



# [ 2018 에스피아이디 컨퍼런스 ] A-SPIICE를 충족하는 AUTOSAR SW 개발

2018. 09. 13

[주]에스피아이디

spid



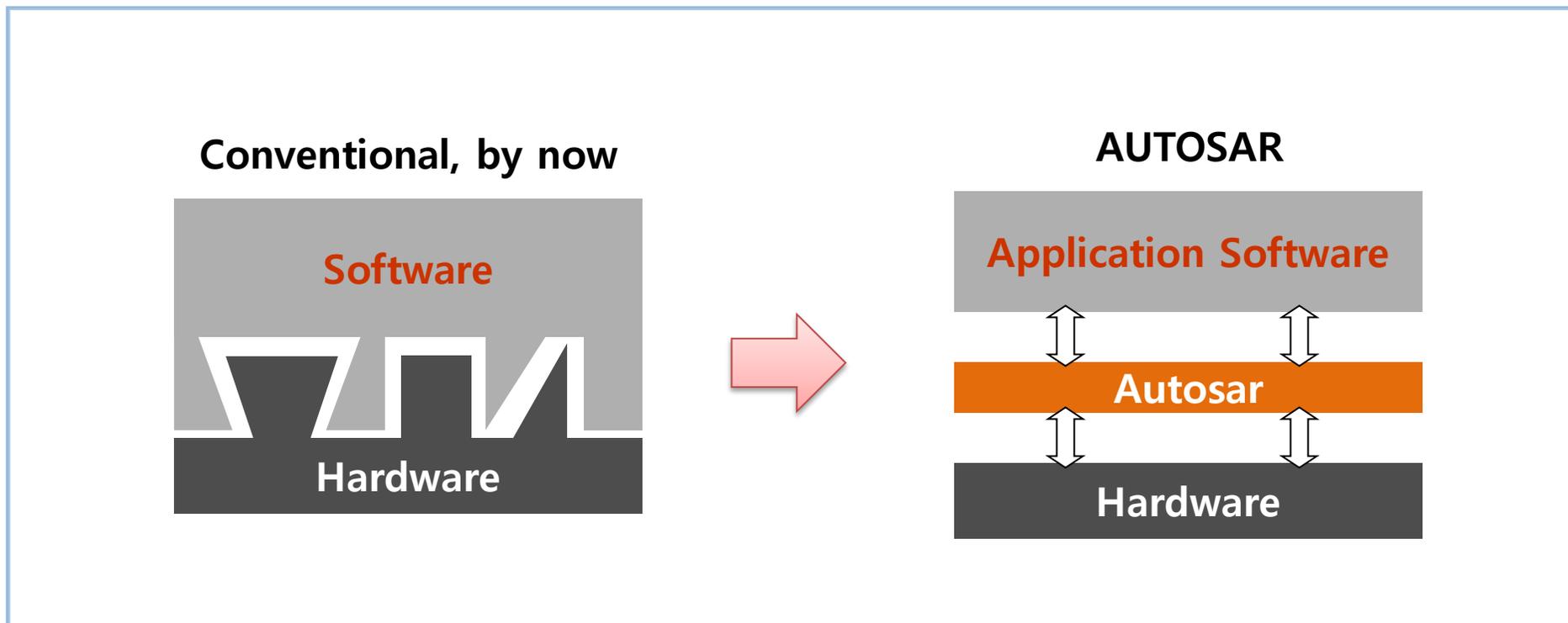
CMMI Institute Partner  
Powered by Carnegie Mellon

# 1. AUTOSAR 개요 및 배경

## ✓ AUTOSAR (AUTomotive Open System ARchitecture)

- Application SW에서 HW 종속적인 코드를 배제하여 재사용성을 극대화
- 표준 인터페이스 활용으로 ECU 간 호환성 극대화
- Supplier는 ECU의 핵심 기능 개발에 집중 가능

