Software Security Investment: The Right Amount of a Good Thing

Chad Heitzenrater[†‡]
chad.heitzenrater@cs.ox.ac.uk
†U.S. Air Force Research Laboratory
Information Directorate
525 Brooks Road
Rome NY 13441, USA

Andrew Simpson[‡]
andrew.simpson@cs.ox.ac.uk
‡Department of Computer Science
University of Oxford
Wolfson Building, Parks Road
Oxford OX1 3QD, UK

***Abstract*—Despite an ever-increasing amount of money and attention devoted to cybersecurity, we continue to see wide-ranging cybersecurity failures. As security practitioners examine new approaches to combat this trend, a growing community has coalesced around secure software development, or 'SWSec', as a best practice. While this movement has highlighted the role engineering process plays in combating the underlying source of vulnerabilities, it has yet to enjoy wide adoption. Anecdotal evidence points to an inability to demonstrate the return on investment (ROI) as a rationale behind this reluctance, and current information security investment models have failed to account for such expenditures. We seek to build upon such models to reflect SWSec investments, with a view to demonstrating the ROI enjoyed by SWSec practice. We summarise our current research toward these ends and identify the research required to fully reflect SWSec alongside current security investments.**

1. Introduction

As cybersecurity failures mount[1], software security ('SWSec') is increasingly seen as an alternative to the current post-deployment, enterprise security mindset. However, as SWSec continues to see devitalised adoption it has become clear that resources are playing a key role in suppressing widespread investment. Even as cybersecurity expenditure increases, companies are finding it hard to comply with the best practices for security[2]. Additionally, traditional approaches such as compliance and enterprise defence are unlikely to be devalued anytime soon (nor should they be, necessarily), further complicating matters. Consideration of SWSec places a further strain on what is often seen as a limited budget in the face of an unmanageable task. Despite a growing consensus that effective endpoint security might not even be possible[3], absent a quantified business case these practices are unlikely to receive the attention — and investment — they deserve. Ultimately, the motivation for new processes, procedures, or expanding budgets rests with such an argument[4].

A set of practices, metrics and models are required to serve as the foundation for the case for SWSec investment. Industry efforts to date, such as the Building Security In Maturity Model (BSIMM) [1] and various Security Development Lifecycles (SDLs) [2], [3] have supplied a basis of practice. Likewise, an emerging body of literature (e.g. [4], [5]) has followed the path of empirical software engineering (e.g. [6]) to produce the metrics necessary to contextualise the contribution of such activities. The final missing piece is the development of models to reason about the role of such processes, and to support the argument for their investment.

2. Problem Statement

In the wider context, envision an overall system development with well-defined boundaries. Here, we consider a system under development that is a stand-alone, greenfield development (i.e. it does not require integration into a larger system) that is centrally managed (i.e. it is not a community development) and follows a traditional development process, such as the waterfall development model. Such a narrow definition is easily broadened, but such constraints will serve the simplicity of exposition. The security investment of a software system can be characterised as a series of decisions spread over the System Development Life Cycle (SDLC) phases, starting with Inception and ending with the system's Disposal.

Consistent with other investment modelling approaches (e.g. [7], [8]), we assume a budget B that will be dedicated to security. Conceptually, B could be a fixed percentage of overall spending or a fixed amount, but importantly B is fixed, limited, and finite[5]. An allocation of B is then dedicated to security at any point in time t (denoted B_s).

[1] As evidenced by sites such as www.breachlevelindex.com.

