Microsoft SDL: Agile Development

Nick Coblentz, CISSP, Microsoft MVP
Senior Security Consultant
AT&T Consulting
Nick.Coblentz@gmail.com
http://nickcoblentz.blogspot.com
http://twitter.com/sekhmetn

OWASP
November 11, 2010

The OWASP Foundation
http://www.owasp.org

Bio

- AT&T Consulting:
  - Application Security
    - Penetration testing
    - Code review
    - Architecture and design reviews
    - Application security program development
    - Secure development methodology improvement

- Research & Presentations
  - **OWASP AppSec Research & Ireland 2010 Conferences**
  - **ISSA Journal: Web Application Security Portfolios**
  - **SAMM Interview Template**
  - **Reducing Info Disclosure in ASP.NET Web Services and WCF Data Services**
  - Turn Application Assessment Reports into Training Classes
  - Observed Secure Software Development Stages
  - Vulnerability Tracking, Workflow, and Metrics with Redmine
  - Using Microsoft's AntiXSS Library 3.1

Adrian Lane:
http://securosis.com/blog/agile-development-and-security/

Microsoft SDL For Agile Released

Source: http://blogs.msdn.com/sdl/archive/2009/11/10/announcing-sdl-for-agile-development-methodologies.aspx

Microsoft SDL

Microsoft Security Development Lifecycle (SDL)

Components:

- Best Practices
- Processes
- Standards
- Security Activities
- Tools

Goal:

"minimize security-related vulnerabilities in the design, code, and documentation and to detect and eliminate vulnerabilities as early as possible in the development life cycle."

Microsoft® Security Development Lifecycle

Which Software?

SDL applies to software that:

- Is used in Business environments
- Stores or transmits PII
- Communicates over the Internet or other networks

Microsoft® SQL Server® 2008

Microsoft® Office Live

bing™

Windows Live™

Source: Microsoft's Product Website

SDL Principles and Process

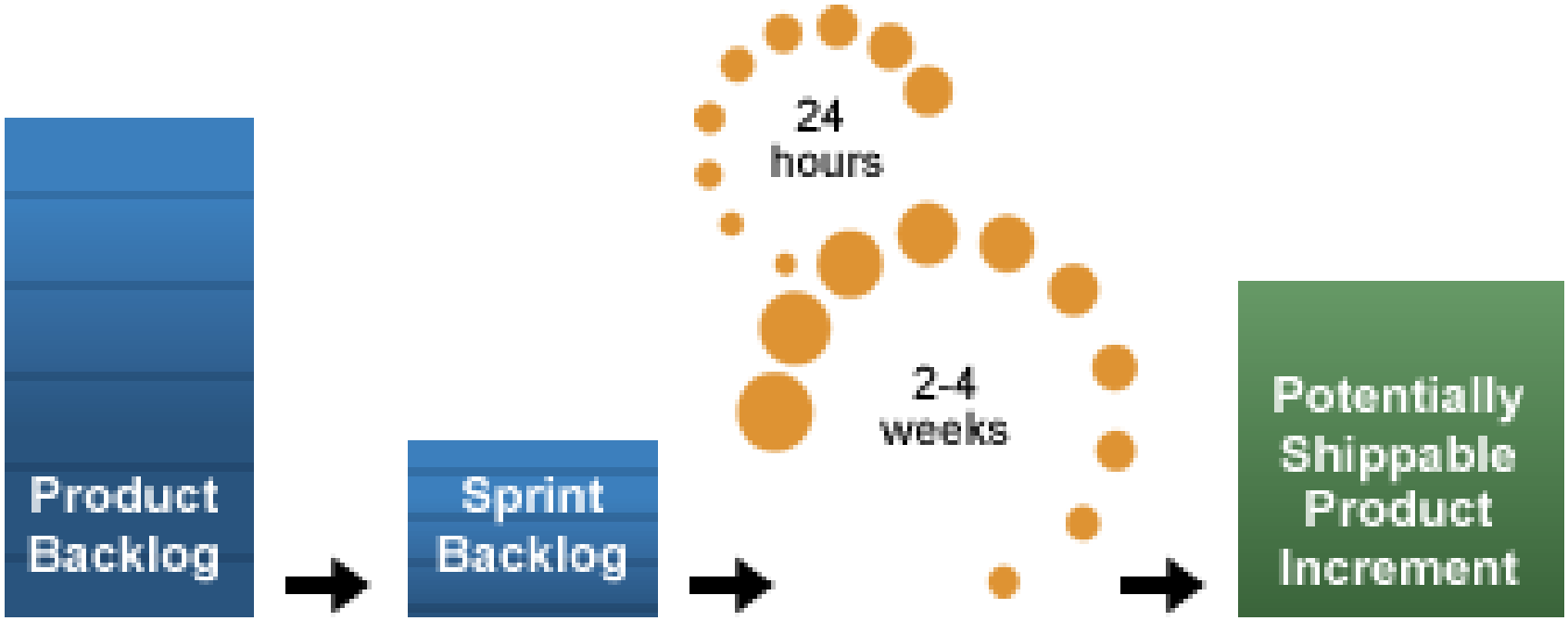
SD3+C

- Secure by Design
- Secure by Default
- Secure in Deployment
- Communications

PD3+C

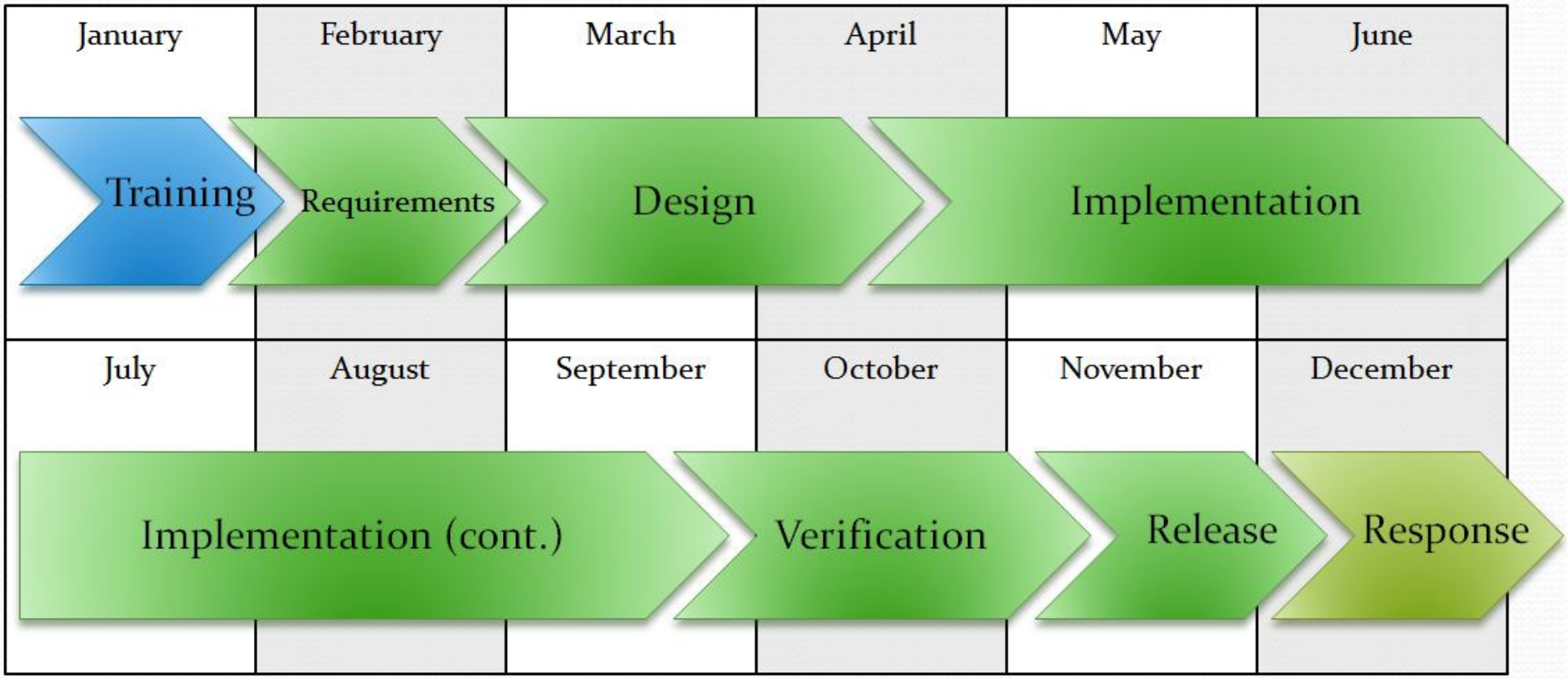
- Privacy by Design
- Privacy by Default
- Privacy in Deployment
- Communications

What is Agile Development?



