Agile Security

Tilmann Haak, XING AG, Germany
<tilmann.haak@xing.com>
@TilmannHaak

Agile Security

1. What is “Agile”?
2. Security within agile software development
3. Using agile methods within the security team

What is “Agile”?

- Influenced by “Lean Manufacturing” and “Lean Production” / “Lean Engineering”
  (based on Toyota Production System)
- Emphasis self-organization
- Empowers the team
- Intended to reduce or avoid “waste” and overburden
- Very popular in software development teams
- It’s about “flow” and “pace”
- Deliver quickly, respond to emerging requirements

Prejudices

- Agile is chaos
- Agile methods lead to insecure software
- Waste of time, esp. daily standups
- It's just a buzzword
- Too complicated
- It's just "sticky notes on the wall"
- Does not work
- Only works for small teams

Agile Manifesto (brief)

"…while there is value in the items on the right,
we value the items on the left more."

[…]

"We have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan"

http://www.agilemanifesto.org/

SECURE
AGILE SOFTWARE DEVELOPMENT

Scrum & Kanban

- There are two main methods:
  Scrum and Kanban
- Both methods could be combined, but I've not seen this so far
- Both have large communities
- Good literature available
- A lot of good articles and publications are freely available

Scrum

- Scrum uses iterations or “sprints” of usually one or two weeks
- Scrum is used by many teams in software development
- The team commits itself to a set of user-stories, issues, tickets for the next sprint
- Scrum uses time-boxing
- At the end of the iteration there has to be a new (working) release available
- Scrum has at least three defined roles: Product owner, Scrum master, and team

Kanban

- Kanban is often used by service or support teams
- No sprints
- Pull system: late process stages pull items from earlier process stages
- Work-in-progress limit
- Kaizen (continuous improvement): Incremental, evolutionary change
- Often employs a Kanban wall with cards for visualization
- Tasks have priorities
- Issues move forwards, never backwards

Waste? Muda?

Anything that doesn't add value from the customer's perspective.

- Unnecessary functionality or code
- Bureaucracy
- Slow communication (esp. internally)
- Unclear requirements
- Started, but unfinished tasks
- Everything that keeps your from doing your actual job

Security within agile development

- Every team works differently
- Many development teams are cross-functional (e.g. frontend, backend, UX, design, PO, QA, …)
- Interfering with a team's current iteration is always problematic
- Most agile teams are very productive, expect tons of stuff to review and test
- Learn agile vocabulary, e.g. technical debt, NFR, software entropy, grooming, t-shirt sizes

How to fail

- Waterfall security will fail (clash of cultures)
- Screw up the team's estimation by adding requirements after the grooming session
- Let the product owner decide whether to implement a security feature or not
- Be an "impediment"
- Ignore the gatekeeper function of the product owner
- Expect someone to read 50+ pages requirements

Requirements…

We tried to discuss BSI's guidelines for development of secure web applications with our software engineers.

"60 pages? Are you kidding? Do you really expect someone to read this?"

Well… Even the OWASP ASVS has 47 pages…

How to succeed

- Make sure your developers understand the value of security for their customers
- Think agile: use evil user stories, ab-use cases
- Consult and advice early to avoid impediments
- Have practical tips and solutions available
- Ensure proper security training for developers
- Involve developers: Hack sessions are quite popular (and they are mutually beneficial)
- Ignore the gatekeeper function of the product owner ;-)

Agile Security Toolbox

Evil User Stories

"As an evil user I want to get chewing gum without paying for it."

Ab-Use Cases

List of steps, usually interactions, between a malicious actor and a system to achieve a goal.

