



## HOW TO BECOME AUTOMOTIVE SPICE COMPLIANT

# 자동차에 적용되는 A-SPICE의 효과적인 준수 방안 최신 자동차 소프트웨어 엔지니어링

지난 5월 31일 임베디드 솔루션 전문기업 MDS테크놀로지사가 개최한 ‘자동차 SW 개발자 컨퍼런스 2017’에는 ISO/IEC 15504(SPICE) 독일 의장인 베르트 힌델(Bernd Hindel) 박사가 기조연사로 초대됐다. 힌델 박사는 자동차 SW 개발 프로세스 국제 표준 “Automotive SPICE(이하 A-SPICE)” 인증기관인 iNTACS의 설립자로, SPICE의 국제 표준 수립을 주도한 인물이다.

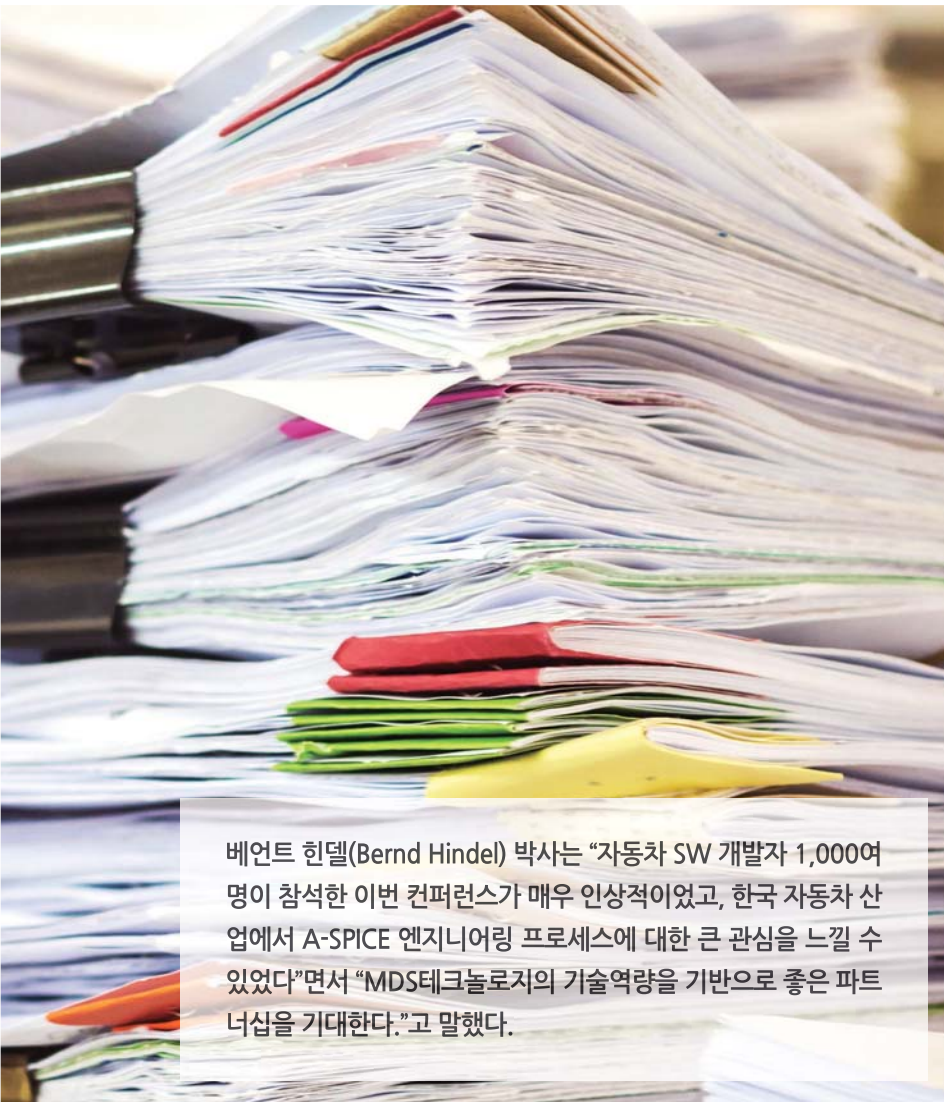
정리 | 한 상 민 기자 \_ han@autoelectronics.co.kr

**차** 량 내 전자제어장치(ECU)에 탑재되는 소프트웨어의 품질 향상을 목적으로 유럽의 주요 완성차 업체가 공동 개발한 A-SPICE의 초판이 발행된 지 10년 만인 지난 2015년에 최신 버전 v3.0 (A-SPICE 3.0)이 릴리즈됐다. A-SPICE 3.0은 “대상 제품

개발 프로젝트의 품질 개선”에 주안점을 두고 있으며, 최근엔 자동차 기능 안전성 국제 표준(ISO 26262)이 요구하는 프로세스 인프라 구축을 목적으로 완성차 업체들이 협력 업체에 A-SPICE 준수를 요구하고 있다.

