ThoughtWorks®

Lessons learned encouraging **Sensible Conversations** about Security

Lessons learned encouraging security thinking in software development teams

Software delivery consultant, with focus on infrastructure and continuous delivery

Developer / Devops

Technical leadership and advisory roles

Typically projects where you hear the terms 'Agile' 'Transformation' or 'Digital'

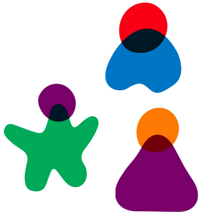
Public sector and private sector

Jim Gumbley

@jgumbley

The problem of security requirements in agile teams

What the functional requirements?



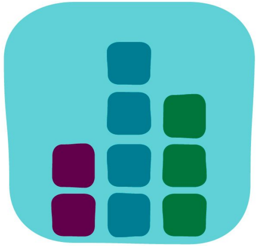
Product owners, Business Analysts, User Researchers

As a bookstore customer,

so that I can buy a book,

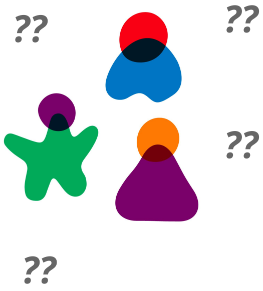
I need to be able to checkout

User Stories, Given When Then, INVEST, Feature Injection, BDD



Product owner prioritisation, MoSoCo. Trade off sliders

What the security requirements?



Experience in risk assessment ?

Knowledge of attackers and threats?

Resources for discovery process?

Technical knowledge of vulnerabilities?

Impetus for discovery process?

Established good practice?

Learning from public sector accreditation

CLAS Consultant

Expert with specific training and certification in information security assurance.

Working within mandatory Secure Policy Framework IS1/2 to assure information security risk management process

Role acts at arms length from development team, reporting to accreditor rather than project

RMADS Document

Documenting risk assessment following guidance in supplement to IS1/2 (Information Risk Management)

