IMPLEMENTATION AND VALIDATION FOR A CONTINUOUS TESTING METHOD IN AUTOMOTIVE SOFTWARE DEVELOPMENT

Yu Zhong

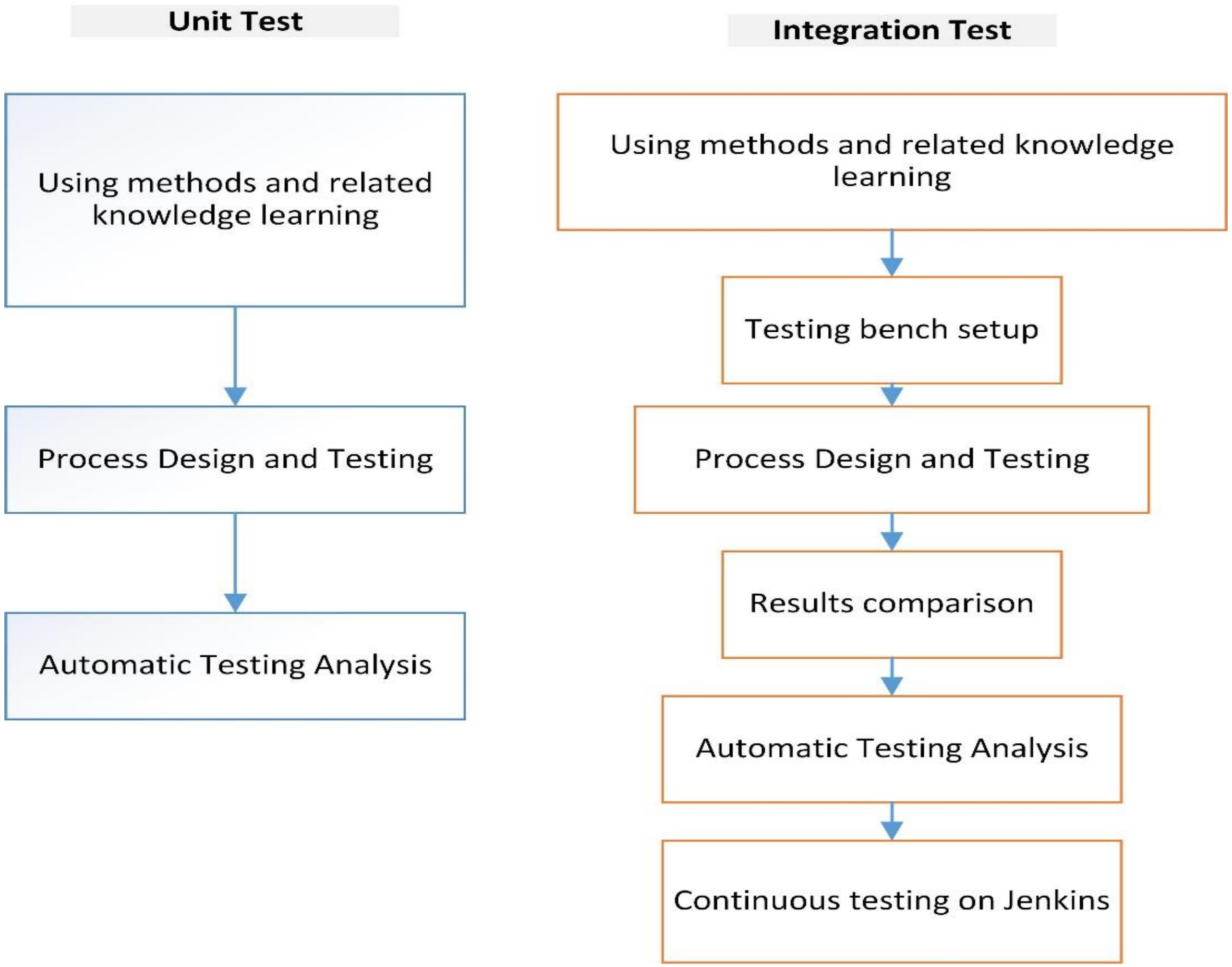
Company Representative: Rupp Christian Robert Bosch GmbH CC-DA/EAU1

Examiner: Prof. Dr. Peter Raab Coburg University of Applied Sciences

BOSCH

Introduction

- Continuous testing
- Unit testing
- Integration testing

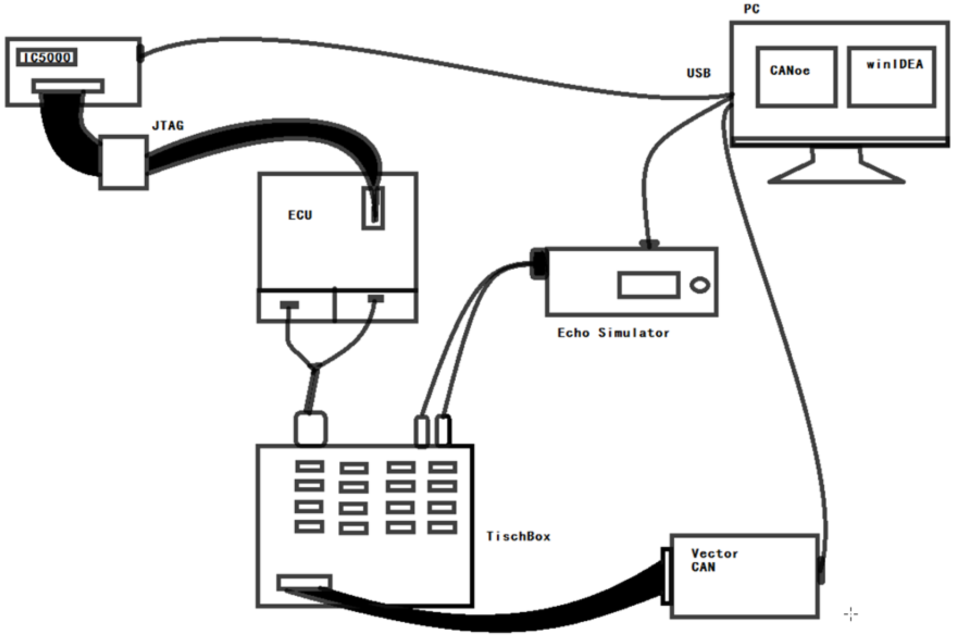

Unit Test
Using methods and related knowledge learning
Process Design and Testing
Automatic Testing Analysis
Integration Test
Using methods and related knowledge learning
Testing bench setup
Process Design and Testing
Results comparison
Automatic Testing Analysis
Continuous testing on Jenkins

Goals

- Continuous test can be executed at any time, even some months or years after they were designed.
- Effort and resources for manually integration testing are reduced and optimized, but at the same time software quality is increased.
- Any developer or tester can build integration tests via Jenkins with only one click.
- Testing and developing process is established in conformance with international standards and norms.
- Any developer can create Jenkins projects with this method according to their own test requirements.
- Unit test and integration test strategy are defined and implemented.
- There are defined concepts for unit test, integration tests and continuous testing which provide guidance on how to execute and evaluate tests and the deployment of continuous testing.

