



**SIEMENS**

Ingenuity for life



Siemens Digital Industries Software

# 차량용 임베디드 애플리케이션 개발 조정

애플리케이션 정의 및 계획

## 개요

ADAS (Advanced Driver Assistance System), 커넥티드 카 기능 등 정교한 전자 장치 및 임베디드 소프트웨어를 활용하는 고급 기능에 대한 수요가 꾸준히 증가하면서 자동차 산업은 급격하게 변화하고 있습니다. 임베디드 애플리케이션 개발은 끊임없이 변화하는 구속조건 하에서 다양한 톨과 팀이 얽혀 협업을 시도해야 하는 미로 같은 작업입니다. 통합 소프트웨어 애플리케이션 엔지니어링 플랫폼은 구속조건과 요구 사항을 완전하고 효과적으로 전달하고 개발 작업이 요구 사항을 정확하게 충족하도록 지원합니다.

Piyush Karkare

# 소개

자동차 산업의 환경은 매우 역동적이고 빠르게 변화합니다. ADAS(Advanced Driver Assistance System), 커넥티드 카 기능 등 정교한 전자 장치 및 임베디드 소프트웨어를 활용하는 고급 기능에 대한 수요가 계속 증가하면서 이러한 변화를 이끌고 있습니다. 이러한 수요는 자동차 산업의 핵심 역량이 기계적 시스템에서 소프트웨어 및 전자 장치로 옮겨가고 있음을 보여줍니다. 경쟁이 치열한 자동차 시장에서 차별화와 혁신은 기계적 시스템이 아닌 소프트웨어와 전자 장치에 점점 더 많이 의존하고 있습니다.

OEM과 공급업체는 차량 라인업 전반에 걸쳐 소프트웨어 기능을 표준화함으로써 최신 차량의 디지털 혁신에 대응하고 있습니다. 폭스바겐 (Volkswagen)은 최근 브랜드 전반에 걸쳐 일관된 운영 체제를 개발할 5천 명의 소프트웨어 전문가로 구성된 새로운 소프트웨어 그룹에 대한 계획을 발표했습니다 (Automotive News Europe, 2019).

이러한 발전 속에서 웨이모 (Waymo), 애플 (Apple)과 같은 기술 선두 기업이 자동차 및 모빌리티 시장에 진입하고 있습니다. 이들은 강력한 소프트웨어 기능과 애플리케이션, 사용자 경험을 설계하고 구축하는 데 정통한 기업입니다. 동시에 수많은 자동차 및 모빌리티 관련 스타트업도 생겨나고 있으며, 매력적인 자동차 플랫폼을 구축하기 위해 처음부터 기술과 자동차 경험의 통합을 모색하고 있습니다. 두 경우 모두에서 차량 소프트웨어 및 전자 장치의 중요성이 커짐에 따라 기존 자동차 브랜드와의 경쟁이 평준화되었습니다.

이와 같은 새로운 과제 외에 기존 자동차 브랜드 간의 경쟁 또한 그 어느 때보다 치열합니다. 2001년에는 소형 크로스오버 SUV 시장에 속한 포드 이스케이프(Ford Escape)의 직접적인 경쟁 모델은 6개 남짓이었습니다. 오늘날 이 부문에서는 주류에서 하이엔드 럭셔리에 이르기까지 21개가 넘는 제조업체가 경쟁하고 있습니다. 경쟁은 더 많은 차량 콘텐츠 (새로운 기능)에 대한 수요를 유발하고, 더 많은 콘텐츠는 복잡성을 높입니다.

이러한 복잡성은 전기 및 전자 (E/E), 소프트웨어 영역에서 압도적으로 두드러집니다. 이 사실은 이러한 영역에 존재하는 요구 사항과 사양의 수를 통해 확인할 수 있는데, 이는 다른 부문에 비해 월등히 많습니다 (그림 1). 예를 들어, 우수한 기능의 중급 차량에는 약 50,000개의 기계

및 규제 요구 사항이 있습니다. 한편, 동일한 차량의 E/E 및 소프트웨어 영역의 경우 요구 사항이 450,000개 이상입니다. E/E와 소프트웨어의 검증 및 테스트 계획은 여러 번에 걸쳐 추적해야 하는 항목의 수를 배가시킵니다.

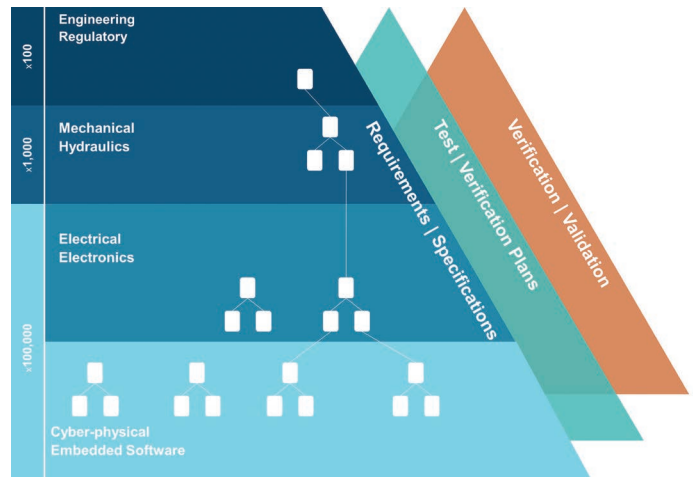


그림 1: 소프트웨어 및 E/E 영역에 대한 요구 사항과 사양은 다른 영역에 비해 훨씬 복잡하고 다양합니다.