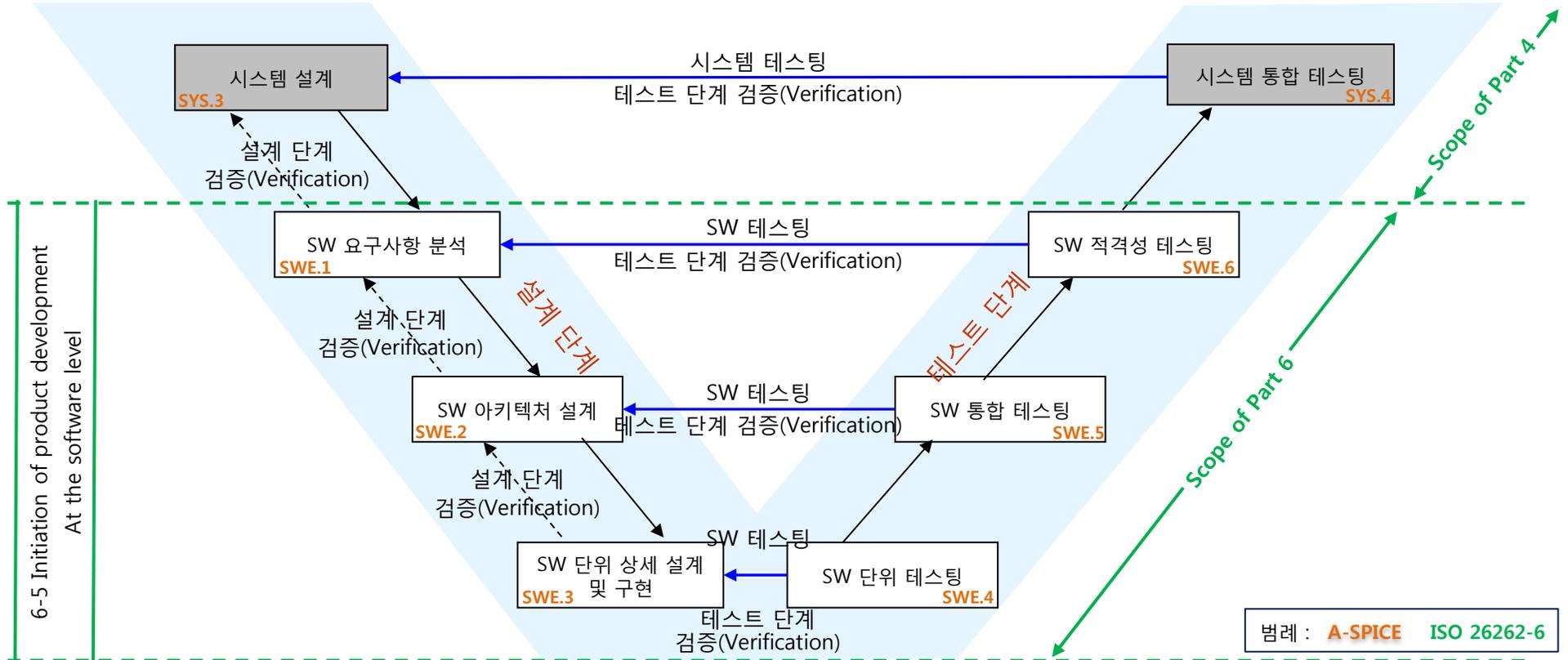
## ✓ ISO 26262-Part6 요구사항을 충족할 수 있도록 진화



# 1. AUTOSAR 개요 및 배경

## ✓ A-SPICE 및 ISO 26262 표준에서 요구되는 Top-Down Approach

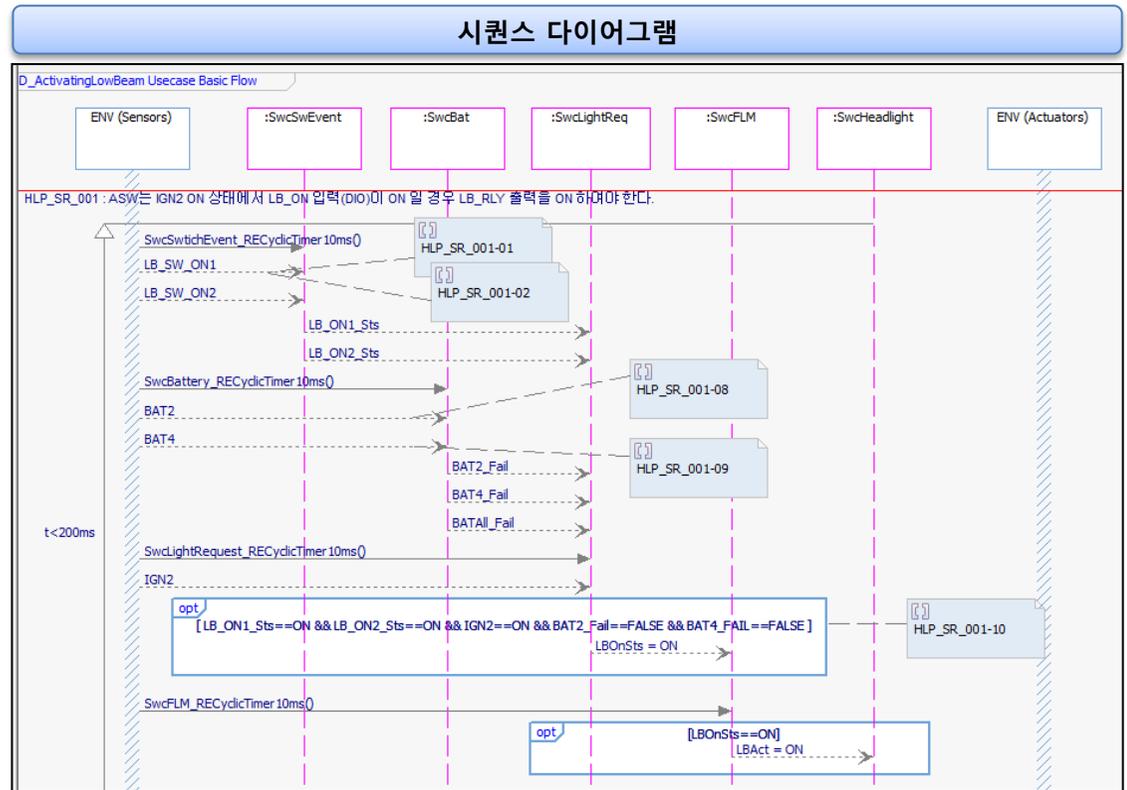
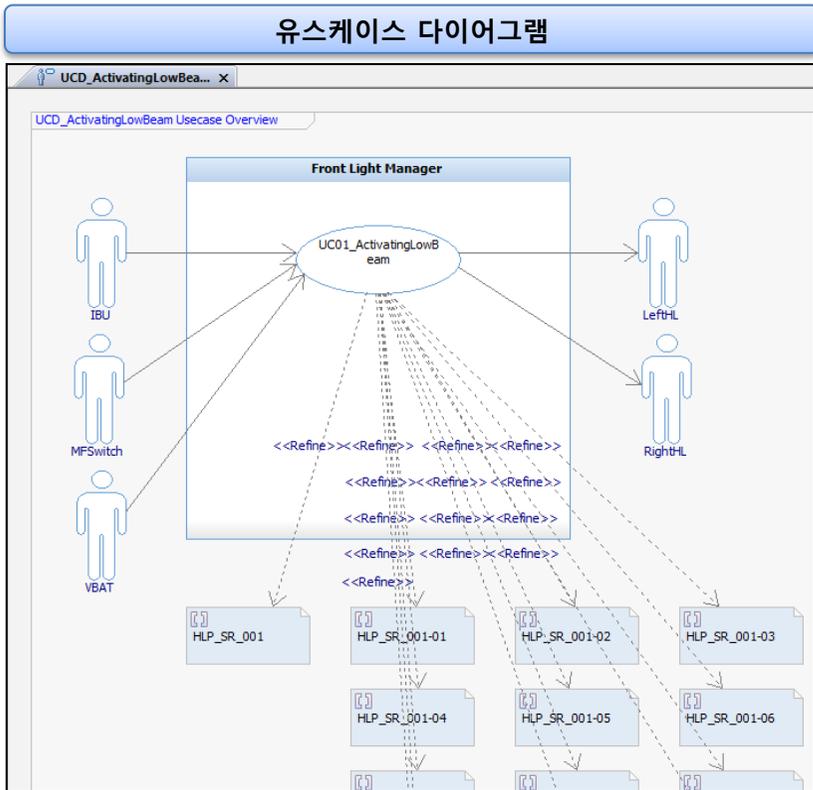
- SRS를 충족하는 SW 아키텍처를 설계하고, 해당 아키텍처에 대한 설계검증(안전분석)을 요구함
- SW 산출물 간 추적성 확보 및 검증(Verification)을 요구함



