Secure Software Engineering

Lehrplan zum Basiskurs

ASQF Secure Software Engineer SSE

Lehrplan Version: V2.1

Letzte Freigabe: 21.08.2024

Copyright und Nutzungsrechte

Das Werk ist urheberrechtlich geschützt. Jede Verwertung außerhalb des Urheberrechtsgesetzes ist ohne Zustimmung des ASQF e.V. unzulässig und strafbar. Das gilt insbesondere für die Weiterverarbeitung, Übersetzung und die Bearbeitung in elektronischen Systemen.

In diesem Dokument wird bei Personenbezeichnungen und personenbezogenen Hauptwörtern i.A. die männliche Form verwendet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung grundsätzlich für alle Geschlechter. Die verkürzte Sprachform dient der besseren Lesbarkeit und beinhaltet keine Wertung.

Autoren

Vera Gebhardt

Sebastian Dengler

Dipl.-Eng. Olga Jaufman

Dr. Thomas Fehlmann

Dr.-Ing. Tobias Koal

Dr. Kristian Trenkel

Max Perner

Reviewer

Prof. Dr. Jürgen Mottok

Dipl.-Ing. Axel Gürtler

Dr.-Ing. Armin Lunkeit

Günter Jung

Torsten Schulz

Dipl.-Ing. Axel Wintsche

Marcel Schwarzmeier

Wir danken den folgenden Mitwirkenden für ihren Beitrag:

Dr.-Ing. Armin Lunkeit

Prof. Dr. Friedrich Holl

Dipl.-Ing. (FH) Hauke Petersen

Prof. Dr. Ivo Keller

Dipl.-Ing. Konstantinos Dalamagkidis, PhD

Dipl.-Wi.-Math. Mareike Roth

Prof. Dr. Jürgen Mottok

Änderungsübersicht

Version	Datum	Autor	Bemerkung
1.0	27.11.2018	Autoren und Reviewer	Erste freigegebene Version
1.1-1.4	-%-	-%-	Zwischenstände
2.0	21.04.2023	Autoren und Reviewer	Freigegebene Aktualisierung und Überarbeitung
2.0	06.12.2023	Reviewer	Aktualisierung und Überarbeitung
2.1	16.08.2024	Reviewer	Aktualisierung Anpassung der LOs und Überarbeitung

Inhaltsverzeichnis

Learning Objectives

1.1.1 LO: Definition Secure Software Engineering (SSE) kennen (K1, 5 min)

1.2.1 LO: Die Einordnung von Safety / Funktionale Sicherheit und Security / IT-Sicherheit verstehen (K2, 15min)

1.2.2 LO: Definitionen Grundbegriffe aus dem Sicherheitsmanagement verstehen (K2, 15min)

1.2.3 LO: Notwendigkeit von Standards und Regelwerke erkennen (K1, 5min)

1.3.1 LO: Die Security Triade kennen (K1, 5min)

1.3.2 LO: Die Bedeutung und das Verfahren des Secure SW Engineering verstehen (K2, 15 min)

1.4.1 LO: Den Begriff Datenschutz kennen (K1, 5min)

1.4.2 LO: Begriffe der Datenschutz Vorschriften erkennen (K1, 5 min)

1.4.3 LO: Die für Datenschutz wichtige Begriffe aufzählen können (K1, 5 min)

1.5.1 LO: Einbettung des konstruktiven SSE in Prozesse und Prozessbeteiligte verstehen (K2, 15min)

1.5.2 LO: Anwendbare Normen mit Bezug zur Administration nennen können (K1, 5min)

1.5.3 LO: Anwendbare Normen mit Bezug zur Software-Entwicklung nennen können (K1, 5min)

1.5.4 LO: Weitere Standards nennen können (K1, 5min)

1.5.5 LO: Einbettung des konstruktiven SSE in Lieferketten verallgemeinern (K2, 15min)

1.6.1 LO: Einbettung des SSE in Organisationen und in Prozesse beschreiben können (K2, 15 min)

1.6.2 LO: Zusammenhänge der wesentlichen Aktivitäten und Einflussfaktoren auf SSE im Lebenszyklus verstehen (K2, 15min)

1.6.3 LO: IT-Sicherheit als Qualitätsmerkmal im Software-Lebenszyklus verstehen (K2, 15min)

1.7.1 LO Offene Reifegradmodelle benutzen können (K3, 60 min)

2.1.1 LO: Grundlagen des Requirements-Engineerings kennen(K1, 5min)

2.1.2 LO: Qualitätsmerkmale Nachvollziehbarkeit, Testbarkeit und Umsetzbarkeit kennen (K1, 5min)

2.1.3 LO: Spezielle Bestimmungsmethoden für Anforderungen kennen (K1, 5min)

2.1.4 LO: Quellen weiterer Anforderungen kennen (K1, 5min)

2.1.5 LO: Verfolgbarkeitsketten/-graphen, Versionierung kennen (K1, 5min)

2.1.6 LO: Für SSE wichtige Qualitätsmerkmale kennen – mit Übung (K3, 60 min)

2.1.7 LO: Einbindung in bestehende Softwareentwicklung kennen (K1, 5min)

2.2.1 LO: Den Begriff Risiko kennen (K1, 5min)

2.3.1 LO: Definition von Modell und Gründe für modellbasiertes Vorgehen kennen (K1, 5min)

2.3.2 LO: Diagramme zum Finden von Vertrauensgrenzen und Angriffsflächen konstruieren können (K2, 15 min)

2.4.1 Unterschiedliche Herangehensweisen verstehen (K2, 15min)

2.4.2 LO: Verstehen, wie man schützenswerten Güter identifiziert und priorisiert (K2, 15min)

2.4.3 LO: Verschiedene Methoden der Bedrohungsanalyse kennen (K1, 5 min)

2.4.4 LO: Verstehen des Einflusses von Sicherheitslücken auf die IT-Sicherheit bzgl. einzusetzender Methoden und Metriken (K2, 15 min)

