Secure Software Engineering
Basic course syllabus

ASQF Secure Software Engineer SSE

Curriculum Version: V2.0

Last release: 06.12.2023

In this document, the masculine form is generally used for personal names and personal nouns. In the interests of equal treatment, the corresponding terms apply to all genders. The abbreviated form of language is used for better readability and does not imply any valuation.

Authors

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

We thank the following contributors for their contribution:

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

Change overview

Version	Date	Author	Comment
1.0	27.11.2018	Authors and reviewers	First released draft version
1.1-1.3	-%-	-%-	Interim results
1.4	15.02.2023	Authors and reviewers	Second released draft version
2.0	06.12.2023	Reviewers	update and revision

Learning Objectives

1.1.1 LO: Know the definition of Secure Software Engineering (SSE) (K1, 5 min)

1.2.1 LO: Know the classification of Safety / Functional Safety and Security / IT Security (K1, 5min)

1.2.2 LO: Know definitions of basic terms from safety management (K2, 15min).

1.2.3 LO: Recognize the need for standards and regulations (K1, 5min).

1.3.1 LO: Know the Security Triad (K1, 5min)

1.3.2 LO: Understand the meaning and process of Secure SW Engineering (K2, 15 min)

1.4.1 LO: Know the concept of privacy (K1, 5min)

1.4.2 LO: Know the main features of the GDPR (or similar BSI) regulations (K1, 5 min).

1.4.3 LO: Be able to list the terms important for data protection (K1, 5 min)

1.5.1 LO: Understand embedding of constructive SSE in processes and process participants (K2, 15min).

1.5.2 LO: Be able to name applicable standards with reference to administration (K1, 5min).

1.5.3 LO: Be able to name applicable standards related to software development (K1, 5min).

1.5.4 LO: Be able to name further standards (K1, 5min)

1.5.5 LO: Generalize embedding of constructive SSE in supply chains (K2, 15min).

1.6.1 LO: Being able to describe the embedding of SSE in organizations and in processes (K2, 15 min)

1.6.2 LO: Understand interrelationships of key activities and influencing factors on SSE in the life cycle (K2, 15min).

1.6.3 LO: Understand IT security as a quality feature in the software life cycle (K2, 15min)

1.7.1 LO Being able to use open maturity models

2.1.1 LO: Understanding the basics of requirements engineering (K2, 15min)

2.1.2 LO: Know quality characteristics traceability, testability and feasibility (K1, 5min)

2.1.3 LO: Know special determination methods for requirements (K1, 5min).

2.1.4 LO: Know sources of further requirements (K1, 5min)

2.1.5 LO: Know traceability chains/graphs, versioning (K1, 5min).

2.1.6 LO: Know quality characteristics important for SSE - with exercise (K3, 60 min)

2.1.7 LO: Know integration into existing software development (K1, 5min).

2.2.1 LO: Know the concept of risk (K1, 5min)

2.3.1 LO: Know definition of model and rationale for model-based approach (K1, 5min).

2.3.2 LO: Be able to construct diagrams to find confidence limits and attack surfaces (K2, 15 min)

2.4.1 Understanding different approaches (K2, 15min)

2.4.2 LO: Understand how to identify and prioritize assets worth protecting (K2, 15min).

2.4.3 LO: Know different methods of threat analysis (K1, 5 min).

2.4.4 LO: Understand the impact of vulnerabilities on IT security in terms of methods and metrics to be used (K2, 15 min).

