# Secure Software Engineering
# Curriculum for the basic course

ASQF Secure Software Engineer SSE

Syllabus Version: V2.1

Last release: 21/08/2024

In this document, the masculine form is generally used when referring to persons and personal nouns. Corresponding terms apply to all genders in the interests of equal treatment. The abbreviated form is intended to improve readability and does not imply any judgement.

## Authors

Vera Gebhardt

Sebastian Dengler

Dipl.-Eng. Olga Jaufman

Dr Thomas Fehlmann

Dr 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 would like to 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

Overview of changes

| Version | date | Author | Remark |
|---|---|---|---|
| 1.0 | 27.11.2018 | Authors and reviewers | First released version |
| 1.1-1.4 | -%- | -%- | Intermediate results |
| 2.0 | 21.04.2023 | Authors and reviewers | Released update and revision |
| 2.0 | 06.12.2023 | Reviewer | Updating and revision |
| 2.1 | 16.08.2024 | Reviewer | Update adjustment<br>of the LOs and revision |

# Table of contents


1. Basic understanding of SSE and its objectives ..... 11

1.1 Definition SSE ..... 11

1.1.1 LO: Know the definition of Secure Software Engineering (SSE) (K1, 5 min) ..... 11

1.2 Meaning of the basic terms from safety management ..... 11

1.2.1 LO: Understanding the categorisation of safety / functional safety and security / IT security (K2, 15min) ..... 11

1.2.2 LO: Definitions Understanding basic terms from safety management (K2, 15min) 11

1.2.3 LO: Recognising the need for standards and regulations (K1, 5min) ..... 11

1.3 Goals ..... 12

1.3.1 LO: Knowing the security triad (K1, 5min) ..... 12

1.3.2 LO: Understanding the meaning and process of Secure SW Engineering (K2, 15 min) 12

1.4 Data protection attribute ..... 12

1.4.1 LO: Know the term data protection (K1, 5min) ..... 12

1.4.2 LO: Recognise terms of data protection regulations (K1, 5 min) ..... 12

1.4.3 LO: List the terms that are important for data protection (K1, 5 min) ..... 13

1.5 Cross-company security ..... 13

1.5.1 LO: Understanding the embedding of constructive SSE in processes and process participants (K2, 15min) ..... 13

1.5.2 LO: Be able to name applicable standards with reference to administration (K1, 5min) 13

1.5.3 LO: Be able to name applicable standards related to software development (K1, 5min) 13

1.5.4 LO: Being able to name further standards (K1, 5min) ..... 13

1.5.5 LO: Generalise the embedding of constructive SSE in supply chains (K2, 15min) 14

1.6 SW lifecycle and processes - an overview ..... 14

1.6.1 LO: Be able to describe the embedding of SSE in organisations and processes (K2, 15 min)14

1.6.2 LO: Understanding the relationships between the main activities and factors influencing SSE in the life cycle (K2, 15min) ..... 14