2.4.5 LO: Eine Methode (Attack Trees) zur Bedrohungsanalyse anwenden können. (K3, 60 min)

2.4.6 Risikoanalyse als Priorisierungsmaßstab kennen (K1, 5 min)

3.1.1 LO: Ansätze und Methodik kennen (K1, 5min)

3.1.2 LO: Bedeutung der Softwarearchitektur verstehen (K2, 15min)

3.1.3 LO: Moderne Softwarearchitektur verstehen (K2, 15min)

3.2.1 LO: Eigenschaften moderner Software-Architektur verstehen (K2, 15min)

3.2.2 LO: Methoden moderner Software-Architektur verstehen (K2, 15min)

3.2.3 LO: Verstehen des Messens von Privacy (K2, 15 min)

3.3.1 LO: Modernes Sicherheitsdesign kennenlernen (K1 5min)

3.3.2 LO: Schnittstellenanalyse kennen (K1, 5min)

3.3.3 LO: Den Sinn geeigneter Schutzmaßnahmen im Sicherheitsdesign kennen (K1, 5min)

3.4.1 LO: Vor- und Nachteile von Intrusion Detection kennen (K1, 5min)

3.4.2 LO: Geschützte Datenablagen verstehen (K2, 15 min)

3.4.3 LO: Integration des konstruktiven SSE in Organisation verstehen (K2, 15min)

4.1.1 LO: Motivation und Ziele im Software-Test kennen (K1, 5 min)

4.1.2 LO: Grundwissen zum Softwaretest kennen (K1, 5 min)

4.1.3 LO: Statische und dynamische Tests kennen (K1, 5 min)

4.1.4 LO: Black Box / White Box kennen (K1, 5 min)

4.2.1 LO: Typischen Angriffswege erkennen (K1, 5min)

4.3.1 LO: Verstehen von Methoden zur Schwachstellenanalyse von Architekturen (K2, 15min)

4.3.2 LO: Systematisches Testing mittels Tools kennen lernen (K1, 5 min)

4.3.3 LO: Statische Analyse Techniken kennen (K1, 5 min)

4.3.4 LO: Dynamische Analyse Techniken kennen (K1, 5 min)

4.4.1 LO: Bewertung und Nachweis der IT-Sicherheit kennen (K1, 5min)

5.1.1 LO: DevOps Zyklus kennen (K1 / 5min)

5.1.2 LO: Typische Vorteile und Nachteile des automatisierten Testens im Lebenszyklus einer Software nennen können (K1 / 5 min)

5.1.3 LO: Begriff Systemüberwachung kennen (K1, 5min)

5.1.4 LO: Rollen und Aufgaben in DevSecOps kennen (K1, 5min)

5.1.5 LO: Bereitstellungsumgebung und automatisierte Auslieferung kennen (K1, 5 min)

5.2.1 LO: Definitionen zu Identitäten

5.2.2 LO: Kernbegriffe von Deployment und Betrieb erklären können (K2, 15 min)

5.2.3 LO: Die Notwendigkeit von Informationssicherheit im Deployment und Betrieb erkennen können (K2, 15min)

5.3.1 LO: Sicheres Deployment ausführen können (K3, 60 min).

5.3.2 LO: Unterschiede zwischen Zugangskontrollmethoden wiedergeben können (K1, 5 min)

5.3.3 LO: Vorteile und Nachteile von Virtualisierung

5.4.1 LO: Die Hauptmerkmale von Patch-Management und Software-Vulnerability-Management auflisten können (K2, 15min).

5.4.2 LO: Incident Response als wichtigen Geschäftsprozess beim Betrieb von Software kennen (K1, 5min)

5.4.3 LO: Begriffe und Aktivitäten bezüglich Beschaffung und Außerbetriebnahme auflisten können (K1, 5min)

5.5.1 LO: Fehlerkultur und Kritikfähigkeit veranschaulichen (K2, 15 min)

5.5.2 LO: Security & Vulnerability Management kennen (K1, 5 min).

5.5.3 LO: Sprint am Beispiel Scrum erkennen (K2 / 15min)

5.6.1 LO: Begriffe im Lebenszyklus kennen (Erweiterung zu Kapitel 1.5) (K1, 5 min)

5.6.2 LO: Anwenden des Security Development Life Cycle (K3, 60 min)

Taxonomiestufe 1: Kennen (K1)

Taxonomiestufe 2: Verstehen (K2)

Taxonomiestufe 3: Anwenden (K3)

Lehrplaninhalte

1. Grundverständnis für SSE und dessen Ziele
2. Bedrohungsanalyse und Anforderungen
3. Engineering & Architektur
4. Security Testing
5. Lifecycle & Prozesse

Business Outcomes

Der Geschäftsnutzen (Business Outcomes) durch die Teilnahme an einer Ausbildung basierend auf dem vorliegenden Lehrplan für den Teilnehmer und dessen Organisation stellt sich wie folgt dar:

- erlangt Verständnis für Erfordernis und Nutzen von Security
- kann Sicherheitsanforderungen definieren
- kann Sicherheitsanforderungen umsetzen
- versteht Prozesse/Konzepte/Methoden und kann bei deren Einführung unterstützen
- erweitert seine Kompetenz und bringt dieses Wissen gezielt ein

Zielgruppe

Alle Stakeholder die am SW-Entwicklungsprozess beteiligt sind, zum Beispiel:

- Informatiker und Ingenieure
- SW-Architekten und Entwickler
- Entwicklungs- und Testteams
- Qualitätsbeauftragte und Projektmanager

Secure Software Engineering ist die Anwendung von disziplinierten, quantifizierbaren Ansätzen für die Entwicklung, den Betrieb und die Wartung von sicherer Software nach systematischen Methoden.

1. Grundverständnis für SSE und dessen Ziele

1.1 Definition SSE

1.1.1 LO: Definition Secure Software Engineering (SSE) kennen (K1, 5 min)

Das Ziel von SSE ist es, eine Software zu schreiben, die nach Anforderungen und aktuellem Wissen einem Angreifer keine Schwachstellen zur Verfügung stellt.

- **Secure Software Engineering**

ist die Anwendung eines systematischen, disziplinierten, quantifizierbaren Ansatzes für die Entwicklung, den Betrieb und die Wartung von sicherer Software nach anerkannten und wiederholbaren Methoden.

1.2 Bedeutung der Grundbegriffe aus dem Sicherheitsmanagement

1.2.1 LO: Die Einordnung von Safety / Funktionale Sicherheit und Security / IT-Sicherheit verstehen (K2, 15min)

Safety und Security (Funktionale Sicherheit / Cybersicherheit) haben Gemeinsamkeiten und Unterschiede, die z. T. gleichzeitig und z. T. parallel zu beachten sind. Der Risikobegriff enthält eine Bewertung für ein Schadensszenario.

