiple non-zero points may exist.
Primary Use	To explain a market price. The intersection point is the market price.	To set an optimum budget. Each non-zero intersection point is a local optimum budgeting point.
Secondary Uses	To predict price dependence on variations in supply and demand.	To identify the impact of funding variations on risk reduction. To identify adjustments for changes in protection cost and effectiveness. To explain a sensible lack of funding.

III. CONSTRUCTING A VALUE CURVE

A Value curve traces the set of points that represents the actual economic value achieved by reducing the risk of a successful attack by various target percentages. For example, if a particular action plan will reduce the likelihood by half (50%) that any piracy, IP theft attempt, or other attack event will succeed, then completing that action plan has a particular economic value to the software owner. The owner should be able to identify the value of the 50% risk reduction target, perhaps by analyzing the expected impact on profits, brand image, and other business considerations. Assigning a value to the 50% risk reduction target creates the first of multiple points that will be used to construct a Value curve.

It is important to note that the value assigned to a risk reduction target is *not* the cost that the owner *expects to pay* to achieve that target, but *instead* what the owner *would be willing to pay*. This amount is the owner's perceived value of a protection scenario that achieved the target, based on the expected beneficial value of having reduced the number of successful attacks.

However, identifying only a single point is not enough. The owner should also identify a second action plan that will reduce the likelihood of successful attack by a different amount, perhaps by only one-fourth (25%). Since the risk reduction is lower and attacks are more likely to be successful, this risk reduction level should have lower value. This provides another point for plotting, and the Value curve begins to take shape.

One data point that merits consideration is the impossible situation of complete (100%) risk reduction and a guaranteed absence of any successful attacks at all. In practice, this cannot be achieved with certainty, although if it could, it would have a relatively high value, when compared with 25% and 50% risk reduction targets. Assigning a value to the theoretical 100% risk reduction target creates the third point defining our hypothetical Value curve.

Another important data point, and one that is easy to understand, is the zero point (0%, $0). In general, Value curves should start at (0%, $0), or maybe even with a deficit. This is because the owner will not perceive any value in a protection plan that offers no risk reduction at all. The (0%, $0) point will be discussed in more detail later. For now though, it provides the fourth point in the example Value curve.

Connecting the four points described thus far, the end points at 0% and 100%, along with the two example points 25% and 50%, creates a rudimentary Value curve, perhaps similar to the Moderate Value curve of Fig. 2. In Fig. 2, the example points are plotted as "Risk Reduction Target, %" versus "Protection Value, $" and a curve is formed by connecting these four points. Analyzing the values of other potential risk reduction targets can refine the Value curve to produce a smoother shape.

A Value curve traces the set of points that answers the following question: "How much would an owner be willing to pay in $Y, to achieve a risk reduction of X%, for a particular software package?" The owner's perceived value of achieving a specified risk reduction target should account for both the value of the software itself, and a discount for the uncertainty of the protection's effectiveness.

As can be seen in Fig. 2, the greater the value of the underlying software, the higher the Value curve will rise against the protection value axis. For example, a curve reflecting High Value indicates a greater protection value for a

given risk reduction target, whereas a curve reflecting Low Value indicates a lower protection value for a given risk reduction target. The differences can be explained with the following scenario: A developer has two different software packages that are scheduled for release. One is likely to generate a significant amount of profit, but only if competition can be limited for an extended period of time, whereas the other is either less valuable or else is intended for only a short-duration sales period. During the short-duration sales period, few events are likely to occur in time to significantly damage expected profits. As the importance of risk reduction decreases, possibly due to decreases in valuation of the underlying asset, a Value curve shifts downward.