Security Stories

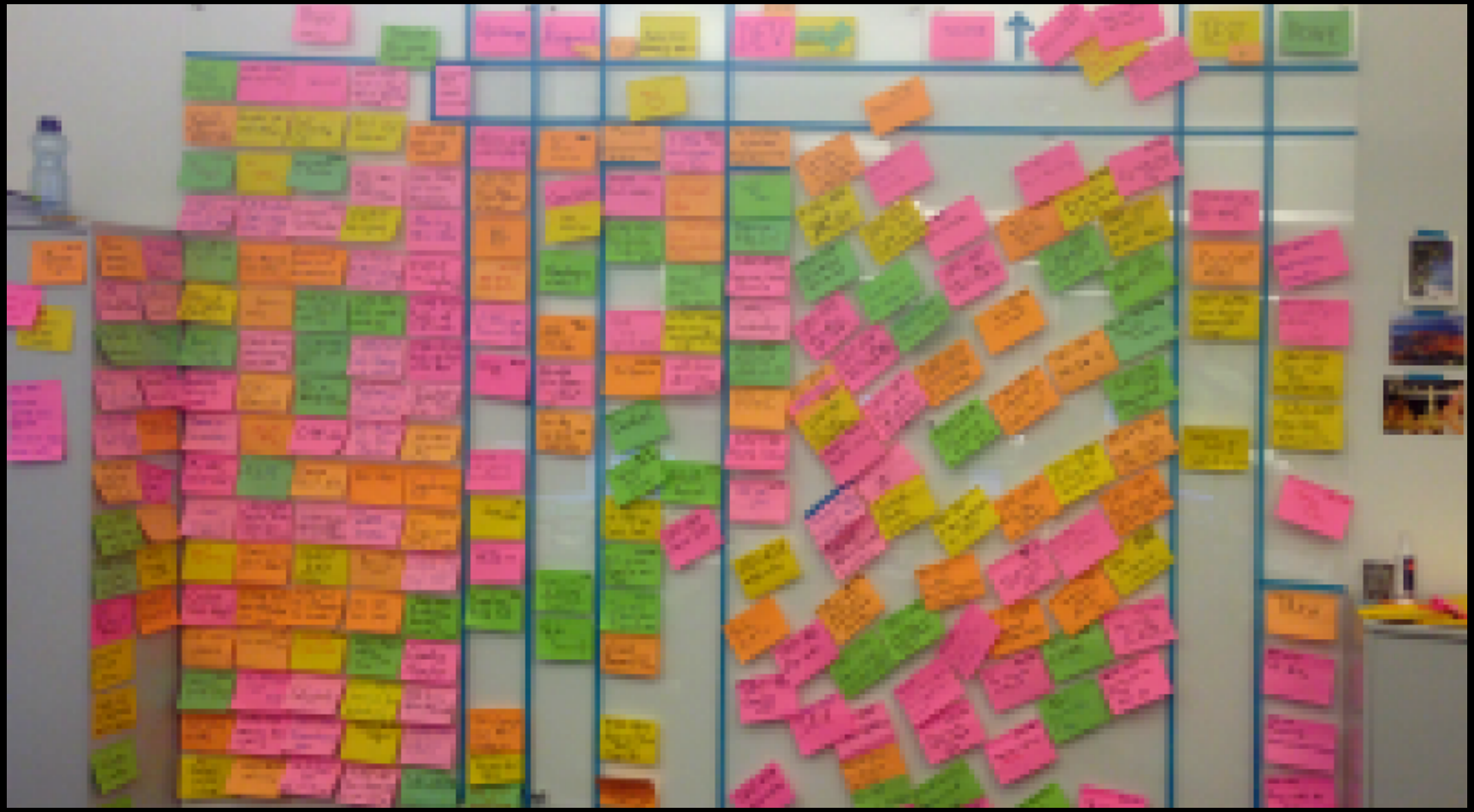
As an engineer I want to ensure that chewing gum is only given out after *correct* payment.

LIVE DEMO

VOLUNTEERS NEEDED

AGILE
SECURITY TEAM

Our Kanban Wall



Our Kanban Wall (2)

Agile Security Team

- Based on past two years experience
- XING is an agile company, all development, service and engineering teams work agile
- Introduced in the security team shortly after I've joined XING
- Enhanced visibility of our work within and outside the team
- Progress is observable
- Bottlenecks show up pretty quick

Problems

- Sync issue tracking system and Kanban wall
- Colleagues working remote
- Sensitive or confidential issues
- People taking photos, external visitors
- Too many long running tasks eat up wall space
- Issue multiplication, if several applications/ teams are affected (one card or 15?)

My personal observations (1)

- Easy to keep track of current issues
- Difficult to track issues that require work from several other teams
- Daily standup gives good feedback regarding progress of the tasks within the team
- It is clear what is next
- Impressive effect on managers ;-)
- The wall helps to explain what you are doing

A quick look at the wall

My personal observations (2)

- Good to actually get things done
- Easier to handle high workload
- Work-in-progress limit helps to focus
- Increased job satisfaction
- Good transparency
- Sensitive information on the wall is problematic
- Every Kanban wall is unique

Try it out!

- For security teams I recommend Kanban
- If possible get an experienced agile coach
- Use visualization and priorities
- Just start with what you have, improve and refine on the way

Done

I assume you have some questions or comments…

tilmann.haak@xing.com

@TilmannHaak

Further Reading

- Kanban, Successful Evolutionary Change for Your Technology Business; David Anderson
- Agile Software Development with Scrum; Ken Schwaber, Mike Beedle

Links

- Evil User Stories: https://www.owasp.org/index.php/Agile_Software_Development:_Don%27t_Forget_EVIL_User_Stories
- Abuse Case Models (McDermott, Fox; 1999): http://www.acsa-admin.org/1999/papers/wed-b-1030-john.pdf
- Limited WIP Society: http://www.limitedwipsociety.org/
- Manifesto for Agile Software Development: http://agilemanifesto.org/
- OWASP ASVS: https://www.owasp.org/images/5/58/OWASP_ASVS_Version_2.pdf
- BSI Guidelines regarding development of secure web applications (German): https://www.bsi.bund.de/DE/Publikationen/Studien/Webanwendungen/index_htm.html