Integrating Security in Agile projects

Elena Kravchenko
Efrat Wasserman
Secure Development vs. Agile Development
Secure Agile Development

We believe **Agility** and **Security** are on the same **Team**!
Introduction

Elena:

• HPE Software **Security Lead** for HPE’s Application Delivery Management (ADM) Business Unit
• 25+ years of software engineering, last 4 years in product security
• MSc in Applied Mathematics from Leningrad State University

Efrat:

• Senior **Program Manager** in HPE SW StormRunner Load (SRL) *
• 17+ years in Software industry, out of which 9 years as a Program manager
• BSc in Computer Science and Mathematics
• MBA in Business Management and Marketing
What we will discuss today

- Challenges
- Secure development lifecycle management (SLM)
- Agile development practices
- Planning and coordination
- Practical tips
HPE ADM organization and portfolio

- **12** Products
- **5** Countries
- **~400** Developers
SLM framework

- Education
- Requirement and planning
- Design
- Implement
- Verify
- Release
- Response
- Deploy


Planning | Design | Dev | QA | Feature Freeze | Code freeze | GA

Freeze
Our product overview

- Light weight web base application for performance testing in the cloud
- 5 development teams, ~50 people
- 6 versions a year (every 2 months)
- SaaS product, continues delivery, hotfix

Persona: professional performance engineers
Managed in Agile methodology
7+ groups collaboration
Smart, Simple, Scalable
Agile development practices

- Ceremonies
- Minimal shippable product (MSP)
- Prioritized backlog flexible to change
- Strict heart beat (HB)
- Tool for agile project management
- High automation coverage

- No Sprints
- 1 HB = 1 Release = 1 Sprint
- No heterogenic scrum teams
- No scrum master (Feature leader role added)
- Additional untraditional ceremonies
- Additional unclassical configurations in the project management tool
SLM adjustments for Agile project

- **Education**
- **Requirement and planning**
- **Design**
- **Implement**
- **Verify**
- **Release**
- **Response**
- **Deploy**

**Steps:**
- Content review
- Feature based TM
- Design Review/Mitigation
- Security risk assessment
- Sign Off

**Cycle:**
- Continuous iteration
- Timelapse indicator
External security governance aspects

Security expertise:

- Application security
- Penetration testing

Main checkpoints embedded in process:

- Content review (*Security impact* labeling)
- Design review (Feature based *threat modelling*)
- Security assessment (*Penetration testing*)
Security Impact

Any content level:
- Theme, Epic, Feature, User Story etc.

Know the product:
- Security controls implemented in the product
- Data managed by the product

Ask the feature lead:
- Which security controls are affected?
- Which data types are affected?
- Which new risks might be inserted?
The **Open Web Application Security Project (OWASP)** is a worldwide not-for-profit organization focused on improving the security of software.

The **OWASP Top Ten** represents a broad consensus about what the **most critical** web application **security flaws** are.

[https://www.owasp.org/index.php/Main_Page](https://www.owasp.org/index.php/Main_Page)
The OWASP Top Ten - 2013

A1 Injection
A2 Broken Authentication and Session Management
A3 Cross-Site Scripting (XSS)
A4 Insecure Direct Object References
A5 Security Misconfiguration
A6 Sensitive Data Exposure
A7 Missing Function Level Access Control
A8 Cross-Site Request Forgery (CSRF)
A9 Using Components with Known Vulnerabilities
A10 Unvalidated Redirects and Forwards
Security Impact defined for every feature:

- Complete release security status
- Accurate documentation of concerns and decisions
- List of features for design review
- Prioritized backlog for security assessment
Penetration testing flavor:

- Think as an attacker
- Know product as a developer
- Work according to time limits
- Work in controlled environment
- Get all required support from development team
Best practices

Processes:

• Security assessment cycle is part of every release **timeline**
• **Dedicated security environment** preparation with predefined data
• Security bucket is allocated for development teams in **every release planning**
• Security criteria is a part of the **release criteria**
• **Status and risks** reviewed as part of all project management meetings, including **retrospective**
• Project **ceremonies**
Ceremonies:

• Bi weekly sync meetings:
  • Content review
  • Design reviews
  • Security assessment review and summary
• Demo and handover training every release
Best practices

Tools:

• **Manage** security content as part of the general release content
• **Audit** every security task in project management tool
• Additional **configurations** in the project management tool
  • Item name **prefix**: [Security]
  • Tags: **Security Impact** and **Security Comments**
  • Security defect **template**
Roles

Security champion
- **Tactical** technical focal point
  - Product insights
  - Technical demonstration
  - Code referee
  - Security ambassador ("double agent")

Architect
- **Strategical** technical focal point
  - Design review
  - Product Road map
  - Architecture adjustments
  - Provide balance

Project manager
- **Orchestrator** and key stakeholder
  - Processes leadership
  - Meetings management
  - Tasks auditing
  - Team engagement
  - Continuous improvement
Tips

• **Know** your partner
• **Drop** the text book definitions! Take the **suitable** agile flavor and **optimize** it
• **Cooperate** and back up one another
• Work **iteratively** on basis of **retrospectives**
• Meet & **Sync periodically** either via calendar invite or “coffee tests”
• Use tools to **audit everything**, be an inseparable part of the project content and success criteria
• **Promote and formalize** integration of secure development into all agile projects in your organization
Don’t say “security doesn’t work with agile”

Be flexible, adjust yourself and your processes, find a good partner and you will succeed together!
Thank you!