# 1. AUTOSAR 개요 및 배경

## ✓ AUTOSAR 표준 및 도구에서 필요한 사항 : ① 동적구조 설계 능력

- 요구사항 분석 및 동적구조 분석/설계 능력
- 유스케이스 다이어그램, 시퀀스 다이어그램 등 필요



# 1. AUTOSAR 개요 및 배경

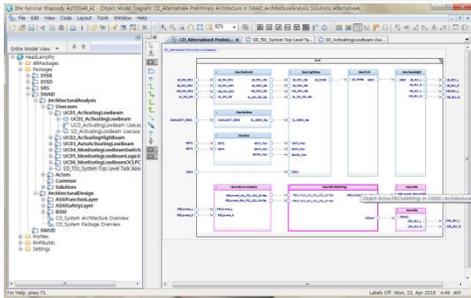
- ✓ AUTOSAR 표준 및 도구에서 필요한 사항 : ② 요구사항 추적성 모델링 능력
  - SW 요구사항과 설계 개념 간 추적링크 확보 능력
  - 요구사항 추적성 매트릭스 출력 필요

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
		(T1) Requirements Traceability Matrix (SWUnit to SRS)																									
Requirements Traceability Matrix		Requirement	SR													SSR											
			HLP_SR_001	HLP_SR_001-01	HLP_SR_001-02	HLP_SR_001-03	HLP_SR_001-04	HLP_SR_001-05	HLP_SR_001-06	HLP_SR_001-07	HLP_SR_001-08	HLP_SR_001-09	HLP_SR_001-10	HLP_SR_001-11	HLP_SR_001-12	HLP_SR_001-13	HLP_SR_002	HLP_SR_003	HLP_SSR_001	HLP_SSR_002	HLP_SSR_003	HLP_SSR_004	HLP_SSR_005	HLP_SSR_006			
		Covered	X			X	X	X																			
SWUnit		Rtld	6	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SwcSwEvent	X	4	X			X	X	X																			
SwcLightReq	X	1	X																								
SwcFLM	X	1	X																								
SwcHeadlight	X	1	X																								
SwcBat	X	1	X																								
SwcSLSens	X	1	X																								
SwcCalData		0																									
SwcSwSafeMngr	X	2																			X	X					
SwcLogicSafeMngr	X	1																						X			
SwcFBCurrentSens	X	3																							X	X	X
SwcFBCTSafeMngr	X	3																							X	X	X
SwcHMI	X	3																							X	X	X
SwcDRL	X	3																							X	X	X
OS		0																									
DEM		0																									

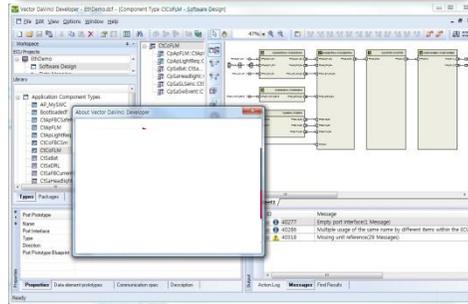
# 2. AUTOSAR ASW 개발 절차 개요

## ✓ AUTOSAR ASW 개발 절차

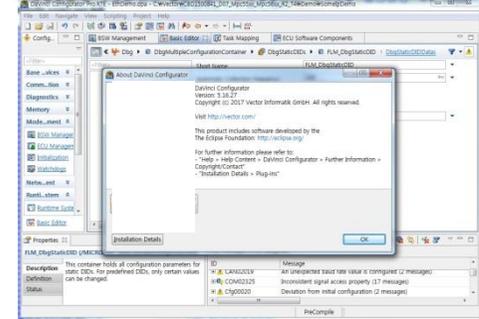
① SW 아키텍처 설계 (UML 도구)



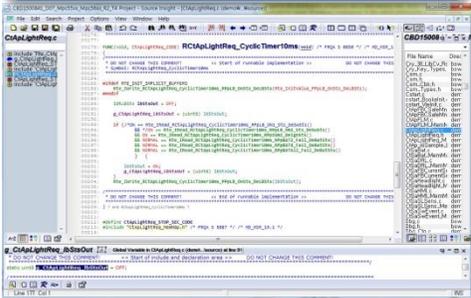
② AUTOSAR ASW 설계 (AUTOSAR 설계 도구)



③ RTE 설정 및 코드 생성 (AUTOSAR RTE 설정 도구)



④ AUTOSAR ASW 구현 (텍스트 편집기)