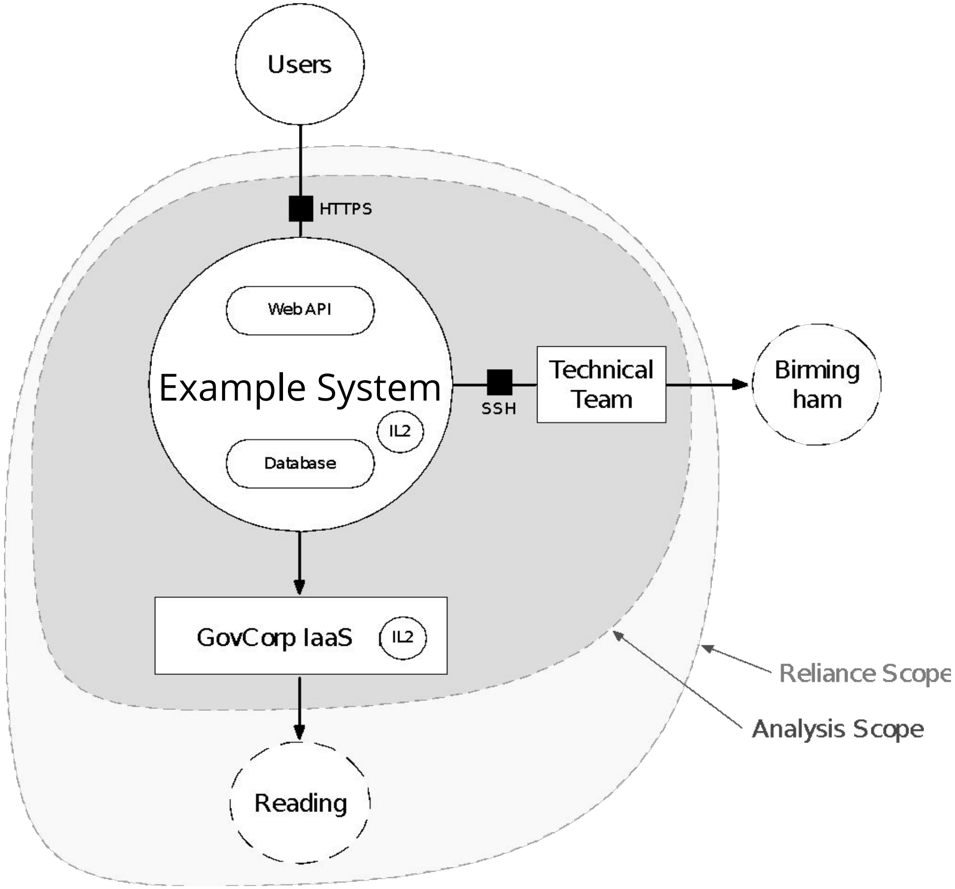
Penetration Testing

Requiring a penetration test - agreeing scope then assurance of mitigations and followups

SIRO sign off

Working with accreditor to present documentation set to Senior Information Risk Officer for approval to operate system

Scope Diagrams



Really helpful visual context on what we are protecting

Shows what is under our control, where we are relying on others

Shows source of threats to system

Shows target of threats to system

Shows structure and topology of these elements to assist in designing controls

Easy to draw up on a whiteboard and get shared understanding with the team

Risk statements

A **disaffected employee** who is a **directly connected administrator** deliberately compromises the **confidentiality** of the **customer database** having a potential **impact on the personal finances of many people**

Risk and impact statements could be negotiated with the product owner and the business

Controls derived from risk *- led to additional scope going into the backlog, additional design sessions with the delivery team*

Led to sensible conversations with CLAS consultant and the accreditor:

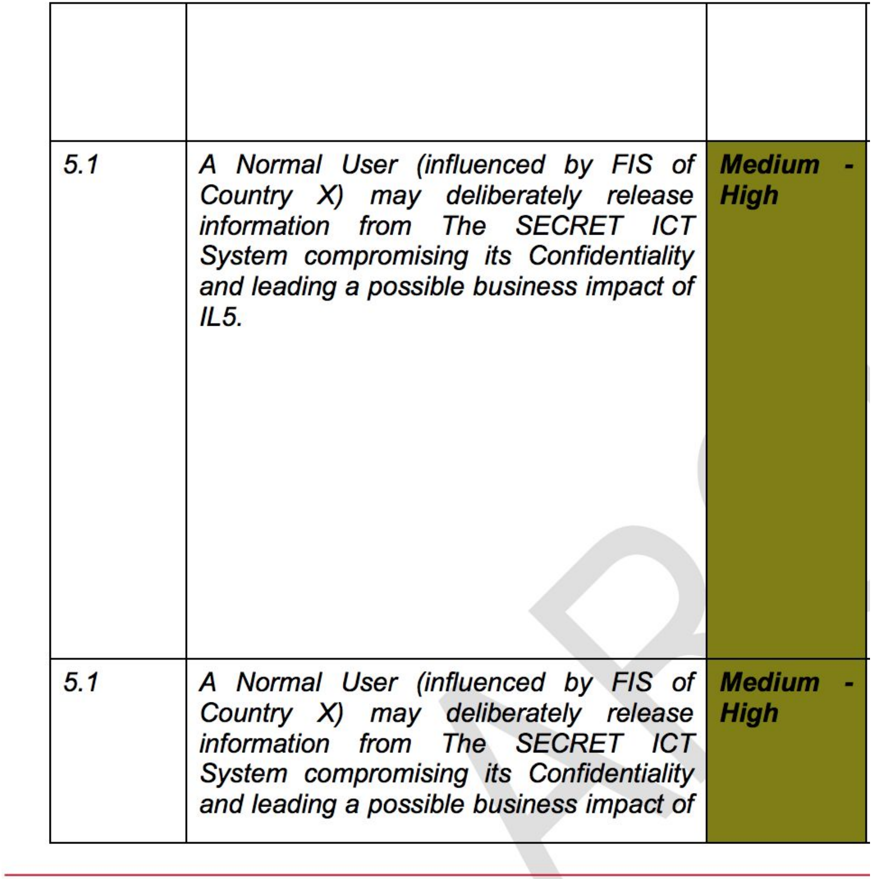
- *Is AES256 strong enough?*
- *Is 20 minutes enough for a session timeout?*
- *We don't think there's much we can do about keyloggers in libraries?*

