

Automotive Tester

Certification



Version 1.0

March 2011

Automotive Tester Board



Table of Contents

- 1. Module 1: Principles of Testing (7 hours) 4
 - 1.1. Introduction to the Automotive Tester course 4
 - 1.1.1 Introduction to the structure of the "Automotive Tester" course (30 minutes) 4
 - 1.1.2 Testing throughout the software life cycle (60 minutes) 4
 - 1.2. Test process and test management 5
 - 1.2.1 Principles of test management (15 minutes) 5
 - 1.2.2 The fundamental test process (30 minutes) 5
 - 1.3. Specification-based test design techniques (Black Box) for the Automotive Tester 5
 - 1.3.1 Introduction (15 minutes) 5
 - 1.3.2 Equivalence partitioning (20 minutes) 5
 - 1.3.3 Boundary value analysis (20 minutes) 6
 - 1.3.4 Decision tables (30 minutes)..... 6
 - 1.3.5 State transition testing (30 minutes)..... 6
 - 1.3.6 Summary of the specification-based test design techniques (10 minutes) 6
 - 1.4. Structure-based test design techniques (White Box) for the Automotive Tester 7
 - 1.4.1 Introduction (20 minutes) 7
 - 1.4.2 Statement coverage (10 minutes) 7
 - 1.4.3 Branch coverage (20 minutes) 7
 - 1.4.4 Path coverage (10 minutes) 7
 - 1.4.5 Simple condition coverage (15 minutes) 7
 - 1.4.6 Decision coverage (5 minutes)..... 8
 - 1.4.7 Multiple condition coverage (10 minutes)..... 8
 - 1.4.8 Summary of the structure-based test design techniques (10 minutes).. 8
 - 1.5. Reviews..... 8
 - 1.5.1 Purpose and benefit of reviews (20 minutes) 8
 - 1.5.2 Review approach (20 minutes)..... 8
 - 1.5.3 Review types (20 minutes)..... 8
 - 1.5.4 Selection of the test design technique (20 minutes)..... 9
- 2. Module 2: Testing in Automotive Engineering (5 hours) 9
 - 2.1. Software development in automotive engineering 9

2.1.1	Principles of software testing in automotive engineering (30 minutes)	9
2.1.2	Automotive Product Emergence Process (PEP) (30 minutes)	9
2.1.3	Sample levels (30 minutes)	10
2.1.4	Integration of software development into the automotive product emergence process (30 minutes)	10
2.1.5	Concept of levels of integration (30 minutes)	10
2.2.	Testing in a virtual environment	10
2.2.1	Introduction (50 minutes)	10
2.2.2	Hardware-in-the-Loop (HiL) (40 Minuten)	11
2.2.3	Software in the loop (SiL) (30 Minuten)	12
2.3.	Specific features of automotive software development	12
2.3.1	AUTOSAR (10 minutes)	12
2.3.2	Operating modes (10 minutes)	13
2.3.3	Proliferation of options (20 minutes)	13
3.	Module 3: Standards Relevant for Automotive Engineering (9 hours)	13
3.1.	Functional Safety – ISO 26262	13
3.1.1	Introduction (15 minutes)	13
3.1.2	Purpose (30 minutes)	14
3.1.3	Relevance for companies (15 minutes)	14
3.1.4	Automotive Safety Integrity Level (180 minutes)	14
3.2.	Automotive SPICE	16
3.2.1	Introduction and purpose (30 minutes)	16
3.2.2	Level of maturity dimension (30 minutes)	16
3.2.3	Process dimension (60 minutes)	16
3.2.4	Test management in Automotive SPICE	17
3.2.5	Testing as engineering and supporting process according to Automotive SPICE (60 minutes)	18
3.3.	Comparison between Automotive SPICE and ISO 26262 (30 minutes)	18
4.	Glossary	20

Copyright Notice

This document may be copied in its entirety, or extracts made, if the source is acknowledged.

Copyright Notice © gasq Service GmbH (hereinafter called gasq®)

1. Module 1: Principles of Testing (7 hours)

1.1. Introduction to the Automotive Tester course

1.1.1 Introduction to the structure of the Automotive Tester course (30 minutes)

The certified Automotive Tester gains an overview of important test techniques and standards used in the field of automotive software development. The syllabus comprises three modules:

- Principles of testing
- Testing in automotive engineering
- Standards relevant to automotive engineering

Thus, the Automotive Tester syllabus is a separate training program that complements the existing Foundation and Advanced Levels.

1.1.2 Testing throughout the software life cycle (60 minutes)

Testing throughout the software life cycle

The complexity of software development requires test-based quality assurance. Testing is therefore a major aspect of the development process. The individual development models, which can be classified as sequential, iterative and incremental models, differ only in their forms.

