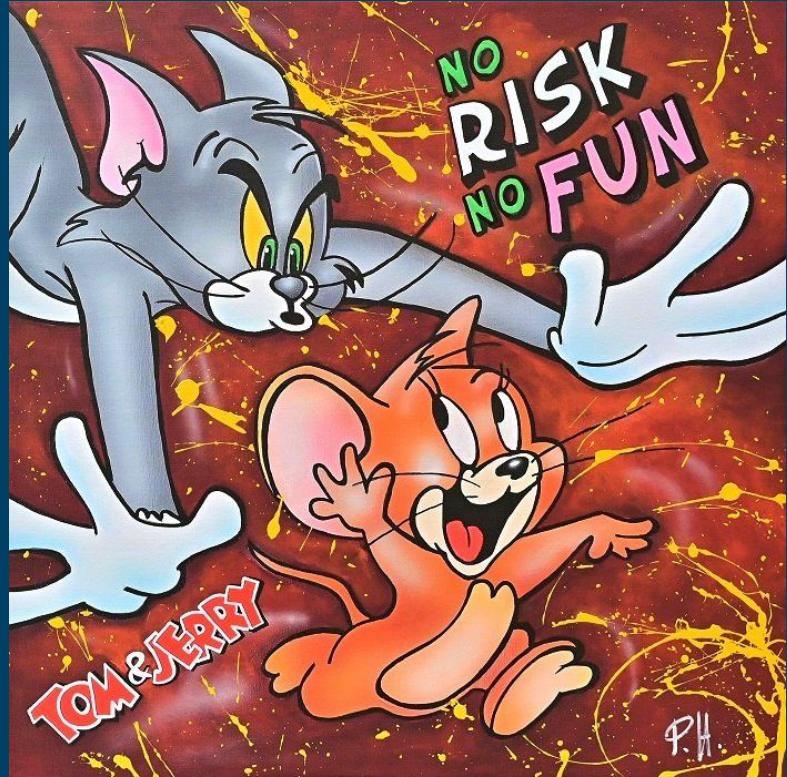


Sichere Software- entwicklung

No risk no fun?

@jasie@machteburch.social
twitter.com/ja_sie
jasie.de/sse



Quelle: peterheylands.com

Über mich



ICSP
Cyber Security Practitioner



ISTQB®
Certified Tester
Foundation Level
Agile Tester

@jasie@machteburch.social • twitter.com/ja_sie • jasie.de •



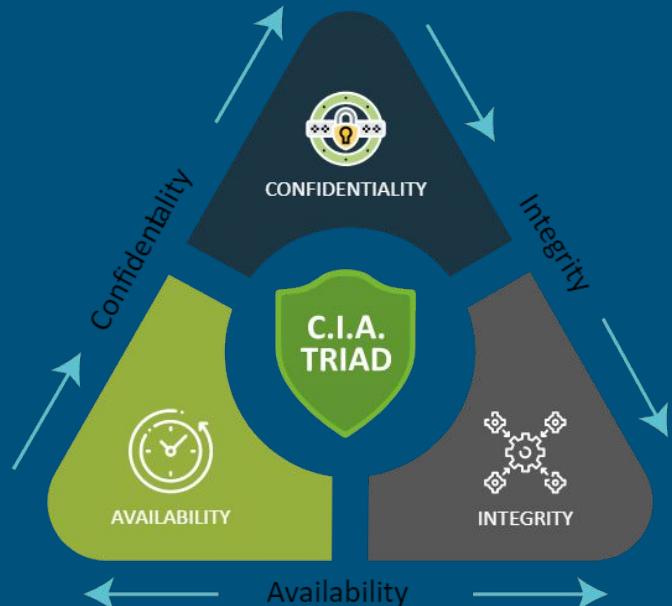
Gliederung

- I. Einführung
- II. Grenzen & Nutzen
- III. Basis-Anforderungen
- IV. In der Praxis

I) Einführung

Begriffe

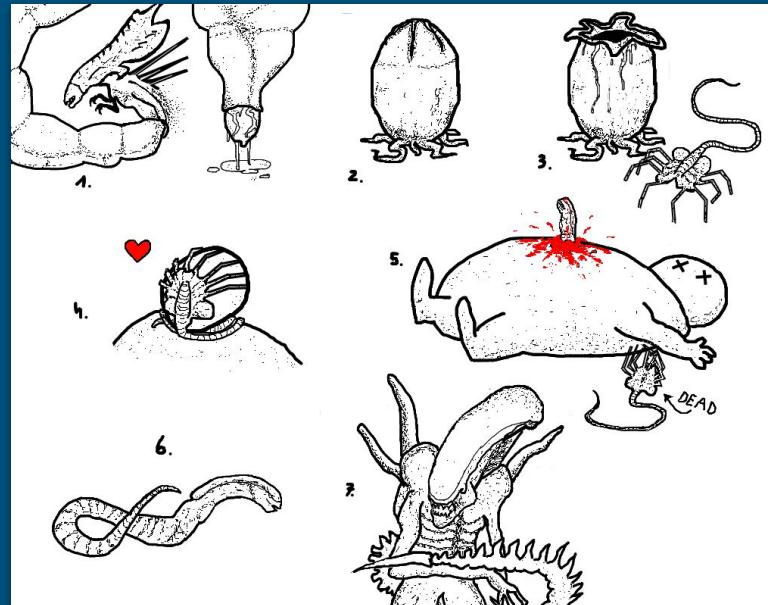
- **Softwaresicherheit**
 - Vertraulichkeit - Integrität - Verfügbarkeit
(CIA Schutzziele)
 - Böswilligkeit und Unabsichtlichkeit