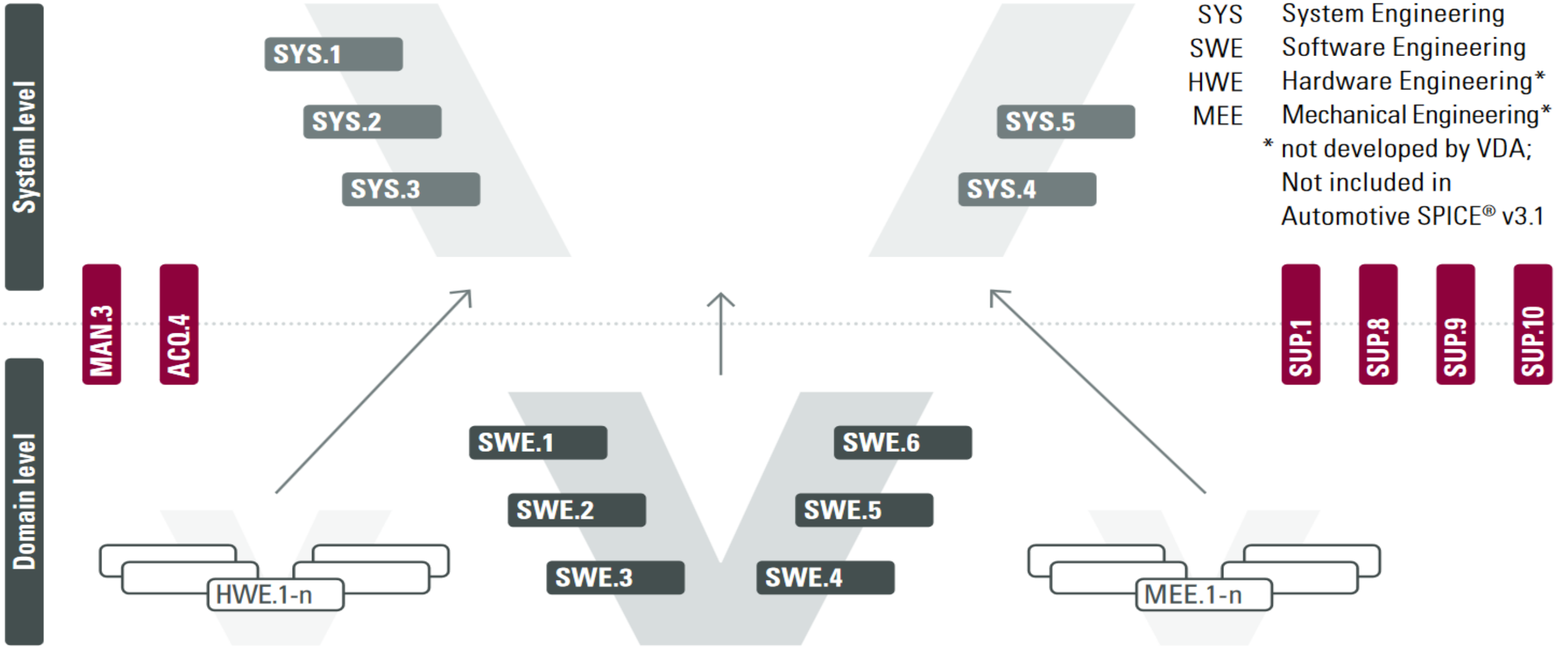
HIL Test Bench

- The test bench are composed of the following components:
- iC5000, JTAG, park pilot ECU, Tischbox, Echo simulator, Vector interface.



Automotive SPICE and ISO 26262

- Automotive Software Process Improvement and Capability dEtermination.

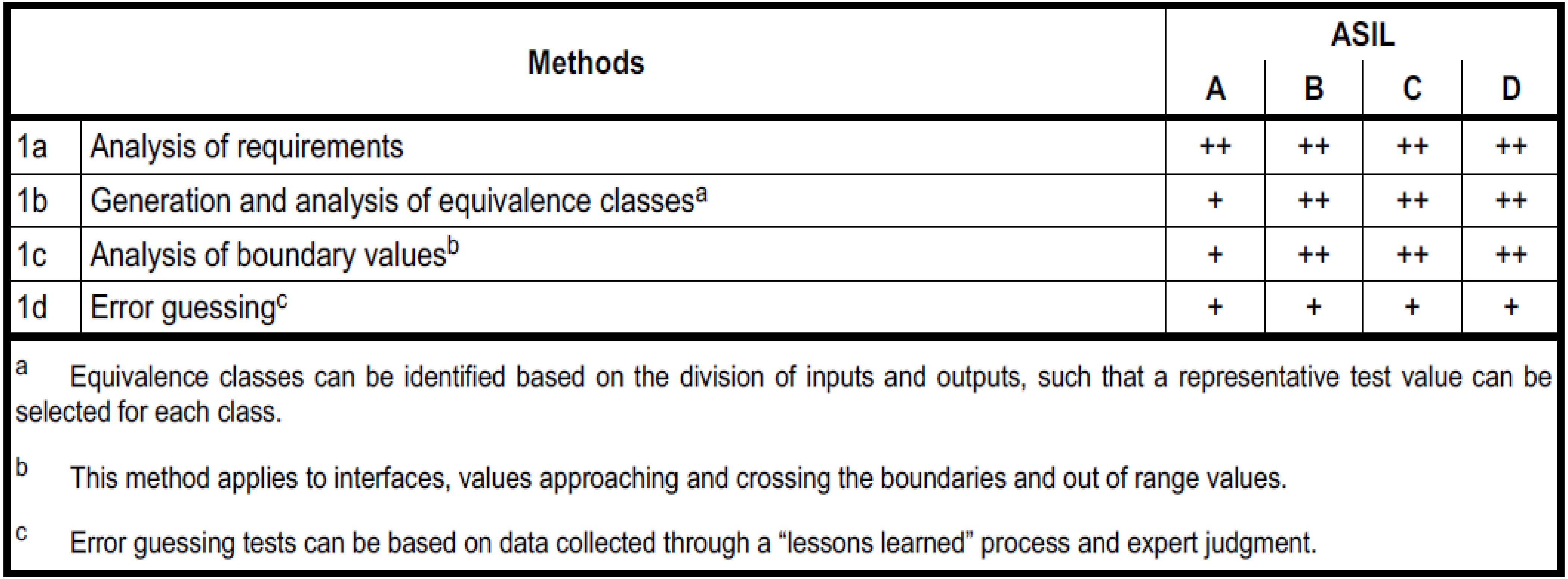


SYS	System Engineering
SWE	Software Engineering
HWE	Hardware Engineering*
MEE	Mechanical Engineering*

* not developed by VDA; Not included in Automotive SPICE® v3.1

Automotive SPICE and ISO 26262

- Automotive Software Process Improvement and Capability dEtermination.



