Automotive SPICE v4.0

A-SPICE(Automotive Software Process Improvement dEtermination)은 ISO 330를 기반으로 개발된 자동차 분야의 산업계 통용 표준 모델입니다.**

'23.12 v4.0이 공식 배포되었고, '24.03부터 1년간 transition phase를 거친 후 v4.0만 유효한 버전이 될 예정입니다. v4.0은 아래와 같은 배경 및 동기를 고려하여 개정이 진행되었습니다.

【 배경 】

2017.11 ASPICE v3.1, ASPICE guideline 활용 현황을 보니..

- 심사 결과의 재현성과 비교성이 개선되지 않음
- 심사 기간 증가 (52% 심사가 5일 이상)
- VDA BGB Guideline 내용이 순수 체크리스트로 활용
- 심사 대상 프로젝트의 문맥(context)에 대한 이해보다 형식성 증가
- 프로젝트 팀원들의 동기 저하

@ VDA QMC survey on assessment quality in 2022

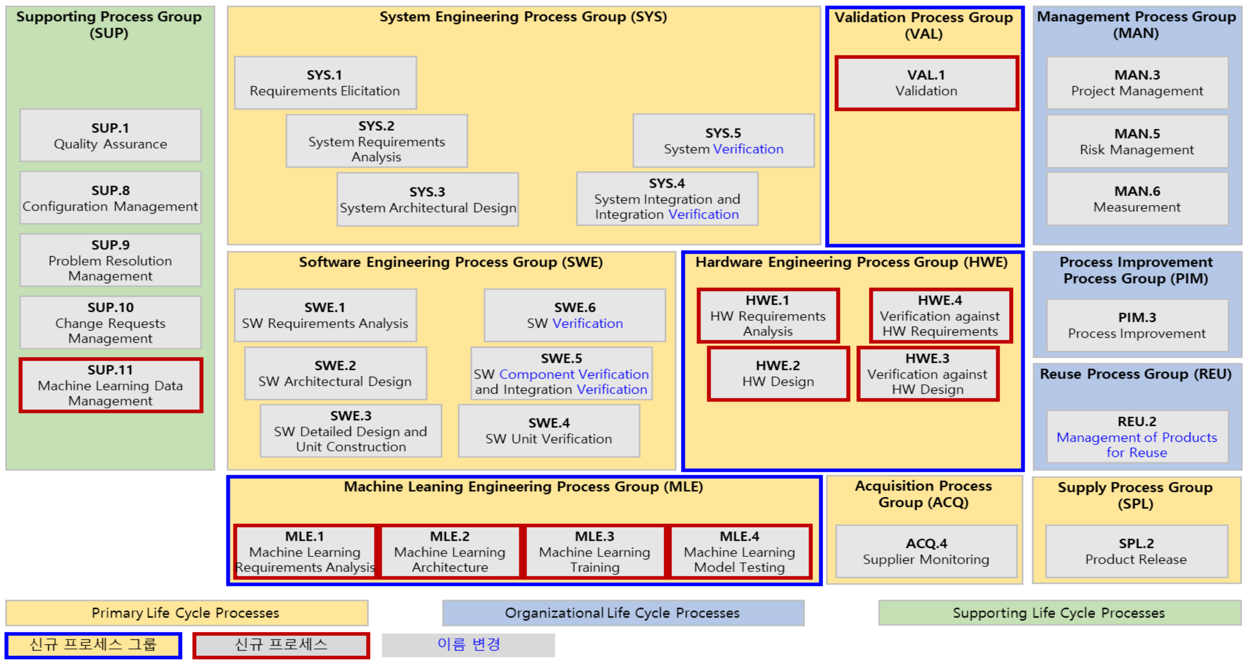
【 동기 】

이를 위해 ASPICE 4.0으로 개정이 진행되고 있음

- 심사 결과의 재현성과 비교성을 최대한 달성
- 심사 효율성 개선
- 최근 engineering state-of-the-art 반영 (e.g. Functional safety, Cybersecuity, Machine learning, new supplier..)
- 중복된 내용/평가 축소
- 잘못된 해석의 가능성 최소화
- 심사 모델 재구조화

A-SPICE v4.0은 10개 프로세스가 삭제되고, 3개 프로세스 그룹 및 10개 프로세스가 추가되는 큰 변화가 포함되어 있습니다.

- 삭제된 프로세스 : ACQ.3/11/12/13/14/15, SPL.1, SUP.2/4/7
- 추가된 프로세스 그룹 : Hardware engineering, Machine learning engineering, Validation
- 추가된 프로세스 : VAL.1, HWE.1/2/3/4, MLE.1/2/3/4, SUP.11
- 이름 변경된 프로세스 : SYS.4/5, SWE.5/6, REU.2


