ASQF®

Das Expertennetzwerk

QUALITY ENGINEERING FOR THE INTERNET OF THINGS (IOT QE) FOUNDATION LEVEL SYLLABUS

ASQF/GTB CERTIFIED PROFESSIONAL FOR IOT
FOUNDATION LEVEL

Syllabus Version 1.0E

31.1.2018

Copyright and User Rights

Authors

Aida Boukhris (Friedrich-Alexander-Universität Erlangen-Nürnberg), Vera Gebhardt (tecmata GmbH), Alexander Gladisch (T-Systems MMS GmbH), Daniel Hummel (IT-P GmbH), Günter Jung (imbus AG), Ralf Mack (Atos Information Technology GmbH), Matthias Pruksch (sepp.med GmbH), Axel Rennoch (Fraunhofer-Institut für Offene Kommunikationssysteme), Alfred Richter (DB Systel GmbH), Nils Röttger (imbus AG), Ina Schieferdecker (Fraunhofer-Institut für Offene Kommunikationssysteme / TU Berlin), Jessica Schiffmann, Günter Schneider (Sulzer GmbH)

Reviewers

Valeria Franzitta (Robert Bosch GmbH), Jan Markus Giesen (Brunel Car Synergies GmbH), Thomas Haase (T-Systems MMS GmbH), Helmut Pichler (Anecon GmbH), Frederik Teichert (HILSTER Testing Solutions GmbH)

Revision History

Version	Date	Author	Remarks
1.0	21.12.2017	Authors and reviewers	First released version (German)
1.0E	31.1.2018	Authors and reviewers	Translation to English

List of Contents

List of Learning Objectives

IoT-QE LO 1 (K1) Know what IoT means [5]

IoT-QE LO 2 (K1) Know that QE is highly relevant for IoT [5]

IoT-QE LO 3 (K2) Explain the specific aspects of IoT and the associated challenges for QE [15]

IoT-QE LO 4 (K2) Explain the relevance of QE in the context of IoT [15]

IoT-QE LO 5 (K1) Know how data is used to add value in IoT [10]

IoT-QE LO 6 (K2) Give an overview of the quality attributes which are relevant for IoT [10]

IoT-QE LO 7 (K2) Explain the quality attributes which are also relevant for IoT operations [15]

IoT-QE LO 8 (K1) Know the significance of standards and regulatory requirements [5]

IoT-QE LO 9 (K1) Know about functional quality attributes [5]

IoT-QE LO 10 (K2) Explain the security and safety challenges facing IoT systems [25]

IoT-QE LO 11 (K2) Explain the requirements for the interoperability of IoT systems [10]

IoT-QE LO 12 (K1) Explain the quality attributes robustness and resilience for IoT systems [10]

IoT-QE LO 13 (K2) Explain the requirements for the maintainability and portability of IoT systems [15]

IoT-QE LO 14 (K2) Explain the special challenges of performance efficiency quality attributes (time behavior und resource utilization) for IoT systems [10]

IoT-QE LO 15 (K2) Explain the relevance of ethical aspects for IoT [10]

IoT-QE LO 16 (K3) Evaluate the quality attributes of a system and derive requirements for the IoT system [60]

IoT-QE LO 17 (K1) Know about selected IoT reference architectures [10]

IoT-QE LO 18 (K2) Explain the elements of an IoT architecture using AIOTI as an example [15]

IoT-QE LO 19 (K2) Explain the layers of IoT architectures using AIOTI as an example [15]

IoT-QE LO 20 (K2) Explain the functions of layers in IoT architectures using AIOTI as an example [15]

IoT-QE LO 21 (K2) Explain the specific influence of data on IoT architectures [10]

IoT-QE LO 22 (K3) Construct an IoT system architecture from an IoT reference architecture [60]

IoT-QE LO 23 (K2) Explain constructive QE [15]

IoT-QE LO 24 (K1) Know the best practices in IoT [5]

IoT-QE LO 25 (K2) Explain DevOps for IoT [15]

IoT-QE LO 26 (K3) Explain the consequences of using the data-driven IoT business model [60]

IoT-QE LO 27 (K2) Explain the meaning of product and system variants for IoT [15]

IoT-QE LO 28 (K2) Explain the importance of QE for the operational phase of IoT systems [15]

IoT-QE LO 29 (K3) Perform an impact analysis of the IT security and safety quality attributes on constructive QE [60]

IoT-QE LO 30 (K2) Explain the trade-off between usability, maintainability and IT security [15]

IoT-QE LO 31 (K2) Explain the trade-off between resilience, robustness and performance [15]

IoT-QE LO 32 (K2) Explain the trade-off between connectivity, interoperability and IT security [15]

IoT-QE LO 33 (K1) Know the advantages of an agile approach [10]

IoT-QE LO 34 (K1) Know the advantages of automated approaches [10]

IoT-QE LO 35 (K1) Know the the need for monitoring during IoT system operations [spread across chapter 5]

IoT-QE LO 36 (K2) Explain the challenges of distributed tests for IoT systems [spread across chapter 5]

IoT-QE LO 37 (K2) Explain the special challenges of testing IoT solutions such as their openness, degree of distribution, changeability, scalability and variability [10]

IoT-QE LO 38 (K3) Define and prioritize test objectives for IoT [30]

IoT-QE LO 39 (K2) Explain the specific test levels for IoT [15]

IoT-QE LO 40 (K3) Prioritize test objectives according to their risk [30]

IoT-QE LO 41 (K2) Explain the test approach [15]

IoT-QE LO 42 (K2) Name the special aspects of testing IoT systems and give examples of IoT tests at different test levels [15]

IoT-QE LO 43 (K2) Explain the need for automating IoT tests [10]

IoT-QE LO 44 (K2) Explain the fundamental test process in the context of IoT [10]

IoT-QE LO 45 (K2) Explain the generic test architecture and the interaction and application of tools [20].

IoT-QE LO 46 (K2) Explain IoT test architectures and typical IoT test objects [15]

IoT-QE LO 47 (K2) Explain the principal aspects of IoT test automation architectures [15]

IoT-QE LO 48 (K2) Explain the usefulness and limitations of traditional testing techniques when applied to IoT systems [15]

IoT-QE LO 49 (K2) Explain the particular challenges for testing the security aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture [15]

IoT-QE LO 50 (K2) Explain the particular challenges of testing the interopability aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture [15]

IoT-QE LO 51 (K2) Explain the particular challenges for testing the performance aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture[15]

IoT-QE LO 52 (K2) Explain the challenges of checking for conformity and certification [15]

IoT-QE LO 53 (K1) Know the relevant standards and lifecycles in the context of IoT [15]

IoT-QE LO 54 (K2) Know and understand the focus of interrelated lifecycles within the IoT contex [15]

IoT-QE LO 55 (K2) Understand the interdisciplinary nature of the IoT lifecycle [15]

IoT-QE LO 56 (K2) Know the stakeholders in the IoT lifecycle and understand their significance [15]

IoT-QE LO 57 (K3) Understand the need to continue QE activities after rollout [30]

0 Introduction

Qualification scheme

This syllabus offers support to industry in the form of techniques and guidelines for the quality assurance and safeguarding of IoT solutions. It provides a de-facto standard qualification scheme and glossary. The syllabus was created on the basis of industry best practices by a working group of experts from the “Arbeitskreis Software Qualität und Fortbildung” (ASQF e.V.) and the German Testing Board (GTB e.V.). This working group will continually maintain the syllabus.

The subject of Internet of Things (IoT) opens up completely new possibilities for people and companies to simplify workflows, acquire information and offer services. At the same time it presents society and industry with major challenges. IoT is an area that is still developing; initial standards are under development but THE standard does not yet exist. In addition, the integration of IoT with other subjects such as Industry 4.0 and Big Data is still in its initial stages. The syllabus therefore provides an overview of the principles and approaches which align to the current status of trendsetting standardization initiatives.

Training courses may only be performed by training providers who are accredited by ASQF / GTB. Course participants may be examined by a certification body which is approved by ASQF / GTB. Those participants who pass the exam will be issued by the certification body with the qualification certificate “Certified Professional for IoT, Foundation Level”.

Business Outcomes

Based on this syllabus, training participants and their organizations benefit from the following business outcomes:

IoT-QE_BO01_Awareness: Understanding of the special challenges of QE in the context of IoT.

IoT-QE_BO02_Standards: Trouble-free cooperation within and together with IoT Teams enabled through knowledge of standards and use of a common glossary.

IoT-QE_BO03_Expertise: Application and mastery of QE in an IoT context through the transfer of “classic” QE expertise and the acquisition of specific QE expertise in IoT.

IoT-QE_BO04_QE-in-the-Organisation: Improvement of QE in an IoT context within the organization through the support of IoT teams and transfer of QE expertise within the organization.

IoT-QE_BO05_Personal-Development: Personal development of training course participants through the acquisition of expertise in a demanding subject with perspective for the future.

Learning objectives and cognitive levels of knowledge

The learning objectives in this syllabus are derived from the business outcomes. Each syllabus chapter contains learning objectives, each of which is allocated a cognitive knowledge level. Appendix C provides a description of the levels used in this syllabus.

Foundation level courses normally contain learning objectives with the following cognitive levels:

K1: Know

K2: Understand

K3: Apply

The cognitive level also has an influence on the training time provided (indicated as a number of minutes within brackets [ ]) and the type of possible exam questions.

1 Motivation [60]

Terms

Internet of Things, (IoT)	Definition of the ISO JTC 1/SWG 5: “An infrastructure of interconnected objects, people, systems and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react.” [ISO 14]
Digital Twin	The digital representation of a physical object, a process or a system in the virtual world. This representation includes both the structure and the dynamic behavior of the object over its entire lifecycle.

1.1 Quality Engineering the Internet of Things: What is it? (10 Min)

IoT-QE LO 1 (K1) Know what IoT means [5]

What is the Internet of Things?

The term IoT represents an infrastructure of interconnected objects (“things”), people, systems and other sources of information which enables them to process information from the physical and digital worlds and to react to it (see also ISO JTC 1/SWG 5).

In essence, IoT is neither a new nor a firmly defined technology with a fixed system definition. Internet of Things means that physical objects are increasingly capable of being represented, integrated and orchestrated in digital environments by a digital representation of themselves (known as a “Digital Twin”). This changes the technological basis and the application landscapes of software-based solutions.

IoT-QE LO 2 (K1) Know that QE is highly relevant for IoT [5]

Quality Engineering for IoT

The subject of IoT presents society and industry with a major challenge. Technical aspects such as architectures and components as well as sociological frameworks such as privacy, IT security, data protection and ethics all interact with each other. This demands a re-think concerning the priorities and different characteristics of technical approaches, quality criteria and processes. "Quality Engineering" is not the pure verification and validation of an implementation; it is the proactive assurance that quality criteria are achieved from the very first steps of development. This approach has established itself in many software development contexts and is of decisive importance for the Internet of Things.

1.2 Specific features of QE4IoT (30 Min)

IoT-QE LO 3 (K2) Explain the specific aspects of IoT and the associated challenges for QE [15]

IoT is characterized by:

- the combination of highly heterogeneous hardware and software elements and the large numbers of interacting components.
- local networking via intranet and global networking via the internet.
- a variety of different technologies and communications protocols at the application level and for interconnections.
- mobile end devices and sensors/actuators with a wide range of hardware resources and versions of operating systems.
- processing of volatile, heterogeneous data in large quantities.
- the dynamic changes in structures and components in a „living“ and open system.

- the horizontal and vertical integration required when using various platforms for edge and cloud computing.

This presents particular challenges for Quality Engineering with regard to:

- interdisciplinary working – the various technical domains demand close cooperation between specialists with different ways of working and levels of knowledge. They often don't speak the same technical language.
- the large variety of different equipment and variants – business cases must take into account standards conformity and interface agreements in the selection of suitable variants.
- complex operational scenarios – test scenarios often cannot even come close to covering the later operational situation.
- the lack of accessibility to equipment – end devices are often not accessible for an analysis or defect resolution.
- vulnerability to attacks – systems which are interconnected via the Internet are in general vulnerable to attacks and must be protected over their entire product lifecycles.
- strongly competing quality requirements – these are often in conflict with each other. For example, a high level of security may conflict with performance or the need for simple access.