2.4.5 LO: Be able to apply a threat analysis method (Attack Trees). (K3, 60 min)

2.4.6 Know risk analysis as a prioritization standard (K1, 5 min)

3.1.1 LO: Understanding approaches and methodology (K2, 15min)

3.1.2 LO: Understand the importance of software architecture (K2, 15min)

3.1.3 Understanding modern software architecture (K2, 15min)

3.2.1 Understand characteristics of modern software architecture (K2, 15min)

3.2.2 Understand methods of modern software architecture (K2, 15min)

3.2.3 LO: Understanding the measurement of privacy (K2, 15 min)

3.3.1 LO: Learn about modern safety design (K1 5min)

3.3.2 LO: Know interface analysis (K1, 5min)

3.3.3 LO: Know the meaning of appropriate protective measures in safety design (K1, 5min).

3.4.1 LO: Know advantages and disadvantages of intrusion detection (K1, 5min)

3.4.2 LO: Understanding protected data stores (K2, 15 min)

3.4.3 LO: Understand embedding of constructive SSE in organization (K2, 15min).

4.1.1 LO: Know motivation and goals in software testing (K1, 5 min)

4.1.2 LO: Know basic knowledge about software testing (K1, 5 min)

4.1.3 LO: Know static and dynamic tests (K1, 5 min)

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

4.2.1 LO: Understand the typical ways of attack (K2, 15min)

4.3.1 LO: Understand methods for vulnerability analysis of architectures (K2, 15min).

4.3.2 LO: Learn about systematic testing using tools (K1, 5 min)

4.3.3 LO: Know static analysis techniques (K1, 5 min)

4.3.4 LO: Know dynamic analysis techniques (K1, 5 min)

4.3.5 LO: Learn about systematic testing using tools (K1, 5 min)

4.4.1 LO: Know assessment and evidence of IT security (K1, 5min).

5.1.1 LO: Know DevOps cycle (K1 / 5min)

5.1.2 LO: Being able to name typical advantages and disadvantages of automated testing in the life cycle of a software (K1 / 5 min)

5.1.3 LO: Know term system monitoring (K1, 5min)

5.1.4 LO: Know roles and tasks in DevSecOps (K1, 5min).

5.1.5 LO: Know deployment environment and automated delivery (K1, 5 min)

5.2.1 LO: Definitions of identities

5.2.2 LO: Being able to explain core concepts of deployment and operation (K2, 15 min)

5.2.3 LO: Be able to recognize the need for information security in deployment and operation (K2, 15min).

5.3.1 LO: Be able to perform secure deployment (K3, 60 min).

5.3.2 LO: Be able to reproduce differences between access control methods (K1, 5 min)

5.3.3 LO: Advantages and disadvantages of virtualization

5.4.1 LO: Be able to list the main features of patch management and software vulnerability management (K2, 15min).

5.4.2 LO: Know incident response as an important business process in the operation of software (K1, 5min).

5.4.3 LO: Be able to list terms and activities related to procurement and decommissioning (K1, 5min).

5.5.1 LO: Illustrate error culture and critical faculties (K2, 15 min)

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

5.5.3 LO: Recognize Sprint using Scrum as an example (K2 / 15min)

5.6.1 LO: Understand that there are different life cycle models (extension to chapter 1.5)

5.6.2 LO: Know life cycle terms (K1, 5 min)

5.6.3 LO: Apply the Security Development Life Cycle (K3, 60 min)

Taxonomy level 1: Know (K1)

Taxonomy level 2: Understanding (K2)

Taxonomy level 3: Apply (K3)

Curriculum content

1. Basic understanding of SSE and its goals
2. Threat analysis and requirements
3. Engineering & Architecture
4. Security Testing
5. Lifecycle & Processes

Business Outcomes

The business benefits (business outcomes) of participating in training based on this curriculum to the participant and their organization are as follows:

gains an understanding of the need for and benefits of security

Can define safety requirements

Can implement safety requirements

Understands processes/concepts/methods and can assist in their implementation

expands his competence and brings this knowledge in a targeted manner

Target group

All stakeholders involved in the SW development process, for example:

Computer scientists and engineers

SW architects and developers

Development and test teams

Quality representatives and project managers

Secure Software Engineering is the application of disciplined, quantifiable approaches to the development, operation, and maintenance of secure software using systematic methods.

1. Basic understanding of SSE and its goals

1.1 Definition SSE

1.1.1 LO: Know the definition of Secure Software Engineering (SSE) (K1, 5 min).

The goal of SSE is to write software that, according to requirements and current knowledge, does not expose vulnerabilities to an attacker.

- **Secure Software Engineering**

is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of secure software according to accepted and repeatable methods.