Focused Expertise

CLAS consultants and accreditors brought a deep understanding of:

- *the organisation’s risk tolerance*
- *who was likely to attack*
- *how similar systems had been protected*
- *the technology of defense and attack*
- *business process and fraud*
- *network technology*
- *risk transfer*
- *evaluating cloud and SaaS / ISO 27001*

Critically an expert knowledge of the risk assessment process, which is not simple - able to do the deep thinking



5.1	*A Normal User (influenced by FIS of Country X) may deliberately release information from The SECRET ICT System compromising its Confidentiality and leading a possible business impact of IL5.*	***Medium - High***
5.1	*A Normal User (influenced by FIS of Country X) may deliberately release information from The SECRET ICT System compromising its Confidentiality and leading a possible business impact of*	***Medium - High***

RMADS document

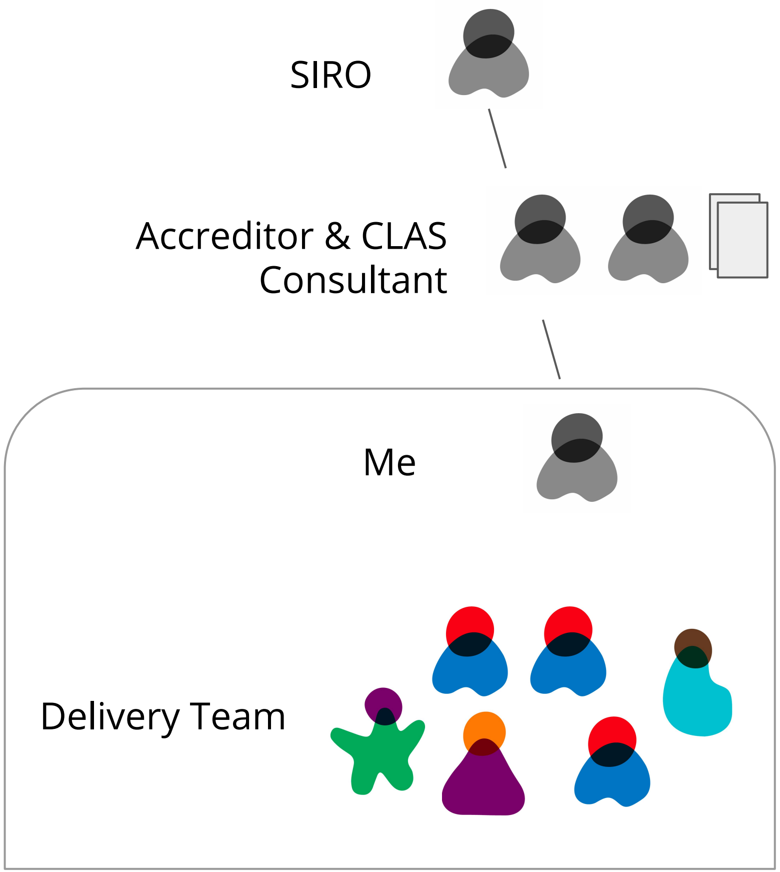
To have a record of the risk management process makes sense

RMADS difficult to comprehend without support from an expert

Can get too big (too many risk statements, hard to understand risk statements) and then hard to drive action

“The risk assessment process described is intended to stimulate thought about risk. It is not intended to simply generate paperwork” - IS1/2 Preface

Connection with delivery team



Given IS1/2 is an assurance process - arm's length from delivery team makes sense