Supporting Process Group (SUP)
SUP.1 Quality Assurance
SUP.8 Configuration Management
SUP.9 Problem Resolution Management
SUP.10 Change Requests Management
SUP.11 Machine Learning Data Management
System Engineering Process Group (SYS)
SYS.1 Requirements Elicitation
SYS.2 System Requirements Analysis
SYS.3 System Architectural Design
SYS.4 System Integration and Integration Verification
SYS.5 System Verification
Validation Process Group (VAL)
VAL.1 Validation
Management Process Group (MAN)
MAN.3 Project Management
MAN.5 Risk Management
MAN.6 Measurement
Software Engineering Process Group (SWE)
SWE.1 SW Requirements Analysis
SWE.2 SW Architectural Design
SWE.3 SW Detailed Design and Unit Construction
SWE.4 SW Unit Verification
SWE.5 SW Component Verification and Integration Verification
SWE.6 SW Verification
Hardware Engineering Process Group (HWE)
HWE.1 HW Requirements Analysis
HWE.2 HW Design
HWE.3 Verification against HW Design
HWE.4 Verification against HW Requirements
Process Improvement Process Group (PIM)
PIM.3 Process Improvement
Reuse Process Group (REU)
REU.2 Management of Products for Reuse
Machine Leaning Engineering Process Group (MLE)
MLE.1 Machine Learning Requirements Analysis
MLE.2 Machine Learning Architecture
MLE.3 Machine Learning Training
MLE.4 Machine Learning Model Testing
Acquisition Process Group (ACQ)
ACQ.4 Supplier Monitoring
Supply Process Group (SPL)
SPL.2 Product Release
Primary Life Cycle Processes
Organizational Life Cycle Processes
Supporting Life Cycle Processes
신규 프로세스 그룹
신규 프로세스
이름 변경

Automotive SPICE v4.0 major changes

Major change 1 Plug-in concept, VDA scope 변경

Plug-in concept이 확장되며 VDA scope에서 Recommended VDA scope으로 이름이 변경되었고, 프로젝트 상황에 따라 일부를 선정하여 적용하도록 개정되었습니다.

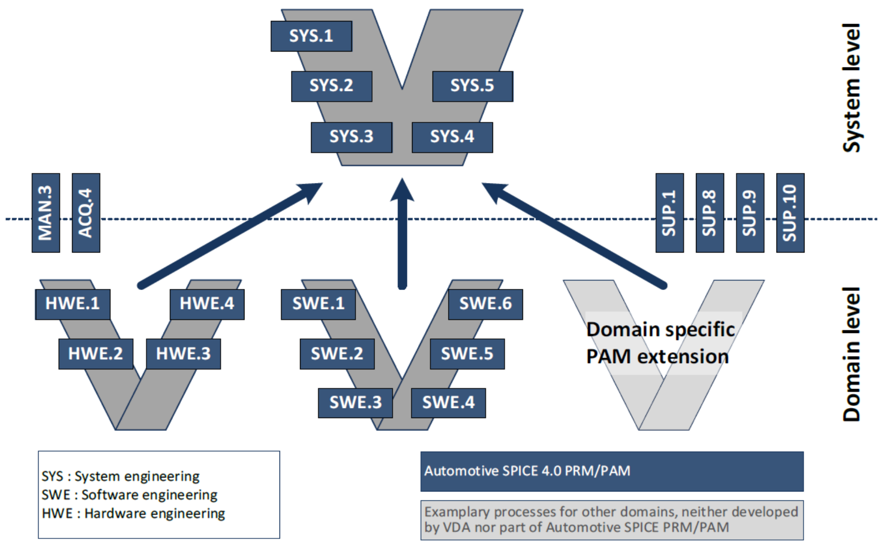
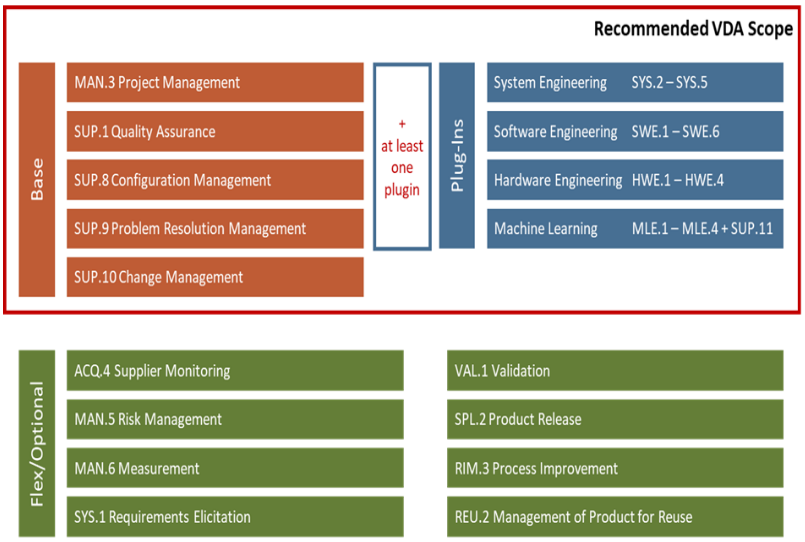

SYS.1
SYS.2
SYS.5
SYS.3
SYS.4
System level
MAN.3
ACQ.4
SUP.1
SUP.8
SUP.9
SUP.10
HWE.1
HWE.4
HWE.2
HWE.3
SWE.1
SWE.6
SWE.2
SWE.5
SWE.3
SWE.4
Domain specific PAM extension
Domain level
SYS : System engineering
SWE : Software engineering
HWE : Hardware engineering
Automotive SPICE 4.0 PRM/PAM
Examplary processes for other domains, neither developed by VDA nor part of Automotive SPICE PRM/PAM

Figure C.1 — The "Plug-in" concept



Recommended VDA Scope

Base	MAN.3 Project Management	+ at least one plugin	Plug-Ins	System Engineering SYS.2 – SYS.5
	SUP.1 Quality Assurance			Software Engineering SWE.1 – SWE.6
	SUP.8 Configuration Management			Hardware Engineering HWE.1 – HWE.4
	SUP.9 Problem Resolution Management			Machine Learning MLE.1 – MLE.4 + SUP.11
	SUP.10 Change Management			

Flex/Optional	ACQ.4 Supplier Monitoring	VAL.1 Validation
	MAN.5 Risk Management	SPL.2 Product Release
	MAN.6 Measurement	RIM.3 Process Improvement
	SYS.1 Requirements Elicitation	REU.2 Management of Product for Reuse