1.2 Meaning of the basic terms from safety management

1.2.1 LO: Know the classification of Safety / Functional Safety and Security / IT Security (K1, 5min).

Safety and security (functional safety / cybersecurity) have similarities and differences, some of which must be considered simultaneously and others in parallel.

- Safety (functional safety) protects people and the environment from the device.
- Security protects device from manipulation
- Safety demands security

1.2.2 LO: Know definitions of basic terms from safety management (K2, 15min).

Basic terms from security management that are relevant for security are:

- Danger
- Hazard: Scenario from functional safety
- Threat
- Risk
- Vulnerability
- Probability of occurrence
- Damage potential
- Threat scenarios Cyber Security

1.2.3 LO: Recognize the need for standards and regulations (K1, 5min).

This is not standard training. There are specific standards and compliance requirements that must be followed.

1.3 Targets

1.3.1 LO: Know the Security Triad (K1, 5min)

There are abstract protection goals from which concrete protection goals are derived. The CIA triad is central to security. There are many other protection goals that can be defined as required.

- Confidentiality: No unauthorized access allowed
- Integrity: Do not allow unauthorized modifications

- Availability: Do not allow unauthorized modifications

1.3.2 LO: Understand the meaning and process of Secure SW Engineering (K2, 15 min).

SSE consists of various procedures, some of which interact with IT administration.

- Procedures:

- From regulatory requirements,
- Define business goals and project objectives for SSE,
- Derive quality characteristics, in detail: IT security
- their prioritization and
- Understand derivation of security requirements and architectures

- Difference IT administration and SSE

- IT administration: operating the infrastructure
- SSE: Generate software.
- SSE may impose requirements on IT administration (Intended use)

1.4 Attribute privacy

1.4.1 LO: Know the concept of data protection (K1, 5min)

It is necessary to minimize the collection of sensitive data and manage it properly.

- The GDPR provides for severe fines. Even without the claim of compliance to a standard or further security regulations, legal constraints are present here.

1.4.2 LO: Recognize terms of data protection regulations (K1, 5 min)

The GDPR explicitly lists principles for the processing of personal data.

- There are 6 principles in the GDPR
- Compliance with the GDPR is mandatory by law and punishable by a fine.

1.4.3 LO: Being able to list the terms that are important for data protection (K1, 5 min)

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

1.5 Cross-company security

1.5.1 LO: Understand embedding of constructive SSE in processes and process participants (K2, 15min).

SSE can be embedded in processes and process participants according to various standards and frameworks.

- Catalog modules and safety standards
- BSI IT basic protection
- Division of tasks in supply chain and life cycle
- Know standards, norms, best practices

1.5.2 LO: Be able to name applicable standards related to administration (K1, 5min).

- ISO 27000 series
- BSI standards [5]

- BSI 200-1: [6] ISMS
- BSI 200-2: [7] IT-Grundschutz Methodology
- BSI-200-3: [8] Risk Analysis based on IT-Grundschutz
- IEC 62443, especially -2 and -3

1.5.3 LO: Be able to name applicable standards related to software development (K1, 5min).

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

1.5.4 LO: Be able to name other standards (K1, 5min).

- NIST Common Criteria
- PCI DSS
- ITIL

1.5.5 LO: Generalize embedding of constructive SSE in supply chains (K2, 15min).

Supply chains are important, both for companies and for products. Various standards and frameworks describe procedures for securing supply chains. One approach should be used to secure the supply chain.

- Catalog modules and safety standards

- BSI IT basic protection

- Division of tasks in supply chain and life cycle
- Know standards, norms, best practices

1.6 SW Lifecycle and Processes - An Overview

1.6.1 LO: Be able to describe the embedding of SSE in organizations and in processes (K2, 15 min).

To achieve the goals of SSE, SSE must become an integral part of the company's culture and business processes.

- Explanation of the differences between lifecycle and process

- SW lifecycle (software lifecycle) - describes the process of software development with the aim of providing a software to the customer.
- Process - procedure model for the realization of the SW lifecycle - e.g. V-model, spiral model, ...