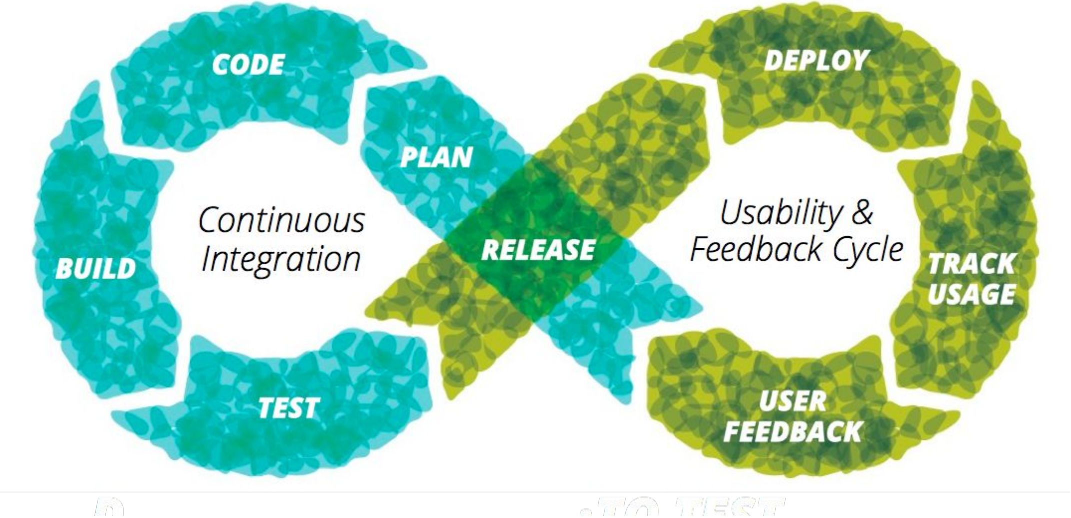
However, developers - although intimately involved in developing and designing the product felt isolated from the conclusions of the RMADS in day to day decisions

Marking the document OFFICIAL-SENSITIVE didn’t help, although unlikely developers would have been able to interpret it fully without help

It was a lot of work for me to manage both sides of the relationship and share context in open and honest way

Transferring good practice into agile delivery

Connecting it to agile software delivery: Update continuously



Agile delivery is based on team who work in iterations

We want the team to add security controls as they build - “baking security in”

Requirements continuously update to reflect changes in:

- *the needs of the organisation*
- *the threat it faces*
- *changes in the vulnerability of technologies and platforms*

Not enough experts for every team

There is a simple solution - adding an information security specialist to the delivery team

However

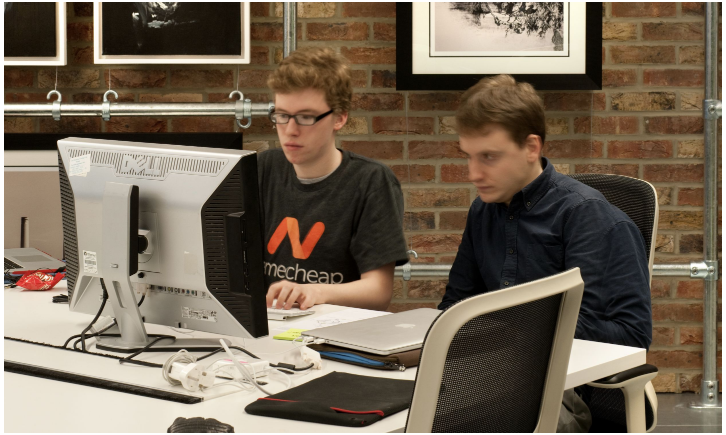
- *Need folks with good expertise with experience of / comfortable working in agile teams*
- *Skills shortages mean the economics won’t make sense for every team*

