SYLLABUS

Course Name: **Embedded Software Testing**

Course No.: **MA-C1**

Course Duration: **45 days**

Course Description: The embedded software test training highlights the development and test process as well as the related dependencies, extensions and interactions. This way, you can exploit synergies and perform tests efficiently. Numerous hands-on exercises on software and hardware help you implement what you learned in the training.

1. Introduction to Embedded Systems
    a. System Development
    b. Embedded Software

2. Foundation of software testing
    a. Testing terminology
    b. Fundamental testing process

3. Development and Test Process in the W-Model (V-Model Extension)
    a. Testing in the software lifecycle
        i. Life cycle models
        ii. Test-oriented development process
        iii. Testing real embedded systems
        iv. Case studies of tests of embedded software

4. Test-Driven Development, TDD
    a. Advantages of TDD
    b. Embedded TDD strategies
    c. TDD example

5. Development stages: analysis, design, implementation
    a. Test stages: component test, integration test, system test, acceptance test
    b. Test types: functional, non-functional, structure-oriented test
    c. Post test, regression test, maintenance test
    d. Standards relevant to testing
    e. Developing testable software
    f. Definition of terms: unit, module, component
    g. Difference between debugging and testing

6. Code Metrics
    a. Lines of code, cyclomatic number according to McCabe, Halstead metric
    b. Using metrics in the test process

7. Design for Test
    a. S.O.L.I.D. principles
    b. Single responsibility, open closed
    c. Liskov substitution
    d. Interface segregation, dependency inversion

8. Testing Object-Oriented Software
    a. Testing class hierarchies
    b. Testing methods of a class
    c. Testing class relations
    d. Testing inheritance
    e. Testing polymorphic class hierarchies

9. Requirements and Test Cases
    a. Requirements engineering fundamentals and terms
        i. Factors influencing requirements engineering
        ii. Incorporation into development processes

    b. Agile models
        i. V Model
        ii. Types of requirements
        iii. Functional, behavioural and structural requirements
        iv. Non-functional requirements

    c. Interface requirements Elicitation, analysis and documentation of requirements
        i. Defining goals
        ii. System context and system boundary
        iii. Elicitation techniques, e.g. interviews, workshops and use cases
        iv. Structure and content of requirements specifications
        v. Documentation using natural language as well as graphical and formal methods

    d. Requirements validation
        i. Acceptance criteria
        ii. Quality criteria
        iii. Reviews and inspections

    e. Requirements management
        i. Traceability of requirements

ii. Measurements and status tracking
iii. Prioritization of requirements
iv. Change management

10. Test case management
    a. Test creation
    b. Equivalence Partitioning
    c. Boundary Value Analysis
    d. Decision Table
    e. State Transition
    f. Exploratory Testing
    g. Error Guessing

11. Static Tests
    a. Review process: document review, coder review, inspection, walkthrough
    b. Static analysis
    c. Control flow analysis, data flow analysis
    d. Using static analysis for improving maintainability
    e. Tool-aided static code analysis
    f. MISRA C 2012 Introduction & Guidelines

12. Dynamic Tests
    a. Black-box: equivalence class construction, threshold testing, decision table testing, state transition testing, use case testing
    b. White-box: statement testing/ coverage, decision testing/ coverage, condition testing/ coverage
    c. Experience-based techniques: error guessing, explorative testing
    d. Systematic approaches for developing test cases
    e. Selection criteria for test methods
    f. Test method assessment

13. Integration of System Components – Overview
    a. Integration test, system test, acceptance test
    b. Integration of Hardware and Software
    c. Test execution on hardware

14. Further Activities in the Test Process – Overview
    a. Test management, planning, control
    b. Risk management, fault and incident management
    c. Configuration management and version control
    d. Software quality features according to ISO 9126
    e. Test documents according to IEEE 829
    f. Test tools types, selection and implementation

15. Dynamic Testing With Tessy
    a. Project preparation
    b. Test Environment Editor (TEE)
    c. TESSY project setup

16. Structure of the graphical user interface
    a. Project structure setup
    b. Add and analyze Modules
    c. Test Interface Editor (TIE)

17. Create test cases
    a. Test case design (test case and test steps)
    b. Test Data Editor (TDE)
    c. Usage of pointers
    d. Advanced Stub functions

18. Test execution
    a. Test runs
    b. Batch tests

19. User code basics (repeat counters, call trace, evaluation macros)
    a. Results
    b. Displayed test results
    c. Coverage measurement (Coverage Viewer)
    d. Creating test reports
    e. Database Backup and reuse of tests

20. Classification Tree Method (introduction)
    a. CTM introduction (test design)
    b. CTE introduction with an example
    c. CTE workshop – test design related to the specification
    d. CTE project – test creation from the design to the test execution

21. Test management
    a. Test management in context
    b. Risk-based test and other approaches for test prioritization
    c. Test documentation
    d. Test estimation
    e. Defining and using test metrics

22. Hands-on Exercises
    a. Exercises on test analysis, test design, test implementation, test execution, test report
    b. Execution of code Review
    c. On Target Testing: Execution of Blackbox and Whitebox tests with Tessy, with and without HW
    d. Test data determination using the classification tree method with CTE
    e. Determining code metrics using the tools
    f. The Tasking/Arm/ Keil µVision and a Cortex™-M evaluation board are used for tests on HW

References: Class notes and Course materials