- Objectives:

- Security effectiveness
- Safety efficiency

- As part of the requirements engineering, the security requirements are to be recognized

- Security requirements should be marked accordingly and a severity should be set for the case of violation
- The security tests must be specified on the basis of the security requirements.
- The procedure here is similar to the procedure for the development of safety functions - safety requirements are determined as part of the requirements engineering and evaluated for severity/criticality.

- If security engineering is integrated into the processes, the impact is minimized; retrofitting security is difficult, sometimes impossible, and requires significantly more effort.

1.6.2 LO: Understand interrelationships of key activities and influencing factors on SSE in the life cycle (K2, 15min).

- The software life cycle consists of (requirement => analysis and design => implementation => testing => commissioning, operation and maintenance, => decommissioning and decommissioning) Characterization of the respective life cycle phases
- There are possibilities of using aspects of security engineering within software development
- Non-technical influencing variables (finances, resources, image) affect vulnerability remediation in the context of the software lifecycle.
- Development and documentation according to a process that implements compliance (adherence to an applicable standard) is a method of achieving secure software by means of secure design and secure software engineering.

1.6.3 LO: Understand IT security as a quality feature in the software life cycle (K2, 15min).

- Security as additional activities

- Security activities influence existing development processes.
- Security activities are called and supported by project management at an early stage.
- Security activities use existing processes for RE, testing, etc. and extend them with security elements.
- Security activities, however, should not duplicate existing processes.

- Security activities define certain requirements and control them. Therefore, one improves the maturity level of existing processes.
- Security activities can take precautions in advance so that, for example, patching and vulnerability management function more smoothly.
- Security must be viewed as an additional activity in the software lifecycle. Associated with this, additional tools and methods must also be integrated into the existing process. It should not be an additional process.

1.7 Maturity models

1.7.1 LO Be able to use open maturity models (K3, 60 min)

- Using OpenSamm as an example, carry out a site assessment of your own processes
- Practical example
- Set of slides officially approved and usable.

2. Threat analysis and requirements

2.1.1 LO: Understand the basics of requirements engineering (K2, 15min).

- RE is used for the efficient and error-free development of complex systems.
- The goal is to achieve a common understanding of the system to be developed between the contractor and the client.
- Requirements managers and requirements engineers exist for this purpose
- These should not only be experienced, but have a training record.
- A procedure for RE is known.
- The basic activities of the RE are known.

2.1.2 LO: Know quality characteristics traceability, testability and feasibility (K1, 5min)

- Traceability means that every request can be traced back bilaterally.
- Each task is linked to the intermediate implementation steps and necessary test cases.
- Implementability means that it is (technically) possible to realize the requirement
- Testability is the property of a requirement to be testable by (automatic) tests or to be statically validated.
- There are others. These three are particularly important quality characteristics of requirements for SSE

2.1.3 LO: Know special determination methods for requirements (K1, 5min).

- z. E.g. Square , Common Criteria, OpenSamm, ...

2.1.4 LO: Know sources of further requirements (K1, 5min).

- From third parties

- Customer requirements
- Company policies
- Best Practices

- Legal requirements

- BSI Act (regarding critical infrastructures)
- DSGVO (GDPR)

- Product requirements

- Medical Devices

- from own process

- Company policies
- Data classifications / Threat modeling
- Functional specification / Use cases
- Modeling (e.g. misuse cases)

2.1.5 LO: Know traceability chains/graphs, versioning (K1, 5min).

There is professional configuration management. A secure process should use these methods. When creating, changing and fulfilling requirements, it is necessary to version them. Traceability chains/graphs are methods to make the effects of changes visible.

2.1.6 LO: Know quality characteristics important for SSE - with exercise (K3, 60 min).

In addition to the security quality characteristics (confidentiality, integrity, non-repudiation, traceability, authenticity), there are some quality characteristics that are decisive for the achievement of security goals. Particularly noteworthy for security stakeholders:

- Reliability
- Fault tolerance
- Modifiability
- Analyzability
- Modifiability

2.1.7 LO: Know integration into existing software development (K1, 5min).

Various standards recommend the use of a software development process. No process is prescribed so that SSE can integrate with existing development processes.