- Safety (funktionale Sicherheit) schützt Menschen und Umwelt vor Device (Gerät)
- Security schützt Device vor Manipulation
- Safety fordert Security ein

1.2.2 LO: Definitionen Grundbegriffe aus dem Sicherheitsmanagement verstehen (K2, 15min)

Für Security relevante Grundbegriffe aus dem Sicherheitsmanagement sind:

- Gefahr
- Gefährdung: Szenario aus Funktionaler Sicherheit
- Bedrohung
- Risiko
- Schwachstelle
- Eintrittswahrscheinlichkeit
- Schadenspotential
- Bedrohungsszenarien Cyber Security

1.2.3 LO: Notwendigkeit von Standards und Regelwerke erkennen (K1, 5min)

Dies ist keine Normenschulung. Es existieren spezifische Standards und Complianceanforderungen, die beachtet werden müssen.

1.3 Ziele

1.3.1 LO: Die Security Triade kennen (K1, 5min)

Es existieren abstrakte Schutzziele, von denen konkrete Schutzziele abgeleitet werden. Zentral in der Security ist die CIA-Triade. Es existieren viele weitere Schutzziele, die nach Bedarf selbst definiert werden können.

- Vertraulichkeit: Kein unautorisierter Zugang ermöglicht
- Integrität: Keine unautorisierten Modifikationen ermöglichen
- Verfügbarkeit: Keine unautorisierten Modifikationen ermöglichen

1.3.2 LO: Die Bedeutung und das Verfahren des Secure SW Engineering verstehen (K2, 15 min)

SSE besteht aus verschieden Vorgehensweisen, die zum Teil in Wechselwirkungen mit der IT-Administration stehen.

- Vorgehensweisen:
- Aus regulatorischen Anforderungen,
- Unternehmenszielen und Projektzielen für SSE definieren,
- Qualitätsmerkmale ableiten, im Detail: IT-Sicherheit
- deren Priorisierung und
- Ableitung von Sicherheitsanforderungen und Architekturen verstehen
- Unterschied IT-Administration und SSE
- IT-Administration: Betrieb der Infrastruktur
- SSE: Erzeugen von Software.
- SSE kann Anforderungen an IT-Administration stellen (Bestimmungsgemäßer Gebrauch)

1.4 Attribut Datenschutz

1.4.1 LO: Den Begriff Datenschutz kennen (K1, 5min)

Es ist notwendig, das Sammeln von sensiblen Daten zu minimieren und diese richtig zu verwalten.

- Die DSGVO sieht empfindliche Geldbußen vor. Auch ohne den Anspruch an compliance zu einer Norm oder weitere Sicherheitsvorschriften liegen hier gesetzliche Zwänge vor.

1.4.2 LO: Begriffe der Datenschutz Vorschriften erkennen (K1, 5 min)

Die DSGVO führt explizit Grundsätze für die Verarbeitung personenbezogener Daten auf.

- Es gibt 6 Grundsätze in der DSGVO
- Die Beachtung der DSGVO ist gesetzlich zwingend und mit einem Bußgeld bewehrt.

1.4.3 LO: Die für Datenschutz wichtige Begriffe aufzählen können (K1, 5 min)

- Personally Identifiable Information (PII)
- Personal Health Information (PHI)
- Personal Financial Information (PFI)
- Privacy Policy

1.5 Unternehmensübergreifende Sicherheit

1.5.1 LO: Einbettung des konstruktiven SSE in Prozesse und Prozessbeteiligte verstehen (K2, 15min)

SSE kann nach verschiedenen Standards und Frameworks in Prozesse und Prozessbeteiligte eingebettet werden.

- Katalog-Bausteine und Sicherheitsstandards
- BSI IT-Grundschutz
- Aufgabenteilung in Lieferkette und Lebenszyklus
- Standards, Normen, Best Practices kennen

1.5.2 LO: Anwendbare Normen mit Bezug zur Administration nennen können (K1, 5min)

- ISO 27000 Serie
- BSI-Standards [5]
- BSI 200-1: [6] ISMS
- BSI 200-2: [7] IT-Grundschutz Methodologie
- BSI-200-3: [8] Risk Analysis based on IT-Grundschutz
- IEC 62443, ins besonders -2 und -3

1.5.3 LO: Anwendbare Normen mit Bezug zur Software-Entwicklung nennen können (K1, 5min)

- ISO 27034
- IEC 62443, insbesondere -4-1, -4-2 und -3-2
- ISO/SAE 21434
- IEC 60601-4-5 & IEC 81001-5-1
- GAMP

1.5.4 LO: Weitere Standards nennen können (K1, 5min)

- NIST Common Criteria
- PCI DSS
- ITIL

1.5.5 LO: Einbettung des konstruktiven SSE in Lieferketten verallgemeinern (K2, 15min)

Lieferketten sind wichtig, sowohl für Unternehmen als auch für Produkte. Verschiedene Normen und Frameworks beschreiben Vorgehensweisen zur Absicherung von Lieferketten. Eine Vorgehensweise sollte genutzt werden, um die Lieferkette abzusichern.

- Katalog-Bausteine und Sicherheitsstandards
- BSI IT-Grundschutz
- Aufgabenteilung in Lieferkette und Lebenszyklus
- Standards, Normen, Best Practices kennen

1.6 SW-Lifecycle und Prozesse – Ein Überblick

1.6.1 LO: Einbettung des SSE in Organisationen und in Prozesse beschreiben können (K2, 15 min)

Um die Ziele des SSE zu erreichen, muss SSE ein integraler Bestandteil der Unternehmenskultur und der Geschäftsprozesse werden.

