



Die Macht zur Veränderung: Security- Kompetenz im Team etablieren

Lars Hermerschmidt, REWE digital
Jan-Niclas Strüwer, Fraunhofer IEM
Benji Trapp, REWE digital

Heise DevSec, Köln

 **Fraunhofer**
IEM

REWE
DIGITAL



Lars Hermerschmidt

Product Owner Security Engineering, REWE digital

INDUSTRY

- Building software security programs since 2017
- Coach and trainer for software security
- Helping in DevOps transformation

RESEARCH FOCUS

- LangSec, eliminating injections
- Threat Modeling automation
- Evidence based software security programs
 - security capability model Security Belts

Contact Me!





Jan-Niclas Strüwer

Research Associate, Fraunhofer IEM

RESEARCH FOCUS

- Static code analysis and secure engineering
- Hierarchical KPI system to assess software security and quality
- Qualitative assessment of software dependencies

INDUSTRY BACKGROUND

- >6 years experience in applied research in the area software engineering
- Coach and trainer for software security

Contact Me!





Benjamin-Yves Trapp

Technical Product Owner, REWE digital

BACKGROUND

- Ehemaliger DevSecOps Engineer, Security Analyst und Cyber Defense Expert
- Nun unterwegs als Red Team Operator und Coach
- > 12 Jahre an Security Erfahrung

OTHER TOPICS

- Studium der Technische Informatik und Biotechnologie
- Erfahrungen in der Chemie-, Einzelhandel- und Banken/Versicherungsbranche
- Blogging über DevOps und Security
- Entwickeln von (Security) Tools und Malware

Contact Me!



Security ist schwierig

„Einfach einen Standard implementieren“

Es gibt viele Standards: ISO27001, OWASP SAMM, OWASP DSOMM, Security Belts

Security ist ein abstraktes Gut wie Umweltschutz

Engineers müssen Tools und Methoden kennen und anwenden können



Frederick Taylor

Security ist schwierig

„Einfach einen Standard implementieren“

Es gibt viele Standards: ISO27001, OWASP SAMM, OWASP DSOMM, Security Belts

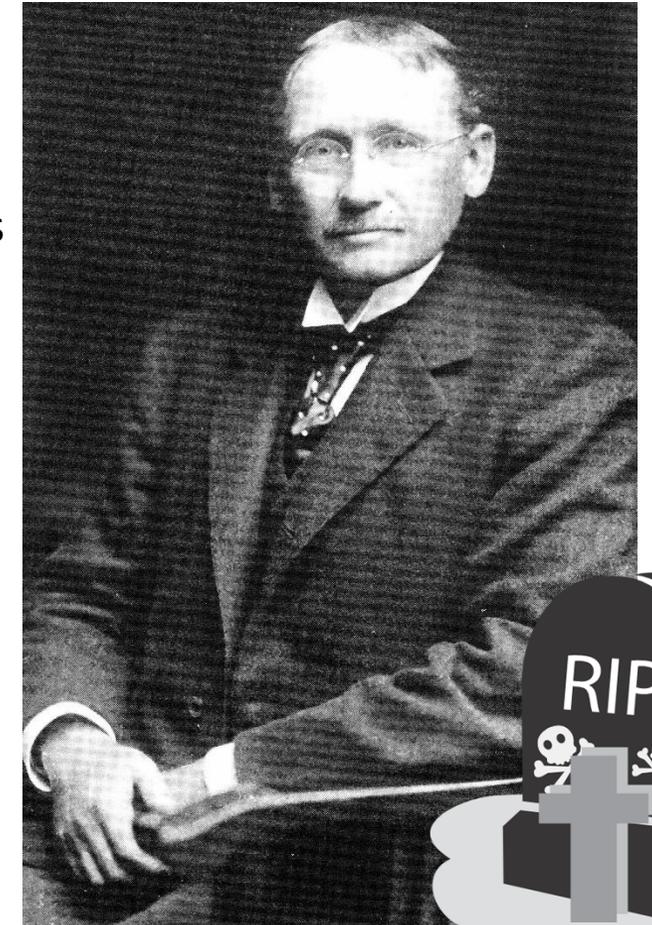
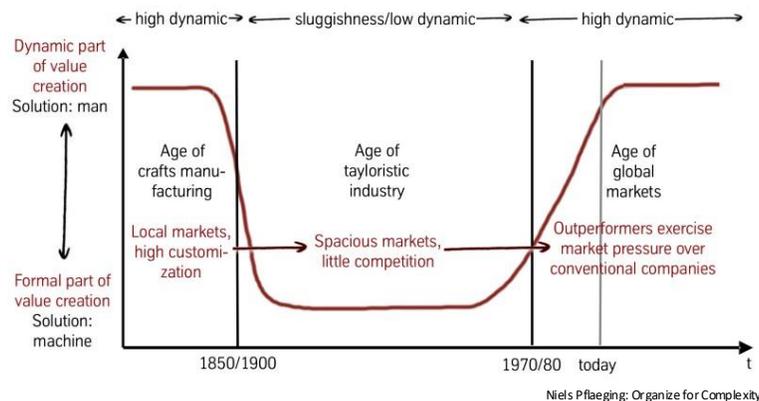
Security ist ein abstraktes Gut wie Umweltschutz