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

1.7 Maturity models ..... 15
1.7.1 LO Being able to use open maturity models (K3, 60 min) ..... 15
2. Threat analysis and requirements ..... 16
2.1 General ..... 16
2.1.1 LO: Know the basics of requirements engineering (K1, 5min) ..... 16
2.1.2 LO: Know the quality characteristics of comprehensibility, testability and practicability (K1, 5min) ..... 16
2.1.3 LO: Know special determination methods for requirements (K1, 5min) ..... 16
2.1.4 LO: Know the sources of further requirements (K1, 5min) ..... 16
2.1.5 LO: Know traceability chains/graphs, versioning (K1, 5min) ..... 17
2.1.6 LO: Knowing important quality characteristics for SSE - with exercise (K3, 60 min) 17
2.1.7 LO: Knowing integration into existing software development (K1, 5min) ..... 17
2.2 Terms ..... 17
2.2.1 LO: Knowing the concept of risk (K1, 5min) ..... 17
2.3 Threat analysis in the design phase ..... 18
2.3.1 LO: Know the definition of a model and reasons for a model-based approach (K1, 5min) 18
2.3.2 LO: Be able to construct diagrams to find confidence limits and attack surfaces (K2, 15 min) 18
2.4 Methods of threat and risk analysis ..... 18
2.4.1 Understanding different approaches (K2, 15min) ..... 18
2.4.2 LO: Understand how to identify and prioritise assets worthy of protection (K2, 15min) 18
2.4.3 LO: Know different methods of threat analysis (K1, 5 min) ..... 19
2.4.4 LO: Understanding the impact of security vulnerabilities on IT security in terms of methods and metrics to be used (K2, 15 min) ..... 19
2.4.5 LO: Be able to apply a method (Attack Trees) to analyse threats. (K3, 60 min)19
2.4.6 Knowing risk analysis as a standard for prioritisation (K1, 5 min) ..... 19
3. Engineering & Architecture ..... 20
3.1 Concepts ..... 20
3.1.1 LO: Knowing approaches and methodology (K1, 5min) ..... 20
3.1.2 LO: Understanding the importance of software architecture (K2, 15min) ..... 20
3.1.3 LO: Understanding modern software architecture (K2, 15min) ..... 20
3.2 Architecture ..... 21
3.2.1 LO: Understanding the characteristics of modern software architecture (K2, 15min) ..... 21

3.2.2 LO: Knowing methods of modern software architecture (K1, 5min) ..... 21
3.2.3 LO: Understanding the measurement of privacy (K2, 15 min) ..... 21
3.3 Design ..... 21
3.3.1 LO: Get to know modern security design (K1 5min) ..... 21
3.3.2 LO: Knowing interface analysis (K1, 5min) ..... 21
3.3.3 LO: Know the purpose of suitable protective measures in security design (K1, 5min) 22
3.4 Techniques and integration in organisations ..... 22
3.4.1 LO: Know the advantages and disadvantages of intrusion detection (K1, 5min) ..... 22
3.4.2 LO: Understanding protected data storage (K2, 15 min) ..... 22
3.4.3 LO: Understanding the integration of constructive SSE in organisation (K2, 15min) ..... 22
4. Security Testing ..... 23
4.1 Basic knowledge of software testing ..... 23
4.1.1 LO: Knowing motivation and goals in software testing (K1, 5 min) ..... 23
4.1.2 LO: Knowing the basics of software testing (K1, 5 min) ..... 23
4.1.3 LO: Knowing static and dynamic tests (K1, 5 min) ..... 23
4.1.4 LO: Know Black Box / White Box (K1, 5 min) ..... 23
4.2 Typical attack methods from a test perspective ..... 23
4.2.1 LO: Recognise typical attack paths from a test perspective (K1, 5min) ..... 23
4.2.2 LO: Understanding the typical errors (K2, 15min) ..... 24
4.3 Analysing the security architecture ..... 24
4.3.1 LO: Understanding methods for analysing the vulnerability of architectures (K2, 15min) 24
4.3.2 LO: Getting to know systematic testing using tools (K1, 5 min) ..... 24
4.3.3 LO: Know static analysis techniques (K1, 5 min) ..... 24
4.3.4 LO: Know dynamic analysis techniques (K1, 5 min) ..... 25
4.4 Reminder: Assessment and proof of IT security ..... 25
4.4.1 LO: Knowing the evaluation and proof of IT security (K1, 5min) ..... 25
5. Lifecycle & processes ..... 26
5.1 Deployment & Operation ..... 26
5.1.1 LO: Know DevOps cycle (K1 / 5min) ..... 26
5.1.2 LO: Be able to name typical advantages and disadvantages of automated testing in the life cycle of software (K1 / 5 min) ..... 26
5.1.3 LO: Know the term system monitoring (K1, 5min) ..... 26
5.1.4 LO: Know roles and tasks in DevSecOps (K1, 5min) ..... 26