※ 참고 : VDA QMC ASPICE guideline v2.0

Rules

- Sponsor/Customer가 Scope 정의
- **Recommended VDA Scope** 선정은 **Basic scope + 1개 이상의 Plug-Ins**으로 되어야함
- Flex/Optional 파트는 project 고유 context에 따라 선택될 수 있음
- Recommended VDA Scope 선정 예)
  - 구 VDA scope = Basic + SYS + SWE
  - Electronic dev. = Basic + SYS + HWE
  - Software only = Basic + SWE

Automotive SPICE v4.0 major changes

Major change 2 Traceability diagram 개정

Engineering process group이 대폭 수정됨에 따라 System-Software, System-Hardware, Software-Machine learning 간 traceability diagram이 정의되었습니다.

- Bidirectional traceability BP & consistency BP 통합
- Test case가 verification measure로 변경
- Validation measure와 관련된 추적 추가
- SW integration verification measure ↔ SW detailed design 추적 추가

【System - Software】

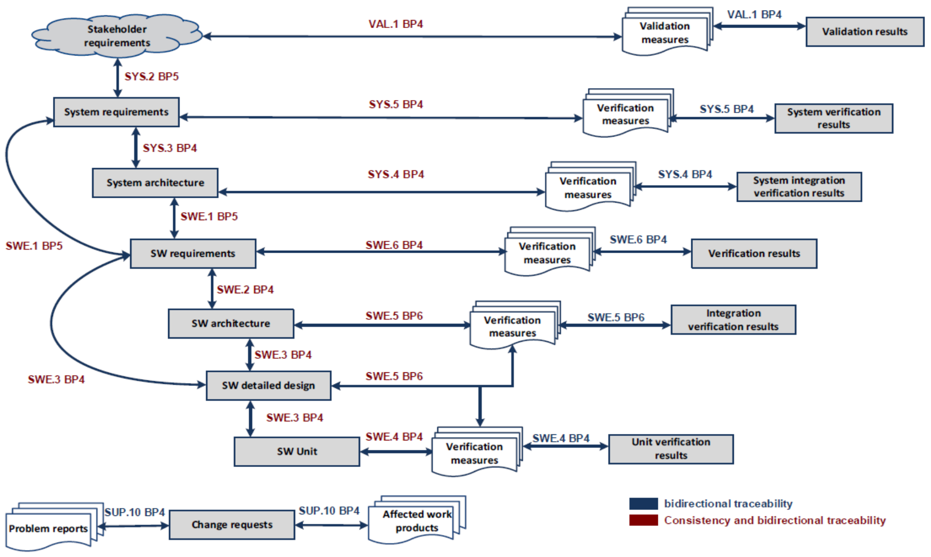


Figure C.2 — Consistency and traceability between system and software work products

【System - Hardware】

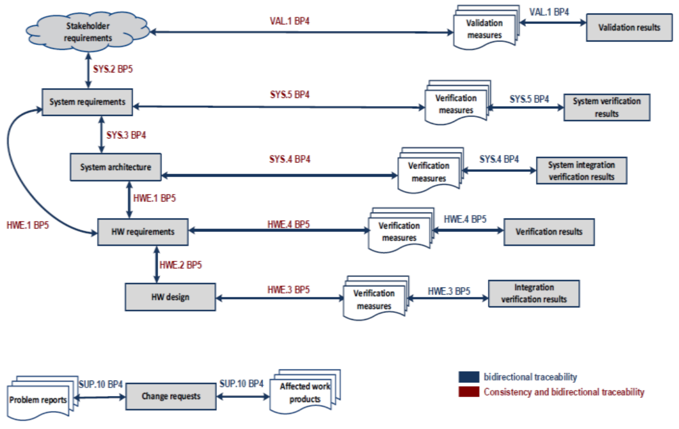


Figure C.3 — Consistency and traceability between system and hardware work products

【Software - Machine learning】

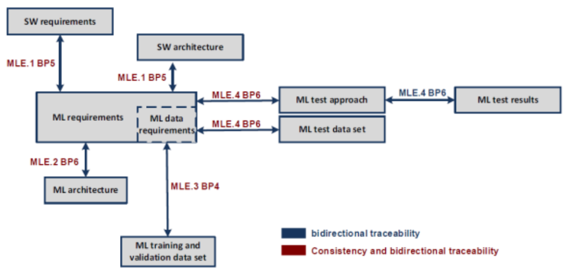


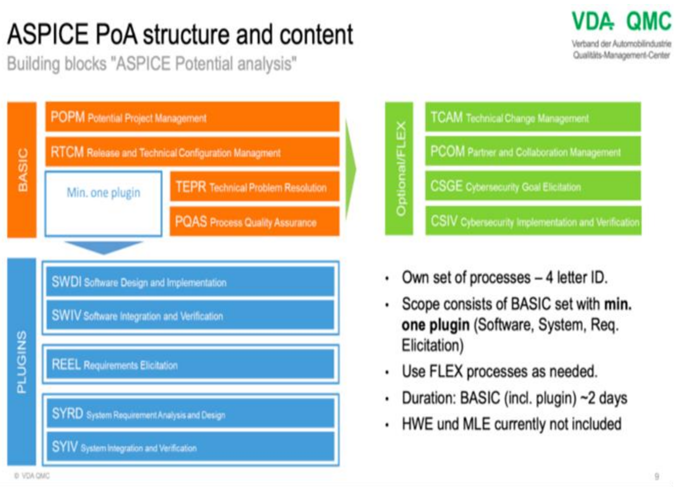
Figure C.4 — Consistency and traceability between ML work products

Automotive SPICE v4.0 major changes

Major change 3 PoA(Potential analysis) 신규 개발 중

차량의 complexity 및 integration 범위 증가로 신생 후보 협력업체가 급증하고, 이에 협력업체 선정 시 빠르고 신뢰할 수 있는 분석 방법이 필요하여 A-SPICE PoA(Potential analysis)라는 별도 PAM이 개발되고 있으며, 주요 특징은 다음으로 예상되고 있습니다.