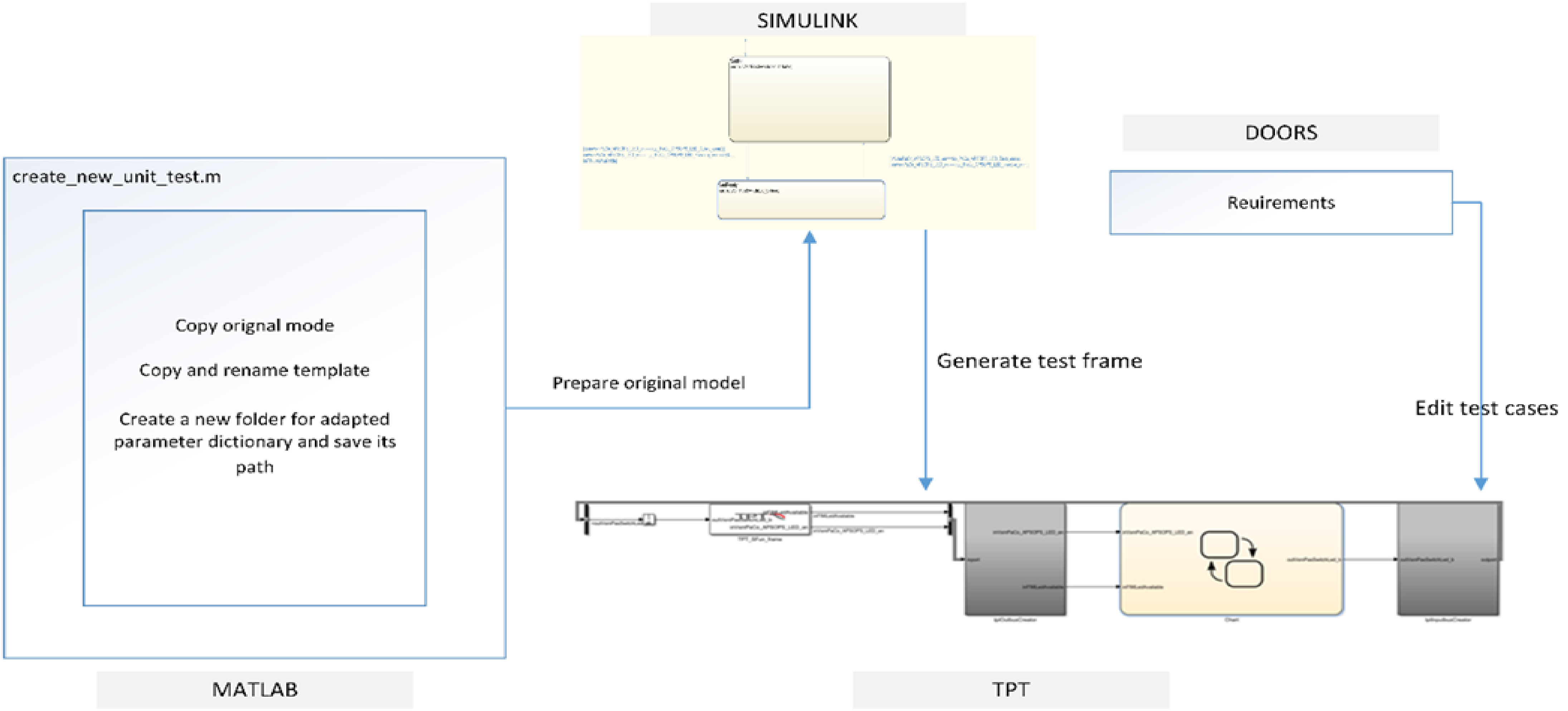
Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Generation and analysis of equivalence classes[a]	+	++	++	++
1c	Analysis of boundary values[b]	+	++	++	++
1d	Error guessing[c]	+	+	+	+

[a] Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.

[b] This method applies to interfaces, values approaching and crossing the boundaries and out of range values.

[c] Error guessing tests can be based on data collected through a "lessons learned" process and expert judgment.

Unit test



SIMULINK

DOORS

Reuirements

create_new_unit_test.m

Copy orignal mode

Copy and rename template

Create a new folder for adapted parameter dictionary and save its path

Prepare original model

Generate test frame

Edit test cases

MATLAB

TPT

Work test flow of unit test

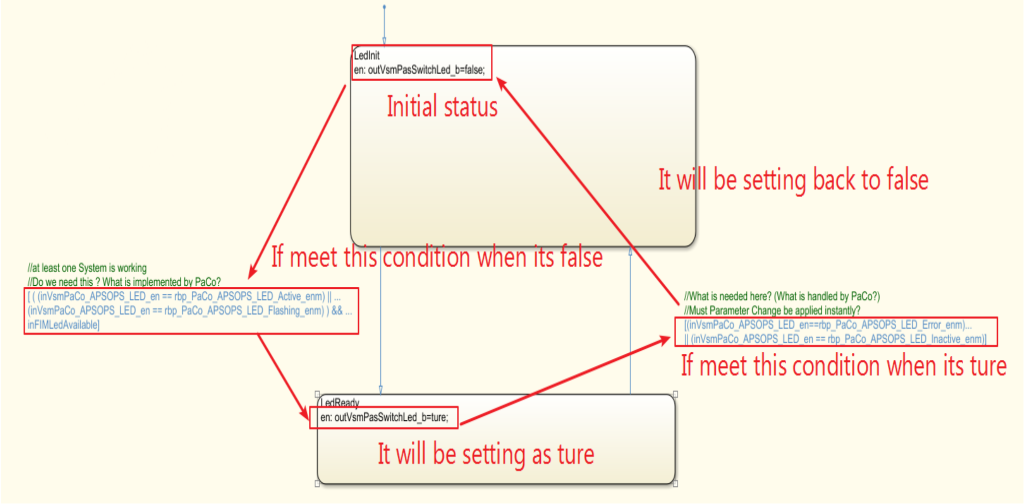
Unit test

- Unit testing is a level of software testing where individual units of a software are tested.