What changes with the Internet of Things? – Quality Engineering for the Internet of Things / QE4IoT

IoT-QE LO 4 (K2) Explain the relevance of QE in the context of IoT [15]

As a result of their complexity and dynamic nature (see section 3.1), IoT architectures have a high level of criticality with regard to, for example, their security requirements. This places a high demand on constructive and analytical Quality Engineering.

Quality Engineering has the task of fully considering the quality attributes of a product across its entire lifecycle. This lifecycle extends from the conceptual stage to development, test and the series production and quality control. The monitoring of functions and services plays a role during the operations phase. Even operational retirement can be an issue for Quality Engineering since accessibility, data protection or environmental issues play a role. Effective quality planning and assurance prevent or detect defects early in the development process, in the production and in operation. This contributes significantly to the acceptance of the product and ultimately to the economic success of products and services.

Quality Engineering consists of both constructive and analytical quality activities:

- Constructive quality activities focus on avoiding defects right from the start. Orientation on best practices and standard architectures in development and production workflows and processes, as well as in the set-up of work environments enable experience from the market to be utilized. Additionally, the consideration of testability, the planning of quality checks during development and production, as well as the planning of the subsequent operational phase concerning required service levels all belong to constructive quality activities.
- Analytical quality activities focus on the early detection of defects. Static techniques such as reviews can be used to detect defects and check models, whereas dynamic quality assurance performs tests such as functional tests, load tests, acceptance tests, usability tests, security tests and penetration tests.

1.3 IoT business is data business – what does that mean for QE? (10 Min)

IoT-QE LO 5 (K1) Know how data is used to add value in IoT [10]

IoT systems enable new types of far-reaching data-driven business models via the Internet. By using the Internet as a communications platform, physical components in a network of sensors, actuators and both central and distributed systems can be connected to each other. This enables real-world data to be made available which is increasingly up to date, extensive, complete but also heterogeneous to business processes.

The available components and the type and quality of the data they provide will be highly dynamic in nature and not always be fully plannable. Analysis of this data can therefore lead to insights which could not previously have been predicted. This means that data has an major significance for the IoT business. The continuous adaption of products and solutions across the entire lifecycle, including operations, will be needed.

Aspects such as data aggregation, filtering and the entire complex issue of data protection must be considered in the planning and operation of IoT applications. This includes data which is personal, business-relevant and security-relevant as well as any data-relevant aspects which might impact on the protection or endangering of privacy.

1.4 The Smart Home Example (20 Min)

The intelligent networking of a family home (Smart Home) with its household equipment, entertainment electronics and additional sensors for lighting, thermostats, charging equipment or air conditioning has long been a typical application area for IoT. The various devices and instruments communicate their status and control information to a server via a local network such as a router within the house. This server can be set up within or outside the house and controlled by the house inhabitants using their end devices.

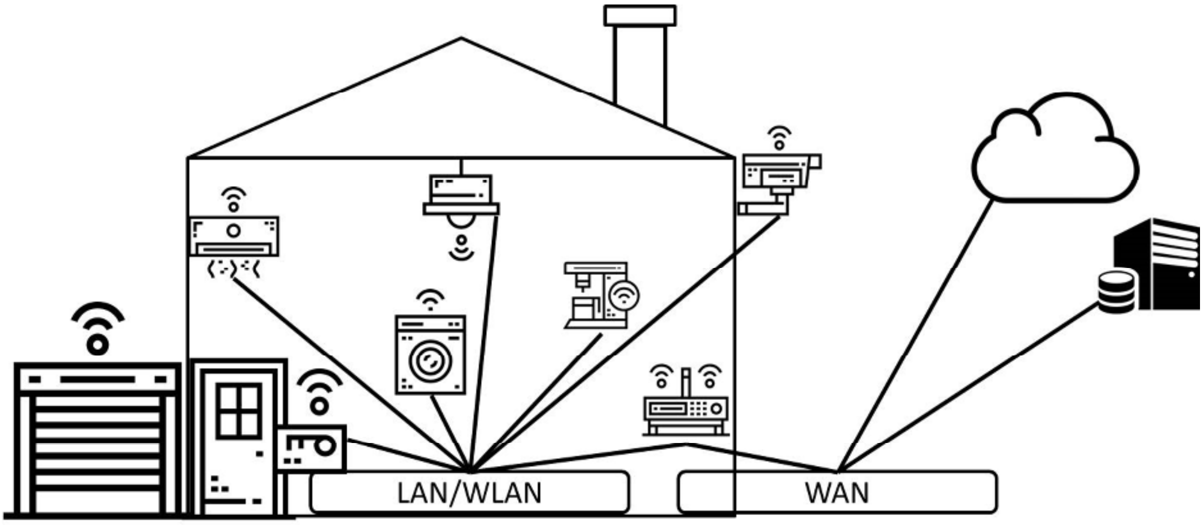

Diagram 1: Smart Home

Icons made by Freepik, samshizone & Retinaicons from www.flaticon.com

Applications like Smart Home may use their own messaging formats and transport protocols for the transfer of data from devices and sensors and for the control commands issued by the users. In order to process and construct data it is possible to differentiate between devices relating to the application and its users, the network components and the IoT entities.

Processing the IoT data can be performed independent of the specific situation and the quality requirements and can take place within the house (Fog Computing), near the house's location (Edge Computing) or in the Cloud (Cloud Computing).

2 Constructive QE – Quality Attributes [175]

Terms

Quality attribute	[ISTQB 17]: (1) A feature or characteristic that affects an item's quality [IEEE 610]. (2) A set of attributes of a software product by which its quality is described and evaluated. A software quality characteristic may be refined into multiple levels of sub-characteristics [ISO 9126]. Quality characteristics are functionality, reliability, usability, efficiency, maintainability and portability [ISO 9126].
Functional security (Safety)	The capability of the software product to achieve acceptable levels of risk of harm to people, business, software, property or the environment in a specified context of use. [ISTQB 17] The part of the overall safety that depends on a system or equipment operating correctly in response to its inputs. The detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequence of the hazardous event. [IEC 61508]
IT-security (Security)	Attributes of software that bear on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data [ISTQB 17] The ability of a software product to protect information and data such that unauthorized persons or systems may not read or change them and only authorized people and systems have access. [ISO/IEC 12207:1995]
Robustness	The degree to which a component or system can function correctly in the presence of invalid inputs or stressful environmental conditions. [ISO/IEC/IEEE 24765:2010(E)]
Resilience	Resilience is the ability of an organization to overcome disruptive risks [ISO Guide 73]. Concerning technical systems, resilience is the ability to not completely fail when disruptions or partial failures occur, but to keep providing principal system services.
Performance	The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate [ISTQB 17].
Interoperability	The capability of the software to interact with one or more specified components or systems [ISTQB 17]
Maintainability	The ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier, or adapted to a changed environment [ISTQB 17]7
Product certificate	Independent expert evidence that a product meets its requirements

2.1 Introduction [30]

Knowledge of quality attributes, their meaning and priority provides the basic foundation for Quality Engineering. The first step is to know which quality attributes need to be addressed. Standards have been established in industry to support this.

The following chapter provides an overview of the quality attributes which have relevance and priority for systems, products and components in the IoT area. Their significance in an IoT context is covered, as well as their influence on IoT architectures, processes and Quality Engineering for IoT.

2.1.1 Overview of the relevant IoT quality attributes (10 Min)

IoT-QE LO 6 (K2) Give an overview of the quality attributes which are relevant for IoT [10]

Quality attributes are characteristics which shape the quality of a system, product or service. General examples are usability, reliability, conformity, aesthetics, durability, security and user experience relating to products or competency of providing services.

Quality Engineering for IoT combines different disciplines such as system engineering, software engineering for applications, enterprise software or embedded software, and quality management. Each of these disciplines can introduce its own specialties, standards and definitions of quality attributes. These are defined partly in general standards such as ISO/IEC 25010 [ISO/IEC 25010], ISO/IEC/IEEE 24765 [ISO/IEC/IEEE 24765:2010(E)] and ISO9126 [ISO 9126] or belong to domain-specific standards. A common language and a shared understanding about quality attributes is a prerequisite for effectiveness and quality in the cooperation between members of an interdisciplinary IoT project.

The standard which is widely used in software development is ISO/IEC 25010 [ISO/IEC 25010]. This standard combines software quality attributes into the following groups: functional suitability, reliability, usability, security, performance efficiency, portability, maintainability and compatibility.

The following quality attributes require particular attention in the IoT context:

Security [ISO/IEC 12207:1995]: The ability of a software product to protect information and data such that unauthorized persons or systems may not read or change them and only authorized people and systems have access.

Safety [ISTQB 17]: The capability of the software product to achieve acceptable levels of risk concerning harm to people, business, software, property or the environment in a specified context of use.

Interoperability ([ISTQB 17]**)**: Enable communications by using compatible data formats and connectivity between devices from different producers.

Robustness ([ISTQB 17]**) and resilience**: Operations in harsh environments and maintaining (partial) functionality when incidents occur.

Maintainability ([ISTQB 17]**)**: The ease with which a software product can be modified to correct defects, to meet new requirements, to make future maintenance easier, or adapted to a changed environment.

Performance ([ISTQB 17]**)**: The degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate.

2.1.2 Operating an IoT System: Challenges occur already at the concept stage (15 Min)

IoT-QE LO 7 (K2) Explain the quality attributes which are also relevant for IoT operations [15]

The operational aspects of an IoT system must already be considered at the design stage (Build to Run). Aspects to consider include functionality for monitoring the operational situation, monitoring system capacity, recovery and maintenance.

The concept for the operational phase must be created with the following aspects of service quality in view:

Availability – the necessary preconditions for the achievement of agreed availability levels must be created. When changes take place, the operational functionality must continue to be fully or at least partially achieved. The changes must not have any side-effects on functionality.

Robustness: The amount by which a system or a component can tolerate exceptional situations. Examples of such situations are heat, cold and vibration but also high volumes of data, limited communications connectivity or a variable energy supply.

Resilience: Ability to handle disruptions and to provide user support in case of disruptive incidents, whether these should relate to systems, service failures or reductions of services. In such cases suitable staff or automatic procedures and services must be available in order to restore operations as soon as possible and/or to support the user.

Performance: Suitability of the available capacity – depending on the operational situation it may be necessary to add capacity which must be installed before bottlenecks occur.

Reliability: Robustness with regard to, for example, weather, vandalism, deliberate attempts at disruption or manipulation, incorrect use, invalid entries etc.

The most important operational parameters are captured and evidence gathered of fulfilling agreed quality attributes. The capture of selected metrics for performance and resource usage must be planned.

2.1.3 Standards (5 Min)

IoT-QE LO 8 (K1) Know the significance of standards and regulatory requirements [5]

Ensuring the correct application of standards is one of the principal tasks of quality engineering:

- Identification of relevant norms, standards and certifications and their further development.
- Implementation in concrete specifications for the product, the system or the services.
- Checking the associated requirements.

Conformity marks and certificates enable conformity to be documented and strengthen trust in the product from customers and users. Legal and regulatory demands (e.g., electrical safety, guidelines from government department) must be applied.

Company-specific standards must be considered and best practices should be known and used. Typical examples for standards are ISO/IEC 2700x Information technology – Security techniques, [IEC61508] Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems or the BSI guidelines on information security.

2.2 Quality attributes with particular relevance for IoT [85]

The IoT quality attributes according to ISO/IEC 25010 [ISO/IEC 2510] are discussed below.

2.2.1 Functional suitability (5 Min)

IoT-QE LO 9 (K1) Know about functional quality attributes [5]

Functional quality attributes according to ISO/IEC 25010 relate to the fulfillment of business requirements placed on a system, product or service.

- **Functional Completeness** – degree to which the set of functions covers all the specified tasks and user objectives.
- **Functional Correctness** – degree to which a product or system provides the correct results with the needed degree of precision

- **Functional Appropriateness** – degree to which the functions facilitate the accomplishment of specified tasks and objectives. Is the implemented functionality too complicated or elaborate?