Engineers müssen Tools und Methoden kennen und anwenden können

„Teams machen einfach Security selber“: DevSecOps Organisation

Informationssicherheitsbeauftragte, Führungskräfte, und Product Owner können

Engineers nicht sagen wie Software Security funktioniert



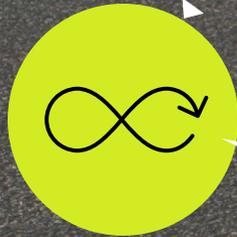
Frederick Taylor 1856 - 1915



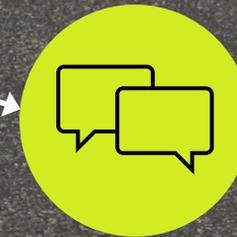
OWASP SAMM



**Security Belts
& Champions**



**Continuous Improvement
& Coaching**



**Experience Report
Internal & External Coaches**

Technical Agile Samman Coaching

Grundlage: Ensemble Programming aka Mob Programming

Rollen: Typist, Navigator, Coach

Coach

„Teaching from the back of the room“

beobachtet und gibt Hilfestellungen

schreibt keinen Code für das Team

Hält Learning Hours

Lernen durch tägliche Wiederholung

Ziel: Neue Gewohnheiten sind präsenter als Alte

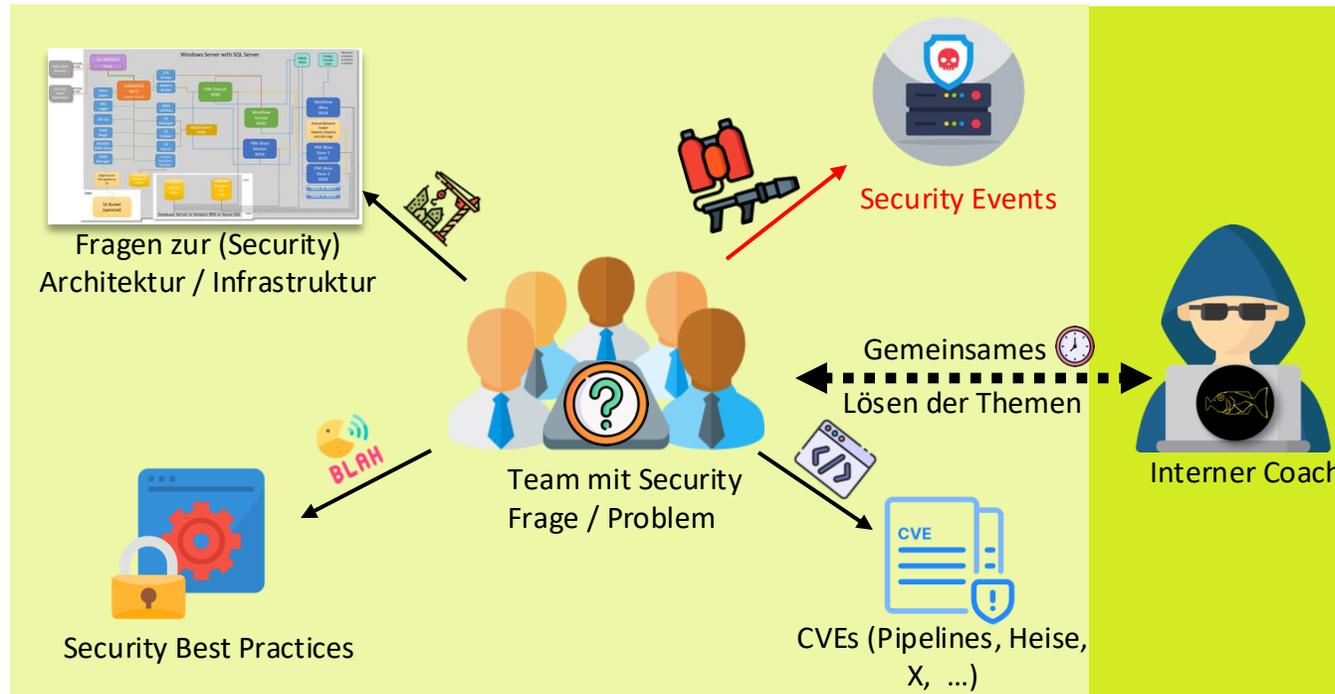
Chartering Workshop

Ziel: Coach und Team vereinbaren ein Lernziel und Zeitraum



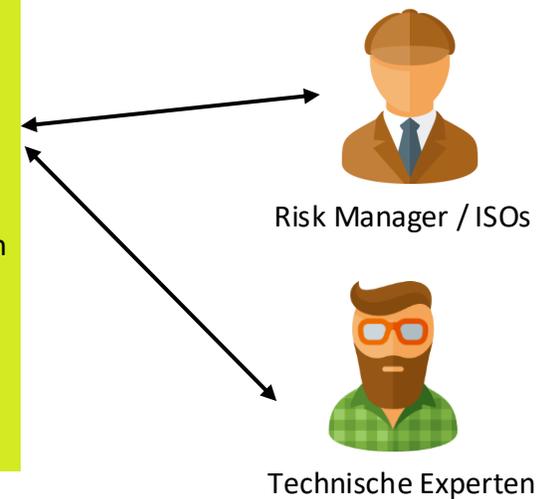
Erfahrungen als interner Coach

Ausgangslage



Ziel

- Wichtige Security Fragen/Probleme der Teams nachhaltig lösen
- Erstkontakt & Direkthilfe bei Security Events
- Vermittlung an Experten bei Bedarf
- Security Babel Fish



Erfahrungen als externer Coach

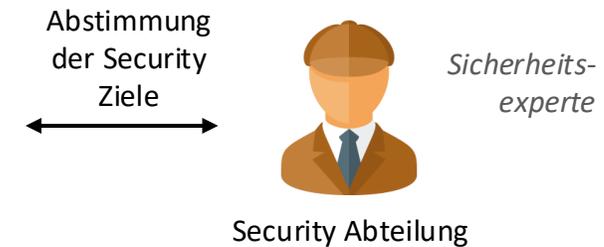
Ausgangslage



Cross functional Team ~10 Teammitglieder (devs, ops, architects, ba, tester, ...)

Ziel

- Security Aktivitäten mit Bezug zur täglichen Arbeit des Teams
- Kein one-size-fits-all approach und direkte Unterstützung für den Security Champion