5.1.5 LO: Know the deployment environment and automated delivery (K1, 5 min) 26
5.2 Deployment in organisations ..... 26
5.2.1 LO: Definitions of identities ..... 26
5.2.2 LO: Be able to explain core concepts of deployment and operation (K2, 15 min) ... 26
5.2.3 LO: Recognise the need for information security in deployment and operation (K2, 15min) 27
5.3 Measures in deployment ..... 27
5.3.1 LO: Be able to perform secure deployment (K3, 60 min). ..... 27
5.3.2 LO: Be able to describe the differences between access control methods (K1, 5 min) 27
5.3.3 LO: Advantages and disadvantages of virtualisation ..... 27
5.4 Incident Response & Vulnerability Management ..... 27
5.4.1 LO: Be able to list the main features of patch management and software vulnerability management (K2, 15min). ..... 27
5.4.2 LO: Recognising incident response as an important business process in the operation of software (K1, 5min) ..... 28
5.4.3 LO: Be able to list terms and activities relating to procurement and decommissioning (K1, 5min) ..... 28
5.5 Team and organisational development ..... 28
5.5.1 LO: Illustrate error culture and ability to take criticism (K2, 15 min) ..... 28
5.5.2 LO: Knowing Security & Vulnerability Management (K1, 5 min). ..... 28
5.5.3 LO: Recognising a sprint using Scrum as an example (K1, 5min) ..... 28
5.6 Process models ..... 28
5.6.1 LO: Knowing terms in the life cycle (K1, 5 min) (extension to chapter 1.5) ..... 28
5.6.2 LO: Applying the Security Development Life Cycle (K3, 60 min) ..... 29
Appendix ..... 30
A. Abbreviations, terms and glossary ..... 30
B. Learning objective / Cognitive levels of learning (not relevant to the exam) ..... 30
Taxonomy level 1: Knowing (K1) ..... 30
Taxonomy level 2: Understanding (K2) ..... 30
Taxonomy level 3: Apply (K3) ..... 30

## Learning Objectives

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

1.2.1 LO: Understanding the categorisation of safety / functional safety and security / IT security (K2, 15min)

1.2.2 LO: Definitions Understanding basic terms from safety management (K2, 15min)

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

1.3.1 LO: Knowing 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 term data protection (K1, 5min)

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

1.4.3 LO: List the terms that are important for data protection (K1, 5 min)

1.5.1 LO: Understanding the 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: Generalise the embedding of constructive SSE in supply chains (K2, 15min)

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

1.6.2 LO: Understand the relationships between the main activities and factors influencing SSE in the life cycle (K2, 15min)

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

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

2.1.1 LO: Know the basics of requirements engineering (K1, 5min)

2.1.2 LO: Know the quality characteristics of comprehensibility, testability and realisability (K1, 5min)

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

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

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

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

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

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

2.3.1 LO: Know the definition of a model and reasons for a 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 prioritise assets worthy of protection (K2, 15min)

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

2.4.4 LO: Understanding the impact of security 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 method (Attack Trees) for threat analysis. (K3, 60 min)

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

3.1.1 LO: Know approaches and methodology (K1, 5min)

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

3.1.3 LO: Understanding modern software architecture (K2, 15min)

3.2.1 LO: Understanding the characteristics of modern software architecture (K2, 15min)

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

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

3.3.1 LO: Get to know modern security design (K1 5min)

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

3.3.3 LO: Know the purpose of suitable protective measures in security design (K1, 5min)

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

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

3.4.3 LO: Understanding the integration of constructive SSE in organisation (K2, 15min)

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

4.1.2 LO: Know the basics of software testing (K1, 5 min)

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

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

4.2.1 LO: Recognise typical attack routes (K1, 5min)

4.3.1 LO: Understanding methods for analysing the vulnerability of architectures (K2, 15min)

4.3.2 LO: Get to know 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)

4.4.1 LO: Knowing the evaluation and proof of IT security (K1, 5min)

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

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

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

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

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

5.2.1 LO: Definitions of identities

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

5.2.3 LO: Recognise 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 describe the differences between access control methods (K1, 5 min)

5.3.3 LO: Advantages and disadvantages of virtualisation

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

5.4.2 LO: Recognising 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 relating to procurement and decommissioning (K1, 5min)