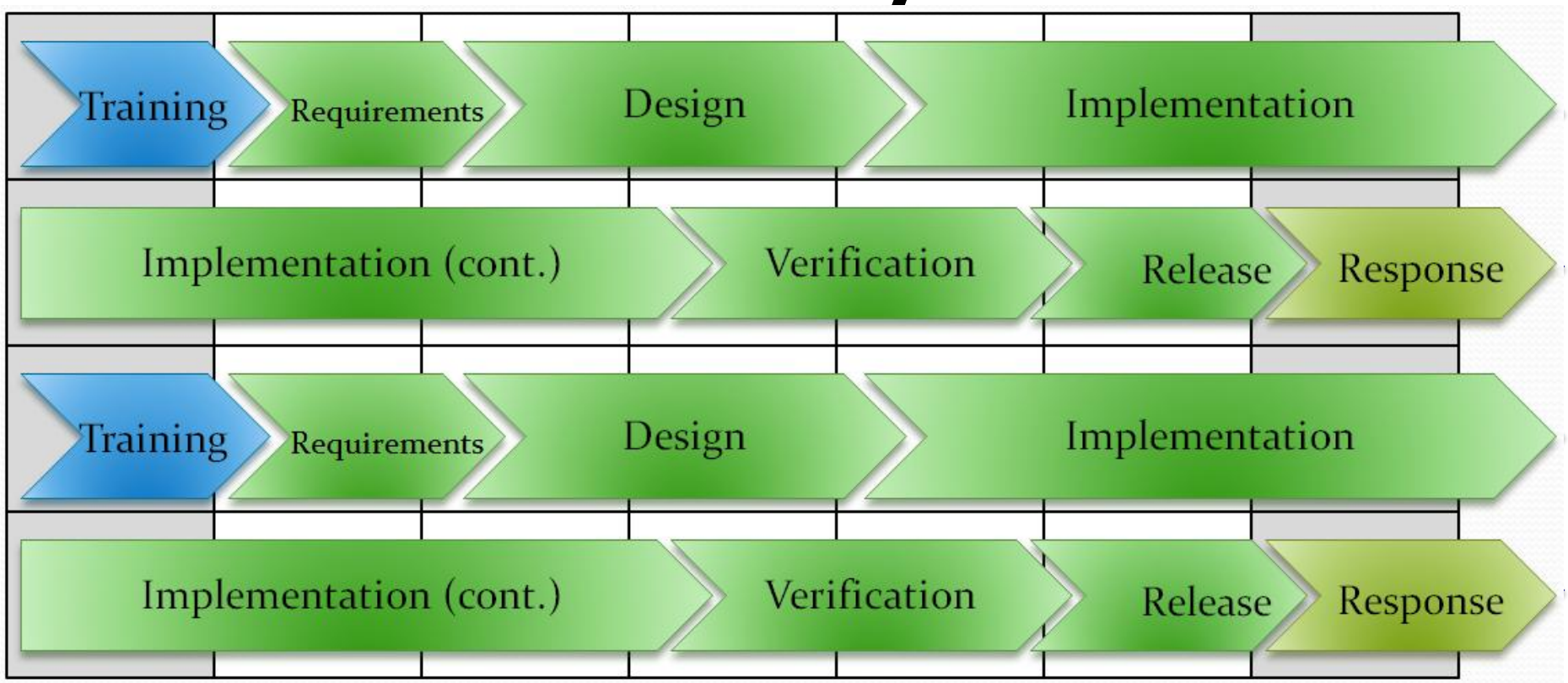
Source: http://www.scrumalliance.org/pages/what_is_scrum

SDLC (Waterfall Methodology)