힌델 박사는 자동차 SW의 복잡성이 증가하고 안전 및 보안 이슈가 강조됨에 따

라 SW 및 시스템 엔지니어링이 점점 더 중요해질 것이라며, ISO 26262, A-SPICE 등 개발 프로세스의 유기적인 통합을 통해 개발 생산성을 높여야 한다고 강조했다. 그는 “자율주행 혁명으로 향후 5년 이내에 자동차 산업의 기존 비즈니스 체계가 허물어지고 제조업체보다 SW 개발업체가 더 높은



베언트 힌델(Bernd Hindel) 박사는 “자동차 SW 개발자 1,000여 명이 참석한 이번 컨퍼런스가 매우 인상적이었고, 한국 자동차 산업에서 A-SPICE 엔지니어링 프로세스에 대한 큰 관심을 느낄 수 있었다”면서 “MDS테크놀로지의 기술역량을 기반으로 좋은 파트너십을 기대한다.”고 말했다.



베언트 힌델(Bernd Hindel) CEO,  
Method Park

- 독일 에를랑겐 대학교 컴퓨터공학 박사
- 독일 에를랑겐 대학교 교수  
(소프트웨어 엔지니어링, 프로젝트 관리)
- 폭스바겐 자동차대학, 아우구스부르크 대학교,  
뉘르츠부르크 대학교 초청 강사
- iNTACS 설립자
- ISO/IEC 15504 (SPICE) 독일 의장

이익을 창출하는 시대가 올 것”이라며 “SW 오류로 인한 사고발생, 해킹 등 안전과 보안 관련 이슈를 해결하는 것이 향후 과제”라고 지적했다. 다음은 힌델 박사의 강연 내용(주제: How to become Automotive SPICE compliant)을 정리하여 소개한다.

## CMMI에서 SPICE, 그리고 A-SPICE로

성숙도 모델(Maturity Models)의 목표는 제품의 품질을 향상시키는 것이다.

“제품의 품질은 결국 개발 프로세스의 품질과 동등하다”는 가설이 있는데, 이는 개발 프로세스가 좋으면 좋은 품질의 제품을 만들 수 있다는 것을 의미한다. 이런 성숙도 모델은 능력 수준(Capability Level)에 의해 정의되며 프로세스 능력 수준은 레벨 0에서 레벨 5까지

6단계로 구분된다. 자동차 산업에서는 보통 레벨 3(Managed) 정도면 적합 판단을 받는다.

성숙도 모델은 관련 모범사례(Best Practice)와 각 레벨에 맞게 측정할 수 있는 측정 프레임워크(Measurement Framework)로 구성된다.

초기 성숙도 모델로는 CMM(Capability Maturity Model)을 들 수 있다. CMM은 카네기멜론대학 부설 연구개발센터인 소프트웨어공학연구소(Software Engineering Institute, SEI)가 미국 국방성의 후원 하에 만든 표준이다. 미국 기업뿐만 아니라 여러 나라 기업들도 소프트웨어 성숙도 모델이 필요하긴 했지만, 라이선스 비용을 미 국방성에 지불하기를 원치 않았다. 결국 1980년대에 소프트웨어가 많이 필요했던 영국군 측에서 CMM이 아닌 새로운 모델을 찾기 시작하면서 유럽 내 독자적인 프로젝트가 시작됐다. 당시 이 프로젝트를 부트스트랩

## 내용 요약

- A-SPICE는 전 세계 자동차 엔지니어링 분야에서 가장 많이 사용하는 개발 프로세스 평가모델이 될 것이다.
- 2015년에 발표된 최신 버전 ‘A-SPICE 3.0’이 올해 의무화될 예정이다. 대부분의 OEM들은 A-SPICE를 사용해 그들의 부품 협력업체(Tier 1)를 평가할 것이다.
- 대부분의 Tier 1은 A-SPICE를 사용해 엔지니어링 프로세스를 개선할 수 있다.

(Bootstrap)이라고 불렀는데 다임러, 보쉬, 알카텔 등이 이 프로젝트에 참여했다.

부트스트랩은 조직의 소프트웨어 개발 및 관리 프로세스의 품질을 평가하고 향상시키는 방법으로, CMM의 방식과 아주 흡사했다. 영국군은 부트스트랩의 표준 작업이 거의 끝나갈 무렵 이 프로젝트에 기반한 ISO 표준을 재정하게 됐다. 즉 부트스트랩이 ISO 15504 TR(기술보고서)이 된 것이다. SPICE라고도 불리는 이 표준은 1995년도에 공식 발표되었는데, 이후 시장에서는 CMMI와 SPICE가 대표적인 성숙도 모델로 쓰이게 됐다.

2000년도로 접어들며 자동차 업계에서는 소프트웨어의 중요성을 인식하게 되었다. OEM들은 대부분의 소프트웨어가 티어(Tier) 1과 티어 2 업체에서 사용하게 될 것을 인지했고, 소프트웨어를 조립 라인에 제공할 때 검토하는 것만으로는 충분치 않다고 판단했다. 협력업체의 전반적인 개발 프로세스를 검토하고, 해당 개발 프로세스가 좋으면 제품도 좋을 것이라는 판단을 하게 된 것이다. 때문에 OEM들은 개발 프로세스 평가 수단으로서 CMMI와 ISO 15504 중 어느 성숙도 모델을 사용해야 할지 결정해야 했다.