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

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

5.5.3 LO: Recognising a sprint using Scrum as an example (K2 / 15min)

5.6.1 LO: Knowing terms in the life cycle (extension to chapter 1.5) (K1, 5 min)

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

Taxonomy Level 1: Knowing (K1)

Taxonomy level 2: Understanding

(K2) Taxonomy level 3: Applying (K3)

**Curriculum content**

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

**Business Outcomes**

The business outcomes for the participant and their organisation as a result of participating in training based on this curriculum are as follows:

- gains an understanding of the need for and benefits of security
- can define security requirements
- can implement safety requirements
- understands processes/concepts/methods and can support their introduction
- expands his expertise and makes targeted use of this knowledge

**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 officers and project managers

Secure software engineering is the application of disciplined, quantifiable approaches for the development, operation and maintenance of secure software using systematic methods.

# 1. Basic understanding of SSE and its objectives

## 1.1 Definition SSE

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

The aim of SSE is to write software that, according to requirements and current knowledge, does not make vulnerabilities available 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 recognised and repeatable methods.

## 1.2 Meaning of the basic terms from safety management

### 1.2.1 LO: Understanding the categorisation of safety / functional safety and security / IT security (K2, 15min)

Safety and security (functional safety / cyber security) have similarities and differences, some of which must be considered simultaneously and some in parallel. The concept of risk contains an assessment for a damage scenario.

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

### 1.2.2 LO: Definitions Understanding basic terms from safety management (K2, 15min)

Basic security management terms relevant to security are

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

### 1.2.3 LO: Recognise the need for standards and regulations (K1, 5min)

This is not a standards training course. There are specific standards and compliance requirements that must be observed.

## 1.3 Goals

### 1.3.1 LO: Knowing 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 unauthorised access possible
- Integrity: Do not allow unauthorised modifications
- Availability: Do not allow unauthorised 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 corporate goals and project goals for SSE,
- Derive quality features, in detail: IT security
- their prioritisation and
- Understanding the derivation of security requirements and architectures
- Difference between IT administration and SSE
- IT administration: operation of the infrastructure
- SSE: Creating software.
- SSE can place requirements on IT administration (intended use)

## 1.4 Data protection attribute

### 1.4.1 LO: Know the term data protection (K1, 5min)

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

- The GDPR provides for severe fines. Even without the requirement for compliance with a standard or further security regulations, there are legal constraints here.

### 1.4.2 LO: Recognise the terms of data protection regulations (K1, 5 min)

The GDPR explicitly sets out 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: 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: Understanding the 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.

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

### 1.5.2 LO: Be able to name applicable standards with reference 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, in particular -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 further standards (K1, 5min)

- NIST Common Criteria
- PCI DSS
- ITIL

### 1.5.5 LO: Generalise the 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.

- Catalogue modules and safety standards
- BSI IT baseline protection
- Division of tasks in the 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 organisations and processes (K2, 15 min)

In order to achieve the goals of SSE, SSE must become an integral part of the corporate culture and business processes.

- Explanation of the differences between lifecycle and process
- Software lifecycle - describes the process of software development with the aim of providing software for the customer.
- Process - Procedure model for the realisation of the SW lifecycle - e.g. V-model, Spiral model, ….
- Goals:
- Security effectiveness
- Safety efficiency
- The security requirements must be recognised as part of requirements engineering
- Security requirements must be labelled accordingly and a severity level should be defined in the event of a breach
- The security tests must be specified on the basis of the security requirements.
- The procedure here is comparable to the procedure for the development of safety functions - safety requirements are determined as part of requirements engineering and assessed for severity/criticality
- If security engineering is integrated into the processes, the impact is minimised; retrofitting security is difficult, sometimes impossible, and involves significantly more effort.