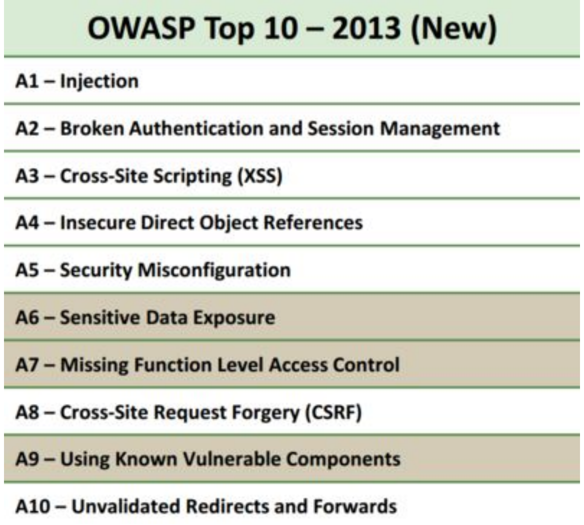
Trying not to reinvent the wheel

OWASP Top Ten

Pro bono project staffed with graduates, no sensitive data, not supporting critical processes

Wrote out cards with the OWASP top ten written on them and talked them through in a workshop with developers





OWASP Top 10 – 2013 (New)
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 Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards

Each developer took a card, or set of cards away to research:

- *Who or why an attacker might use that attack*
- *The technical mechanism and how to protect against it*

Each developer then wrote up story cards to control against the attack, and worked with product owner to prioritise.

Lots of learning! Saw a big improvement in the protections built into the system

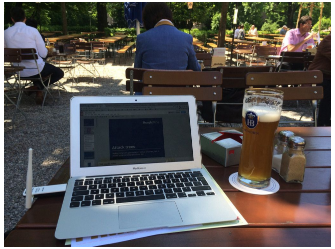
Just OWASP Top Ten likely too limiting for most projects. No risk assessment!

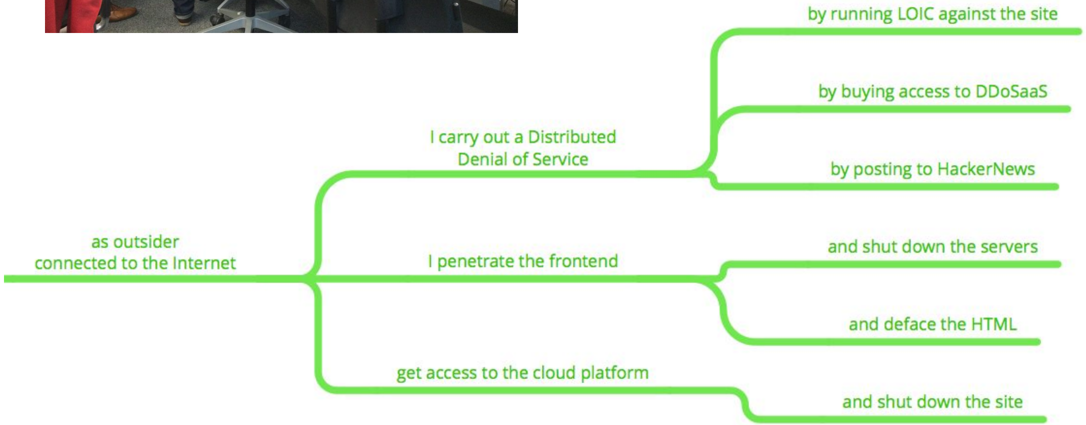
Building attack trees

Paired with a colleague with security expertise in Germany to support inception of a new platform for a client

Workshops with development team, more senior technical stakeholders, folks with business and product expertise

Delivered a set of risk prioritised attack trees with recommended controls to guide development





as outsider connected to the Internet
I carry out a Distributed Denial of Service
by running LOIC against the site
by buying access to DDoSaaS
by posting to HackerNews
I penetrate the frontend
and shut down the servers
and deface the HTML
get access to the cloud platform
and shut down the site