- Erläuterung der Unterschiede zwischen Lifecycle und Prozess
- SW-Lifecycle (Software-Lebenszyklus) - beschreibt den Prozess der Softwareentwicklung mit dem Ziel der Bereitstellung einer Software für den Kunden.
- Prozess – Vorgehensmodell für die Realisierung des SW-Lifecycle – z. B. V-Modell, Spiralmodell, ….
- Ziele:
- Sicherheitseffektivität (security effectiveness)
- Sicherheitseffizienz
- Als Teil des Requirements Engineerings sind die Security Anforderungen zu erkennen
- Security Anforderungen sind entsprechend zu kennzeichnen und es sollte eine Schwere für den Fall der Verletzung festgelegt werden
- Auf Basis der Security Anforderungen sind die Security Tests zu spezifizieren.
- Das Vorgehen ist hier vergleichbar zum Vorgehen bei der Entwicklung von Safety-Funktionen – Safety-Anforderungen werden als Teil des Requirements-Engineerings ermittelt und einer Schwere/Kritikalität bewertet
- Wird Security Engineering in die Prozesse integriert, wird der Impact minimiert, Nachrüsten von Security ist schwierig, z. T. unmöglich, und hat einen deutlich höheren Aufwand.

1.6.2 LO: Zusammenhänge der wesentlichen Aktivitäten und Einflussfaktoren auf SSE im Lebenszyklus verstehen (K2, 15min)

- Der Software-Lebenszyklus besteht aus (Anforderung => Analyse und Design => Implementierung => Test => Inbetriebnahme, Betrieb und Wartung, => Deinstallation und Außerbetriebnahme) Charakterisierung der jeweiligen Lebenszyklusphasen

- Es gibt Möglichkeiten des Einsatzes von Aspekten des Security Engineerings innerhalb der Softwareentwicklung
- Nicht-technische Einflussgrößen (Finanzen, Ressourcen, Image) nehmen Einfluss auf die Behebung von Sicherheitslücken im Kontext des Software-Lebenszyklus.
- Die Entwicklung und Dokumentation nach einem Prozess, welcher Compliance (Regeltreue zu einer anzuwendenden Norm) umsetzt, ist eine Methode, mittels Secure Design und Secure Software Engineering zu gesicherter Software zu gelangen.

1.6.3 LO: IT-Sicherheit als Qualitätsmerkmal im Software-Lebenszyklus verstehen (K2, 15min)

- Security als zusätzliche Aktivitäten
- Security Aktivitäten beeinflussen bestehenden Entwicklungsprozesse.
- Security Aktivitäten werden vom Projektmanagement frühzeitig aufgerufen und unterstützt.
- Security Aktivitäten nutzen bestehende Prozesse für RE, Testing etc. und erweitern diese um Security-Elemente.
- Security Aktivitäten sollen allerdings keine bestehenden Prozesse duplizieren.
- Security Aktivitäten definieren bestimmte Anforderung und kontrollieren diese. Darum verbessert ein solcher den Reifegrad bestehender Prozesse.
- Security Aktivitäten können im Vorfeld bereits Vorsorgen, dass z. B. Patching und Vulnerability Management reibungsloser funktioniert.
- Die Security ist als zusätzliche Aktivitäten im Software-Lebenszyklus zu betrachten. Es sind damit verbunden, auch zusätzliche Tools und Methoden in den bestehenden Prozess zu integrieren. Es sollte kein zusätzlicher Prozess sein.

1.7 Reifegradmodelle

1.7.1 LO Offene Reifegradmodelle benutzen können (K3, 60 min)

- Am Beispiel OpenSamm eine Standortbestimmung der eigenen Prozesse durchführen
- Praxisbeispiel
- Foliensatz offiziell freigegeben und verwendbar.

2. Bedrohungsanalyse und Anforderungen

2.1 Allgemein

2.1.1 LO: Grundlagen des Requirements-Engineerings kennen (K1, 5min)

- RE dient der effizienten und fehlerarmen Entwicklung komplexer Systeme.
- Ziel ist, ein gemeinsames Verständnis über das zu entwickelnde System zwischen Auftragnehmer und Auftraggeber zu erreichen.
- Es existieren Requirementsmanager und Requirementsengineer hierfür
- Diese sollten nicht nur erfahren sein, sondern einen Ausbildungsnachweis haben.
- Ein Ablauf für RE ist bekannt.
- Die Grundtätigkeiten des RE sind bekannt.

2.1.2 LO: Qualitätsmerkmale Nachvollziehbarkeit, Testbarkeit und Umsetzbarkeit kennen (K1, 5min)

- Nachvollziehbarkeit bedeutet, dass sich jede Anforderung bilateral zurückverfolgen lässt.
- Jede Aufgabe ist mit den Zwischenschritte der Implementierung und den nötigen Testfällen verknüpft.
- Umsetzbarkeit bedeutet, dass es (technisch) möglich ist, die Anforderung zu realisieren
- Testbarkeit ist die Eigenschaft eine Anforderung, durch (automatische) Tests prüfbar zu sein oder statisch validiert zu werden.
- Es gibt weitere. Diese drei sind für SSE besonders wichtige Qualitätsmerkmale von Requirements

2.1.3 LO: Spezielle Bestimmungsmethoden für Anforderungen kennen (K1, 5min)

- z. B. Square , Common Criteria, OpenSamm, …

2.1.4 LO: Quellen weiterer Anforderungen kennen (K1, 5min)

- von Dritten
- Kundenanforderungen
- Unternehmensrichtlinien
- Best Practices
- Gesetzliche Vorgaben
- BSI-Gesetz (bzgl. Kritische Infrastrukturen)
- DSGVO (GDPR)
- Produktanforderungen
- Medizinprodukte
- aus dem eigenen Prozess
- Unternehmensrichtlinien

- Data classifications / Threat modeling
- Funktionale Spezifikation / Use cases
- Modellierung (z. B. Misuse cases)

2.1.5 LO: Verfolgbarkeitsketten/-graphen, Versionierung kennen (K1, 5min)

Es gibt professionelles Konfigurationsmanagement. Ein securer Prozess sollte diese Methoden nutzen. Bei der Erstellung, der Veränderung und Erfüllung von Anforderungen ist es nötig, diese zu versionieren. Verfolgbarkeitsketten/-graphen sind Methoden, die Auswirkungen von Änderungen sichtbar zu machen.

