

FORD LDM Localization - 功能 #3402

Task # 3240 (进行中): FORD文档输出

功能 # 3349 (进行中): SWQA文档

功能 # 3365 (进行中): TDR_RQT_003701_705586 Software Testing

TDR_RQT_003701_705586 002 Software Unit Testing

2025-03-24 08:09 - 力常 张

状态:	进行中	开始日期:	2025-03-21
优先级:	普通	计划完成日期:	
指派给:	槐 杨	% 完成:	50%
类别:		预期时间:	0.00 小时
目标版本:	FORD_TDR_Documents	耗时:	0.00 小时
描述 参考软件需求：DR-003701-708354 Software Unit Testing 需求描述 本设计规则支持RQT-003701-705586 “ 软件测试 ”，适用于所有采用非AUTOSAR操作系统、AUTOSAR Classic、AUTOSAR Adaptive或第三方软件的电子模块软件，并作为该需求的符合性验证方法。 软件单元测试要求： 所有含软件的电子模块（无论软件为手工编写、自动生成或由第三方提供），均需根据其ASIL等级（如表1所列）进行单元测试。 表1-软件单元测试要求 ASIL评级最低要求的单元测试覆盖率 QM100%报表覆盖率 A/B100%报表覆盖率100%分行覆盖率 C/D100%报表覆盖率100%MC/DC覆盖率 设计规范 单元测试确保已构建的每个源代码模块能按照详细设计阶段定义的规格要求运行。单元测试采用白盒（玻璃盒）测试技术，通过代码结构设计测试用例，覆盖所有语句、判断、谓词、域及循环终止条件。 最有效的项目能在项目生命周期中尽早识别并修复错误。单元测试可在开发周期早期发现问题，其目标是对程序的每个部分进行隔离验证，证明各独立单元的正确性。若错误未能被早期检测到，项目将面临更高的时间与成本 本目标失控风险。单元测试和同行评审能提升PCE（阶段遏制有效性）。PCE是衡量软件团队在防止软件问题从错误/问题产生的阶段/源头逃逸方面的有效性指标。 不实施单元测试和同行评审会增加软件错误逃逸至编码/实现阶段之外，最终成为客户所遇缺陷/故障的概率。错误是在正确时间早期被发现的问题，而缺陷/故障则是在错误时间后期被发现的问题。 汽车安全完整性等级（ASIL）：标准定义了四个ASIL等级：ASIL A、ASIL B、ASIL C、ASIL D。ASIL D对产品提出最高完整性要求，ASIL A则要求最低。被标识为QM（质量管理）的危害不规定任何安全性要求。 修正条件/判定覆盖：功能覆盖与分支覆盖的组合有时也称为判定覆盖。该准则要求程序中每个入口点和出口点至少被调用一次，且程序中的每个判定均取遍所有可能结果至少一次。条件/判定覆盖要求同时满足判定覆盖和条件覆盖。对于安全关键型应用，通常要求满足修正条件/判定覆盖（MC/DC）。 代码覆盖率：代码覆盖率是描述程序源代码被测试覆盖程度的指标，属于白盒测试的一种形式，用于识别未被测试用例执行的程序区域。 语句覆盖：语句覆盖是一种白盒测试设计技术，要求源代码中所有可执行语句至少被执行一次。 语句覆盖率=（已执行语句数/总语句数）*100； 语句覆盖-程序中的每条语句是否都被执行？ 语句覆盖范围覆盖什么？1.未使用的语句2.死代码3.未使用的分支 Decision Coverage: Decision coverage reports the true or false outcomes of each Boolean expression. In this coverage, expressions can sometimes get complicated. Therefore, it is very hard to achieve 100% coverage. Decision Coverage = (Number of Decision Outcomes Exercised)/Total Number of Decision Outcomes) Branch Coverage: In the branch coverage, every outcome from a code module is tested. For example, if the outcomes are binary, you need to test both True and False outcomes. Branch coverage - Has each branch (also called DD-path) of each control structure (such as in if and case statements) been executed? For example, given an if statement, have both the true and false branches been executed? Branch Coverage = (Number of Executed Branches)/(Total Number of Branches) Branch coverage testing offers the following advantages:			

- Allows you to validate all the branches in the code
- Helps you to ensure that no branched lead to any abnormality of the program's operation
- Allows you to find those areas which are not tested by other testing methods

历史记录

#1 - 2025-03-24 09:16 - 力常 张

- 描述 已更新。

#2 - 2025-03-24 16:17 - 稚媛 黄

- 主题 从 DR-003701-708354 软件单元测试 变更为 TDR_ROT_003701_705586 002 Software Unit Testing

- 描述 已更新。

#3 - 2025-04-08 10:24 - 涛 陆

- 状态 从 新建 变更为 进行中

- 指派给 从 力常 张 变更为 槐 杨

- % 完成 从 100 变更为 50

请确认主机需求，并调整单元测试规则及报告

#4 - 2025-06-03 10:51 - 稚媛 黄

- 主题 从 TDR_ROT_003701_705586 002 Software Unit Testing 变更为 TDR_RQT_003701_705586 002 Software Unit Testing