



---

## Cigital SDL Case Study Outline – Insurance Vertical

---

### **Executive Summary**

This case study follows the adoption of a secure software development lifecycle based on the Microsoft SDL at a large insurance company (the Company). The case study describes both the business drivers leading up to the Company's recognizing the need for incorporating the SDL within their development process as well as the initial roll out of the SDL.

A combination of changes in the regulatory landscape and results of third-party security assessments of some applications led the Company to evaluating better methods for managing their software security risk. The Company had been using an approach that relied heavily on penetration testing, but the third-party assessments showed that the Company's applications still contained high risk security vulnerabilities. The company used a combination of Microsoft consultants and Cigital, a Microsoft SDL Pro Network member, to help them optimize their risk assessment and mitigation process by integrating security activities into the Company's software development lifecycle.

Cigital is a leading software security and quality consulting firm focusing on helping organizations improve software. We help organizations ensure their software is secure and reliable while also improving how they build and deploy software using a combination of proven methodologies, tools, and best practices that are tuned to meet each client's unique requirements. Cigital has enabled some of the most well-known organizations in financial services, communications, insurance, hospitality, e-commerce and government to reduce their mission-critical software business risks.

### **Background**

Before implementation of the SDL, the Company's secure development practices would best be described as "below Basic". The Information Security group was primarily concerned with network and infrastructure security. The Information Security group had security policies concerning application security, but was unable to get wide-spread adoption or adherence to these policies by the Company's software development teams. The Company did have a practice of regularly-scheduled penetration testing that combined both network and application testing.

The Audit group felt that the Company were spending too much money on penetration testing and asked the CISO to find a more effective process at managing risk in their application software. Along with the Risk and Compliance group, the Information Security group sponsored a set of third-party assessments for the Company's consumer-facing applications. These assessments found several vulnerabilities that the penetration testing had



missed. One of these vulnerabilities was serious enough that the Company was at risk from losing a license it needed to conduct daily business. Because they clearly proved the inefficacy of the process followed so far, these assessments were used to build the business case for a more comprehensive (SDL Optimization Level 3) enterprise-wide secure software development life cycle program.

## **Using the SDL Optimization Model for an Enterprise-wide Solution**

Executive management, led by the CISO, decided that an enterprise-wide SDL-based solution was the most cost effective and approved a project to bring the Company to SDL Optimization Level 3. The “SDL Program” was created as a three-year program run out of the Information Security office. The SDL Program was split into three phases of which the first phase is complete. The first phase was to define the roadmap for adopting each of the SDL’s five capability areas and pilot the SDL within two application development teams. The goal of the first phase was to get the two development teams to the SDL Optimization Level 2 standard. The second phase was to try and scale the program across an entire business unit using the lessons learned from the pilots. By the end of the second phase, the goal was to reach SDL Optimization Level 3 for a single business unit. The third phase consisted of rolling out the SDL across all of the Company’s business units.

The Company’s healthcare business unit was chosen to pilot the SDL for several reasons:

- The business was developing a direct-to-consumer sales channel through the web and a customer portal. These applications were rated as high-risk by the Compliance and Legal group because they managed personally identifiable information (PII) and were accessible via the Internet by anyone.
- The applications in the healthcare business unit were being modified to address changes in the regulatory environment to avoid financial penalties and protect the Company’s employees from prosecution for non-compliance. For example, some of the required changes were:
  - Removing requirements to gather PII such as a US Social Security Number to meet more stringent requirements prescribing reduced collection of information.
  - Supporting data access use and disclosure regulations such as the:
    - Health Insurance Portability and Accountability Act (HIPAA)
    - European Commission’s Directive on Data Protection
    - European Standards on Confidentiality and Privacy in Healthcare
- Many of the security vulnerabilities found during the assessments that could have led to disclosures of PII were found in applications run within the healthcare business unit.



## Implementing Security Activities within Development Phases

The SDL Program started with a combination of internal groups and external consultants developing changes to the Company’s software development process, internal security standards and policy, internal training, and tools. The project team decided that the Company would base the overall secure development life cycle on Microsoft’s SDL for two reasons:

1. It was flexible enough to accommodate both their .NET and Java application teams.
2. Microsoft SDL was comprised of activities that are designed to integrate with their current RUP-based software development and overall project management processes.

The program team decided on a set of core security activities to target integration within the software development life cycle. These security activities were taken from the SDL Capability areas and then mapped to the Company’s development lifecycle phases. Table 1 shows the security activities defined by the SDL Program team for the Company’s security program.

Table 1 - Security Activities

Development Phase	SDL Capability	Security Activity
On-boarding and Training	Training, Policy and Organization	<ul style="list-style-type: none"> <li>• SDL Awareness Campaign</li> <li>• Role-specific SDL training</li> </ul>
Functional and Non-functional technical requirements	Requirements and Design	<ul style="list-style-type: none"> <li>• Security Requirements</li> <li>• Abuse Case Definition</li> </ul>
Architecture and Design	Requirements and Design	<ul style="list-style-type: none"> <li>• Threat Modeling</li> <li>• Attack Surface Analysis and Reduction</li> <li>• Secure Design Review</li> </ul>
Development	Implementation	<ul style="list-style-type: none"> <li>• Secure Code Guidelines</li> <li>• Secure Code Reference Card</li> <li>• Secure Code Review</li> </ul>
System Integration Testing	Verification	<ul style="list-style-type: none"> <li>• Fuzz Testing</li> <li>• Manual Security Testing</li> </ul>
Deployment	Release and Response	<ul style="list-style-type: none"> <li>• Security Scorecard</li> </ul>



## Step 1 – Awareness Training

Rather than start implementing the SDL directly within the application development teams, the Company first rolled out awareness training campaign. The awareness training campaign was designed to convey the motivations and justification for the SDL. A critical piece of the awareness training campaign was the development of pre and post training assessments that bookend the awareness training delivery. Through the assessments, the SDL Program team was able to track both the effectiveness of the training campaign and incorporate participant’s suggestions to improve it. The initial offering was management briefings that started with the business unit heads across the Company and then down to their direct reports. The business-unit head awareness training was targeted at the senior management to ensure that they understood the size and nature of the business risk and that the Company’s executive management was on board and supported the program. Ensuring that the business unit heads supported the SDL Program was critical because they allocated resources and budget for all development project.

