Agile and Secure: Can We Be Both?

Keith Landrus
Director of Technology
Denim Group Ltd.
keith.landrus@denimgroup.com
(210) 572-4400

Copyright © 2006 - The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the Creative Commons Attribution-ShareAlike 2.5 License. To view this license, visit http://creativecommons.org/licenses/by-sa/2.5/

The OWASP Foundation
http://www.owasp.org/
The Agile Practitioner’s Dilemma

Agile Forces:
- More responsive to business concerns
- Increasing the frequency of stable releases
- Decreasing the time it takes to deploy new features

Secure Forces:
- More aggressive regulatory environment
- Increasing focus on need for security
- Traditional approaches are top-down, document centric
Objectives

- Background
- Goals of Agile Methods
- Goals of Secure Development Lifecycle (SDL)
- Review the Momentum of Agile Methods
- Look at An Integrated Process
- Challenges & Compromises
Notable Agile Methods

- eXtreme Programming (XP)
- Feature Driven Development (FDD)
- SCRUM
- MSF for Agile Software Development
- Agile Unified Process (AUP)
- Crystal Clear
- Dynamic Systems Development Method (DSDM)
Manifesto for Agile Software Development

*Individuals and interactions* over processes and tools

*Working software* over comprehensive documentation

*Customer collaboration* over contract negotiation

*Responding to change* over following a plan

Source: [http://www.agilemanifesto.org/](http://www.agilemanifesto.org/)
Agile’s Core Values

- Communication
- Simplicity
- Feedback
- Courage
Principles of Agile Development

- Rapid Feedback
  - The system is appropriate for the intended audience.
- Simple Design
  - The code passes all the tests.
  - The code communicates everything it needs to.
- Incremental Change
  - The code has the smallest number of classes and methods.
- Embracing Change
- Quality Work
Agile Practices

- The Planning Game
  - Customer: scope, priorities and release dates
  - Developer: estimates, consequences and detailed scheduling

- The Driving Metaphor

- Shared Vision

- On-Site Customer
  - Development iterations or cycles that last 1-4 weeks.
  - Release iterations as soon as possible (weekly, monthly, quarterly).

- Small Releases
More Agile Practices

- Test Driven
  - Programmer tests guide the development process. Red, Green, Refactor
  - Customer tests provide feedback to the team that the system is working as expected.

- Collective Ownership

- Coding Standards

- Pair Programming

- Continuous Integration
  - Continuously build, deploy and execute all of the system’s tests multiple times per day.
Agile Methods strive to...

- Adapt to ever-changing customer needs.

- Bring together small teams of highly talented individuals and remove obstacles that get in the way of developing quality systems.

- Maintain a strong emphasis on testing.
A secure product is one that protects the confidentiality, integrity, and availability of the customers’ information, and the integrity and availability of processing resources under control of the system’s owner or administrator.

-- Source: Writing Secure Code (Microsoft.com)
A Secure Development Process...

- Strives To Be A Repeatable Process
- Requires Team Member Education
- Tracks Metrics and Maintains Accountability

Sources:

“Writing Secure Code” 2nd Ed., Howard & LeBlanc

“The Trustworthy Computing Security Development Lifecycle” by Lipner & Howard
Secure Development Principles

- SD³: Secure by Design, Secure by Default, and in Deployment
- Learn From Mistakes
- Minimize Your Attack Surface
- Assume External Systems Are Insecure
- Plan On Failure
- Never Depend on Security Through Obscurity Alone
- Fix Security Issues Correctly
Secure Development Practices

- Education, Education, Education
- Threat Modeling
- Secure Coding Techniques
- Security Testing
- Security Code Reviews
Microsoft’s Secure Development Lifecycle (SDL)

- Requirements
- Design
- Implementation
- Verification
- Release

Security Training
- Security Kickoff & Register With SWF
- Security Design Best Practices
- Security Architecture & Attack Surface Review
- Threat Modeling

- Create Security Documentation and Tools For Product
- Prepare Security Response Plan
- Security Push
- Pen Testing
- Final Security Review

Security Servicing & Response Execution

Requirements → Design → Implementation → Verification → Release → Support & Servicing
SDL: Requirements Phase Activities

- Determine (or make contact with) the security advisor “security buddy”

- Identify key security objectives for the system

- Consider Security Feature Requirements
SDL: Design Phase Activities

- Define Security Architecture and Design Guidelines
- Document the Attack Surface
- Conduct Threat Modeling
- Define Supplemental Ship Criteria
SDL: Implementation Phase Activities

- Apply Common Coding Standards
- Apply Security-Testing Tools
- Apply Static-Analysis Code Scanning Tools
- Conduct Security Code Reviews
SDL: Verification Phase Activities

- Conduct the “Security Push”
  - Additional Security Code Reviews
  - Focused Security Testing
SDL: Release, Support & Servicing Activities

- Conduct the Final Security Review (FSR) Prior to Release
- Prepare to Respond to Vulnerability Reports
- Learn from Errors and Mistakes
Observations of the SDL in Practice

- Threat Modeling is the Highest-Priority Component

- Penetration Testing Alone is Not the Answer

- Tools Should be Complementary

- Microsoft’s experience has indicated that the SDL has been effective at reducing security vulnerabilities in their products.
Dr. Dobb’s says Agile Methods Are Catching On

41% of organizations have adopted an agile methodology

65% have adopted one or more agile techniques

Of the 2,611 respondents doing agile...

