Challenges of Software Security in Agile Software Development

Dr. Panayotis Kikiras
INFS133
March 2015

Agenda

- Lean Principles and Agile Development
- Usable Security
- Secure software development in Agile environment
- Prioritizing Security
- Conclusions

Lean Thinking in Agile Development

- Eliminate Waste — does it add end user value?
- Amplify Learning — validated learning
- Decide as Late as Possible — real options
- Deliver as Fast as Possible — fast learning
- Empower the Team — mastery, autonomy, purpose
- Build Integrity In — perceived and conceptual integrity
- See the Whole — simplify structure, optimize behaviour

Nordic Reading: http://www.fokkusu.fi/agile-security

Definition of Secure

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.

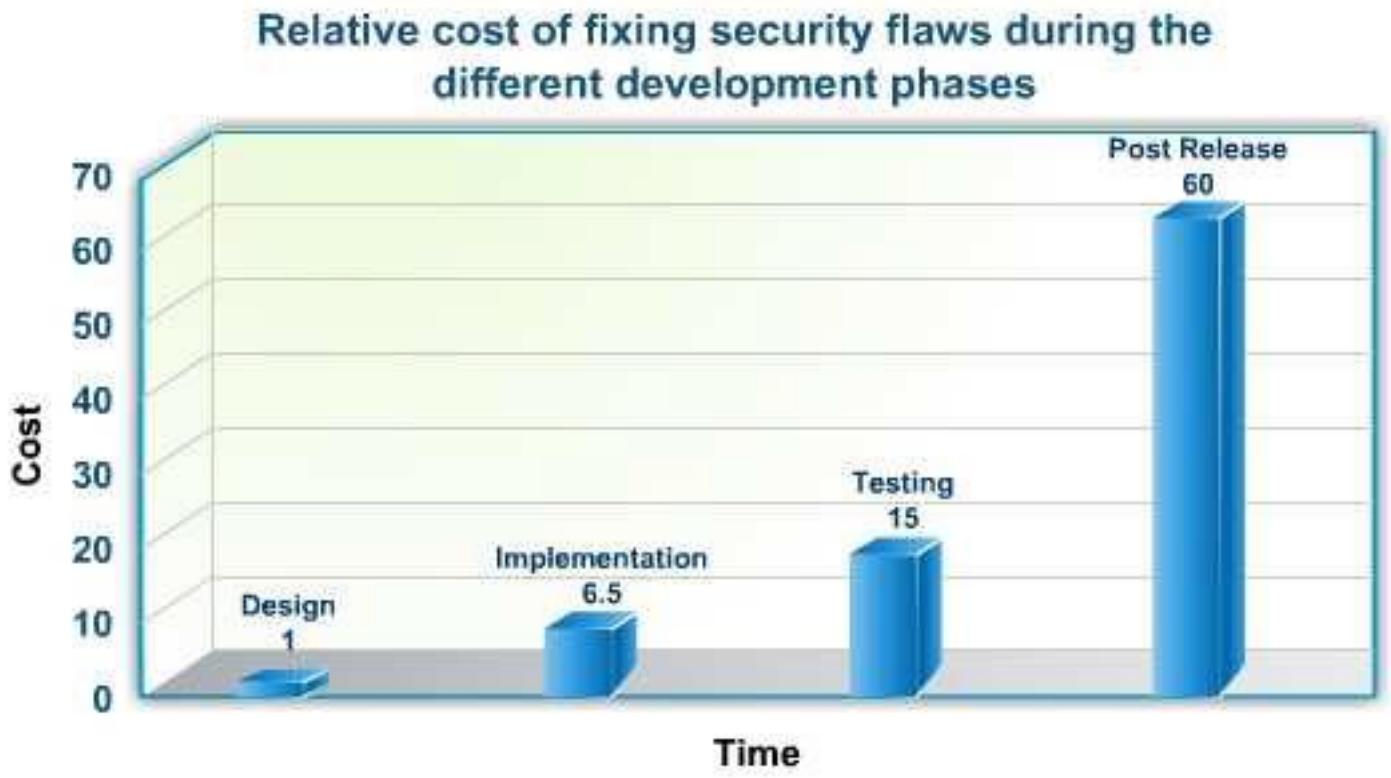
-- Source: Writing Secure Code
(Michael Howard, David LeBlanc)