E/E 및 소프트웨어 영역 내의 복잡성도 상당하지만, 영역 간에 요구 사항을 계단식으로 적용하고 시행하는 일은 훨씬 더 복잡할 수 있습니다. E/E, 소프트웨어, 기계 영역은 상호 작용 지점이 많으며 매우 상호 의존적입니다. 요구 사항에는 예측은 차치하고 이해조차 상당히 어려운 영역 간 관계가 존재하는 경우가 많습니다. 그러나 팀에서는 각 설계 요구 사항과 이러한 요구 사항이 더 큰 시스템에 미치는 영향을 파악해야 합니다.

OEM과 공급업체가 프로세스와 가치 사슬을 구축하려고 노력하는 와중에도 설계 변경과 제품 진화는 지속적으로 일어납니다. 중요한 것은 영역 전반에 걸쳐 관련 데이터 간의 일관성과 품질을 유지하는 능력입니다. 이를 위해서는 다양한 이해 관계자가 여러 플랫폼과 조직에서 수행하는 작업 간에 전반적인 일관성과 호환성을 보장하는 것이 핵심입니다. OEM 및 공급업체는 차량 플랫폼 수준 기능에서 논리적 할당 및 실제 소프트웨어 애플리케이션

개발에 이르는 각 수준의 설계 추상화를 통해 작업을 조율할 수 있어야 합니다.

OEM은 서비스 중심 아키텍처로 전환하고 있습니다. 이러한 전환의 특징으로 ECU 하드웨어를 더 적은 수의 도메인 관리 유닛으로 통합하는 것을 꼽을 수 있으며, 서비스와 기능이 ECU 추상화에 대해 포팅된 소프트웨어를 통해 수행됩니다. 이러한 아키텍처는 애플리케이션 수준에서 적절한 분해가 이루어지도록 각 서비스가 실행되는 위치를 추적하는 작업에 상당한 신중함을 더합니다. 서비스로 분해되고 ECU 추상화로 매핑되는 기능-함수 아키텍처는 변경 사항과 효과를 추적하기 위한 효과적이고 유기적이면서 확장 가능한 접근 방식입니다.

이러한 팀 또는 추상화 간의 활동 조율이 중요한 과제일 수 있지만 임베디드 애플리케이션 개발의 구성은 특히 어렵습니다. 차량용 임베디드 애플리케이션 개발은 끊임없이 변화하는 구속조건 하에서 다양한 톨과 팀이 얽혀 협업을 시도해야 하는 미로 같은 작업입니다. 이러한 환경으로 인해 협업이 어려워지고, 그 결과 문제가 발생하곤 합니다. 오늘날 소프트웨어 시스템은 소프트웨어 애플리케이션의 복잡성과 중요성으로 인해 프로그램 위험을 유발하는 주된 원인으로 손꼽힙니다. 소프트웨어 결함과 구속조건이 잦은 변경으로 인해 생산 지연, 예산 초과, 보증 비용 증가 등이 발생합니다.

이 백서의 목적에 따라 차량용 임베디드 소프트웨어 애플리케이션은 다음과 같은 세 가지 부문으로 나눌 수 있습니다.

- 임베디드 애플리케이션 정의 및 계획
- 임베디드 애플리케이션 개발 및 품질 보증
- 임베디드 애플리케이션 제공 및 모니터링

이 백서에서는 첫 번째 부문인 임베디드 애플리케이션의 정의 및 계획에 초점을 맞춥니다.

# 애플리케이션 정의 및 계획

일부 OEM은 차량 내 소프트웨어에 대해 기능 중심적 아키텍처 기반 방식을 성공적으로 구축했습니다. 차량 수준의 기능 요구 사항, 사양, 구속조건이 완전히 정의되고 소프트웨어 애플리케이션 추상화에 계단식으로 적용될 경우, 이러한 방식을 통해 애플리케이션 정의 및 계획이 시작됩니다. 이러한 요구 사항, 사양, 구속조건에 따라 차량 플랫폼 또는 차량 수준의 기능 설계 의도가 정의됩니다 (그림 2). 예컨대 차량 엔지니어는 차선 유지 지원, 어댑티브 크루즈 컨트롤, 전방 충돌 경고 및 완화 등의 ADAS 기능을 지원하는 차량 기능에 대한 사양 및 구속조건을 정의합니다.

이어서 소프트웨어 엔지니어는 기능 수준의 정의를 애플리케이션의 요구 사항과 정의로 분해하기 시작합니다. 이로 인해 특정 소프트웨어 기능은 다음 단계 분해를 촉진합니다. 실제로 소프트웨어 기능 분해와 차량 플랫폼 전반에 걸친 표준화는 개발 과정에서 꾸준히 진행됩니다 (그림 3). 소프트웨어 엔지니어는 높은 수준의 요구 사항에서 끊임없이 기능을 분해해 차량의 기능 모음을 구현합니다. 엔지니어가 이러한 기능을 개발하면 기능 균일성이 극대화되고, 기능적 상호 작용이

SWF - 고객/엔지니어링 특징  
 SWf - 분해된 소프트웨어 기능  
 HWf - 분해된 하드웨어 기능  
 LC - 논리적 구성 요소  
 PC - 물리적 구성 요소