Quelle: websitesecuritystore.com

Begriffe

- **Sichere Softwareentwicklung**
 - CIA in jeder Phase des Entwicklungszyklus'
 - Planung & Spezifikation
 - eigentliche Entwicklung ♥
 - Tests & Freigabe
 - Auslieferung
 - Inbetriebnahme
 - Anpassungen



© Noel4 / deviantart.com

Meine Motivation

[ISO 27001]

*Informationstechnik – Sicherheitsverfahren –
Informationssicherheitsmanagementsysteme – Anforderungen*



© iso.org

→ A.14.2.1 **Richtlinie für sichere Entwicklung:**

“Regeln für die Entwicklung von Software [...] sind festgelegt und bei Entwicklungen in der Organisation angewendet.”

Quelle: IT Crowd



Eure Motivation

[ISO 25010]

*System und Software-Engineering –
Qualitätskriterien und Bewertung von System und Softwareprodukten (SQuaRE)*



© iso.org

Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability
<ul style="list-style-type: none">• Functional Completeness• Functional Correctness• Functional Appropriateness	<ul style="list-style-type: none">• Time Behaviour• Resource Utilization• Capacity	<ul style="list-style-type: none">• Co-existence• Interoperability	<ul style="list-style-type: none">• Appropriateness• Recognizability• Learnability• Operability• User Error Protection• User Interface Aesthetics	<ul style="list-style-type: none">• Maturity• Availability• Fault Tolerance• Recoverability	<ul style="list-style-type: none">• Confidentiality• Integrity• Non-repudiation• Authenticity• Accountability	<ul style="list-style-type: none">• Modularity• Reusability• Analysability• Modifiability• Testability	<ul style="list-style-type: none">• Adaptability• Installability• Replaceability

BSI IT-Grundschutz

CON: Konzeption und Vorgehensweise

- ↓ CON.1 Kryptokonzept
- ↓ CON.2 Datenschutz
- ↓ CON.3 Datensicherungskonzept
- ↓ CON.6 Löschen und Vernichten
- ↓ CON.7 Informationssicherheit auf Auslandsreisen
- ↓ CON.8 Software-Entwicklung
- ↓ CON.9 Informationsaustausch
- ↓ CON.10 Entwicklung von Webanwendungen
- ↓ CON.11.1 Geheimschutz VS-NUR FÜR DEN DIENSTGEBRAUCH (VS-NfD)



© LaundryFactory | redbubble.com

[BSI CON.8]

II) Nutzen & Grenzen

Grenzen 😞

- Usability
- keine absolute Sicherheit
- niemals fertig
- Zeit

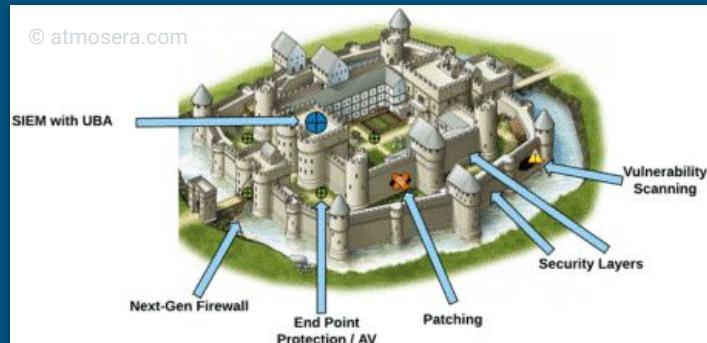
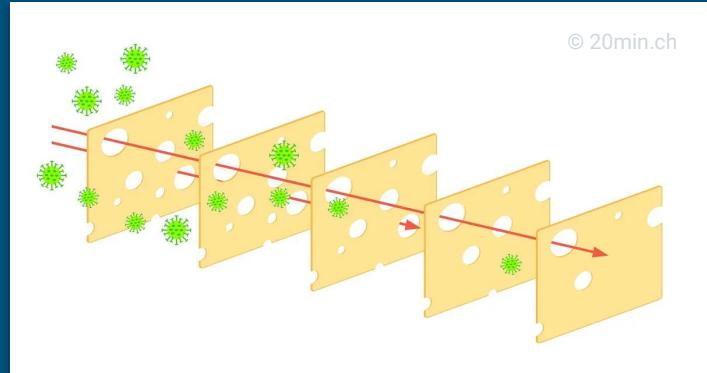


Quelle: Big Bang Theory

Nutzen 😊

- reduziertes Risiko bzgl. Schutzziele CIA
- Nachweis gegenüber Kunde
- Zertifizierungen / Wettbewerbsvorteil
- Business Opportunity