⑤ 빌드 및 Target 업로드



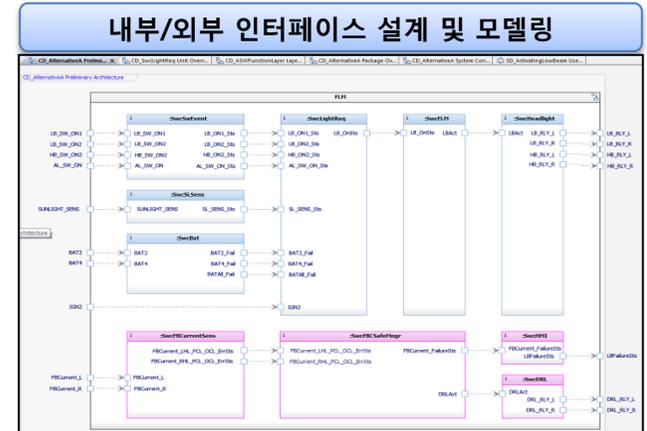
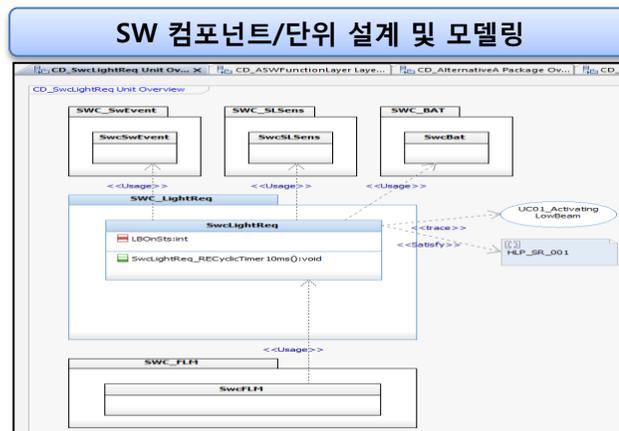
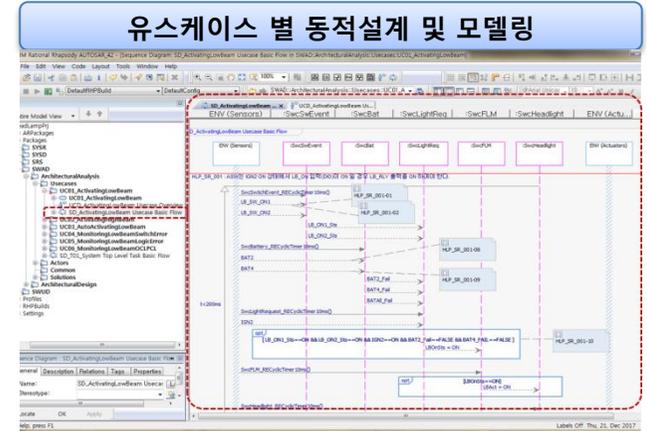
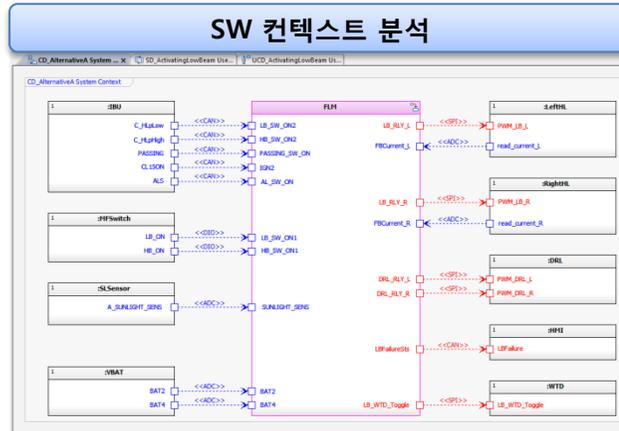
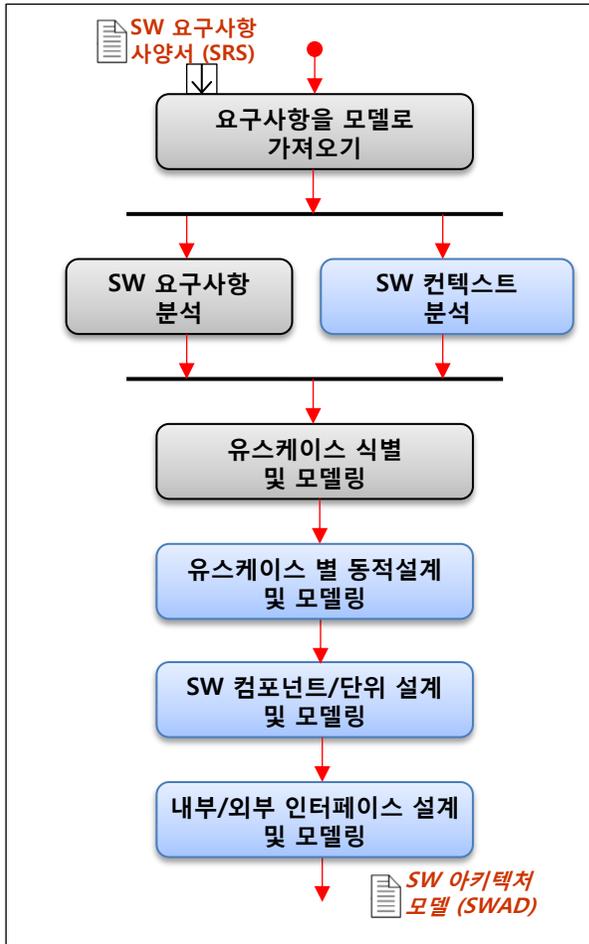
⑥ 실행



# 3. SW 아키텍처 설계

## 정적구조 설계 : ① SW 컨텍스트 분석 및 외부 인터페이스 식별

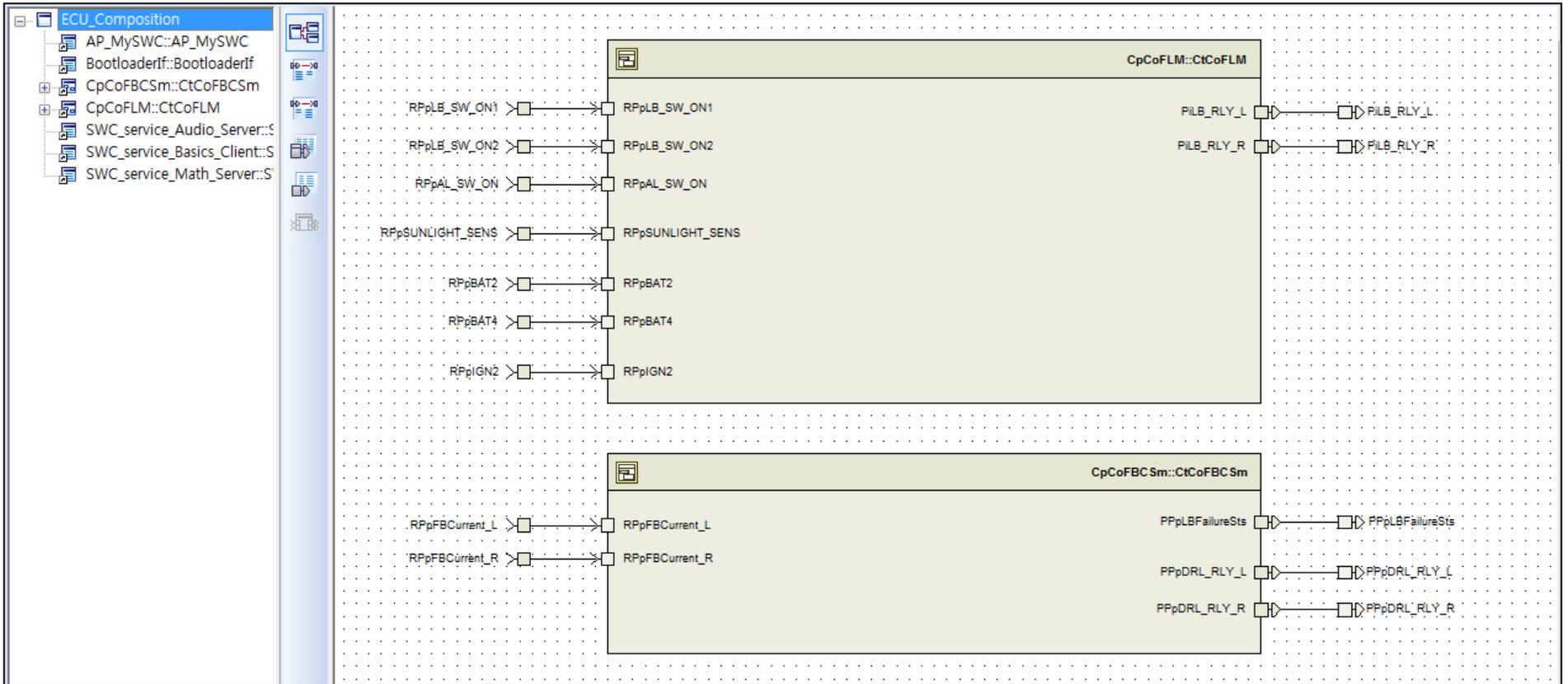
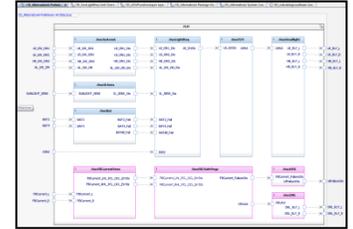
- SW 컨텍스트를 분석하여 SW 시스템 경계를 정의하고, 외부 인터페이스를 식별
- SW 컨텍스트를 클래스 다이어그램으로 작성