In particular the functionality of intelligent self-learning systems requires special test procedures because the system's behavior continues to change as it learns.

2.2.2 IoT Security (25 Min)

IoT-QE LO 10 (K2) Explain the security and safety challenges facing IoT systems [25]

The IoT security of an IoT system or product takes into account both the vulnerability of the system itself (**IT security/ security**) and the dangers which may arise from the system (**functional security / Safety.**

Internet connectivity increases dramatically the exposure to dangers caused by manipulative attacks and penetration attempts, as well as unwanted physical influences (destruction, theft, manipulation).

The development of safe hardware and software which comply with market standards (e.g., [ISO 27034]) influences the security of the equipment and the protection of processes information.

The following ISO/IEC 25010 quality attributes are relevant to this subject:

- **Confidentiality** – degree to which a product or system ensures that data are accessible only to those authorized to have access (e.g., by ensuring encrypted storage and communications).
- **Integrity** – degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data (e.g., by using encryption and check sums).
- **Non-repudiation** – degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later (e.g., by protecting evidence from corruption).
- **Accountability** – degree to which the actions of an entity can be traced uniquely to the entity.
- **Authenticity** – degree to which the identity of a subject or resource can be proved to be the one claimed. Associated: **Identifiability,** i.e. the components of a system can always be uniquely identified.

Legal requirements relating to data protection must be strictly applied; especially (in Germany) the State Laws on Data Protection (BDSG). Personal data may only be processed under particular conditions. Ethical aspects must be considered such as the protection of individuals' private autonomy, their privacy or their trust.

Requirements for the assurance of functional security (safety) of IoT systems are given in general form in standard IEC 61508 [IEC 61508] as well as in various branch-specific standards such as ISO 26262 Road Vehicles Functional Safety, ISO 50128 Railway applications, ISO 13849 Safety of Machinery, etc.

2.2.3 Compatibility (10 Min)

IoT-QE LO 11 (K2) Explain the requirements for the interoperability of IoT systems [10]

Compatibility – the ability to use the IoT systems, products or their components created by different producers together or adapt them to each other.

IoT systems, products or components must potentially be able to coexist and work together with components and platforms from different and varying producers at all levels of the IoT architecture. Two sub-attributes of compatibility therefore play a significant role for IoT systems:

- **Interoperability** – Systems and components can exchange and use information with each other. Apart from the connectivity between the components, compatible data formats and protocols are needed, together with a common interpretation of the data.
- **Coexistence** – Systems and components can use the same infrastructure without adversely influencing their respective functionality.

2.2.4 Robustness and Resilience (10 Min)

IoT-QE LO 12 (K1) Explain the quality attributes robustness and resilience for IoT systems [10]

IoT products or components are often exposed to harsh conditions. As Things in the real world they must be protected above all from environmental influences; frequently they must cope with special physical conditions such as heat, cold and vibration but also high volumes of data, limited communications connectivity or a variable energy supply. The implementation of self-administering devices supports product resilience.

Additionally, the attributes robustness and resilience are of course also of major importance for complete systems, including their possible involvement in the Cloud.

2.2.5 Maintenance and Portability (15 Min)

IoT-QE LO 13 (K2) Explain the requirements for the maintainability and portability of IoT systems [15]

In the area of IoT the producers of IoT devices which are designed for a long life (e.g., cars, production machines or high-value household equipment) face the challenge that the security, interoperability and maintenance of their devices must be supported over many years and keep up with changing IoT processes (e.g., closing security gaps, supporting new communications formats).

Typical types of maintenance are corrective, perfective, predictive and preventative. Even though the Internet connectivity enables maintenance to be automated, including the predictive and preventative aspects, many IoT devices are not continuously connected to the Internet and only have limited accessibility.

The ISO/IEC 25010 [ISO/IEC 25010] standard considers the subject of maintenance above all from the perspective of software:

- **Modularity** – degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components
- **Reusability** – degree to which an asset can be used in more than one system, or in building other assets
- **Analyzability** – degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified
- **Modifiability** – degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality
- **Testability** – Effort and effectiveness with which test criteria can be set and the test execution can be performed.

ISO/IEC 25010 [ISO/IEC 25010] divides up the quality attribute **Portability** as follows:

- **Adaptability** – degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments (including individualization, if the adaption is performed by an end-user)
- **Installability** – degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment
- **Replaceability** – degree to which a product can replace another specified software product for the same purpose in the same environment

2.2.6 Performance (10 Min)

IoT-QE LO 14 (K2) Explain the special challenges of performance efficiency quality attributes (time behavior und resource utilization) for IoT systems [10]

Achieving good time behavior and low resource usage represents for many IoT products and components a major challenge, since the devices produced often need to be small and inexpensive and may have no external power source.

Modern transport technologies and protocols (e.g., LTE-M, LoRa, SigFox) are optimized for low data rates and long range with low power consumption, so bandwidth limits and potentially high latency times must be considered in the design.

The performance quality attribute for IoT systems, product and components is defined in ISO/IEC 25010 [ISO/IEC 25010] with three particularly relevant sub-attributes:

- **Time Behavior** – degree to which the response and processing times and throughput rates meet requirements when performing functions
- **Resource utilization** – degree to which requirements are met for the amounts and types of resources used (e.g., power, storage space) when performing functions
- **Capacity** – degree to which the maximum limits of a product or system parameter meet requirements

2.2.7 Ethical aspects regarding IoT (10 Min)

IoT-QE LO 15 (K2) Explain the relevance of ethical aspects for IoT [10]

The increasing interconnection of Things and the growing use of automation, data analysis and self-learning machines has lead to a feeling of loss of control for users and operators of IoT systems. The protection of privacy and other decisions concerned with ethics is partly left to machines. This challenge must be taken into consideration in the construction and validation of IoT systems.

Ethics is concerned with the regulation and evaluation of human actions. When evaluating ethical aspects the context in which the IoT system is used and the sociological context (e.g., morals, culture, laws) of the user and operator must be taken into account. When doing this it must be remembered that ethical decisions can have different outcomes depending on the moral and cultural background in various geographical locations. Important aspects for the evaluation of ethical questions include legality, justice, respect, freedom to decide, environmental protection and sustainability. In this sense ethical aspects can influence all quality attributes of an IoT system. Ethics play in increasingly important role in IoT projects. The Quality Engineer needs to recognize potential ethical implications in order to address these in the construction and assurance of the IoT system. However, the Quality Engineer does not bear responsibility for this; the response to ethical questions must be demanded from quality management and/or the project leadership.

Examples:

1. Protection of privacy: Which sociological standards (or laws) muss be observed in order not to damage someone’s right of self-determination (e.g., due to the possibility to monitor a patient’s medical diagnosis?)

   It would be illegal if medical values were to be transferred not only from the equipment taking the measurements to the application, but also to the patient’s health insurance without the their consent.

2. Freedom to decide: which information may not be withheld from the user of a software application so that they may continue to decide freely?

   With regard to security and sustainability, it is questionable when the user’s navigation system proposes a quicker route but does not permit a proposal for a slower or lower risk route.

2.3 Scenario-based exercise [60]

IoT-QE LO 16 (K3) Evaluate the quality attributes of a system and derive requirement for the IoT system [60]

Role play on the basis of a realistic example scenario of an IoT product:

- The course participants form groups of approx. 3-4 members. Each group shall identify and prioritize the most important quality attributes.
- From these the two most important attributes is selected and for each of them several requirements on the system from the viewpoint of QE4IoT are described. (25 min)
- Finally, the selections are presented in the plenary sitting and discussed. (25 min)

2.3.1 Example for the Exercise on Quality Attributes (60 Min)

(Based on [oneM2M 16], chapter 7.3 *Secure remote patient care and monitoring*)

Applications in the area of digital medicine (E-Health) increasingly allow remote monitoring, diagnosis and treatment of patients and their health parameters. They at least partially remove the need for a visit to the doctor's practice or a local care of the patient by a doctor or care workers. This leads to considerable cost savings and avoids effort and inconvenience. In addition, a complete management of chronic illnesses in enabled, which allows patients to remain independent for longer periods in an environment which is familiar to them.

The measurement of various medical parameters takes place with the support of medical and other sensors in and on the body or in the vicinity of the patient. The information can be remotely read and analyzed with the help of suitable applications. Alarms can be automatically issued by the sensors to those providing help as soon as life-threatening situations are recognized or limits are exceeded. Messages can also be sent to care workers or family members when less serious anomalies are recognized. Alternatively, systems like this can also be used by people who are with the patient to set off particular actions, such as changes to applied doses of medication and obtaining remote support.

Note: in many countries, including Germany, the protection of personal data (and in particular health-related data) is strictly regulated. Any data protection violations are heavily punished.

E-Health-Systems can contain protected data at several levels of sensitivity. Great care is needed to ensure that access to various types of data is only possible for the assigned user groups (e.g., patient, doctor, care worker, family).

Involved groups:

- Patients who use sensors for measuring their medical values.
- Operators of E-Health applications, who supply the sensors, operate the system for monitoring measurements and provide services for the processing of messages to care workers, etc.
- Medical and care staff (health care workers, doctors etc.) and other administrative service providers (e.g., accounts office, insurance) who must have restricted access to selected medical data.
- Technical service providers, such as network providers, software suppliers, etc.

Reasons for accessing data:

- New measurement data are issued by a medical IoT device.
- An analysis of received medical data has been provided (e.g., alarm, messaging) which demands a reaction.
- A request for sensitive medical data is received from an authorized person.
- A new participant (e.g., a new doctor) is authorized for a medical scenario.

3 Constructive QE – IoT Architecture [125]

Terms

Edge Computing	Decentralized data processing based on the partial evaluation of sensor data on the fringe of the network (the „edge“) in preparation for uploading to the Cloud.
Fog Computing	Decentralized data processing based on the partial evaluation of data in a local network in preparation for uploading to the Cloud.
Reference model	An abstract framework for understanding the principal relationships between the entities in an environment, and the development of common standards or specifications for supporting that environment.

3.1 IoT Reference Architectures [65]

Adequate architectures which are domain-optimized are the technical basis for the quality of a system. Knowledge of architectural specifics and the requirements on the architecture are important parts of Quality Engineering in an IoT context.

3.1.1 Reference Architectures (10 Min)

IoT-QE LO 17 (K1) Know about selected IoT reference architectures [10]

Reference architectures are models and references for a class of architectures. They define the architecture of a system from several different points of view. One of the most important of these viewpoints defines the elements in the architecture. In addition, reference architectures describe interactions (data communication, synchronization of activities) independent of the fundamental platform. They offer generic model guidelines and rules for the development of a specific system architecture and provide, according to ISO/IEC CD 30141 [ISO/IEC CD 30141]:

- The description of the characteristics in an IoT system.
- The definition of the domains in an IoT system.
- The description of the IoT systems and its elements.
- The description of the interoperability between the elements of an IoT system.

An overall and extensive standardization of reference architectures for IoT does not exist at the moment. Domain-independent and domain-specific reference architectures are currently under development. The reference architecture AIOTI HLA (High Level Architecture) from the "Alliance for Internet of Things Innovation" (AIOTI) [AIOTI 16] is one of the most prominent (status 2017) domain- independent IoT reference architectures and serves in this syllabus as a guideline for IoT architectural subjects. Apart from that there are many domain-independent and domain-specific standardization approaches.

The prominent reference model for IoT used in the German Industry 4.0 initiative is called RAMI 4.0. This model combines the essential elements of Industry 4.0 into a three-dimensional layered model [RAMI 4.0 15].

There are various commercial and Open Source solutions for IoT IT platforms in the Cloud. Hardware platforms and Tool Kits enable even small projects to start development of IoT solutions.