Zeit für Real Talk

Aus dem Leben eines internen und externen Security Coaches

Entscheidungen im Gehirn

Gewohnheiten kann man nicht ändern, nur Neue lernen

System 1: Thinking Fast

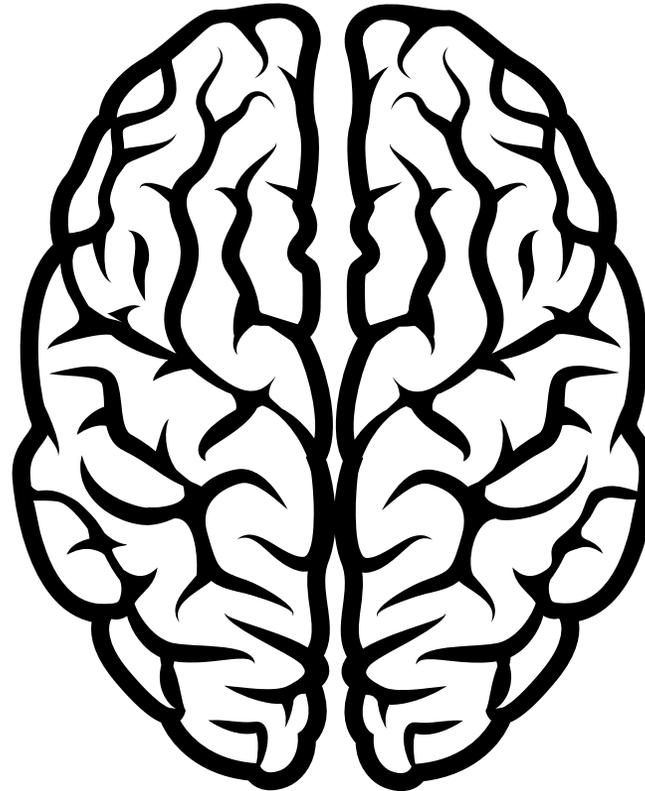
Gewohnheiten

Intuition

Unterbewusst

Autopilot

Reagieren ohne nachdenken



System 2: Thinking Slow

Rational

Ist anstrengend

Logisches Denken

Neue Herangehensweise

Neues Verhalten

Veränderungen – Interner Coach

Impact auf Security Architektur

- Coaching führte zu Veränderung der Security Architektur
- Reflektion über den Status Quo
- Umleiten der Probleme in konkrete Lösungen
- Schaffen einer gemeinsamen Kommunikationsbasis und Sprache
 - Wirkt inner- und außerhalb des Teams



IT-Training vs. Kata 型

IT Training

Wissen wird einmal präsentiert

Trainee wendet Wissen 1x an und bekommt kein Feedback vom Trainer, ob das gut oder richtig war.

Teile landen im System 2



Kata 型

Festgelegte Abfolge von Bewegungen ohne Gegner

Wird bis zur Perfektion geübt

Trainer gibt Feedback

Durch Wiederholung in System 1 verankert



Coding Kata

Aus Software Craftsmen Community

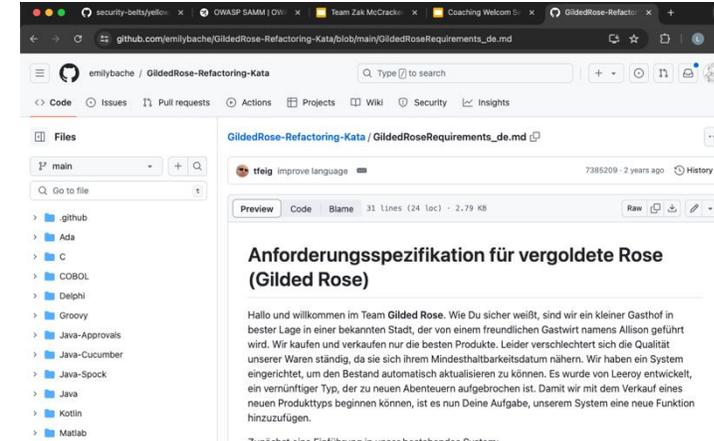
Abfolge von Engineering Schritten wie Refactoring und TDD

Ohne Gegner Produktiv Code

Wird bis zur Perfektion geübt

Annahme

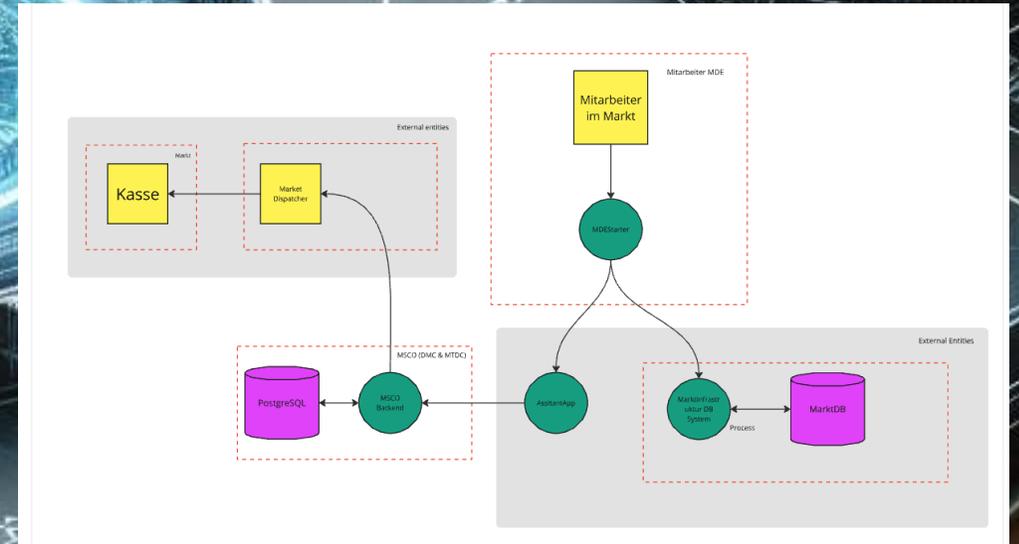
Katas verändern das Verhalten eines Teams