Defense in Depth



III) Basis-Anforderungen

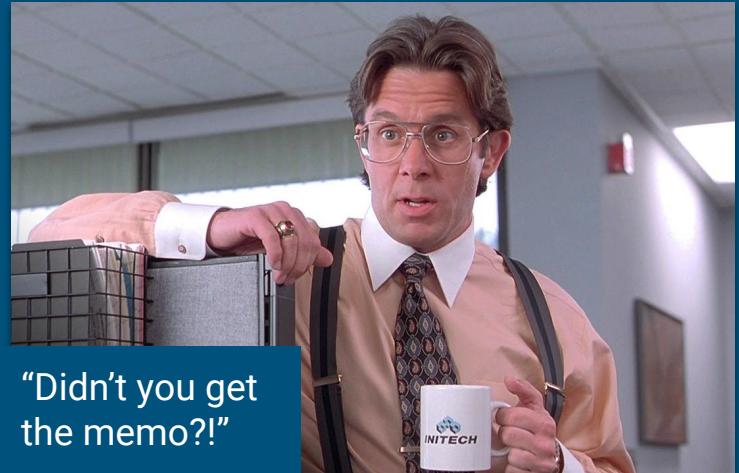
Auswahl eines **Vorgehens- modells zur Software- Entwicklung**

Basis-Anforderung 1

1) Vorgehensmodell

- “Sicherheitsanforderungen des Auftraggebers an **Vorgehensweise** im **Vorgehensmodell** integriert”
- “Personal in Methodik des **Vorgehensmodells** geschult”

[CON.8.A2]



“Didn’t you get
the memo?!“

Quelle: Office Space

1) Vorgehensmodell

Beispiele:

- Security Acceptance Criteria
- “AbUser”-Stories
aka Evil User Stories
- Security Stories

*"As (**malicious user**), I want (**malicious activity**), in order to (**business impact**)".*

"As an attacker, I want to try to guess a user's password by sending a large number of authentication requests in parallel to log in to his session".

*"As (**squad role**), I want to (**activity**), in order to (**prevent an evil user story from happening**)".*

"As a developer, I want to **set up a mechanism to block user accounts** after 5 attempts **to avoid brute force attacks**".

Auswahl einer Entwicklungs- umgebung

Basis-Anforderung 2

2) Entwicklungsumgebung

- “*Liste erforderlicher und optionaler Auswahlkriterien für eine Entwicklungsumgebung vom Verantwortlichen für die Software-Entwicklung erstellt*”
- “*Entwicklungsumgebung anhand der vorgegebenen Kriterien ausgewählt*”

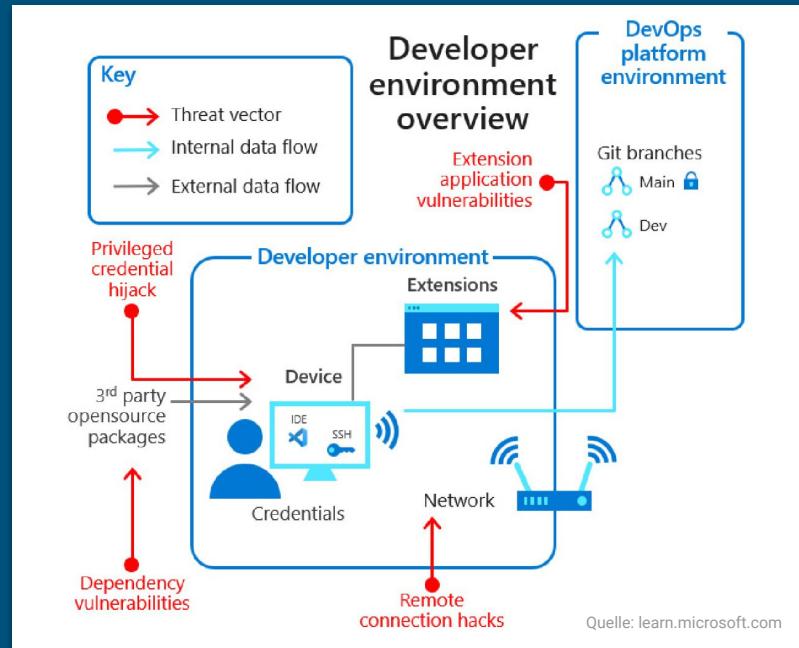


[CON.8.A3]

2) Entwicklungsumgebung

Beispiele:

- erforderlich:
sicheres OS,
Auth via SSH
- optional:
welche IDE,
welches Git Tool



Sicheres **Systemdesign** in der zu entwickelnden Software

Basis-Anforderung 3

3) Sicherer Systemdesign /1

- “alle **Eingabedaten** vor der Weiterverarbeitung geprüft und **validiert**”
- “bei Client-Server-Anwendungen Daten grundsätzlich auf **Server** validiert”

[CON.8.A5]



3) Sicherer Systemdesign /1

Beispiele:

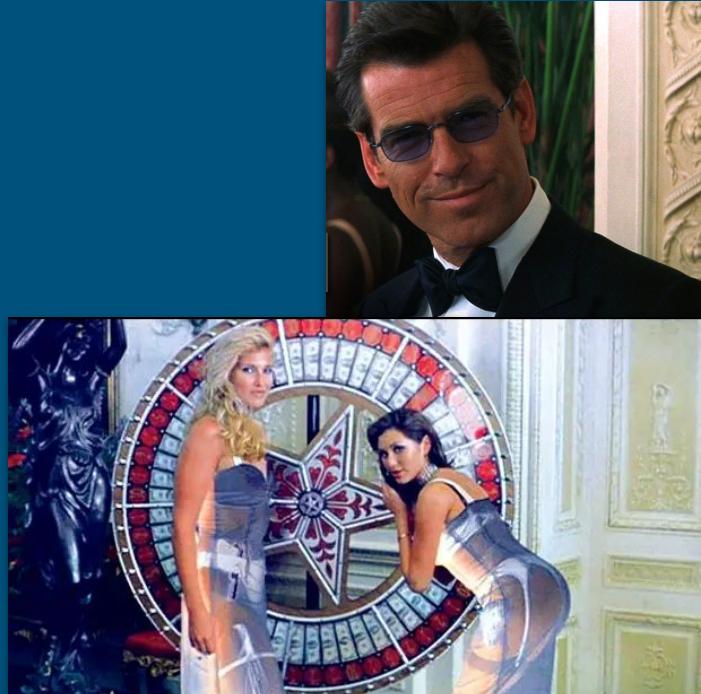
- Client:
E-Mail-Input, Pflichtangaben, ...
- Server:
Authentication, Dateiformat, ...