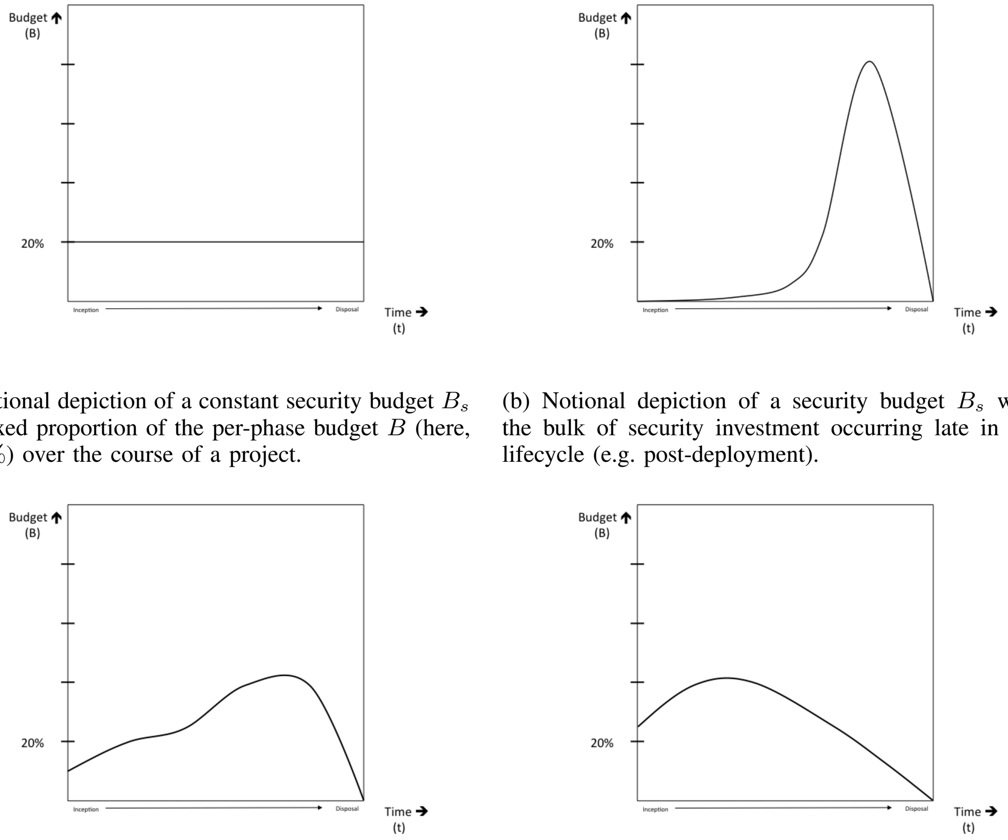
[2] For example, recent reports have cited cost as a barrier to the adoption of the NIST Cybersecurity Framework; see http://www.darkreading.com/attacks-breaches/nist-cybersecurity-framework-adoption-hampered-by-costs-survey-finds/d/d-id/1324901.

[3] http://www.fiercecio.com/story/it-losing-confidence-cybersecurity-products-frustrated-impact-productivity/2016-05-03

[4] http://www.cio.com/article/3063738/security/it-leaders-pick-productivity-over-security.html

[5] This view is separate, but not incompatible, with the important question of how to establish B relative to other constraints (as captured in approaches such as [9]). Here, we concern ourselves with the relative investment between SDLC phases once B has been established.

Approved for Public Release; Distribution Unlimited: 88ABW-2016 3044 20160616



(a) Notional depiction of a constant security budget B_s as a fixed proportion of the per-phase budget B (here, as 20%) over the course of a project.

(b) Notional depiction of a security budget B_s with the bulk of security investment occurring late in the lifecycle (e.g. post-deployment).

(c) Notional depiction of a security budget B_s with investment that starts early in the lifecycle and slowly increases.

(d) Notional depiction of a security budget B_s with investment at the early stages, leading to reduced investment in later phases.

Figure 1: Depictions of security investment over the System Development Lifecycle (SDLC).

Projected onto a Cartesian space, we find that B_s may take on a variety of forms; Figures 1a–1d present conceptualisations of how such security investment might occur.

When considering SWSec as part of the security investment, a logical question arises: Is there an allocation of project resources that provides a more efficient outcome? Figure 1b presents what might arguably be considered the current state — a very low security investment effort over the early stages of the process, quickly rising to consume an ever-increasing share of the project costs. In addition to questions of efficiency, such a profile allocates resources to security to the potential detriment of other investment opportunities (e.g. new functionality, or code maintenance) in later phases. Figure 1a, while more controlled, might not depict a more favourable allocation, as it would imply that security investment in any given phase is equally beneficial. Many might consider Figure 1c to be the most likely scenario to result in effective investment, with security investment ramping up as the security concerns — and system artefacts — become more tangible. However, the SWSec community might argue that an investment profile such as that illustrated by Figure 1d is the most appropriate. This reflects a belief that up-front investment is the most effective means to achieving security, reducing the need for later investment into activities such as accreditation, patching, and breach remediation.

The question of which depiction, if any, reflects a more efficient reality is the focus of this research. For the case of a fixed B this reflects a greater (or at least equivalent) return on investment to considerations of post-deployment defence only. Given the evidence suggesting that economic factors are a leading motivator for the implementation of new information security processes or procedures — and the low confidence practitioners have in their ability to demonstrate the ROI of such initiatives[6] — this question is important to the SWSec community. To make the case for a given outlay of B, we seek a model for information security investment that incorporates expenditure throughout the various lifecycle phases.