- Advantages:
  - Makes the Process Agile
  - Improves the quality of the code
  - Finds Software Bugs Early
  - Facilitates Changes and Simplifies Integration
  - Developers can execute regression tests
  - Reduce Costs
- Disadvantages:
  - Not every bug can be catch during unit
  - Losing amount of time for testing complex cases
  - Testing have to finished even design is not clear
  - Do not show the absence of errors
  - Sometimes test cases and scenarios even more complicated than model itself

Unit test

- ▶ Unit testing is a level of software testing where individual units of a software are tested.



```
LedInit
en: outVsmPasSwitchLed_b=false;
```

Initial status

It will be setting back to false

If meet this condition when its false

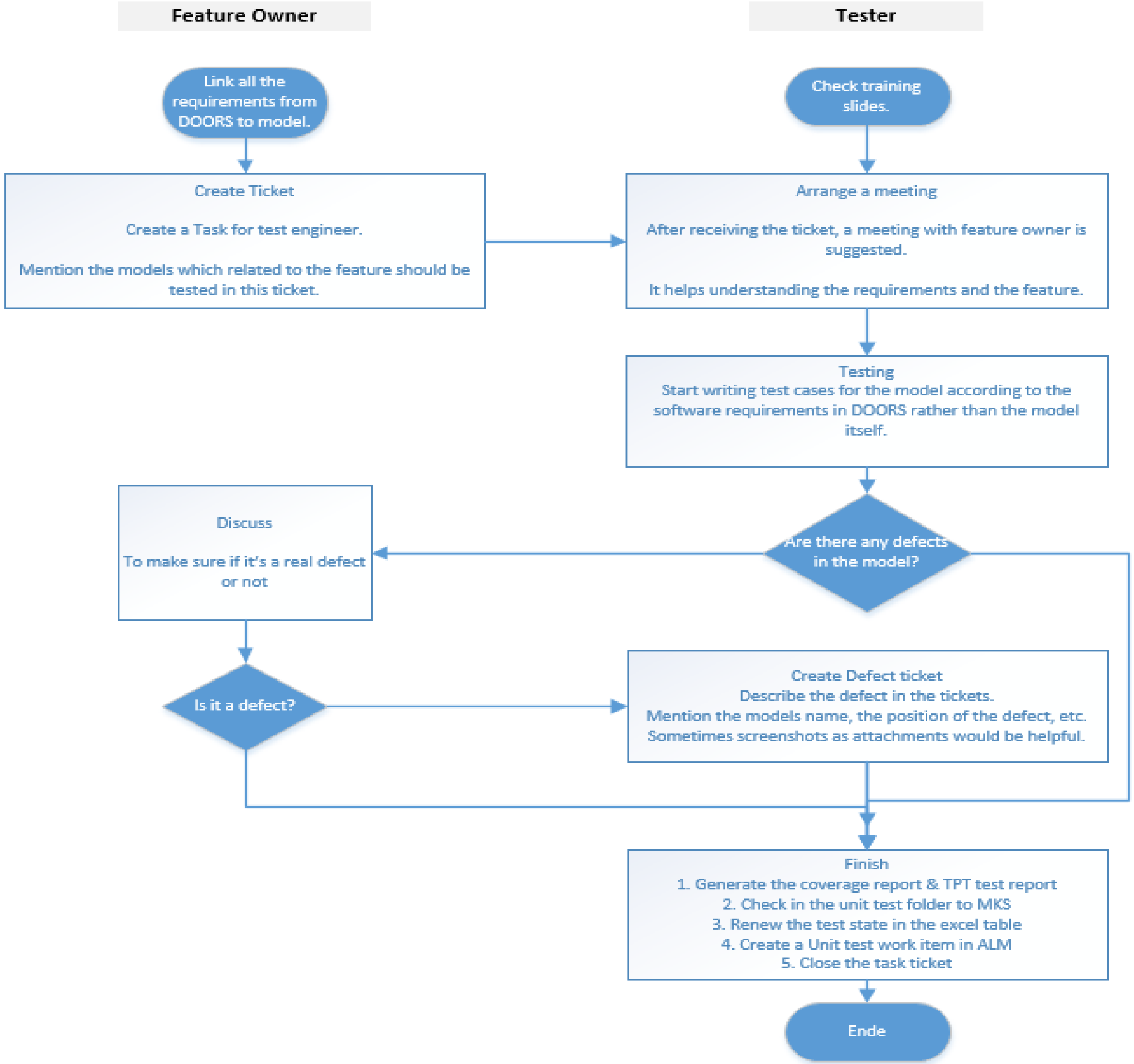
```
//at least one System is working
//Do we need this ? What is implemented by PaCo?
[ ( (inVsmPaCo_APSOPS_LED_en == rbp_PaCo_APSOPS_LED_Active_enm) || ...
(inVsmPaCo_APSOPS_LED_en == rbp_PaCo_APSOPS_LED_Flashing_enm) ) && ...
inFIMLedAvailable]
```

```
//What is needed here? (What is handled by PaCo?)
//Must Parameter Change be applied instantly?
[(inVsmPaCo_APSOPS_LED_en==rbp_PaCo_APSOPS_LED_Error_enm)...
|| (inVsmPaCo_APSOPS_LED_en == rbp_PaCo_APSOPS_LED_Inactive_enm)]
```

If meet this condition when its ture

```
LedReady
en: outVsmPasSwitchLed_b=ture;
```

It will be setting as ture



Feature Owner

Tester

Link all the requirements from DOORS to model.

Create Ticket

Create a Task for test engineer.

Mention the models which related to the feature should be tested in this ticket.

Check training slides.

Arrange a meeting

After receiving the ticket, a meeting with feature owner is suggested.

It helps understanding the requirements and the feature.

Testing
Start writing test cases for the model according to the software requirements in DOORS rather than the model itself.

Are there any defects in the model?

Discuss

To make sure if it's a real defect or not

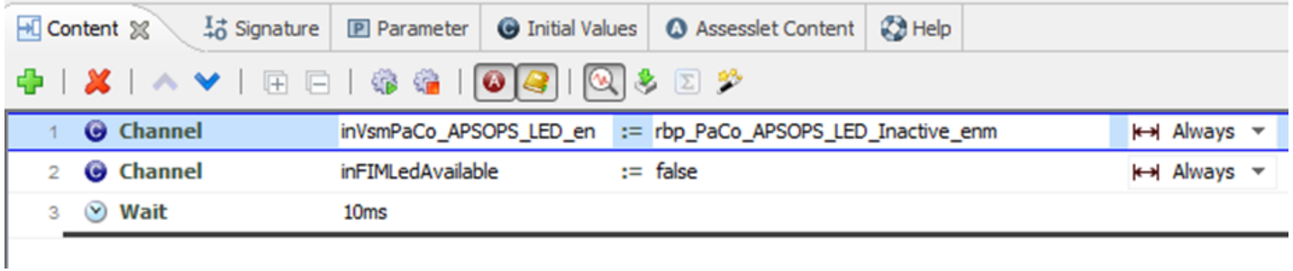
Is it a defect?

Create Defect ticket
Describe the defect in the tickets.
Mention the models name, the position of the defect, etc.
Sometimes screenshots as attachments would be helpful.