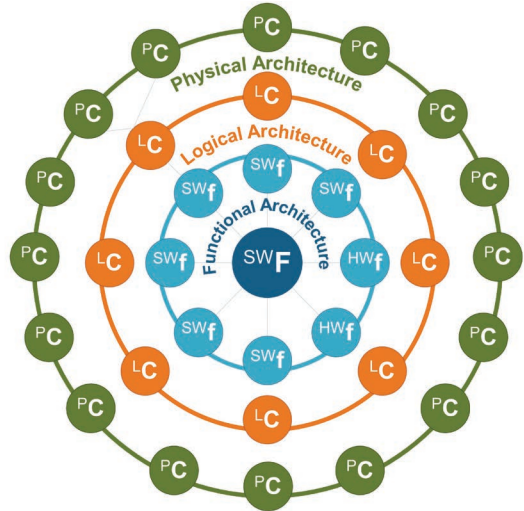


그림 2: 소프트웨어 특징은 소프트웨어 기능으로 분해되고 논리적 구성 요소에 할당되며, 부품 번호에 따라 물리적 구성 요소에 매핑됩니다.

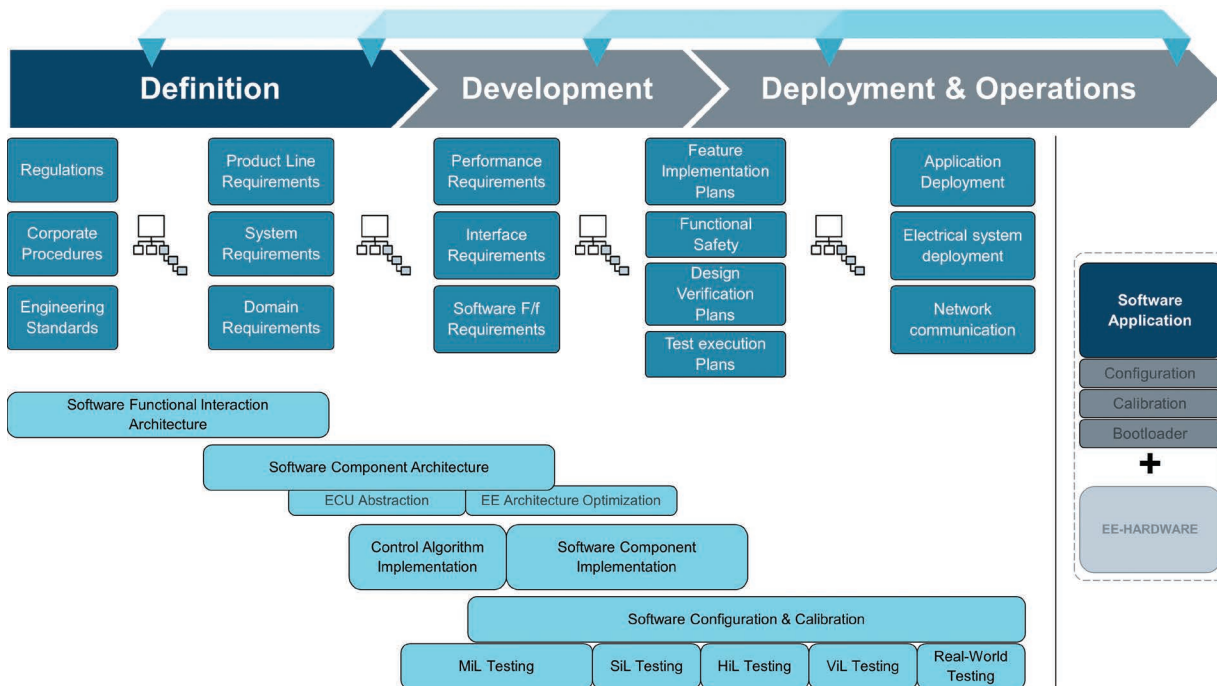


그림 3: 요구 사항과 구속조건을 기능적 분해는 개발 내내 이루어지는 지속적인 과정입니다.



최적화되므로 차량 플랫폼 전반으로 기능을 확장할 수 있습니다. 엔지니어는 이 방식을 활용할 수 있는 영역에서 다른 시스템이나 조직의 기능을 재사용할 수도 있습니다.

이 단계에서는 애플리케이션의 안전 개념을 고려하고 위험 관리 전략을 개발하는 것도 중요합니다. 강력한 안전 요구 사항과 검증 및 확인 전략은 시스템 수준에서 정의되며, 임베디드 소프트웨어 애플리케이션 개발 수준까지 분해됩니다. 소프트웨어 엔지니어는 실패 모드, 영향 및 진단 분석 (FMEA/FMEDA), 위험 요소 분석 및 위험 평가 (HARA), 기타 안전 요구 사항 및 분석을 개발합니다. 엔지니어는 ISO 26262 지침을 충족하도록 이러한 여러 프로세스를 문서화하고 추적합니다.

**주요 과제**

임베디드 애플리케이션 개발 복잡성으로 인해 프로세스를 수동으로 관리하기가 상당히 어렵고 시간이 많이 듭니다(그림 4). 한편, 차량용 임베디드 소프트웨어 엔지니어는 모든 차종을 지원하는 양질의 소프트웨어를 촉박한 일정 내에 제공해야 하는 압박을 받고 있습니다. 엔지니어가 이러한 기대치를 충족하기 위해 노력하는 와중에도 인터페이스, 기능적 동작, 하드웨어 통합, 하드웨어 제한 및 네트워크 대역폭 문제와 관련된 시스템 수준의 변경 사항이 빠른 속도로 진행되는 소프트웨어 개발 작업을 저해할 수 있습니다. 이러한 시스템 수준 변경 사항을 추적하는 작업은 기한 준수 여부를 좌우할 정도로 어려운 과제입니다. 더욱이 이러한 변경 사항을 추적하지 못하면 변경이 이뤄진 시점에 따라 많은 비용을 들여 재작업을 해야 할 수 있고, 이는 개발을 더 지연시키고 잠재적으로 실행 위험이 추가될 수 있습니다.