[6] http://www.forbes.com/sites/stevemorgan/2016/03/09/worldwide-cybersecurity-spending-increasing-to-170-billion-by-2020/#72fb63a576f8

3. Modelling SWSec

Although the information security economics literature is not lacking in investment models, to date none explicitly account for investment into SWSec. Existing modelling approaches have previously been classified into different types [10], to include accounting models (e.g. [7], [11]), game-theoretic models (e.g. [8]), and macroeconomic-focused input/output models (e.g. [12]). Even if focus is limited to accounting models alone, such 'standard' cybersecurity economic models are not easily applied to software security [13].

Alternatively, one might consider software-specific investment models, in the vein of empirical software engineering (e.g. [6]). In such models, a lack of definitive security metrics can be a hindrance, leading to the use of surrogate measures with questionable applicability [14]. The resulting investment decisions may be ill-defined, or based upon faulty data; worse, this limited view fails to consider the inevitable post-deployment security investments captured by information security models.

The focus of this research is the extension and integration of these two approaches. While software engineering's tradition of quantified data collection has begun to place attention on security vulnerabilities, yet to be addressed is how such data applies within the economic context. Examination of a few popular accounting models finds a consistent theme in the insight the defender has into the durability of the system.

- The *Gordon-Loeb Model (GL)* [7] is widely considered to be the standard for information security economic models, employing a breach probability function based on the defender's risk stance. GL uses a single vulnerability parameter v to incorporate the "probability that without additional security, a threat that is realised will result in the information set being breached" [7].
- The *Iterated Weakest Link (IWL) Model* [11] is a discrete-time model in which the adversary will always attack the least protected point at each time t, as long as it is economically viable. IWL employs an uncertainty parameter σ and an attack gradient Δx to characterise the defender insight into security and the difficulty of successful attack, respectively.

Our research seeks to combine empirical software engineering with information security economics, characterising the contribution of SWSec as the resilience of the system prior to post-deployment security. Such a definition is consistent with the goals of SWSec, which seeks to "build security in" to systems [15]. The result are models that (although at times abstract) can be used to guide investments throughout the development lifecycle. An approach will now be illustrated, using IWL as a basis.

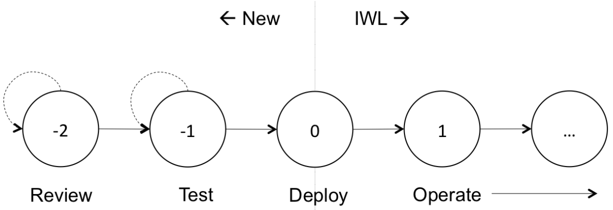


Figure 2: Depiction of the integrated SSE-IWL process, with process investment steps ($t = -2, -1$) that complement the system-level security investment ($t = 1, \ldots, t_{\text{max}}$). This is anchored by the deployment point ($t = 0$), whereby the values set by the software process are fed into the system model (in this case, the IWL).

3.1. Example: The IWL-SSE Model

We illustrate our concept of interest by summarising the IWL Secure Software Engineering (IWL-SSE) model, as presented in [16]. IWL-SSE was developed as a companion model to the IWL, such that it can be executed independently with the goal of setting the model's initial conditions. (The reader is referred to [11] for details of the IWL.) It represents aspects of the SSE process prior to the system's deployment (i.e. absent attacker input), resulting in favourable IWL initial conditions: reduction of uncertainty σ and an increased 'attack gradient' Δx (the increase in costs for each successive attack, related to the 'hardness' of the system). Investment is then optimised over $t_{max} > 0$ time periods, with the SSE occurring at time ($t = -2, -1$) and deployment at time $t = 0$.

In IWL-SSE, defender investment may occur during Architecture and Design ($t = -2$, denoted AD) and Implementation and Test ($t = -1$, denoted IT). Each phase may occur zero or more times (i), succeeding with a probability α (finding a flaw via review) or β (finding a bug or flaw via test). This results in an overall investment by phase of

$$I_{\{\text{AD,IT}\}} = (i \cdot c) + i \cdot (\text{eff} \cdot e)$$

where

$$\begin{aligned} i \in & \ \{i_{\text{AD}}, i_{\text{IT}}\} \\ c \in & \ \{c_{\text{AD}}, c_{\text{IT}}\} \\ \text{eff} \in & \ \{\alpha, \beta\} \end{aligned} \qquad (1)$$