3) Sicherer Systemdesign /2

- “*bei Fehlern oder Ausfällen von Komponenten des Systems keine schützenswerten Informationen preisgegeben*”
- “*Schützenswerte Daten verschlüsselt übertragen und gespeichert*”

[CON.8.A5]



Quelle: James Bond

3) Sicherer Systemdesign /2

Beispiele:

- Security by Obscurity
 - Development mode & Production mode



3) Sicherer Systemdesign /3

- *“Benutzer-Authentisierung und Authentifizierung entsprechend Sicherheitsanforderungen der Anwendung”*
- *“Passwörter mit **sicherem Hashverfahren** gespeichert”*

[CON.8.A5]



Quelle: dr-weedy.com

3) Sicherer Systemdesign /3

Beispiele:

- Auth-Anforderung:
2FA,
Sperre bei x Fehlversuchen
- PW-Hashverfahren:
MD5 Argon2

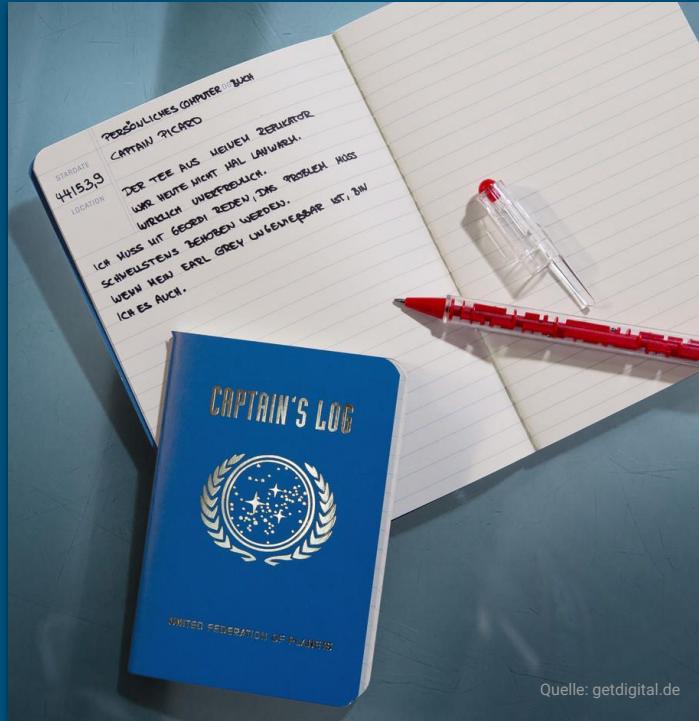


Quelle: kaspersky.de

3) Sicherer Systemdesign /4

- “*sicherheitsrelevante Ereignisse protokolliert*”
- “*Programmcode und Konfigdateien bereinigt ausgeliefert*”

[CON.8.A5]



Quelle: getdigital.de

3) Sicherer Systemdesign /4

Beispiele:

- Protokolle:
Rechteänderungen, DB-Fehler, ...
- bereinigter Code:
keine Secrets, keine Kommentare...



Quelle: Der Tatortreiniger

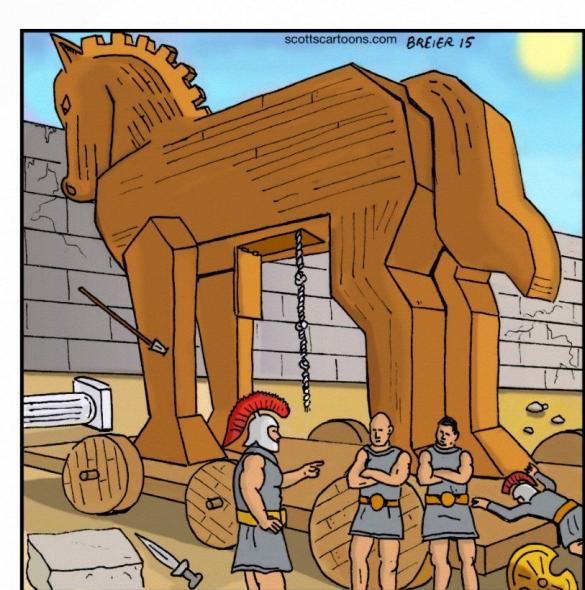
Verwendung von **externen** **Bibliotheken** aus vertrauens- würdigen Quellen

Basis-Anforderung 4

4) Bibos aus vertrauenswürdigen Quellen

- "externe **Bibliotheken** aus **vertrauenswürdigen** Quellen bezogen"
- "**Integrität** der externen Bibliotheken vor Verwendung sichergestellt"

[CON.8.A6]



"Norton! McAfee! How did you NOT detect this as a threat? You're both useless, you know that?"