2.1.6 LO: Für SSE wichtige Qualitätsmerkmale kennen – mit Übung (K3, 60 min)

Neben den Security Qualitätsmerkmalen (Vertraulichkeit, Integrität, Nicht-Abstreitbarkeit, Nachvollziehbarkeit, Authentizität) gibt es einige Qualitätsmerkmale, die für das Erreichen von Security Zielen maßgeblich sind. Besonders zu beachten für Security-Stakeholder:

- Zuverlässigkeit
- Fehlertoleranz
- Änderbarkeit
- Analysierbarkeit
- Modifizierbarkeit

2.1.7 LO: Einbindung in bestehende Softwareentwicklung kennen (K1, 5min)

Verschiedene Standards empfehlen die Anwendung eines Softwareentwicklungsprozess. Es wird kein Prozess vorgeschrieben, damit SSE sich in bestehende Entwicklungsvorgänge integrieren kann.

- Methode schafft Compliance als prüfbare Metrik.
- Durch Einbindung in Prozesse werden Methoden wiederholbar.
- Methoden müssen nachvollziehbar und wiederholbar sein.

2.2 Begriffe

2.2.1 LO: Den Begriff Risiko kennen (K1, 5min)

Es gibt unterschiedliche Standards, damit auch unterschiedliche Definitionen des Begriffs Risiko, abhängig von z.B. Anwendungsfall.

- Hier: Risiko ist „Schwere des möglichen Schadens zusammen mit Wahrscheinlichkeit des Schadenseintritts“.
- Die Anwendung der Risikoanalyse ermöglicht Priorisierung von Gefährdungen und Aufwand von Maßnahmen.

2.3 Bedrohungsanalyse in der Designphase

2.3.1 LO: Definition von Modell und Gründe für modellbasiertes Vorgehen kennen (K1, 5min)

Modelle sind Abstraktionen des Software-Systems. Sie stellen bestimmte Aspekte in den Fokus. Dadurch werden z. B. Datenflüsse klar einschätzbar.

- Modelle sind vereinfachte Darstellungen der Realität.

2.3.2 LO: Diagramme zum Finden von Vertrauensgrenzen und Angriffsflächen konstruieren können (K2, 15 min)

Durch das Definieren von Vertrauensgrenzen wird Handlungsbedarf sichtbar.

- Angriffsflächen werden so sichtbar.
- Maßnahmen werden abgeleitet.
- Ein Diagramm kann ein einfaches Modell sein. (Kontext-, Zustands-, Use-Case-, Datenfluss-, …)

2.4 Methoden der Bedrohungs- und Risikoanalyse

2.4.1 Unterschiedliche Herangehensweisen verstehen (K2, 15min)

Eine Bedrohungsanalyse kann unterschiedlich fokussiert werden

- Asset (angreifbare Güter),
- Attacker (Wünsche/Ziele oder Vorgehensweisen des Angreifers),
- Vulnerabilities (welche Schwachstellen typischerweise ausgenutzt werden),
- Risk (Ursache der Risiken)

Diese Vorgehensweisen haben unterschiedliche Vor– und Nachteile. Blick auf

- Assets vernachlässigen Blickwinkel des Angreifers → Perspektivenwechsel ist wertvoll.
- Attacker sind zwangsläufig unvollständig.
- Vulnerability ist reaktiv.

2.4.2 LO: Verstehen, wie man schützenswerten Güter identifiziert und priorisiert (K2, 15min)

Zur Risikoanalyse gehören Risikoabschätzung (Identifikation, Einschätzung und Bewertung) und Risikobehandlung.

- In der Risikoanalyse wird eine Bedrohung aus einer Liste von Verwundbarkeiten und jeweiligen Ausnutzungs- und Eintrittswahrscheinlichkeiten eingeschätzt.
- Die Risikominderung besteht aus Maßnahmen, die das vorhandene Risiko einer Bedrohung reduzieren sollen.
- Das Restrisiko verbleibt nach Durchführung aller Risikomaßnahmen.

2.4.3 LO: Verschiedene Methoden der Bedrohungsanalyse kennen (K1, 5 min)

Verschiedene Methoden der Bedrohungsanalyse können angewendet werden. Die Auswahl entspricht der Neigung und Erfahrung der Anwender. Darum sollten verschiedene Methoden bekannt sein.

- Attack Trees
- CVSS
- Octave
- STRIDE
- Trike

2.4.4 LO: Verstehen des Einflusses von Sicherheitslücken auf die IT-Sicherheit bzgl. einzusetzender Methoden und Metriken (K2, 15 min)

Bei der Verwendung von 3rd Party Software und Komponenten ist ein Verständnis für Klassifikationen und Metriken gemeldeter Sicherheitslücken wichtig.

- CVE (Quelle für gemeldete Sicherheitslücken)
- CWE (Klassifizierungen von Sicherheitslücken)
- CVSS (Metrik)

2.4.5 LO: Eine Methode (Attack Trees) zur Bedrohungsanalyse anwenden können. (K3, 60 min)

- Übung am Beispiel.

2.4.6 Risikoanalyse als Priorisierungsmaßstab kennen (K1, 5 min)

Eine Risikoanalyse liefert Metriken zur Gefährdungsbeurteilung. Diese dienen dazu, die gefundenen Risiken zu kategorisieren, klassifizieren und priorisieren. Dies ist ein wichtiger Input für eine Entscheidungsfindung bezüglich Maßnahmen.

- Aus Risikoanalyse folgen Security Requirements.
- Minderung des Risikos durch Maßnahmen
- Erneute Bewertung des Restrisikos, sowie Akzeptanz des verbleibenden Restrisikos.

3. Engineering & Architektur

3.1 Konzepte

3.1.1 LO: Ansätze und Methodik kennen (K1, 5min)

Ansätze wie Design Prinzipien ermöglichen es in Architektur und Design bewährte Vorgehensweisen zum Erreichen des Security Requirements einzusetzen. Zu Methoden wie Design Patterns und Coding Guidelines gibt es vorgefertigte Anwendungen für Security:

- Secure Design Principles
- Minimize Attack Surface
- Establish Secure Defaults
- Principle: Least Privilege
- Principle: Defense in Depth
- Keep Security Simple
- Secure Design Patterns
- Secure Coding

3.1.2 LO: Bedeutung der Softwarearchitektur verstehen (K2, 15min)

Definition für **Softwarearchitektur:** Strukturierte oder hierarchische Anordnung der Systemkomponenten sowie Beschreibung ihrer Beziehungen.