시스템 수준의 변경 사항 및 업데이트 구현은 여러 도메인과 조직이 관여하는 활동입니다. 새시 그룹의 새로운 사양은 전기 배선 하네스 또는 전비 중량 고려 사항과 관련된 문제를 유발할 수 있으며, 이로 인해 파워트레인 기능에 새로운 요구 사항이 적용되며, 이는 다시 차체 제어 그룹의 애플리케이션에도 영향을 미칠 수 있습니다. 팀에서 개발 주기를 진행해 나가면서 변경 사항의 시작점과 영역 및 조직 전반에 대한 영향을 추적하고 파악하기란 거의 불가능에 가까워지고 있습니다. 이로 인해 변경 사항을 완전히 파악하는지 여부와 상관없이 소프트웨어 애플리케이션이 변경될 수 있습니다.

ECU, 센서, 차량 기능을 구현하는 데 필요한 기타 하드웨어는 다양한 구현을 지원하기 위해 통합되거나 수정될 수 있습니다. 다른 버전이 개발되거나 하드웨어가 통합되면서 임베디드 소프트웨어 애플리케이션은 이러한 버전에서도 최적으로 작동해야 합니다. 이를 위해 엔지니어는 각 하드웨어 또는 운영 체제의 사양과 요구 사항을 포착해 지원되는 다양한 하드웨어 버전 및 운영 체제에서 최적의 기능을 제공하도록 애플리케이션을 구성해야 합니다.

작업자 퇴직 또는 이직으로 누락되거나 애매한 정보를 분명하게 제공할 수 없는 경우, 설계 요구 사항의 명확성과 품질이 모든 수준에서 저하될 수 있습니다. 이와 같은 내부 지식 손실은 개발 과정의 중대한 장애물이 될 수 있습니다. 또한 여러 이해 관계자가 잘못되거나 오래된 모델로 작업할 경우, 소프트웨어와 기타 모델의 통합이 어려워질 수도 있습니다. 실제로 일부 그룹은 모델과 별도로 작업해 차량의 특정 구성 요소에 대한 실제 소스를 왜곡하게 될 수 있습니다. 이렇게 사소해 보이는 비일관성이 이후 상당한 지연과 비용 부담이 큰 재작업을 유발합니다.

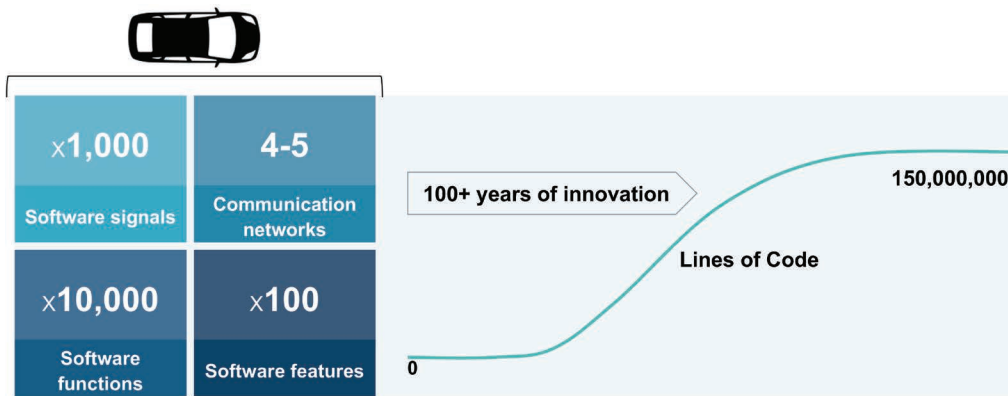


그림 4: 소프트웨어 복잡성이 커지면 애플리케이션 개발을 관리하기가 매우 어려워집니다.

# 통합 플랫폼과 임베디드 애플리케이션 정의 및 계획

임베디드 애플리케이션 정의 및 계획 프로젝트는 애플리케이션과 시스템 수준 간 연결을 통해 시스템 수준의 기능 또는 제품 정보를 받아들이고 추적해야 합니다. 이러한 연결은 시스템 수준의 변경을 통한 능동적이고 유기적인 정렬을 지원하며, 이를 통해 프로젝트는 개발 중에도 전체적인 시스템의 맥락을 유지할 수 있습니다. 소프트웨어 요구 사항의 정의 또는 변경은 새로운 요구 사항에 맞추기 위한 소프트웨어 아키텍처 및 모델 변경 및 제어 알고리즘에 맞추기 위한 업데이트를 트리거합니다. 이제 이와 같은 모델링 작업은 코드 수준의 변경 사항이 트리거되기 전에 원하는 결과에 대한 애플리케이션 및 시스템 수준에서 확인 및 검증될 수 있습니다.