### 1.6.2 LO: Understand the relationships between the main activities and factors influencing SSE in the life cycle (K2, 15min)

- The software life cycle consists of (requirements => analysis and design => implementation => testing => commissioning, operation and maintenance, => deinstallation and decommissioning) Characterisation of the respective life cycle

phases

- There are ways of utilising aspects of security engineering within software development
- Non-technical factors (finances, resources, image) influence the elimination of security vulnerabilities in the context of the software life cycle.
- The 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: Understanding 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 for and supported by project management at an early stage.
- Security activities utilise existing processes for RE, testing etc. and expand these to include security elements.
- However, security activities should not duplicate existing processes.
- Security activities define certain requirements and monitor them. This is why it improves the maturity level of existing processes.
- Security activities can already ensure in advance that, for example, patching and vulnerability management function more smoothly.
- Security should be viewed as an additional activity in the software life cycle. It also involves integrating additional tools and methods 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)

- Use the example of OpenSamm to determine the status of your own processes
- Practical example
- Slide set officially approved and usable.

# 2. Threat analysis and requirements

## 2.1 General

### 2.1.1 LO: Know the basics of requirements engineering (K1, 5min)

- RE is used for the efficient and error-free development of complex systems.
- The aim 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
- They should not only be experienced, but also have proof of training.
- A procedure for RE is known.
- The basic activities of the RE are known.

### 2.1.2 LO: Know the quality characteristics of comprehensibility, testability and realisability (K1, 5min)

- Traceability means that every request can be traced bilaterally.
- Each task is linked to the intermediate implementation steps and the necessary test cases.
- Feasibility means that it is (technically) possible to realise 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 features 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 the sources of further requirements (K1, 5min)

- from third parties
- Customer requirements
- Company guidelines
- Best Practices
- Legal requirements
- BSI Act (regarding critical infrastructures)
- DSGVO (GDPR)
- Product requirements
- Medical devices
- from your own process
- Company guidelines

- Data classifications / Threat modelling
- Functional specification / use cases
- Modelling (e.g. misuse cases)

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

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

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

In addition to the security quality features (confidentiality, integrity, non-repudiation, traceability, authenticity), there are some quality features that are decisive for achieving security goals. Particular attention should be paid to security stakeholders:

- reliability
- Fault tolerance
- Modifiability
- Analysability
- Modifiability

#### 2.1.7 LO: Knowing 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 be integrated into existing development processes.

- method creates compliance as a verifiable metric.
- Methods become repeatable through integration into processes.
- Methods must be comprehensible and repeatable.

### 2.2 Terms

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

There are different standards, and therefore also different definitions of the term risk, depending, for example, on the application.

- Here: risk is "severity of the possible damage together with the probability of the damage occurring".
- The use of risk analysis enables the prioritisation of hazards and the cost of measures.

## 2.3 Threat analysis in the design phase

### 2.3.1 LO: Know the definition of a model and reasons for a 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 confidence limits reveals the need for action.

- Attack surfaces become visible.
- Measures are derived.
- A diagram can be a simple model. (context, state, use case, data flow, ...)

## 2.4 Methods of threat and risk analysis

### 2.4.1 Understanding different approaches (K2, 15min)

A threat analysis can be focussed in different ways

- Asset (tangible goods),
- Attacker (attacker's wishes/goals or methods),
- Vulnerabilities (which vulnerabilities are typically exploited),
- Risk (cause of the risks)

These approaches have different advantages and disadvantages. View of

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

### 2.4.2 LO: Understand how to identify and prioritise assets worthy of protection (K2, 15min)

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

- In the risk analysis, a threat is assessed from a list of vulnerabilities and the respective probabilities of exploitation and occurrence.
- Risk minimisation consists of measures 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)

Various methods of threat analysis can be used. The choice depends on the inclination and experience of the user. Various methods should therefore be known.

- Attack Trees
- CVSS
- Octave
- STRIDE
- Trike

### 2.4.4 LO: Understanding the impact of security 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 vulnerabilities)
- CWE (classifications of security vulnerabilities)
- CVSS (metric)

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

- Exercise using an example.

### 2.4.6 Know risk analysis as a standard for prioritisation (K1, 5 min)

A risk analysis provides metrics for risk assessment. These are used to categorise, classify and prioritise the risks identified. This is an important input for decision-making regarding measures.