Architektur trifft grundlegende Entscheidungen über ein Software-System. Dies ist vor Implementierungsbeginn notwendig, um bestimmte Security Ziele erreichbar zu machen. Fehler an dieser Stelle sind im späteren Verlauf des Produktlebenszyklus möglicherweise nicht oder nur sehr teuer zu behandeln.

3.1.3 LO: Moderne Softwarearchitektur verstehen (K2, 15min)

Ein risikobasierter Ansatz muss angewendet werden. Dies ist bei der Auswahl von Architekturmustern und im Design bei Technologien und Methoden wichtig, um Security zu konstruieren. Moderne SW-Architektur basiert auf bewussten Entscheidungen über Modularisierung und Verwendung von Plattformen, um Wartbarkeit zu erreichen.

- Ziel von Moderner Softwarearchitektur:
- Verbesserung bzgl. Qualitätsmerkmale wie Anpassbarkeit, Wartbarkeit, Änderbarkeit gegenüber z.B. Monolithen.
- Moderne Softwarearchitektur erlaubt Analysierbarkeit und sollte Fehlertolerant sein.
- Angriffe/Fehler sollten erkannt werden und Verfügbarkeitsanforderungen sollten erfüllt werden.
- Es gibt einen wichtigen Unterschied bezüglich Administrativen/Operative und Software-Engineering Maßnahmen.

3.2 Architektur

3.2.1 LO: Eigenschaften moderner Software-Architektur verstehen (K2, 15min)

Für Security sind können einige wichtige Eigenschaften in der Architektur verankert und genutzt werden.

- Monitoring
- Tracking
- Tracing

3.2.2 LO: Methoden moderner Software-Architektur kennen (K1, 5min)

Bestimmte Frameworks erleichtern die Automatisierung von Methoden. Einige Methoden sind Security-Spezifisch und werden hier genauer betrachtet:

- Microservices am Beispiel Docker und Kubernetes
- Intrusion Detection mittels Pattern Recognition
- Data Movement Encryption Methoden
- Key Management Methoden

3.2.3 LO: Verstehen des Messens von Privacy (K2, 15 min)

Daten und Informationen, die mit Menschen in Verbindungen gebracht werden können, sind nach der DSGVO besonders schützenswert.

- Bewertung des Wertes der Daten
- Bewertung der Maßnahmen des Schutzes

3.3 Design

3.3.1 LO: Modernes Sicherheitsdesign kennenlernen (K1 5min)

Die Überwachung von bestimmten Systemvariablen, wie z. B. die Dateigrößen oder Netzwerkverkehr kann Hinweise auf unberechtigten Zugriff geben.

- Das Design muss ermöglichen, dass Schnittstellen stetig überwacht werden. Secure Boot: Überprüfung der kommenden und vorangegangenen Ereignisse im Bootablauf.
- Beispiele für modernes Sicherheitsdesign sind zum Beispiel unter Schlagworten wie Zero Trust und Gegenmaßnahmen zu Tainted Input zu finden.

3.3.2 LO: Schnittstellenanalyse kennen (K1, 5min)

Schnittstellen stellen Zugriff auf Daten und Funktionen bereit. Die Analyse des Designs muss prüfen, dass nur autorisierte Zugriffe stattfinden.

- HW/SW Systeme können auf verschiedenen Wegen angegriffen werden.
- Nur benötigte Schnittstellen dürfen implementiert werden, da Schnittstellen die Angriffsfläche vergrößern. Nicht mehr benötigte Schnittstellen müssen entfernt werden.

3.3.3 LO: Den Sinn geeigneter Schutzmaßnahmen im Sicherheitsdesign kennen (K1, 5min)

- Geeignete Schutzmaßnahmen wirken präventiv
- Minimieren die Möglichkeit von Anwenderfehlern
- Sind bequem
- Sind verständlich

3.4 Techniken und Integration in Organisationen

3.4.1 LO: Vor- und Nachteile von Intrusion Detection kennen (K1, 5min)

Intrusion Detection bezeichnet Maßnahmen zur Überwachung, die bösartiges und versehentliches Fehlverhalten aufdecken.

- Vorgehensweisen
- Vorteile von Intrusion Detection
- Nachteile von Intrusion Detection

3.4.2 LO: Geschützte Datenablagen verstehen (K2, 15 min)

Jede gespeicherte Information und ausführbare Funktion können von einem Angreifer manipuliert sein. Geschützte Datenablagen ermöglichen einen hohen Schutz vor Auslesen und Manipulation.

- Wichtig für besonders sensiblen sicherheitsrelevanten Informationen.
- Umsetzung innerhalb eines besonders geschützten Hardware-Bereiches.

3.4.3 LO: Integration des konstruktiven SSE in Organisation verstehen (K2, 15min)

- Definition konstruktives SSE
- Trennung von Daten und Funktionen
- Zugriffsverwaltung
- Rollenkonzept
- organisatorische Einbettung von Sicherheitsmanagement

4. Security Testing

4.1 Grundwissen zum Softwaretest

4.1.1 LO: Motivation und Ziele im Software-Test kennen (K1, 5 min)

- Korrekte Umsetzung der Funktionalität auf Basis der (Security-) Anforderungen prüfen
- Bewertung und Erhöhung des Vertrauens durch Softwarequalität
- Umsetzung von Security-Zielen prüfen durch Security-Test
- Durch gezieltes, systematisches und effektives Vorgehen möglichst viele Fehlerwirkungen, Fehlertrends und Sicherheitsschwachstellen erkennen und nachweisen

4.1.2 LO: Grundwissen zum Softwaretest kennen (K1, 5 min)

- Definition Test → ISTQB® Certified Tester Foundation Level
- Anwendung unterschiedliche Testlevel
- Entwurfsmethoden für Testfälle (Äquivalenzklassenbildung, Grenzwertanalyse, …)
- Grundzüge des klassischen Softwaretests
- Ablauf und Schwerpunkte beim Test von Hardware – Security Prozessor, Smart Cards, …
- Fundamentaler Testprozess nach ISTQB®

4.1.3 LO: Statische und dynamische Tests kennen (K1, 5 min)

- Statische Tests ohne Programmausführung z. B.:
  - Statische Codeanalyse
  - Review
  - String Suche am Binary
  - Hash-Wert-Vergleich
- Dynamische Tests während der Programmausführung: z. B.:
  - Validierungstest
  - Fuzzing