Learning objective: (K1) You know the following development models and are able to classify them as sequential, iterative and incremental models: waterfall model, V model, spiral model as well as RUP (Rational Unified Process) and agile development models.

Objective of testing

Testing is intended to determine the software development quality, to discover failures and to create preventive measures to avoid errors. Testing does not imply troubleshooting. Any test comprises a defined test objective and a test process. Tests can be performed at different test levels.

Learning objective:

(K1) You are able to describe the purpose of software testing.

(K2) You are able to describe the following test levels: component test; integration test; system test and acceptance test; hardware test; software test; and functional integration test.

Test process within the context of business processes

The test process itself is not a productive process on its own. The activity of testing always occurs after development; therefore testing depends on the progress of development. The quality doctrine, on the one hand, requires independent testers; an efficient troubleshooting process, on the other hand,

requires quick and formal communication between testing and development. These aspects require that testing is integrated in comprehensive and supporting business processes.

Learning objective: (K2) You are able to describe the purpose and significance of testing in requirement management, configuration management, change management, risk management and deviation management.

1.2. Test process and test management

1.2.1 Principles of test management (15 minutes)

Testing is teamwork. The test team must be directed by a test manager. The test manager, on the one hand, tracks the business objectives by defining concrete test objectives; he/she communicates the test progress within the company. On the other hand, the test manager has to continually manage the test team in line with current circumstances and the partial results already achieved.

Learning objective: (K1) You are able to describe the tasks and necessity of test management.

1.2.2 The fundamental test process (30 minutes)

In order to track a test objective, a test process must be defined which considers the following phases according to the fundamental test process: planning and control; analysis and design; implementation and realization; evaluation and reporting; and related testing activities. Each test phase has roles. We distinguish first of all between the Test Manager, the Test Analyst and the Technical Test Analyst.

Learning objective: (K2) You are able to describe the fundamental test process and to define the individual phases.

(K1) You are able to list the roles of the fundamental test process and assign them to the process phases.

(K2) You are able to describe the components of a test specification.

1.3. Specification-based test design techniques (Black Box) for the Automotive Tester

1.3.1 Introduction (15 minutes)

In specification-based test design techniques, the test draft is created without knowing the inner structure of the test object. Based on the specification or comparable documents and the expectations implied, test scenarios are designed according to specific procedures. Comparing observed results and expected results helps to identify deviations. Not all specification-based test design techniques are used in automotive software development. Accordingly a selection of specification-based test design techniques will be discussed in the following. Additional techniques are conveyed in the general syllabus training courses.

Learning objective: (K1) You are able to describe specification-based test design techniques and what they are used for.

1.3.2 Equivalence partitioning (20 minutes)

In view of the assumption that all representatives of an equivalence partition show the same behavior, testing one representative per each valid and invalid equivalence partition will suffice. Partitions can be created both for input and for output values.

Learning objective: (K2) You are able to describe the failures detected during equivalence partitioning.

(K2) You are able to determine the test case coverage for equivalence partitions.

(K2) You are able to determine the representatives for equivalence partitions.

1.3.3 Boundary value analysis (20 minutes)

An incorrect implementation of boundaries is a frequent source of errors in software development so that boundary errors in particular can occur that are not found in equivalence partitioning. Therefore implementation and interpretation boundary errors are identified when the boundary values of an equivalence partition are reviewed.

Learning objective: (K2) You are able to describe the failures detected during boundary value analysis.

(K2) You are able to determine the test case coverage for boundary values.

(K2) You are able to determine the boundary values of equivalence partitions.

1.3.4 Decision tables (30 minutes)

Decision tables are used to analyze functions with many combinations of input values. Input values can be discrete values as well as representatives from equivalent partitions or boundary values. All possible decisions are listed in tables and converted to test cases.

Learning objective: (K2) You are able to describe the use of decision tables and you know for what types of errors they are used.

(K2) You are able to determine the test case coverage based on a decision table.

(K3) You are able to create decision tables.

1.3.5 State transition testing (30 minutes)

In development and for test case design, finite state machines are created, particularly for embedded systems with discrete states, in order to be able to describe the transitions from one state into another. The models provide different possibilities for test case creation using various metrics for determining the test case coverage: state coverage, n-switch coverage and state transition tables that consider both valid and invalid coverage.

Learning objective: (K2) You are able to describe the use of finite state machines for testing and you can explain for what error types test cases are used based on finite state machines.

(K2) You are able to determine the test case coverage for state coverage, n-switch coverage and using state transition tables.

(K3) You are able to create finite state machines and design test cases according to the degrees of coverage of state coverage, n-switch coverage and using state transition tables.