In such a situation, when the different curves are plotted on the same graph, a Value curve constructed for more valuable software will be above a Value curve constructed for less valuable software. An example of a Low Value curve is one generated for a video game, with an expected short-duration, high-volume sales period that enables relatively quick cost-recovery and profit. By the time a competitor can bring a similar product to market, or software crackers can facilitate wide-spread piracy [2], sales will already be slowing due to market saturation. Even the value of a 100% risk reduction is limited for such an example, because sales in the consumer video game market is largely spurious, driven by emotional appeal rather than economic utility, and tied to anticipation of a release date. So sales will initially be brisk, but then could taper off. In contrast, the High Value curve could represent a business-to-business software package with a relatively stable long-term sales expectation, for which widespread piracy or competition can produce a noticeable effect on revenue.

In light of these examples, the following general comments about Value curves should be easily understood:

1. The Value curves start at $0 for 0% risk reduction. There is no value, if there is no benefit.

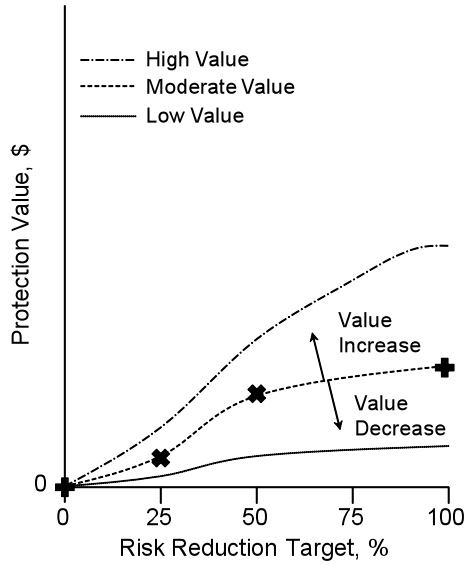

Fig. 2. Value Curves

2. The curves have a limited maximum value for the theoretical, but impossible, case of 100% risk reduction.
3. The curves likely taper off as risk reduction approaches 100%.
4. The curves are monotonically non-decreasing, although they may not be monotonically increasing.
5. Higher value, longer term reliance, and greater criticality raises the protection value for a given risk reduction target.

IV. Constructing an Effectiveness Curve

An Effectiveness curve traces the set of points that represents the actual costs that are necessary to obtain threat reductions at various levels. An Effectiveness curve is needed for each Value curve and must be at the same scope (title-specific or covering an entire product line), although a single Effectiveness curve may be copied for use with Value curves that are at the same scope.

There are two primary differences between an Effectiveness curve and a Value curve. The risk reduction parameter for an Effectiveness curve is the actual risk reduction that is achieved in practice, whereas for a Value curve, the risk reduction parameter is the target (desired) amount of risk reduction.

The actual risk reduction values can be determined empirically, using analyses of historical data for similar activities, or more expensively, by using red team testing results. For example, certain protection packages may be empirically determined to prevent attacks only by unskilled attackers, but not sophisticated ones. However, if a more expensive protection package were used, a greater number of attackers could be defeated.

Multiple levels of risk reduction activities should be used in determining cost data, ranging from very low-cost plans, up through highly expensive ones. To generate an Effectiveness curve, a software owner could contact suppliers of protection services to request a plurality of cost estimates. Similar to the process of constructing a Value curve, various scenarios should be analyzed, so that a set of points can be plotted and connected to create a curve.

Similar to the difference between an actually achieved threat reduction and a threat reduction target, there is also a difference between actual protection cost, used in an Effectiveness curve, and protection value, used in a Value curve. Protection cost can be determined by analyzing historical cost data and price quotes. Note that there is a difference between this cost and the earlier-described protection value. The cost is the price of available options, whereas the value is what the owner is willing to pay. These two numbers will generally differ.

An Effectiveness curve traces the points that answer the following question: "How much will it cost to achieve a risk reduction of X%?" An Effectiveness curve is somewhat similar to a supply curve, because it indicates what is available in the market, whereas a Value curve is more closely related to a demand curve, because it indicates how much a consumer (here, a software owner) is willing to spend. In the protection market, the software owner is the consumer.