- Security requirements follow from risk analysis.
- Risk minimisation through measures
- Reassessment of the residual risk and acceptance of the remaining residual risk.

# 3. Engineering & Architecture

## 3.1 Concepts

### 3.1.1 LO: Know approaches and methodology (K1, 5min)

Approaches such as design principles make it possible to use proven procedures in architecture and design to achieve the security requirements. Prefabricated applications for security are available for methods such as design patterns and coding guidelines:

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

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

Definition of **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 achieve certain security goals. Errors at this point may be impossible or very expensive to deal with later in the product life cycle.

### 3.1.3 LO: Understanding modern software architecture (K2, 15min)

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

- The goal of modern software architecture:
- Improvement with regard to quality features such as customisability, maintainability, changeability compared to e.g. monoliths.
- Modern software architecture allows analysability and should be fault-tolerant.
- Attacks/errors should be recognised and availability requirements should be met.
- There is an important difference between administrative/operational and software engineering measures.

## 3.2 Architecture

### 3.2.1 LO: Understanding the characteristics of modern software architecture (K2, 15min)

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

- Monitoring
- Tracking
- Tracing

### 3.2.2 LO: Know methods of modern software architecture (K1, 5min)

Certain frameworks facilitate the automation of methods. Some methods are security-specific and are analysed 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.

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

## 3.3 Design

### 3.3.1 LO: Get to know modern security design (K1 5min)

The monitoring of certain system variables, such as file sizes or network traffic, can provide indications of unauthorised access.

- The design must enable interfaces to be constantly monitored. Secure boot: Checking the upcoming and previous events in the boot process.
- Examples of modern security design can be found under buzzwords 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 authorised access takes place.

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

### 3.3.3 LO: Know the purpose of suitable protective measures in security design (K1, 5min)

- Suitable protective measures have a preventive effect
- Minimise the possibility of user errors
- Are comfortable
- Are understandable

## 3.4 Techniques and integration in organisations

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

Intrusion detection refers to monitoring measures that detect malicious and inadvertent misbehaviour.

- Procedures
- Advantages of intrusion detection
- Disadvantages of intrusion detection

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

Any stored information and executable function can be manipulated by an attacker. Protected data storage provides a high level of protection against reading and manipulation.

- Important for particularly sensitive security-relevant information.
- Realisation within a specially protected hardware area.

### 3.4.3 LO: Understanding the integration of constructive SSE in organisation (K2, 15min)

- Definition of constructive SSE
- Separation of data and functions
- Access management
- Role concept
- Organisational 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 the functionality based on the (security) requirements
- Evaluating and increasing trust through software quality
- Check implementation of security objectives through security testing
- Recognise 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 the basics of software testing (K1, 5 min)

- Definition Test → ISTQB® Certified Tester Foundation Level
- Application of different test levels
- Design methods for test cases (equivalence class formation, limit value analysis, …)
- Basics of classic software testing
- Procedure and focal points when testing hardware - security processor, smart cards, …
- Fundamental test process according to ISTQB®

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

- Static tests without programme execution, e.g:
  - Static code analysis
  - Review
  - String search on binary
  - Hash value comparison
- Dynamic tests during programme 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 between black box / white box testing

## 4.2 Typical attack methods from a test perspective

### 4.2.1 LO: Recognise typical attack paths from a test perspective (K1, 5min)

- Presentation of typical attacks from the tester's perspective
- Code modification by programme = virus

- DLL Injection, DLL Hooking
- Man in the Middle
- Attack through automated tools

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

Representation of typical attacks:

Code modification using a virus or DLL Hooking Man

in the Middle

Presentation of typical errors that lead to security problems: Inadequate

protection of passwords (salting and hashing)

Lack of encapsulation (firewalls, side paths, user input validation, prevention of e.g. SQL injection)

Striking examples from history → Sensitisation

### 4.3 Analysing the security architecture

#### 4.3.1 LO: Understanding methods for analysing the vulnerability of architectures (K2, 15min)

- Dynamic analysis, manual analysis of a model representation
- Agent-based through additionally installed software
- Information Gathering: Comparison of CWE, CVE and other publicly accessible sources