# 4. AUTOSAR ASW 설계

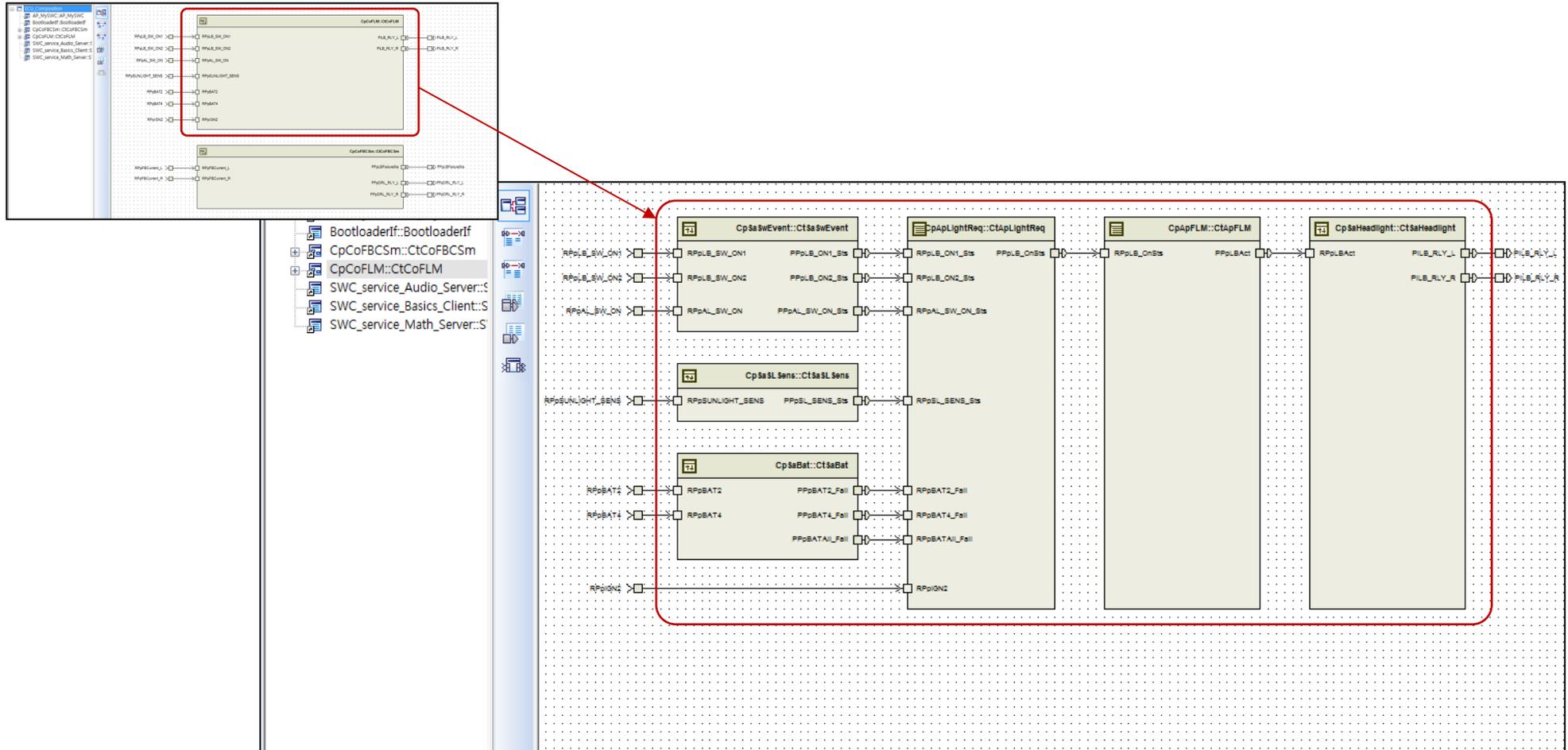
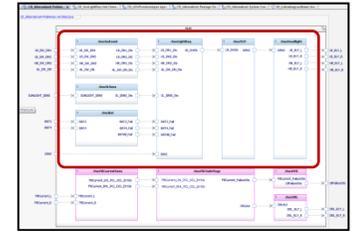
## ✓ AUTOSAR ASW 아키텍처 설계 : CompositionType 레벨

- 일반기능 CompositeType : CtCoFLM 설계
- 안전기능 CompositeType : CtCoFBCSm 설계



# 4. AUTOSAR ASW 설계

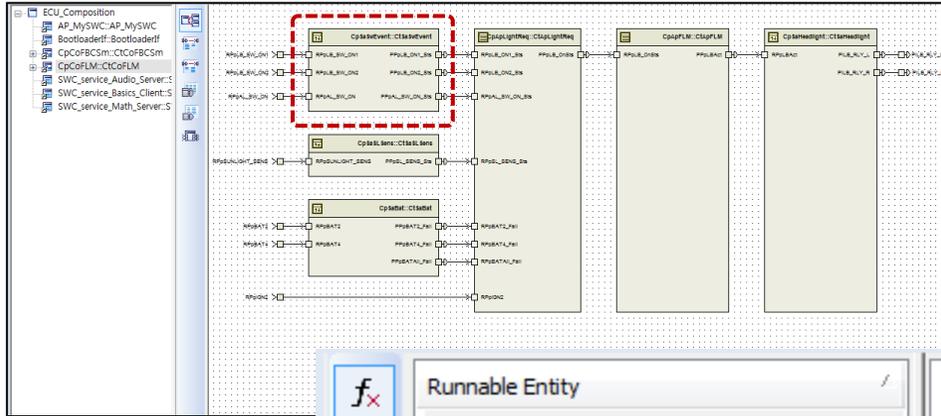
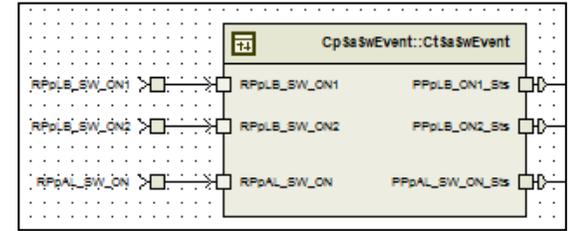
- ✓ AUTOSAR ASW 아키텍처 설계 : (Atomic) SW Component 레벨
  - 일반기능 CompositeType : CtCoFLM 상세 설계