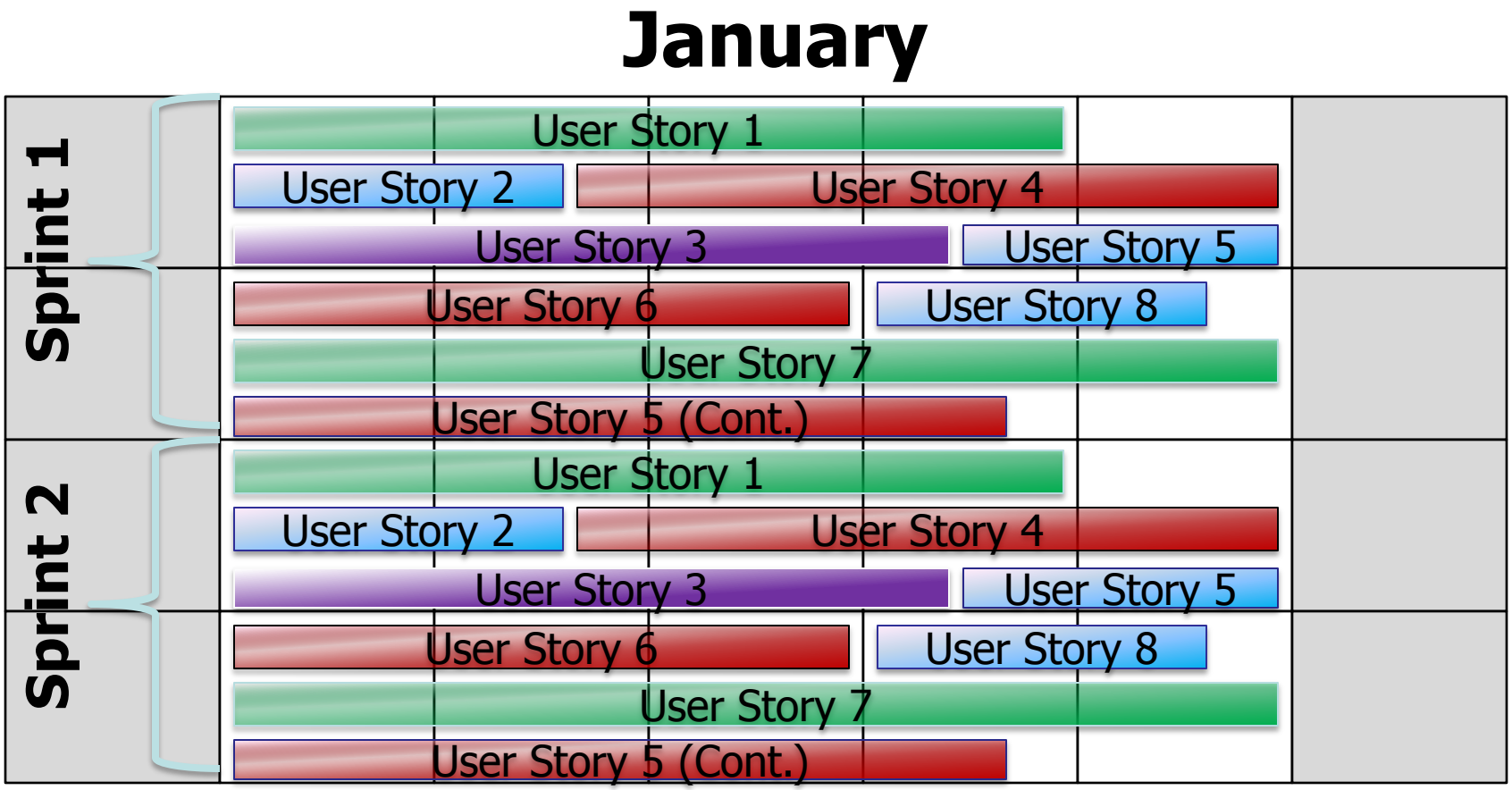


This Is NOT Agile Development

January



Agile Development



Agile Development

Source: http://www.scrumalliance.org/pages/what_is_scrum

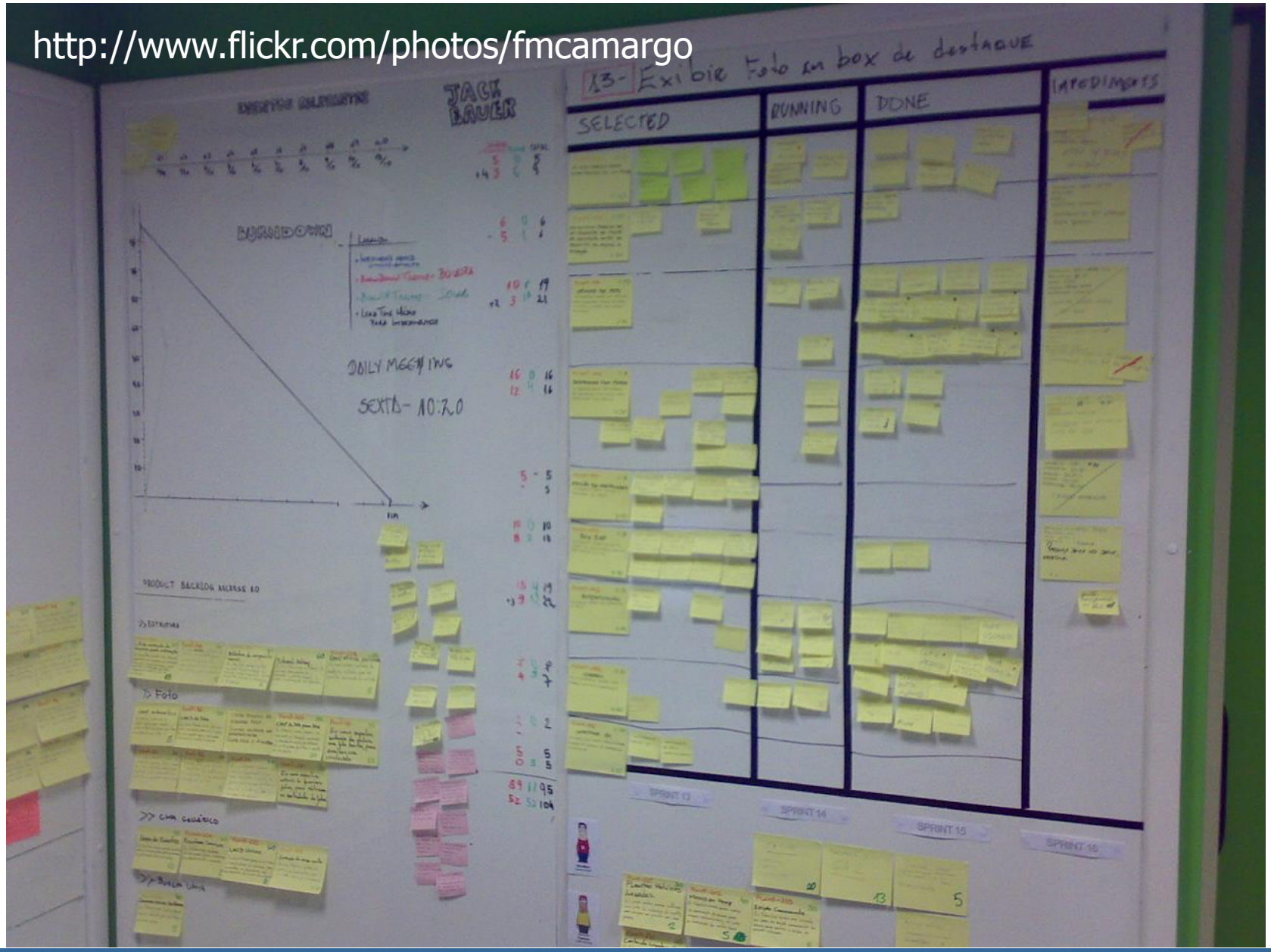
- Cross-functional, self-organizing teams
- Short, time-boxed development iterations
- Delivery of small functional stories
- No *extensive* up front design or documentation

Planning and Design

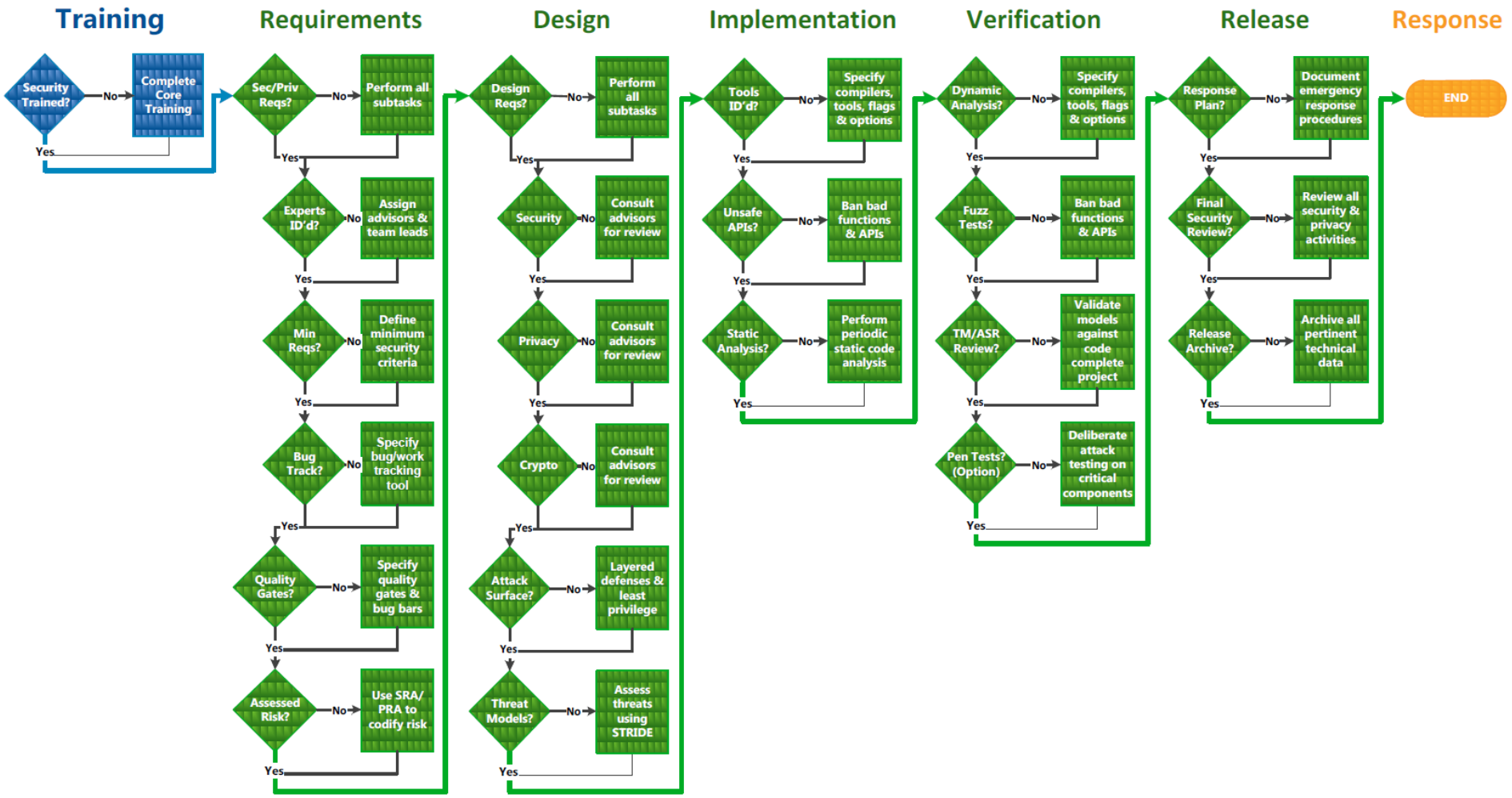
Planning and Design (cont.)

http://www.flickr.com/photos/acarlos1000

User Stories and Documentation



SDL SECURITY ACTIVITIES



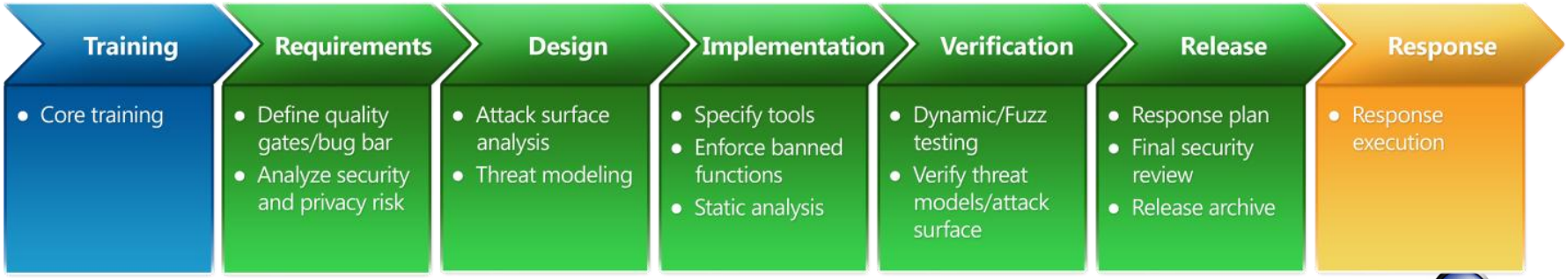
Source: Simplified Implementation of the Microsoft SDL