#### 4.3.2 LO: Get to know systematic testing using tools (K1, 5 min)

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

- Feedback based Application Security Testing (FAST)
- There are advanced test procedures for the security sector (e.g. fuzzing, genetic algorithms, software module test ...)
- Extensive information (test procedures, tools, approaches) is available from the BSI. See appendix, description of different types of tools and examples (e.g. Valgrind)
- 100% code/branch coverage, randomised input checks such as fuzzing, compliance checks for rules and coding guidelines cannot be achieved by manual test procedures, for example.
- Know sources for testing frameworks: OWASP, SP800, BSI

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

- Static analysis procedure
- Results of the static analysis
- Selection of techniques
- Tool families for security testing

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

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

### 4.4 Reminder: Assessment and proof of IT security

#### 4.4.1 LO: Knowing the evaluation and proof of IT security (K1, 5min)

- Risk analysis and metrics
- Schemes and procedures 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 of DevOps
- Know the "Shift Left" concept (early error prevention, error detection, Troubleshooting is more economical)

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

- Automated tests and DevOps life cycle
- 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)

- System definition
- 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 of DevSecOps
- Deployment and operation components: roles, components, tasks

### 5.1.5 LO: Know the 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 organisations

### 5.2.1 LO: Definitions of identities

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

### 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 and CI/CD (Continuous Integration / Continuous Deployment) and reference to IT security
- Test automation

### 5.2.3 LO: Recognise the need for information security in deployment and operation (K2, 15min)

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

## 5.3 Measures in deployment

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

Secure deployment is important in order to fulfil the Secure Design Principles.

- List elements of secure deployment, explain their respective characteristics and advantages 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 describe the 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 virtualisation

- 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))
- Communicating 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 (e.g. CERT reports)
- Definition of patch, interaction with security properties of software
- Activities for a secure rollout of new functions and bug fixes
- Software vulnerability scanner/vulnerability scanner

#### 5.4.2 LO: Recognising 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 in favour of responsible disclosure of security issues

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

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

### 5.5 Team and organisational development.

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

- Culture for safety-conscious thinking
- Qualification of employees
- Resources, tools and methods for error culture
- Establish activities in the process with regard to error culture

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

- The SVM provides procedures, methods and processes for analysing weaknesses at an organisational level.

#### 5.5.3 LO: Recognising a sprint using Scrum as an example (K1, 5min)

- Know the process and events of a sprint
- Know the terms: Product Backlog, Sprint Backlog, Backlog Item, User Stories, Increment, Sprints, Definition of Done
- be able to name roles: Scrum Master, Product Owner, Developer

### 5.6 Process models

#### 5.6.1 LO: Knowing terms in the life cycle (K1, 5 min) (extension to chapter 1.5)

- Life cycle is understood in different ways depending on the context. Here is an overview of relevant life cycle definitions.
- Difference Safety Security
- The characteristics of safety-orientated process models can be named and clearly identified and assigned within an example process.

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

- The reference process of the Security Development LifeCycle and the mapping of specialities on sequential and iterative/agile process models are to be applied using the following example
- Know the differences between sequential and iterative process models using exemplary processes and understand the respective activities of security engineering within these process models.
- Understanding the Security Development LifeCycle as a reference process
- Exercise using an 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 the exam)

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

### *Taxonomy level 1: Knowing (K1)*

The learner retrieves information stored in memory (e.g. concepts, isolated facts, sequences, principles, ways and means). Typical observable performances are recognising, naming, designate.
Keywords: remember, recognise, reproduce, know

### *Taxonomy level 2: Understanding (K2)*

The learner justifies or explains statements on the topic. Typical observable performances are describe, summarise, compare, classify, justify, explain, give examples of test concepts.
Keywords: summarise, generalise, abstract, classify, compare, transfer, contrast, explain, interpret, translate, represent, infer, deduce, categorise, 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. Typical observable performances are: performing, applying, assessing, determining, designing, analysing.
Keywords: apply, use, execute, utilise, understand procedure, apply procedure