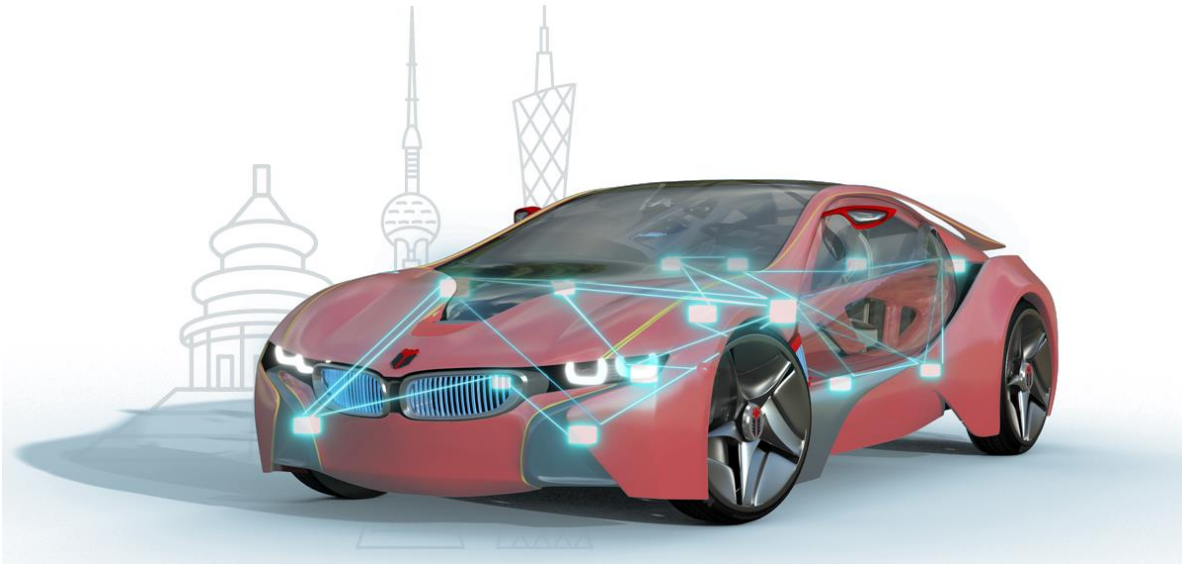
知从 MCAL 开发流程介绍
ZC MCAL DEVELOPMENT PROCESS INTRODUCTION

知从工程服务

ZC ENGINEERING SERVICE



知从 MCAL 开发流程介绍
ZC MCAL DEVELOPMENT PROCESS INTRODUCTION

知从工程服务 ZC ENGINEERING SERVICE

1 概述 INTRODUCTION

知从科技可为客户提供 MCAL 的软件产品开发服务。MCAL 软件包依赖于不同的芯片，一般由各家芯片厂商提供，并通过 EB Tresos 或者知从木牛配置工具进行配置和生成代码。

ZC can provide MCAL software product development services for chip companies. MCAL software packages rely on different chips and are generally provided by various chip manufacturers, and are configured and code generated through EB Tresos or ZC.MuNiu configuration tool.

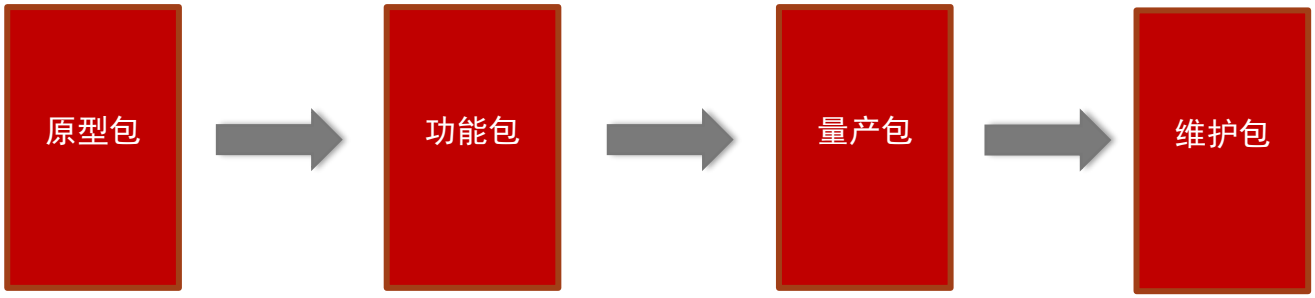
知从科技在汽车电子领域具有丰富的工程经验，采用英飞凌、恩智浦、瑞萨、意法等多家芯片，完成过 BMS、MCU、HCU、BCM、Gateway 等控制器产品的量产开发工作。目前已经同英飞凌、恩智浦、意法半导体等多家芯片厂商建立合作关系。

ZC has rich engineering experience in the field of automotive electronics. It has completed mass production and development of controller products such as BMS, MCU, HCU, BCM, Gateway, etc. using Infineon, NXP, Renesas, and ST. At present, it has established cooperative relationships with many chip manufacturers such as Infineon, NXP, and ST.

2 MCAL 开发服务介绍 MCAL DEVELOP SERVICE INTRODUCTION

知从科技可根据客户需求实现 MCAL 开发服务，并发布原型包，功能包，量产包，维护包。

ZC can implement three stages of MCAL configuration development services based on customer requirements. The service stages include: Demo engineering configuration, customer demand engineering configuration, and customized development services.



- 原型包 Prototype Package

目标：让客户了解知从产品；收集完整客户需求。

Target: To make the customer know ZC products ; to collect complete customer requirements.

内容：针对确定芯片和编译器，包含基础的软件模块。

Content: Contains the basic software modules for the deterministic chip and compiler.

质量：实现部分需求，基本测试。不可上车实验。

Quality: Implement partial requirements, basic testing. Do not experiment in the car.

- 功能包 Function Package

目标：实现客户完整需求。

Target : To implement customer's complete requirements.

内容：最多可以分两次发布；芯片和编译器选项不可再更改。

Content: Release at least twice; cannot change the chip and compiler options.

质量：完成客户全部需求，基本测试。不可上车实验。

Quality: Complete all customer requirements, basic testing. Do not experiment in the car.

- 量产包 Mass Production Package

目标：没有新的需求变更，保证客户量产需求。

Target: There is no new requirement changes so that can ensure customer requirements of mass production.

内容：客户需求完全实现的软件版本。

Content: The version of the software that the customer requirements fully realized.

质量：经过完整测试后的版本，可以上车。

Quality: The fully tested version is ready to be tested in the car.

- 维护包 Maintenance Package

目标：整车量产后的软件维护。

Target: Software maintenance after vehicle mass production.

内容：客户反馈软件缺陷的修复。

Content: Repair of customer's feedback about software defects.