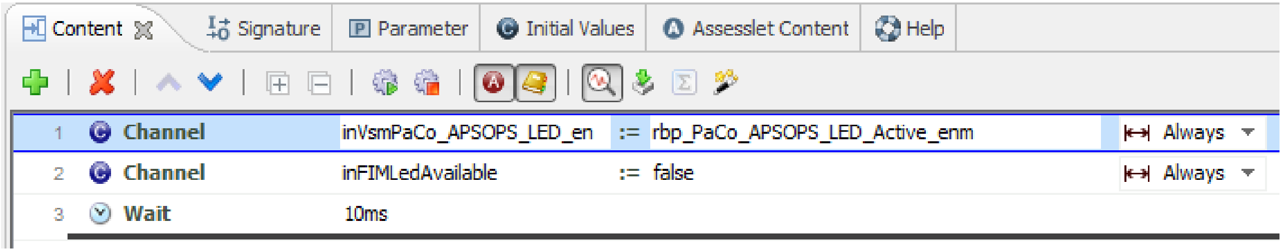
Finish
1. Generate the coverage report & TPT test report
2. Check in the unit test folder to MKS
3. Renew the test state in the excel table
4. Create a Unit test work item in ALM
5. Close the task ticket

Ende

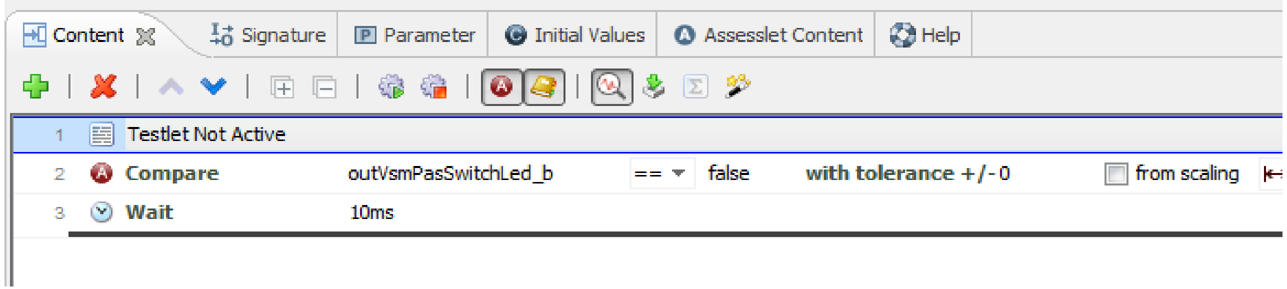
Unit test



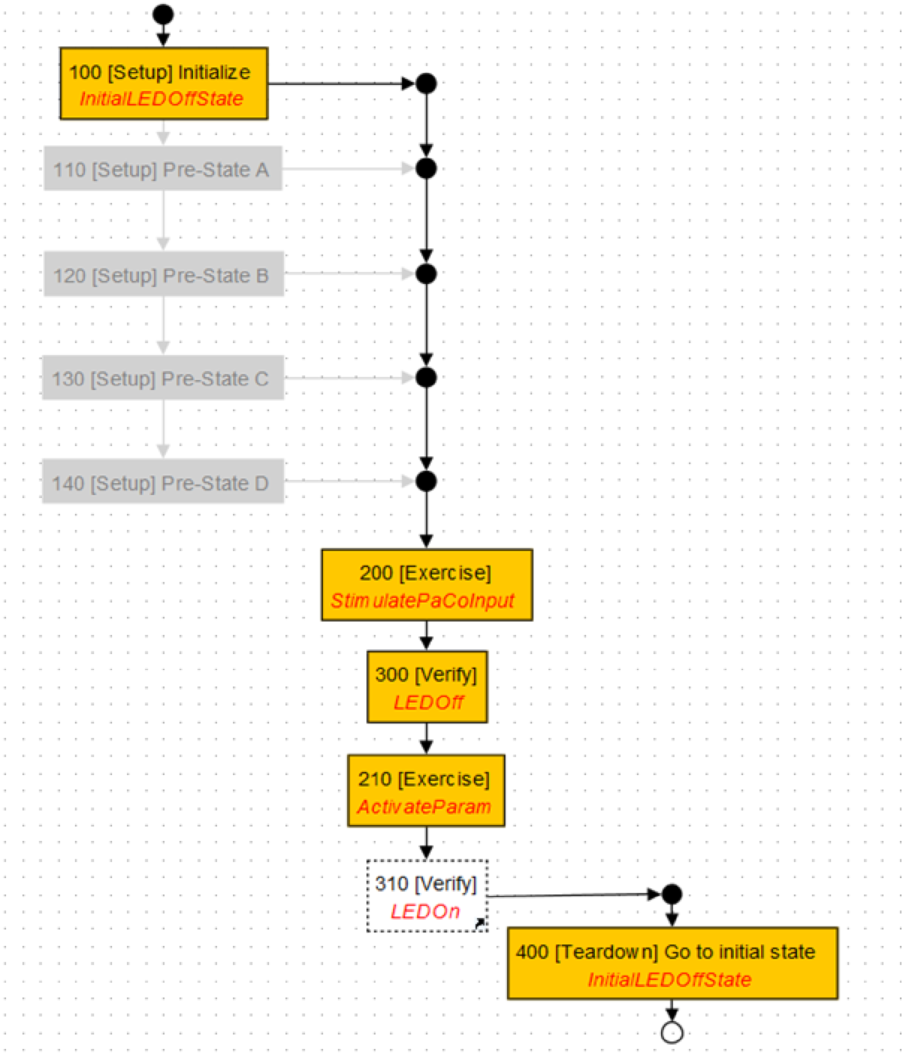
1	Channel	inVsmPaCo_APSOPS_LED_en	:=	rbp_PaCo_APSOPS_LED_Inactive_enm	Always
2	Channel	inFIMLedAvailable	:=	false	Always
3	Wait	10ms			