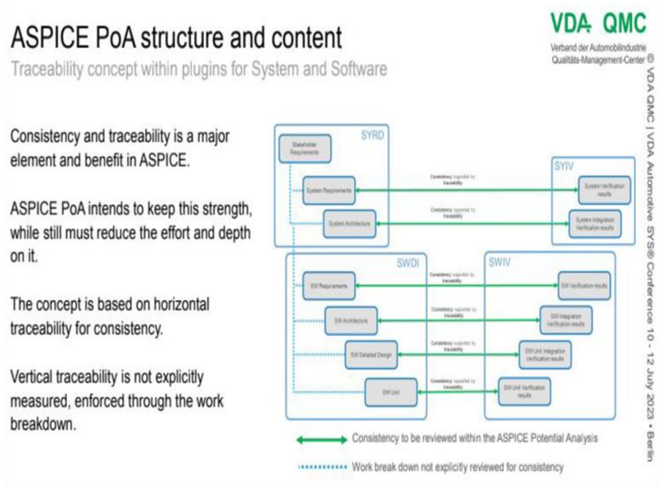
- Process capability가 아닌 risk evaluation 중심
- Base + Domain 최소 1개 (HWE, MLE: 현재 미포함), CL1만 대상
- NPLE가 아닌 Red-yellow-green으로 평가
- 2일정도 공인 ASPICE assessor가 진행


VDA QMC
Verband der Automobilindustrie
Qualitäts-Management-Center
BASIC
POPM Potential Project Management
RTCM Release and Technical Configuration Managment
Min. one plugin
TEPR Technical Problem Resolution
PQAS Process Quality Assurance
Optional/FLEX
TCAM Technical Change Management
PCOM Partner and Collaboration Management
CSGE Cybersecurity Goal Elicitation
CSIV Cybersecurity Implementation and Verification
PLUGINS
SWDI Software Design and Implementation
SWIV Software Integration and Verification
REEL Requirements Elicitation
SYRD System Requirement Analysis and Design
SYIV System Integration and Verification
© VDA QMC
9

ASPICE PoA structure and content

Building blocks "ASPICE Potential analysis"

- Own set of processes – 4 letter ID.
- Scope consists of BASIC set with **min. one plugin** (Software, System, Req. Elicitation)
- Use FLEX processes as needed.
- Duration: BASIC (incl. plugin) ~2 days
- HWE und MLE currently not included


VDA QMC
Verband der Automobilindustrie
Qualitäts-Management-Center
© VDA QMC | VDA Automotive SYS® Conference 10 - 12 July 2023 • Berlin
SYRD
Stakeholder Requirements
System Requirements
System Architecture
SYIV
System Verification results
System Integration Verification results
SWDI
SW Requirements
SW Architecture
SW Detailed Design
SW Unit
SWIV
SW Verification results
SW Integration Verification results
SW Unit Integration Verification results
SW Unit Verification results
Consistency supported by traceability
Consistency to be reviewed within the ASPICE Potential Analysis
Work break down not explicitly reviewed for consistency

ASPICE PoA structure and content

Traceability concept within plugins for System and Software

Consistency and traceability is a major element and benefit in ASPICE.

ASPICE PoA intends to keep this strength, while still must reduce the effort and depth on it.

The concept is based on horizontal traceability for consistency.

Vertical traceability is not explicitly measured, enforced through the work breakdown.

- **BASIC : 4개로 구성**
  . PM(POPM), CM(RTCM), PRM(TEPR), QA(PQAS)
- **Plugins : 3개로 구성**
  . System : Requirement & design(SYRD), integration & verification(SYIV)
  . SW : design & implementation(SWDI), integration & verification(SWIV)
  . Requirements elicitation(REEL)
- **Flex : 4개로 구성**
  . CRM(TCAM), Partner management(PCOM)
  CS goal elicitation(CSGE), CS integration & verification(CSIV)

- ASPICE의 traceability & consistency는
  개발 산출물의 horizontal, vertical이 모두 정의된 것에 반해
- **ASPICE PoA의 traceability & consistency는**
  빠르고 효과적인 점검을 위해 **horizontal만 포함**

SOLUTIONLINK

Automotive SPICE v4.0 major changes

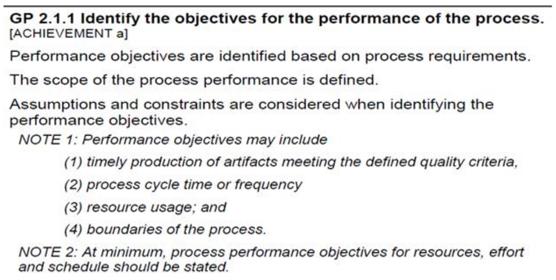
Major change 4

Strategy/Plan BP 삭제

→ But, 신규 BP 추가 또는 다른 BP에 흡수

v3.1에서는 CL1 performed 및 CL2 managed에 정의된 의미적 차이와 CL1 각 프로세스의 strategy BP 의미적 혼선이 있어서 v4.0에서는 CL1에 strategy/plan BP는 삭제되고 CL2에서 strategy/plan을 모두 요구하도록 변경되었습니다.