## 제품 통합

구현부터 시스템 수준 맥락을 망라하는 명확한 시야는 복잡한 자동차 임베디드 소프트웨어 영역의 필수 요소입니다 (그림 5). 소프트웨어 엔지니어는 애플리케이션 정의의 구성 요소가 시스템 수준의 기능 정의 및 하드웨어 구속조건과 어떻게 연결되는지 알고 있어야 합니다.

시스템 수준의 기능 컨텍스트를 유지해야 모든 프로젝트 결과물이 시스템 수준의 요구 사항과 구속조건, 종속성과 일치할 수 있습니다. 시스템 수준에서 변경이 이뤄지는 경우 특히 그렇습니다. 어댑티브 크루즈 컨트롤 또는 새시 하드웨어 모듈을 구현하기 위해 자동 비상 제동과 어댑티브 크루즈 컨트롤 기능을 결합하는 새로운 요구 사항이 정의될 수 있습니다. 소프트웨어 엔지니어는 시스템 수준 컨텍스트를 유지해 이러한 변경 사항이 개발 중인 애플리케이션에 미치는 영향과 필요한 변경 사항을 이해할 수 있습니다. 엔지니어는 여러 차종을 빠르게 지원하는 탁월한 품질의 소프트웨어를 개발하면서 이러한 관계를 파악할 수 있어야 합니다.

임베디드 애플리케이션 개발을 위해 하나로 통합된 확장형 플랫폼은 영역과 조직 전반에 걸쳐 소프트웨어 개발 활동과 시스템 수준 정의 간의 다양하고 복잡한 관계를 관리하는 데 있어 매우 중요합니다. 이러한 플랫폼을 통해 작업 항목에 대한 요구 사항을 조율하고 소프트웨어 아키텍처 전반에 걸쳐 변경 사항을 계단식으로 적용하고 구현할 수 있습니다.

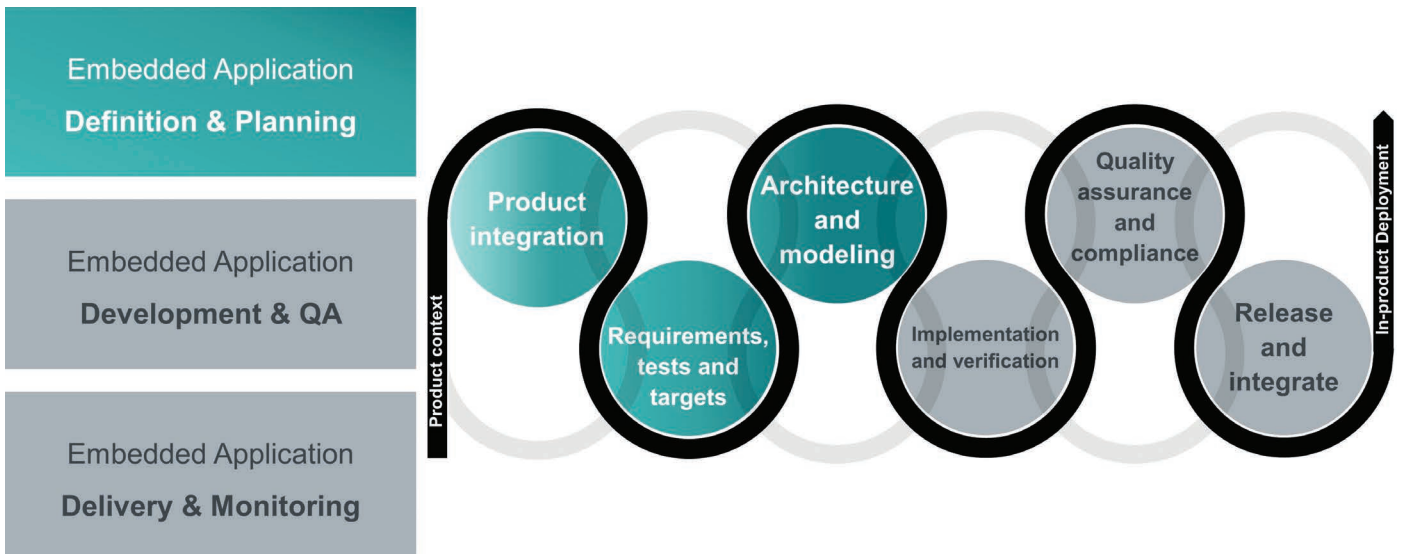


그림 5: 임베디드 애플리케이션 개발 프로세스 전반에 걸쳐 차량 수준의 기능 정의 및 하드웨어 구속조건에 대한 명확한 시각화가 필요합니다.

Polarion과 같은 고급 소프트웨어 개발 조율 툴은 시스템 수준 기능 정의를 사용하고 추적해 시스템 수준 변경과 애플리케이션 개발 간 직접적인 연결을 생성하므로 애플리케이션과 전체 시스템 개발 간에 동기화 상태를 유지할 수 있습니다 (그림 6). 예를 들어, 모든 스티어링 제어 특징과 기능, 그 중에서도 특히 풀-드리프트 (pull-drift) 보상이나 차선 유지 지원을 관리하기 위해 시스템 정의에서 파워 스티어링 제어 모듈 (PSCM)용 애플리케이션을 요구할 수 있습니다. 조율 툴은 시스템 수준에서 관련 요구 사항, 사양, 차량 수준 모델, 구성 및 교정 매개 변수 정의 등을 가져올 수 있습니다.