하지만 CMMI는 치러야 하는 비용이 만만치 않았다. 라이선스 비용을 SEI 측에 지불해야 했기 때문이다. 결국 비용 문제로 독일의 주요 자동차 업체인 다임러, BMW, 포르쉐, 아우디, 폭스바겐 등은 SPICE를 평가 모델로 도입하기로 결정했다.

더욱이 CMMI는 미 국방성이 주도했기에 모델에서 무엇인가를 바꾸려면 승인을 받아야 했다. CMMI 기반으로는 업계에 특화된 프로세스 모델로 개발할 수 없었던 것이다. 자동차 산업은 업계에서 사용하는 용어를 사용하기 원했고, 그 용어가 모델에 반영되기를 원했다. CMMI는 이를 허용하지 않은 반면, ISO에서는 그렇게 할 수 있는 가능성이 있었다. 예를 들어 ISO 15504의 경우, 관심 분야별로 프로세스 평가 모델을 자체 용

어로 구성할 수 있었다. SPICE에 여러 개의 버전이 있는 것은 이러한 이유에서다. 현재 우주공학, 금융, 의료 분야 등의 SPICE가 있는데, 각각의 분야별로 해당 SPICE 버전을 만들 수 있음을 보여준다.

아우디, BMW, 다임러, 포르쉐, 폭스바겐이 참여하고 있는 HIS 그룹은 2003년도에 SPICE를 사용하기로 결정했다. A-SPICE 1.0 버전은 2005년에 발표했는데, 이것이 바로 ISO 15504이다. ISO 표준의 경우 5년마다 새로 갱신되는데, ISO 15504의 새로운 버전인 ISO 33000이 2015년에 발표됐다. A-SPICE도 여기에 맞춰야 했기에 새로운 A-SPICE 3.0을 제정했다.

A-SPICE 3.0은 ISO 33000에 기반했으며, 독일자동차협회(Verband Der Automobilindustrie, VDA)에 의해 제정됐다. 배포는 워킹그룹 13(AK 13)에서 맡고 있다.

## “학습”이 중요하다

### Basics & Abstractions

A-SPICE나 SPICE는 반드시 사용해야 하는 규제가 아니라 프로세스 성숙도 평가 모델이라는 점을 기억할 필요가 있다. 성숙도 모델은 “개선”을 중요시 여기

며, SPICE(Software process Improvement and Capability Evaluation)에서의 “I” 역시 “Improvement”의 약자다.

A-SPICE는 구입해서 쓸 수 있는 프로세스가 아니다. 여러 모범 사례와 이를 어떻게 배워야 하는지에 대한 아이디어이다. 즉 A-SPICE는 프로세스 개선을 위한 학습이 주된 내용이며, 관련된 모범사례를 취할 때 학습할 내용이 무엇인지를 정의해야 한다. 한 마디로 학습을 소프트웨어 엔지니어링 분야에 적용한 것이라고 할 수 있다.

A-SPICE에는 Basics와 Abstractions가 들어가 있는데, 이것을 이해하면 A-SPICE를 준수하는 프로세스를 마련할 수 있다.

A-SPICE의 가장 큰 장점 중 하나는 요구사항 이행을 증명하기 위한 추적성(Traceability)이다. A-SPICE는 추적성 측면에서 CMMI보다 훨씬 더 명확한 요구사항을 제시한다. 각 고객의 요구사항에 대한 추적성을 유지하기 위해서는 필요한 정보를 수집하고 문서화해야 하는데, 여기에 테스트를 위한 요구사항이 포함되는 것은 아니다. 추적성이 제대로 확보될 경우 시간을 많이 절약할 수 있다. 물론 엔지니어링 톨로 추적성을 확보하기란 쉽지 않지만, 일단 확보하면 평균 개발 엔지니어링 속도가 빨라지고 전체 프로젝트에서 무슨 일이 일어났는지도 쉽게 알 수 있게

표 1 | CMMI와 Automotive SPICE 개념의 맵핑

CMMI	Automotive SPICE
Process Area	Process
Purpose	Process Purpose
Specific Goals	Process Outcomes
Specific Practices	Base Practices
Subpractices	-
Typical Work Products	Output Work Products
-	Work Product Characteristics
Generic Goals	Process Attributes
Generic Practice	Generic Practices
Generic Practice Elaborations	-
-	Generic Resources
Examples	-
Amplifications	-
Capability Levels	Capability Levels

된다. 여기서 “알 수 있다”는 것은 결국 학습을 통해 배운다는 것을 의미한다.

엔지니어링에는 여러 가지 기술을 사용할 수 있기 때문에 가이드라인(Guidelines)이 필요하다. 어떤 기술이나 방법, 툴을 사용하든지 일정 수준 이상의 품질이 담보될 수 있게끔 하기 위해서다.

또한 프로젝트에서 의사결정을 내려야 하는 시점에 결정을 도와주는 전략(Strategies)이 필요하다. 어떻게 보면 이것이 엔지니어링 Practices인 것이다.