The construction of an Effectiveness curve will be described using Figs. 3 and 4. The easiest point to explain is the zero point (0%, $0). Spending nothing results in essentially no protection. Thus, an Effectiveness curve will generally start at (0%, $0). Two representative cases are illustrated in Figs. 3 and 4: Ineffective and Highly Effective, which are plotted as "Protection Cost, $" versus "Actual Risk Reduction, %" in Fig. 3.

The Ineffective curve shows a relatively ineffective protection scenario because, as indicated in Fig. 3, the actual risk reduction achieved is fairly low, even for significant expenditures. Rotating the graph of Fig. 3 produces the graph shown in Fig. 4. The Ineffective curve of Fig. 4 has the same relationship between actual risk reduction and protection cost as the curve illustrated in Fig. 3, although Fig. 4 provides a slightly different perspective. As illustrated in Fig. 4, achieving even comparatively low risk reduction results in rapidly escalating costs.

In contrast, the Highly Effective curve represents a protection scenario in which meaningful risk reduction is achievable, although for a non-trivial cost. In Fig. 3, the Highly Effective curve indicates that greater risk reduction is available for a given cost than is provided by the Ineffective protection. It should be noted that protection quality may be related to price, and attempting to cut protection costs by selecting less-expensive solutions may damage effectiveness disproportionately to cost savings. Rotating the Highly Effective curve from Fig. 3 to produce the perspective shown in Fig. 4, illustrates that, while costs do climb with increasing risk reduction, the cost changes are more reasonable with the Highly Effective curve than with the Ineffective curve.

Fig. 4 illustrates another concept: As protections improve, an Effectiveness curve leans rightward, shifting down. This means that expenditures may be reduced, while achieving the same risk reduction. An example of improving protection effectiveness is an efficiency improvement as protection engineers gain experience. However, if new attack technology is developed and promulgated, actual risk reduction will generally decrease for a given expenditure level. In this case, an Effectiveness curve shifts upward in Fig. 4.

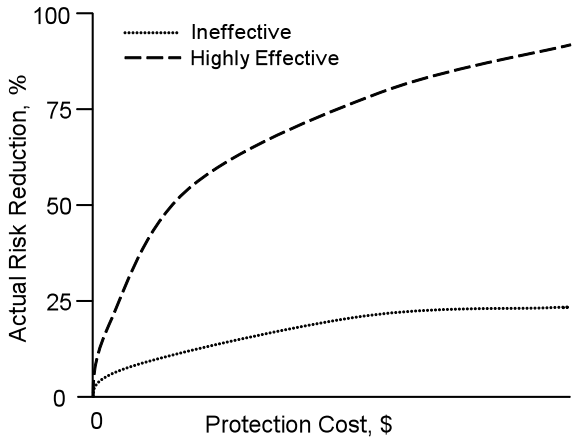

Fig. 3. Effectiveness Curves

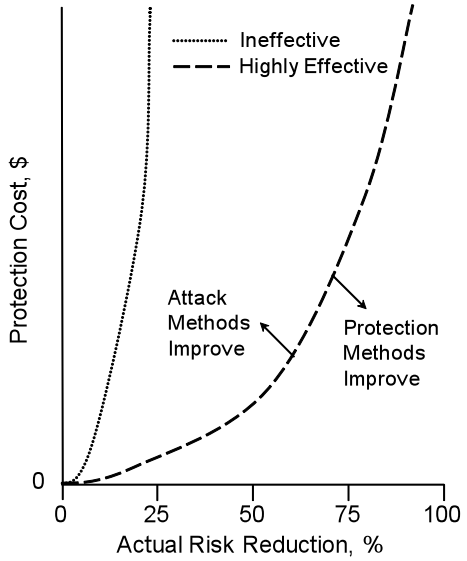

Fig. 4. Rotated Effectiveness Curves

No Effectiveness curve will reach 100% risk reduction, even with excessively high costs. In light of these examples, the following general comments about Effectiveness curves can be more easily understood:

1. The Effectiveness curves start at $0 for 0% reduction. There is no benefit, if there is no effort expended.
2. The curves never reach 100% risk reduction.
3. Protection cost rapidly escalates as risk reduction approaches 100%.
4. The curves are monotonically non-decreasing, although they may not be monotonically increasing.
5. Changes in technology affect an Effectiveness curve's shape and maximum cost endpoint.