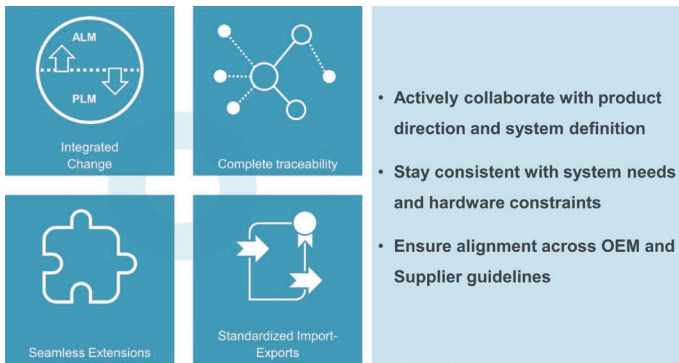


그림 6: 고급 애플리케이션 엔지니어링 관리 솔루션은 시스템 요구 사항을 사용하여 시스템과 애플리케이션 수준 간 직접적인 연결을 생성할 수 있습니다.

이는 전체적인 제품 및 소프트웨어 개발 주기를 통합하므로 제품 통합 단계라고 할 수 있습니다. 제품 통합을 통해 소프트웨어 정의 및 계획, 전체 제품 방향 및 시스템 정의 간 활발한 협업이 가능합니다. 이러한 협업을 통해 임베디드 소프트웨어 애플리케이션이 하드웨어 구속조건 및 높은 수준의 시스템 요구와의 일관성을 유지할 수 있습니다. 이는 또한 OEM과 공급업체가 각자의 지침과 목표를 조정하는 데도 도움이 됩니다.

### 요구 사항, 테스트 및 타겟

시스템 수준 정의 및 하드웨어 구속조건은 시스템 수준 요구 사항으로 코드화됩니다. 이러한 요구 사항을 통해 소프트웨어 엔지니어는 임베디드 애플리케이션 수준의 요구 사항, 노이즈 계수, FMEA (실패 모드 및 효과 분석), 테스트 사례 및 기능적 목표를 분해할 수 있습니다. 이러한 요구 사항은 각 임베디드 소프트웨어 애플리케이션에 따라 다르지만 시스템 수준의 컨텍스트와는 일관됩니다.

규정 및 프로세스 준수를 지원하기 위해 소프트웨어 엔지니어는 ISO26262 지침 및 Automotive-SPICE를 사용하여 애플리케이션 위험을 관리할 수 있습니다. 또한 엔지니어는 Polarion 플랫폼을 사용해 안전 목표, 기능적 안전 개념 및 관련 안전 요구 사항을 관리할 수 있으며, 업스트림으로는 시스템 컨텍스트, 다운스트림으로는 코드 및 모델 라인까지 완벽하게 추적할 수 있습니다. 이를 통해 엔지니어링 팀은 편차, 부적합 및 리콜 수를 줄일 수 있습니다.

통합 임베디드 애플리케이션 개발 플랫폼을 통해 사용자는 각 결과물을 개발하는 데 어떤 툴을 사용하든 모든 주요 관계자 및 이들의 결과물과 원활하게 협업할 수 있습니다 (그림 7). 이러한 협업 덕분에 소프트웨어 요구 사항 또는 하드웨어 구현 상에 변경이 계속 발생해도 모델이나 테스트 계획 같은 애플리케이션 결과물 개발이 일관적으로 이뤄집니다. 애플리케이션 개발 플랫폼에서 제공하는 협업을 통해 팀이 여러 ECU 하드웨어 버전과 운영 체제에 맞게 각 애플리케이션을 최적화할 수 있도록 사양을 파악할 수도 있습니다.

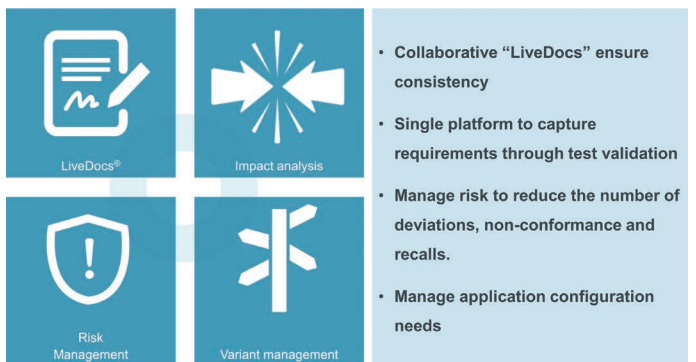


그림 7: 통합 애플리케이션 개발 플랫폼을 사용하면 애플리케이션 모델, 테스트 계획 등을 개발하는 팀 간에 협업과 일관성이 보장됩니다.

### 아키텍처 및 모델링

소프트웨어 아키텍트는 아키텍처 및 모델링 단계에서 작업하며 소프트웨어 모델을 제작하고 소프트웨어 구성 요소 간 상호 작용을 포착해 소프트웨어 아키텍처를 최적화하기 위해 애플리케이션별 요구 사항과 사양, 동작에 관여합니다 (그림 8).