1.3.6 Summary of the specification-based test design techniques (10 minutes)

The test design techniques conveyed are used for different objectives, are subject to different degrees of testing and achieve different quality levels.

Learning objective: (K3) You are able to decide which test design technique is used for what test applications considering the objectives including the test level, the availability of resources and the expected quality.

1.4. Structure-based test design techniques (White Box) for the Automotive Tester

1.4.1 Introduction (20 minutes)

Structure-based test design techniques consider the inner structure of the software. The structure-based test design technique aims at determining implementation errors. The static analysis of the inner structure creates control flowcharts and data flowcharts which serve to determine the test data. Particularly in standards for safety-relevant systems, test cases are required on the basis of structure-based test design techniques. In the future these will be used increasingly in automotive software development. This syllabus includes a selection of structure-based test design techniques that are worth knowing as part of your fundamental knowledge.

Learning objective: (K1) You are able to describe structure-based test design techniques and what they are used for.

1.4.2 Statement coverage (10 minutes)

Complete statement coverage is achieved when each statement in the software is executed in the test at least once.

Learning objective: (K2) You are able to design metrics to elicit test case coverage for statements.

(K2) You are able to create test cases for complete statement coverage.

1.4.3 Branch coverage (20 minutes)

Complete branch coverage is achieved when each branch of the software is gone through at least once.

Learning objective: (K2) You are able to design metrics to elicit test case coverage for branches.

(K2) You are able to create test cases for complete branch coverage.

1.4.4 Path coverage (10 minutes)

Complete path coverage is achieved when a complete coverage of combinations of all possible branches of the software is executed.

Learning objective: (K2) You are able to design metrics to elicit test case coverage for paths.

(K2) You are able to create test cases for complete path coverage.

1.4.5 Simple condition coverage (15 minutes)

In contrast to the test cases that were exclusively designed on the basis of the control flow and lead to statement, branch or path coverage, another strategy of test design considers decision making in a branch. These lead to the metrics of simple condition coverage, decision coverage, minimal multiple condition coverage, defined condition coverage or multiple condition coverage. Simple condition coverage requires that any single condition is executed once as true and once as false.

Learning objective: (K2) You are able to design metrics to elicit test case coverage for simple conditions.

(K2) You are able to create test cases for complete condition coverage.

1.4.6 Decision coverage (5 minutes)

The term 'decision coverage' is introduced to make it possible to compare the metrics.

Learning objective: (K1) You know the term 'decision coverage'.

1.4.7 Multiple condition coverage (10 minutes)

The complete combination of both the positive and the negative execution of any single condition in a decision results in complete multiple condition coverage being achieved.

Learning objective: (K2) You are able to design metrics to elicit multiple condition coverage.

(K2) You are able to create test cases for complete multiple condition coverage.

1.4.8 Summary of the structure-based test design techniques (10 minutes)

The structure-based test design techniques conveyed cause different quality levels and serve to detect various error types.

Learning objective: (K2) You are able to describe which structure-based test design techniques are to be used for what test objectives.

(K3) You are able to compare the quality levels of the test exit criteria of the various test design techniques.

1.5. Reviews

1.5.1 Purpose and benefit of reviews (20 minutes)

Reviews contribute to the static analysis of documents in order to improve and secure the quality of the work progress particularly in early project phases. Reviews can be carried out for all documents of any release state. Depending on the review goal, various review types can be used which can be distinguished as informal and formal reviews. Formal review types are subject to a defined process with specified role assignment.

Learning objective: (K1) You are able to describe the purpose of using reviews.

1.5.2 Review approach (20 minutes)

On the one hand, all formal review types are subject to a review process: review planning, kick-off, individual preparation, review meeting, revision, and review completion. On the other hand, fixed roles are assigned: Review Manager, author, reviewer or inspector, keeper of the minutes or secretary, and moderator.

Learning objective: (K2) You are able to describe the review process.

(K2) You are able to name and explain the roles of a review.

1.5.3 Review types (20 minutes)

Various review types may be applied depending on the review goals and the test object. Two distinctive features help to classify them: on the one side, review approaches are distinguished; on the other side, the test object types. This syllabus merely considers the distinction in terms of the approaches: informal review, walkthroughs, technical reviews, inspections, management reviews and audits.

Learning objective: (K2) You are able to describe the various review types and can decide which review types are used in what cases.

1.5.4 Selection of the test design technique (20 minutes)

Although there is no standard on how to select a test design technique and hardly any documentation on this topic, the challenge of selecting or combining the right test design techniques should be discussed. Various parameters are decisive for this selection, including security-relevant standards or other industry standards, the classification of risks (for example according to ISO 26262), the current test level, the quality features according to the test objective, or the available project resources.