Threat Modeling

Interner Coach

- Initialzündung für Security Themen
- Diskussionsgrundlagen zum schlichten von Debatten
- Vereinfacht Kommunikation
 - Teamintern
 - Über Teamgrenzen hinaus
 - mit ISOs und Risk Manager



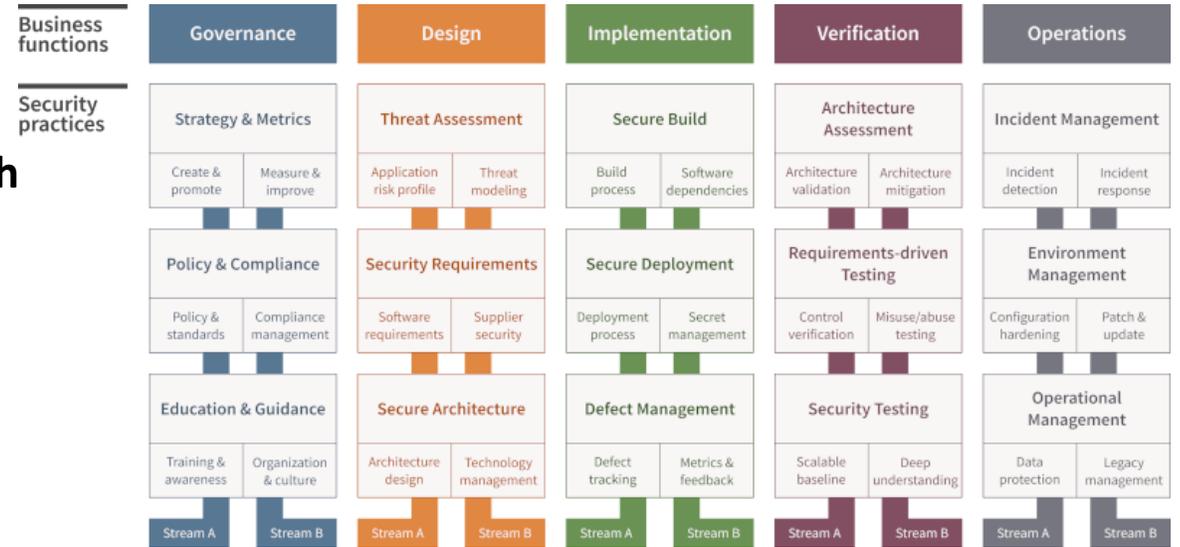
OWASP SAMM

Maturity Assessments

Security Experten führen Assessment des SDLC durch

„Gap to target“ wird gemessen

Teams bekommen die Aufgabe das Ziel zu erreichen



Annahme

Aufzeigen von Verbesserungspotentialen führt zu Verbesserung in den Teams

Erkenntnis

OWASP SAMM ist nicht für Engineers gedacht

- Engineers verstehen es nicht
- Hat keinen Team-Bezug

The screenshot shows a spreadsheet with OWASP SAMM assessment questions and answers. The questions are organized into categories: Patching and Updating, Data Protection, and System Decommissioning / Legacy Management. The questions are numbered 1 through 3, and the answers provide detailed descriptions of the required practices. For example, question 1 under Patching and Updating asks 'Do you identify and patch vulnerable components?' and the answer describes the need for an up-to-date list of components and a regular review of public sources for vulnerabilities. Question 1 under Data Protection asks 'Do you protect and handle information according to protection requirements for data stored and processed on each application?' and the answer describes the need to know the data elements processed and stored by each application. Question 1 under System Decommissioning / Legacy Management asks 'Do you identify and remove systems, applications, application dependencies, or services that are no longer used, have reached end of life, or are no longer actively developed or supported?' and the answer describes the need to manage customer/user migration from older versions for each product and customer/user group.