# 4. AUTOSAR ASW 설계

## ✓ AUTOSAR SW Component의 상세 Runnable 설계

- Runnable Entity 설계
  - Runnable의 Trigger, Port Access 등 설정



Runnable Entity

- RCTsaSwEvent\_CyclicTimer 10ms
- RCTsaSwEvent\_RPpAL\_SW\_ON\_DeALSwSts
- RCTsaSwEvent\_RPpLB\_SW\_ON2\_DeSwSts

---

Runnable Entity

- RCTsaSwEvent\_CyclicTimer 10ms
- RCTsaSwEvent\_RPpAL\_SW\_ON\_DeALSwSts
- RCTsaSwEvent\_RPpLB\_SW\_ON2\_DeSwSts**

---

Runnable Entity

- RCTsaSwEvent\_CyclicTimer 10ms
- RCTsaSwEvent\_RPpAL\_SW\_ON\_DeALSwSts
- RCTsaSwEvent\_RPpLB\_SW\_ON2\_DeSwSts

Trigger	Port Access	Properties	Attributes	Description
Trigger	Disabled in Modes		Name	
10 msec			TMT_RCTsaSwEvent_RPpLB_SW_ON1_10ms	

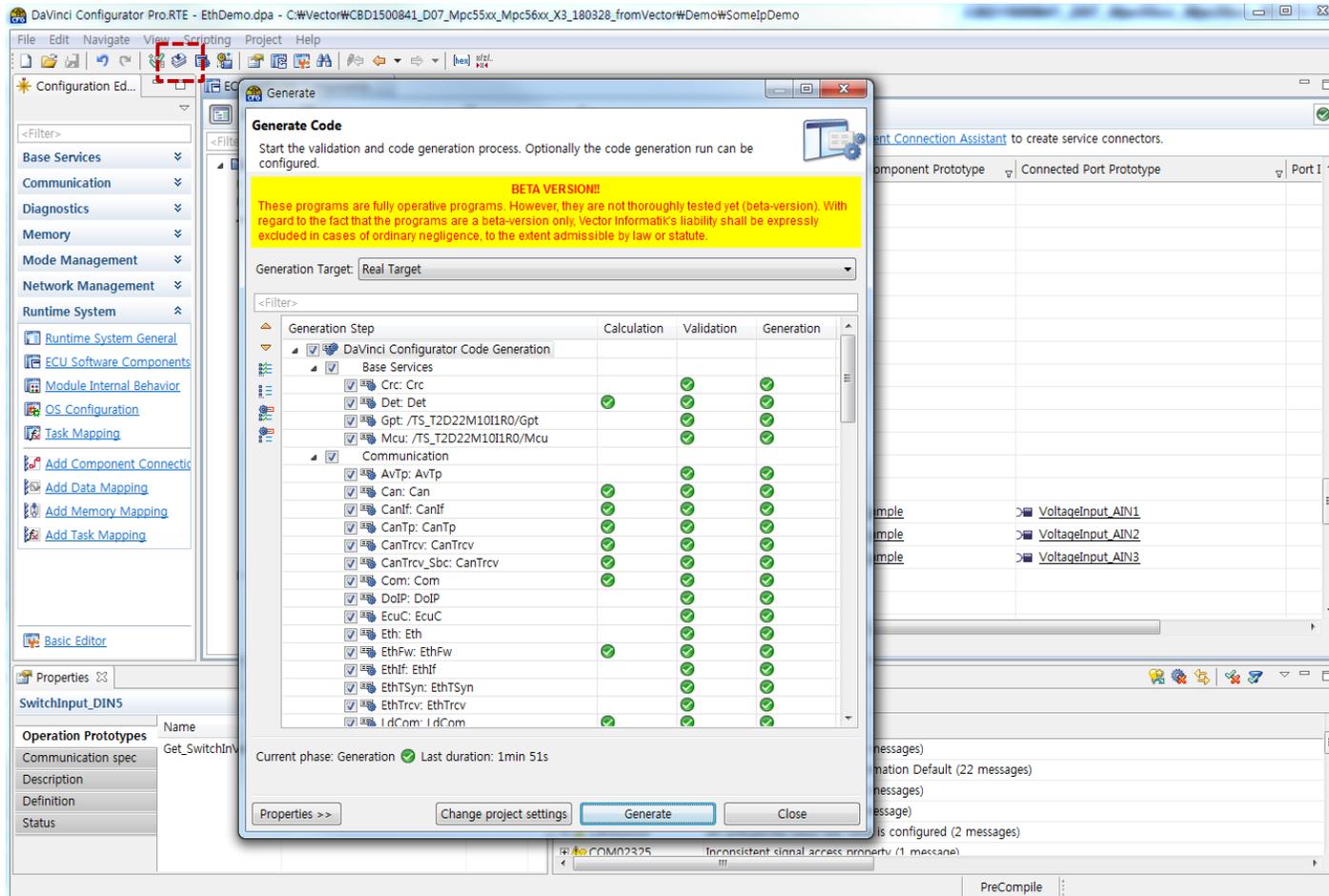
Trigger	Port Access	Properties	Attributes	Description
Trigger	Disabled in Modes		Name	
RPpLB_SW_ON2,DeSwSts			DRT_RCTsaSwEvent_RPpLB_S	

Trigger	Port Access	Properties	Attributes	Description
Trigger	Disabled in Modes		Name	
RPpAL_SW_ON,DeALSwSts			DRT_RCTsaSwEvent_RPpAl	