Here, I represents the cost of the SWSec activity conducted at time t for a given set of iterations, i. The value eff is the effectiveness of the SSE activity (α for reviews and β for tests), and refers generally to the benefit derived from execution of the process. Note that this is related, but not equivalent, to the effectiveness of a particular tool or activity (e.g. static analysis software or formal specification reviews). Finally, e is the cost of conducting the identified fixes; in the initial model, it is assumed that all identified flaws or bugs are fixed. Applied to the two defined phases, this results in the following functional forms:

$$\begin{aligned} t = -2: \quad I_{\text{AD}} &= (i_{\text{AD}} \cdot c_{\text{AD}}) + i_{\text{AD}} \cdot (\alpha \cdot e_{\text{fAD}}) \\ t = -1: \quad I_{\text{IT}} &= (i_{\text{IT}} \cdot c_{\text{IT}}) + i_{\text{IT}} \cdot \left(\beta \cdot \left[\frac{e_{\text{fIT}}}{2^{\alpha i_{\text{AD}}}} + e_{\text{bIT}}\right]\right) \end{aligned} \qquad (2)$$

Here, c_{AD} and c_{IT} represent the cost of conducting a review or test, respectively.

The overall cost for the software process I_{P} is then simply the summation of the phase costs:

$$I_{\text{P}} = I_{\text{AD}} + I_{\text{IT}} \tag{3}$$

The defender benefit derived from SWSec investment is reflected in two ways. The first benefit is in the reduction in the overall uncertainty σ faced by the defender, relative to the overall amount invested in the respective phases. In IWL-SSE an equal amount of uncertainty — and equal weight — is placed on each phase:

$$\sigma = \sigma_{\text{AD}} + \sigma_{\text{IT}} \tag{4}$$

This formulation reflects the number of iterations undertaken, exponentially decreasing asymptotically to 0:

$$\sigma_t = \frac{\sigma_{max}}{2} \cdot \text{eff}^{\frac{1}{i}} \tag{5}$$

with

$$\begin{aligned} \text{eff} \in & \ \{\alpha, \beta\} \\ i \in & \ \{i_{\text{AD}}, i_{\text{IT}}\} \\ t \in & \ \{\text{AD}, \text{IT}\} \end{aligned} \tag{6}$$

Here, σ_{max} refers to the starting level of uncertainty, while eff and i correspond to the effectiveness and iteration count values for phases $t = (-2, -1)$.

An additional payoff is an increase in the gradient of attack, Δx. This is modelled with sub-linear growth and concavity, reflecting diminishing returns (pursuant to the information security economics literature [17]):

$$\Delta x = \sqrt{(1 + \alpha i_{\text{AD}} + \beta i_{\text{IT}})} \tag{7}$$

The overall model is then optimised over ($t = -2, -1, 0, \ \ldots, \ t_{\text{max}}$) using the widely accepted Return on Security Investment (ROSI) metric, which, in turn, is defined by the Annual Loss Expectancy (ALE) [18]:

$$\text{ROSI} = \frac{\text{ALE}_0 - \text{ALE}_1 - \text{average security investment}}{\text{average security investment}} \tag{8}$$

where [19]:

$$\text{ALE} = \text{Expected rate of loss} \times \text{Value of loss} \tag{9}$$

This results in a new metric: the *Return on Secure Software Process (ROSSP)*, which is defined thus.

$$\text{ROSSP} = \text{ROSI}_{\text{SSE}} - \text{ROSI}_{\text{NoSSE}} \tag{10}$$

Here, ROSI_{SSE} is the realised return after SSE investment, while $\text{ROSI}_{\text{NoSSE}}$ is the return without SSE investment.

In [16], it was demonstrated that an increased ROSSP could be derived with a lower initial post-deployment defence outlay. Values for the IWL parameters asset value ($a = 1000$), attacker return ($r = 0.05$), and attack costs ($z = 0.025$) were established consistent with [11]. Values for α and β were set relative to reported values for manual architectural review ($\alpha = 0.6$) and code review ($\beta = 0.3$) effectiveness [20]; although not directly analogous, these represent reasonable, well-supported estimates.