4) Bibos aus vertrauenswürdigen Quellen

Beispiele:

- Vertrauenswürdigkeit:
Infos über den/die Entwickler
- Integrität:
wie oft aktualisiert

The screenshot shows the npmjs.com package page for 'is-odd'. At the top, it displays the package name 'is-odd' with an 'npm v3.0.1' badge, 'downloads 2.3M/month' badge, and a green 'Travis passing' badge. Below this is a brief description: 'Returns true if the given number is odd, and is an integer that does not exceed the JavaScript MAXIMUM_SAFE_INTEGER.' A note encourages users to support the author, Jon Schlinkert. The 'Install' section shows how to install with npm: '\$ npm install --save is-odd'. The 'Usage' section provides an example: 'const isOdd = require('is-odd'); console.log(isOdd('1')) //=> true; console.log(isOdd('3')) //=> true'. To the right, there's a sidebar with package statistics: Weekly Downloads (495,273), Version (3.0.1), License (MIT), Unpacked Size (6.51 kB), Total Files (4), Issues (0), Pull Requests (5), Homepage (github.com/jonschlinkert/is-odd), Repository (github.com/jonschlinkert/is-odd), and Last publish (3 years ago).

The screenshot shows the npmjs.com package page for 'is-ten-thousand'. It features a yellow 'Readme' tab, an 'Explore' tab with a '28 Dependencies' badge, a '1 Dependents' badge, and a '2 Versions' badge. The main description states: 'If you need to know if a number is ten thousand.' The 'Keywords' section lists: original, is-ten-thousand, is-thousand, is-hundred, is-ten, beauty, grace. To the right, there's a sidebar with package statistics: Weekly Downloads (20), Version (0.2.0), License (WTFPL), Unpacked Size (7.08 kB), and Total Files (5).

Quelle: npmjs.com

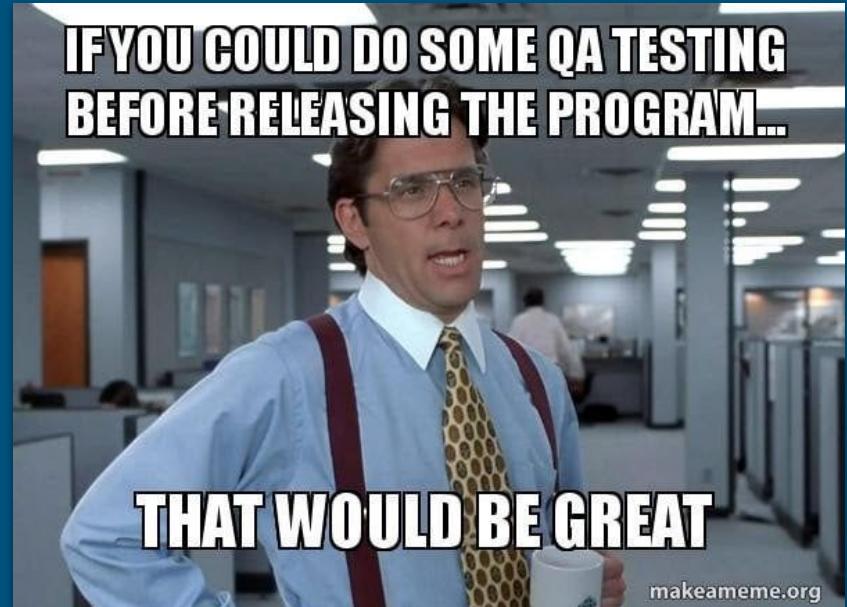
Durchführung von entwicklungs- begleitenden **Software-Tests**

Basis-Anforderung 5

5) Software-Tests /1

- “vor Freigabe entwicklungsbegleitende **Software-Tests** durchgeführt”
- “vor Freigabe Quellcode auf **Fehler gesichtet**”

[CON.8.A7]



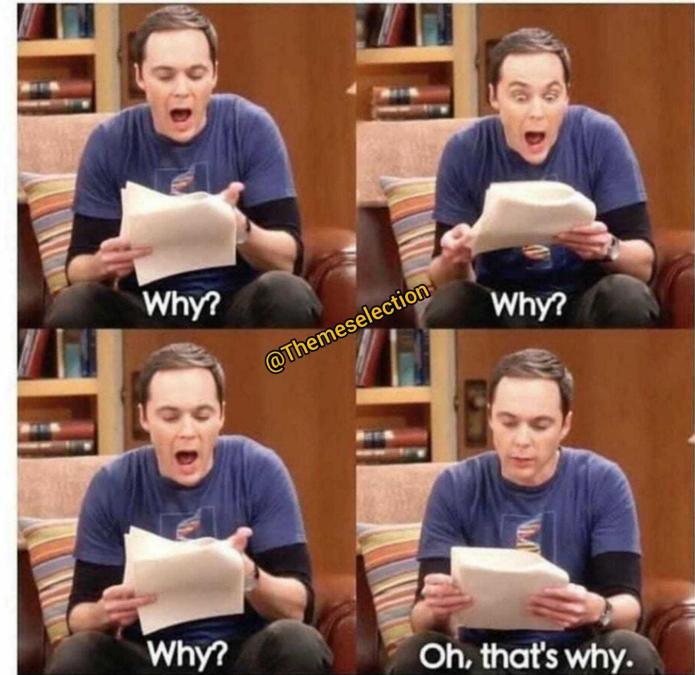
Quelle: Office Space

5) Software-Tests /1

Beispiele

- Komponenten- & Integrationstests als Teil der DoD
- toolgestütztes Code Reviewing, Bitbucket PR, Gitlab MR, ...