V. Overlaying the Curves

Although the Value and Effectiveness curves are plotted on differently-defined axes, the units of the corresponding axes are the same. This allows an overlay of the curves of Figs. 2 and 4 on a single graph, as illustrated in Figs. 1 and 5.

The Value curves are plotted as protection value (units of currency) as a function of risk reduction target (units of percentage). The Effectiveness curves are plotted as protection cost (units of currency) as a function of actual risk reduction (units of percentage). Thus, both x-axes are in units of percentage, ranging from 0% to 100%, and both y-axes are in units of currency, ranging from $0 up to an undefined maximum. In Fig. 5, the three representative cases of Value curves of Fig. 2 are plotted, along with the Highly Effective curve of Fig. 4.

Although it is obscured in Fig. 5, all four curves intersect at (0%, $0). The High Value and Moderate Value curves intersect the Effectiveness curve at a significant distance (>25% risk reduction) away from (0%, $0), although the Low Value curve does not. The intersection points, first noted in Fig. 1, are annotated in Fig. 5. A special National Security curve is illustrated in Fig. 1, and indicates that a high degree of risk reduction is warranted, even with high cost.

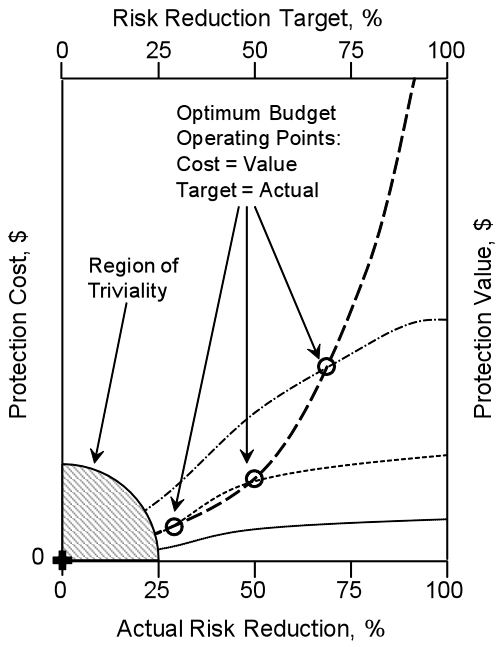

Fig. 5. Curve Overlay

Note that whereas the Value curves are bounded by the currency unit parameter at the theoretical maximum of 100% risk reduction, the Effectiveness curve is unbounded by the currency unit parameter, but cannot reach 100% risk reduction. This condition guarantees that, if a Value curve exceeds an Effectiveness curve at any place to the right of (0%, $0), then there must be at least one non-zero intersection point.

An intersection point is one at which: cost = value and actual = target. At the intersection points, an achievable cost/benefit ratio matches a predetermined cost/benefit ration that is acceptable to the owner. Therefore, an intersection point is a sensible operating point, because the cost incurred has full value when the risk reduction target is achieved. Away from an intersection point, there is a disparity between either cost and value or actual risk reduction and the target risk reduction.

Multiple non-zero intersection points may exist, and selection of the specific operating point from among the multiple intersections may involve additional considerations. If there is no non-zero intersection point, that is, if there is no intersection point other than at (0%, $0), then the optimum operating point is (0%, $0). Thus, in some circumstances, a PVT can provide reasoned justification for not spending anything on software protection.