【 v3.1 】



GP 2.1.1 Identify the objectives for the performance of the process. [ACHIEVEMENT a]

Performance objectives are identified based on process requirements.

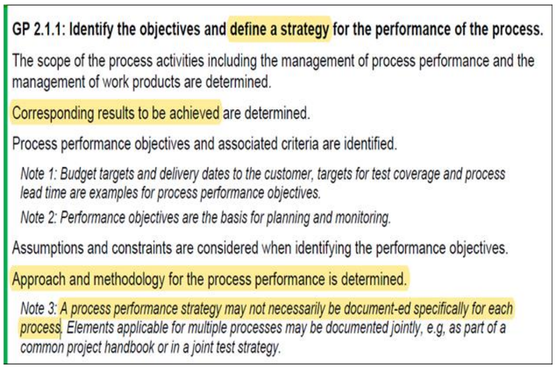
The scope of the process performance is defined.

Assumptions and constraints are considered when identifying the performance objectives.

NOTE 1: Performance objectives may include
(1) timely production of artifacts meeting the defined quality criteria,
(2) process cycle time or frequency
(3) resource usage; and
(4) boundaries of the process.

NOTE 2: At minimum, process performance objectives for resources, effort and schedule should be stated.

【 v4.0】



GP 2.1.1: Identify the objectives and define a strategy for the performance of the process.

The scope of the process activities including the management of process performance and the management of work products are determined.

Corresponding results to be achieved are determined.

Process performance objectives and associated criteria are identified.

Note 1: Budget targets and delivery dates to the customer, targets for test coverage and process lead time are examples for process performance objectives.

Note 2: Performance objectives are the basis for planning and monitoring.

Assumptions and constraints are considered when identifying the performance objectives.

Approach and methodology for the process performance is determined.

Note 3: A process performance strategy may not necessarily be document-ed specifically for each process. Elements applicable for multiple processes may be documented jointly, e.g. as part of a common project handbook or in a joint test strategy.

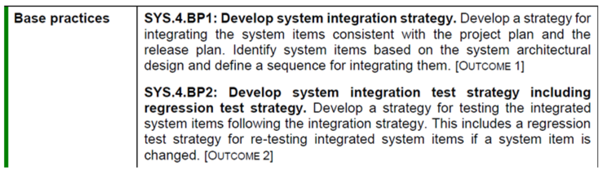
다만, CL2 managed의 key elements로 GP2.1.1에 strategy가 추가됨에 따라..

- V3.1은 일부 process만 strategy가 요구되었으나 v4.0은 CL2 시 모든 영역에 strategy 필요
- Strategy는 objectives and criteria, approach/methodology를 포함함

그러나 CL1에서 strategy가 완전히 사라진 것은 아니며, 의미는 여전히 존재합니다.

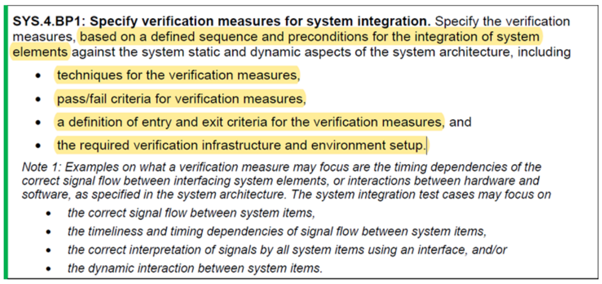
- SYS.4와 같이 기존에 있던 BP에 내용 추가
- 또는 SUP.1과 같이 새로운 BP 추가

【 v3.1 】



Base practices	**SYS.4.BP1: Develop system integration strategy.** Develop a strategy for integrating the system items consistent with the project plan and the release plan. Identify system items based on the system architectural design and define a sequence for integrating them. [OUTCOME 1] **SYS.4.BP2: Develop system integration test strategy including regression test strategy.** Develop a strategy for testing the integrated system items following the integration strategy. This includes a regression test strategy for re-testing integrated system items if a system item is changed. [OUTCOME 2]

【 v4.0】

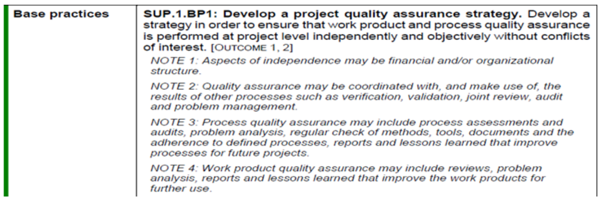


SYS.4.BP1: Specify verification measures for system integration. Specify the verification measures, based on a defined sequence and preconditions for the integration of system elements against the system static and dynamic aspects of the system architecture, including

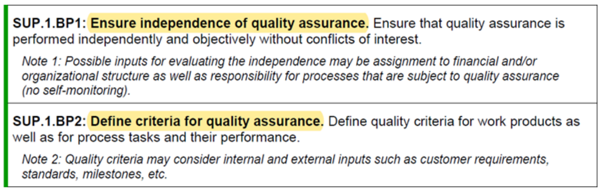
- techniques for the verification measures,
- pass/fail criteria for verification measures,
- a definition of entry and exit criteria for the verification measures, and
- the required verification infrastructure and environment setup.