4.1.4 LO: Black Box / White Box kennen (K1, 5 min)

- Definition Test → Certified Tester Foundation Level
- Unterschied Black-Box / White-Box Testing

4.2 Typische Angriffsverfahren aus Testsicht

4.2.1 LO: Typische Angriffswege aus Testsicht erkennen (K1, 5min)

- Darstellung typischer Angriffe aus Perspektive des Testers
- Code Modification durch Programm = Virus

- DLL Injection, DLL Hooking
- Man in the Middle
- Angriff durch automatisierte Tools

4.2.2 LO: Verstehen der typischen Fehler (K2, 15min)

Darstellung typischer Angriffe:

Code Modification mittels Virus oder DLL Hooking

Man in the Middle

Darstellung typischer Fehler, die zu Security-Problemen führen:

Unzureichender Schutz von Passwörtern (Salting und Hashing)

Mangelnde Kapselung (Firewalls, Seitenwege, User Input Validation, Verhindern von z. B. SQL Injection)

Markante Beispiele aus der Historie → Sensibilisierung

4.3 Analyse der Security Architektur

4.3.1 LO: Verstehen von Methoden zur Schwachstellenanalyse von Architekturen (K2, 15min)

- Dynamische Analyse, händische Analyse einer Modelldarstellung
- Agentenbasiert durch zusätzlich installierte Software
- Information Gathering: Abgleich CWE, CVE und andere öffentlich zugängliche Quellen

4.3.2 LO: Systematisches Testing mittels Tools kennen lernen (K1, 5 min)

Die Vorteile vom Tooleinsatz beim methodenbasierten systematischen Testing kennen.

- Feedback based Application Security Testing (FAST)
- Es existieren fortgeschrittene Testverfahren für den Security-Bereich (z. B. Fuzzing, Genetische Algorithmen, Software Modul Test …)
- Es existieren umfangreiche Informationen (Testverfahren, Tools, Herangehensweisen) beim BSI. S. Anhang, Darstellung verschiedener Arten von Werkzeugen und Nennung von Beispielen (z.B. Valgrind)
- 100%ige Code/Branch- Abdeckung, Randomisierte Eingabeüberprüfung wie Fuzzing sind, Compliance-Check zu Regelwerken und Coding-Guidelines z. B. durch manuelle Testverfahren nicht zu leisten.
- Quellen für Testing Frameworks kennen: OWASP, SP800, BSI

4.3.3 LO: Statische Analyse Techniken kennen (K1, 5 min)

- Vorgehensweise der statischen Analyse
- Ergebnisse der statischen Analyse
- Auswahl von Techniken
- Toolfamilien für Security Testing

4.3.4 LO: Dynamische Analyse Techniken kennen (K1, 5 min)

- Klassische Blackbox Tests
- Fehlerinjektion Test
- Prüfung der Verschlüsselung, Herkunft des Schlüssels
- Übereinstimmung mit Model

4.4 Erinnerung: Bewertung und Nachweis der IT-Sicherheit

4.4.1 LO: Bewertung und Nachweis der IT-Sicherheit kennen (K1, 5min)

- Risikoanalyse und Metriken
- Schemas und Vorgehensweise der Common Criteria
- Begriffe: ToE, EAL, SFRs, SARs

5. Lifecycle & Prozesse

5.1 Deployment & Betrieb

5.1.1 LO: DevOps Zyklus kennen (K1 / 5min)

- CI/CD (Continuous Integration / Continuous Deployment) kennen
- Definition DevOps
- Konzept „Shift Left“ kennen (Frühe Fehlervermeidung, Fehlererkennung, Fehlerbehebung ist wirtschaftlicher)

5.1.2 LO: Typische Vorteile und Nachteile des automatisierten Testens im Lebenszyklus einer Software nennen können (K1 / 5 min)

- Automatisierte Tests und DevOps – Lebenszyklus
- Automatisiertes Testen von Schwachstellen, mittels "Digital Twin" und Test Stubs
- Automatisierung und geänderte /selbstverändernde/lernende Systeme, Aufwand der Automatisierung.

5.1.3 LO: Begriff Systemüberwachung kennen (K1, 5min)

- Definition System
- Bedeutung der kontinuierlichen Systemüberwachung
- Mechanismen für Systemüberwachung (SIEM, IDS, Malware Scanners)

5.1.4 LO: Rollen und Aufgaben in DevSecOps kennen (K1, 5min)

- Definition DevSecOps
- Bestandteile Deployment und Betrieb: Rollen, Komponenten, Aufgaben

5.1.5 LO: Bereitstellungsumgebung und automatisierte Auslieferung kennen (K1, 5 min)

- Bereitstellungsumgebung: Entwicklung, Build, Test, Qualitätssicherung, Staging, Produktiv
- Automatisierte Auslieferung und Integrierung

5.2 Deployment in Organisationen

5.2.1 LO: Definitionen zu Identitäten

- Identitäten aus Security-Sicht
- Authentisierung, Authentifizierung, Autorisierung

5.2.2 LO: Kernbegriffe von Deployment und Betrieb erklären können (K2, 15 min)

- Elemente einer Secure Operations Policy
- IaaS, SaaS, PaaS, Managed Services sowie CI/CD (Continuous Integration / Continuous Deployment) und Bezug zur IT-Sicherheit
- Testautomatisierung

5.2.3 LO: Die Notwendigkeit von Informationssicherheit im Deployment und Betrieb erkennen können (K2, 15min)

- Schutz der Prozessumgebung (Einsatzumgebung) als übergeordnetes Ziel
- Notwendigkeit die Software-Repositories und Buildumgebung auch zu schützen
- Informationssicherheit ist nicht gleich IT-Sicherheit:
- Dokumentation des bestimmungsgemäßen Gebrauchs.

5.3 Maßnahmen im Deployment

5.3.1 LO: Sicheres Deployment ausführen können (K3, 60 min).

Sicheres Deployment ist wichtig, um die Secure Design Principles zu erfüllen.