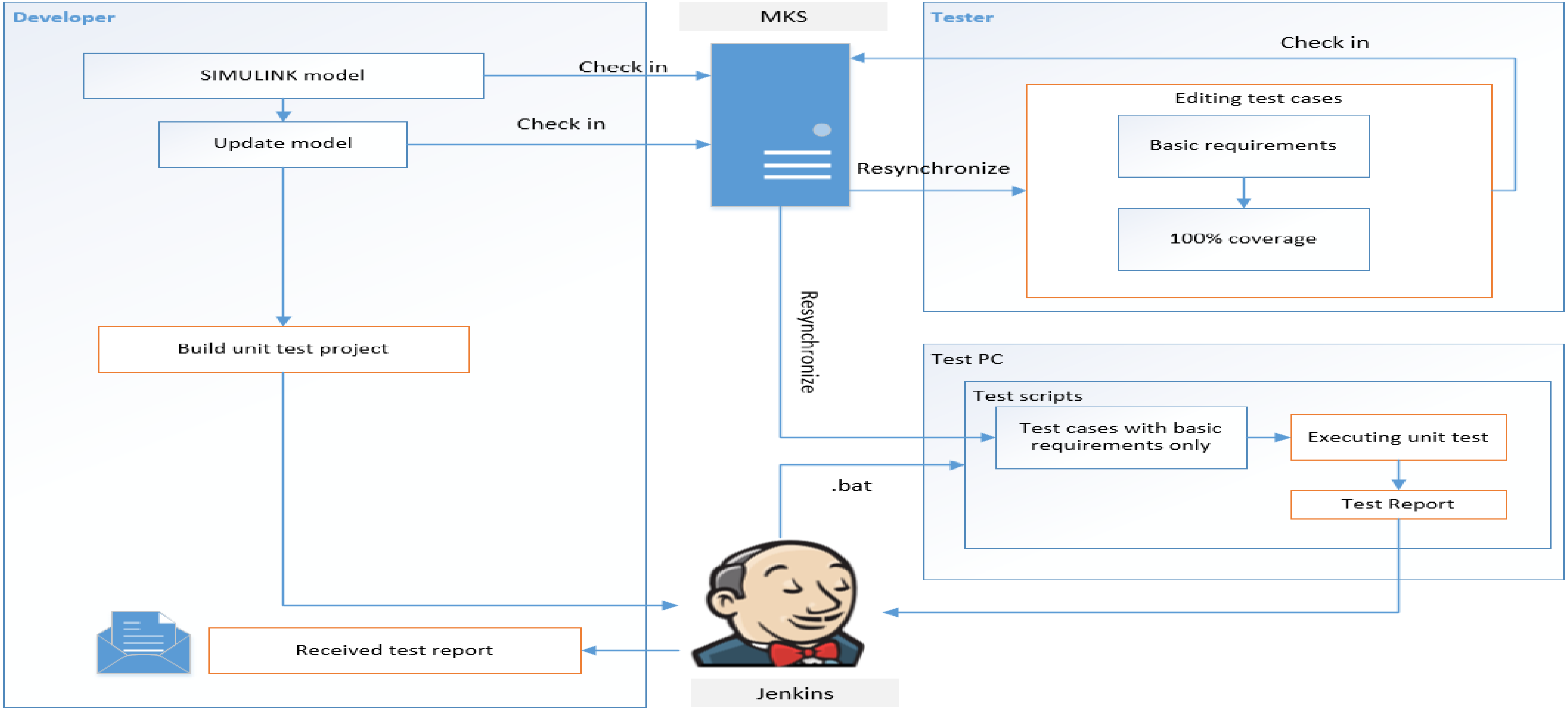
1	Channel	inVsmPaCo_APSOPS_LED_en	:=	rbp_PaCo_APSOPS_LED_Active_enm	Always
2	Channel	inFIMLedAvailable	:=	false	Always
3	Wait	10ms			



1	Testlet Not Active					
2	Compare	outVsmPasSwitchLed_b	==	false	with tolerance +/- 0	□ from scaling
3	Wait	10ms				

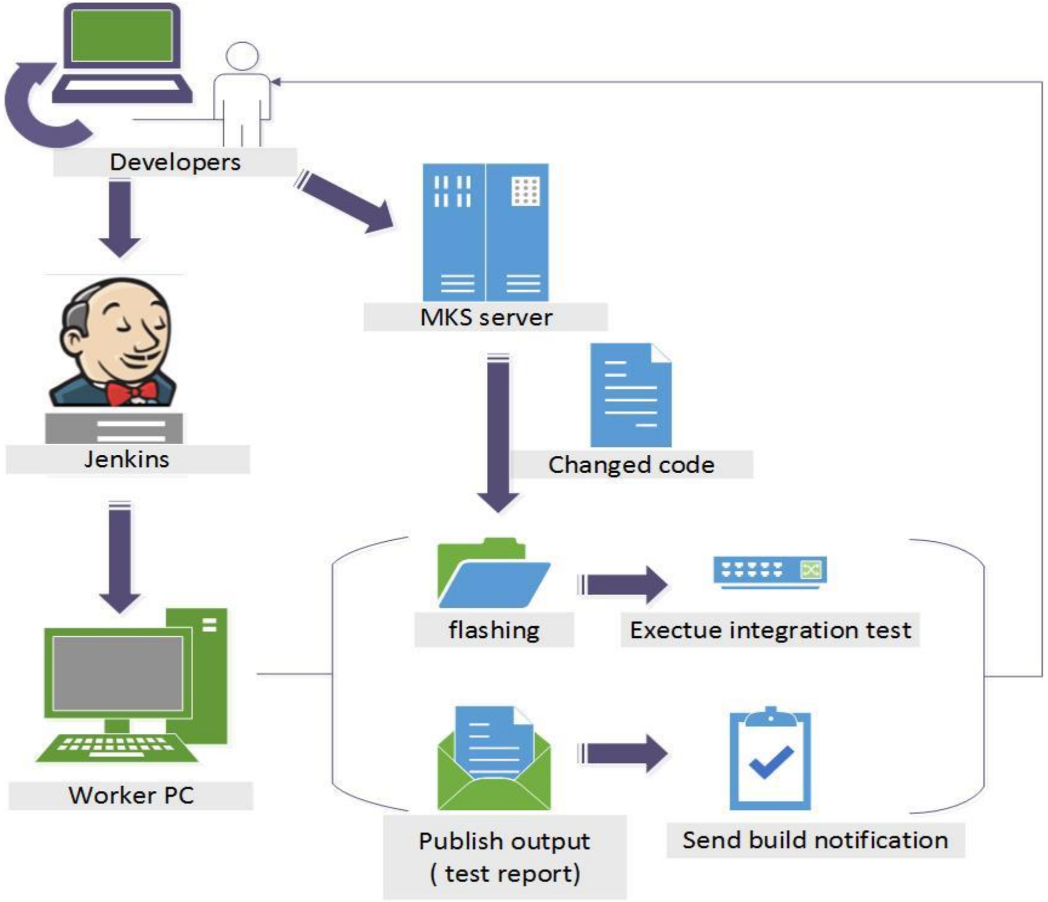


Unit test


Developer
SIMULINK model
Update model
Build unit test project
Received test report
Check in
Check in
MKS
Resynchronize
Resynchronize
Tester
Check in
Editing test cases
Basic requirements
100% coverage
Test PC
Test scripts
Test cases with basic requirements only
Executing unit test
Test Report
.bat
Jenkins