质量：及时响应的缺陷修复，完整的回归测试。

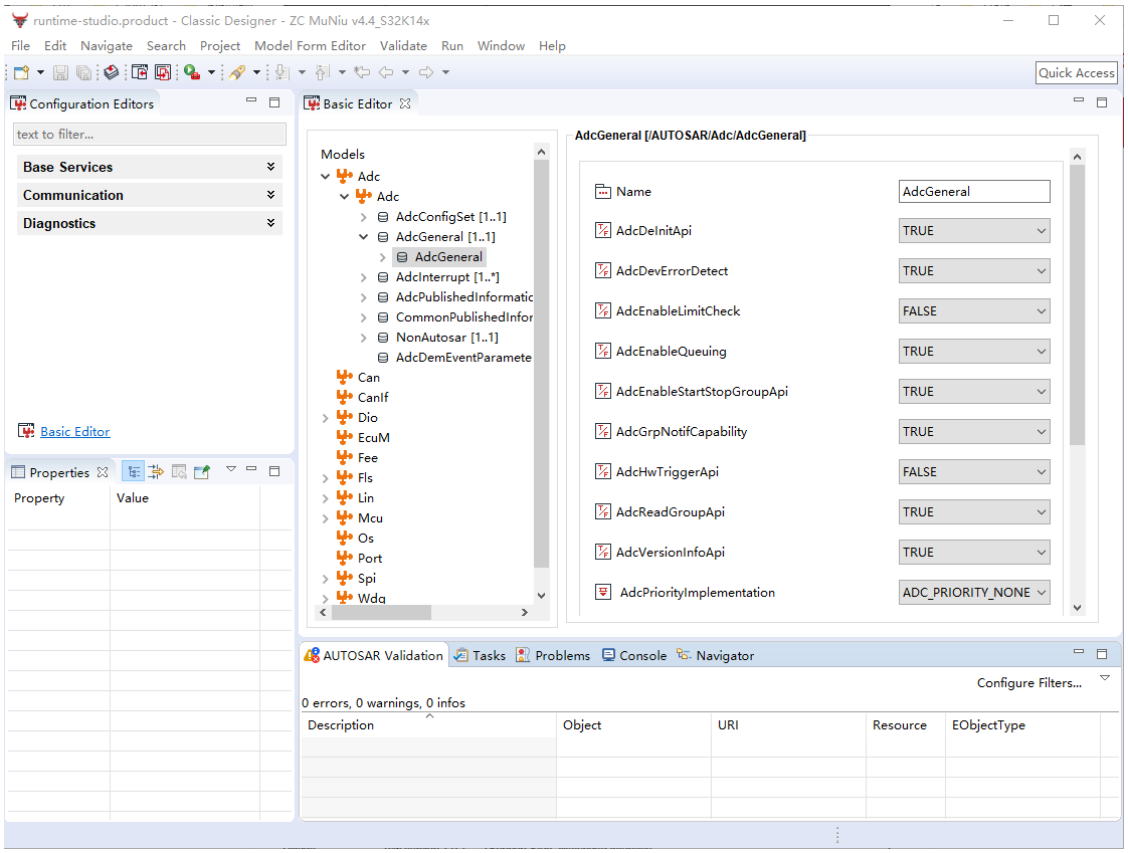
Quality: Prompt response for defect repair, complete regression testing.

周期：量产包交付后一年内免费维护。

Period: Free maintenance within one year after the mass production package is delivered.

除此之外，如果客户对芯片驱动有特定需求，并且 MCAL 中无法满足，则知从科技会根据客户需求，完成 MCAL 的定制开发服务。知从科技使用知从.木牛配置工具实现 MCAL 定制开发的配置功能。

If the customer has specific requirements for chip drivers that cannot be met in MCAL, ZC will complete customized development services for MCAL based on the customer's requirements. Know from the use of technology. ZC.MuNiu configuration tool implements the configuration functions customized by MCAL.


runtime-studio.product - Classic Designer - ZC MuNiu v4.4_S32K14x
File Edit Navigate Search Project Model Form Editor Validate Run Window Help
Quick Access
Configuration Editors
text to filter...
Base Services
Communication
Diagnostics
Basic Editor
Properties
Property
Value
Basic Editor
Models
Adc
Adc
AdcConfigSet [1..1]
AdcGeneral [1..1]
AdcGeneral
AdcInterrupt [1..*]
AdcPublishedInformatic
CommonPublishedInfor
NonAutosar [1..1]
AdcDemEventParamete
Can
CanIf
Dio
EcuM
Fee
Fls
Lin
Mcu
Os
Port
Spi
Wdg
AdcGeneral [/AUTOSAR/Adc/AdcGeneral]
Name
AdcGeneral
AdcDeInitApi
TRUE
AdcDevErrorDetect
TRUE
AdcEnableLimitCheck
FALSE
AdcEnableQueuing
TRUE
AdcEnableStartStopGroupApi
TRUE
AdcGrpNotifCapability
TRUE
AdcHwTriggerApi
FALSE
AdcReadGroupApi
TRUE
AdcVersionInfoApi
TRUE
AdcPriorityImplementation
ADC_PRIORITY_NONE
AUTOSAR Validation
Tasks
Problems
Console
Navigator
Configure Filters...
0 errors, 0 warnings, 0 infos
Description
Object
URI
Resource
EObjectType

3 知从 MCAL 开发流程介绍 ZC MCAL DEVELOP SERVICE INTRODUCTION

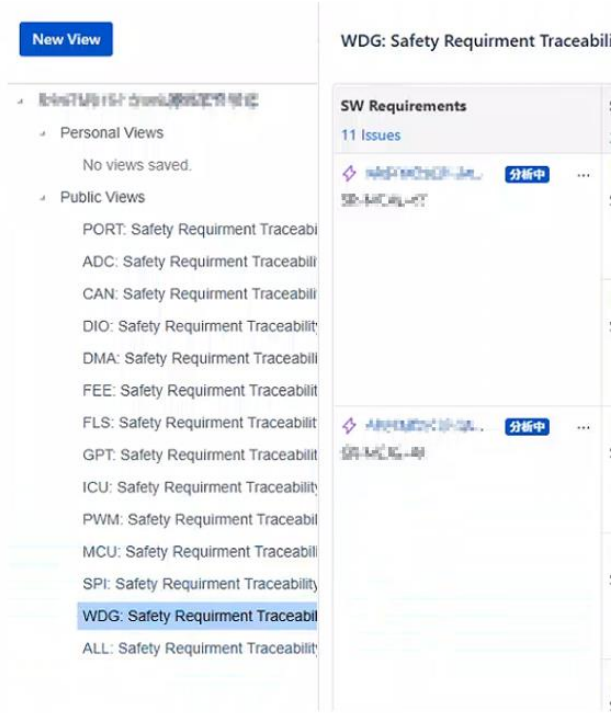
3.1 SYS01 系统需求收集 SYS01 System Requirements Gathering

首先知从会从客户方收集系统需求，包括但不限于：市场需求，产品需求，Autosar 需求，法律法规需求，硬件手册等。

First, the project team will collect system requirements from the client, including but not limited to: market requirements, product requirements, Autosar requirements, legal and regulatory requirements, hardware manuals, etc.