Fig. 5 includes an obscured section nearby (0%, $0). This is a region of uncertainty, in which Effectiveness and Value curves could have a number of essentially meaningless intersection points, due to exceptionally low costs. A PVT should be used with the understanding that only intersection points at meaningful levels of protection and expenditure should be seriously considered as potential budget operating points.

The highlighted intersection points make sense heuristically, because the High Value curve intersection point indicates that greater resources should be devoted to protecting the higher value software than should be devoted to protecting the Moderate Value software.

The PVT of Figs. 1 and 5 are constructed in the following manner: Value curves are plotted using the top x-axis and the right y-axis, whereas the Effectiveness curve is plotted against the bottom x-axis and the left y-axis. The top and bottom axes should have identical range and progression, as should the left and right axes. The Value curves reflect the software owner's predetermined acceptable operating points, while the Effectiveness curve indicates operating points that are available to the owner.

VI. Lack of Market and Imperfect Operating Points

The PVT predicts the lack of a protection market for some scenarios, in which the software value is low enough that protection is not merited. This is a significant difference from supply and demand graphs, and is illustrated in Fig. 6.

In Fig. 6, the Effectiveness curve is above the Value curve for all points outside the region of uncertainty. This means that there is no protection market in this situation. The owner should forego protection for the particular software package associated with this Value curve. However, a protection market can exist if either the Effectiveness curve shifts downward or the Value curve shifts upward.

Fig. 6 illustrates a near-intersection between the Value and Effectiveness curves. If the software value increases (Value curve shifts upward) or else if the cost-effectiveness of the protections increases (Effectiveness curve shifts downward), then the curves will likely touch at this point. A market for protection will spring into existence, and the sensible protection budget will be defined by where the curves contact.

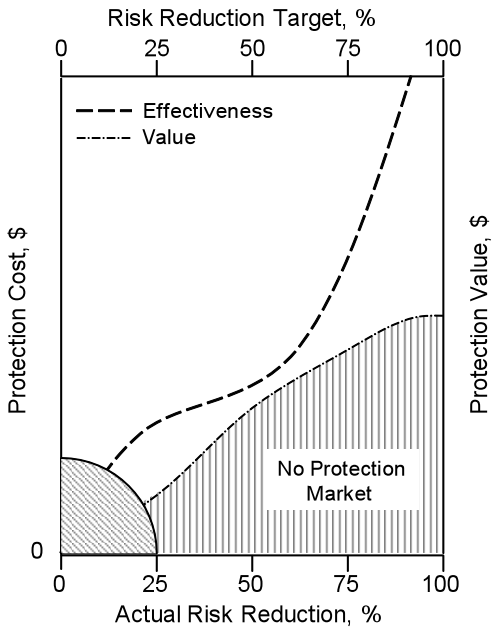

Fig. 6. Lack of Market

Software protection services providers now have an explanation for a lack of customers: When protection costs are too high, or the effectiveness is too low, savvy software owners will recognize the lack of value and opt to simply go without protections. However, if a protection service provider increases value to customers, either by reducing cost without sacrificing quality or by enhancing quality without similarly enhancing the bill, a market can be created.

Fig. 7 illustrates the unfortunate situation of over-funding software protection. In the illustration, the owner is spending for protection according to the diamond symbol on the Effectiveness curve, which is to the right and above the intersection point. The owner is receiving an actual risk reduction that is in excess of the risk reduction at the intersection point, and so is receiving some value for the overspending.

However, when compared with the Value curve, the extra level of protection is costing more than the owner's predetermined value for that achieved level of risk reduction. The predetermined value of the achieved level of risk reduction is indicated by the X symbol on the Value curve. The cost difference can be divided into two parts.

The over-funding is a combination of (1) unnecessary expense and (2) waste. As illustrated, the majority is waste, although since the curves are only notional, this result is not necessarily representative. Without having conducted a proper analysis of value and available options, a software owner risks operating in an inefficient manner, similarly one of that illustrated in Fig. 7.