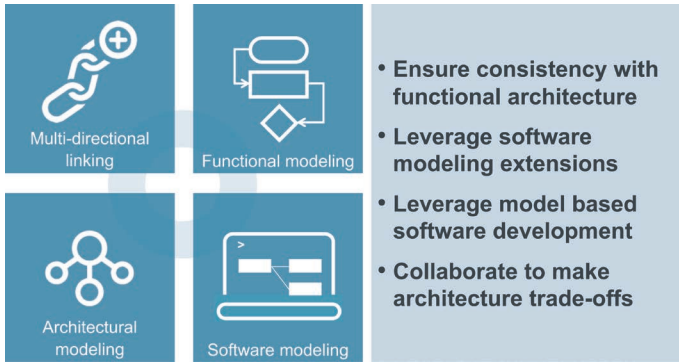


그림 8: 통합 애플리케이션 엔지니어링 플랫폼을 사용하면 엔지니어가 아키텍처 및 모델링 작업을 조율하고 절충안을 분석하며, 모델 기반 소프트웨어 개발을 활용할 수 있습니다.

모델 기반 소프트웨어 개발 방식은 이 같은 압박 속에서 작업하는 소프트웨어 엔지니어를 가장 효과적으로 지원할 수 있습니다. 엔지니어는 견고한 모델에서 시작해 최종 소프트웨어의 잠재적 작동 포함된 풍부한 아키텍처를 개발할 수 있습니다. 또한 모델 기반 방식으로 이러한 압박을 관리하는 데 필요한 속도에 맞춰 아키텍처를 편집하고 리팩토링하며 보강 및 분석할 수 있습니다.

아키텍처 분석, 테스트 및 검증을 개발 주기 초반으로 옮길 수 있습니다. 테스트가 실행되면 그 결과를 다시 소프트웨어 모델에 통합할 수 있습니다. 메카트로닉 솔루션과 같은 다른 시뮬레이션 솔루션과의 상호 운영성은 기능적 목업 인터페이스(FMIs)를 통해 이뤄지며, Simulink와의 상호 운영성은 테스트와 검증에서 다시 소프트웨어 모델로 이어지는 페루프를 형성합니다.

Polarion은 ESD (Embedded Software Designer), MATLAB/Simulink, Enterprise Architect 등의 산업 표준 툴과 직접 연결하는 방식으로 이 결과물 제조 및 생성을 조정합니다. 모델과 구성 요소 상호 작용이 개발되고 성숙함에 따라

Polarion은 이를 애플리케이션 정의에 다시 연결할 수 있습니다. 이렇게 하면 필요한 모든 소프트웨어 구성 요소 상호 작용과 연결된 강력한 아키텍처를 통해 필요한 모든 요구 사항, 사양, 동작을 모델로 표현할 수 있게 됩니다. 이 단계는 모든 코드가 구현되기 전에 소프트웨어 모델을 검증합니다.

아키텍처와 모델링 단계는 다방향 요구 사항 추적 기능으로 시스템 및 소프트웨어 엔지니어가 공통 환경을 통해 차량 소프트웨어 애플리케이션을 설계하고 검증할 수 있게 해줍니다. 이를 통해 다운스트림 설계 의사 결정에 관해 알려주며 이를 반영하는 소프트웨어 아키텍처 및 개별 애플리케이션의 디지털 트윈이 생성됩니다. 이 단계는 기능 아키텍처, 애플리케이션 동작 모델 및 요구 사항 전반에 걸쳐 일관성과 협업을 보장하는 역할도 합니다. 다양한 팀에서 아키텍처 절충안을 만들고 소프트웨어 모델을 활용하여 문제를 식별하며 작동 변동성을 관찰하고 의미 체계를 조사할 수 있습니다.

소프트웨어 팀은 단일 플랫폼에서 애플리케이션 정의 및 계획 단계를 통합해 필요한 툴 세트 전반에 걸쳐 모델링, 코딩, 테스트 실행, 빌드 프로덕션 등에 대한 작업을 할당할 수 있습니다. 소프트웨어 엔지니어는 언제든지 시스템 수준 정의 및 구속조건을 살펴보고 다른 시스템 사용자와 협력할 수 있습니다.

또한 상세 소프트웨어 요구 사항을 만들고, 이후에 변경할 경우 소프트웨어 아키텍처와 모델링의 변경이 트리거됩니다. 이처럼 정제된 요구 사항을 통해 소프트웨어 엔지니어는 제어 알고리즘에 맞춰 이러한 모델을 업데이트할 수 있습니다. 소프트웨어 엔지니어는 코드 수준의 변경 사항이 트리거되기 전에 애플리케이션 및 시스템 수준에서 원하는 결과가 달성됐는지 확인하고 검증하기 위해 MiL (model-in-the-loop) 테스트를 실행할 수도 있습니다.



# 결론

최신 차량의 소프트웨어 콘텐츠는 변화하는 자동차 업계의 요구를 수용하기 위해 놀라운 속도로 성장하고 있습니다. 고급 안전 기능, 차량 연결, 맞춤형 환경 등 차량의 기능에 대한 소비자의 기대치도 훨씬 높아졌습니다. 파워 스티어링, 엔진 관리, 브레이킹 시스템 등 기본적인 차량 시스템도 지난 몇 년에 걸쳐 소프트웨어 영역으로 들어왔습니다. 이와 같은 개별 기능을 이해하기는 어렵지 않지만, 이러한 기능과 다른 차량 시스템 사이에 이루어지는 상호 작용이 증가하면서 새로운 문제가 발생하고 있습니다.

소프트웨어 콘텐츠의 복잡성과 정교함이 증가하고 사이버와 물리적 환경 사이의 상호 의존도가 높아지면서 제품 및 애플리케이션 라이프사이클에 맞는 새로운 조정 수준이 필요한 상황입니다. 이러한 조율을 위해서는 최고 수준의 시스템 정의에서 개별 구현까지 추적할 수 있는 통합 제품-소프트웨어 디지털 스레드를 사용하는 강력한 솔루션이 필요합니다 (그림 9).