SDL Security Activities

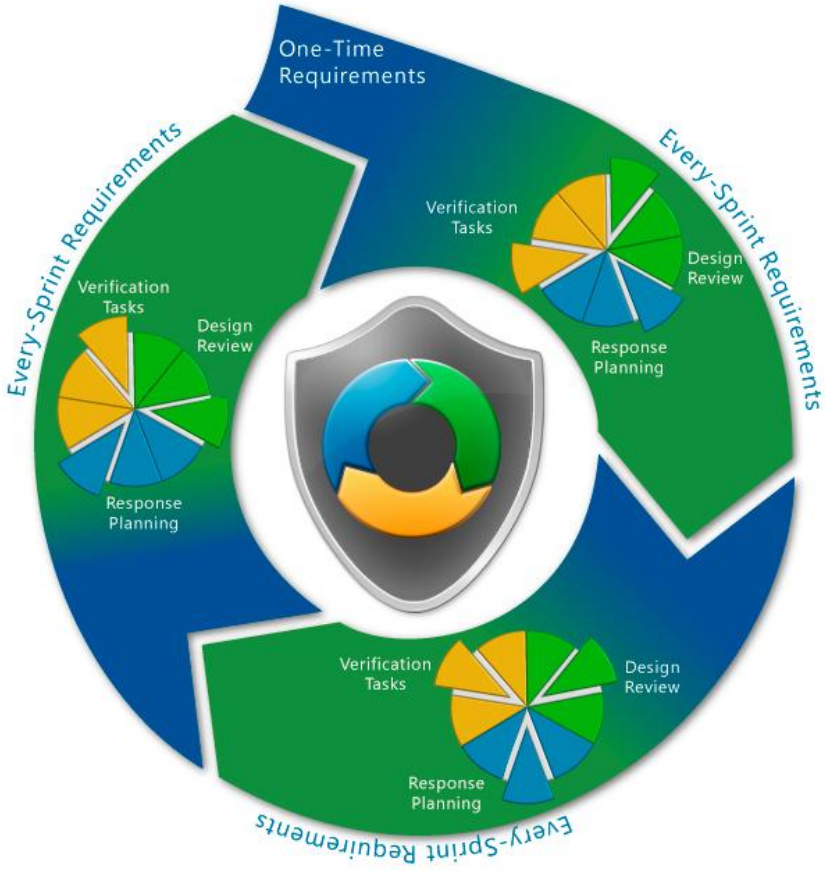
- Training
- Requirements
  - Security Requirements
  - Quality Gates/Bug Bars
  - Security and Privacy Risk Assessment
- Design
  - Design Requirements
  - Attack Surface Reduction
  - Threat Modeling
- Implementation
  - Use Approved Tools
  - Deprecate Unsafe Functions
  - Static Analysis
- Verification
  - Dynamic Program Analysis
  - Fuzz Testing
  - Threat Model and Attack Surface Review
- Release
  - Incident Response Plan
  - Final Security Review
  - Release/Archive
- Optional Activities
  - Manual Code Review
  - Penetration Testing
  - Vulnerability Analysis of Similar Applications

Traditional SDL Pain Points for Agile

- Can't complete all SDL activities in each sprint
- Requirements, architecture, and design evolves over time
- Threat model/documentation becomes dated quickly
- Data sensitivity, protection, and connections to third parties may not be immediately known
- Teams don't include application security specialists



Microsoft SDL For Agile Development



SDL Requirement Categories:

- Every-Sprint
- Bucket
  - Verification Tasks
  - Design Review Tasks
  - Response Planning Tasks
- One-Time

Source: Microsoft SDL v4.1a

Every-Sprint SDL Requirements

"…so essential to security that no software should ever be released without these requirements being met."

Examples:

- Update the threat model
- Communicate privacy-impacting design changes to the team's privacy advisor
- Fix all issues identified by code analysis tools for unmanaged code
- Follow input validation and output encoding guidelines to defend against cross-site scripting attacks

Bucket SDL Requirements

- Teams prioritize the pool of tasks over many sprints
- Each sprint, one task from each bucket completed
- Each tasks must be completed at least every 6 months

Examples:

- Security Verification Tasks
  - Run fuzzing tools
  - Manual and automated code review
- Design Review Tasks
  - Conduct privacy review
  - In-depth threat model
- Response Planning Tasks
  - Define security/privacy bug bar
  - Create support documents

One-Time Requirements

Why?

- Repetition not necessary
- Must occur at the beginning of the project
- Not possible at the beginning of the project

Examples:

- Configure bug tracking system (3 months)
- Identify security/privacy experts (1 month)
- Baseline threat model (3 months)
- Establish a security response plan (6 months)

SDL-Agile Appendix

Appendix Q: SDL-Agile Bucket Requirements

Bucket A: Security Verification

Title	Requirement/ Recommendation	Applies to Online Services	Applies to Managed Code	Applies to Native Code
Debug the application with the Application Verifier enabled	Requirement			X
Disable tracing and debugging in ASP.NET applications	Requirement	X	X	
Investigate and service any reported /GS crashes	Requirement			X
Perform ActiveX control fuzzing	Requirement	X		X
Perform attack surface analysis	Requirement	X	X	X
Perform binary analysis (BinScope)	Requirement	X	X	X
Perform COM object testing	Requirement			X
Perform cross-domain scripting testing	Requirement	X	X	X
Perform file fuzz testing	Requirement	X		X
Perform RPC fuzz testing	Requirement	X		X
Conduct in-depth manual and automated code review for high-risk code	Recommendation	X	X	X
Perform data flow testing	Recommendation	X	X	X