Security is mainly a software problem

- Depending on the source, an estimated **70% to 92%** of security breaches result from vulnerabilities in software.
- Network Security Layer is adequately addressed (firewalls, IDS, IPS, Antivirus).
- A new star is rising though …

The end user

Incentives to Improve



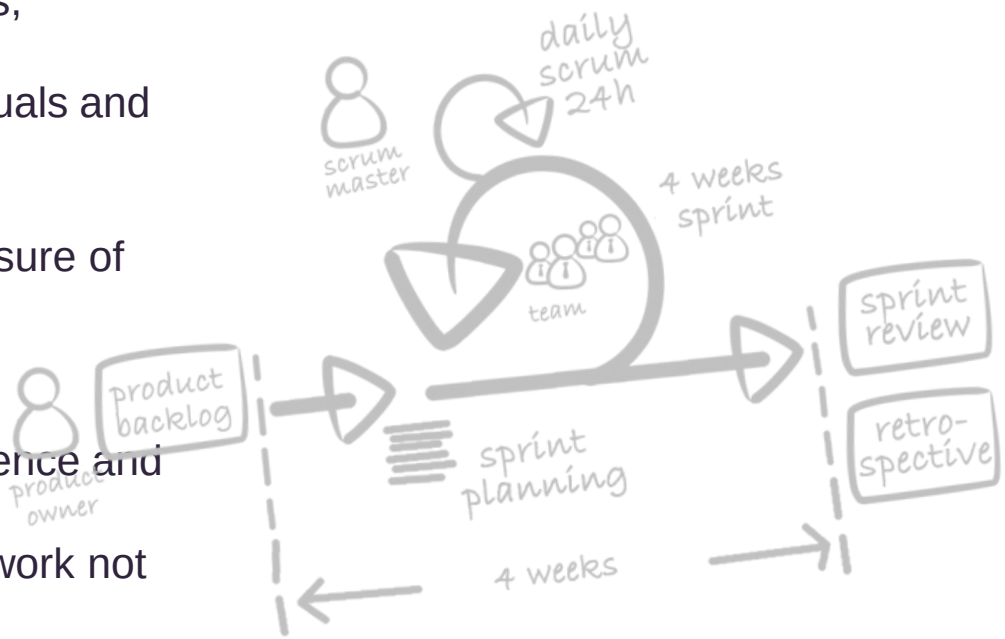
Source: Fundamentals of Secure Architecture – online available at https://knowledge.elementk.com

Where is Scrum now?

- Early and continuous delivery of valuable software

- Welcome changing requirements, even late in development
- Build projects around motivated individuals and trust them to get the job done.

- Working software as the primary measure of progress



- Continuous attention to technical excellence and good design
- Simplicity—maximizing the amount of work not done
- The best architectures, requirements, and designs emerge from self-organizing teams