收集完成后会将客户需求导入需求管理系统，并进行追溯

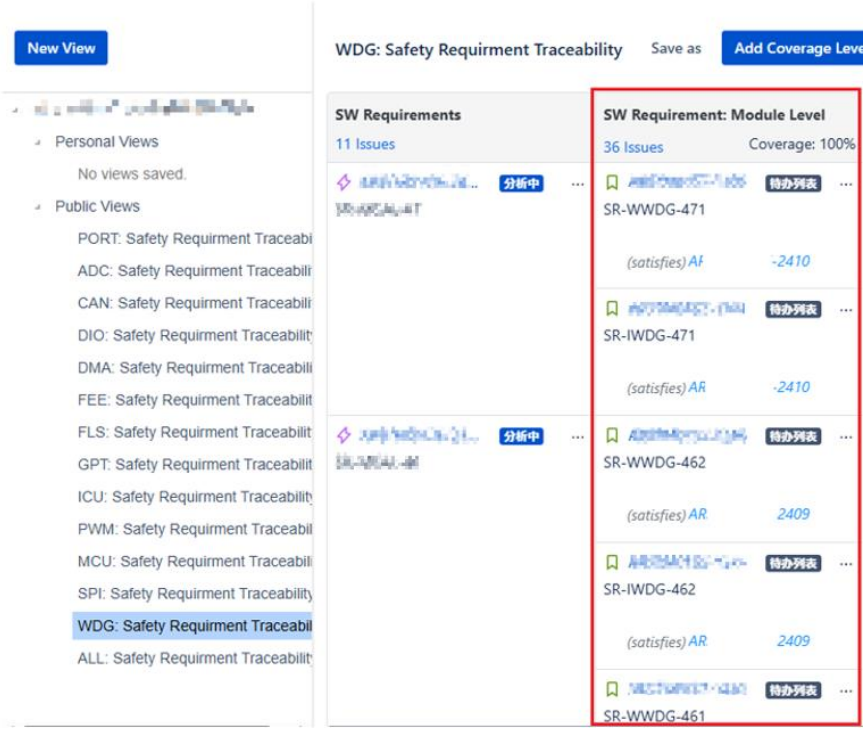
After collection is completed, customer requirements will be imported into system for management and traced



3.2 SWE01 软件需求分析 SWE01 Software Requirements Analysis

将客户需求进行识别，整理出软件需求，并在需求管理系统上进行需求分析，并进行追溯，将软件需求与客户系统需求进行追溯。

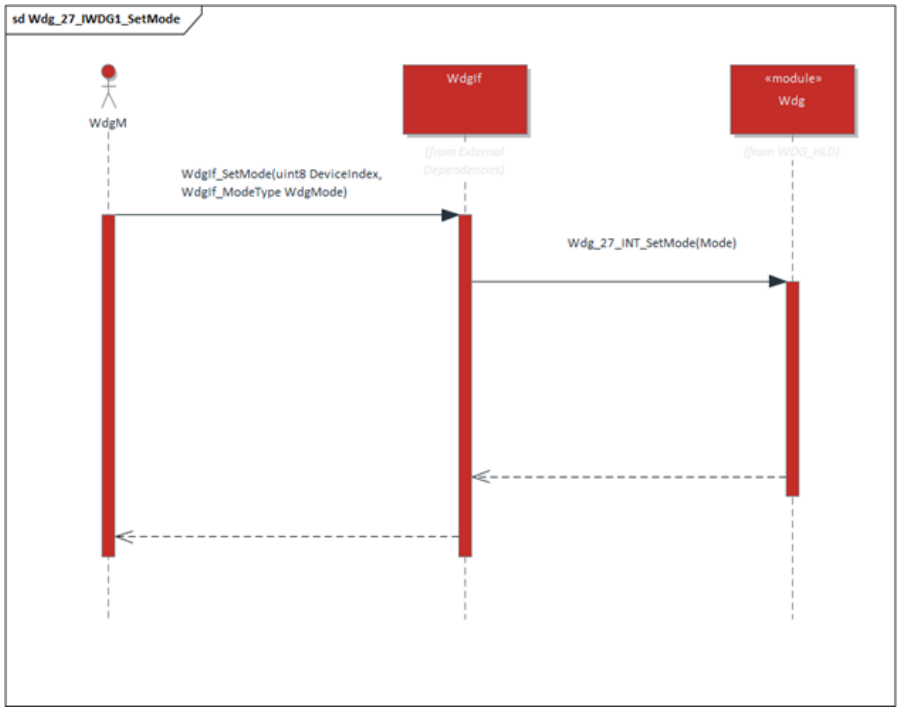
Identify customer requirements, organize software requirements, conduct requirements analysis on system for traceability, and trace software requirements to customer system requirements.



3.3 SWE02 架构设计 SWE02 Architecture Design

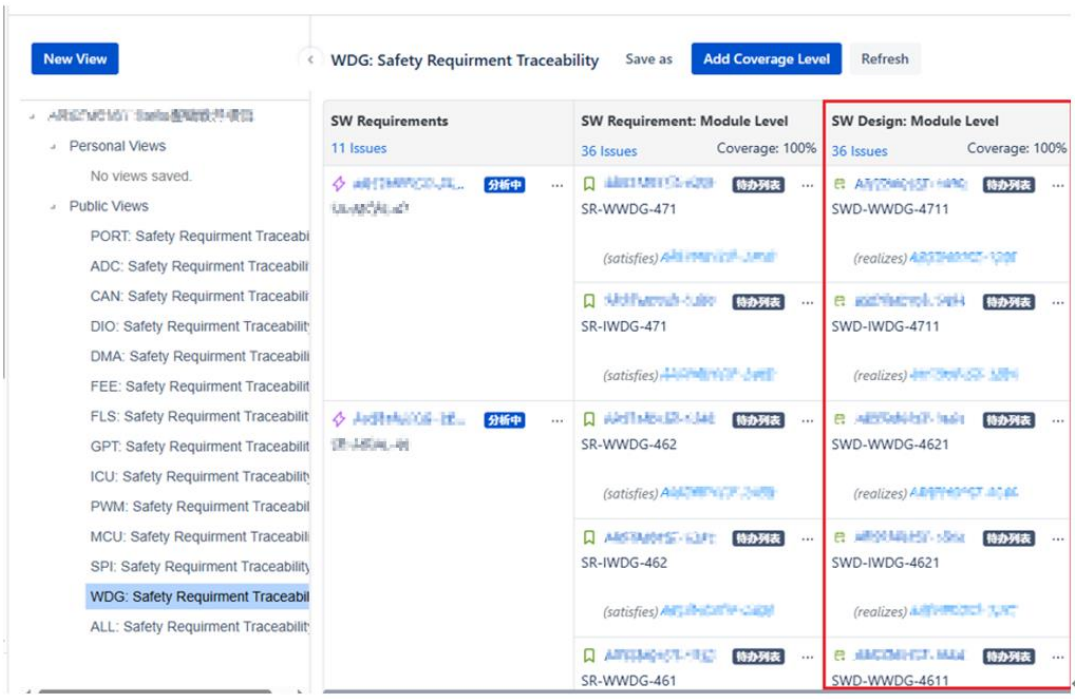
针对每个模块如 Wdg 进行分解，设计出模块间的交互接口，画出静态动态架构图，并在需求管理系统进行追溯。

Decompose each module, such as Wdg, design the interaction interfaces between modules, draw static and dynamic architecture diagrams, and trace them on system for traceability.



接口	编号	描述			
void Wdg_27_IWDG1_Init(const Wdg_27_IWDG1_ConfigType *Wdg_Cfg_Ptr)	SWAD-15-001	Wdg_27_IWDG1_Init 函数应初始化 Wdg 模块和看门狗硬件。 The Wdg_27_IWDG1_Init function shall initialize the Wdg module and the watchdog hardware.			
	I/O	名称	类型	范围	描述
参数		Wdg_Cfg_Ptr	Wdg_27_IWDG1_ConfigType	/	配置集指针 Pointer to configuration set
回归值		None	None	None	None
备注					

接口	编号	描述			
Std_ReturnType Wdg_27_IWDG1_SetMode(WdgIf_ModeType Mode)	SWAD-15-002	此函数用于切换看门狗的模式。 This function is used to set the mode of the watchdog.			
	I/O	名称	类型	范围	描述
参数		Mode	WdgIf_ModeType	WDGIF_OFF_MODE WDGIF_SLOW_MODE WDGIF_FAST_MODE	以下静态配置模式之一： - wdgif_slow_mode - wdgif_fast_mode One of the following statically configured modes:



3.4 SWE03 详细设计和编码 SWE03 Detailed Design and Coding

对每个软件模块进行细分，直至无法分解的软件单元，并对每个软件单元进行静态，动态设计，并在需求管理系统进行追溯。

Subdivide each software module until it reaches software units that cannot be further decomposed, perform static and Motion Design on each software unit, and trace them on system for traceability.