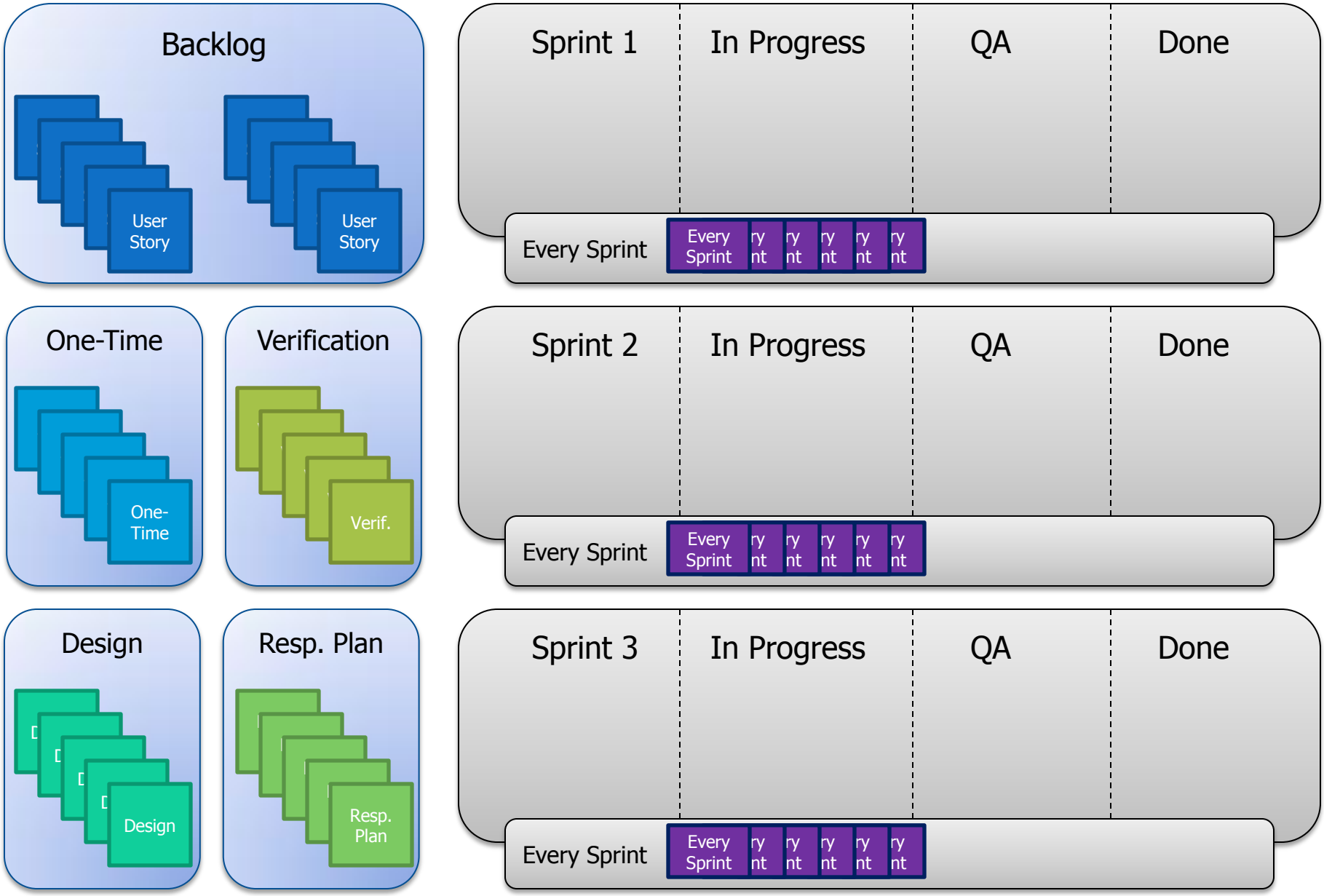
SDL-Agile Appendix: Deadlines

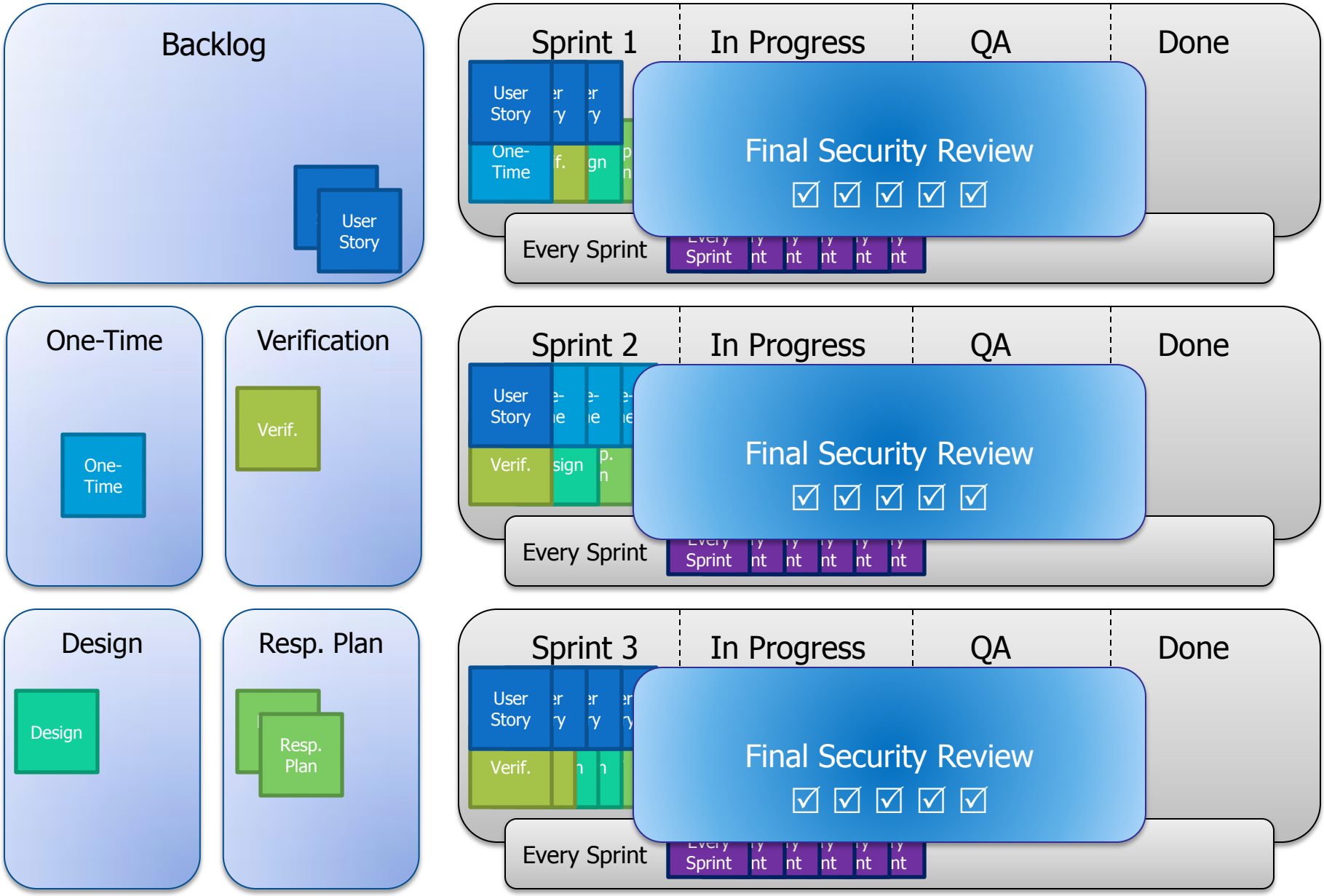
Appendix R: SDL-Agile One-Time Requirements

Title	Requirement/ Recommendation	Completion Deadline (months)	Applies to Online Services	Applies to Managed Code
Avoid writable PE segments	Requirement	6	X	
Configure bug tracking to track the cause and effect of security vulnerabilities	Requirement	3	X	X
Create a baseline threat model	Requirement	3	X	X
Determine security response standards	Requirement	6	X	X
Establish a security response plan	Requirement	6	X	X
Identify primary	Requirement	1	X	X

Final Security Review

- Occurs at the end of every sprint
- Checklist:
  - ☑All every-sprint requirements have been completed
  - ☑No one-time requirements have exceeded deadline
  - ☑At least one requirement from each bucket category has been completed
  - ☑No bucket requirements exceed the six month deadline
  - ☑No security or privacy bugs are open that exceed the severity threshold


Backlog
User Story
User Story
One-Time
One-Time
Verification
Verif.
Design
Design
Resp. Plan
Resp. Plan
Sprint 1
In Progress
QA
Done
Every Sprint
Every Sprint
Sprint 2
In Progress
QA
Done
Every Sprint
Every Sprint
Sprint 3
In Progress
QA
Done
Every Sprint
Every Sprint

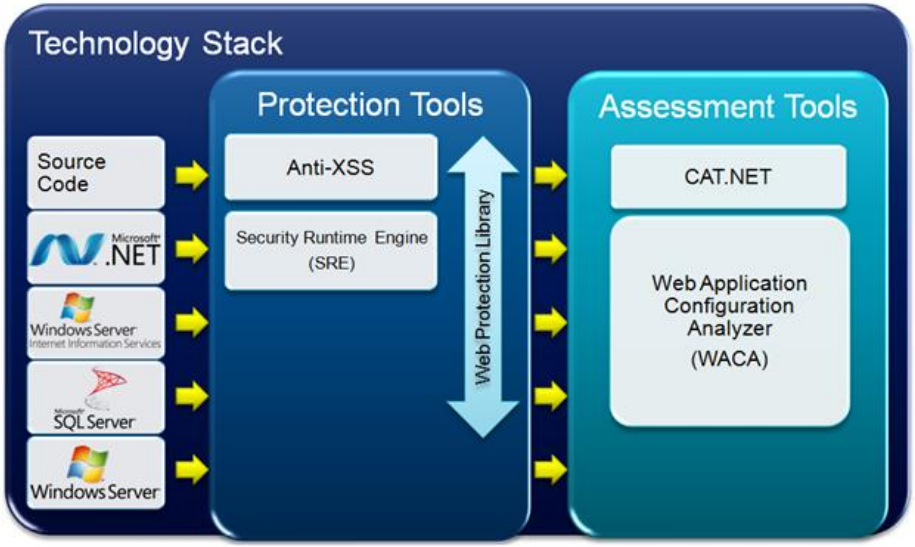

Backlog
User Story
Sprint 1
In Progress
QA
Done
User Story
One-Time
Final Security Review
Every Sprint
Sprint
One-Time
Verification
Verif.
Sprint 2
In Progress
QA
Done
User Story
Verif.
Final Security Review
Every Sprint
Sprint
Design
Design
Resp. Plan
Resp. Plan
Sprint 3
In Progress
QA
Done
User Story
Verif.
Final Security Review
Every Sprint
Sprint
OWASP

Making SDL-Agile Manageable

- Documented standards
- Security training
- Automation
  - Continuous Integration
    - Secure Configuration
    - Security Unit Tests
    - Automated Secure Code Analysis
    - Automated Deployment and Vulnerability Scanning
- Process
  - Continuous updates to the threat model
  - SDL Process Templates for VSTS
  - MSF-Agile + SDL Process Template
- Light on security artifacts/documentation

Making SDL-Agile Manageable

- Tooling
  - Code Analysis/Scanning
    - CAT.NET
    - MiniFuzz
    - BinScope Binary Analyzer
    - Fiddler w/ Watcher
    - FxCop
  - MS Threat Modeling Tool