Integration test

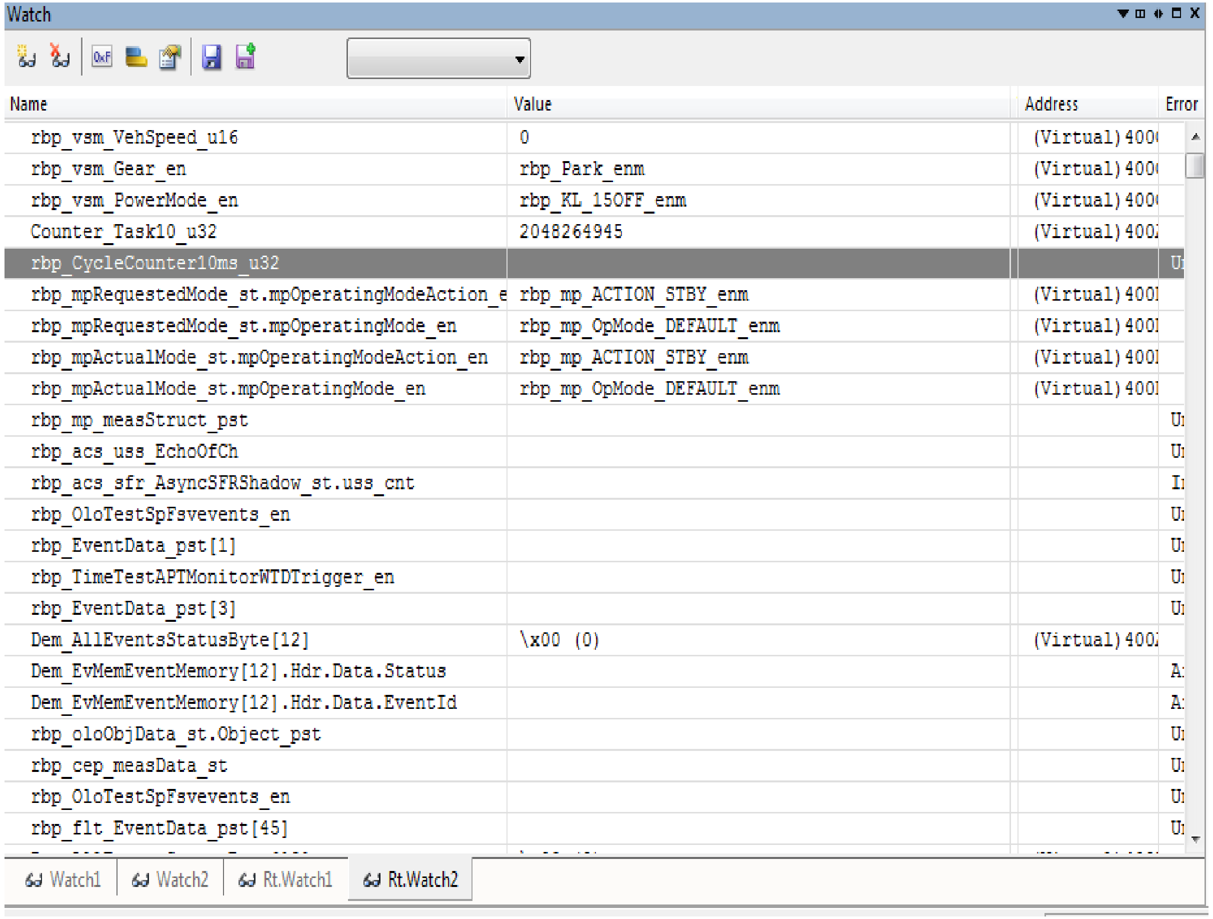
- Integration testing is the phase in software testing in which individual software modules are combined and tested as a group.



Integration test

- **ECU Flashing**
  - Flashing through Merge file.
  - Flashing through Individual files.
  - Flashing through ODIS.

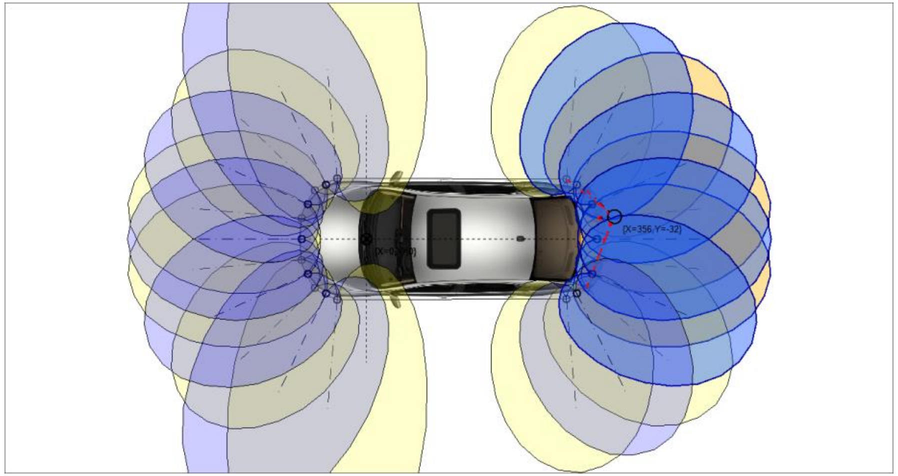
- 5 individual files:
- *P.hex*, *E.hex*, *C.hex*, *R.hex* and *B.hex*.
- P.hex is application hex.
- C.hex is calibration of parameters.
- E.hex is eeprom.
- B.hex is boot loader
- R.hex is boot manager