3.1.2 AIOTI HLA as an IoT Reference Architecture (55 Min)

IoT-QE LO 18 (K2) Explain the elements of an IoT architecture using AIOTI as an example [15]

An important aspect is the static view of the elements in an IoT architecture. The domain model of the AIOTI reference architecture defines the elements of an IoT architecture as follows:

- User: User, human or otherwise
- Thing: physical object
- IoT Service
- Virtual Entity: virtual instance of a physical object
- IoT Device: Interface to the physical possibilities of the physical object

A user interacts with a physical object (Thing) and an IoT service operates as an agent for this interaction. The IoT service is connected with a Virtual Entity which virtually represents the physical object and represents the objects' characteristics in the virtual world. The interaction of the IoT service with the physical object is enabled via an IoT device which also includes the physical capabilities of the Thing.

IoT-QE LO 19 (K2) Explain the layers of IoT architectures using AIOTI as an example [15]

A further important aspect is the dynamic viewpoint which in AIOTI is represented by the AIOTI functional model. This describes functions and interfaces between the elements of an IoT system and consists of three layers:

- **The Application Layer** contains communications and interface functions for the communication between processes.
- **The IoT Layer** contains the specific IoT functionality (e.g., data management) and makes these available via the Application Programming Interfaces (APIs).The IoT layer uses the services of the Network Layer.
- **The Network Layer** groups services at data and control levels. The Network Layer makes transport mechanisms available for user data (near and far communications as well as between entities in the IoT Layer) and control services.

IoT-QE LO 20 (K2) Explain the functions of layers in IoT architectures using AIOTI as an example [15]

The functions within a layer are described in terms of entities:

- **The App-Entity** implements the IoT application logic decentral in devices, gateways or servers. For example, tracking systems for vehicle fleets, remote monitoring of blood sugar etc.
- The **IoT Entity** makes IoT functions and the data they generate available for the App-Entities or other IoT Entities. An IoT Entity uses the lower level Network Layer to send and receive data and for access to the control layer of the network.
- The networks in the Network Layer typically integrate heterogeneous network technologies (e.g., PAN, LAN, WAN, etc.) and network domains connected using the Internet protocol.

Depending on the communications technology used the Network Layer can offer various qualities of service (QoS). In the end the relevant requirements are determined by the Application Layer.

IoT-QE LO 21 (K2) Explain the specific influence of data on IoT architectures [10]

The occasionally high data volumes and the partly limited bandwidth and data storage volumes of the entities within the IoT Layer, combined with an inability to guarantee their continuous availability, make Edge Computing an important architectural approach.

Edge Computing describes the decentral processing of data at the borders (the so-called “edge”) of the network. Data from an IoT system is prepared, aggregated and stored at the edge (e.g., a gateway) and made available directly to applications over the Network Layer or alternatively to a Cloud.

A similar approach is followed by Fog Computing. Fog systems are clusters of IoT systems (rather like small computing centers) which perform a partial evaluation of data in a local network in preparation for uploading to the Cloud. With Edge Computing it’s even more about direct device sensors.

3.2 Representation of IoT Systems with Reference Models [60]

3.2.1 Smart-Home Platform (60 Min)

IoT-QE LO 22 (K3) Construct an IoT system architecture from an IoT reference architecture [60]

The Smart-Home architecture (see chapter 1.4) can be presented with the support of various IoT reference architectures. A further variant of this type of reference model is oneM2M [oneM2M 16]. In this section the items and IT devices involved are illustrated with the help of the elements and terms which oneM2M introduces.

The AIOTI reference architecture can be established on the basis of oneM2M. The AIOTI High Level Architecture (HLA) describes the linkage between oneM2M and the AIOTI HLA Function Model (see [AIOTI 16], Chapter 6.2). oneM2M specifies Application Entities (AE) which make functions available as Common Services Entities (CSE) from the IoT Layer via an API. The lower level Network Services can be used as a horizontal communication platform to establish an additional path of architectural communication. The IoT and Application layers use standard protocols CoAP, MQTT, Websockets and HTTP. In addition, the Network Layer is connected via 3GPP and 3GPP2. The horizontal communication between AEs is not standardized.

oneM2M is a web-based architecture which has been specially designed to create a horizontal service layer which can also be applied to various domains, such as transport and healthcare.

Using this Service Layer Middleware, vertically structured data can also be exchanged horizontally. This communication is implemented by the protocols which include HTTP (Hypertext Transfer Protocol), MQTT (Message Queue Telemetry Transport) and CoAP (Constrained Application Protocol).

Apart from this, meta information such as data structures and APIs are defined so that third-party applications can connect with a oneM2M-System. This enables the exchange of IoT data between different systems and applications.

The vertical model is divided into four layers: Application, Service/Platform (P), Network (N) and Device (D). Any IoT system can define its own special P/N/D.

4 Constructive QE – Processes and Methods [250]

Terms

DevOps	DevOps (a combination of Development and Operations) is a way of working for developing and administering applications. Common impulses, processes and tools enable a more effective and efficient cooperation between the areas of Dev, Ops and quality assurance.
Constructive QE	All-embracing implementation of preventative measures to avoid that something is implemented with defects, in a unsuitable way or with insufficient care.
Agile Software development	A group of software development methodologies based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
Usability	Extent to which a software product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. [ISO 9241]
Updateability	Possibility to extend and/or improve a version of a software product.

4.1 Constructive Quality Engineering [35]

4.1.1 Introduction (15 Min)

IoT-QE LO 23 (K2) Explain constructive QE [15]

IoT systems are characterized by the focus on data and their communication (e.g., between Microcomputer-Units, Embedded Systems, Gateways and Cloud Servers) with a particular focus on the exchange and processing of shared data. Data security has a particular importance with regard to both individual components and the desired interaction of various layers and end-to-end behavior. In this context some quality requirements conflict with each other when constructing IoT systems. This requires awareness for the subject of Quality Engineering.

The motto “prevention is better than cure” is applied. Constructive Quality Engineering involves taking measures in order to assure that the desired outcome is achieved in a suitable way and with the required care.

Constructive quality work has the goal of preventing defects right from the start. Orientation on best practices and standards in the development and production processes and in the set-up of the work environment enables experience to be leveraged. The planning of quality checks during the development and production stages as well as the planning of the subsequent operations phase regarding the achievement of required service levels also belong to the area of constructive quality work. A risk-based approach is recommended when deriving the quality levels of a component or part thereof.

The measures applied in quality work are applied overall to different levels:

- Organization,
- Processes
- Project(s) and
- Product(s).

Constructive QE for IoT systems builds on the fundamental understanding of the quality attributes and architecture of IoT systems. It supports analytical quality assurance with the optimal planning of quality measures.

4.1.2 Processes and Best Practices for IoT Development (5 Min)

IoT-QE LO 24 (K1) Know the best practices in IoT [5]

The goal of having organizations which learn and continuously improve is an integral part of IoT business models. Decision-making is derived from data and can also lead to changes in business processes and the organization.

The setting down and adherence to processes is beneficial to the organization by:

- creating transparency (defect avoidance from understanding),
- laying down responsibilities (identification of participants and their tasks),
- interdivisional communication and coordination,
- process-oriented thinking and acting, and
- building up a basis for further optimization and automation.

Best practices offer a similar but not comparable benefit because they are not binding and not always known. In comparison to and in addition to processes, a best practice is:

- a non-binding recommendation on how to proceed in a particular case,
- more flexible than a standard, and
- in the case of changes to requirements or conditions, easier to replace with a more promising approach.

Due to the special requirements which apply to IoT products and solutions (e.g., time-to-market, number of variations, complexity, data-driven added value), reference should be made to the significance of interdisciplinary teams from the areas of development, operations and quality assurance (DevOps) [Humble 10]. This team carries the overall responsibility for each development.

4.1.3 DevOps for IoT (15 Min)

IoT-QE LO 25 (K2) Explain DevOps for IoT [15]

DevOps describes a way of working taken from the areas of software development and system administration. DevOps is a blend of the words development and IT operations. DevOps permits more effective and efficient cooperation between the areas of development, operations and quality assurance by bringing together common interests, processes and tools. With DevOps the quality of a software-based system, the speed of its development and delivery, and the reliability of its operations are improved by cooperation between the participating teams.

In practice this means, for example:

- developers are more involved in the installation of virtual machines and with aspects of IT security, or with the planning and roll-out of deliveries.
- administrators are more involved with automation in combination with “Infrastructure as Code”. This means that the operations teams can perform the administration and delivery of a particular IT infrastructure, including version control and automated tests, automatically via code instead of manually.
- developers and IT operations staff become accustomed to agreeing and deploying interdivisional Key Performance Indicators (KPIs) as common targets.

The groups involved in DevOps follow contractual goals which may be partially in conflict with each other:

- developers want to implement changes quickly,
- testers want to limit the risks of discrepancies,
- administrators want to guarantee stable operations.

By establishing a suitable DevOps culture based on consensus building the organization is put in a position where it can react quickly and efficiently to changes in the general conditions or business goals. In the context of IoT this means, amongst others,

- vertically, to consider asset management (development and administration of the Things / hardware) and,
- horizontally, to consider the upstream and downstream business processes (Service Definition, Operations, Customer Service, Market Analysis, Business Intelligence)

4.2 Selected Aspects of IoT [90]

4.2.1 IoT Exercise for Data Orientation in IoT (60 Min)

IoT-QE LO 26 (K3) Explain the consequences of using the data-driven IoT business model [60]

The principal value added by IoT systems comes from the data they create themselves or supplement from third parties. Examples for this are:

- real-time information and control (Dashboard, Tracing),
- evaluation of historical data with application for predictive models (Predictive Maintenance, Machine Learning) or
- highly efficient administration of resources (On Demand, Share-Economy) on the basis of usage data.

The value added arises from the creation of options on the basis of extracted information from variable data (see Diagram 2). The services, processes and software for the collection and processing of the data and the processing of results should be subject to continuous improvement.

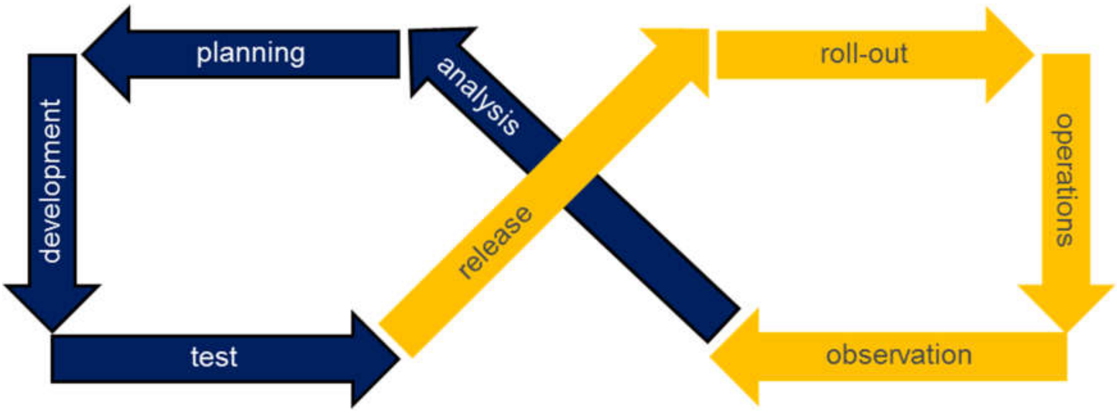

Diagram 2: Data-driven DevOps process

The course participants make groups of 3 to 4 members. Each of these groups prioritizes the most important quality attributes for their tasks and defines a target range. The achieved status is presented to the plenum and briefly discussed (30 min).

In a second round, the basic requirements are changed. Each group prioritized once more the most important quality attributes for their tasks and defines a target range. The achieved status is presented to the plenum and briefly discussed (30 min).

4.2.2 Variants of IoT Systems (15 Min)

IoT-QE LO 27 (K2) Explain the meaning of product and system variants for IoT [15]

In system development different variations of especially functional product characteristics are identified as variants. Variants frequently (but not always) have a common basis from which they develop with specific characteristics for the particular variant e.g., customer-specific modifications. Variants can develop independent of time-based considerations.

In comparison, versions represent different stages in the development of a variant. Versions build on a defined basis and develop over a period of time. Version control captures the changes made and archives them with time-stamp and user identification to enable traceability and permit previous versions to be re-created.