2. Module 2: Testing in Automotive Engineering (5 hours)

2.1. Software development in automotive engineering

2.1.1 Principles of software testing in automotive engineering (30 minutes)

The business processes in automotive product development are characterized by the construction of physical components. From the entrepreneurial point of view, creation of embedded software is regarded as a downstream enabler, although admittedly a large part of today's functionality is enabled exclusively or to a great extent by software. This function-creating software is called embedded software.

The challenge of creating embedded software is firstly its accessibility, secondly the physical and economic constraints of the embedded system, and thirdly the challenge of providing the customer as much product customization as possible despite series production. Based on the function, on the tough real-time requirements, but also on the economic constraints due to quantity effects, firmware programming using languages that make this possible is required. Object-oriented or model-based languages are only slowly becoming accepted.

Learning objective: (K1) You are able to describe the constraints of software development in automotive engineering, particularly in view of the proliferation of options, quantity effects, firmware programming and the problems of software testing resulting from these.

2.1.2 Automotive Product Emergence Process (PEP) (30 minutes)

According to the PEPs of automotive engineering companies, complex products from various disciplines must be developed within no time. Several processes must be attuned and provide results at the right moment. In addition to development this includes, for example, economic processes, logistics, product planning or service planning. These processes include synchronous milestones. Common problems are discussed in software development teams.

Learning objective: (K1) You are able to describe the basic structure of automotive PEPs and are aware of the challenges they pose.

2.1.3 Sample levels (30 minutes)

Quality assurance within the development process of the PEP is accomplished using sample levels (A, B and C-level samples). A-level samples prove the concept's suitability, B-level samples prove the suitability for series production, and C-level samples are samples manufactured with series production tools which prove manufacturability. Release of B-level samples, in general, has economic consequences, because this release causes the supplier's responsibility to be transferred to the customer.

Learning objective: (K2) You are able to distinguish A-level, B-level and C-level samples.

2.1.4 Integration of software development into the automotive product emergence process (30 minutes)

Software development must integrate into the automotive PEP, which came into being based on the view of classical vehicle construction. The technology and significance as well as the complexity due to increasing functional networking are growing faster than can be taken into consideration by any prevailing process definitions. Thus, software development is frequently ranked behind the constructional activities. The requirements for software development therefore result from dependencies on the construction tasks and the constraints implied by PEP. Accordingly, quality assurance is performed on different levels. Many of the quality features belonging to the software are assured during automotive testing. This causes complex communication channels both in development and in testing.

Learning objective: (K2) You are able to explain the challenges of testing in the automotive emergence process, particularly with regard to communication channels.

2.1.5 Concept of levels of integration (30 minutes)

A new concept, the concept of levels of integration, is emerging to meet the concerns of software development. According to predefined integration levels, this concept envisions the gradual integration of completed functions. Accordingly, the relevant test levels with regard to integration are derived from the integration levels.

Learning objective: (K3) You are able to plan integration levels for development of a vehicle and develop relevant test concepts.

2.2. Testing in a virtual environment

2.2.1 Introduction (50 minutes)

Software in vehicles always depends on the embedding hardware. Creation and validation of the software cannot wait for the hardware's completion. Accordingly, virtual environments must be created to simulate the real environment.

Objective

In order to be able to do without the hardware or the target systems, methods have been developed which allow a simulation of the environment. First of all, these include HiL (Hardware-in-the-Loop) and SiL (Software-in-the-Loop), but also MiL (Model-in-the-Loop) simulations. The tests run in virtual environments following the established test processes using the corresponding documentation in accordance with IEEE 829.

Test planning must include the breakdown from the master test plan down to the individual test cases. Design methods and automatic test case generation from models are used to create the test cases

using the appropriate tools. With test management systems the test plans can be automated and used at the test location or in the vehicle. Based on the test results the following documents can be created:

- Execution statistics
- Conclusions on the test coverage
- Test documents

This approach supports tracking of detected defects and attribution to the relevant requirements.

Learning objective: (K3) You are able to specify a test in the virtual environment and outline the documentation of its execution.

2.2.2 Hardware-in-the-Loop (HiL) (40 minutes)

Purpose of HiL

The "Hardware-in-the-Loop" simulation approach describes testing in a virtual environment of real components that were disembedded from the overall system. This type of simulation is used primarily by the vehicle manufacturer who is responsible for the vehicle as an overall system. Accordingly, the vehicle manufacturer bears the responsibility for the interaction of the control devices provided by various suppliers. This simulation helps to prepare for integration into the vehicle.

Learning objective: (K1) You know the purpose and goal of Hardware-in-the-Loop simulation.