Programmers while reviewing the codes



Quelle: Big Bang Theory

5) Software-Tests /2

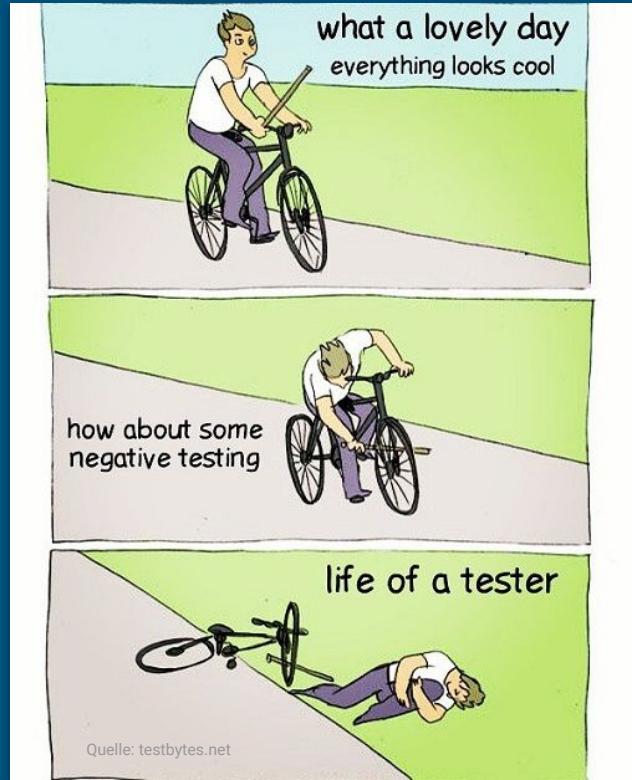
- “*Tests umfassen die **funktionalen** und **nicht-funktionalen** Anforderungen der Software*”
- “**Negativtests** abgedeckt & **kritische Grenzwerte** der *Eingabe* sowie der *Datentypen* überprüft”

[CON.8.A7]

5) Software-Tests /2

Beispiele

- funktional:
Komponententest
- nicht-funktional:
Performanztest
- Negativtest:
neg. Wert, null, n/a



5) Software-Tests /3

- “Software getestet in **Test- und Entwicklungsumgebung** getrennt von **Produktionsumgebung**”
- “automatische **statische Code-Analyse** durchgeführt”

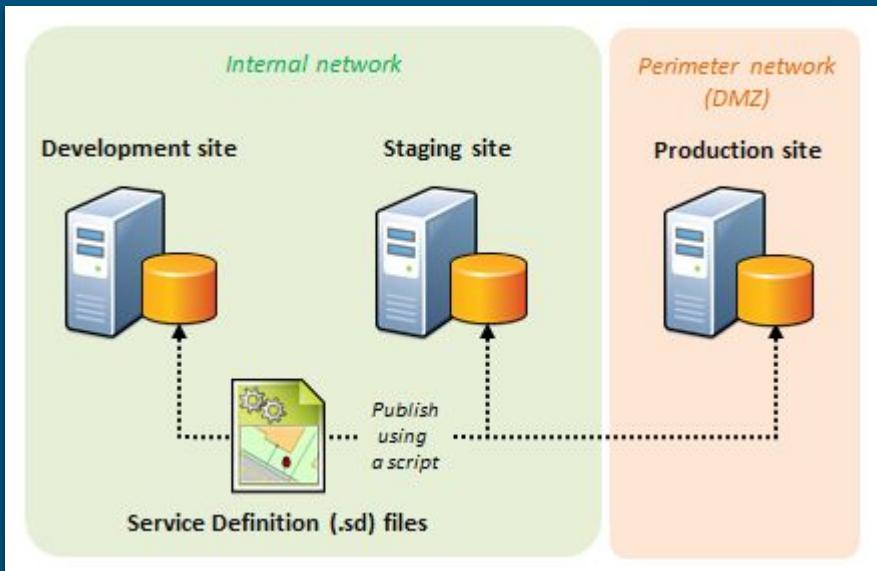
[CON.8.A7]



5) Software-Tests /3

Beispiele

- getrennte Umgebungen:
lokaler Rechner, Staging Server,
Production Server
- statische Analyse:
in IDE u/o in der CD/CI Pipeline



Bereitstellung von Patches, Updates & Änderungen für die entwickelte Software

Basis-Anforderung 6

6) Patches & Updates

- “**sicherheitskritische Patches und Updates für entwickelte Software zeitnah durch Entwickler bereitgestellt**”
- “**für Installations-, Update- oder Patchdateien vom Entwickler Checksummen oder digitale Signaturen bereitgestellt**”