그렇다고 A-SPICE가 요구사항에 대한 추적성만 요구하는 것은 아니다. IBM Rational DOORS나 PTC Integrity와 같은 툴이 추적성을 확보할 수 있게끔 도와주지만, 이들은 실질적으로 전체 요구사항의 50% 정도 밖에 커버하지 못한다. A-SPICE에서는 워크 패키지(Work Packages)에 대한 추적성을 요구하고 있다. 한 마디로 프로젝트 계획을 수립하고 모니터링까지도 포함돼야 한다는 것이다.

우선, 요구사항의 추적을 보자. 요구사항에 대한 추적성이 있어야 한다는 것은 요구사항의 전반적인 라이프사이클을 다 추적해야 한다는 것을 의미한다. 요구사항을 먼저 캡처하고, 분석하고, 디자인을 추출하고, 이를 구현한 후 테스트해야 할 것이다.

요구사항을 분석하는 것은 A-SPICE에서 아주 중요한 사안인데, 분석해야 하는 속성까지도 정의하고 있다. 요구사항의 타당성(feasibility), 테스트 가능성(testability), 요구사항이 미치는 영향(impact), 중요도(criticality), 요구사항을 이행하는 데 드는 비용 추정(estimation)이 포함된다.

워크 패키지에 대한 추적성도 있어야 한다. 이것은 워크 패키지의 라이프사이클을 봐야 한다는 것인데, 이를 위해선 프로젝트 초반에 워크 패키지가 준비돼야 한다. 그 다음 비용에 대한 예측이 있어야 하고 캘린더에 적기 위한 스케줄링이 있어야 한다. 이후 실행과 검토가 이루어져야 한다. 워크 패키

지에 대한 비용 추정과 스케줄링을 다시 해야 하는 경우도 있다. 바로 이러한 워크 패키지가 프로젝트 매니저(PM) 쪽에서 이뤄지고 있는데, 워크 패키지에 대한 추적성이 있어야 한다. 이 분야의 추적을 하게 되면, 많은 것을 배울 수 있다. 엔지니어가 워크 패키지에서 비용 추정을 하고 워크 패키지를 이행하게 되면 처음 예산보다 얼마나 벗어났는지를 알 수 있다. 또한 자신이 얼마만큼 달성할 수 있는지 스스로 알 수 있는데, 여기서 다시 한 번 “학습”의 중요성을 알 수 있다.

이제 앞서 언급한 가이드라인에 관해 얘기해보도록 하겠다. 가이드라인은 방법(methods)이나 툴(tools)을 위해 필요한 것인데, 이것이 A-SPICE의 요구사항이다. ISO 26262 역시 준수를 위한 가이드라인이 필요한데, 안전과 관련된 문화(Safety culture)는 어떻게 되는지, 달성 가능한지에 대한 내용이 포함돼야 한다.

그 다음 전략(Strategies)에 기초한 의사결정이 이뤄지는데, 사실 추정(estimation) 측면에서도 전략적 의사결정이 이뤄져야 한다. 비용뿐만 아니라, 여러 가지 엔지니어링과 관련된 추정을 함에 있어서도 여러 가지 테크닉이 있는데, 그 테크닉을 어떻게 선택하고 나온 추정치를 어떻게 활용할 것인지에 대한 전략도 있어야 한다.

검토(Review) 전략도 있어야 한다. 검토할 수 있는 방법에도 여러 가지가 있을뿐더러, 각 산출물에 대해 어떤 검토를 할 것인가에 대한 전략이 있어야 하기 때문이다.

알다시피 100% 정확하게 테스트 할 수 있는 방법은 없다. 그렇다면 테스트와 관련된 전략, 화이트박스로 할 것인지 블랙박스로 할 것인지, 아니면 좀 더 세부적인 테스트를 할 것인지에 대한 전략이 필요하다. SPICE에서는 “100% 정확할 것 같지 않으면 그에 대비한 전략이 있어야 한다.”고 언급하고 있다. A-SPICE에서 모든 엔지니어링 프로세스에 대한 전략이 필요한 것은 이러한 이유에서이다.

PDCA(Plan-Do-Check-Act)는 개선을 위한 것이다. 개발자는 우선 스스로 목표를 설정해야 한다. 자신이 프로젝트를 통해서 향상되고 있는지 어떻게 알 수 있을까? 그리고 프로젝트 매니저가 다음 프로젝트에서는 더 잘 할 것인지 어떻게 알 수 있을까? 이것을 알아야만 개발팀의 전반적인 개선을 이뤄낼 수 있다.

또 소프트웨어 구조가 스스로 개선이 됐는지, 자신의 구조가 어떻게 개선될 수 있는지 알 수 있을까? A-SPICE에서는 이런 목표와 전략, 그리고 목표를 달성하기 위한 계획, 이를 실행할 수 있는 실행안, 목표들을 모니터링하고 후속 조치할 수 있는, 또 에스컬레이션(escalation) 할 수 있는 부분들이 있어야 한다고 요구하고 있다. 이것도 결국 학습과 관련된 것이다. A-SPICE의 모든 프로세스별로 이것이 다 들어가 있다. A-SPICE를 제정할 당시, 이런 부분이 기본이 되어야 한다고 생각하고 A-SPICE를 제정한 것이다.