Reviews	**Tests**	**ROSI**	**ROSSP** **(IWL, $\sigma = 0$)**	**k**
None, $\sigma = 0$		33.5	–	11
None, $\sigma = 16$		25.1	-8.4	0
1	1	34.5	1.0	5
5	1	40.6	7.1	4
1	5	39.4	5.9	4
8	24	44.6	11.1	3
25	25	42.6	9.1	3

TABLE 1: ROSI and ROSSP for various configurations of secure software engineering ($\alpha = 60\%$, $\beta = 30\%$, $c_{\text{AD}} = 3$, $c_{\text{IT}} = 1$)

Starting with the best-case scenario of low uncertainty ($\sigma = 16$) and a requisite attack gradient ($\Delta x = 1$), various combinations of reviews and tests were examined up to a maximum of $i_{\text{AD}}, i_{\text{IT}} = 25$. The model was then executed in discrete time against multiple rounds of post-deployment security, with the results presented in Figure 3 and Table 1. It was found that in most instances some amount of SWSec supports improved overall ROI (combined pre- and post-deployment outlays) for the given IWL parameters. Importantly, larger values of i_{AD} and i_{IT} result in a lower ROSSP, highlighting limits to the value of increased process. This reflects the widely held belief that SWSec investment has the potential to provide a greater return than purely post-deployment security — but only when applied at the 'right' amount.

3.2. Model Extensions

As an initial model of SWSec investment, IWL-SSE raises many questions regarding the effective allocation of resources and the role SWSec plays relative to post-deployment security. Ongoing work builds upon this model by relaxing the model's assumptions in order to present a more comprehensive view of how investments in secure software development affect efficient overall system security.

Standalone system. Most systems — and especially those likely to come into contact with an adversary — are developed to be inter-connected, and deployed as a part of larger networks. While IWL-SSE examines the role of SWSec with respect to the network security of a single system, in reality most networks comprise enterprises where the security of each individual system may affect connected systems. The original IWL model accounts for this interconnectivity by representing a configuration of defences $\mathbf{d}_t$ as a binary column vector of elements d_i, where $\mathbf{d} \in \{1, 0\}^n$ indicate the implementation (1) or not (0) of any specific defence. Costs for a given configuration for each post-deployment round are then combined into a matrix $\mathbf{C}$, where off-diagonal elements assume positive or negative values indicating their positive or negative effect in the given configuration. Additionally, IWL supports the representation of sunk costs ($\lambda \geq 0$) that result from altering defensive posture, supplying a richer representation of the complexities of post-deployment defence configuration [21].

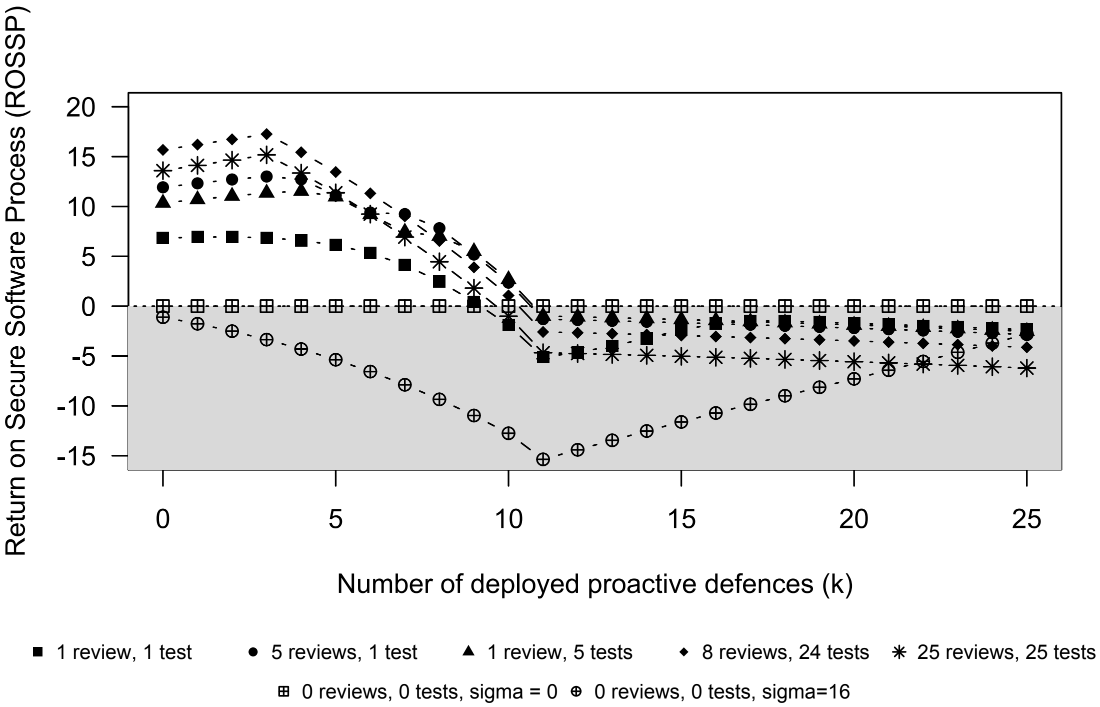


Figure 3: The Return on Secure Software Process (ROSSP) for a variety of dynamic IWL-SSE scenarios. The unfilled boxes represent the case of no process: with high uncertainty (crossed circles), and with absolute certainty (crossed squares). The latter serves as the 'best case' scenario without SSE investment, and serves as the baseline for comparison.