Security Belts

Gamification

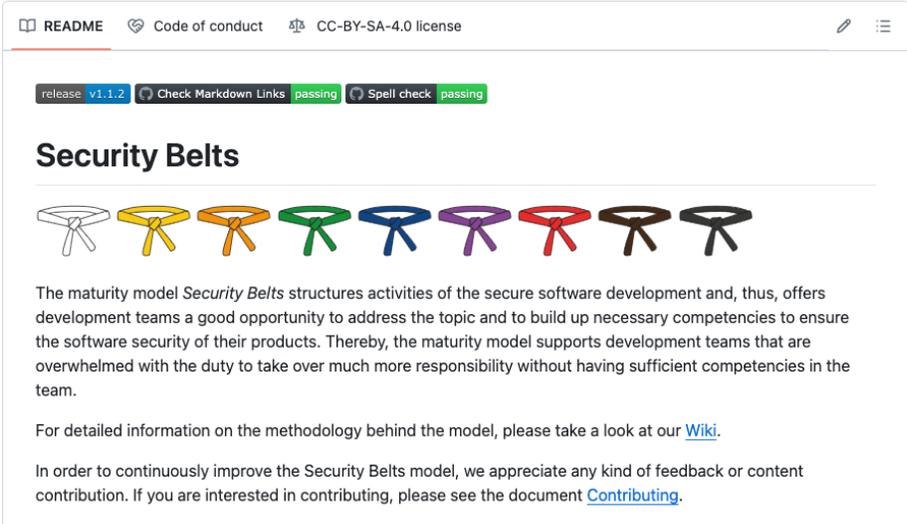
Maturity Modell strukturiert SSDLC Aktivitäten für Entwickler

Gürtel (wie beim Judo) sortieren Aktivitäten nach Schwierigkeitsgrad

Teams bekommen einen Gürtel, wenn sie alle Aktivitäten des Gürtels durchführen

Annahme

Gamification zwischen Teams führt zu Wettrennen um die Erreichung der Gürtel



release v1.1.2 Check Markdown Links passing Spell check passing

Security Belts



The maturity model *Security Belts* structures activities of the secure software development and, thus, offers development teams a good opportunity to address the topic and to build up necessary competencies to ensure the software security of their products. Thereby, the maturity model supports development teams that are overwhelmed with the duty to take over much more responsibility without having sufficient competencies in the team.

For detailed information on the methodology behind the model, please take a look at our [Wiki](#).

In order to continuously improve the Security Belts model, we appreciate any kind of feedback or content contribution. If you are interested in contributing, please see the document [Contributing](#).

<https://github.com/AppSecure-nrw/security-belts>

Security Belts



Neu Priorisierung einzelner Aktivitäten

Identifizieren was die Teams bereits tun und auf diese Aktivitäten aufbauen

Was macht ihr bisher für eure Security?



Externer Coach

Security Belts

Interner Coach

- Startpunkt für die technische und Kulturelle Reise eines Team in Security Themen
- Ad hoc Beratung von Security Champions in Bezug auf Belt-Aktivitäten
- Kontinuierliche Weiterentwicklung der Security Champions



⚠ Achtungspunkt ⚠

- Belt-Aktivitäten sind meist Individualleistung einzelner
- Lernen skaliert nicht auf das gesamte Team
- Was passiert wenn der Securitt Champion das Team verlässt ? → Security Maturity sinkt

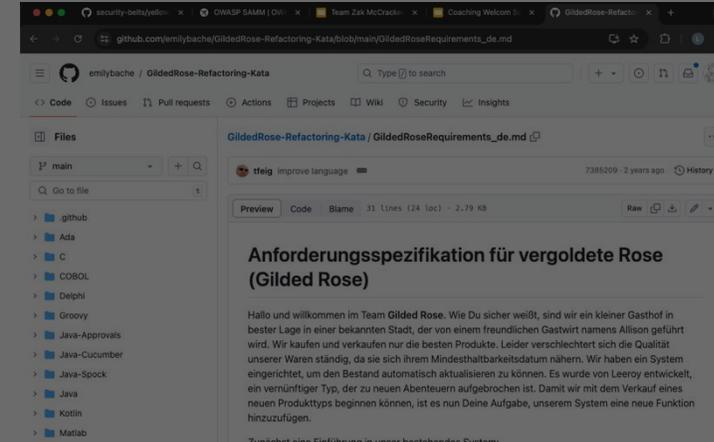
Coding Kata

Aus Software Craftsmen Community

Abfolge von Engineering Schritten wie Refactoring und TDD

Ohne Gegner Produktiv Code

Wird bis zur Perfektion geübt



Annahme

Katas verändern das Verhalten eines Teams

Erkenntnis

Hürde bei der Anwendung von Katas **im Team** auf echte Arbeit
Lernen als Individuum != Lernen als Team