Fig. 8 illustrates another unfortunate situation: under-funding software protection. In the illustrated situation, the owner is spending for protection according to the diamond symbol on the Effectiveness curve, which is to the left and below the intersection point. The owner is receiving an actual risk reduction that is lower than the risk reduction at the intersection point, and so is receiving a risk discount. However, the risk discount is less than the amount of under-funding, so the owner faces excessive risk. Without having conducted a proper analysis of value and available options, a software owner also risks operating in an inefficient manner, similarly to that illustrated in Fig. 8.

VII. SUMMARY

The introduced PVT, enables determination of an optimum software protection budget, allows analysis of the effects of incorrect funding, explains when it is sensible to forego protection, and also enables analyzing technology changes. It shows that protection is only economically sensible when a software owner's perceived value of protection matches at least one available non-zero available solution. As expected, the PVT predicts higher budgets for higher value software.

REFERENCES

[1] Wilson, Kelce and Tapia Garcia, Claudia, "How Much Should You Invest In Patents?" *les Nouvelles*, pp. 47-55, March 2010.

[2] Wilson, Kelce, "Introduction to Software Protection Concepts," *Intellectual Property Today*, pp. 36-41, August 2007.

[3] Wilson, Kelce, "Patenting Computer Security Systems," *Intellectual Property Today*, pp. 34-35, February 2008.

AUTHOR

Kelce Wilson is a patent attorney in the telecomm industry, and was previously an engineer for the US Department of Defense in the fields of software security and weapon system simulation software development. He has an MBA, a J.D. (law degree), a PhD in Computational Electromagnetics, and is a registered patent practitioner with the US Patent Office.

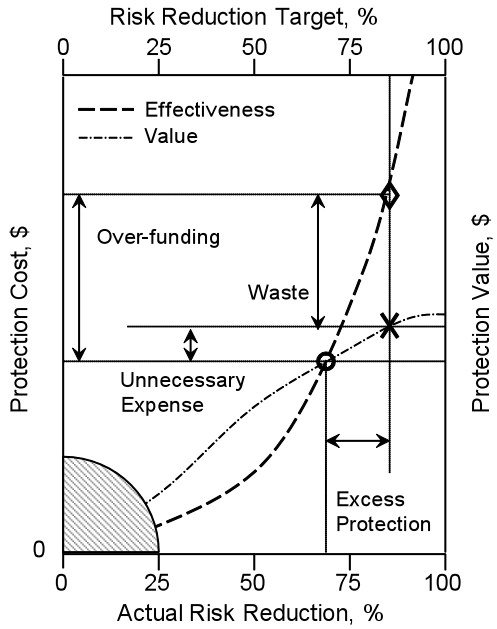

Fig. 7. Over-funding

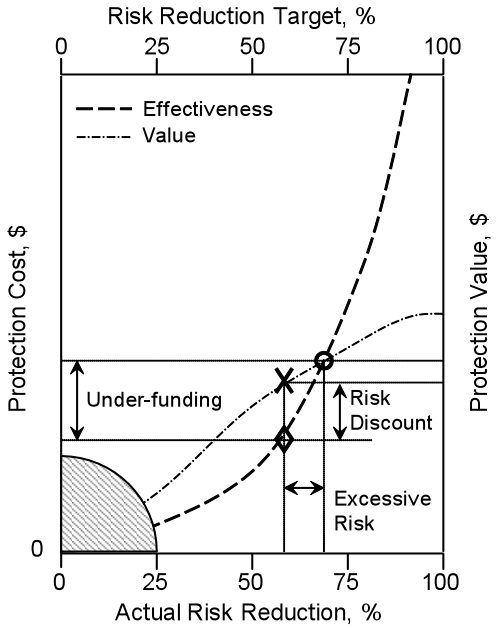

Fig. 8. Under-funding