An IoT System consists of a large number of hardware and software components which are subject to different lifecycles and development speeds (see also Chapter 6). The use of variants and versions adds substantially to this complexity. Due to the continuous development of IoT systems, the following situations (among others) can occur:

- adaptions for relevant groups of customers
- piloting of newly developed components
- replacement of defective or out of date “things”

These continuous developments can result in the parallel operation of different processing sequences within the overall IoT system. This can lead to higher development and maintenance costs and a higher risk for the overall end-to-end process sequence. As an example for this, sensor data with different degrees of precision need different types of processing and can impact decisions with machine learning potential. This can influence the overall IoT system. For these reasons it is highly recommended to oversee the component interfaces, including their precise qualification. Apart from that, a careful management of configurations, variants and versions must be applied, together with an awareness for dynamic operations.

4.2.3 Operation of IoT Systems (15 Min)

IoT-QE LO 28 (K2) Explain the importance of QE for the operational phase of IoT systems [15]

The value of the data obtained in the operational phase demands that requirements are placed on Quality Engineering concerning the roll-out, maintenance, update and decommissioning of an IoT system. These tasks will be performed on a complete or partial system and are typically performed on the „live“object. To avoid system failures, a risk-based planning is recommended which analyzes the modified system for undesired consequences depending on the risk criticality. Approaches for this are, for example:

- the piloting of changes on a partial system, or
- operations-specific regression tests.

Apart from the obligatory need for continuity of operations, changes require that a test be carried out to check for any negative impact on data processing, any undesired effect on users and the achievement of the change’s objectives. In particular the scalability, load and performance and security must be ensured. Once an acceptable test result and remaining risk has been achieved, a complete installation and operational roll-out of the change can be started. This typically results in conflicts between the demanded quality requirements and the frequency with which new versions or variants can be made productive. In particular the business model for an IoT system or partial system can demand that frequent system updates are performed, which can clash with the required quality attributes. Placing quality characteristics at the back of the queue can be a valid strategy if the reduced level of quality can be compensated for with high priority and the resulting damage can be contained within an acceptable corridor.

4.3 Quality Attributes of an IoT System and Constructive QE [105]

4.3.1 IT Security and Safety (60 Min)

IoT-QE LO 29 (K3) Perform an impact analysis of the IT security and safety quality attributes on constructive QE [60]

Apart from the classic area of conflict between the quality attributes of IT security and safety, (see also Chapter 2) differences also exist in terms of the areas to be protected regarding the IT security requirements placed on IoT Things compared to the relatively well-developed IT security for classic IT architectures.

Principal objectives of protection:

- Confidentiality: only authorized users may read and change data.
- Integrity: no changes go unnoticed and each change can be traced.
- Availability: Access is possible only within an agreed time interval.

Additional objectives of protection:

- Authenticity: Genuineness of an object.
- Non-repudiation: no forbidden challenges to performed activities
- Accountability: explicit assignment of responsibilities
- Privacy: protection of personal privacy and data.

Classically, safety requires stable, mostly certified software versions, whereas IT security also demands regular and short-term updates. Due to the data-orientation of IoT not only the primary objectives of protection are in focus, but increasingly also the additional objectives of protection. Whereas the focus rests on the confidentiality and integrity of well secured data that is mostly resident in data centers, the attention is directed at authenticity and accountability for Things that are spread across the world. In this sense the distribution of safety across IT and Things automatically sets requirements for the networking and interaction regarding availability, confidentiality and non-repudiation.

In addition, the protection of privacy represents an extremely important quality requirement which (in Germany) is regulated by European law and the German data protection laws. The use of personal data must be contractually documented and approved by the customer. A contractually invalid connection between data with personal data must be prevented. In this sense it must be appreciated that even Thingscan save personal data which then must be protected accordingly.

Of significance for IT security of architectures is knowledge of and the risk evaluation of typical IoT attack vectors on the various levels of the architecture (see table below). This information is the basis for implementing appropriate protection mechanisms.

Level of Architecture	**Attack Vector**
IoT layer - Devices and connectivity	- physical device interfaces - web interfaces for the devices - network interfaces of the devices - device storage and storage extensions (e.g., SD cards) - device firmware

	- Physical manipulation or theft of device
Network layer (Computation, Aggregation and Storage-services)	- Cloud Web Interfaces - Backend APIs - Update Mechanisms (over the air updates) - Other communication between IoT Layer and Network Layer
Application layer - (Analytics, Visualization und Control)	- Mobile applications - Web applications

In analytical Quality Engineering these attack vectors are assigned a testing focus according to priorities from the risk analysis. (Chapter 5.4.2)

The course participants make groups of 3 to 4 members. Each of these groups formulate concrete requirements. The achieved status is presented to the plenum and briefly discussed (15 min).

4.3.2 The Trade-off between Usability, Maintainability and IT Security (15 Min)

IoT-QE LO 30 (K2) Explain the trade-off between usability, maintainability and IT security [15]

Things and services should be as simply and safely as possible installed, adapted, maintained and de-installed (see also Chapter 6 Lifecycle):

- "Simply" means where possible automatically and without work-intensive manual intervention from users or service operators.
- "Safely" means without violating data protection or other values to be protected.

Users want the integration of already used services and Things in their installed environment to be as seamless as possible, including the intuitive possibility to adapt these to suit their individual wishes.

Compared to this, a service operator must guarantee a contractually agreed and secure operation which in emergency situations must be capable of reverting to manual remote maintenance without the involvement of a local service technician. This is often in conflict with simple usability.

4.3.3 The Trade-off between Resilience, Robustness and Performance (15 Min)

IoT-QE LO 31 (K2) Explain the trade-off between resilience, robustness and performance [15]

The fundamental structure of IoT systems as distributed systems is advantageous for resilience, whereas robustness and performance can be challenged by the many interfaces. This means, for example, that the failure of a central gateway might make an IoT system unavailable. In this case a redundant implementation can improve both robustness and performance, although more system resources would be needed.

4.3.4 The Trade-off between Connectivity, Interoperability and IT security (15 Min)

IoT-QE LO 32 (K2) Explain the trade-off between connectivity, interoperability and IT security [15]

IoT systems and components are interconnected, which requires interoperable interfaces in order to guarantee connectivity. On the other hand, interoperable interfaces offer similar (when not identical) weaknesses regarding their IT security. In addition, an infected IoT device can relatively easily endanger further devices and components over the IoT network.

4.4 Approaches for Continuous Development [20]

4.4.1 Advantages of Agile Procedures (10 Min)

IoT-QE LO 33 (K1) Know the advantages of an agile approach [10]

Traditional development models such as the V-Model, which use upstream specifications and perform downstream verification and validation with the customer, are under pressure, especially for IoT systems. This is due to:

- high effort for the creation of specifications of IoT products, which can be very complex,
- additional technical requirements, which occur during development ,
- new requirements which arise from the customer's need for fast adaptions.

This is where agile development approaches contribute towards efficiency increases with their principles. They are based on continuous improvement processes which are performed in iterations such as in the Plan-Do-Check-Act approach:

- Plan – define objects and processes for achieving goals
- Do – implement the plan and collect data for decisions
- Check – identify any divergences from the plan and goals, and any potential for optimization
- Act (Adjust) – take on the successful aspects and improve

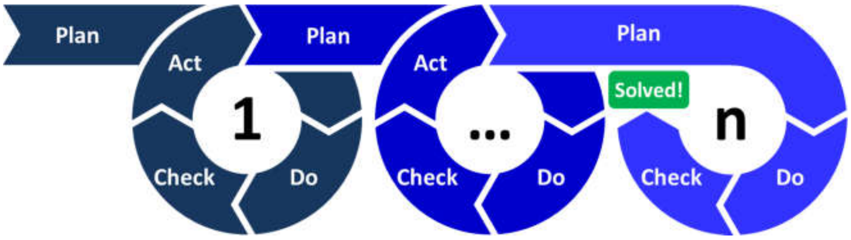

Diagram 3: Agile Development with the Plan-Do-Check-Act Approach

4.4.2 Advantages of Automated Approaches (10 Min)

IoT-QE LO 34 (K1) Know the advantages of automated approaches [10]

The software, services and processes of an IoT system for the collection and processing of data, including the processing of results, are often subject to continuous development.

In addition, the time factor is often highly relevant for IoT systems such as with, for example, the roll-out of security updates. A fast reaction to market developments is also often needed.

Manual intervention in these situations increases risk and leads unavoidably to bottle-necks due to the length of the process sequences involved. Increasing computing capability to give the maximum level of automation is in comparison relatively cost effective, although the effort for the development and maintenance of the automation must also be taken into account. If parallelization can be applied to the automation, reaction times can relatively easily be reduced and the robustness of the automation increased.

In addition, various infrastructures can be used, for example for development, sales, operations, monitoring and maintenance, or the management of the product and customers. These areas should also be automated as much as possible. IoT should in equal measures be subject to these automation approaches and the constructive and analytical approaches to Quality Engineering.

5 Analytical QE (including Test) [260]

Terms

Fuzz Testing	A software testing technique which sends automatically generated invalid inputs to the target system in order to identify broken data structures, or finds invalid data inputs which can result in a worsening of services. ETSI TR 101 583 "A software testing technique used to discover security vulnerabilities by inputting massive amounts of random data, called fuzz, to the component or system." [ISTQB 17]
Conformity	The capability of the software product to adhere to standards, conventions or regulations in laws and similar prescriptions [ISO 9126].
Things under Test	Extension of the term „System under Test" [ISTQB 17] for an SUT, which integrates hyper-physical components.
Data quality	Evaluation of data regarding their correctness, relevance and reliability, as well as their consistency and availability on various systems.

IoT-QE LO 35 (K1) Know the need for monitoring during IoT system operations
[spread across chapter 5]

IoT-QE LO 36 (K2) Explain the challenges of distributed tests for IoT systems
[spread across chapter 5]

5.1 Introduction [10]

IoT-QE LO 37 (K2) Explain the special challenges of testing IoT solutions such as their openness, degree of distribution, changeability, scalability and variability [10]

IoT solutions are in general characterized by their openness, distributed nature, dynamism, scalability and a long operational runtime. For these reasons new approaches to analytical quality assurance are necessary.

For example, the quality assurance that accompanies development must be extended in the period it covers. As a consequence of the DevOps approach, testing, runtime monitoring and certification are interconnected and must be reconsidered. Specifically for IoT it is also necessary to define a further „operations" test level following the usual system and acceptance tests so that possible later changes can be considered, such as interface extensions, exchange of system elements or newly diagnosed deficiencies.

A particular difficulty arises from the question of liability in the event of damages following a security incident or inappropriate use. This is especially the case when for certification purposes at a particular point of time the self-declaration of the manufacturer or operator needs to be relied on.

5.1.1 Test Objectives and Prioritization (30 Min)

IoT-QE LO 38 (K3) Define and prioritize test objectives for IoT [30]

In General the test objectives in the context of IoT include those based on the quality criteria described in ISO/IEC 25010 [ISO/IEC 25010] (see Chapter 3.1), whereby particular attention is given to interoperability, IT security and performance. The connection between functional security and information security must also be considered.

The priority of test objectives relates to the priority of the quality attribute to be tested. Test objectives and their priorities must be continuously evaluated over the lifecycle of the IoT system and where necessary adjusted or extended.

Apart from the functional quality attributes, the following quality attributes have a particularly high significance for the prioritization of test objectives for IoT systems in comparison with "classic" systems:

Reason	**Quality Attribute**
Specific (distributed) architecture	Interoperability Performance and capability Adaptability Robustness and resilience
Interrelated lifecycles and interdisciplinary nature of IoT	Compatibility Maintenance Portability
Interrelated and wide ranging business processes which can be represented in IoT systems	Functional security (safety) IT-Security Privacy Ethical aspects

It is helpful to structure the IoT test requirements and objectives into groups relating to process, system/components and communications protocol.