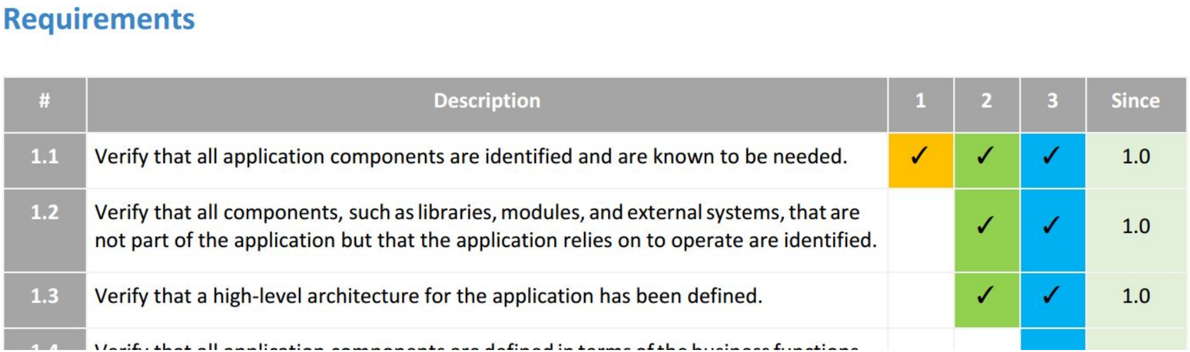
More art than science - much of the advice was derived from expertise rather than the attack tree format. Knowing threats, vulnerabilities, controls. Understanding risk.

Longer term: Output was hard to update and iterate on - ended up being a snapshot exercise rather than something the programme could remix

Application Security Verification Standard

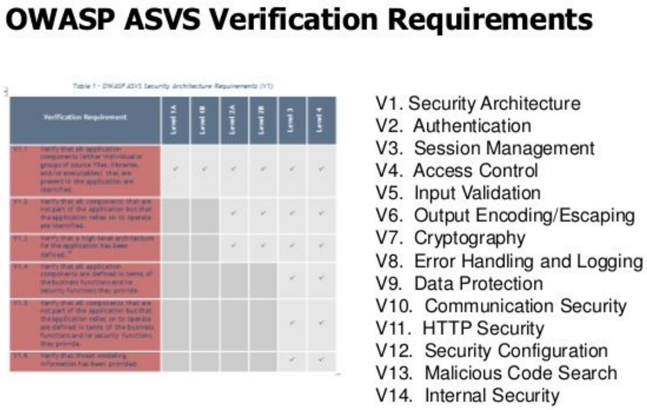
Working as a tech lead on a SaaS data analytics project for a telecoms company - no formal assurance team assigned.

Needed a good baseline to ensure we weren't 'missing anything'.



Requirements

#	Description	1	2	3	Since
1.1	Verify that all application components are identified and are known to be needed.	✓	✓	✓	1.0
1.2	Verify that all components, such as libraries, modules, and external systems, that are not part of the application but that the application relies on to operate are identified.		✓	✓	1.0
1.3	Verify that a high-level architecture for the application has been defined.		✓	✓	1.0
1.4	Verify that all application components are defined in terms of the business functions				



OWASP ASVS Verification Requirements

Table 1 - OWASP ASVS Security Architecture Requirements (V1)

	Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1.1	Verify that all application components (either individual or groups of source files, libraries, and/or executables) that are present in the application are identified.	✓	✓	✓	✓	✓	✓
V1.2	Verify that all components that are not part of the application but that the application relies on to operate are identified.			✓	✓	✓	✓
V1.3	Verify that a high-level architecture for the application has been defined.			✓	✓	✓	✓
V1.4	Verify that all application components are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.5	Verify that all components that are not part of the application but that the application relies on to operate are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.6	Verify that threat modeling information has been provided.					✓	✓

V1. Security Architecture
V2. Authentication
V3. Session Management
V4. Access Control
V5. Input Validation
V6. Output Encoding/Escaping
V7. Cryptography
V8. Error Handling and Logging
V9. Data Protection
V10. Communication Security
V11. HTTP Security
V12. Security Configuration
V13. Malicious Code Search
V14. Internal Security

Just a list of controls - no why - baseline approach - no risk assessment component.