Currently, IWL-SSE represents interdependent post-deployment defences in IWL, but does not account for sunk costs or potential interplay between SWSec and the security enterprise. Consideration of such costs in light of SWSec will require a quantified understanding of how (if at all) secure software engineering investments affect specific post-deployment defences, and the incorporation of these parameters into IWL-SSE is likely to increase return — especially in the case of sunk costs.

Greenfield development. As a consequence of a number of factors — including advances in the design of programming languages, the rise of commercial software development, and active open-source software communities — very few modern software-based systems emerge as a result of software being developed from scratch (so-called 'greenfield' development). Instead, most software developments employ any number of software libraries and interface with various existing software applications (at the very least, an operating system). As a result, considering the security of developed software requires a consideration of other systems, including the software it, in turn, relies upon.

Dependencies must be considered relative to the costs they impose. Information security has long been recognised as an economic externality — resulting in increased indirect costs that are borne by others [22] — and the state of software development is representative of this effect. This construct has often been used to explain the common pattern of security flaws commonly found in commercial software; so-called "patch-and-fix" [23]. This effect is non-negligible, with high-profile vulnerabilities such as Heartbleed[7] being the direct result of utilising external codebases. Extension of IWL-SSE to consider the effect of externalities introduced as a result of operating system dependence and the employment of third-party libraries will provide a more nuanced view of the true cost of modern software development.

Engineering approach. Additionally, IWL-SSE assumes the view of software development as a linear, centrally managed endeavour. While approaches such as the waterfall model and its variants continue to dominate development activities (with, by some estimates, upwards of 30% of projects being developed via such approaches [24], [25]), there is an increasing rise in alternative approaches such as agile methodologies. The specific form of the model follows from this assumption of linearity, but the construction and configuration of the SSE portion of the model (and its parameters) could easily be adapted to represent other approaches. Such extensions include the incorporation of additional types of errors (such as configuration errors), and security-specific testing later in the lifecycle (e.g. penetra-

[7] http://heartbleed.com/

tion testing — as investigated in an extension to the IWL, the IWL-PT model [26]).

4. Application

What is to be gained by employing such models? Ideally, their output provides valuable insight into the investment decision-making process, forming the basis for rational security investment. Yet, as evidenced in Section 3, the treatment is necessarily general, and the operation is stylised: as with all models, they are a simplification of complex issues into essential elements sufficient to gain an understanding of core principles. Three such principles are now presented.

Effects of SWSec. Realising the value contribution of SWSec practice is both the primary motivation and key insight to be gained from model development. Providing a reasoned approach to examining SWSec investment addresses one of the largest obstacles to SWSec adoption: understanding the return on investment. While a critical observer may challenge the validity of a given calculation, approaches such as IWL-SSE are derived in much the same way as the risk management processes corporations have grown to rely upon. The linking of empirical software engineering and security practice hold promise to provide results that, although estimates, are as supportable, grounded and measured as any in common use today. Measures such as ROSSP and similar constructs support comparisons between pre- and post-deployment investment considerations, and move the discussion of such investments into a common language and representation.

Role of pre- and post-deployment security. Security investment has traditionally been a reactionary practice — a race to fix largely preventable problems via add-on solutions [27]. This emphasis on post-deployment strategies has, until now, limited optimisation strategies to single- or dual-phase consideration. Even as the ROSI provided by SWSec investment is rendered calculable, there will continue to be a need for current practices: network security, enterprise integration, third-party appliances, and certification or accreditation schemes to pull these elements together. An understanding of comprehensive security must start with a coherent consideration of the various investments from a holistic, lifecycle-spanning point of view.

Security impact of process, models, and language. As the constructs develop and spur well-designed software engineering studies into SWSec practices, such models may be key to a richer understanding of the role that software engineering decisions play in the realisation of secure systems. For example, it may be possible to more precisely model the per-phase costs (Equation 2) by examining known, estimated or relative values corresponding to specific development practices such as prototyping or iteration (impacting e_{fAD} and e_{fIT}), development languages (affecting β, e_{fIT} and e_{bIT}), tool support (reducing c_{AD} and c_{IT}), and available expertise (driving i, c_{AD} and c_{IT}, as well as α and β). Additionally, models such as IWL-SSE would benefit from richer expressions that capture varied aspects of the engineering practice, such as bug re-introduction.