- At regular intervals, the team reflects on, tunes, and adjusts its behavior

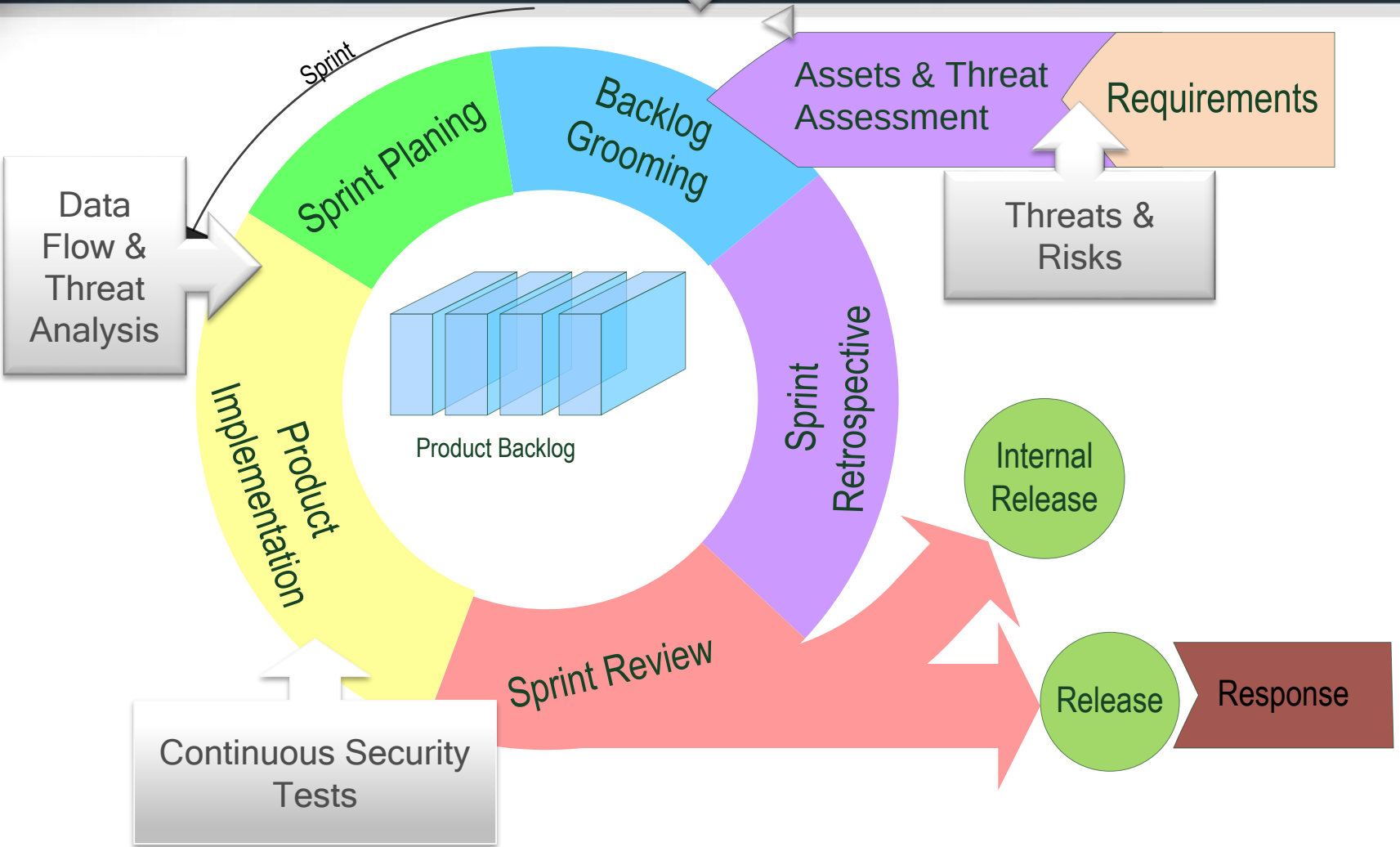
Where are you now?

- You trust that your teams are doing their best for security.
  - Do they?
- No specific care being taken in designing for security unless the customer requires that
  - Does this happens now?
- How a PO prioritizes security if it is not required by the customer?

Usable Security to Eliminate Waste

- Customers in general never ask for security directly
  - The product is expected to be secure
  - As a service to protect the business case
- Sometimes customers and security specialists are overexaggerating
- Teams should provide built-in solutions based on thorough Risk Analysis and Threat Assessment
  - UX: simplify structure, enrich functionality

Securing Scrum



Prioritizing Security: Risk – Adjusted Backlog

- Project Risks – security threats are like anti-value
  - If a risk occurs, takes time and resources away from activities that deliver value.
  - Therefore not only plan to deliver high value early but plan to execute risk avoidance and mitigation activities early too!
- Risk management great fit in Agile development
  - Through iterations we can tackle high-risk areas sooner than later
  - Deal with threats when still exists time and budget to work with them
  - Reduces the amount of effort invested in work that may end up scraped.