CAT.NET: Cross-site Scripting Vulnerability

Analysis Information

Analysis Engine Version	1.0.3455.36250
Created by	[redacted]
Start time	Sunday, February 28, 2010 1:34:46 PM
Stop time	Sunday, February 28, 2010 1:34:47 PM
Elapsed time	00:00:00.6100000
Data flow graph	5 nodes, 5 edges
Targets	C:\Users\[redacted]\Desktop\UnsignedUnecryptedViewStateExploit\UnsignedUnecryptedViewStateEx

Cross-Site Scripting (ACESEC05)

1 results

Result #1

Summary

Problem	A cross-site scripting vulnerability was found through a user controlled variable that enters the application at Default.aspx.cs:21 through the variable stack1 which eventually leads to a cross-site scripting issue at Default.aspx.cs:21.
Resolution	Use the Anti-XSS library to properly encode the data before rendering it
Entry Variable	stack1
Confidence	High

Source Context	Line	Input Variable	Statement
Default.aspx.cs	21		lblPayload.Text = txtBox1.Text;
Default.aspx.cs	21	Return from TextBox.get_Text	lblPayload.Text = txtBox1.Text;

Making SDL-Agile Manageable

- Libraries
  - Web Protection Library (WPL)
    - Encoder/~~Anti-XSS Library~~
    - Security Runtime Engine (SRE)
    - Sanitizer.GetSafeHTML

Web Protection Library - Encoder/~~AntiXSS~~

Encoder Methods

Encoder Class See Also Send Feedback

Microsoft.Security.Application.Encoder

The Encoder type exposes the following members.

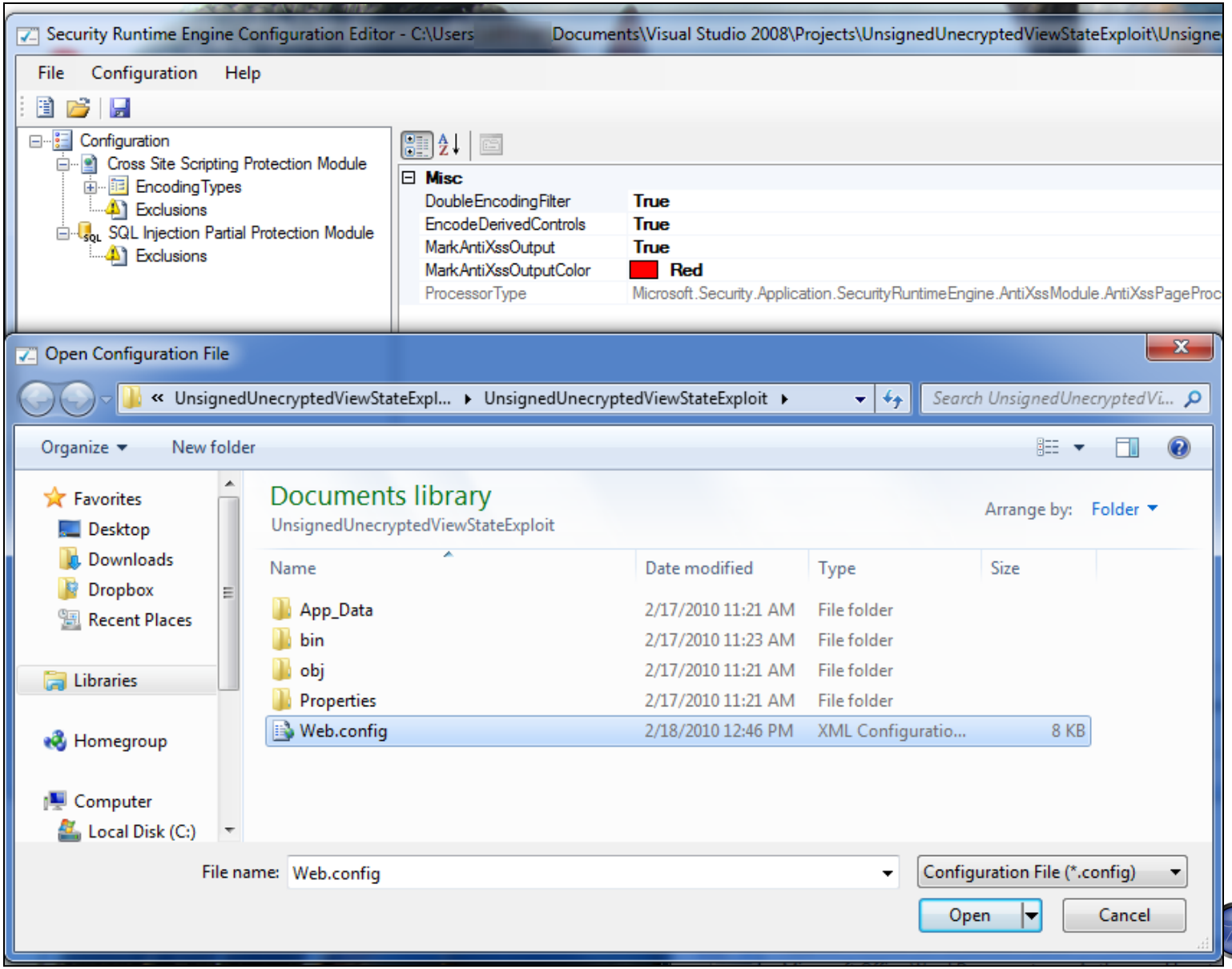
Methods

	Name	Description
S	CssEncode	Encodes input strings used in Cascading Style Sheet (CSS) elements.
S	HtmlAttributeEncode	Encodes input strings for use in HTML attributes.
S	HtmlEncode	Overloaded.
S	JavaScriptEncode	Overloaded.
S	LdapEncode	Encodes input strings used in Lightweight Directory Access Protocol (LDAP) search queries.
S	UrlEncode	Overloaded.
S	VisualBasicScriptEncode	Encodes input strings for use in Visual Basic Script.
S	XmlAttributeEncode	Encodes input strings for use in XML attributes.
S	XmlEncode	Encodes input strings for use in XML.

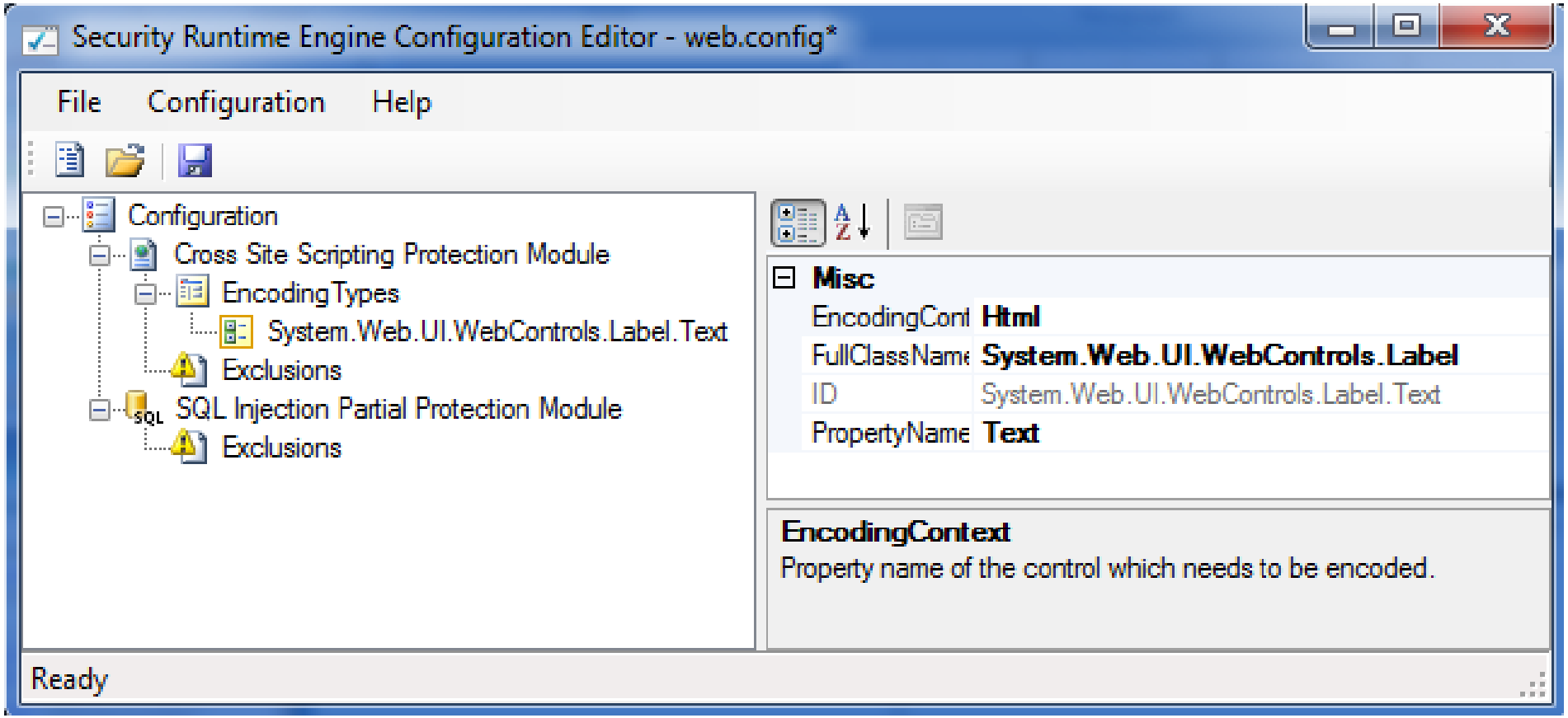
The Security Runtime Engine (SRE)