Learning Hours

4C Model

Connect: Lernatmosphäre im Team schaffen

Concept: Ein neues Fähigkeit, die das Team lernen soll

Concrete: Hands-on Fähigkeit anwenden und ausprobieren

Conclusions: Reihum sagt jeder Teilnehmende was er/sie heute gelernt hat



Learning Hours by Topic

Find a learning hour on a particular topic:

- [Small Steps - Iterative and Incremental](#)
- [Test Design](#)
- [Test Doubles](#)
- [Testable Agile Design](#)
- [Refactoring](#)
- [Code Reading](#)
- [Legacy Code](#)
- [Architecture](#)
- [Behaviour Driven Development](#)
- [DevOps](#)
- [Approval Testing with TextTest](#)
- [Approval Testing in New Development](#)
- [Working in an Ensemble](#)
- [Working with C](#)
- [Git](#)

License: CC-BY-SA-4.0. Attribution: sammancoaching.org



Beck's rules of simple design

In this learning hour we learn about Kent Beck's rules of simple design. There is no 'concrete' part of this learning hour, you don't get to practice using these rules. You should probably follow up this learning hour with a second one where you do so.

Session Outline

- 5 min connect: vote for favourite design guidelines
- 25 min concept exercise: implement FizzBuzz however you like
- 10 min concept: read FizzBuzz code samples
- 10 min concept: YAGNI and Beck's rules of simple design
- 5 min conclusions: note down how TDD affects design



Beck's rules of simple design

In this learning hour we learn about Kent Beck's rules of simple design. There is no 'concrete' part of this learning hour, you don't get to practice using these rules. You should probably follow up this learning hour with a second one where you do so.

- 5 min connect: vote for favourite design guidelines
- 25 min concept exercise: implement FizzBuzz however you like
- 10 min concept: read FizzBuzz code samples
- 10 min concept: YAGNI and Beck's rules of simple design
- 5 min conclusions: note down how TDD affects design

Connect - vote for favourite design guidelines

Make a list of plausible design guidelines that people might find important/useful. Put them up on a shared whiteboard and ask each person to pick their top 4 and mark them with dot votes.

- No duplication
- Single return per function
- Testable (has unit tests)
- Extension points for adding functionality
- Doesn't have unnecessary elements or extension points
- Readable
- Uses naming conventions like m_... for member, f for interface
- Small classes and methods
- Small memory footprint
- Fast speed of execution

Do include language or organization specific guidelines if you know any good ones.

Concept exercise

Give them a starting position with an empty failing test and ask them to implement FizzBuzz. Tell them to do it however they want to, and to follow the design guidelines they want to follow. If no-one does TDD and they all end up with rather simple but less testable code, you might want to give them a quick demo of what solving FizzBuzz with TDD looks like. You should end up with something more like code sample 3 in the next section. Hopefully you won't have to - the ideal is that some pairs use TDD and end up with code like sample 3, and some don't, and end up with code like sample 1 or 2. That's important for the next section.

Concept - TDD changes your design

Print out and pin up the code samples from 'FizzBuzzViaSamples' around the walls of the room. In order: Print out the implementations and the tests. Make someone walk around in their

Evil User Stories

Als verärgerter **Filialeleiter** nutze ich meinen **Zugriff** auf die **Buchungssysteme**, um mir selbst **Geld** zu überweisen.



Als **<Rolle>**, kann ich **<Fähigkeit>**, um **<Mehrwert>** zu bekommen.

— Routine —

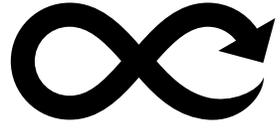
Security bei neuen Features einfach mitdenken!