# 5. AUTOSAR RTE 설정 및 코드 생성

## ✓ RTE Configuration 및 Code Generation

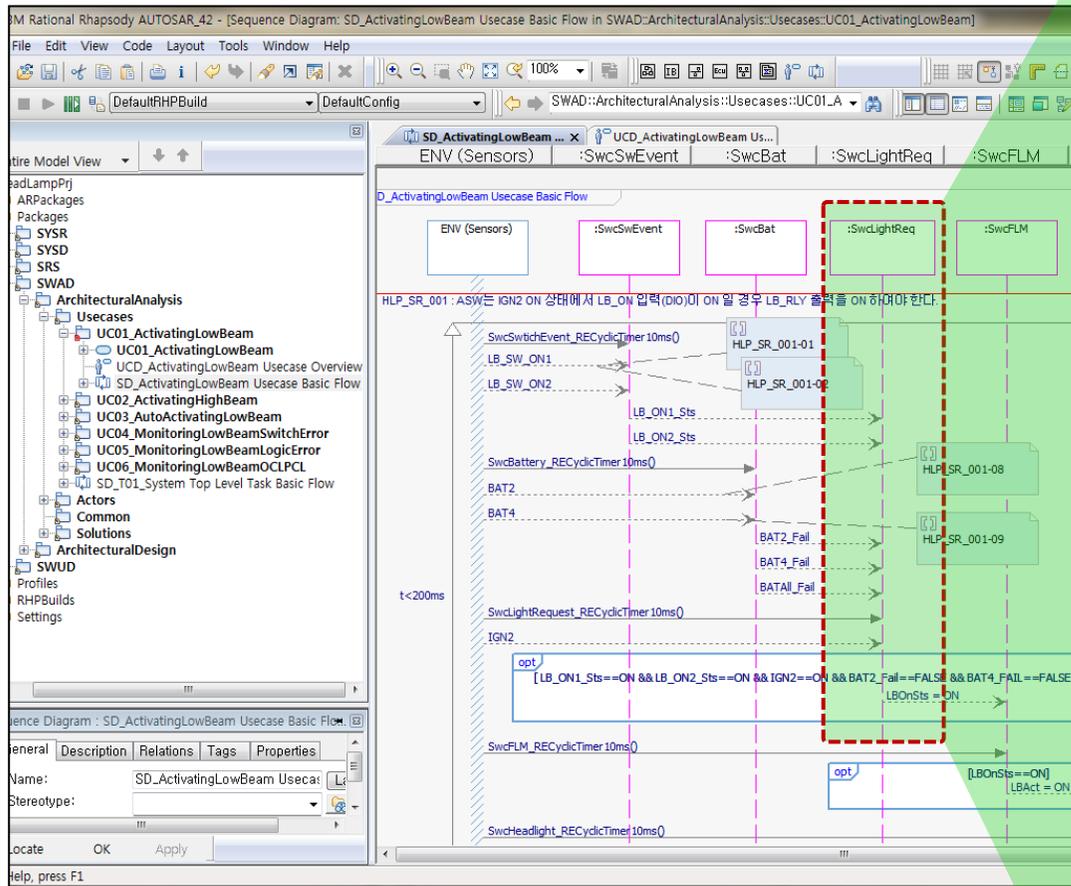
- RTE 설정
- RTE 및 BSW 코드 생성



# 6. AUTOSAR ASW 구현

## ✓ Runnable 구현

- Runnable에 대한 Function 코드 구현

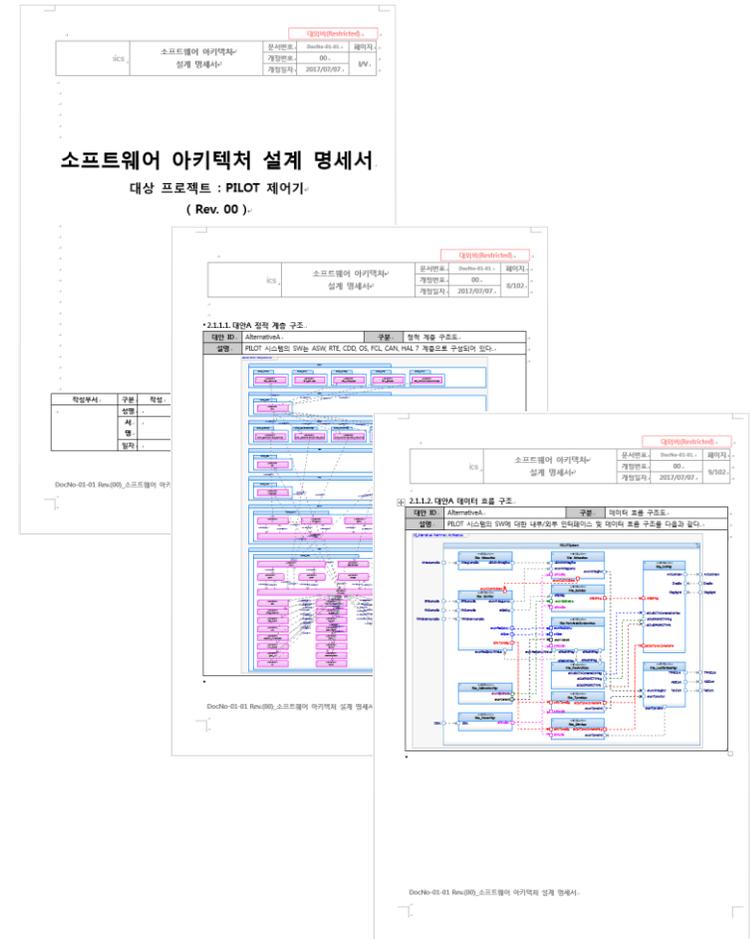
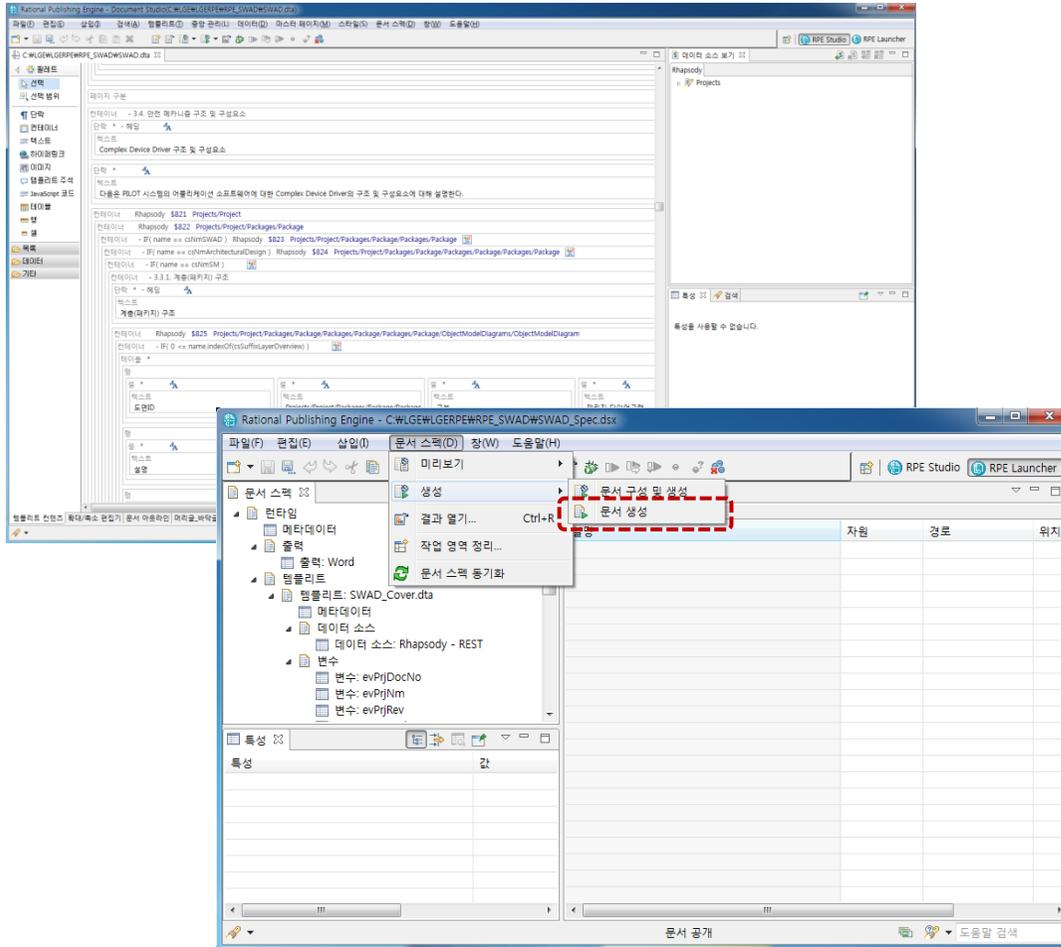


```
00137: /*.....
00138: *
00139: * Runnable Entity Name: RCTApLightReq_CyclicTimer10ms
00140: *
00141: *.....
00142: *
00143: * Executed if at least one of the following trigger conditions occurred:
00144: * - triggered on TimingEvent every 10ms
00145: *.....
00146: *
00147: * Input Interfaces:
00148: * =====
00149: * Implicit S/R API:
00150: * -----
00151: *
00152: * IdtALSwSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPAL_SW_ON_Sts_DeALSwSts(void)
00153: * IdtBatSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBAT2_Fail_DeBatSts(void)
00154: * IdtBatSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBAT4_Fail_DeBatSts(void)
00155: * IdtBatSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBATA11_Fail_DeBatSts(void)
00156: * IdtIgnSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPIGN2_DeIgnSts(void)
00157: * IdtSwSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPLB_ON1_Sts_DeSwSts(void)
00158: * IdtSwSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPLB_ON2_Sts_DeSwSts(void)
00159: * IdtLSensSts Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPSL_SENS_Sts_DeLSensSts(void)
00160: *
00161: * Output Interfaces:
00162: * =====
00163: * Implicit S/R API:
00164: * -----
00165: * void Rte_IWriteRef_RCTApLightReq_CyclicTimer10ms_PPPLB_OnSts_DeLBSts(IdtLBSts data)
00166: * IdtLBSts *Rte_IWriteRef_RCTApLightReq_CyclicTimer10ms_PPPLB_OnSts_DeLBSts(void)
00167: *
00168: *.....
00169: * DO NOT CHANGE THIS COMMENT! << Start of documentation area >> DO NOT CHANGE THIS COMMENT!
00170: * Symbol: RCTApLightReq_CyclicTimer10ms_doc
00171: *.....
00172: *
00173: * DO NOT CHANGE THIS COMMENT! << End of documentation area >> DO NOT CHANGE THIS COMMENT!
00174: *.....
00175: *
00176: * DO NOT CHANGE THIS COMMENT! << Start of runnable implementation >> DO NOT CHANGE THIS COMMENT!