Timing and validity

A HiL simulation can provide the following benefits:

- Simulation before the vehicle is available
- Reproducible test cases
- Simple modification of the system structure
- Safe and nondestructive testing of extreme situations
- Automated test procedure
- Independence from constraints such as environmental impact

The simulation results cannot be transferred unrestrictedly to a real test, because the models are simplified in favor of their real-time capability and cannot reproduce a real test in every respect.

Learning objective: (K2) You know the advantages of HiL simulation and are able to outline them.

Requirement for a HiL simulation

The availability for use of the test object including the hardware is a basic requirement for the performance of a compelling HiL simulation. Accordingly, the development and organization processes must consider that the hardware and electrical interfaces for simulation in a realistic environment are available.

Learning objective: (K2) You know what requirements must be fulfilled for a HiL simulation.

Construction of a test bench

A HiL test bench includes the following components:

- Hardware including control device (test object)
- Simulation environment with real-time capability

- Output and analysis unit

The challenge of HiL simulations is to have these components available as early as possible.

Learning objective: (K2) You know the components of a HiL test bench and are aware of the challenges of guaranteeing that the components interact smoothly.

HiL simulation procedure

During HiL simulation a real control device is connected to a model of its future environment and is tested on it. This enables the analysis of the developed device at an early stage. The complete integrated system consisting of hardware and software is linked with a simulation of the environment using signal flows and is executed under real-time conditions, if possible. The HiL test itself is normally executed as a black box test.

Learning objective: (K3) You know the scope of HiL simulation and are able to estimate the effort the procedure requires.

2.2.3 Software-in-the-Loop (SiL) (30 minutes)

Purpose of SiL

SiL, in contrast to HiL, does not require any special hardware and therefore provides more flexibility for test execution. SiL simulations and tests can be performed at an early stage of software development and provide the possibility of executing tests before the hardware is completed.

Learning objective: (K2) You know the goals of SiL simulation and are able to distinguish it from HiL.

Constraints: Timing and validity

The complete replica of the embedding hardware as a simulation model may result in costly development, because it must be developed using its own development process with appropriate validation.

The time behavior of the software often differs from the time behavior of the hardware. This can be compensated using a synchronization process with a simulated real-time clock. However, the simulation becomes more complex this way.

Learning objective: (K1) You are able to name the most important constraints of SiL simulation.

2.3. Specific features of automotive software development

Automotive software development includes additional specific features for which only a brief overview can be provided here. These range from new initiatives for functional abstraction of the hardware through various operating states, which are not part of the primary focus of consideration, to the challenge of managing any number of variants.

2.3.1 AUTOSAR (10 minutes)

AUTOSAR (AUTomotive Open System ARchitecture) is an initiative of several vehicle manufacturers for abstracting software from the hardware. AUTOSAR provides control devices with an operating system including a virtual communication bus. The initiative furthermore includes requirements with impacts on the test process.

Learning objective: (K1) You are able to reflect the purpose of AUTOSAR.

2.3.2 Operating modes (10 minutes)

Along with the normal operating mode, additional operating modes must be developed and tested, for example the production or the transport mode. The relevant functions as well as the transitions between the operating modes must be considered for testing.

Learning objective: (K1) You are able to describe the impacts of the operating modes on testing.

2.3.3 Proliferation of options (20 minutes)

Customizability of the vehicles also influences the software design options. This results in an almost infinite number of variants, of which only a small portion of the actual variations can be tested. In order to nevertheless achieve as wide as a test coverage possible, reduction algorithms must be used as provided by all-pairs testing or orthogonal array testing.

Learning objective: (K1) You are aware of the problems involved in testing due to the proliferation of options.

(K3) You are able to use the all-pairs and orthogonal array testing methods to reduce the number of test cases.

3. Module 3: Standards Relevant for Automotive Engineering (9 hours)

3.1. Functional Safety – ISO 26262

3.1.1 Introduction (15 minutes)

ISO 26262 will replace IEC 61508 as the standard for functional safety of road vehicles for the automotive industry. This learning unit classifies the historical and legal aspects of the standard, outlines its structure and contents and looks at selected aspects involved in software testing. ISO 26262 ("Road vehicles – Functional safety") is

- an ISO standard for safety-relevant electrical/electronic systems in vehicles that is currently being prepared and
- a process framework and approach model providing the tasks and work products relevant for software testing.
- It defines the methods to be used for testing and development and
- guarantees functional safety of an electrical/electronic system in vehicles during implementation.

Learning objective: (K1) You are able to explain why ISO 26262 is relevant for automotive software testing and you know the aspects it addresses.