Harder to discuss the 'why' with delivery team - "just because".

Harder to discuss with business and prioritise- relying on my own judgements which are not validated.

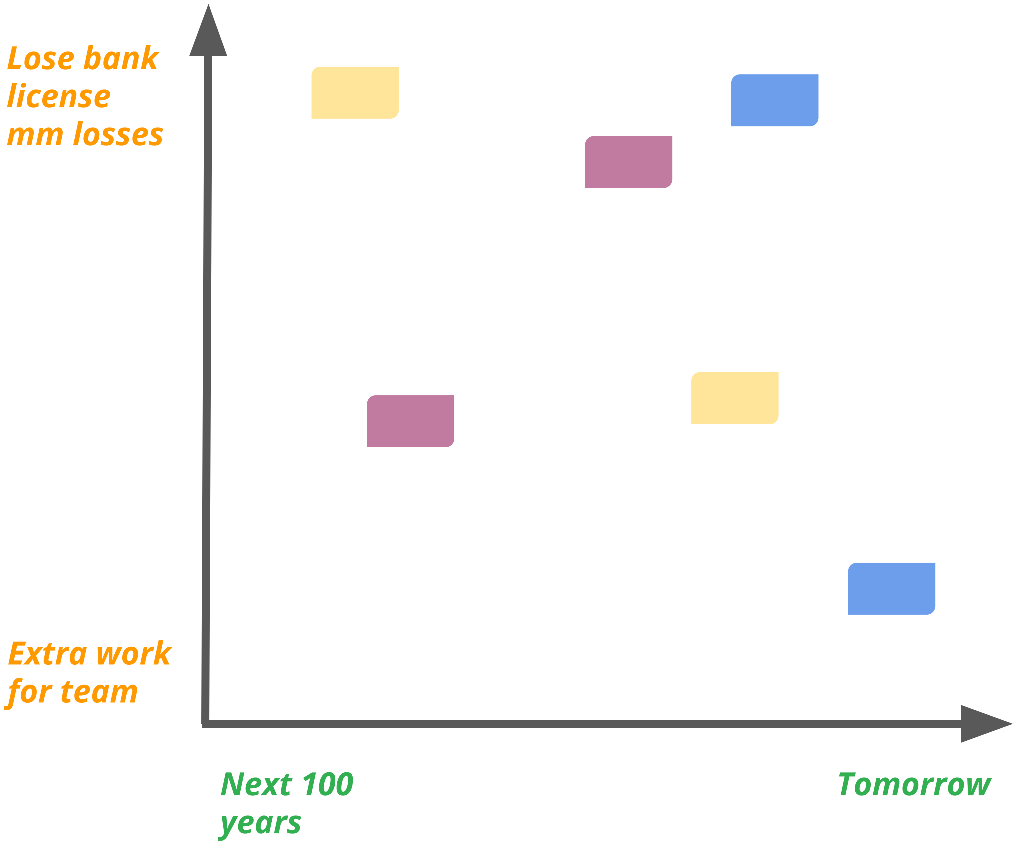
Risk mapping workshop

Awareness session in retail financial services institution which had introduced Agile into software development process

System was being accredited by bank information security group - wanted to connect with developers

- *Team + Security brainstormed attackers with motivations*
- *Ranked them visually via impact - specific to business*
- *Ranked them visually by likelihood - how long is it likely to be before that individual attacks*

Felt like a great session - improved awareness- did it connect with real work in backlog?



Microsoft's Escalation of Privilege Cards
OWASP Cornucopia

Threat Modelling via playing cards! Seemed perfect!

Carried out workshops with both sets of cards in our London Office for various projects. Sent them out to projects - folks tried it out in US and India also.

EoP Cards too Microsoft specific - lots of cards people didn't know how to relate or apply technical language to their use cases - again falling back on expertise

Cornucopia cards had a similar effect - lots of debate about the semantics of the cards

Not clear/explicit how to translate into outcomes