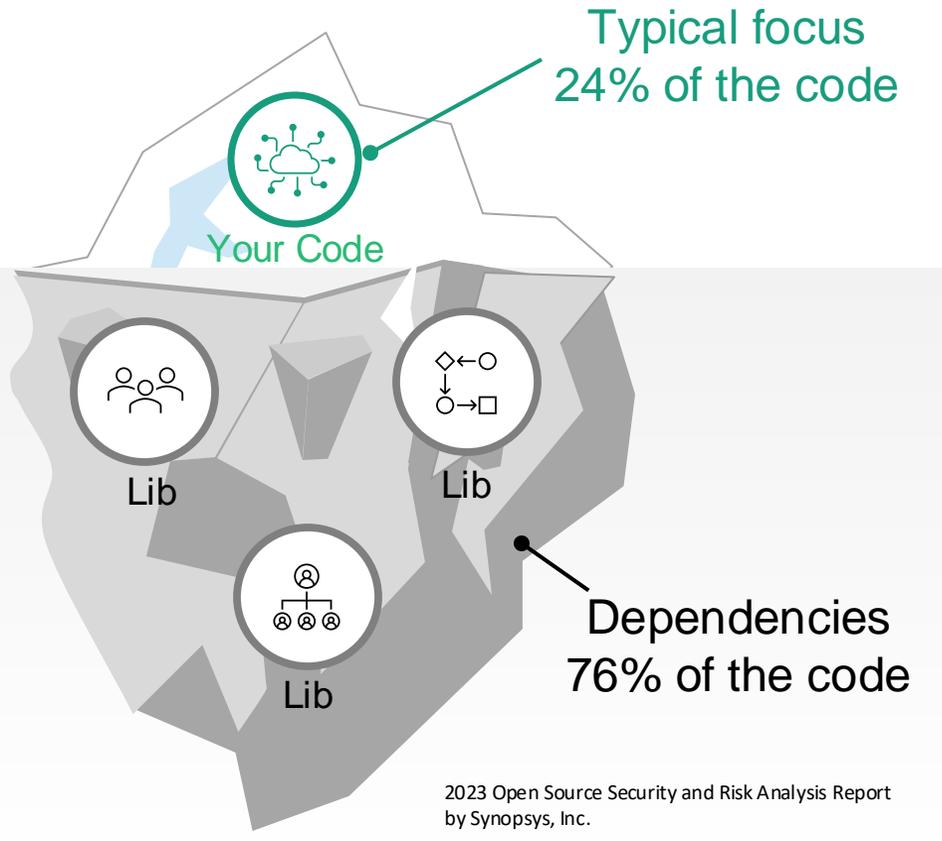
Externer Coach

— Example —

— Theory —



Patchmanagement



Renovate Bot

Technical Solution

Process

78% aller Vulnerabilitäten in der Open Source-Code-Base sind durch unpatchte Abhängigkeiten entstanden. Wie können wir kontinuierlich besser werden?



Externer Coach

State of Open Source Security Report 2019 Snyk Inc.

Awareness

Mob Sessions

- Stärkt/Erzeugt Vertrauen
→ Keine Angst vor Technik
- Holt das Team da ab wo es ist
- Lernen durch “anfassen”
- TDD und Pull Request als stille Helden
- Aufbau von Wissen zu Angriffstechniken
- Fördert die Transferleistung von erlerntem Wissen in konkrete Probleme



Summary der Methodik



Taylor ist tot



README Code of conduct CC-BY-SA-4.0 license

release v1.1.2 Check Markdown Links passing Spell check passing

Security Belts

Capability Modelle zeigen Verbesserungspotential

In order to continuously improve the Security Belts model, we appreciate any kind of feedback or content contribution. If you are interested in contributing, please see the document [Contributing](#).

Lernen als Individuum (Kata) →
Lernen im Team (Mob) →
Verhaltensänderung

Chartering Workshop
Learning Hour

Summary des Ergebnis



Security Awareness durch konstante Betreuung massiv gesteigert

Änderungen brauchen Zeit

Riesiger Impact kann in sehr kurzer Zeit erzielt werden → Coaching als Motor und Mentor für Verhaltensänderung

Leider ist eine dedizierte Langzeit Betreuung eines Teams selten möglich

Linkes Twix/Rechtes Twix ? NEIN Tandem



Erstkontakt über Internen Coach als Wegbereiter

Externer Coach übernimmt für Langzeit Betreuung der Teams

Tandem für optimale Abdeckung

Vorbeugen von Abnutzungseffekten z.B. durch unterschiedliche Methodik



Lars Hermerschmidt

Product Owner Security Engineering
lars.hermerschmidt@rewe-group.com

REWE digital



Jan-Niclas Strüwer

Research Associate
jan-niclas.struewer@iem.fraunhofer.de

Fraunhofer IEM



Benji Trapp

Technical Product Owner Red Team
benjamin.trapp@rewe-group.com

REWE digital