Note 1: Examples on what a verification measure may focus are the timing dependencies of the correct signal flow between interfacing system elements, or interactions between hardware and software, as specified in the system architecture. The system integration test cases may focus on

- *the correct signal flow between system items,*
- *the timeliness and timing dependencies of signal flow between system items,*
- *the correct interpretation of signals by all system items using an interface, and/or*
- *the dynamic interaction between system items.*



Base practices	**SUP.1.BP1: Develop a project quality assurance strategy.** Develop a strategy in order to ensure that work product and process quality assurance is performed at project level independently and objectively without conflicts of interest. [OUTCOME 1, 2] *NOTE 1: Aspects of independence may be financial and/or organizational structure.* *NOTE 2: Quality assurance may be coordinated with, and make use of, the results of other processes such as verification, validation, joint review, audit and problem management.* *NOTE 3: Process quality assurance may include process assessments and audits, problem analysis, regular check of methods, tools, documents and the adherence to defined processes, reports and lessons learned that improve processes for future projects.* *NOTE 4: Work product quality assurance may include reviews, problem analysis, reports and lessons learned that improve the work products for further use.*



SUP.1.BP1: Ensure independence of quality assurance. Ensure that quality assurance is performed independently and objectively without conflicts of interest.

Note 1: Possible inputs for evaluating the independence may be assignment to financial and/or organizational structure as well as responsibility for processes that are subject to quality assurance (no self-monitoring).

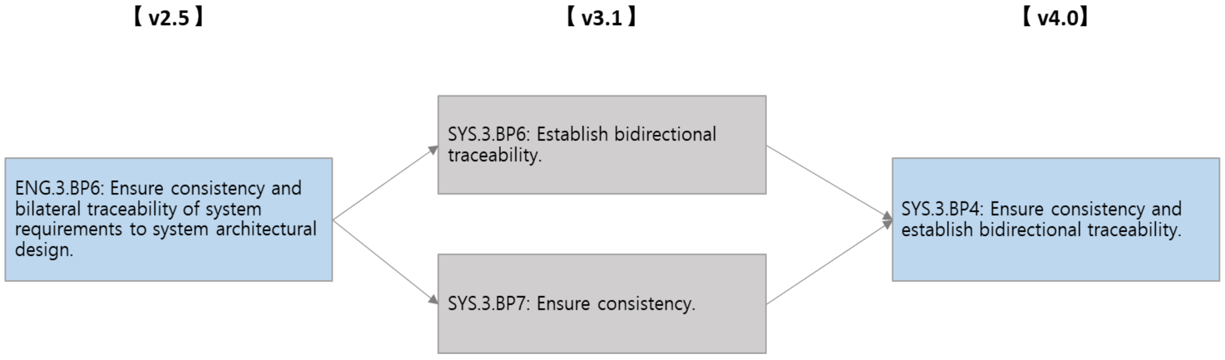
SUP.1.BP2: Define criteria for quality assurance. Define quality criteria for work products as well as for process tasks and their performance.

Note 2: Quality criteria may consider internal and external inputs such as customer requirements, standards, milestones, etc.

Automotive SPICE v4.0 major changes

Major change 5 Traceability, consistency BP 재 통합

v3.1에서는 bidirectional traceability와 consistency를 모두 강조하기 위하여 BP를 분할하여 정의하였으나 분리 후 이점이 없어 v4.0에서는 v2.5와 동일하게 재 통합이 진행되었습니다.

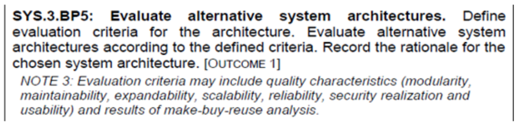

【 v2.5 】
【 v3.1 】
【 v4.0】
ENG.3.BP6: Ensure consistency and bilateral traceability of system requirements to system architectural design.
SYS.3.BP6: Establish bidirectional traceability.
SYS.3.BP7: Ensure consistency.
SYS.3.BP4: Ensure consistency and establish bidirectional traceability.

Major change 6 Evaluation alternative archi. BP 삭제

→ But, Analyze archi.로 확대 정의

v3.1에서는 설계 본질적 가체에 대한 평가로 BP가 활용되지 않고, '대안 식별', '대안의 비교', '다른 대안이 선택되지 않았다'에 중점을 두는 경우가 많았습니다. V4.0에서는 기존 BP는 삭제되고 Analyze architecture BP가 추가되어 아키텍처 분석 관점으로 확대 및 강조되었습니다. 이는 functional safety, cybersecurity의 analysis를 고려한 경향도 있다고 보입니다.

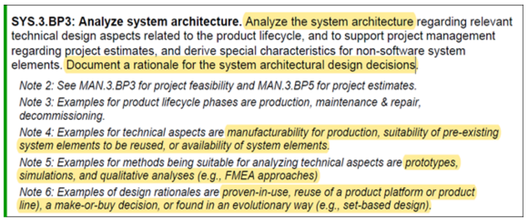
【 v3.1 】



SYS.3.BP5: Evaluate alternative system architectures. Define evaluation criteria for the architecture. Evaluate alternative system architectures according to the defined criteria. Record the rationale for the chosen system architecture. [OUTCOME 1]

NOTE 3: Evaluation criteria may include quality characteristics (modularity, maintainability, expandability, scalability, reliability, security realization and usability) and results of make-buy-reuse analysis.

【 v4.0 】



SYS.3.BP3: Analyze system architecture. Analyze the system architecture regarding relevant technical design aspects related to the product lifecycle, and to support project management regarding project estimates, and derive special characteristics for non-software system elements. Document a rationale for the system architectural design decisions.

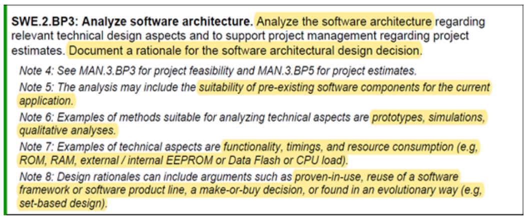
Note 2: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

Note 3: Examples for product lifecycle phases are production, maintenance & repair, decommissioning.