Well-designed functional forms, rooted in empirical software engineering measurements, may be employed to examine vexing questions surrounding software engineering decisions. Moving such concerns from the realm of 'best practice' to reasoned choices that are aligned with system goals may not only improve system security, but also software engineering practice in general.

Further development of SWSec investment modelling is likely to foster further insights not currently envisioned, as such an act is often a catalyst for refutation and study. Security investment economic modelling continues to evolve, and insights will feed further refinement and reduce abstraction. Understanding the promise — and limitation — of these approaches will be key to their successful adoption.

5. Proposals

The IWL-SSE is but a first step towards a robust treatment of the SWSec investment problem. The following key areas of research will need to be tackled if our ultimate ambition is to be fully realised.

Data. Foremost, there is a need for stronger ties to the empirical software engineering community and the development of a culture of quantitative research in this area. Efforts undertaken thus far show promise, but often suffer from limited sampling or a lack of generalisability [28]. The values employed for effectiveness parameters, such as α and β from above, must be supported by well-scoped definitions and studies that incorporate variables pertaining to tools, techniques and expertise. Our research seeks to provide the exposure of ideas required for meaningful dialogue to commence.

Models. As data becomes more robust and expansive, models and functional forms for the various processes must be designed to support existing (or potentially novel) optimisation approaches. The understanding of SWSec practice effectiveness, and its relationship to resource investment (Equation 1) and benefit (Equations 6 and 7), will drive the accuracy of such characterisations. Best practices, such as multi-model execution [29], will enhance the robustness of calculations and foster confidence in results.

Practice. Finally, these approaches must be evangelised, and find their way into practice. While SWSec benefits from an active community of practice, many of its main pillars, such as the Building Security In Measurement Model (BSIMM) [1] and various security development models, lack the theoretical basis to absorb and develop new concepts. One avenue could be the addition of 'conceptual practices' to BSIMM: practices that may not be widely deployed due to complexity or novelty (and therefore won't appear in the BSIMM survey), but are judged by practitioners to be worthy of consideration for their potential contribution. This is a role well-suited to an entity such as the IEEE Cybersecurity Initiative (CYBSI)[8], the aim of which is the improvement of the understanding of cybersecurity issues.

[8] http://cybersecurity.ieee.org/

6. Conclusion

The three proposals of Section 5 are modest, but belie the complexity of understanding lifecycle-wide security planning and investment. While many have drawn parallels between aspects of software engineering and security — relating security to quality, usability, or robustness — the linkage of economics and engineering addresses a root cause of security failure: an inability to manage the activities necessary to impart security from the outset. The current environment demands best practice to be supported by well-grounded theory and rationale in order to support the business case required to introduce SWSec — at the right amount — into common practice.

Acknowledgments

This work builds upon [16], developed in conjunction with Rainer Böhme at the Security and Privacy Lab, Universität Innsbruck. The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] G. McGraw, S. Migues, and J. West, "Building security in maturity model (BSIMM)," PDF, 2015, https://www.bsimm.com/.

[2] "Microsoft security development lifecycle," 2014. [Online]. Available: http://www.microsoft.com/security/sdl/default.aspx

[3] "Category:OWASP CLASP project - OWASP," January 2014. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_CLASP_Project

[4] D. Baca, B. Carlsson, and L. Lundberg, "Evaluating the cost reduction of static code analysis for software security," in *Proceedings of the Third ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS '08)*. ACM, 2008, pp. 79–88.

[5] S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller, "Predicting vulnerable software components," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007)*. ACM, 2007, pp. 529–540.

[6] B. W. Boehm, *Software Engineering Economics*, ser. Prentice-Hall Advances in Computing Science and Technology Series. Englewood Cliffs, N.J: Prentice Hall, 1981.

[7] L. A. Gordon and M. P. Loeb, "The economics of information security investment," *ACM Transactions of Information Systems Security*, vol. 5, no. 4, pp. 438–457, November 2002.

[8] E. Panaousis, A. Fielder, P. Malacaria, C. Hankin, and F. Smeraldi, "Cybersecurity games and investments: A decision support approach," in *Decision and Game Theory for Security*, ser. Lecture Notes in Computer Science (LNCS). Springer International Publishing, 2014, vol. 8840, pp. 266–286.