3.1.2 Purpose (30 minutes)

The draft for ISO 26262 ("Road vehicles – Functional safety") describes the standard for the functional safety of road vehicles. Since July 2009 it has been available as a Draft International Standard. Its publication as a worldwide valid international standard has been planned for mid 2011. As an industry-specific derivation it will then replace the IEC 61508, the currently valid formal legal standard for road vehicles.

A standard such as ISO 26262 has the following goals:

- Specify requirements regarding safety without limiting the scope of solutions
- Not constrain either innovation or competitive differentiation
- Avoid competitive distortions

ISO/DIS 26262 currently addresses:

- Passenger cars with a total permissible weight of max. 3.5 tons

In the automotive industry module strategies are increasingly gaining ground. For this reason very similar systems are used in various vehicle classes. For example, window actuators for passenger cars differ only modestly or not at all from those used for commercial vehicles.

Since ISO 26262 does not explicitly address commercial vehicles (or buses, motorcycles, etc.), IEC 61508 will continue to be the valid formal legal standard for them.

Learning objective: (K2) You are able to name the most important addressees of ISO 26262 and describe the motivation for the standard.

3.1.3 Relevance for companies (15 minutes)

When a standard is published, it contributes to the state-of-the-art in science and technology. As new products and innovations come into existence at a faster pace than the standard's further development, merely implementing the standard will not suffice. Fulfillment of the standard is required to prove that the state-of-the-art of science and technology is met, but this is not enough.

If a standard's requirements are not fulfilled, however, and if in a product liability case the product is criticized as not meeting the state-of-the-art of science and technology, a reversal of the burden of proof may be sought. This may cause difficulties to a greater or lesser extent. Therefore the company should be interested in fulfilling the standard to reduce an incalculable product liability risk. After publication of the standard, all products must be developed according to the development processes specified therein and demonstrate the required product characteristics.

The phase between publication of the Draft International Standard and the final approved standard can be regarded as the introduction phase. With the publication of the ISO/DIS 26262, the companies should now start to implement the standard and to design their in-house processes accordingly.

Learning objective: (K2) You know the relevance of ISO 26262 for companies and you are able to describe the most important aspects of the standard.

3.1.4 Automotive Safety Integrity Level (ASIL) (180 minutes)

Structure of ASIL (30 minutes)

The "Automotive Safety Integrity Level" (ASIL) measures the safety relevance of a failure. The following three parameters are considered:

- Exposure (E): the frequency of situations in which the failure is relevant
- Controllability (C): the controllability of the failure when it occurs
- Severity (S): the scale of damage, when the failure cannot be controlled

Summary of the automotive standard ISO 26262:

- The standard for the first time describes the functional safety of road vehicles. It serves as a reference for the state-of-the-art of science and technology with regard to the functional safety of road vehicles at the time of its publication.
- It is possible and reasonable to extend its use to other vehicle classes.
- The standard provides a method to determine the Automotive Safety Integrity Level (ASIL).
- However, the three parameters of the basis of evaluation (Exposure E, Controllability C and Severity S) allow plenty of wiggle room.

Learning objective: (K2) You know the principles of ASIL and you are able to describe the most important components and parameters.

ASIL calculation procedure (30 minutes)

Based on the three parameters E, C and S, ASIL is determined on a scale from A to D (or QM for systems that are not relevant for safety), where A is the lowest level and D is the highest one. The requirements of ISO 26262 are finally to be implemented depending on the calculated ASIL.

While in IEC 61508 the method for determining the "Safety Integrity Level" (SIL) is described merely for information, the method for determining the ASIL in ISO 26262 is now specified as a standard. ISO/DIS 26262 gives leeway to determine the three parameters E, C and S.

With regard to Controllability C and Severity S the actual vehicle configuration plays a major role. The method of ISO 26262 considers this only implicitly. The challenge of ASIL classification is to find a method which results in an ASIL structure which is as consistent as possible.

Learning objective: (K2) You are able to describe how to calculate ASIL and how to design the process of evaluation.

(K3) You are able to determine the ASIL of a function.

Risk evaluation techniques (90 minutes)

Testing serves to assure quality. Efficient testing requires that the risk associated with software is known. ISO 26262 includes a technique of risk evaluation based on information about the probability of damage occurrence. Difficulties arise from the fact that information must be gathered at an early stage of product creation. Risk evaluation is differentiated in the following techniques:

- Determination of the scale of damage
- Determination of the damage frequency
- Estimation of the risk of compensation

The following common approaches can be the basis for the practical support of risk evaluation:

- Risk landscape
- Critical incidents reporting
- PAAG / HAZOP (Hazard and Operability)
- FMECA (*Failure Mode and Effects and Criticality Analysis*)
- FTA (Fault Tree Analysis) and impact analysis
- Value at Risk (VaR) technique
- Safety graph: table classification, check lists, Delphi method

Learning objective: (K3) You have gained an overview of the practical approaches to risk evaluation, and you are able to evaluate and identify risks.