- "The Security Runtime Engine (SRE) is an HTTP module that acts like a gatekeeper to protect ASP.NET web applications from cross-site scripting (XSS) attacks."
- "It works by inspecting each control that is being reflected by ASP.NET and then automatically encoding data of vulnerable controls in their appropriate context."
- SRE Configuration Editor GUI Tool

The Security Runtime Engine (SRE)



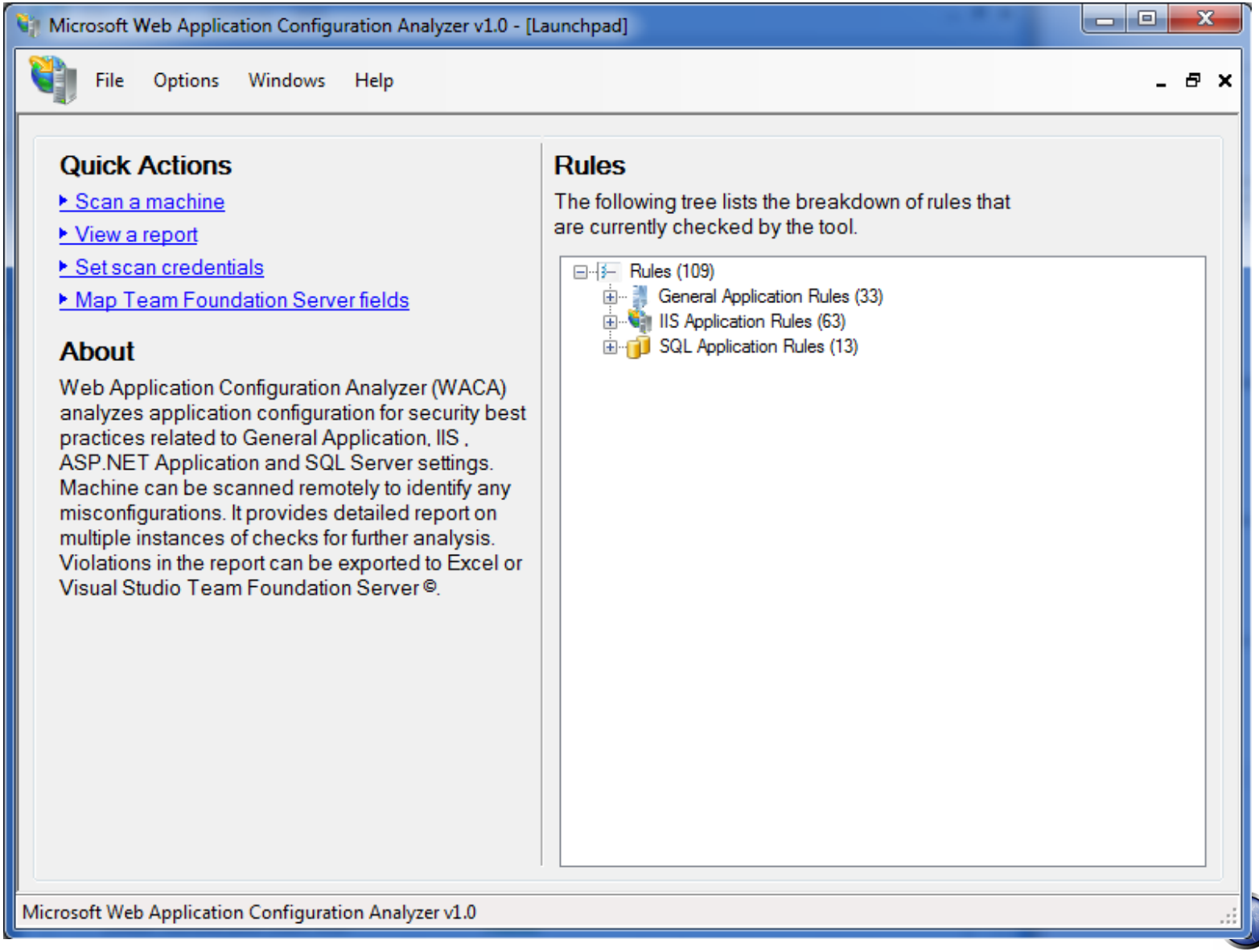
The Security Runtime Engine (SRE)



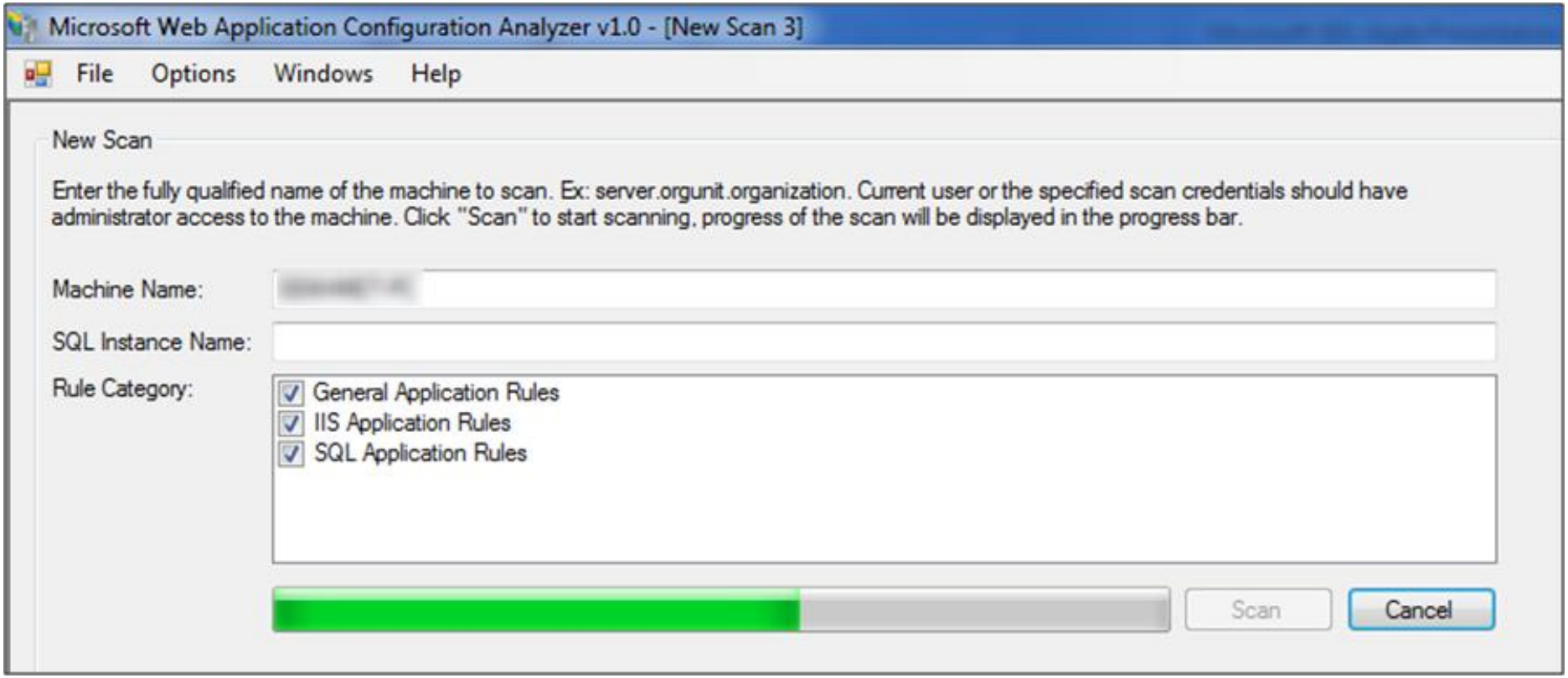
Making SDL-Agile Manageable

- Deployment
  - Web Application Configuration Analyzer (WACA)
  - Microsoft Baseline Security Analyzer
  - Web.config Security Analyzer (WCSA)

