Agile & DevOps Security & Audit

John Tannahill, CA, CISM, CGEIT, CRISC, CSX-P
jtannahi@rogers.com

Areas of Coverage

- Agile Security & Control
- DevOps Security & Control
- Audit Tools & Techniques

Manifesto for Agile Software Development - Values

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Note: See Agile Principles

3

Agile Software Development

- Understand Agile Roles (e.g. Stakeholder, Product Owner, Architecture Owner, Team Lead, Team Members)
- Methodology Overview
  - Prioritized List
  - Iterations (e.g. 2-4 weeks)
  - Release

4

Agile Planning

- Release
- Iteration
- Daily
  - Yesterday I did ...
  - Today I'm going to ...
  - Roadblocks are ...

5

Agile Approaches & Methodologies

- Scrum
- eXtreme Programming (XP)
- Kanban

6

Scrum Projects

- Kick-off Meeting
  - Project Backlog
  - User Stories
- Sprints (1-3 weeks iterations)
- Sprint Lifecycle
  - Planning
  - Execution
  - Review
  - Rinse and Repeat

7

Roles

- Scrum Master
  - Team Leader / Facilitator
- Product Owner
- Technical Lead
- Developers
- Tester

8

User Stories / Requirements

- As a [user],
- I want to [do this thing],
- So that I can[accomplish this goal]

- Prioritized => Project Backlog
- Feature Estimation (how long user story will take)

- Requirements
  - User Story
  - Acceptance Criteria
  - Tasks to implement the story

9

Agile / DevOps Tools

- Post-It Notes / Whiteboards
- Wikis / Sharepoint
- Maven
- Subversion
- Git
- CVS
- SVN
- Hudson
- Jenkins
  - OWASP ZAP Plugin
- Docker
- Issue Tracking
  - Bugzilla
  - Redmine
- Monitoring
  - Nagios
  - InfluxDB
  - Log Management
  - Sensu

10

Other Considerations

- Planning Poker
- Velocity
- Test Driven Development (TDD)
- Continuous Integration

11

Agile Security & Control

- Agile Risk Management
- Threat Modelling
- Agile Life Cycle Controls
- Building Security into Agile (SecDevOps)
- Microsoft Security Development Lifecycle (SDL)
- Security Stories
- Security Verification
- Key Control Practices

12

Agile - Key Practices

- 1. Start with Agile guidance and an Agile adoption strategy.
- 2. Enhance migration to Agile concepts using Agile terms
- 3. Continuously improve Agile adoption at both the project level and organization level.
- 4. Seek to identify and address impediments at the organization and project levels.
- 5. Obtain stakeholder/customer feedback frequently.
- 6. Empower small, cross-functional teams.
- 7. Include requirements related to security and progress monitoring in your queue of unfinished work (the backlog).
- 8. Gain trust by demonstrating value at the end of each iteration.
- 9. Track progress using tools and metrics.
- 10. Track progress daily and visibly.
- *(Source: U.S. Government Accountability Office, Effective Practices and Federal Challenges in Applying Agile Methods)*

13

Microsoft Security Development Lifecycle (SDL)

- Training
- Requirements
- Design
- Implementation
- Verification
- Release
- Response

14

Security User Stories

- Threat model (on-going)
- Define abuse user stories
- Security features
- Security acceptance criteria

15

Key Control Practices

- Threat Modelling
- Security Stories
- Configuration and Patch Management Tasks
- Daily Tests
- Every Sprint Tests
- Vulnerability Assessment
- Penetration Test

16

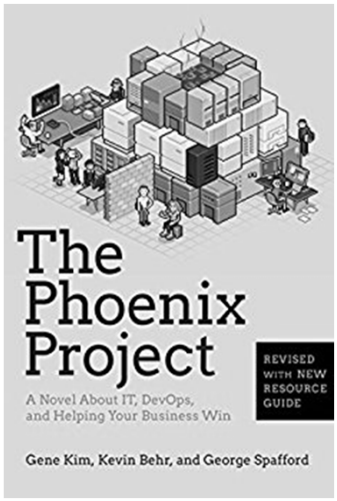
DevOps Security & Control

Introduction to DevOps

- DevOps Principles
- Automation
- Configuration Management
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Continuous Monitoring
- SecDevOps

18

The Phoenix Project



19

DevOps Resources (isaca.org)

- DevOps Overview
- DevOps Practitioner Considerations

20

DevOps Principles

- Short for Development and Operations (original concept)
  - E.g. Includes quality assurance, testing, security and release management
- Builds on Agile Development Concepts
- Continuous Delivery Pipeline

21

Automation

- Developer Environment
  - Development Tools
    - Maven
    - Grunt
  - Deployment
  - Production-Like Environment
    - VM
    - Docker
    - Cloud

22

Automation

- Revision Control System
  - Continuous Integration
  - Git
- Build Server
  - Jenkins
- Code Repositories
  - GitHub, GitLab
- Package Managers

23

Automation

- Test Environments
  - Integration
  - Staging
- Release Management
- Deployment / Configuration Management
  - Puppet, Chef, Salt, Ansible
  - Vagrant /Virtualization
  - Docker
  - AWS / Azure

24

Continuous ...

- Integration
  - Developers regularly integrate work with that of the rest of the developers on their team and then test the integrated work
- Delivery
  - Automate deployment software to testing, system test, staging, and production environments
- Deployment
  - E.g. Docker Containers and Images
  - Orchestration
- Monitoring
  - Data and metrics to stakeholders about applications at different stages of delivery cycle

25

SecDevOps

- Integrate security with business, development and operations
- “Shift security left”

26

10 Important Controls (DevOps Practitioner Considerations –isaca.org)

- Automated software scanning
- Automated vulnerability scanning
- Web application firewall
- Developer application security training
- Software dependency management
- Access and activity logging
- Documented policies and procedures
- Application performance management
- Asset management and inventorying
- Continuous auditing and/or monitoring

27

DevOps Tools

- Integrated Development Environment (IDE)
- Issue Tracker
- Version Control / Source Code Management
- Build Automation
- Continuous Integration (CI) Server

28

Audit Approach

- Based on Agile Methodology
- Understand Agile artifacts and map to audit evidence requirements
  - Security and Control Stories
- Understand SecDevOps
- Understand tools / function
- See control selection: DevOps Practitioner Considerations and related assessment criteria

29

Audit Checklists

- ISACA Audit / Assurance Programs

30