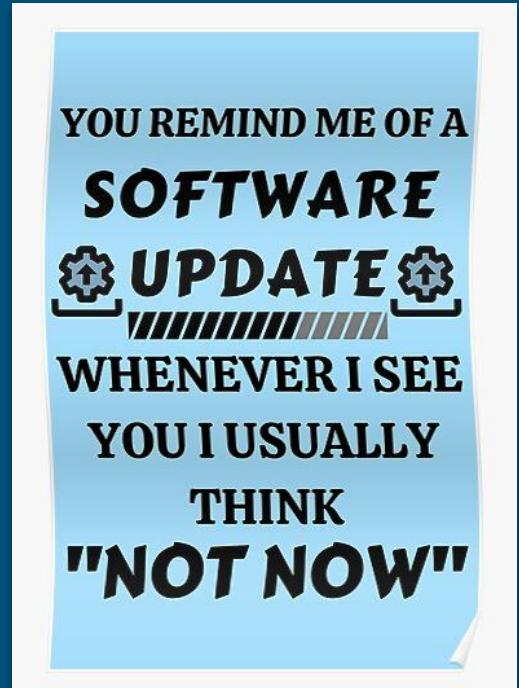
[CON.8.A8]



6) Patches & Updates

Beispiele:

- je Quartal 1 Ticket für Patches
- Trennung nach Bugfix - Minor - Major



Quelle: redbubble

Versions- verwaltung des Quellcodes

Basis-Anforderung 7

7) Versionsverwaltung

- “Quellcode des Entwicklungsprojekts über geeignete **Versionsverwaltung** verwaltet”
- “Versionsverwaltung nicht ohne **Datensicherung**”

[CON.8.A10]



Quelle: Star Trek

7) Versionsverwaltung

Beispiele:

- 'Git-Flow' Branching-Modell
- Pull/Merge Requests
- Konfiguration



Überprüfung von externen Komponenten

Basis-Anforderung 8

8) Überprüfung externer Komponenten

*"unbekannte externe Komponenten,
deren Sicherheit nicht durch etablierte
und anerkannte Peer-Reviews oder
vergleichbares sicherstellbar, auf
Schwachstellen überprüft"*

[CON.8.A20]



Quelle: colourbox.de

8) Überprüfung externer Komponenten

Beispiel:

- Vulnerability Scanner Tools:
npm audit, SonarCube, BlackDuck
- CVE Verzeichnisse:
cve.mitre.org, ...



Quelle: Star Trek

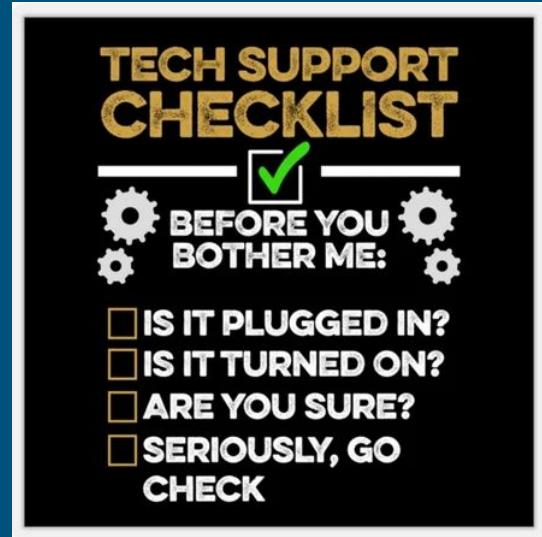
Zusammenfassung CON.8

- 1) Auswahl eines **Vorgehensmodells**
- 2) Auswahl einer **Entwicklungsumgebung**
- 3) Sicheres **Systemdesign**
- 4) **Bibliotheken** aus vertrauenswürdigen Quellen
- 5) Entwicklungsbegleitende **Software-Tests**
- 6) **Patches**, Updates und Änderungen
- 7) **Versionsverwaltung** des Quellcodes
- 8) Überprüfung von **externen Komponenten**

IV) In der Praxis

In der Praxis

- **Richtlinie** für Sichere Softwareentwicklung
- **Security Checkliste** je Projekt
- 1 Verantwortlicher je Projekt
- DoD / Acceptance Criteria