- Method creates compliance as an auditable metric.
- By integrating them into processes, methods become repeatable.
- Methods must be traceable and repeatable.

2.2 2.2 Terms

2.2.1 LO: Know the concept of risk (K1, 5min).

There are different standards, thus also different definitions of the term risk, depending on e.g. use case.

- Here: risk is "severity of possible harm together with probability of harm occurring".
- The application of risk analysis enables prioritization of hazards and effort of measures.

2.3 Threat analysis in the design phase

2.3.1 LO: Know definition of model and rationale for model-based approach (K1, 5min).

Models are abstractions of the software system. They focus on certain aspects. This makes it possible to clearly assess data flows, for example.

- Models are simplified representations of reality.

2.3.2 LO: Be able to construct diagrams to find confidence limits and attack surfaces (K2, 15 min).

Defining trust limits reveals the need for action.

- Attack surfaces thus become visible.
- Measures are derived.

- A diagram can be a simple model. (context, state, use case, data flow, ...)

2.4 Threat and risk analysis methods

2.4.1 Understanding different approaches (K2, 15min)

A threat analysis can be focused in different ways

- Asset (tangible goods),
- Attacker (desires/goals or approaches of the attacker),
- Vulnerabilities (which vulnerabilities are typically exploited),
- Risk (cause of the risks)

These approaches have different advantages and disadvantages. View

- Assets neglect attacker's point of view → perspective change is valuable.
- Attackers are inevitably incomplete.
- Vulnerability is reactive.

2.4.2 LO: Understand how to identify and prioritize assets worth protecting (K2, 15min).

Risk analysis includes risk assessment (identification, estimation and evaluation) and risk treatment.

- In risk analysis, a threat is assessed from a list of vulnerabilities and respective probabilities of exploitation and occurrence.
- Risk mitigation consists of actions designed to reduce the existing risk of a threat.
- The residual risk remains after all risk measures have been implemented.

2.4.3 LO: Know different methods of threat analysis (K1, 5 min).

Different methods of threat analysis can be applied. The selection corresponds to the inclination and experience of the users. That is why different methods should be known.

- Attack Trees
- CVSS
- Octave
- STRIDE
- Trike

2.4.4 LO: Understand the impact of vulnerabilities on IT security in terms of methods and metrics to be used (K2, 15 min).

When using 3rd party software and components, an understanding of classifications and metrics of reported vulnerabilities is important.

- CVE (source for reported security vulnerabilities)
- CWE (security vulnerability classifications)
- CVSS (Metric)

2.4.5 LO: Be able to apply a threat analysis method (Attack Trees). (K3, 60 min)

- Exercise by example.

2.4.6 Know risk analysis as a prioritization standard (K1, 5 min).

A risk analysis provides metrics for risk assessment. These are used to categorize, classify and prioritize the risks found. This is an important input for decision making regarding measures.

- Risk analysis is followed by security requirements.
- Mitigation of the risk through measures
- Re-evaluation of the residual risk, and acceptance of the remaining residual risk.

3. Engineering & Architecture

3.1 Concepts

3.1.1 LO: Understanding approaches and methodology (K2, 15min)

Approaches such as design principles make it possible to use best practices in architecture and design to achieve the security requirement. For methods such as design patterns and coding guidelines, there are ready-made applications for 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: Understand the importance of software architecture (K2, 15min).

Definition for **software architecture:** Structured or hierarchical arrangement of system components and description of their relationships.

Architecture makes fundamental decisions about a software system. This is necessary before the start of implementation in order to make certain security goals achievable. Errors at this point may be impossible or very expensive to deal with later in the product lifecycle.

3.1.3 LO: Understanding Modern Software Architecture (K2, 15min)

A risk-based approach must be applied. This is important in the selection of architectural patterns and in the design of technologies and methods to construct security. Modern software architecture is based on conscious decisions about modularization and use of platforms to achieve maintainability.