功能描述
Function Description

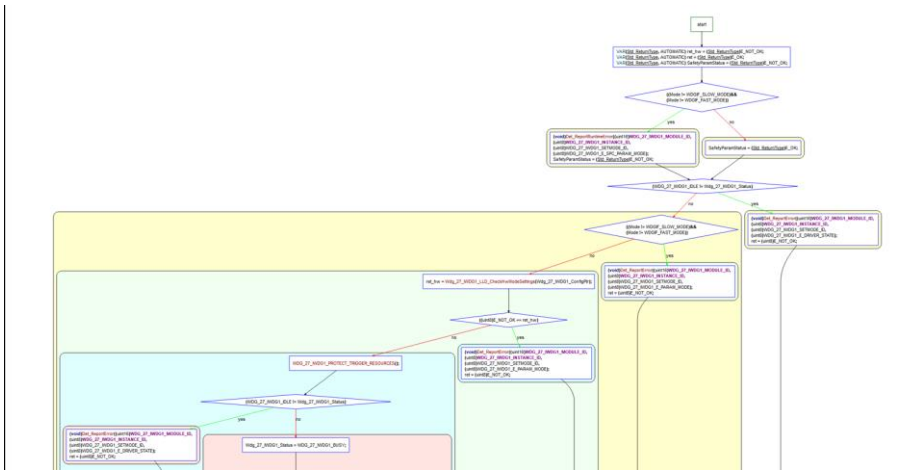
名称 Name	设计编号 Design ID	描述 Description	同步/异步 Sync/Async	备注 Remark
Std_ReturnType Wdg_27_IWDG1_SetMode(WdgIf_ModeType Mode)	SWE03Arlwdg00002	此函数用于切换看门狗的模式。 This function is used to set the mode of the watchdog.	同步Synchronous	/

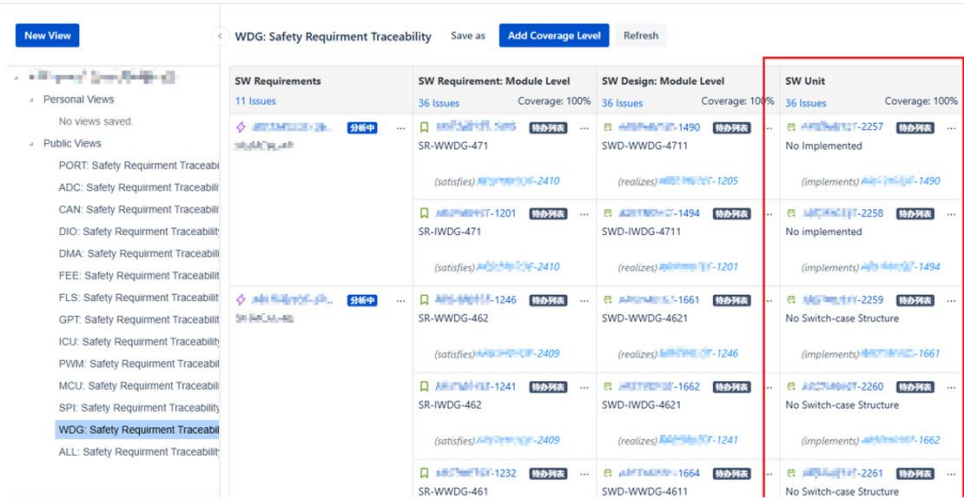
参数/配置数据
Parameters/Configuration Data

名称 Name	类型 Type	范围 Range	描述 Description
Mode	WdgIf_ModeType	WDGIF_OFF_MODE WDGIF_SLOW_MODE WDGIF_FAST_MODE	以下静态配置模式之一: - wdgif_slow_mode、 - wdgif_fast_mode。 One of the following statically configured modes: - WDGIF_SLOW_MODE, - WDGIF_FAST_MODE.

局部变量
Local Variable

名称 Name	类型 Type	范围 Range	描述 Description
ret_hw	Std_ReturnType	E_OKE_NOT_OK	下层接口的运行结果 Running results for low level interfaces
ret	Std_ReturnType	/	用于存储函数的执行结果状态 Used to store the execution result state of a function
SafetyParamStatus	Std_ReturnType	/	用于存储安全参数检查的结果状态 Used to store the result status of security parameter checks





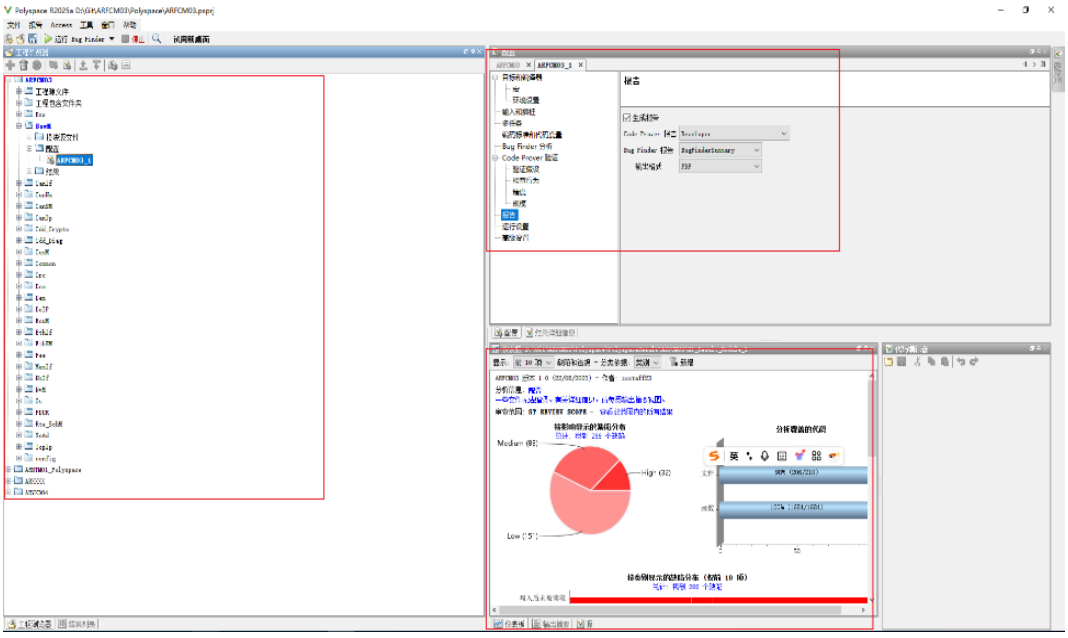
3.5 SWE04 单元测试 Unit Test

静态分析工具：Polyspace，用于代码质量检查和性能优化：包括总结报告，违反项详细记录及影响分析，修正建议等。

Static analysis tool: Polyspace, used for code quality inspection and performance optimization: including summary reports, detailed records of violations and impact analysis, correction suggestions, etc.

