

SBQ

SECURE BUSINESS QUARTERLY



Return on
Security Investment
Calculating the Security Investment Equation

Tangible ROI through Secure Software Engineering

by Kevin Soo Hoo, Andrew W. Sudbury and Andrew R. Jaquith

Tangible ROI through Secure Software Engineering

Research conducted by Kevin Soo Hoo,
Andrew W. Sudbury, and Andrew R. Jaquith

Application security is usually addressed as vulnerabilities are discovered, after an application has been developed. In contrast, generally accepted software-engineering principles hold that software flaws are less expensive to fix earlier in the application-development process.

Do the cost savings justify an early investment in secure software engineering (SSE)?

To answer this question, *SBQ* asked researchers from @stake (the publisher of *SBQ*) to examine the value of incorporating expert security analysis at various stages in the application-development cycle.

Findings indicate that significant cost savings and other advantages are achieved when security analysis and secure engineering practices are introduced early in the development cycle. The return on investment ranges from 12 percent to 21 percent, with the highest rate of return occurring when analysis is performed during application design.

These results imply a tangible bottom-line increase based on cost reduction. “Soft” indirect costs, such as a decrease in market value after a vulnerability, have purposely been excluded from this research.

Methodology

Using a combination of public and proprietary research data on the application-development process, we have built a quantitative, time-phased model to calculate return on investment of secure software engineering practices when applied during different phases of development.

Key model assumptions included:

Security Analysis Metrics

- SSE analysis expense, per phase
- Number of security defects found, per phase
- Percentage of vulnerabilities fixed

Financial Assumptions

- Discount rate

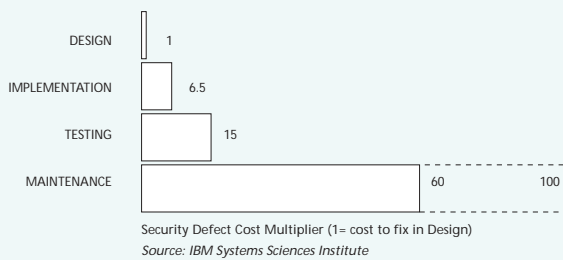
Defect Remediation Costs

- Cost to fix defects, per class of defect
- Patch release frequency
- Cost multiplier, per phase

Security Analysis Assumptions

The model assumes that SSE analysis of an average enterprise application will typically uncover approximately seven significant security defects that could have been detected — and corrected — during design. Of these, it was assumed that due to the prohibitive expense associated with remedying certain deep structural flaws, software developers will elect to fix only four. Both of these assumptions are drawn from proprietary @stake

Remediation Cost Multiplier by Phase



application-vulnerability research, which is in turn based on an exhaustive analysis of over 500 real-world data points from public and private sources.

Security Defect Remediation Costs

Based on anecdotal and empirical evidence from leading software companies and software quality assurance (SQA) experts, security defect fixes, when done during the testing phase, cost anywhere from \$2,000 to \$9,000 each, depending on complexity. To avoid biasing the net present value (NPV) calculations, we did not factor in the costs associated with severe defects that would have required a complete system redesign.

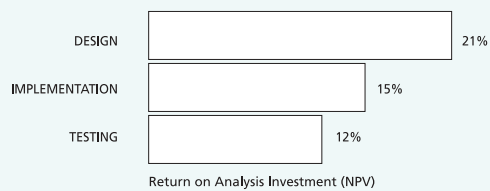
Cost Multiplier

To measure the relative costs of fixing security defects at different points in the application-development lifecycle, we created cost ratios for design, implementation, testing, and maintenance phases. According to SQA empirical research, one dollar required to resolve an issue during the design phase grows into 60 to 100 dollars to resolve the same issue after the application has shipped.

Discount Rate and Patch Release Frequency

To determine ROI based on NPV dollars, the model assumes a discount rate of 10 percent *per annum*, a twenty-month software development cycle, and periodic patch releases three months apart.

ROI by Phase of Introduction of SSE Principles



Analysis cost calculations include only direct costs associated with the programming effort needed to resolve security defects. Indirect costs such as loss of goodwill, reputation, and functionality are difficult to quantify and would have biased the results.

Observations and Results

We found that the return on investment of SSE analysis was 21 percent when initiated early in the design process. The later that security was addressed in the development cycle, the costlier it became.

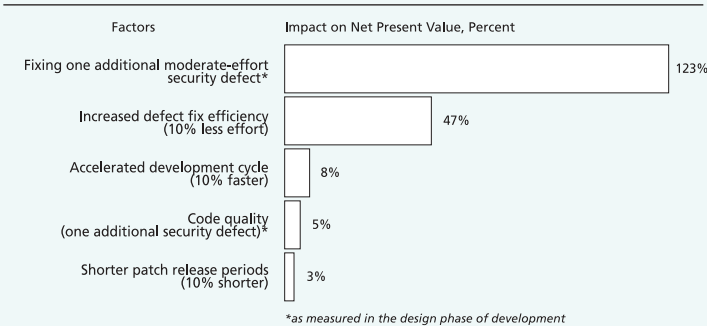
Driving results at later stages in the development process include the number of security defects found and resolved, the cost of resolving issues, and the relative increase in those costs.

For example, the cost of fixing four security defects found in a typical enterprise-class application totaled \$24,000 during the testing stage. If the defects had not been discovered until after deployment, the cost could have soared to nearly \$160,000, exclusive of indirect costs such as loss of goodwill or trust, or public relations expenses.

We performed nominal sensitivity analysis on five key inputs to understand the relative importance of each factor to the final results. The results are *most* sensitive to those factors directly related to fixing security defects, namely the number resolved and effort required. If a firm elects to fix one additional defect, NPV increases by 123 percent. Interestingly, the act of *detecting* security defects

Tangible ROI through Secure Software Engineering (continued)

Sensitivity Analysis




does not affect NPV substantially; *finding* one additional defect impacts NPV much less than fixing one. Moreover, factors such as the timing of patch releases and length of the development cycle were found to have minimal effect due to the relatively short time horizon.

As a final note, we found that NPV increased substantially when fixes could be performed more efficiently. Although it is beyond the scope of this article, we expect that developer efficiency can be significantly enhanced by such activities as developer training on secure coding practices, automated security testing tools, and knowledge-transfer activities.

Conclusion

For the typical enterprise application, the benefits associated with direct development cost savings and early detection of security issues are conclusive and outweigh the initial investment in security analysis. Since nearly three-quarters of security-related defects are design issues that could be resolved inexpensively during the early stages, a significant opportunity for cost savings exists when secure software engineering principles are applied during design. The standard practice today of

waiting until the end of the development cycle to deal with security is wasteful. Building security into applications from the start improves reliability, avoids potentially embarrassing and costly incidents, and — as our analysis shows — ultimately saves money. 

Kevin Soo Hoo has a PhD in Engineering-Economic Systems at Stanford University. His research is focused on computer security risk management.

Andrew W. Sudbury is currently an MBA student at the MIT Sloan School of Management. A founder of Café Liberty, he has worked in technology development and information security for the last 10 years.

Andrew R. Jaquith is a Director of @stake. He has over 10 years' experience as a consultant and technology implementer and has served leading firms in the financial services, supply-chain, and manufacturing sectors.

Secure Business Quarterly



ILLUSTRATION: JAMES STEINBERG

Defining the Value of Strategic Security

SBQ

SECURE BUSINESS QUARTERLY

Secure Business Quarterly • 196 Broadway • Cambridge, MA 02139 USA
www.s bq.com • e-mail: info@s bq.com

Secure Business QuarterlySM is an @stakeSM publication.