CISQ Quality Standard Overview and Integrating Security into SDLC

By Shahid N. Shah
This and many of my other presentations are available at

http://www.SpeakerDeck.com/shah

@ShahidNShah

shahid@shah.org
Who is Shahid?

• 25+ years of software engineering and multi-discipline complex IT implementations (Gov., defense, health, finance, insurance)

• 12+ years of high-security, regulated IT and safety-critical medical devices experience (blog at http://healthcareguy.com)

• 15+ years of technology management experience (government, non-profit, commercial)
What’s this talk about?

Theme

Security cannot be bolted on at the end of the system development lifecycle (SDLC)

Key Takeaways

• Security is an emergent property of the overall quality of a system. The Consortium for IT Software Quality (CISQ), led by Object Management Group (OMG), does a great job of explaining how security and quality are tightly integrated.

• Security must be fully integrated in to the SDLC. Microsoft’s SDL is a great approach to doing so.
What is SDL?

• Security Development Lifecycle (SDL) is a system development process introduced by Microsoft in 2004 for improving Security in their products

• Can be used with any software development model
What is CISQ, where does SDL fit?

Consortium for IT Software Quality produces computable metrics standards for software quality and focuses on establishing standards for:

- Software Quality
- Functional Sizing

Software quality characteristics selection include:

- Reliability
- Performance Efficiency
- Security
- Maintainability

More info: http://it-cisq.org/standards-page/
CISQ Quality Characteristics

Software Product Quality Characteristics from ISO/IEC 25010.
SDL and Regulatory Compliance

Security & compliance contribute significantly to an enterprise's reputation.

Risk management is the approach an organization uses to remove or reduce harm to its assets and personnel.

- FISMA
- FedRAMP (FISMA for Cloud)
- HIPAA & Omnibus
- PCI
- FDA Regulations
- etc...
Injecting a security lifecycle into any SDLC

Source: Microsoft
SDL into SDLC benefit: Security

- Microsoft SQL Server: 91% Fewer Vulnerabilities in SQL Server 2005
- Removal of 50% of Security Vulnerabilities during development would reduce Configuration Management, Support costs by 75% each – Gartner Research
- Fixing security post development costs roughly 3 times more than the cost of built-in security
SDL into SDLC benefit: Compliance

- Improves compliance with a holistic, multidisciplinary approach.
- Enables organizations to take a proactive vs. reactive stance.
- Eliminates redundancies and coordinates processes.
SDL into SDLC benefit: Lower Costs

- **NIST:** code fixes after release can result in 30 times the cost of fixes performed during design phase.
- The **Forrester** Consulting State of Application Security study reported that organizations implementing an SDL process showed better ROI results than the overall surveyed population.
- **Aberdeen Group** demonstrated adopting an SDL process generates a stronger return on investment (four-times higher).
SDLC methodology is no excuse

Security and compliance can be injected into any type of development methodology
Creating your own “Secure SDLC”

Requirements
- Security Requirements
- Privacy Requirements

Design
- Threat Modelling
- Security Design Review

Development
- Static Code Analysis
- Peer Code Review

Testing
- Security Test Plan
- Security Test Cases

Deployment
- Final Security Review
- Security Monitoring, POA&Ms
Injecting Security into Requirements

• Key phase
• Establish Security requirements
• Establish Security Quality requirements
• Establish Security Code review process requirements
• Establish Privacy and Compliance requirements
Injecting Security into Design

• Prepare Security Design requirements
  – Like Cryptographic, validate design with requirements

• Identify attack surfaces and design system

• Identify data flows

• Perform Threat Modelling using Tools like MS Threat Modelling Tool
  – MS Tool
Injecting Security into Implementation

- Use approved tools and compiler and linker settings
- Deprecate un-safe functions
- Remove dead code
- Perform static code analysis
- Integrate security scanners in to continuous integration (CI) tools such as Jenkins, Cruise Control
- Provide reports of the tool runs, including unit tests, integration test results visibly to stakeholders
Injecting Security into Testing

• Run Dynamic Security Scanners
  – App Scan, Skipfish, Veracode etc...
  – **CSX Tool**, **MS Starter Kit**

• Perform Fuzz Testing
  – **OWASP Fuzz Testing guide**

• Perform Attack Surface Vector review
  – **MS Attack Surface Analyzer**
Injecting Security into Deployment

- Create an Incident Response Plan
- Perform Continuous Monitoring of the Security Controls
- Conduct Final Security Review
- Certify Release and Archive
Who’s responsible for all this?

<table>
<thead>
<tr>
<th>Phase</th>
<th>Primary Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements/Initiation</td>
<td>• AO (Authorizing Official)/Business Owner</td>
</tr>
<tr>
<td></td>
<td>• CIO, Requirements Manager, Configuration Management Manager</td>
</tr>
<tr>
<td></td>
<td>• Product and Program Management Team</td>
</tr>
<tr>
<td></td>
<td>• Operations Team, Information System Security Officer</td>
</tr>
<tr>
<td></td>
<td>• Enterprise Architect</td>
</tr>
<tr>
<td>Design</td>
<td>• System Architect</td>
</tr>
<tr>
<td></td>
<td>• Developers</td>
</tr>
<tr>
<td>Development</td>
<td>• Director of Development</td>
</tr>
<tr>
<td></td>
<td>• Developers</td>
</tr>
<tr>
<td>Testing</td>
<td>• Test Director</td>
</tr>
<tr>
<td></td>
<td>• Testers</td>
</tr>
<tr>
<td>Deployment</td>
<td>• Operations Head</td>
</tr>
<tr>
<td></td>
<td>• DevOps</td>
</tr>
</tbody>
</table>

- Follow the practices recommended in the NIST’s RMF Framework for better effective integration of security into SDLC
- Follow the guidelines in the NIST SP 800-53 for Security Controls to get an overview on security requirements
Our experience

• Difficult to start, easier to adopt if management buys off, training is done, right roles are incorporated, and project estimation includes all the required work.

• If you don’t get more time, at least start with OWASP Top 10 security flaws and recommendations to address them.
What do we do at Netspective?

• Institute Security in to SDLC using NUP (Netspective Unified Process)
• Identify Security Controls (NIST SP 800-53) and assess them as the changes occur in system assets with tools such as Netspective Opsfolio
• Track compliance as the changes occur to assets using Netspective Opsfolio
• Monitor the system reliability with tools such as Netspective Watchtower
Tools

- Server scanning tools (Nessus)
- Application scanning tools (IBM App Scan, OWASP ZAP Proxy, Veracode, Skipfish etc...)
- Network scanning tools (IDS, Intrusion detection)
- Pen testing tools
- Netspective NUP
- Netspective Opsfolio
- Netspective Watchtower
Summary

Major lesson
It’s better, faster, and cheaper to integrate security into your system development lifecycle.

All the tools and approaches you need are readily available, most for free.

Key Takeaways
- **Security is an emergent property of the overall quality of a system.** CISQ, led by OMG, does a great job of explaining how security and quality are tightly integrated.
- **Security must be fully integrated in to the SDLC.** Microsoft’s SDL is a great approach to doing so but you can roll your own approach as well.
Thank You

Visit
http://www.netspective.com
http://www.healthcareguy.com
E-mail shahid.shah@netspective.com
Follow @ShahidNShah
Call 202-713-5409