- Elemente von Sicherem Deployment auflisten, ihre jeweiligen Eigenschaften und Vorteile erklären und beispielhaft einer Bedrohung eine Sicherheitsmaßnahme zuordnen können
- Erklärung der Begriffe "Environment Hardening" und "Secure Defaults" anhand von Beispielen

5.3.2 LO: Unterschiede zwischen Zugangskontrollmethoden wiedergeben können (K1, 5 min)

- DAC Discretionary Access Control
- MAC Mandatory Access Control
- RBAC Role based Access Control

5.3.3 LO: Vorteile und Nachteile von Virtualisierung

- Hypervisor kontrolliert Anwendungen während der Ausführung
- Erhöhter Ressourcenverbrauch
- Techniken für die Gewährleistung der Integrität und Verfügbarkeit (z.B. Load-Balancing, Datenreplizierung, Redundanz)
- Hardware-basierte Technologien für Trusted Computing und für die Verwaltung von kryptographischem Material (wie TPM (Trusted Platform Module) and HSM (Hardware Security Module)
- Vermitteln des Konzepts von Root of Trust und Chain of Trust

5.4 Incident Response & Vulnerability Management

5.4.1 LO: Die Hauptmerkmale von Patch-Management und Software-Vulnerability-Management auflisten können (K2, 15min).

- Definition "Patch Management" und "Software Vulnerability Management"
- Inputs für SVM und benötigte Quellen (z.B. CERT-Berichte)
- Definition Patch, Wechselwirkung mit Security Eigenschaften von Software
- Aktivitäten für einen sicheren Rollout neuer Funktionen und Bug Fixes
- Software Vulerability Scanner/Schwachstellenscanner

5.4.2 LO: Incident Response als wichtigen Geschäftsprozess beim Betrieb von Software kennen (K1, 5min)

- Bestandteile einer Richtlinie für "Incident Response"
- Eigenschaften eines effektiven Response-Teams
- Motivation für Ursachenanalyse (z.B. Root-Cause-Analysis)
- Argumente für verantwortungsvolle Offenlegung von Sicherheitsproblemen

5.4.3 LO: Begriffe und Aktivitäten bezüglich Beschaffung und Außerbetriebnahme auflisten können (K1, 5min)

- Aktivitäten bei der Beschaffung von Software
- Definitionen ILM (Information Lifecycle Management, SLA (Service Level Agreement) und EoL (end of life)
- Maßgebliche Bestandteile einer EoL-Policy
- Möglichkeiten für den Umgang mit Daten z.B. ordnungsgemäße und gesetzeskonforme Vernichtung von Datenträgern

5.5 Team- und Organisations- Entwicklung.

5.5.1 LO: Fehlerkultur und Kritikfähigkeit veranschaulichen (K2, 15 min)

- Kultur für sicherheitsbewusstes Denken
- Qualifikation der Mitarbeiter
- Ressourcen, Tools und Methoden für Fehlerkultur
- Aktivitäten im Prozess bzgl. Fehlerkultur etablieren

5.5.2 LO: Security & Vulnerability Management kennen (K1, 5 min).

- Das SVM liefert Vorgehen, Methoden und Abläufe zur Schwachstellenanalyse auf organisatorischer Ebene.

5.5.3 LO: Sprint am Beispiel Scrum erkennen (K1, 5min)

- Ablauf und Events eines Sprints kennen
- Begriffe kennen: Product Backlog, Sprint Backlog, Backlog Item, User Stories, Inkrement, Sprints, Definition of Done
- Rollen nennen können: Scrum Master, Product Owner, Developer

5.6 Vorgehensmodelle

5.6.1 LO: Begriffe im Lebenszyklus kennen (K1, 5 min) (Erweiterung zu Kapitel 1.5)

- Unter Lebenszyklus wird je nach Kontext Unterschiedliches verstanden. Hier ein Überblick relevanter Lebenszyklus-Definitionen.
- Unterschied Safety Security
- Die Merkmale sicherheitsorientierter Vorgehensmodelle können benannt und innerhalb eines Beispielprozesses klar identifiziert und zugeordnet werden.

5.6.2 LO: Anwenden des Security Development Life Cycle (K3, 60 min)

- Der Referenzprozesses des Security Development LifeCycle und Abbildung der Spezialitäten auf sequenziellen und iterativen/agilen Vorgehensmodellen soll am Beispiel angewendet werden
- Anhand exemplarischer Prozesse die Unterschiede zwischen sequenziellen und iterativen Vorgehensmodellen kennen und die jeweiligen Aktivitäten des Security Engineerings innerhalb dieser Vorgehensmodelle verstehen.
- Verstehen des Security Development LifeCycle als Referenzprozess
- Übung am Beispiel.

Anhang

Im Folgenden werden mitgeltende Dokumente und Informationen aufgeführt.

A. Abkürzungen, Begriffe und Glossar

Siehe Glossar.

B. Lernziel / Kognitive Ebenen des Lernens (Nicht prüfungsrelevant)

Auszug aus [ISTQB 11]:
Die folgende Taxonomie für Lernziele bildet die Grundlage des Lehrplans. Jeder Inhalt wird entsprechend den zugeordneten Lernzielen geprüft.

Taxonomiestufe 1: Kennen (K1)

Der Lernende ruft im Gedächtnis gespeicherte Informationen (z.B. Begriffe, isolierte Fakten, Abfolgen, Prinzipien, Mittel und Wege) ab. Typische beobachtbare Leistungen sind erkennen, nennen,
bezeichnen.
Schlüsselworte: sich erinnern, erkennen, wiedergeben, kennen

Taxonomiestufe 2: Verstehen (K2)

Der Lernende begründet oder erläutert Aussagen zum Thema. Typische beobachtbare Leistungen sind
beschreiben, zusammenfassen, vergleichen, klassifizieren, begründen, erklären, Beispiele für Testkonzepte nennen.
Schlüsselworte: zusammenfassen, verallgemeinern, abstrahieren, klassifizieren, vergleichen, auf etwas übertragen, etwas gegenüberstellen, erläutern, interpretieren, übersetzen, darstellen, rückschließen, folgern, kategorisieren, Modelle konstruieren, erklären, Beispiele geben, begründen, verstehen

Taxonomiestufe 3: Anwenden (K3)

Der Lernende überträgt erworbenes Wissen auf gegebene neue Situationen oder wendet sie zur Problemlösung an. Typische beobachtbare Leistungen sind: ausführen, anwenden, beurteilen, ermitteln, entwerfen, analysieren.
Schlüsselworte: anwenden, einsetzen, ausführen, nutzen, Verfahren verstehen, Verfahren anwenden