5.1.2 Specific Test Levels for IoT (15 Min)

IoT-QE LO 39 (K2) Explain the specific test levels for IoT [15]

Test Level	**Example**	**Remarks**
Acceptance test / system test or certification according to general test and integration requirements	Information security Conformity with supporting protocols Conformity with standardized sequences	Dependencies between usage profiles must be considered (e.g., private vs. industrial application, military application). Conformity in this sense relates to standards and standards-like documents.
Integration test for the embedding of the test object into its (test) environment	Compatibility Interoperability	Can strongly depend on the specific usage scenarios of the test objects. System environment can include high levels of or possibly not fully foreseen behavior (e.g., future new services) Environment can also be created by simulation.

Operational/ Diagnosis test in the production environment and possibly also during the productive phase (e.g., passive tests for monitoring of behavior in operation)	Presence of required services (e.g., production tests by the producer), Test scenarios for sustaining operations	Since the system environment cannot completely recreate the anticipated behavior which may change, tests and analysis must also be required in operation. Triggers are, for example, new deficiencies or updates that cannot be executed in a lab setting. No „continuous“ tests without user permission. It is possible that associated additional security risks (IT security and safety) must be considered.

5.1.3 Risk Analysis (30 Min)

IoT-QE LO 40 (K3) Prioritize test objectives according to their risk [30]

Safety-critical IoT systems demand a special consideration of criticality by performing a risk analysis for the entire IoT system. The results of the risk analysis are used to derive the priorities and test objectives. The ETSI EG 203 251 (Risk-based Security Assessment and Testing Methodologies, Chapter. 7.2 ff) serves as a reference for the integration of risk analysis in a test development process.

5.1.4 Test Activities in the Lifecycle (15 Min)

IoT-QE LO 41 (K2) Explain the test approach [15]

Efficient and effective testability is above all for IoT systems an important pre-condition for a quality product. The test analysis and test planning must therefore be carried out for each phase of the lifecycle, including the maintenance phase. In the maintenance phases it is necessary to perform both traditional monitoring and predictive maintenance aspects. The early definition of the test system makes the task of creating test interfaces in the system easier. All testing activities are therefore checked in each phase of the lifecycle and adapted to the needs of the particular phase. Behavioral tests in the productive phase (i.e. in operation) are becoming increasingly important and are performed by the operator, for example, when updates occur at maintenance intervals.

5.2 Testability und Test Automation [25]

5.2.1 Specialties of IoT Testing (15 Min)

IoT-QE LO 42 (K2) Name the special aspects of testing IoT systems and give examples of IoT tests at different test levels [15]

The specialties of testing IoT systems as an extension to the „classic“ software and protocol testing are listed in the following table [Schieferdecker 16].

Layer	**Specialty**	**Test variations in addition to „classic“ software and protocol testing**
IoT Layer - Devices and connectivity	High level of significance for security, conformity /interoperability and data quality	Real-time testing Embedded systems testing GUI testing (for management software) Security testing

Network Layer - platform (computation, aggregation and storage services)	High level of significance for security, conformity /interoperability and availability	Performance and scalability testing Service testing GUI and Usability Testing (for management software) Security testing
Application layer- (analytics, visualization and control)	High level of significance for security and usability	GUI, Usability and (mobile) app testing Performance and scalability testing Security testing Crowd testing

Table 1: Specialties of IoT Testing

5.2.2 Test automation (10 Min)

IoT-QE LO 43 (K2) Explain the need for automating IoT tests [10]

A high degree of test automation must be attained in all test phases in order to guarantee an effective test. The reasons for this are:

- Assuring quality in the lifecycle is connected with a high degree of regression tests.
- The time to market factor has a major and persistent significance.
- The complexity and dynamism of the system context is high for IoT products.
- Manual procedures have a higher risk of error compared to automated procedures.

5.3 Test Process and Test Architecture [60]

5.3.1 The Fundamental Test Process (10 Min)

IoT-QE LO 44 (K2) Explain the fundamental test process in the context of IoT [10]

The fundamental test process [ISTQB 11] is extended in the context of IoT by runtime phases with monitoring and Watch Dogs. This means the system grows successively together with the test system:

- Planning and Control: a part of the lifecycle management
- Analysis and Design: extended risk analysis
- Implementation and Execution: Automation and runtime
- Evaluation and report: continuous
- Completion: first with the termination of an IoT solution

5.3.2 Test Automation Architectures (20 Min)

IoT-QE LO 45 (K2) Explain the generic test architecture and the interaction and application of tools [20].

In order to provide testability and test automation for IoT it is necessary to formulate a complete IoT Test Automation Architecture (IoT TAA) which is generated from the IoT Reference Architecture (IoT RA). [ISTQB 16]

The generic Test Automation Architecture is made up of the individual systems shown in Diagram 4. [ISTQB 16]. The IoT test system can always be traced back to the generic ISTQB Test Automation Architecture, independent of the actual hardware or virtualized components used.

The planning and design of tests is based on a wide variety of IoT test architectures [Jäkel 17], which derive in particular from the generic Test Automation Architecture (e.g., due to several different communications protocols). Different test architectures can be used for different test requirements, test objectives and test phases. These need to be integrated into the overarching test architecture of the test system.

As a result of the requirements, IoT test architectures used for the development of IoT systems must be to a large degree automated and make use of tools.

Monitoring and test access points need to be planned to enable testing in the productive phase which ensure running operations are not impacted. Especially these test access points must be protected.

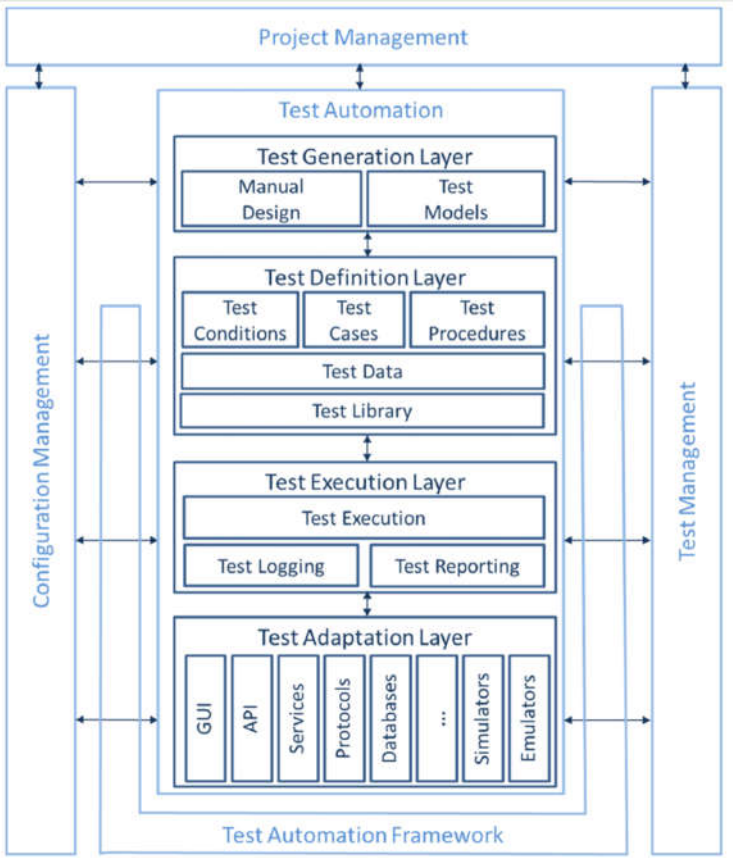

Diagram 4: Generic Test Automation Architecture [ISTQB 16]

5.3.3 IoT Test Architectures (30 Min)

IoT-QE LO 46 (K2) Explain IoT test architectures and typical IoT test objects [15]

Since IoT systems are by nature distributed, the test architectures and appropriate process strategies (amongst them the improvement of efficiency thru virtualization of the entire system) are also distributed. The following test architectures are typical for IoT test systems (see [Jäkel 17] for examples):

- Device-based IoT test architecture (e.g., for testing Retroboxes or Gateways) according to the AIOTI IoT Layer.
- Service-based IoT test architecture (e.g., for the data-oriented testing of dashboards in the Cloud) according to the AIOTI Application Layer.
- Infrastructure-based IoT test architecture (e.g., for the testing of oneM2M functional elements) according to the AIOTI Network Layer.

The SUT and the test system can swap roles for various requirements. This means it can make sense to use a component in one case as an SUT and in another case as a test system which stimulates the SUT.

It is therefore necessary for the test environment to be completely integrated into the process and tool environment (typically over the entire lifecycle and in a DevOps process environment).

IoT-QE LO 47 (K2) Explain the principal aspects of IoT test automation architectures [15]

Principal aspects of the Test Automation Architecture [ISTQB 16] are:

- Understanding the technology of the System Under Test (SUT) as well as it's integration with the Test Automation System (TAS). Specifically for IoT:
  - IoT test interfaces are typically at protocol and service levels.
  - A careful analysis of the test interfaces is needed to ensure an implementation which is future safe.
  - Typical interaction between SUT and TAS are event driven and peer-to-peer.
  - Boundaries for the SUT are decisive for an efficient and effective TAS. These can vary for different test requirements.
- Understanding the test environment. Specifically for IoT:
  - For IoT the simulation of test environments has much higher significance than with a „classic" test.
  - Possible variability in the system boundaries between SUT and TAS must be taken into account in the test architecture.
  - The maintainability of the test environment plays an important role.
  - The real operational environment of the IoT product must be considered.
  - The integration of the test environment with DevOps tools must be considered.
- Time, effort and complexity of the implementation (planning and controlling).
- Ease with which the implementation can be used. (Design aligned to the user profile of the tester as user). Specifically for IoT:
  - Since IoT development projects are highly interdisciplinary, the usability requirements on the must take into account all participating roles.

5.4 Test Techniques [75]

5.4.1 Important IoT Test Techniques (15 Min)

IoT-QE LO 48 (K2) Explain the usefulness and limitations of traditional testing techniques when applied to IoT systems [15]

Testing techniques, including those for test automation and tools, are not fundamentally new; they are a special selection of established approaches which consider the IoT-specific characteristics of the SUT and specifics from the requirements analysis. Techniques for interoperability, security and performance play a very significant role for practical applications. This involves a high degree of automation which is complemented by appropriate manual tests, especially exploratory tests (e.g., for mobile devices in various environments).

Model-driven analysis strategies and model-based testing make optimal analytical approaches available (also for the definition of system boundaries for the SUT) and are an important best practice for tests of IoT systems. Online MBT [ISTQB 17] is an answer to the need for increasing development of the test design in the dynamic IoT situation.

Analytical QE is accompanied by a chain of techniques and tools. Many tools are available, sometimes also as Open Source tools. In addition, standardized test descriptions exist for specific domains and their protocols (e.g., ETSI in the telecommunication, automotive and Autosar domains) typically using descriptions such as TTCN-3.

During the course of executing operational and runtime tests it can become necessary later on to refine the relevant test expectations.

5.4.2 Security Test (15 Min)

IoT-QE LO 49 (K2) Explain the particular challenges for testing the security aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture [15]

Security tests have a major significance for IoT. Aspects of IT security have often not been considered in the required depth, partly because of insufficient economic motivation and insufficient expertise in interdisciplinary projects. In addition, different components of IoT solutions are frequently developed by separate teams, which hinders an overarching consideration of security.

On the other hand IoT systems are excellently suited to the construction of Botnets. A Botnet is a group of automated programs designed to impact system security. They run on interconnected computers whose network connections, local resources and data are available to them without the consent of the owner. A wide range of attacks, such as Denial of Service can be executed using Botnets. Security tests in the area of IoT must address all layers of architecture in sufficient depth. This requires an overall approach to security tests. IoT security is not just about device security! The assurance of an individual component is not sufficient for the security of the overall system.

The following table shows the most important areas of focus for testing the individual layers of architecture (see Chapter 4.3.1).