The business unit head awareness training was followed up by role-specific training targeting each of the main roles: business analyst, architect, developer, tester and managers within the application development teams. The training focused on teaching individuals the responsibilities and activities specific to their role within their organization’s implementation of the SDL. All of the training was customized to highlight the Company’s specific problems found during the assessments, discussion of the severity of the problems and quotes from senior management showing support for the program.

## Step 2 – Integration with the Project Management Office

While the training was being rolled out, other members of the SDL Program team engaged the Company’s Project Management Office (PMO). The Company’s PMO owned many of the gates within the software development lifecycle, so it was important to ensure that the PMO’s process considered the necessary security process steps, artifacts, and assurance activities. For example, the PMO needed to update the list of required deliverables of the design phase to include the results from Threat Modeling. Table 2 shows the specific security deliverables for each development phase.

Table 2 - PMO Process Deliverables and Gates

Development Phase	Security Activity	Deliverables and Process Gates
On-boarding and	• SDL Awareness Campaign	• Pre and Post course



Training	<ul style="list-style-type: none"> <li>• Role-specific SDL training</li> </ul>	student evaluations
Functional and Non-functional technical requirements	<ul style="list-style-type: none"> <li>• Security Requirements</li> <li>• Abuse Case Definition</li> </ul>	<ul style="list-style-type: none"> <li>• Security Requirements integrated with other functional and non-functional requirements</li> </ul>
Architecture and Design	<ul style="list-style-type: none"> <li>• Threat Modeling</li> <li>• Secure Design Review</li> </ul>	<ul style="list-style-type: none"> <li>• Threat Models</li> <li>• Secure Design Review portions of the Security Scorecard complete</li> </ul>
Development	<ul style="list-style-type: none"> <li>• Secure Code Guidelines</li> <li>• Secure Code Reference Card</li> <li>• Secure Code Review</li> </ul>	<ul style="list-style-type: none"> <li>• Static Analysis tools run and results reviewed by Information Security</li> <li>• Secure Code Review portions of the Security Scorecard complete</li> </ul>
System Integration Testing	<ul style="list-style-type: none"> <li>• Fuzz Testing</li> <li>• Manual Security Testing</li> </ul>	<ul style="list-style-type: none"> <li>• Fuzz Testing and Manual Security Testing results integrated into bug-tracking and bug triage process</li> </ul>
Deployment	<ul style="list-style-type: none"> <li>• Security Scorecard</li> </ul>	<ul style="list-style-type: none"> <li>• Final PMO sign-offs on Security Scorecard</li> </ul>

The Information Security group worked with the PMO to develop a common set of security requirements for all applications as well as a Security Scorecard. The scorecard addresses the specific security concerns that are reflected in the security requirements as well as addressing specific technical issues that must be addressed for all application. Each phase of the development cycle has specific sections of the Security Scorecard and the PMO uses the scorecard at the end of the cycle as part of the “go to production” evaluation.

### Step 3 – Pilot the SDL with Development Teams

During the rollout of the SDL to the initial application development teams, members of the project acted as the security advisor to the team. In addition, another SDL Program team member coached the project management and development managers on how to implement the SDL. For architects, the security advisor helped review designs to ensure that the issues in



the Security Scorecard were addressed. The additional coaching was used to reinforce the lessons learned during role-specific training.

For use within the pilots, the SDL Program decided to purchase a static-analysis tool and to assemble a red team within the Information Security group to run the tool and provide analysis to the development teams. They established an internal secure code reviewing service which development teams are required to use. The SDL Program team decided on a central approach for the pilots as a way to build internal knowledge and expertise about using the tool before trying to implement it business unit or enterprise wide.

At the end of the two pilots using the SDL, the same external assessments were done on the two applications. The result was that there was a notable reduction of the common web application vulnerabilities. The Risk and Compliance office was very happy with these results. The feedback from the development teams was mixed. Some of the team felt that the SDL did not add much overhead and they appreciated the “additional testing”. Some, however, felt that the additional security activities were cumbersome and confusing.

## **Next Steps**

Senior management within the healthcare business unit decided that the SDL should be rolled out to additional healthcare applications. The SDL Program team is taking the lessons learned from the first two applications and revising the process, adding additional standards and making the technical guidance more specific to the company’s development framework. The SDL Program team is evaluating tools and processes for automating both static and dynamic security vulnerability analysis with each the development teams rather than providing these capabilities as a central service. The central services work well and provide the required expertise to run the tools and interpret the results, but the Information Security group is worried whether they have sufficient capacity to service all of the application teams within all of the Company’s business units.

## **Authors**

Scott Matsumoto, Principal Consultant, Cigital, Inc.

Chin Dou, Technical Manager, Cigital Inc.

Brian Mizelle, Managing Principal, Cigital Inc.