A	B	C	D	E	F	G	H	I
源代码指标		函数圈复杂度	函数代码行数	函数嵌套数量	函数中可能路径数的上限值	文件中的静态全局变量和函数的数量	注释中字符数量和代码中字符数量的比例	不通过原因
模块名	函数名	STCYC <=10	STLIN <= 200	STMIF <=5	STPTH <= 300	STSCT <= 30	STCDN <= 1.2	
Adc.c	Adc_CheckNormalConversionStatus	3	59	1	4	8	0.99	填写不通过原因
	Adc_CheckStartGroupConversionErrc	5	43	1	16			
	Adc_DisableGroupNotification	4	28	3	4			
	Adc_EnableGroupNotification	4	28	3	4			
	Adc_GetGroupStatus	3	29	2	3			
	Adc_GetStreamLastPointer	4	40	3	4			
	Adc_GetVersionInfo	2	18	1	2			
	Adc_Init	4	42	2	6			
	Adc_ProcessInjectedConversion	2	12	1	2			
	Adc_ProcessNormalConversion	2	15	1	2			
	Adc_ReadGroup	5	41	4	5			
	Adc_SetupResultBuffer	5	44	4	5			
	Adc_StartGroupConversion	7	54	1	48			
	Adc_StopGroupConversion	6	54	4	7			

误差列表					
MISRA-C Rules	违反的规则号	代码行数	违规代码	规则规定	不修正的原因
MISRA-C:2012 Rule-2.2 There shall be no dead code	2981	689	VAR(uint8, AUTOMATIC) runtimeError = 0U;	This initialization is redundant. The value of this object is never used before being modified.	此函数需要宏定义
MISRA-C:2012 Rule-20.7 Expressions resulting from	3432	162	P2VAR(Adc_ValueGroupType, AUTOMATIC, ADC_APPL_DATA) Adc_ResultsBufferPtr[ADC_GROUPS_MAX_NB];	Simple macro argument expression is not parenthesized.	读写寄存器
MISRA-C:2012 Dir-4.4 Sections of code should not	2053	998	#endif	This block comment appears to comment out source code.	MemMap
MISRA-C:2012 Rule-20.7 Expressions resulting from the expansion of macro	3432	212	CONSTP2VAR(Adc_UnitStatusType, AUTOMATIC, ADC_APPL_CONST) Adc_UnitsStatus[ADC_HW_UNITS_NB] =	Simple macro argument expression is not parenthesized.	强制类型转换
MISRA-C:2012 Dir-4.4 Sections of code should not be "commented out"	2053	1129	#endif	This block comment appears to comment out source code.	读写寄存器
MISRA-C:2012 Rule-2.1 A project shall not contain	1503	843	FUNC(Std_ReturnType, ADC_CODE) Adc_ReadGroup(VAR(Adc_GroupType, AUTOMATIC) Group, P2VAR(Adc_ValueGroupType, AUTOMATIC,	The function '%1s' is defined but is not used within this project.	强制类型转换
MISRA-C:2012 Rule-20.7 Expressions resulting from	3432	843	FUNC(Std_ReturnType, ADC_CODE) Adc_ReadGroup(VAR(Adc_GroupType, AUTOMATIC) Group, P2VAR(Adc_ValueGroupType, AUTOMATIC,	Simple macro argument expression is not parenthesized.	寄存器赋值



动态测试 Tessy：通过需求分析，错误猜测，等价类，边界值设计生成测试用例。

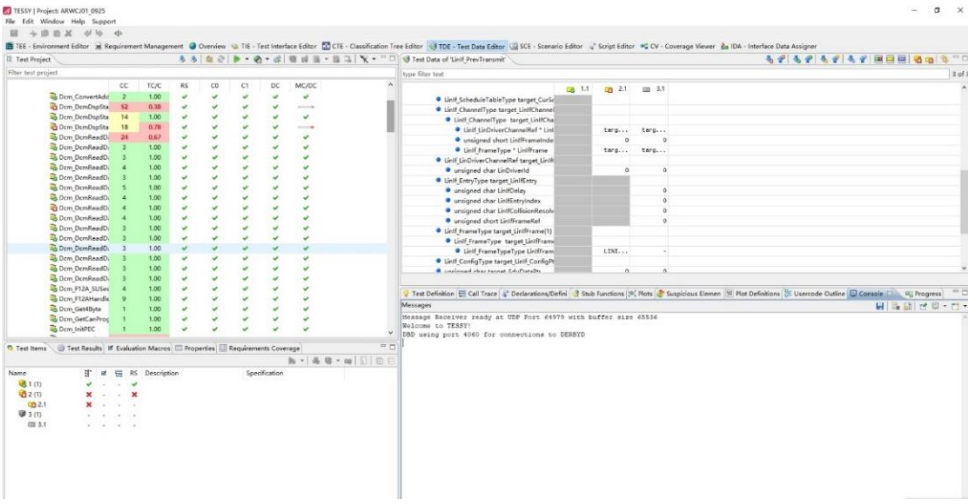
Dynamic testing with Tessy: Generate test cases through requirement analysis, error guessing, equivalence class, and boundary value design.

判定覆盖（Decision Coverage）

条件覆盖 (Conditon Coverage)

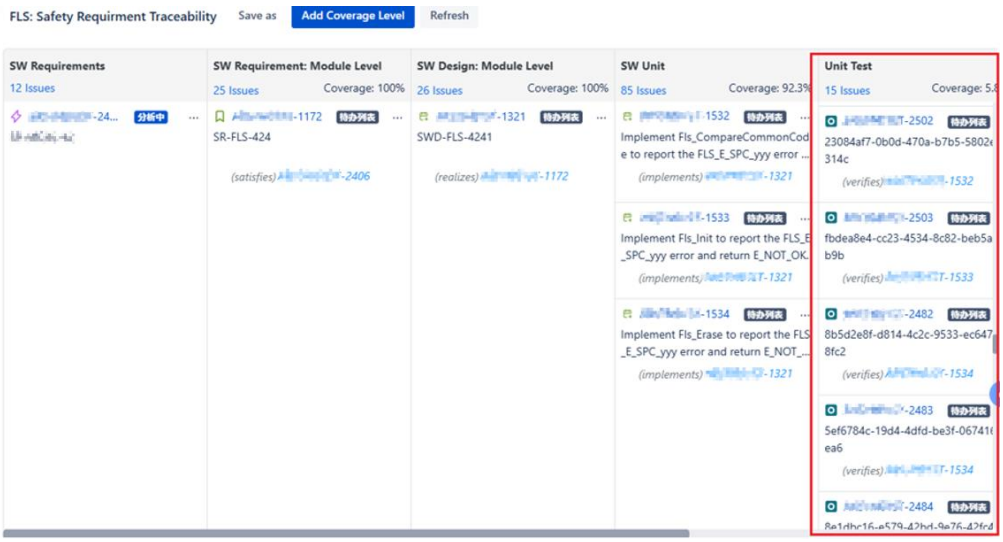
修正判定条件覆盖（MC/DC）

覆盖度均达到 100% Coverage all reached 100%



单元测试追溯同样使用需求管理系统进行追溯：

Unit test traceability also uses system for tracing:



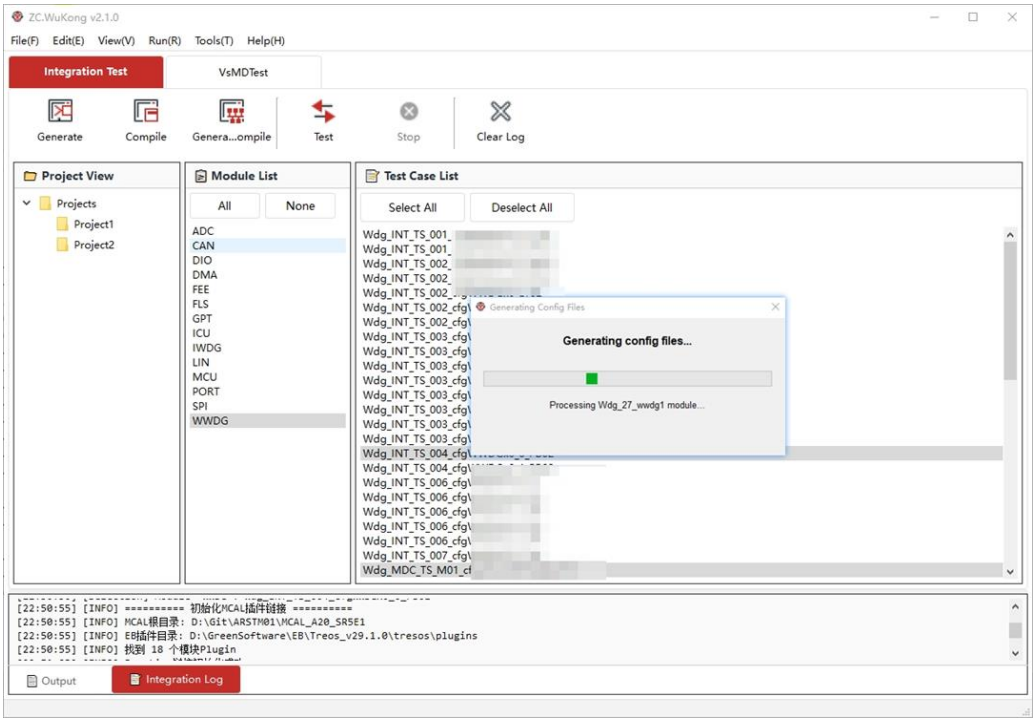
3.6 SWE05 软件集成和集成测试 Software Integration and Test

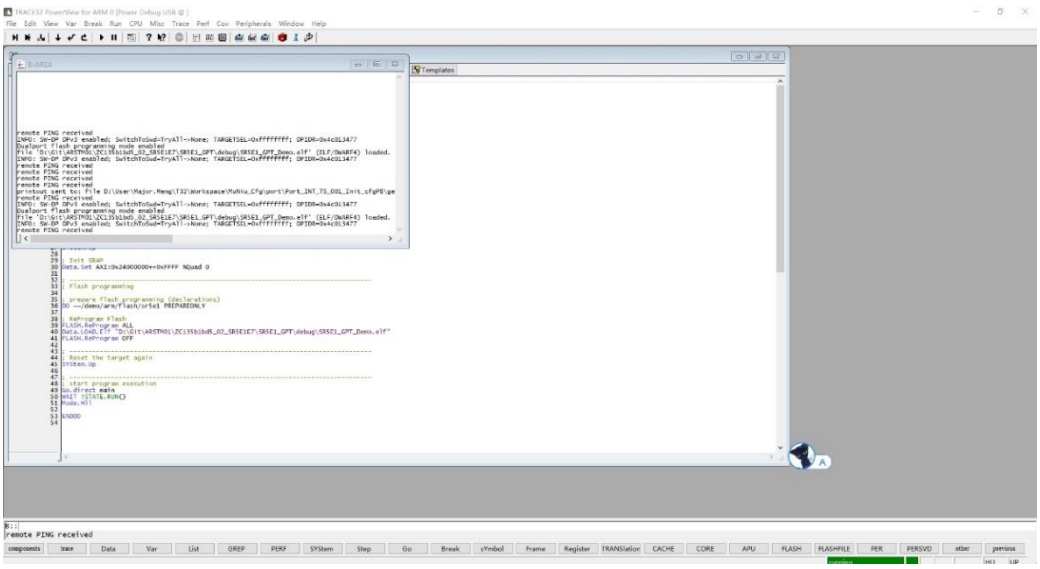
MCAL 全模块集成测试：知从悟空测试工具，包含以下内容

- Python：完整的自动化测试框架
- MuNiu / EB Tresos： 设计多组配置及测试用例
- Cygwin+Hightec/Tasking：实现自动化编译
- Trace32：实现自动烧录，执行测试及读取测试结果

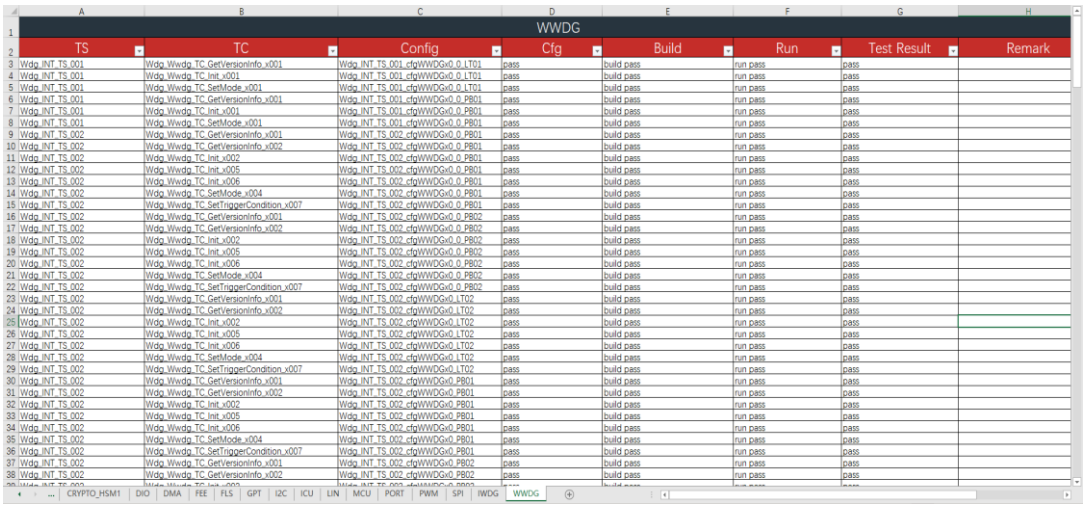
MCAL Full Module Integration Test: Known from WuKong Testing Tool, including the following content

- Python: A Complete Automated Testing Framework
- MuNiu / EB Tresos: Design multiple sets of configurations and test cases
- Cygwin+Hightec/Tasking: Implement automated compilation
- Trace32: Implement automatic programming, execute tests, and read test results