Note 4: Examples for technical aspects are manufacturability for production, suitability of pre-existing system elements to be reused, or availability of system elements.

Note 5: Examples for methods being suitable for analyzing technical aspects are prototypes, simulations, and qualitative analyses (e.g., FMEA approaches)

Note 6: Examples of design rationales are proven-in-use, reuse of a product platform or product line), a make-or-buy decision, or found in an evolutionary way (e.g., set-based design).



SWE.2.BP3: Analyze software architecture. Analyze the software architecture regarding relevant technical design aspects and to support project management regarding project estimates. Document a rationale for the software architectural design decision.

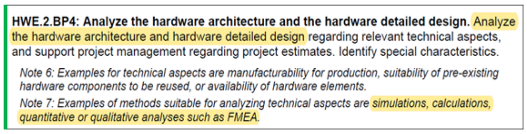
Note 4: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

Note 5: The analysis may include the suitability of pre-existing software components for the current application.

Note 6: Examples of methods suitable for analyzing technical aspects are prototypes, simulations, qualitative analyses.

Note 7: Examples of technical aspects are functionality, timings, and resource consumption (e.g, ROM, RAM, external / internal EEPROM or Data Flash or CPU load).

Note 8: Design rationales can include arguments such as proven-in-use, reuse of a software framework or software product line, a make-or-buy decision, or found in an evolutionary way (e.g, set-based design).



HWE.2.BP4: Analyze the hardware architecture and the hardware detailed design. Analyze the hardware architecture and hardware detailed design regarding relevant technical aspects, and support project management regarding project estimates. Identify special characteristics.

Note 6: Examples for technical aspects are manufacturability for production, suitability of pre-existing hardware components to be reused, or availability of hardware elements.

Note 7: Examples of methods suitable for analyzing technical aspects are simulations, calculations, quantitative or qualitative analyses such as FMEA.

v4.0에서는

- SYS.3 : 기술적 분석(생산을 위한 제조 가능성, legacy system element 재사용 적합성 등) 및 정량적 분석(FMEA 등) 수행. <u>아키텍처 설계결정 근거 명시 필요</u>
- SWE.2 : 기술적 분석(기능, 동작시간, 리소스, legacy SWC의 재사용 적합성 등) 및 정량적 분석(FMEA 등) 수행. <u>아키텍처 설계 근거 명시 필요</u>
- HWE.2 : Block diagram 및 schematic diagram 대상으로 정량적/정성적 분석(FMEA 등) 수행. <u>아키텍처 설계 근거 명시 필요</u>