## 후진이 아닌 지속적인 발전을 위해

### ISO/IEC 330xx = A-SPICE 3.0

A-SPICE의 프로세스 능력 수준은 레벨 0부터 5까지 6단계로 구성된다(그림 1). 여기서 먼저 짚고 넘어 가야 할 단계가 레벨 2이다. 레벨 2는 정의(define)하고 검토(review)하는 단계이다. 이는 프로세스에 의해 생성된 작업 산출물이 적절하게 관리되는 정도를 측정하는 것으로, 일반적으로 작업 산출물에 대한 요구사항과 문서화 및 컨트롤을 위한 요구사항을 정의하며 이를 만족시키기 위해 작업 산출물을 검토하고 조정한다.

레벨 3에서는 표준 프로세스가 존재하는지, 해당 프로세스가 맞춤화와 피드백을 통해 개선될 수 있는지를 확인한다.

A-SPICE는 하나의 표준화된 프로세스를 선호하지 않는다. 사내에 어떤 표준화된 프로세스가 있어서 변경이 불가능하다면 이는 A-SPICE의 원칙에 어긋난다. A-SPICE는 프로세스에 대해 끊임없이 의문을 갖기를 원하며, 프로세스를 프로젝트에 맞게끔 맞춤형하고 사후에는 해당 프로세스가 프로젝트에 적합했는지 피드백해주어야 한다. 프로세스의 어떤 부분이 개선될 수 있는지도 알려주어야 한다.

레벨 4는 추정(estimation)과 측정(measurement)의 단계인데, 현재 자동차 OEM 들은 레벨 4를 요구하지 않는다. 대부분의 기업들은 안전을 위한 프로세스나 보안을 위한 프로세스는 레벨 3을 요구한다. 하지만 향후에는 레벨 4까지 적용될 것으로 예상된다.

### A-SPICE 프로세스 영역

ISO/IEC 330xx 시리즈는 거의 60개의 프로세스 영역(process areas)으로 구성된다. A-SPICE는 31개의 프로세스 영역이 있다. 이전 버전이 소프트웨어 계열만 검토했다면 2015년 7월 15일 발표된 A-SPICE 3.0에서는 시스템을 구성하는 메커니컬, 하드웨어, 소프트웨어에 대한 상위 개념으로 시스템 엔지니어링 그룹을 배치했다. 미래에는 A-SPICE 평가를 하드웨어 개발과 메커니컬 개발 쪽에서도 할 수 있게 될 것이다. 그림 2에서 빨간 테두리로 표시된 부분이 HIS 그룹에서 권장하는 것들이다.

요구사항에 대해 평가를 하려면, 프로세스별로 두세 시간의 인터뷰가 필요하다. A-SPICE의 경우 31개 영역에 대해 모두 인터뷰를 하기 때문에 많은 비용이 소요될 것이다. 이에 대해 HIS 그룹은 15개 영역을 선정했고, HIS 범위(scope)에 기초해 15개의 프로세스 영역을 평가(assessment)하는데 1주일이면 충분하게 됐다.