A security risk can be prioritized like any other feature

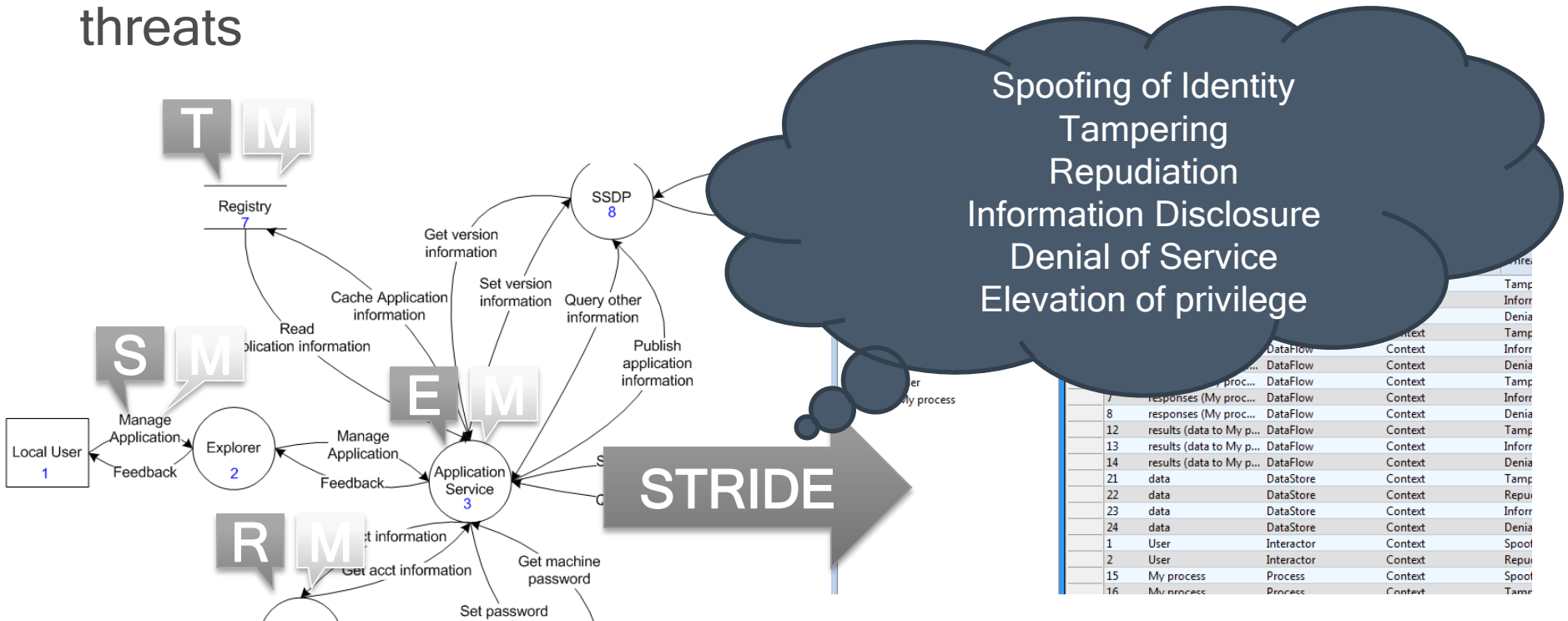
Prioritized Feature list with ROI Values		Prioritized Risk list -Ordered by Severity
Must	5000€	Risk i (Risk Impact (in€,points)XRisk Propability (in %))
Must	4000€	Risk 2 (8000€X50%=4000€)
Should	3000€	Risk 3 (3000€X25%=1500€)
Should	2000€	Risk 4 (2500€X25%=625€)
Could	1000€	Risk 5 (500€X20%=100€)

Risk-adjusted Backlog

Prioritized risk list	Prioritized requirements list	**Risk Adjusted Backlog**
Risk 1 (9000€X50%=4500€)	Requirement 1 5000	Requirement 1 5000
Risk 2 (8000€X50%=4000€)	Requirement 2 4000	**Risk 1 (9000€X50%=4500€)**
Risk 3 (3000€X25%=1500€)	Requirement 3 3000	Requirement 2 4000
Risk 4 (2500€X25%=625€)	Requirement 4 2000	**Risk 2 (8000€X50%=4000€)**
Risk 5 (500€X20%=100€)	Requirement 5 1000	Requirement 3 3000

Analyze the Dataflow

- Realization of User Stories whose acceptance criteria requires detailed look on potential threats
- Dataflow and STRIDE Analysis support identification of threats



Conclusions

- Also part of the Security Scrum process
  - Continuous Integration Testing
    - Explicit regression for acceptance criteria
    - Code Analysis (SAN 25)
    - Fuzzy Testing
  - Secure Coding Guidelines
- Adding Security to Scrum process is necessary and possible
  - Backlog Prioritization based on identified Risks
  - Modeling threats in user stories (business and technical)
  - Integrated security testing
- Incorporating experiences from Scrum teams (incl. explicit vs. implicit stories)