How to use ASIL (30 minutes)

Determination of ASIL and the activities involved require time-intensive work in the early phases of product development and test planning. Risk analysis that is comprehensively executed helps to prioritize the tests in the sense of risk-based testing, which results in time and resources being conserved without having to accept significant cuts in product quality. The following aspects are accounted for ASIL and provide major benefits:

- Scheduling
- Determination and administration of test levels (both on the component and system levels)
- Creation and monitoring from the OEM to the supplier

Learning objective: (K2) You are aware of the areas in which the ASIL calculation can be used and of the benefits that can be achieved with it.

3.2. Automotive SPICE

3.2.1 Introduction and purpose (30 minutes)

In 2005 the industry-specific standard Automotive SPICE was published by the Special Interest Group Automotive. The standard was derived from ISO 15504 for software process assessment. This binding approach is used for objective process evaluation with the resulting process improvement on the project and organization levels. Automotive SPICE includes:

- Process reference model (PRM)
- Process assessment model (PAM)

Automotive SPICE basically has two dimensions:

- Process
- Capability

Learning objective: (K1) You know the origin of Automotive SPICE and are able to name its targets and dimensions.

3.2.2 Capability dimension (30 minutes)

The processes of the standard are based on ISO 12207 which was extended or adapted by automotive-specific supplements. The capability levels correspond to the six levels of process maturity as defined in ISO 15504. Assessments can be performed using an ISO 15504-compatible assessment model.

Learning objective: (K2) You are able to explain the assessment model with all six levels of process maturity.

3.2.3 Process dimension (60 minutes)

Within the framework of an Automotive SPICE assessment the maturity of any single process is evaluated. It includes indicators for all processes to evaluate to which extent the processes are executed. The processes used in Automotive SPICE can be distinguished in two groups:

- Primary life cycle processes
- Organizational life cycle processes

These are divided further into:

- (MAN) Management Process Group
- (ENG) Engineering Process Group
- (SUP) Supporting Process Group
- (ACQ) Acquisition Process Group
- (RIN) Resource and Infrastructure Process Group
- (OPE) Operation Process Group

In order to track the test objective, a test process must be defined which considers the test-specific components of the Automotive SPICE processes specified.

Learning objective: (K2) You know all processes including the indicators for an evaluation of the extent to which the processes must be executed, and you are able to describe these processes including their corresponding sub-items.

(K2) You understand the relevance of testing the processes and are able to explain them.

3.2.4 Test management in Automotive SPICE

Testability analysis (15 minutes)

Testability analysis forms the basis for software testing. It aims at guaranteeing the testability of the current software version. Evaluation of the testability basically requires the knowledge of

- System architecture
- System features

These are not exclusively based on the requirements.

Learning objective: (K1) You know the purpose of the testability analysis.

Testing techniques in Automotive SPICE (45 minutes)

Automotive SPICE allows for both static analyses and dynamic testing techniques for quality evaluation. A static analysis analyzes the test object without running it. A major advantage is the fact that no executable code is necessary. In dynamic tests the software is run. Executable software is required which is run under defined constraints with specific input values. In particular, vehicle control device function tests are performed in this way. All three common test specification approaches are used:

- Black box testing
- Gray box testing
- White box testing

Learning objective: (K2) You know the difference between static and dynamic analysis as well as between black box testing, gray box testing and white box testing with regard to Automotive SPICE.

Test documentation (90 minutes)

Automotive SPICE uses the established industry standards of IEEE 829 for the creation of test documentation, too. Documents such as test strategy, test concept, master test concept, level test concept, test plan, test specification, test protocol, problem report and test summary report are taken from IEEE 829.

Learning objective: (K2) You are able to explain the required documents and you can work with the documents of IEEE829.

3.2.5 Testing as engineering and supporting process according to Automotive SPICE (60 minutes)

The test-oriented engineering and supporting processes in Automotive SPICE mix test levels and other techniques of quality assurance:

- Software integration test
- Software test
- System integration test
- System test
- Quality assurance
- Reviews

Learning objective: (K2) You know the engineering and supporting processes of Automotive SPICE and are able to explain their relevance with regard to software testing.

3.3. Comparison between Automotive SPICE and ISO 26262 (30 minutes)

The two above mentioned standards are not built upon each other and pursue different goals. Therefore their requirements do not merge seamlessly. Their use and any competing aspects must be coordinated according to the company's test strategy and the corresponding project goals. The strengths and weaknesses of the two standards must be compared.