- Goal of Modern Software Architecture:

- Improvement regarding quality features like adaptability, maintainability, changeability compared to e.g. monoliths.
- Modern software architecture allows analyzability and should be fault tolerant.
- Attacks/failures should be detected and availability requirements should be met.
- There is an important difference regarding administrative/operational and software engineering measures.

3.2 Architektur

3.2.1 LO: Understand features of modern software architecture (K2, 15min).

For security, some important properties can be anchored and used in the architecture.

- Monitoring
- Tracking

- Tracing

3.2.2 LO: Understanding methods of modern software architecture (K2, 15min)

Certain frameworks facilitate the automation of methods. Some methods are security-specific and are examined in more detail here.

- Microservices using the example of Docker and Kubernetes
- Intrusion detection using pattern recognition
- Data Movement Encryption Methods
- Key Management Methods

3.2.3 LO: Understanding the measurement of privacy (K2, 15 min).

Data and information that can be linked to people are particularly worthy of protection under the GDPR.

- Evaluation of the value of the data
- Evaluation of the measures of protection

3.3 Design

3.3.1 LO: Learn about modern security design (K1 5min).

Monitoring certain system variables, such as file sizes or network traffic, can provide indications of unauthorized access.

- The design must allow interfaces to be constantly monitored.
- Examples of modern security design can be found, for example, under keywords such as Zero Trust and countermeasures to Tainted Input.

3.3.2 LO: Know interface analysis (K1, 5min).

Interfaces provide access to data and functions. The analysis of the design must check that only authorized accesses take place.

- HW/SW systems can be attacked in several ways.
- Only required interfaces may be implemented, since interfaces increase the attack surface. Interfaces that are no longer required must be removed.

3.3.3 LO: Know the meaning of appropriate protective measures in security design (K1, 5min).

- Suitable protective measures have a preventive effect
- Minimize the possibility of user error
- Are comfortable
- Are understandable

3.4 Techniques and integration in organizations

3.4.1 LO: Know the advantages and disadvantages of intrusion detection (K1, 5min).

Intrusion detection refers to monitoring measures that detect malicious and accidental misconduct.

- Procedures

- Advantages of Intrusion Detection
- Disadvantages of Intrusion Detection

3.4.2 LO: Understanding protected data stores (K2, 15 min)

Any stored information and executable function can be manipulated by an attacker. Protected data stores enable a high level of protection against readout and manipulation.

- Important for particularly sensitive security-related information.
- Implementation within a specially protected hardware area.

3.4.3 LO: Understand integration of constructive SSE in organization (K2, 15min).

- Definition constructive SSE
- Separation of data and functions
- Access management
- Role concept
- organizational embedding of safety management

4. Security Testing

4.1 Basic knowledge of software testing

4.1.1 LO: Know motivation and goals in software testing (K1, 5 min).

- Check correct implementation of functionality based on (security) requirements
- Evaluation and increase of trust through software quality
- Verify implementation of security objectives through security test
- Identify and prove as many error effects, error trends and safety vulnerabilities as possible through a targeted, systematic and effective approach

4.1.2 LO: Know basic knowledge about software testing (K1, 5 min).

- Definition Test → ISTQB® Certified Tester Foundation Level
- Application different test levels
- Design methods for test cases (equivalence class formation, limit value analysis, ...)
- Basic features of classical software testing
- Process and focus of testing hardware - security processor, smart cards, ...
- Fundamental test process according to ISTQB®

4.1.3 LO: Know static and dynamic tests (K1, 5 min).

- Static tests without program execution e.g.:
    - Static code analysis
    - Review
    - String search on binary
    - Hash value comparison
- Dynamic tests during program execution: e.g.:
    - Validation test
    - Fuzzing

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

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

4.2 Typical attack methods from a test perspective

4.2.1 LO: Understand typical attack routes (K2, 15min).

- Representation of typical attacks from the tester's perspective
- Code modification by program = virus
- DLL Injection, DLL Hooking
- Man in the Middle
- Attack through automated tools

4.2.2 LO: Understanding of the typical errors (K2, 15min).

Representation of typical attacks:

Code modification by means of virus or DLL hooking
Man in the Middle

Presentation of typical errors that lead to security problems:

Insufficient protection of passwords (salting and hashing)
Lack of encapsulation (firewalls, side paths, user input validation, preventing e.g. SQL injection)

Striking examples from history → Sensitization

4.3 Analysis of the security architecture

4.3.1 LO: Understand methods for vulnerability analysis of architectures (K2, 15min).

- Dynamic analysis
- Agent-based through additionally installed software
- Information Gathering: CVE and other publicly available sources

4.3.2 LO: Learn about systematic testing using tools (K1, 5 min)

- Advanced test methods exist for the security area (e.g. fuzzing, genetic algorithms, ...)
- Extensive information (test procedures, tools, approaches) exists at BSI. See appendix for a description of different types of tools and examples (e.g. Valgrind).
- 100% code/branch coverage, randomized input verification such as fuzzing are, compliance check to rule sets and coding guidelines e.g. by manual testing procedures cannot be provided.
- Know sources for testing frameworks: OWASP, SP800, BSI

4.3.3 LO: Know static analysis techniques (K1, 5 min).

- Procedure of the static analysis
- Results of the static analysis
- Technique selection
- Tool families for security testing

4.3.4 LO: Know dynamic analysis techniques (K1, 5 min).

- Classic black box tests
- Error injection test
- Verification of the encryption, origin of the key
- Conformity with model

4.3.5 LO: Learn about systematic testing using tools (K1, 5 min)

Know the benefits of using tools in method-based systematic testing.

- Feedback based Application Security Testing (FAST)
- Fuzzy Testing
- Genetic algorithms
- Software module test

4.4 Reminder: IT security assessment and proof

4.4.1 LO: Know IT security assessment and evidence (K1, 5min).

- Risk analysis and metrics
- Schemes and procedure of the Common Criteria
- Terms: ToE, EAL, SFRs, SARs

5. Lifecycle & Processes

5.1 Deployment & Operation

5.1.1 LO: Know DevOps cycle (K1 / 5min)

- Know CI/CD (Continuous Integration / Continuous Deployment)
- Definition DevOps
- Know the concept of "Shift Left" (early fault prevention, fault detection, fault correction is more economical).

5.1.2 LO: Being able to name typical advantages and disadvantages of automated testing in the life cycle of a software (K1 / 5 min)

- Automated testing and DevOps lifecycle

Automated testing of vulnerabilities, using "Digital Twin" and test stubs

- Automation and modified/self-modifying/learning systems, automation effort.
-

5.1.3 LO: Know the term system monitoring (K1, 5min).

- Definition system
- Importance of continuous system monitoring
- Mechanisms for system monitoring (SIEM, IDS, malware scanners)

5.1.4 LO: Know roles and tasks in DevSecOps (K1, 5min).

- Definition DevSecOps
- Components Deployment and Operation: Roles, Components, Tasks

5.1.5 LO: Know deployment environment and automated delivery (K1, 5 min).

- Deployment environment: development, build, test, quality assurance, staging, production
- Automated delivery and integration

5.2 Deployment in organizations

5.2.1 LO: Definitions on identities

- Identities from a security perspective
- Authentication, authentication, authorization

5.2.2 LO: Be able to explain core concepts of deployment and operation (K2, 15 min)

- Elements of a Secure Operations Policy
- IaaS, SaaS, PaaS, Managed Services as well as Continuous Integration and relation to IT Security
- Test automation

5.2.3 LO: Be able to recognize the need for information security in deployment and operations (K2, 15min).

- Protection of the process environment (operational environment) as the overriding goal
- Need to protect the software repositories and build environment as well
- Information security is not just IT security:
- Documentation of the intended use.

5.3 Deployment measures

5.3.1 LO: Be able to perform secure deployment (K3, 60 min).

Secure deployment is important to meet Secure Design Principles.