Architectural layer	**Attack Vector**	**Technique**
IoT-Layer - Devices and connectivity	- physical device interfaces - web interfaces for the devices - network interfaces of the devices - device storage and storage extensions (e.g., SD cards) - device firmware - Physical manipulation or theft of device	- Data flow analysis / Proxy / Man in the Middle-attacks - Test of web weak points - Search for sensitive data (passwords, keys, …) and manipulation of data - Analysis of firmware - Search for protocol weak points or incorrect configurations e.g., unencrypted communications - Side-channel attack
Network Layer - platform (computation, aggregation and storage services)	- Cloud Web Interfaces - Backend APIs - Update mechanisms (over the air updates) - Other communication between IoT Layer and Network Layer	- Test for web weak points - Data flow analysis / Proxy / Man in the Middle-attacks - Spoofing of end devices - Search for logical weaknesses in overall concept - Making use of well-known

		protocol weaknesses
Application layer- (analytics, visualization and control)	- Mobile applications - Web applications	- Test of web weaknesses - Test of sensitive data in mobile devices

Table 2: Focus for Security Tests

5.4.3 Interoperability Test (15 Min)

IoT-QE LO 50 (K2) Explain the particular challenges of testing the interoperability aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture [15]

Interoperability tests are a type of functional test. They evaluate the ability of the software product to interact with one or more specified components or systems. With this type of test execution data are exchanged between two selected systems (client / server). This data contains both correct inputs according to the protocol and deliberately defective data which tests the stability (reaction) of the system. Note: Normally the interoperability tests are executed after successful completion of conformity tests (e.g., with TTCN-3 technology).

Important techniques for the interoperability test are:

Test Object	**Technique**	**What is tested?**
Technical interoperability	Basic tests of connectivity and communications protocols	Coupling of hardware / software components in order to ensure basic communication.
Syntactic interoperability	Targeted checks of messages and the syntax of abstract data formats. Use of encoders and decoders	Correct use of syntax for e.g., HTML, XML or ASN.1 data structures
Semantic interoperability	Execution of sample scenarios and user scenarios, possibly with support from standardized Use Case catalogs	Checks on whether the implementation of the interconnected components/systems follows a common interpretation.
	Standardized catalog of test objectives in tabular form which provides the inputs for configurations, sequences of triggers and observations about the involved components or systems	Standardized test objectives. Plugtests™

Table 3: Important techniques for the interoperability test

Plugtests™ are collections with which the producers of electronic equipment or software can test the interoperability of their products in conjunction with products from other producers. They are not published, have a duration of between three and five days, and are prepared and supported by a neutral institution (e.g., ETSI). A standardized catalog of test objectives in tabular form is used which provides the inputs for configurations, sequences of triggers and observations about the involved components or systems.

5.4.4 Performance Test (15 Min)

IoT-QE LO 51 (K2) Explain the particular challenges for testing the performance aspects of IoT solutions and the application of appropriate testing techniques for different layers of the IoT architecture [15]

IoT Systems are distributed systems which can process huge amounts of data at the application level. A suitable architecture must be selected (e.g., with the use of elements from Edge Computing) to enable the efficient processing of these heterogeneous data streams. The scalable performance of each IoT component is an aspect of the quality assurance which is tested in development and monitored in production. As a result of the typically high volume of transactions required for a performance test, automation is of decisive importance.

Example of techniques and tools for performance tests [ISTQB 16] are shown below:

Technique	Tool example	Explanations
Dynamic Analysis	Dynamic analysis tools uncover defects which can only be revealed when the program is run (e.g., time dependent defects and memory bottle-necks)	These are typically used for the component and integration tests, as well as for tests of middleware.
Performance test, Load test, Stress test	Performance test tools monitor and protocol how a system responds under various simulated usage conditions with regard to the number of parallel users, ramp-up behavior as well as the frequency and relative proportions of transactions. The load is created by simulating virtual users which execute a selected group of transactions. These are distributed over various test machines which are generally known as load generators.	Software test with expected and extreme loads submitted to a running system. The behavior of the system is then observed and analyzed.
Monitoring	Test monitors continuously analyze, verify and record the usage of specific system resources and issue warnings of possible problems in providing services.	E.g., for simulating event data

5.4.5 Product Certification (15 Min)

IoT-QE LO 52 (K2) Explain the challenges of checking for conformity and certification [15]

As a result of the many and varied types of IoT devices and IoT services, certification has an extremely high level of importance. Certification involves the checking of products, processes and services to ensure conformance to the requirements of standards and additional normative documents [DIN EN ISO/IEC 17065:2013-01]. The results from inspections and checks performed in a test lab are evaluated by a certification authority and where appropriate a certificate or at least a recognized seal of approval is issued. Certifications are based on guiding principles, directives, norms and standards. At present however there is no “IoT Standard”, but instead a collection of high-level norms which are still too immature.

The main focus for IoT is the evaluation of IT security:

- Functional security requirements
- Stability

- Conformity and vulnerability to errors of individual communication protocols typically used in IoT.

Their principles do not differ from the IT security principles from the IT area. The following generic standards can be applied [Wardaschka 17]:

- Protocol-specific IoT norms and standards
- Standards with functional requirements
- Standards with non-functional requirements (performance, availability, reliability, documentation, work processes)
- IoT standards which are still to be developed

6 Lifecycle [90]

Terms

IIoT	Industrial Internet of Things relates to the use of IoT in the manufacturing industry (Industrial Internet or Industry 4.0)
Industry 4.0	Industry 4.0 is a futuristic project of the German Government and stands for „the fourth industrial revolution“. Principal characteristics of this fourth industrial revolution are the individualization and hybridization of products and the integration of interdisciplinary stakeholders and business processes.
aggregated	Data and information from various sources are summarized.

As a final consideration for this chapter, the relationship between the activities in the overall lifecycle is reconsidered. In the context of IoT a wide variety of different lifecycles from IoT devices, software such as application services, and infrastructure such as networks come together.

A generally valid IoT lifecycle is currently not variable in the referenced standards (see Chapter 2). This syllabus is therefore based on the simplified lifecycle representation of ISO/IEC CD 30141 which lists the phases Initiate, Build, Develop, Operate, Update and Decommission (see also Chapter 3).

6.1 The Phases of the IoT Lifecycle [30]

6.1.1 Relevant Standards in an IoT Context with their Lifecycle Definitions (15 Min)

IoT-QE LO 53 (K1) Know the relevant standards and lifecycles in the context of IoT [15]

The meaning of the lifecycle term is understood differently depending on the context. The following table provides and overview of the relevant lifecycle definitions in the area of IoT.

	Phases	*Focus*
Software lifecycle (ISTQB)	(normal phases): conception, requirements, design, implementation, test, installation, operation and maintenance, and occasionally decommissioning	IT / Software product
RAMI 4.0 lifecycle	Lifecycle phases of a type are defined as: development and use/maintenance. Lifecycle phases of an instance are defined as: production and use/maintenance.	Manufacturing / Industry 4.0, emphasizes the interconnection of the value chains.
AIOTI HLA Data-lifecycle	Obtain/collect, Create/derive, Use, Store, Share/disclose, Archive, Destroy/Delete	The data lifecycle is considered to be on a par with lifecycles of other dimensions

Services lifecycle	Phases are dependent on the customer's information basis, from the service content and the service level.	Consists, for example, of logistical planning, aspects of project management, reports and identification for software / operational installations/ application installation, replication, data storage, archiving, data migration, tests and customer interactions
IT- lifecycle	Typical phases are: Concept / Implementation / Test-Acceptance / Document-Handbook / Training / Go Live / Support / Change Request /Decommission / Disposal	IT Systems and software in focus
ISO 15288	Supplier agreements, technology, company (management), project	System level
IEC 61508	Functional security – general requirements, concept phase, application domain definition, analysis phase, assignment phase, implementation phase (HW/SW/System/other measures), overall installation, validation, overall operations - maintenance-repair, decommissioning, withdrawal.	E/EE/PE systems –technical requirements and management pre-conditions, requirements on processes
ISO 26262	Functional security management, concept phase, system development with HW/SW including their interfaces, production and operation.	Technical requirements and management pre-conditions, requirements on processes and techniques.

Table 4: Lifecycle Definitions in the area of IoT

6.1.2 Interrelated Lifecycles in the IoT Context with their Phases and their Significance from the QE Viewpoint (15 Min)

IoT-QE LO 54 (K2) Know and understand the focus of interrelated lifecycles within the IoT contex [15]

According to ISO/IEC CD 30141 [ISO/IEC CD 30141] the product lifecycle consists of the phases, Build, Develop, Operate, Update und Decommission. Table 5 describes these phases. Quality Engineering is relevant for all of them.

In a typical IoT environment the participating entities (e.g., devices, services, and infrastructure) may easily find themselves in different phases of their individual lifecycles. For Industry 4.0 or the Smart Manufacturing scenarios this has been explicitly addressed in RAMI 4.0.

Industry 4.0 enhances added value by digitalizing existing added value chains and further connections. In doing this the field of view is extended as with, for example, a factory and an association of factories, or entire production chains together with integrated partners in engineering and logistics which are considered right up to the end customer.

Phase	***Description***
Initiate	The development or the integration of an entity to be used in an IoT

Phase	*Description*
	context (e.g. device, service, and infrastructure) is initiated.
Build	The IoT entity is designed and the required pre-conditions such as IoT infrastructure and organizational structures are created.
Develop	The IoT entity is developed and rolled out in the target environment.
Operate	The IoT entity is in its IoT target environment operated and supporting customer services provided.
Update	The IoT entity is maintained by performing corrections and rolling out both functional and non-functional improvements.
Decommission	The IoT entity has reached the end of its lifecycle and is removed from operation.

Table 5: Phases in the IoT Product Lifecycle according to ISO/IEC CD 30141

In order to support the sometimes very different lifecycles, RAMI 4.0 differentiates between Type and Instance when considering lifecycles. Types are the potential objects, and are used in order to coordinate the planning process between the business partners. Instances are the concrete completed objects such as delivery items, machines and also the end product.

A Type exists from the initial idea and ends with its release to production; it therefore represents the basis for production. Finished products are instances of Types. If these instances are delivered to customers they are initially considered there to be Types until they are integrated into a system and thereby become Instances.

In an IoT context the agreement of Quality Engineering activities within the different lifecycle phases of the relevant entities is of great importance. At the same time this means that all the phases within an IoT lifecycle (as shown in Table 5) must be supported by QE activities. Due to the high level of complexity in the IoT and IIoT areas, the planning and execution of verification and validation activities and the selection of suitable Quality Engineering measures and techniques requires specific qualified staff for each phase.

6.2 Understanding the Special Significance of an Interdisciplinary Approach for the IoT Lifecycle [30]

6.2.1 The Interdisciplinary Nature of the IoT Lifecycle (15 Min)

IoT-QE LO 55 (K2) Understand the interdisciplinary nature of the IoT lifecycle [15]

The various perspectives in the IoT lifecycle relate to the particular phase under consideration, the data and services to be provided and for whom they are to be made available. Generally, the collected data are sent to a cloud-based service where they are aggregated with other data and information from various other sources and then used in interaction with the end user.

The enormous variety of possibilities in IoT underscores the need for an interdisciplinary approach. The following aspects have an impact on this variety of possibilities:

- Data collection, aggregation and safe communication
- The selection of available solutions for Gateways
- The software and tools used for the IoT applications
- The widest range of services
- The diversity of IoT platforms and frameworks

This variety of technical infrastructure is extended by the various experts from their respective branches and in this way increases the interdisciplinary nature of the IoT lifecycle.

On top of this comes the interdisciplinary coordination for each phase regarding products and processes at technical, organizational and management (business) levels.

6.2.2 Stakeholders and their Importance in the IoT Lifecycle (15 Min)

IoT-QE LO 56 (K2) Know the stakeholders in the IoT lifecycle and understand their significance [15]

Fundamentally the processes, roles and responsibilities for the relevant services, data and technical infrastructure of each product, device and branch are anchored within the IoT lifecycle.

Additional influences on the Quality Engineering activities come from third parties; these can be complete alliances and ecosystems. Strategic alliances are common in the areas of product development, sales, technology, service and research and development. They allow the marketing and sales organizations of the involved companies to make their product portfolio and the status of their digitalization “IoT enabled”.

An ecosystem is a dynamic mix of associations which are functionally interrelated; it is a relational structure of associations in the IoT space. Simply put, it is a overarching cooperation between companies. For example, there may be associations of suppliers, technology companies, franchise takers and deliverers. The requirements and work procedures within these alliances and ecosystems significantly influence the activities and workflows within the IoT lifecycle and thereby also the associated planning and execution of Quality Engineering.

