MCAL Development Overview

Content

RVC MCAL includes:

1. RD: Requirement Development : để tester và dev biết ‘what to do’ and ‘has it done’

* RD chuyển yêu cầu của khách, thị trường,… thành các tính năng
* Requirement specfication được tạo ra dựa trên project plan
* How to do RD: tìm các requirement không rõ ràng, dang dở, mơ hồ,… và sửa lại bằng cách tìm hiểu xem : có trùng với yêu cầu khác ? vi phạm kiến trúc AUTOSAR hay HWM ?,…; bên cạnh đó check Impact work product và review lại RD (bởi team member)

1. FD: Functional Design: xác định các thành phần của phần mềm, gán các requirement vào tính năng và xác định interface giữa các component.

* Functional designing + Functional design document cần theo sát với Requirement Specification

1. Detail design (DD): định nghĩa sâu hơn các hành vi để đáp ứng yêu cầu từ FD document và các giao tiếp giữa các software unit và processing logic của unit đó ở level có thể tác động đến code và thiết kế của Unit Test Specification

* DD document phải theo sát FD document
* RVC có general DD (common source in generation tool) và module DD (specific source)

1. Coding and debug process (CD phase)

* Generation Tool (TCODE – C# internal) : tạo source và header files từ input (CDF, translation files,…) kết hợp với Driver code để hiện thực MCAL modules
* Driver (ECODE – C external) : function/APIs để hiện thực MCAL modules
* Generation Tool đc sử dụng bởi user kết hợp với Driver source code để compile
* PDF (arxml type - external): trữ mọi parameters của module, sử dụng generation tool để lưu parameters (Davinci, Spectrum,…)
* Sample application: bao gồm CDF, linker files và sample test program (\*.c, \*.h), được dùng để compile với sản phẩm của CD để đảm bảo tính hiện thực của driver
* Stub files: \*.c, \*.h hoặc arxml của modules khác liên quan đến MCAL driver modules hiện tại
* BSWMDT (arxml): XML format chứa thông tin của MCAL modules: API, properties, memory map định nghĩa bởi AUTOSAR
* Static analysis: analyse phần mềm ngay cả khi phần mềm chưa chạy, ngược lại với dynamic analysis
* UM: User Manual : hướng dẫn user sử dụng như API, tính năng, mã lỗi. ECODE và TCODE có UM riêng

1. Testing

* Gồm Integration Testing (IT), Unit Testing (UT), System Test (ST)
* GIT(Generation Integration Test) : confirm error, GUT (Generation Unit Test) for TCODE and DIT (Driver Integrated Test), DUT(Driver Unit Test) for ECODE
  + GIT: đảm bảo mọi case lỗi, cảnh báo được test
  + DIT: test các functional behavior dựa vào requirement, có thể cần board để test
  + GUT: test từng unit sử dụng White Box Testing method
  + DUT: test từng unit trên thanh ghi + functional bahavior với State Transition Testing -> đảm bảo read/write đúng
* Sysytem testing: đảm bảo hoạt động đúng như ý khách khi hoạt động trên Multi-Core environment

1. Peer review

Gồm:

* Peer review: được review bởi member trong team và chuyên gia về work product
* Design review: kiểm tra tính hoàn thiện của từng process trong Design Phase bởi member và các thành viên có sự quan tâm và Quality Department (chưa cần software qualification)
* Release review: kiểm tra trước khi có được xuất bản hay ko (trừ MP)