Quelle: KJ / etsy.com

In der Praxis

Checkliste_CON.8.xlsx | © BSI

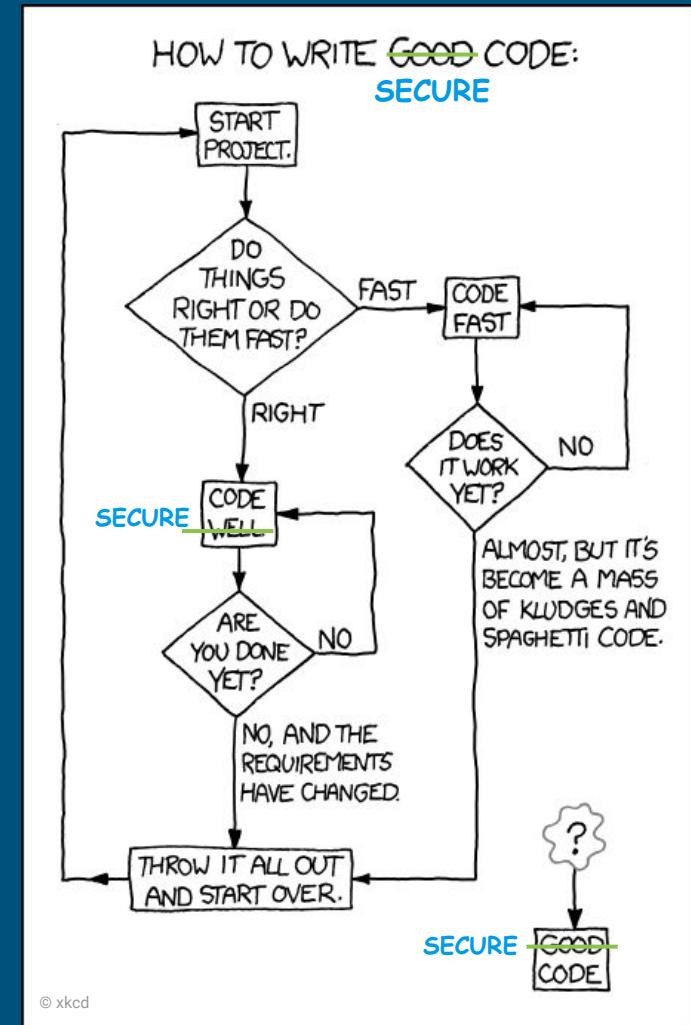
A	B	C	D	E	F	G	H	I	J
ID	ID-Anforderung	Titel	Inhalt	Typ	Entbehrlich	Begründung für Entbehrlichkeit	Umsetzung	Umsetzung bis	Verantwortlich
1									
2		Baustein: CON.8 Software-Entwicklung							
3		Kompendium: 2023							
4									
5									
6	CON.8.A2	Auswahl eines Vorgehensmodells	Ein geeignetes Vorgehensmodell zur Software-Entwicklung MUSS festgelegt werden.	Basis					
7	CON.8.A2	Auswahl eines Vorgehensmodells	Anhand des gewählten Vorgehensmodells MUSS ein Ablaufplan für die Software-Entwicklung erstellt werden.	Basis					
8	CON.8.A2	Auswahl eines Vorgehensmodells	Die Sicherheitsanforderungen der Auftraggeber an die Vorgehensweise MÜSSEN im Vorgehensmodell integriert werden.	Basis					
9	CON.8.A2	Auswahl eines Vorgehensmodells	Das ausgewählte Vorgehensmodell, einschließlich der festgelegten Sicherheitsanforderungen, MUSS eingehalten werden.	Basis					
10	CON.8.A2	Auswahl eines Vorgehensmodells	Das Personal SOLLTE in der Methodik des gewählten Vorgehensmodells geschult sein.	Basis					
11	CON.8.A3	Auswahl einer Entwicklungsumgebung	Eine Liste der erforderlichen und optionalen Auswahlkriterien für eine Entwicklungsumgebung MUSS von Fachverantwortlichen für die Software-Entwicklung erstellt werden.	Basis					

Fazit

Fazit



© Artistry Team / RedBubble



© xkcd

Quellen

1. IT-Grundschutz-Kompendium: **CON.8 Software-Entwicklung** | BSI 2023
bsi.bund.de/.../Grundschutz/IT-GS-Kompendium_Einzel_PDFs_2023/03_CON_Konzepte_und_Vorgehensweisen/CON_8_Software_Entwicklung_Edition_2023.html
2. IT-Grundschutz-Kompendium: **Checklisten** | BSI 2023
bsi.bund.de/.../Grundschutz/Kompendium/checklisten_2023.html?nn=128568
3. iso25000.com: **ISO/IEC 25010**
iso25000.com/index.php/en/iso-25000-standards/iso-25010/
4. ScienceSoft: **Sichere Softwareentwicklung: Schritt-für-Schritt-Anleitung** | D. Dmitry Nikolaenya, 2019
scnsoft.de/blog/sichere-softwareentwicklung

Weiterführend /1

1. zu CON 8.A.2: **Ein Vorgehensmodell für die Entwicklung sicherer Software** | A Lunkeit, W. Zimmer 2016
syssec.at/de/veranstaltungen/.../dachsecurity2016/papers/DACH_Security_2016_Paper_22B3.pdf
2. zu CON 8.A.3: National Cyber Security Center: **Secure development and deployment guidance – Secure your development environment**
ncsc.gov.uk/collection/developers-collection/principles/secure-your-development-environment
3. zu CON 8.A.5: Medium: **Password Hashing: Scrypt, Bcrypt and ARGON2** | M. Prezioso 2019
medium.com/analytics-vidhya/password-hashing-pbkdf2-scrypt-bcrypt-and-argon2-e25aaf41598e
4. zu CON 8.A.6: Medium: **Choosing a Third-Party Library** | D. Scheider 2021
dana-scheider.medium.com/choosing-a-third-party-library-e8b0f7aa9497

Weiterführend /2

5. zu CON 8.A.7: Basiswissen Softwaretest | T. Linz 2019
6. zu CON 8.A.8: **Wie man mit 3rd Party Libraries umgeht: Strategien für den erfolgreichen Einsatz von Fremdbibliotheken** | 2015
entwickler.de/iot/wie-man-mit-3rd-party-libraries-umgeht-strategien-fur-den-erfolgreichen-einsatz...
7. zu CON 8.A.10: Medium: **Code Security – The Importance of Securing Your Version Control** | L. Oliff 2018
medium.com/@lukeocodes/the-importance-of-securing-your-version-control-131841429994
8. zu CON 8.A.20: DNSstuff: **Top 15 der kostenpflichtigen und kostenfreien Vulnerability-Scanner-Tools** | 2020
dnsstuff.com/de/network-vulnerability-schwachstellen-scanner