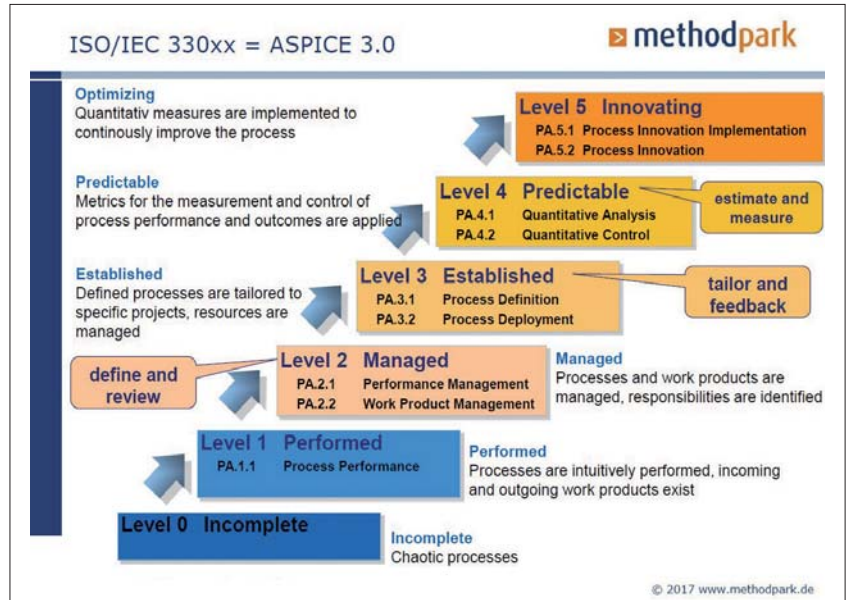


그림 1 | ISO/IEC 330xx = A-SPICE 3.0

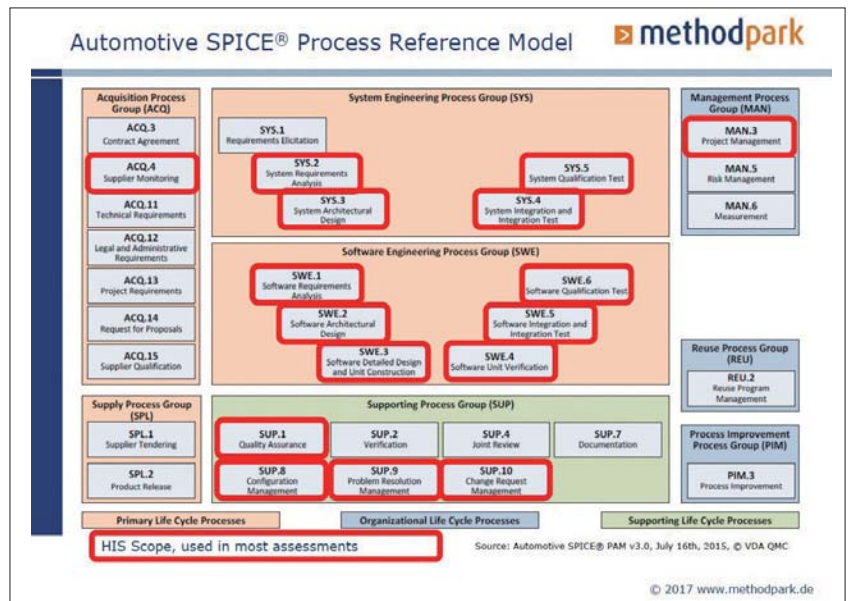


그림 2 | Automotive SPICE 프로세스 참조 모델

레벨 1에서는 양방향 추적성(bidirectional traceability)과 일관성(consistency)이 중요하다(그림 3). SPICE가 90년대에 만들어졌기 때문에, A-SPICE의 근간은 V-cycle이다. V-cycle은 소프트웨어 엔지니어링 프로세스(SWE)와 시스템 엔지니어링 프로세스(SYS)로 구성돼있다. 이 추적성은 V-cycle에 기초해 정의되는데, 단계마다 추적성이 있어야

한다. 추적성은 프로젝트를 보다 잘 이해할 수 있도록 도와주는 기초이자 효율적인 엔지니어링을 위해서도 필요하기 때문이다.

만약 프로젝트에 변경 요청을 하려면, 추적성이 확보되어야만 제대로 디자인을 바꾸고 요구사항을 바꿀 수 있다.

레벨 1은 추적성이 없으면 불가능하다. 레벨 1의 50% 정도가 결국은 추적성과 일관

성에 관련된 것이기 때문인데, 따라서 추적이 없다면 레벨 1 평가는 생각할 수 없다.

레벨 2에서는 개선을 위한 목표를 설정한다. 프로젝트 관리자로서 개선하기 위해 무엇을 해야 하고, 개발자로서 스스로 어떻게 개선할 수 있는지 목표를 설정하는 것이다. Part 2.1에서는 워크 패키지를 계획 및 모니터링하거나 성과 관리, 역할 분담, 인력 배치 시에도 목표 설정하는 것을 다룬다. Part 2.2는 템플릿과 형상(Configuration) 관리, 검토와 관련된 것이다.

레벨 3의 경우에는 프로세스를 정의하고 이를 맞춤형 방식으로 사용하는 것이다. 여기에는 추정과 모니터링, 형상 관리와 검토가 들어간다. 또한 표준 프로세스와 피드백, 프로세스 맞춤화, 팀원의 스킬 관리 등도 포함된다.

다시 정리하면, 레벨 1은 어떤 소프트웨어 엔지니어링에서의 장인정신(Craftsmanship)이라고 할 수 있다. 추가적으로 배울 것은 없으며 대학에서 배운 정도면 충분하다. 레벨 2는 프로젝트를 개선하기 위한 자세라고 할 수 있다. 형상 관리 후 검토하고 템플릿을 준수하며 이미 있는 것을 재사용하고자 하는 자세를 가져야 한다는 것이다. 레벨 3은 잘된 점을 기억하고 이를 다시 활용하는 자세이다. 때문에 레벨 3에서는 경험을 문서화해 이것을 프로세스로 정착시키는 것이다. 간단하게 보이지만, 이러한 성숙도 단계까지 올라가서 적용하기는 쉽지 않다. 담당자가 기술적인 부분에 대해 이해하고, 개선을 위한 자세를 갖추고, 좋은 프로세스가 무엇인지를 파악해서 이것을 재사용하게끔 하는데 적은 시간이 소요되는 것이 아니기 때문이다.

## A-SPICE의 도입

그림 6단계로 구성된 A-SPICE를 단계적으로 어떻게 도입할 것인가? 우선 전

반적인 비즈니스의 목표와 목적을 파악해야 한다. 고객이 레벨 3 준수를 요구할 때면 이미 늦었다. 그 전에 먼저 시작했어야 하고 항상 목표도 있어야 한다. 현업에 엔지니어링을 개선해 시간을 절감하고, 현장에서 오류 수를 줄이겠다는 목표가 있어야 한다. 이런 목표가 있어야만 프로세스 모델 도입 시 도움이 되는 것이다.

경영진과 개발자들이 참여하는 워크숍을 통해 어떤 목표를 세워야 하고, 개발자 입장에서 무엇을 개선해야 하는지 파악해야 한다. 워크숍 등을 통해 목표를 정의한 후에는 갭 분석(Gap analyses)을 해야 한다. 이를 위해 현재 사용하고 있는 프로세스를 먼저 분석해야 하는데, 엑셀이든 PPT이든 보유하고 있는 자체 프로세스를 A-SPICE를 통해 평가하면 된다. 즉 기존의 것과 비교를 하라는 것인데, 어떤 차이가 있는지 파악하고, 우선순위를 정해 전에 정했던 목표를 달성할 수 있는지 확인해야 한다.