To put it in less abstract terms, the following third parties belong to those who have a direct influence on the QE tasks within the IoT lifecycle:

- Laws, regulations
- Standards, norms
- Evaluation and certification bodies
- Workers‘ councils, staff councils
- Associations and federations
- Ethical guidelines

Using the example of product liability, QE measures are directly influenced by several of these third parties. Laws contain requirements on the characteristics of a product which is to be rolled out. Independent of whether the product is a service, a device or a tool, there are requirements with regard to the security of the product. These requirements must be obeyed and proof of this provided over the entire lifecycle of the product. This proof is obtained with the help of norms, standards and certificates which require concrete Quality Engineering measures. The standards even include requirements for ethical fundamentals such as, for example, the ISO 26262-2, which explicitly demands a “culture of security-aware thinking”

Associations and federations such as the VDA (Association of German Automotive Industries e.V.) or the VDE (Association of Electro technology, Electronics and Information Technology e.V.) supply standards and guidelines relating to processes and quality requirements.

The IoT lifecycle contains not only the already mentioned services, data, technical infrastructures, communications structures and ecosystems, it also contains the human factor. Workers’ councils, staff councils, trade unions, professional bodies and work protection laws are all examples of influencing third parties. All of the resulting requirements and regulations must be integrated, planned, executed, controlled and optimized for the processes, activities and therefore also the Quality Engineering measures in the IoT lifecycle.

6.3 Continuing QE Activities after Roll-Out [30]

IoT-QE LO 57 (K3) Understand the need to continue QE activities after rollout [30]

As a result of the many devices, different services and disciplines, as well as long lifecycle duration, there are special requirements to be aware of for IoT systems which also apply after delivery and during the operational phase.

Example: UC "Plug-In Electrical Charging Vehicles and power feed in home scenario" from oneM2M Use Case collection [oneM2M 16]

In this Use Case (see Diagram 5) diverse devices and systems from various industrial domains interact with each other and their lifecycles are not synchronized.

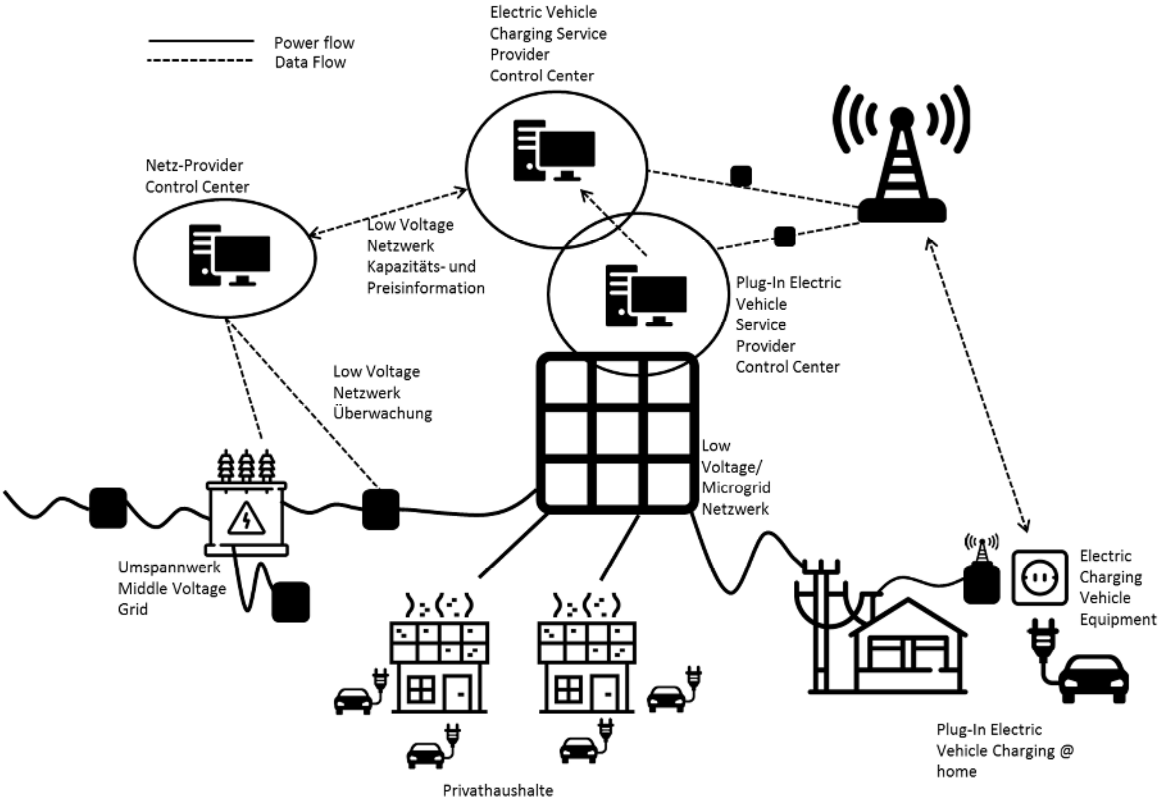
Power flow
Data Flow
Electric Vehicle Charging Service Provider Control Center
Netz-Provider Control Center
Low Voltage Netzwerk Kapazitäts- und Preisinformation
Plug-In Electric Vehicle Service Provider Control Center
Low Voltage Netzwerk Überwachung
Low Voltage/ Microgrid Netzwerk
Umspannwerk Middle Voltage Grid
Electric Charging Vehicle Equipment
Plug-In Electric Vehicle Charging @ home
Privathaushalte

Diagram 5: Changing Process for an Electric Car (Icons made by Freepik, samshizone & Retinaicons from www.flaticon.com)

The application of quality attributes during the operation of a particular participating device or system can be immediately affected if quality defects arise in another system. These quality issues can arise when first used or after a longer period of use.

It is therefore decisive that QE activities support all phases of the lifecycle, even in the phases Operate und Update which follow the rollout. This can be achieved, for example, by monitoring selected quality attributes and introducing corrective proactive measures if threshold values are exceeded.

7 Appendix A Glossary of Terms

8 Appendix B References

[Anderson 11] Anderson, Michael; Anderson, Susan Leigh (Hrsg.), Machine Ethics (2011)

[Bandyopadhyay 11] Debasis Bandyopadhyay, Jaydip Sen: Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Pers Commun. 58, 49-69 (2011)

[Bendel 16] Bendel, Oliver: 300 Keywords Informationsethik: Grundwissen aus Computer- Netz- und Neue-Medien-Ethik sowie Maschinenethik (2016)

[AIOTI 16] AIOTI WG03 – loT Standardisation, High Level Architecture (HLA) – Release 2.1 (2016) https://aioti.eu/wp-content/uploads/2017/03/AIOTI-WG3-IoT-High-Level-Architecture-Release_2_1.pdf

[ETSI 2016] ETSI EG 203 251 V1.1.1, Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies (2016) http://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_60/eg_203251v010101p.pdf

[Humble 10] Jez Humble, David Farley: Continuous Delivery. Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley, Upper Saddle River (2010)

[IEC 61508] Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements, (2010) https://www.vde-verlag.de/iec-normen/preview-pdf/info_iec61508-1%7Bed2.0%7Db.pdf

[IEEE 1028] IEEE Standard 1028-2008 - IEEE Standard for Software Reviews and Audits (2008) https://standards.ieee.org/findstds/standard/1028-2008.html

[ISO 14] ISO JTC 1/SWG 5, Internet of Things (IoT) Preliminary Report 2014 (2014) https://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/internet_of_things_report-jtc1.pdf

[ISO/IEC/IEEE 24765] ISO/IEC/IEEE 24765:2010: Systems and software engineering -- Vocabulary

[ISO/IEC 25010] ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models (2011) https://www.iso.org/standard/35733.html

[ISO 27034] ISO/IEC 27034:2011, Information technology — Security techniques — Application security (2011) http://www.iso27001security.com/html/27034.html

[ISO/IEC CD 30141] ISO/IEC CD 30141, Internet of Things Reference Architecture (IoT RA) (2016) https://www.w3.org/WoT/IG/wiki/images/9/9a/10N0536_CD_text_of_ISO_IEC_30141.pdf

[ISO/IEC/IEEE 42010] ISO/IEC/IEEE 42010:2011, Systems and software engineering -- Architecture description (2011) https://www.iso.org/standard/50508.html

[ISO 9126] ISO/IEC 9126-1:2001, Software engineering -- Product quality -- Part 1: Quality model

https://www.iso.org/standard/22749.html

[ISTQB 11] International Software Testing Qualifications Board: ‘Certified Tester Foundation Level Syllabus’ V 2011 1.0.1 (2011)32

[ISTQB 16] ISTQB® Certified Tester Advanced Level Syllabus “Test Automation Engineer” (2016)
http://www.Istqb.Org/Certification-Path-Root/Test-Automation-Engineer.html

[ISTQB 17] ISTQB®/GTB Standardglossar der Testbegriffe Deutsch / Englisch Version 3.11 (2017)

[Jäkel 17] Frank-Walter Jäkel et al, R2.2: Testarchitekturen (2017)
http://www.iot-t.de/wp-content/uploads/sites/11/2017/07/IoT-T_R2.2.pdf

[Kuhlen 04] Kuhlen, Rainer. Informationsethik: Umgang mit Wissen und Informationen in elektronischen Räumen (2004)

[oneM2M 16] oneM2M Technical Report, Use Cases Collection (2016)
http://www.onem2m.org/images/files/deliverables/Release2/TR-0001-Use_Cases_Collection-V2.4.1.pdf

[RAMI 4.0 15] Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0), Martin Hankel und Bosch Rexroth (2015)

[Riedel 16] Oliver Riedel et al, Modellbasierte modulare Shopfloor IT - Integration in die Werkzeuge der Digitalen Fabrik (2016)
http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-3162488.pdf

[Rötzer 16] Rötzer, Florian (Hrsg.): Programmierte Ethik: Brauchen Roboter Regeln oder Moral? (2016)

[Schieferdecker 16] I. Schieferdecker et al, Das Ende der Unsicherheit – Quality Engineering für IoT (2016)
https://www.sigs-datacom.de/uploads/tx_dmjournals/Schieferdecker_Metzger_Rennoch_IOT_16.pdf

[VDI/ZVEI 15] VDI/ZVEI Statusreport Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) (2015)
https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf

[Wardaschka 17] André Wardaschka et al, R4.1: Stand IoT Labore / Auswahl viel versprechender Protokolle (2017)
http://www.iot-t.de/wp-content/uploads/sites/11/2017/07/IoT-T_R4.1.pdf

9 Appendix C Learning Objectives: Cognitive Levels of Learning

Extract from [ISTQB 11]:

The following learning objective are defined as applying to this syllabus. Each topic in the syllabus will be examined according to the learning objective for it.

Level 1: Remember (K1)

The candidate will recognize, remember and recall a term or concept.

Keywords: Identify, remember, retrieve, recall, recognize, know

Examples:

Can recognize the definition of "failure" as:

- "Non-delivery of service to an end user or any other stakeholder" or
- "Actual deviation of the component or system from its expected delivery, service or result"

Level 2: Understand (K2)

The candidate can select the reasons or explanations for statements related to the topic, and can summarize, compare, classify, categorize and give examples for the testing concept.

Keywords: Summarize, generalize, abstract, classify, compare, map, contrast, exemplify, interpret, translate, represent, infer, conclude, categorize, construct models

Examples:

Can explain the reason why tests should be designed as early as possible:

- To find defects when they are cheaper to remove
- To find the most important defects first

Can explain the similarities and differences between integration and system testing:

- Similarities: testing more than one component, and can test non-functional aspects
- Differences: integration testing concentrates on interfaces and interactions, and system testing concentrates on whole-system aspects, such as end-to-end processing

Level 3: Apply (K3)

The candidate can select the correct application of a concept or technique and apply it to a given context.

Keywords: Implement, execute, use, follow a procedure, apply a procedure

Examples:

- Can identify boundary values for valid and invalid partitions
- Can select test cases from a given state transition diagram in order to cover all transitions