00177: * Symbol: RCTApLightReq_CyclicTimer10ms
00178: *.....
00179: FUNC(void, CTApLightReq_CODE) RCTApLightReq_CyclicTimer10ms(void) /* PRQA S 0850 */ /* MD_MSR_19.8 */
00180: {
00181: *.....
00182: * DO NOT CHANGE THIS COMMENT! << Start of runnable implementation >> DO NOT CHANGE THIS COMMENT!
00183: * Symbol: RCTApLightReq_CyclicTimer10ms
00184: *.....
00185:
00186: #ifdef RTE_INIT_IMPLICIT_BUFFERS
00187: Rte_IWriteRef_RCTApLightReq_CyclicTimer10ms_PPPLB_OnSts_DeLBSts(Rte_Initvalue_PPPLB_OnSts_DeLBSts);
00188: #endif
00189:
00190: IdtLBSts lbStsOut = OFF;
00191:
00192: g_ctApLightReq_LbStsOut = (uint8) lbStsOut;
00193:
00194: if (ON == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPLB_ON1_Sts_DeSwSts()
00195:     && ON == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPLB_ON2_Sts_DeSwSts()
00196:     && ON == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPIGN2_DeIgnSts()
00197:     && NORMAL == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBAT2_Fail_DeBatSts()
00198:     && NORMAL == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBAT4_Fail_DeBatSts()
00199:     && NORMAL == Rte_IRead_RCTApLightReq_CyclicTimer10ms_RPPBATA11_Fail_DeBatSts()
00200:     ) {
00201:     lbStsOut = ON;
00202:     g_ctApLightReq_LbStsOut = (uint8) lbStsOut;
00203: }
00204: Rte_IWriteRef_RCTApLightReq_CyclicTimer10ms_PPPLB_OnSts_DeLBSts(lbStsOut);
00205:
00206: *.....
00207: * DO NOT CHANGE THIS COMMENT! << End of runnable implementation >> DO NOT CHANGE THIS COMMENT!
00208: *.....
00209: } > end RCTApLightReq_CyclicTimer10ms ?
00210: } > end RCTApLightReq_CyclicTimer10ms ?
```

# 7. SW 설계 산출물 출력

## ✓ SW 설계서 자동 출력

- “SW 모델”로부터 “SW 아키텍처 설계서” 문서를 자동 출력
- “SW 모델”로부터 “SW 단위 상세 설계서” 문서를 자동 출력



Smart  
System  
Software

Process

Product  
Professional  
People

Durable  
Delivery  
Deployment

# SPID

Improvement

Innovation  
Intelligent

spid

최광목 수석 컨설턴트 / Process 사업본부 / (주)에스피아이디

서울시 강남구 선릉로93길 27, 아름빌딩 4층 (135-513)

02-3453-5345 / Fax: 02-3453-5346 / kmchoi@espid.com

[www.espid.com](http://www.espid.com) / [www.espid.co.kr](http://www.espid.co.kr)