그 다음은 프로세스 구조(Architecture)를 정의해야 한다. 회사마다 필요한 구조가 다를 텐데, 구조에는 인트라 프로세스(Intra process) 구조와 인터 프로세스(Inter process) 구조 두 가지가 있다. 건물에 비유하자면, 인트라 프로세스 구조는 건설되고 있는 하나의 건물이고, 인터 프로세스 구조는 여러 건물이 있는 도시라고 보면 된다. 인트라 프로세스 구조에는 역할 정의(Role descriptions), 템플릿(Templates), 툴(Tools), 라이프사이클(Lifecycles), 워크 인스트럭션(Work instructions), 체크리스트(Checklists) 등이 있다. 이런 용어들은 프로세스를 이야기할 때 흔히 사용되며, 이것을 메타 모델(Meta Model)이라고도 부른다.

각 기업마다 사용하는 용어가 다르기 때문에, 프로세스를 정의하기 위해서는 먼저 개발자들이 어떤 용어를 사용하는지 파악하는 것이 중요하다.

이제 추상화(abstraction)를 해야 한

다. 프로세스 자산(Process assets) 유형 혹은 프로세스 엘리먼트(process element) 유형들을 추상화해야 하는데, 이러한 프로세스 엘리먼트 유형들이 나중에 프로세스에 사용 되는 것이다.

메타 모델은 그림 4와 같은 모양이 될 수 있다. 엔지니어링 툴이 있고, 역할 정의가 있고, 페이즈(Phase)와 마일스톤(milestone)이 있다. 또 프로세스와 활동 그룹(Activity Group)이 있고 의사결정(Decision)과 활동(Activity)이 있다. 이러한 구조는 기업마다 있으며, 모든 사람들이 이것을 다 이해하고 난 후에야 프로세스를 만들 수 있다.

다음은 인터 프로세스 구조다. 이 구조에는 프로세스가 여러 개 들어가게 되는데, 영역별로 구분 지어져 있다. 매니지먼트 프로세스(전략 관리, 비즈니스 플랜, 기업 지배구조), 밸류 프로세스(연구, 제품 개발, 제조, 영업 및 마케팅, 서비스 및 지원), 지원 프로세스(금융, 인적자원, 정보기술, 구매, 법률)가 있을 것이다. 밸류 프로세스는 기업마다 다른데, 이러한 프로세스들은 보통 계층(hierarchy)이 있다. 제품 개발은 시스템 엔지니어링, 하드웨어 엔지니어링, 소프트웨어 엔지니어링, 품질관리(QM)로 세분화될 것이다. 이것이 인터 프로세스 구조로서 사전에 정의해야 하는 것이다.

이런 구조를 모두 정의한 후에는 조직을 정의해야 한다. 누가 프로세스를 바꿀 수 있는지, 프로세스 오너, 프로세스 작성자, 프로세스 아키텍트, 프로세스 디자이너 등이 누구인지 정의해야 한다. 일반적인 소프트웨어 프로젝트와 비슷하다고 할 수 있다. 팀이 구성되어 프로세스를 정의하고 변화를 관리할 수 있도록 해야 한다. 이렇게 먼저 조직이 마련되고 나서 작업을 시작해야 한다.

그 다음에 코치들이 있어야 한다. 코치들은 정립된 프로세스를 이해하고 프로젝트 팀원들이 프로세스를 준수하면서 작업을

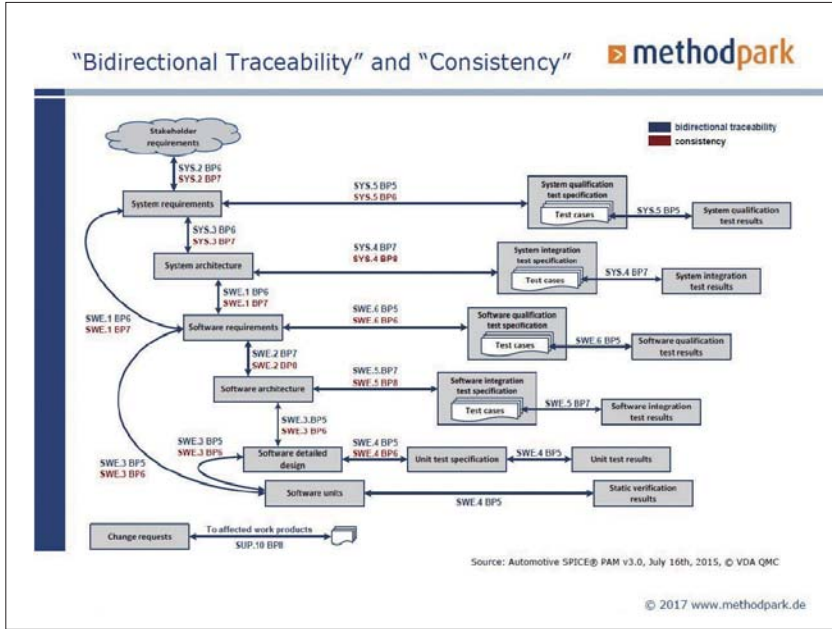


그림 3 | 양방향 추적성 및 일관성 (출처: Automotive SPICE® PAM v3.0, July 16th, 2015, © VDA QMC)

