Mastering Security in Agile/Scrum, Case Study

Anu Puhakainen & Juha Sääskilahti
Ericsson
Network Security Competence Hub
Presentation Outline

- Introduction
  - Agile Transformation background
- Case Study: Security in Agile
- Five takeaways
Introduction
From Waterfall to Agile

- Deliverable
- Sprint
  - Sprint Review Meeting
  - Team
  - Sprint Backlog
  - Sprint ~2 weeks
  - Daily Scrum Meeting
  - Scrum Master

- Product Owner
  - Product Backlog
  - Sprint Planning Meeting

- Scrum
  - 24h

- Risk Analysis
- Analysis of Security
- Requirements
- Security Design
- & Implementation
- Security Testing
- Security Maintenance
- & Patching

- Requirements & Analysis
- Design
- Coding
- Verification
- Operation

Deliverable
Agile Transformation

- Major R&D Agile Transformation
  - Ericsson Finland as forerunner ~500 R&D employees working in software development for mobile networks
- Not only process change – also a big cultural change!
From This
Through This
To This
CASE STUDY: Security in Agile
Background to the Case Study

- R&D Transformation Case linked to
  - TiViT Cloud SW Research Project initiated 2010
    - Multi-branched research, including Agile
- Problem statement for Security in Agile
  - Current Agile/Scrum models do not have security embedded
What have we researched until now?

- Agile Transformation – yes
- But ... How is Security embedded?
  - How to make sure products developed with agile/scrum/lean are secure?
- Develop good practice for global Ericsson R&D
  - Theory meets practice – or does it?
Starting Point - Risk Analysis (RA)

- Old methodology
  - Suited for product releases with relatively long interval
- Agile brings new requirements
  - More frequent product releases
  - More dynamic feature changes (short lead time)
Tangible outcome: New RA method

- Promises:
  - Minimal preparation work required prior to workshop
  - Workshop of ½ - 2 days for a full product
    - For new features, very quickly … 15min(?)
  - More fluid workshops; mind-maps instead of matrixes
    - More motivating for participants
    - Using xMind (but any mind-map is ok)
- Templates
- Iterated and experimented 10-15 times before outlining Agile RA methodology
Risk Management with Agile/Continuous Integration

- Business Level Risk Analysis – updated every time product backlog changes
- Technical Level Risk Analysis – every time sprint starts
- Validation of Risks
  - Every check-in
  - Every Product Release
What else has been achieved so far?

- Security awareness – one key learning
  - Security much more visible now
- Learning from other companies and organizations
  - Research consortium, SafeCode…
  - Don’t try this alone at home!
Next area to address in detail: Security Testing

- Objective to find a good working model:
  - Set 1: Every Check-In
  - Set 2: Every Scrum
  - Set 3: Every Epic/sprint
  - Set 4: Every Product Release to Customer
- Automate what you can
  - Some tests should not too be automated
Security Requirements Management in Agile

- Non-functional requirements (e.g., Security requirements) – challenge in Agile
- 2 fundamental problems
  - Which requirements to choose
  - How to formulate the chosen requirements into Agile User Stories
    - Negative user stories? – How to confirm by testing?
### Example

<table>
<thead>
<tr>
<th></th>
<th>As a(n) architect/developer, I want to ensure <strong>AND</strong> as QA, I want to verify use of controlled format string</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D</strong></td>
<td>Adhere to SAFECode’s Fundamental Practices for Secure Software Development for preventing format string issues.</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>Scan source code for such violations using code analyzer tools, e.g., Coverity.</td>
</tr>
<tr>
<td><strong>A/D</strong></td>
<td>Conduct false positive analysis of flagged issues.</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>Fix format string issues analyzed as confirmed.</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>Use fuzz testing tool to verify that no process/system crashes/hangs exist. If they do, fix them and re-run the tool.</td>
</tr>
</tbody>
</table>

- Minimize Use of Unsafe String and Buffer Functions
- Use Canonical Data Formats
- Use Static Analysis Tools
- Perform Fuzz/Robustness Testing

---

Software Security Guidance for Agile Practitioners  
www.safecode.org
And we continue with these as well

- Processes
  - Finalizing Process – security control points
    - How to add controls without sacrificing ’agile model’
- Organization
  - Who should have which competence?
- Measuring security...
  - No good metrixes for product security
Takeaways
How to Apply Security in Agile

- Apply security ‘agilely’
  - Bit by bit; no ‘one-big-shot’
  - Adjust on the fly, give room for iteration
- Allocate sufficient resources
- Take learnings from other companies
- Make use of existing material
For security in agile, define strategy for:

- Organization – Security Roles, Responsibilities
- Process – Security Control Points
- Security Requirements Management
- Risk Management
- Security Verification
Questions?