- List elements of secure deployment, explain their respective features and benefits, and be able to assign a security measure to a threat as an example.
- Explanation of the terms "Environment Hardening" and "Secure Defaults" using examples

5.3.2 LO: Be able to reproduce differences between access control methods (K1, 5 min).

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

5.3.3 LO: Advantages and disadvantages of virtualization

- Hypervisor controls applications during execution
- Increased resource consumption
- Techniques for ensuring integrity and availability (e.g. load balancing, data replication, redundancy)
- Hardware-based technologies for trusted computing and for the management of cryptographic material (such as TPM (Trusted Platform Module) and HSM (Hardware Security Module)
- Communicate the concept of Root of Trust and Chain of Trust.

5.4 Incident Response & Vulnerability Management

5.4.1 LO: Be able to list the main features of patch management and software vulnerability management (K2, 15min).

- Definition of "Patch Management" and "Software Vulnerability Management
- Inputs for SVM and required sources
- Definition patch, interaction with security properties of software
- Activities for a secure rollout of new features and bug fixes
-

5.4.2 LO: Know Incident Response as an important business process in the operation of software (K1, 5min).

- Components of an "Incident Response" policy
- Characteristics of an effective response team
- Motivation for root cause analysis (e.g. root cause analysis)
- Arguments for responsible disclosure of security issues

5.4.3 LO: Be able to list terms and activities related to procurement and decommissioning (K1, 5min).

- Activities in the procurement of software
- Definitions ILM (Information Lifecycle Management, SLA (Service Level Agreement) and EoL (end of life)
- Relevant components of an EoL policy
- Possibilities for handling data e.g. proper and legally compliant destruction of data media

5.5 Team and Organizational Development.

5.5.1 LO: Illustrate error culture and criticality (K2, 15 min).

- Culture for safety-conscious thinking
- Employee qualification
- Resources, tools and methods for error culture
- Establish activities in the process regarding error culture

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

- The SVM provides procedures, methods and processes for vulnerability analysis at the organizational level.

5.5.3 LO: Recognize Sprint using Scrum as an example (K2 / 15min)

- Know the process and events of a sprint
- Know terms: Product Backlog, Sprint Backlog, Backlog Item, User Stories, Increment, Sprints, Definition of Done.
- Roles you can name: Scrum Master, Product Owner, Developer

5.6 Process models

1.1.1 LO: Understand that there are different life cycle models (extension to chapter 1.5) (K1, 5 min)

- Life cycle is understood in different ways depending on the context. Here is an overview of relevant life cycle definitions.

5.6.1 LO: Know life cycle terms (K1, 5 min).

- Difference Safety Security
- The characteristics of safety-oriented process models can be named and clearly identified and assigned within an example process.

5.6.2 LO: Apply the Security Development Life Cycle (K3, 60 min)

- The reference process of the Security Development LifeCycle and mapping of the specialties on sequential and iterative/agile procedure models is to be applied by example
- Using exemplary processes, know the differences between sequential and iterative process models and understand the respective security engineering activities within these process models.
- Understanding the Security Development LifeCycle as a reference process
- Exercise by example.

Appendix

Applicable documents and information are listed below.

A. Abbreviations, terms and glossary

See glossary.

B. Learning Objective / Cognitive Levels of Learning (Not relevant to exam).

Excerpt from [ISTQB 11]:
The following taxonomy for learning objectives forms the basis of the curriculum. Each content is tested according to the assigned learning objectives.

Taxonomy level 1: Know (K1)

The learner retrieves information stored in memory (e.g., terms, isolated facts, sequences, principles, ways and means). Typical observable performances are recognize, name, designate.
Keywords: remember, recognize, recall, know

Taxonomy level 2: Understanding (K2)

The learner justifies or explains statements about the topic. Typical observable performances are describe, summarize, compare, classify, justify, explain, give examples of
Give examples of test concepts.
Key words: summarize, generalize, abstract, classify, compare, apply to something, contrast something, explain, interpret, translate, represent, infer, conclude, categorize, construct models, explain, give examples, justify, understand

Taxonomy level 3: Apply (K3)

The learner transfers acquired knowledge to given new situations or applies it to solve problems. problem solving. Typical observable performances are execute, apply, assess, determine, design, analyze.
Keywords: apply, employ, perform, use, understand procedures, apply procedures.