Learning objective: (K4) You are able to combine the strengths of the standards Automotive SPICE and ISO 26262 for the creation of the test concept as required by the project and the company's test strategy.

Bibliography:

Balzer, H. (1998): Lehrbuch der Software-Technik : Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Heidelberg, Berlin, Spektrum Akademischer Verlag.

Bender, K. (2005): Embedded Systems – qualitätsorientierte Entwicklung. Berlin Heidelberg

Heinrich, Lutz J. (1992): Informationsmanagement : Planung, Überwachung und Steuerung der Informations-Infrastruktur. 4., vollständig überarbeitete und ergänzte Auflage. München, Wien, Oldenbourg.

Thaller, G. E. (1997): Der Individuelle Software-Prozess : DIN EN ISO 9001 für Klein- und Mittelbetriebe. Kaarst, bhv.

Thaller, G. E. (2000): Software-Test : Verifikation und Validation. Hannover, Heise. Erstes Kapitel der Auflage von 2002:

Winter, Mario (1999): Qualitätssicherung für objektorientierte Software - Anforderungsermittlung und Test gegen die Anforderungsspezifikation. Schirmacher, Arne (2001/2002): Testdokumentation nach ANSI/IEEE 829.

VDA Qualitätsmanagement Center, Automotive SPICE

Markus Müller, Klaus Hörmann, Lars Dittmann, Jörg Zimmer, Automotive SPICE in der Praxis 1. Auflage dpunkt Verlag Heidelberg 2007 Automotive SPICE

Olaf Kindel, Mario Friedrich: *Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis.* dpunkt.verlag, 2009

Kai Borgeest: *Elektronik in der Fahrzeugtechnik.* ATZ/MTZ-Fachbuch, 2008

ISO/DIS 26262-1, *Functional safety*

Automotive SPICE, *Prozess Assessment Model Release v2.3, 2007*

4. Glossary

HiL analysis unit

An analysis unit in the HiL bench processes the input signals and calculates the interim values or output signals using existing information.

Assessment

Gathering of the characteristics of the performance and processes of an organization compared to a model with the goal of evaluating and improving the processes or process capability.

HiL input/output unit

The input/output unit provides the connection to the outside world in form of exchanging data with the virtual environment of the HiL simulation. The signals between the real special hardware (simulation hardware) and the computer unit are exchanged using an input/output unit.

AUTOSAR

AUTomotive Open System Architecture (AUTOSAR) is an international standard in the automotive industry. It describes an open and standardized software architecture for vehicle development which is developed and shared by automotive manufacturers, automotive suppliers and tool manufacturers.

Operating mode (automotive)

Defines a special profile which is optimized for a certain task (for example, transport or production).

Real-time capability

Real-time capability of a system means that a system must react to an event within a specified time frame.

Trial

This is a technique for testing the test object in view of the basic goals of development or production. The product emergence process must progress correspondingly.

FTA (Fault Tree Analysis)

The Fault Tree Analysis is a technique of reliability and safety analysis. The goal is to determine possible combinations of reasons that may lead to certain unwanted events. When an FTA is performed, a graphical logical tree structure is created to show the correlations.

Hardware-in-the-Loop (HiL) simulation

HiL is a simulation technique, which executes a real electronic control device or a mechatronic component using its inputs and outputs in a virtual system environment.

Integration level

Incremental planning of the integration of software functions.

MISRA-C

This is a standard for programming policies of the C language in the automotive industry which was compiled by MISRA (Motor Industry Software Reliability Association).

PAAG or HAZOP

PAAG stands for *Prognose* (prognosis), *Auffinden der Ursache* (finding the reason), *Abschätzen der Auswirkungen* (estimating the effects) and *Gegenmaßnahmen* (counteraction) and is a safety technology method. It was developed to investigate the safety of technical systems. HAZOP stands for **Hazard** and **Operability** and has similar approaches to PAAG.

Automotive product emergence process (Automotive PEP)

The automotive product emergence process describes the workflows from the idea for a new vehicle to its development, production and marketing.

Safety Integrity Level (SIL)

Method for classifying safety-critical systems in safety levels.

Software-in-the-Loop (SiL) simulation

SiL is a software testing and/or calibration approach where ECU software is connected in the lab to a virtual environment using inputs and outputs.

System structure

The system structure is derived from the limits of the system, the relationships between the elements and the interactions of the system elements with the environment.

Simultaneous engineering

Simultaneous engineering represents the integrated and parallel execution of both product and process design, which are work processes that are normally executed serially.

Virtual environment

Intended environment which is simulated by a computer to emulate the real world.