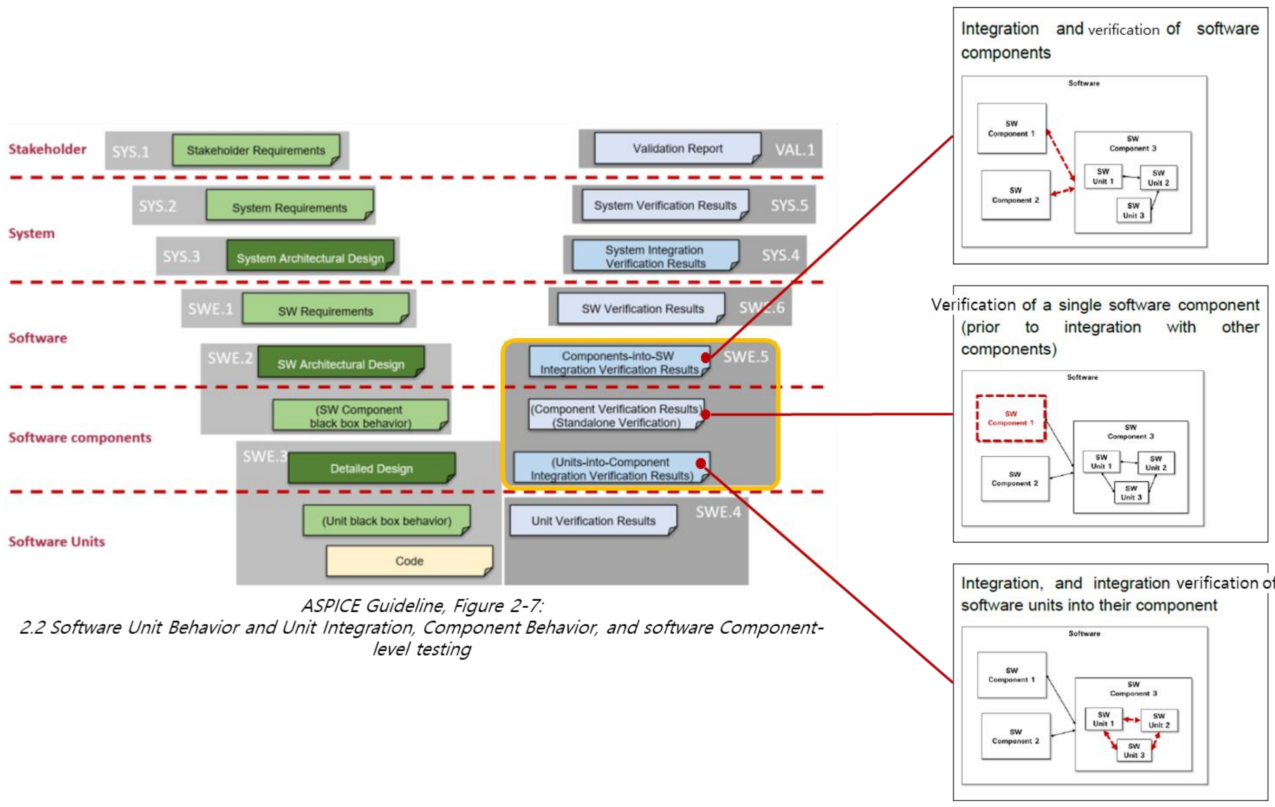
Automotive SPICE v4.0 major changes

Major change 7 SWE.5 단계 세분화

v3.1에서는 integration은 unit → item → integrated SW로 세분화 정의되어 있으나, integration test에 대해서는 세부적인 단계가 정의되어 있지 않았습니다. 또한 traceability에 대해서도 SW architecture와 SW integration test case만 명시되어 있었습니다. 이로 인해 integration test는 component into large elements만 고려하는 것으로 오해가 되는 경우가 발생하였습니다.

이를 해결하기 위해 v4.0에서는 다음과 같은 변화가 있습니다.

- SW integration verification 단계 정의 : unit into verification, SWC into SW integration verification (SWE.5.BP4)
- SWC stand-alone test 신규 추가(SWE.5.BP5)
- 추적/일관성 BP 추가 : SWE.5(integration verification measure)는 SWE.2(SW architecture) 외 SWE.3(SW detailed design)와도 추적 및 일관성 확보



Stakeholder
SYS.1
Stakeholder Requirements
Validation Report
VAL.1
System
SYS.2
System Requirements
System Verification Results
SYS.5
SYS.3
System Architectural Design
System Integration Verification Results
SYS.4
Software
SWE.1
SW Requirements
SW Verification Results
SWE.6
SWE.2
SW Architectural Design
Components-into-SW Integration Verification Results
SWE.5
Software components
(SW Component black box behavior)
(Component Verification Results) (Standalone Verification)
SWE.3
Detailed Design
(Units-into-Component Integration Verification Results)
Software Units
(Unit black box behavior)
Unit Verification Results
SWE.4
Code
Integration and verification of software components
Verification of a single software component (prior to integration with other components)
Integration, and integration verification of software units into their component
Software
SW Component 1
SW Component 2
SW Component 3
SW Unit 1
SW Unit 2
SW Unit 3

ASPICE Guideline, Figure 2-7:
2.2 Software Unit Behavior and Unit Integration, Component Behavior, and software Component-level testing

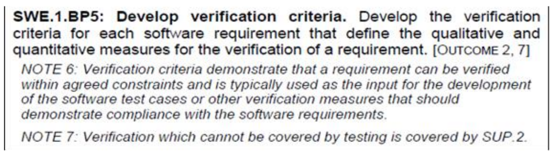
Automotive SPICE v4.0 major changes

Major change 8 Verification criteria BP 삭제

→ But, Requirements의 characteristics로 새롭게 정의

v3.1에서는 요구사항은 반드시 검증 가능하여야 하기 때문에 이를 강조하기 위하여 verification criteria BP가 추가되었습니다. 그러나 BP1(specify)와 BP5(verification criteria) 평가에 일관성 이슈가 발생하는 경우도 존재하며, verification criteria에 대한 별도 문서/정보가 존재해야 한다는 오해도 발생하였습니다. 이를 해결하기 위해 v4.0에서는 BP1(specify)에 요구사항이 갖추어야 하는 기본적인 품질 특성으로 흡수하며 BP1을 강화하였습니다.

【 v3.1 】

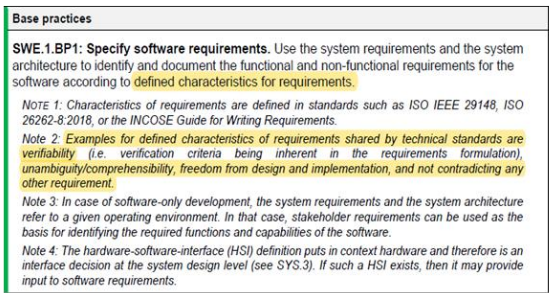


SWE.1.BP5: Develop verification criteria. Develop the verification criteria for each software requirement that define the qualitative and quantitative measures for the verification of a requirement. [OUTCOME 2, 7]

NOTE 6: Verification criteria demonstrate that a requirement can be verified within agreed constraints and is typically used as the input for the development of the software test cases or other verification measures that should demonstrate compliance with the software requirements.

NOTE 7: Verification which cannot be covered by testing is covered by SUP.2.

【 v4.0】



Base practices

SWE.1.BP1: Specify software requirements. Use the system requirements and the system architecture to identify and document the functional and non-functional requirements for the software according to defined characteristics for requirements.

NOTE 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide for Writing Requirements.

Note 2: Examples for defined characteristics of requirements shared by technical standards are verifiability (i.e. verification criteria being inherent in the requirements formulation), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement.

Note 3: In case of software-only development, the system requirements and the system architecture refer to a given operating environment. In that case, stakeholder requirements can be used as the basis for identifying the required functions and capabilities of the software.

Note 4: The hardware-software-interface (HSI) definition puts in context hardware and therefore is an interface decision at the system design level (see SYS.3). If such a HSI exists, then it may provide input to software requirements.

Others 그 외 주요 변경 사항

- 다양한 검증 방법 고려하여 test → verification으로 용어 변경
- Item 용어는 functional safety와 의미가 충돌되어 삭제
- Process의 Output 명칭 변경 : work product → information item
- Note 성격 보완 : requirements/checklist 활용 소지 있는 경우 informative 성격으로 개선