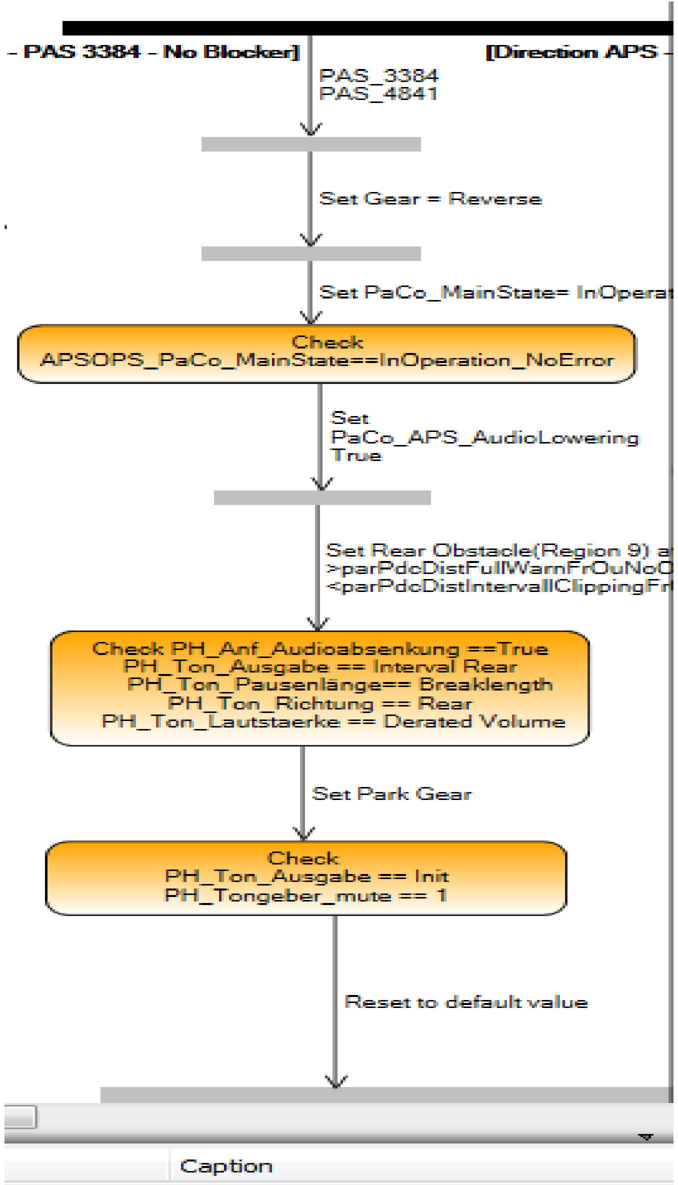


Name	Value	Address	Error
rbp_vsm_VehSpeed_u16	0	(Virtual)400	
rbp_vsm_Gear_en	rbp_Park_enm	(Virtual)400	
rbp_vsm_PowerMode_en	rbp_KL_15OFF_enm	(Virtual)400	
Counter_Task10_u32	2048264945	(Virtual)400	
rbp_CycleCounter10ms_u32			U
rbp_mpRequestedMode_st.mpOperatingModeAction_e	rbp_mp_ACTION_STBY_enm	(Virtual)400	
rbp_mpRequestedMode_st.mpOperatingMode_en	rbp_mp_OpMode_DEFAULT_enm	(Virtual)400	
rbp_mpActualMode_st.mpOperatingModeAction_en	rbp_mp_ACTION_STBY_enm	(Virtual)400	
rbp_mpActualMode_st.mpOperatingMode_en	rbp_mp_OpMode_DEFAULT_enm	(Virtual)400	
rbp_mp_measStruct_pst			U
rbp_acs_uss_EchoOfCh			U
rbp_acs_sfr_AsyncSFRShadow_st.uss_cnt			I
rbp_OloTestSpFsvevents_en			U
rbp_EventData_pst[1]			U
rbp_TimeTestAPTMonitorWTDTrigger_en			U
rbp_EventData_pst[3]			U
Dem_AllEventsStatusByte[12]	\x00 (0)	(Virtual)400	
Dem_EvMemEventMemory[12].Hdr.Data.Status			A
Dem_EvMemEventMemory[12].Hdr.Data.EventId			A
rbp_oloObjData_st.Object_pst			U
rbp_cep_measData_st			U
rbp_OloTestSpFsvevents_en			U
rbp_flt_EventData_pst[45]			U

Integration test

- **Manual Setting in CANoe**
- We can set and modify the CAN signals with CANoe restbus simulation (at most time statically)which means that we have to set these signals manually , by clicking and write signals values .





- PAS 3384 - No Blocker]

[Direction APS -

PAS_3384
PAS_4841

Set Gear = Reverse

Set PaCo_MainState= InOperat

Check
APSOPS_PaCo_MainState==InOperation_NoError

Set
PaCo_APS_AudioLowering
True

Set Rear Obstacle(Region 9) a
>parPdcDistFullWarnFrOuNoO
<parPdcDistIntervallClippingFr

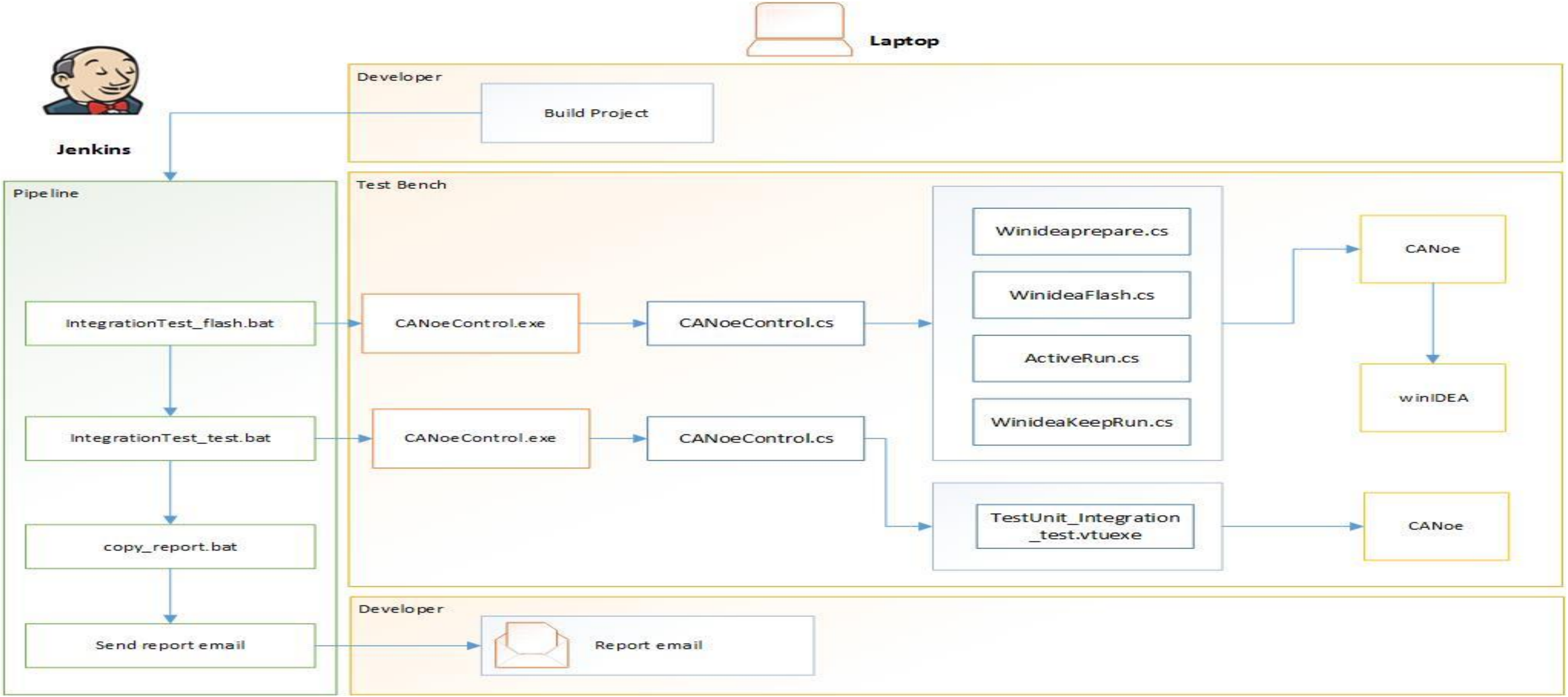
Check PH_Anf_Audioabsenkung ==True
PH_Ton_Ausgabe == Interval Rear
PH_Ton_Pausenlänge== Breaklength
PH_Ton_Richtung == Rear
PH_Ton_Lautstaerke == Derated Volume

Set Park Gear

Check
PH_Ton_Ausgabe == Init
PH_Tongeber_mute == 1

Reset to default value

Continuous testing



THANK YOU

Yu Zhong

Leonberg, Germany. 26.02.2019

BOSCH