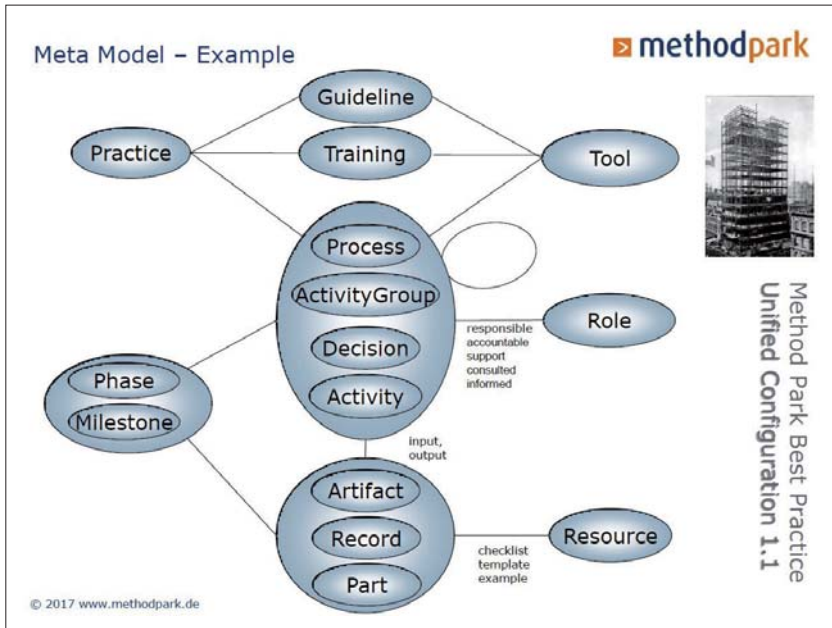


그림 4 | Meta Model - Example

할 수 있게끔 해야 한다. 프로젝트 팀마다 한 명의 코치를 둘 것을 권장한다.

그 다음에 프로세스에 대한 모델링 작업을 할 수 있다. 지금까지 용어를 정의하고, 메타 모델을 만든 후, 프로세스 영역과 프로세스 구조를 정의했다. 그 다음 조직을

만들었고 역할(role)과 메소드 설명(Method Descriptions)을 했다. 그 다음은 프로세스 영역별로 정의를 하는데, 프로세스 개요(Process Overview)를 마련하고 필요한 프로세스 영역별로 필요한 활동(Activities), 산출물과 역할을 정의한 다음에 검토(Review)

를 해야 한다. 그 다음에는 템플릿, 체크리스트, 문서의 라이프사이클이 정의되고 리뷰되고 있는지 파악해야 한다.

그 다음에 컴플라이언스(Compliance)를 확인해야 한다. 그 다음 갭 분석을 하게 된다. 이후 테일러링 가이드라인(Tailoring Guidelines)도 정의해야 한다. 레벨 3부터는 테일러링 가이드라인이 있어서 향후 프로세스를 개선할 수 있기 때문이다. 그 방법이 바로 메소드와 툴들이 되겠다. 여기에는 바로 메소드 설명과 툴 설명을 마련해 프로세스에 포함시키고, 트레이닝(Trainings) 절차가 마련되어 팀이 새로운 프로세스에 대해 교육을 받고, 코치가 정해져 무슨 일을 해야 하는지 결정해야 한다.

마지막에는 측정(Measurement)이 포함되어야 한다. 측정은 프로세스 자체에 들어가 그 목표를 달성했는지 여부를 파악할 수 있도록 하는 것이다. 기존 프로젝트보다 개선됐는지 여부를 파악하기 위해 프로젝트 매니저가 사용하는 기준 중 하나가 측정치를 맞췄는지 여부를 보는 것이다.

이러한 과정을 거치게 되면, 향후 프로세스를 전사적으로 전개할 때 하나의 프로젝트를 파일럿 프로젝트로 진행해, 성공적이면 다른 프로젝트에 적용해 코치와 함께 전체적으로 적용할 수 있게 되는 것이다. 측정치들에 대한 평가는 이후 이루어지며, 워크숍에서 정했던 목표가 달성됐는지를 볼 수 있다. 이렇게 함으로써 A-SPICE의 궁극적인 목적인 학습조직이 생기게 되는 것이다.

정리하자면, 먼저 A-SPICE와 ISO 26262를 이해해야 한다. 그리고 AUTOSAR와 모델 기반 설계 툴과 같은 메소드가 필요하다. 또 DOORS나 Integrity와 같은 툴도 필요하다. 무엇보다도 사람이 이런 부분들에 대해 충분히 이해하고 효율적으로 일할 수 있어야 한다. 모든 과정에는 시행착오가 있기 마련이다. A-SPICE는 퇴보가 아닌 지속적인 발전을 위한 것이다. AE