[9] J. Peeters and P. Dyson, "Cost-effective security," *IEEE Security & Privacy Magazine*, vol. 5, no. 3, pp. 85–87, 2007.

[10] R. Rue, S. L. Pfleeger, and D. Ortiz, "A framework for classifying and comparing models of cyber security investment to support policy and decision-making," in *Proceedings of the 6th Annual Workshop on the Economics of Information Security (WEIS 2007)*, 2007.

[11] R. Böhme and T. Moore, "The iterated weakest link: A model of adaptive security investment," in *Proceedings of the 8th Annual Workshop on the Economics of Information Security (WEIS 2009)*, 2009.

[12] E. Andrijcic and B. Horowitz, "A macro-economic framework for evaluation of cyber security risks related to protection of intellectual property," *Risk Analysis*, vol. 26, no. 4, pp. 907–923, 2006.

[13] S. Neuhaus and B. Plattner, "Software security economics: Theory, in practice," in *11th Annual Workshop on the Economics of Information Security (WEIS 2012)*, 2012.

[14] M. G. Jaatun, "Hunting for aardvarks: Can software security be measured?" in *IFIP International Cross-Domain Conference and Workshop (CD-ARES)*, ser. Lecture Notes in Computer Science, G. Quirchmayr, J. Basl, I. You, L. Xu, and E. Weippl, Eds., vol. 7465. Springer, 2012, pp. 85–92.

[15] G. McGraw, "Software security," *IEEE Security & Privacy*, vol. 2, no. 2, pp. 80–83, March 2004.

[16] C. Heitzenrater, R. Böhme, and A. C. Simpson, "The days before zero day: Investment models for secure software engineering," in *Proceedings of the 15th Workshop on the Economics of Information Security (WEIS 2016)*, June 2016.

[17] S. E. Schechter, "Toward econometric models of the security risk from remote attack." *IEEE Security & Privacy*, vol. 3, no. 1, pp. 40–44, 2005.

[18] R. Böhme and T. Nowey, "Economic security metrics," in *Dependability Metrics*, ser. Lecture Notes in Computer Science (LNCS). Springer Berlin Heidelberg, 2008, vol. 4909, pp. 176–187.

[19] T. Tsiakis and G. Stephanides, "The economic approach of information security," *Computers & Security*, vol. 24, no. 2, pp. 105–108, 2005.

[20] B. Boehm and V. R. Basili, "Software defect reduction top 10 list," *IEEE Computer*, vol. 34, no. 1, pp. 135–137, January 2001.

[21] R. Böhme and T. Moore, "The ?iterated weakest link? model of adaptive security investment," *Journal of Information Security*, vol. 7, no. 02, p. 81, 2016.

[22] C. Herley, "So long, and no thanks for the externalities: The rational rejection of security advice by users," in *Proceedings of the 2009 Workshop on New Security Paradigms Workshop (NSPW '09)*. New York, NY, USA: ACM, 2009, pp. 133–144.

[23] T. Moore and R. Anderson, "Economics and internet security: A survey of recent analytical, empirical and behavioral research," PDF, Computer Science Group, Harvard University, Tech. Rep. TR-03-11, 2011.

[24] C. J. Neill and P. A. Laplante, "Requirements engineering: The state of the practice," *IEEE Software*, vol. 20, no. 6, pp. 40–45, Nov 2003.

[25] ——, "Requirements engineering: The state of the practice revisited," PDF, 2008. [Online]. Available: https://www.projectsmart.co.uk/white-papers/requirements-engineering-the-state-of-the-practice-revisited.pdf

[26] R. Böhme and M. Félegyházi, "Optimal information security investment with penetration testing," in *Decision and Game Theory for Security*, ser. Lecture Notes in Computer Science (LNCS). Springer Berlin Heidelberg, 2010, vol. 6442, pp. 21–37.

[27] T. Armerding, "Is security really stuck in the dark ages?" May 2015, [Online; posted 22-May-2015]. [Online]. Available: http://www.csoonline.com/article/2925351/data-protection/is-security-really-stuck-in-the-dark-ages.html#social

[28] R. Scandariato, J. Walden, and W. Joosen, "Static analysis versus penetration testing: A controlled experiment." in *24th IEEE International Symposium on Software Reliability Engineering (ISSRE 2013)*. IEEE Computer Society, 2013, pp. 451–460.

[29] S. L. Pfleeger and R. Rue, "Cybersecurity economic issues: Clearing the path to good practice," *IEEE Software*, vol. 25, no. 1, pp. 35–42, January 2008.