- 37% using eXtreme Programming
- 19% using Feature Driven Development (FDD)
- 16% using SCRUM
- 7% using MSF for Agile Software Development

Source: http://www.ddj.com/dept/architect/191800169
Agile Teams are “Quality Infected”

- 60% reported increased productivity
  - 6% reported a decrease

- 66% reported improved quality

- 58% improved stakeholder satisfaction
  - 3% reported a decrease
 Adoption Rate for Agile Practices

Of the respondents using an agile method…

- 36% have active customer participation
- 61% have adopted common coding guidelines
- 53% perform code regression testing
- 37% utilize pair programming
Let’s Look at Some Specific Agile Methods

- eXtreme Programming (XP)
- Feature Driven Development (FDD)
- SCRUM
- MSF for Agile Software Development
eXtreme Programming (XP)

- Light-weight, small-to-medium sized teams
- Work on things that really matter every day
- Get the most possible value out of every development week
- Takes commonsense principles and practices to extreme levels.
Feature Driven Development (FDD)

**Startup Phase**
- Develop an Overall Model
- Build Features List
- Planning

**Construction Phase**
- Design by Feature
- Build by Feature

Source: [http://featuredrivendevelopment.com/](http://featuredrivendevelopment.com/)
SCRUM

- Commonly Used to Enhance Existing Systems
- Feature Backlog
- 30 Day Sprints
- Daily Team Meeting

Source: http://www.controlchaos.com/
MSF for Agile Software Development

- Adapted from the MSF’s Spiral / Waterfall Hybrid

- Product definition, development and testing occurs in overlapping iterations

- Different iterations have a different focus
Let’s Look at an Integrated Process

Making Agile Trustworthy

Iteration 0
- Project Setup Plan

Iteration 1
- Plan Develop & Test Feedback

Repeat as needed
- Plan Develop & Test Feedback

Iteration n
- Develop & Test Release Product
Project Roles

- Product Manager / Customer
- Program Manager / Coach
- Architect
- Developer
- Tester
- Security Adviser
Project Setup

- **Education & Training** *(include Security)*
  - Developers
  - Testers
  - Customers

- **User Stories / Use Case Development**

- **Architecture Decisions** *(spikes)*
Release Planning

■ User Stories / Use Cases Drive…
  ▸ Acceptance Test Scenarios
  ▸ Estimations may affect priorities and thus the composition of the release
  ▸ Inputs for Threat Modeling
  ▸ Security Testing Scenarios

■ Finalize Architecture & Development Guidelines
  ▸ Common Coding Standards *(include security)*
  ▸ Conduct Initial Threat Modeling (assets & threats)
  ▸ Designer’s Security Checklist
Iteration Planning

- 1-4 Weeks in Length (2 weeks is very common)

- Begins with an Iteration Planning Meeting
  - User Stories are broken down into Development Tasks
  - Developers estimate their own tasks
  - Document the Attack Surface (Story Level)

- Never Slip the Date
  - Add or Remove Stories As Necessary
### Anatomy of a 2 Week Iteration

<table>
<thead>
<tr>
<th>Day 1:</th>
<th>Days 2 &amp; 3:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Iteration Planning Meeting</td>
<td>- Architectural spikes</td>
</tr>
<tr>
<td>- Developers signup for tasks</td>
<td>- Agile Modeling</td>
</tr>
<tr>
<td></td>
<td>- Attack surface &amp; Threat Modeling</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Days 4 - 9:</th>
<th>Day 10:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Developers complete tasks</td>
<td>- Iteration close out</td>
</tr>
<tr>
<td>- Testers implement automated acceptance tests</td>
<td>- Security testing</td>
</tr>
</tbody>
</table>

**Day 9:**
- Security Code Review
Executing an Iteration

- Daily Stand-ups

- Continuous Integration
  - Code Scanning Tools
  - Security Testing Tools

- Adherence to Common Coding Standards and Security Guidelines

- Pair Programming
  - New Features, Refactoring, Hazardous Components

- Developer’s Checklist
Stabilizing a Release

- Just like any other iteration

- Schedule Defects & Vulnerabilities based on customer priorities

- Final Security Review (FSR)
Challenges & Compromises

- Balance of Code Review vs. Pair Programming
- SDL Techniques practices in small doses throughout the duration of the project
- Threat Modeling performed against a moving target
Can We Be Both?

- Communication
- Simplicity
- Feedback
- Courage
- Trustworthy
Book Resources

- Extreme Programming Explained: Embrace Change, Kent Beck, Addison Wesley

- Planning Extreme Programming, Kent Beck and Martin Fowler, Addison Wesley

Article Resources

- The New Methodology, Martin Fowler

- The Trustworthy Computing Security Development Lifecycle, Steve Lipner and Michael Howard

- Survey Says: Agile Works in Practice, Scott Ambler

- SCRUM Development Process, Ken Schwaber, Advanced Development Methods
Web Site Resources

- http://www.agilealliance.org
- http://www.xprogramming.com
- http://www.featuredrivendevelopment.com
- http://www.controlchaos.com
- http://msdn.microsoft.com/vstudio/teamsystem/msf/msfagile