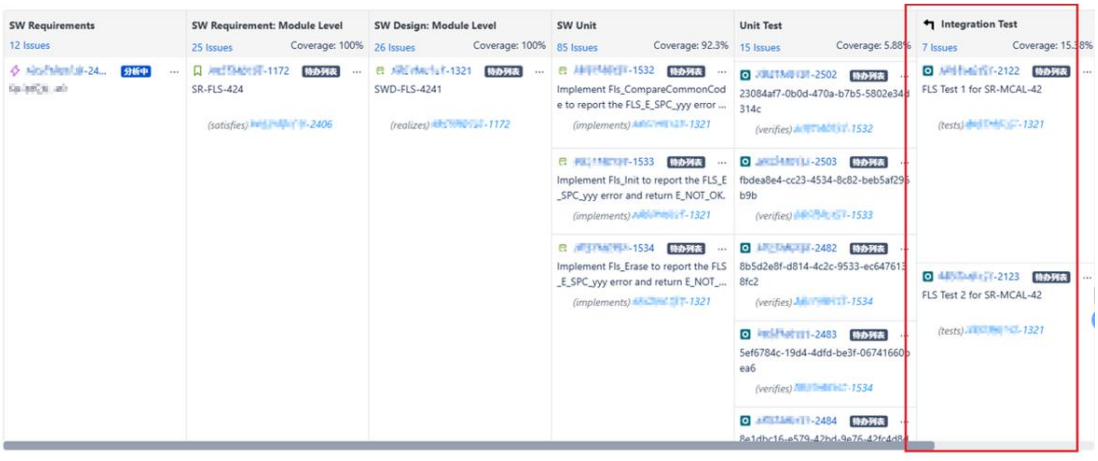
```
########################################
Clean Done!
########################################
2025-09-01 15:58:28 CST - DEBUG - Run command: make CFG_DIR="TestCase/test_wdg_wwdg" MAK_DIR="TestCase/test_w
dg_wwdg/make/Wdg_INT_TS_001.mak" GENERATE_DIR="EBtresos_Cfg/wdg_27_wwdg1/Wdg_INT_TS_001_cfgWWDGx0_0_LT01/gene
rate/epc" EXTRA_C_FLAGS="-DEXTERNAL_TIMER_SERVICE -DM4_WDG_MIP=Wdg_27_WWDG1 -DM4_WDG_BUILD_MODE=VARIANT-LINK-
TIME -DM4_WDG_WWDG_INSTANCE=WWDG1 -DM4_WDG_WWDG_PRESCALER=DIV_128 -DM4_WDG_FAST_TIMEOUT=0.04 -DM4_WDG_FAST_WI
NDOW_TIMEOUT=0.02 -DM4_WDG_SLOW_TIMEOUT=0.05 -DM4_WDG_SLOW_WINDOW_TIMEOUT=0.025 -DM4_WDG_INTERNAL_TRIGGER_CHA
NNEL=TIM2_CH0 -DM4_WDG_DEFAULT_MODE=WDGIF_SLOW_MODE -DM4_WDG_DEV_ERROR_DETECT=0 -DM4_WDG_DISABLE_DEM_REPORT=1
 -DM4_WDG_SLOW_WN_MODE=0 -DM4_WDG_WWDG_IWDG_TEST=0" SILENT="0" MODULE="Wdg_27_WWDG1" CURRENT_CFG_SET="Wdg_INT
_TS_001_cfgWWDGx0_0_LT01" -j8
2025-09-01 15:58:28 CST - INFO - Attempt 1/1 for Wdg_27_WWDG1 - Wdg_INT_TS_001_cfgWWDGx0_0_LT01
TestCase/test_wdg_wwdg/make/wdg_wwdg_common.mk:28: ##### MIP:                  Wdg_27_WWDG1
TestCase/test_wdg_wwdg/make/wdg_wwdg_common.mk:29: ##### VENDOR_API_INFIX:      WWDG1
TestCase/test_wdg_wwdg/make/wdg_wwdg_common.mk:60: ##### TEST_COMMON_CFG:
TestCase/test_wdg_wwdg/make/wdg_wwdg_common.mk:61: ##### DERIVATIVE_FOLDER:    SR5E1CM7
TestCase/test_wdg_wwdg/make/wdg_wwdg_common.mk:62: ##### ALL:    SR5E1CM7
TestCase/test_wdg_wwdg/make/Wdg_INT_TS_001.mak:408: ../MCAL_A20_SR5E1_0.4.00_Unsigned_With_noM4/Rte_TS_T40D68
```



WWDG							
TS	TC	Config	Cfg	Build	Run	Test Result	Remark
Wdg_INT_TS_001	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_001_cfgWWDGx0_0_LT01	pass	build pass	run pass	pass	
Wdg_INT_TS_001	Wdg_Wwdg_TC_Init_x001	Wdg_INT_TS_001_cfgWWDGx0_0_LT01	pass	build pass	run pass	pass	
Wdg_INT_TS_001	Wdg_Wwdg_TC_SetMode_x001	Wdg_INT_TS_001_cfgWWDGx0_0_LT01	pass	build pass	run pass	pass	
Wdg_INT_TS_001	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_001_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_001	Wdg_Wwdg_TC_Init_x001	Wdg_INT_TS_001_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_001	Wdg_Wwdg_TC_SetMode_x001	Wdg_INT_TS_001_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x002	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x002	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x005	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x006	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetMode_x004	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetTriggerCondition_x007	Wdg_INT_TS_002_cfgWWDGx0_0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x002	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x002	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x005	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x006	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetMode_x004	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetTriggerCondition_x007	Wdg_INT_TS_002_cfgWWDGx0_0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x002	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x002	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x005	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x006	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetMode_x004	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetTriggerCondition_x007	Wdg_INT_TS_002_cfgWWDGx0_LT02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x002	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x002	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x005	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_Init_x006	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetMode_x004	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_SetTriggerCondition_x007	Wdg_INT_TS_002_cfgWWDGx0_PB01	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x001	Wdg_INT_TS_002_cfgWWDGx0_PB02	pass	build pass	run pass	pass	
Wdg_INT_TS_002	Wdg_Wwdg_TC_GetVersionInfo_x002	Wdg_INT_TS_002_cfgWWDGx0_PB02	pass	build pass	run pass	pass	

集成测试追溯同样使用需求管理系统进行追溯:

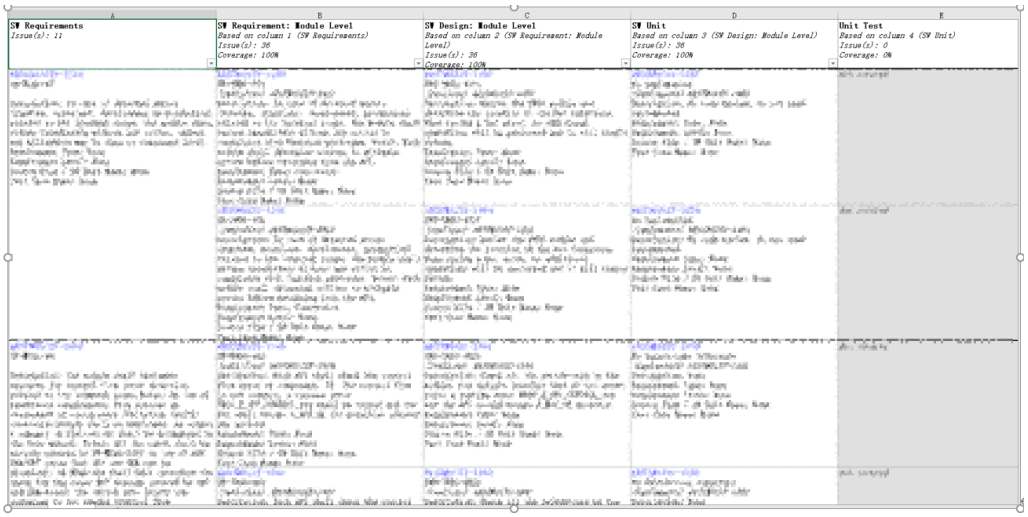
Integration test traceability also uses system for traceability:



3.7 ASPICE 追溯矩阵 traceability matrix:

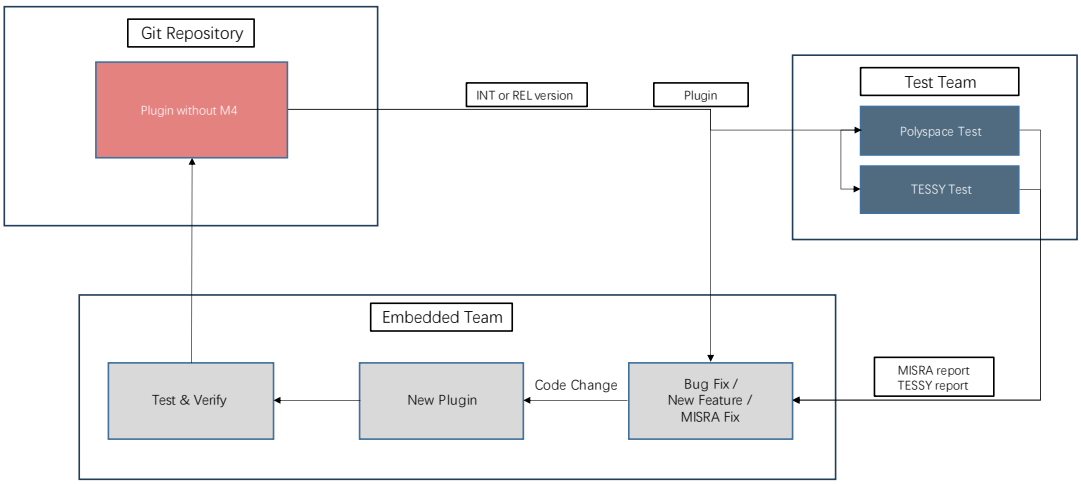
最后针对完整的开发及测试导出追溯矩阵

Finally, export the traceability matrix for the complete development and testing:

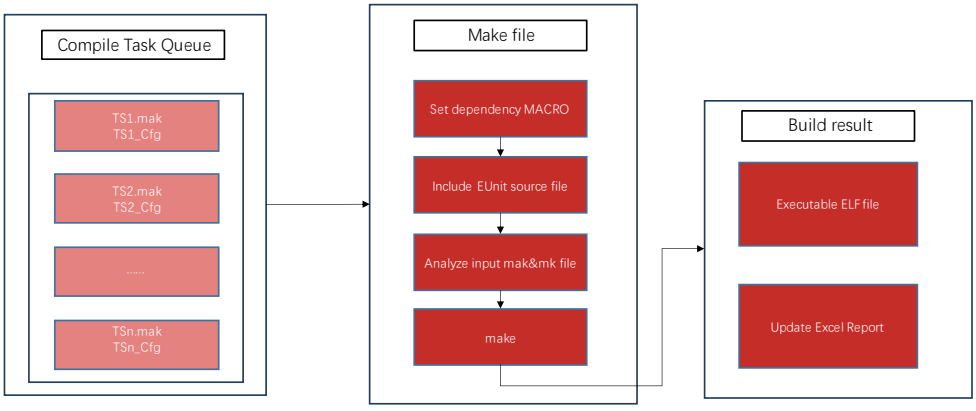


3.8 开发及测试自动化流程 Development and Testing Automatic Flow

SWE03&04 Develop process



SWE05 Compile Procedure



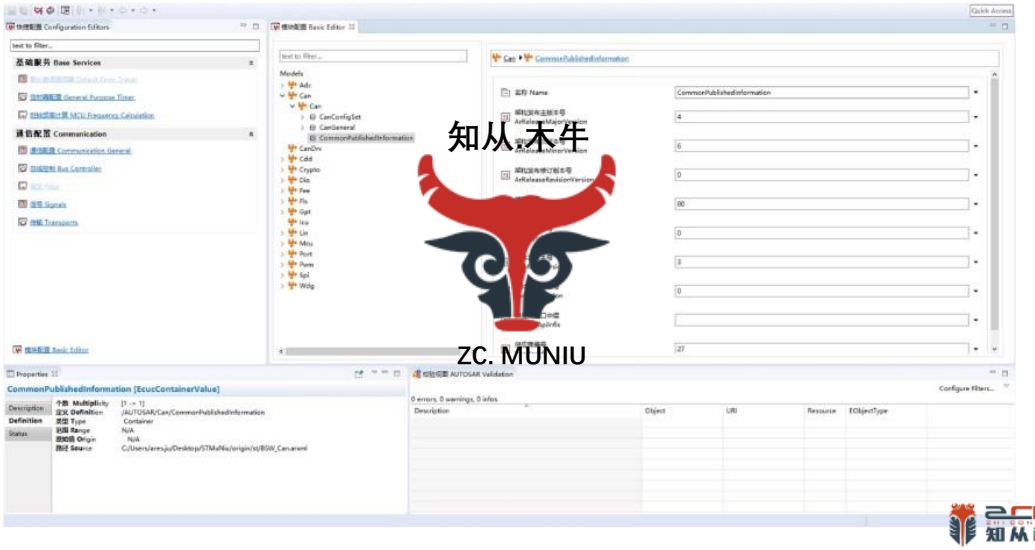
4 交付物 RELEASE PRODUCTS

交付物 RELEASE PRODUCTS	
ProjectDevelopmentSchedule	SafetyManual
WeeklyProjectReport	SoftwareIntegrationTestReport
MilestoneReport	SoftwareResourceAnalysisReport
ListOfFunctionalSafetyOutput	SoftwareTestReport
SafetyCase	QuestionAndAnswerLog
SafetyPlan	Test Specification IT
SafetyLibrarySoftwareSafetyAnalysisReport	Code Coverage IT
SoftwareDependentFailureAnalysisReport	Traceability Matrix
WarningReport	Test Specification UT
Code	SW Unit Design
StaticAnalysisReport	Safety Concept
DynamicAnalysisReport	SRD approved

5 工具链介绍 TOOL CHAIN

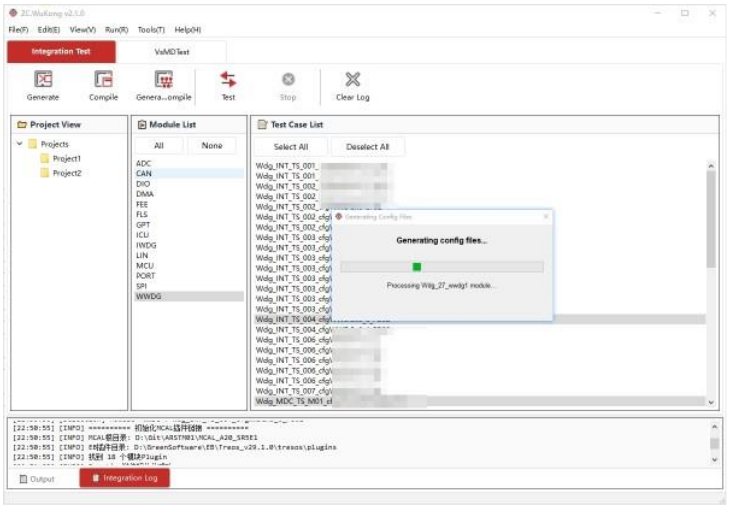
5.1 知从木牛配置工具 ZC.MuNiu Configuration Tool

配置工具Configuration Tool：木牛 MUNIU



5.2 知从悟空 MCAL 测试工具 ZC.WuKong MCAL Test Tool

ZC 悟空MCAL测试工具



知从悟空测试工具可以通过 EB_Tresos、知从木牛等MCAL配置工具对MCAL进行不同配置及测试用例的设计。

并通过Makefile文件和Cygwin环境对测试代码和MCAL工程进行自动编译，并可以通过劳特巴赫等调试器进行自动烧录，自动运行测试代码及测试。