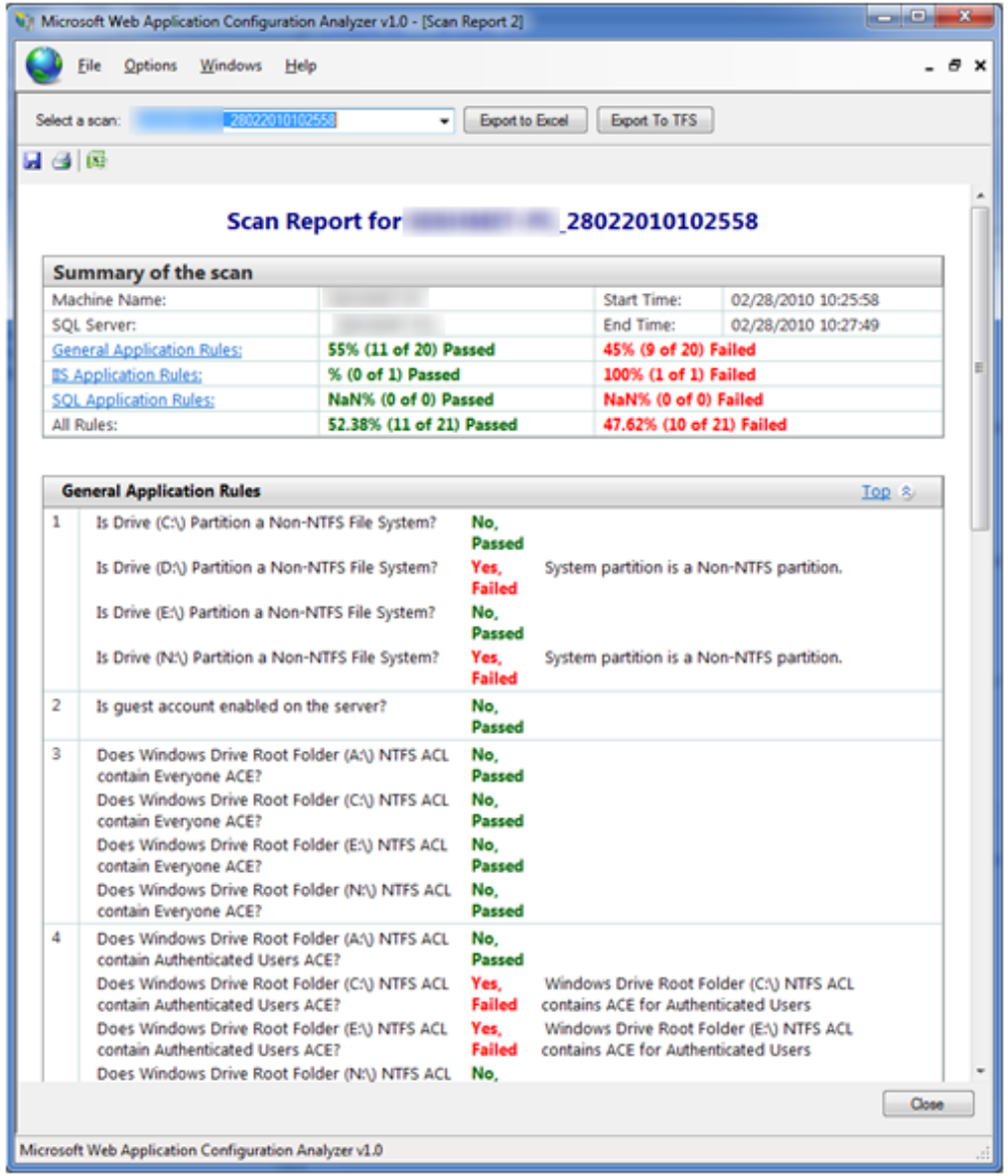
Web Application Configuration Analyzer



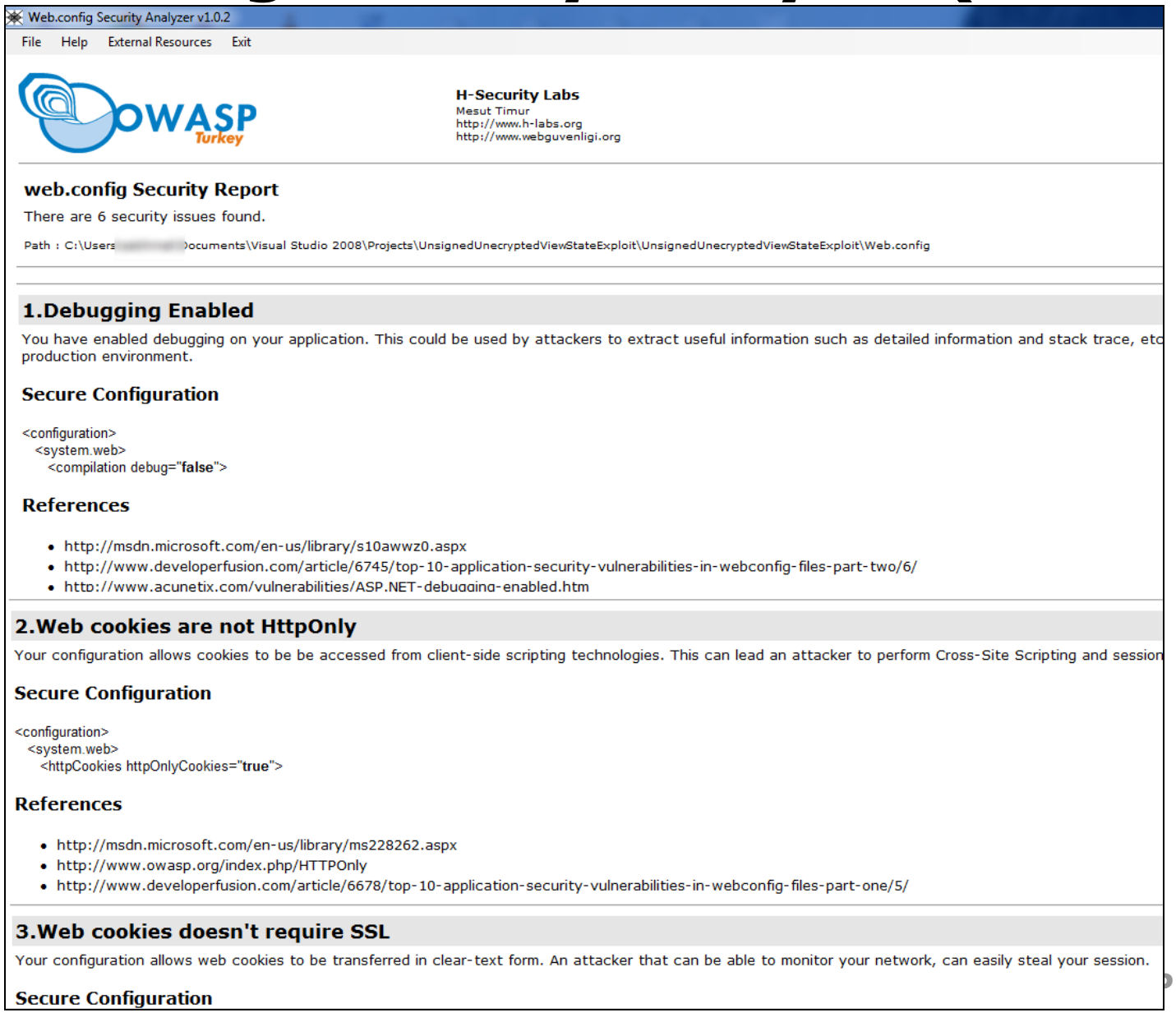
Web Application Configuration Analyzer



Web Application Configuration Analyzer



Web.config Security Analyzer (WCSA)



Making SDL-Agile Manageable

- **Education**, **secure coding standards, automation** and **tools** play a significant role in making secure Agile development efficient and economical
- Don't forget:
  - Periodic manual security activities are also a must
  - All of this must fit within a **repeatable, mature process**

Summary and Questions

More Information:
http://www.microsoft.com/sdl

Nick Coblentz, CISSP
Senior Consultant, AT&T Consulting
Nick.Coblentz@gmail.com
http://nickcoblentz.blogspot.com
http://www.twitter.com/sekhmetn

- Microsoft releases SDL-Agile Guidance in Nov. 2009
- Treats SDL Activities like team-prioritized User Stories
  - 3 Categories: One-time, Every-time, and Bucket
- Increased success with the implementation of training, automation, tools, and standards