이러한 솔루션으로 소프트웨어 엔지니어링 팀은 각 결과물과 관련된 시스템 컨텍스트에 대한 명확한 지식을 바탕으로 애플리케이션 개발 작업을 수행할 수 있습니다. 팀에서는 협업을 통해 아키텍처 절충안을 만들고 보다 민첩하게 변경 사항을 관리할 수 있습니다. 이러한 솔루션은 또한 검증 및 확인 과정부터 애플리케이션 정의,

계획 및 개발에 이르는 페루프를 구현해 지속적으로 품질을 최적화하고 개선합니다.

하지만 애플리케이션 정의 및 계획 단계는 점점 컴퓨터화되는 차량을 만드는 데 필요한 첨단 소프트웨어 애플리케이션을 개발하는 첫 단계에 불과합니다. 계획과 정의가 완료되고 나면, 소프트웨어 엔지니어는 모델을 실제 코드로 구현하고 높은 품질을 위해 테스트를 진행해야 합니다. 또한 애플리케이션은 애플리케이션이 배포될 차량의 하드웨어와 운영 체제 변형에 맞춰 구성되어야 합니다. 이어지는 하위 프로세스에도 통합 임베디드 애플리케이션 개발 플랫폼의 이점이 동일하게 적용됩니다.

궁극적으로 조직, 엔지니어링 영역, 기능 추상화 전반에 걸쳐 소프트웨어 애플리케이션 개발을 효과적으로 관리할 수 있는 기업은 디지털화된 자동차 산업에서 성공을 구가할 최적의 입지를 확보할 수 있습니다. 이는 차량 운영 체제 및 펌웨어의 사용 증가로 인해 소프트웨어 기능의 ECU 하드웨어에 대한 의존도가 낮아지는 상황에서 특히 그렇습니다. 각 애플리케이션과 시스템 구속조건, 잠재적 하드웨어 구속조건을 추적하는 강력한 방법론은 성공으로 가는 핵심입니다. 자동차 임베디드 애플리케이션 개발을 위한 통합 플랫폼은 이러한 능력이 마련될 기반이 됩니다.

Siemens 솔루션이 귀사에 어떤 도움을 줄 수 있는지 자세히 알아보려면 [siemens.com/aes](https://www.siemens.com/aes)에서 블로그, 백서, 팟캐스트, 제품 동영상, 웨비나, 솔루션 기능, 인포그래픽 등을 확인해 보십시오.

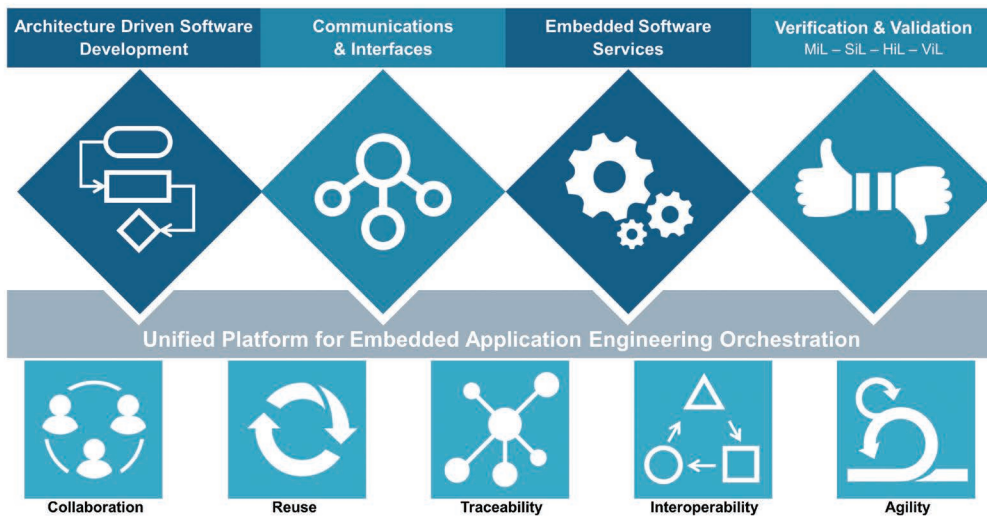


그림 9: 임베디드 애플리케이션 엔지니어링 조정을 위한 통합 플랫폼은 추적 기능을 제공하고 애플리케이션 개발 과정 전반에 걸쳐 협업을 촉진합니다.

## Siemens Digital Industries Software

### 본사

Granite Park One  
5800 Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 972 987 3000

### 미주 지역

Granite Park One  
5800 Granite Parkway  
Suite 600  
Plano, TX 75024  
USA  
+1 314 264 8499

### 유럽 지역

Stephenson House  
Sir William Siemens Square  
Frimley, Camberley  
Surrey, GU16 8QD  
+44 (0) 1276 413200

### 아태 지역

Unit 901-902, 9/F  
Tower B, Manulife Financial Centre  
223-231 Wai Yip Street, Kwun Tong  
Kowloon, Hong Kong  
+852 2230 3333

## Siemens Digital Industries Software 소개

Siemens Digital Industries Software는 엔지니어링, 제조 및 전자 설계가 미래와 만나는 디지털 엔터프라이즈를 실현하기 위한 혁신에 박차를 가하고 있습니다. Siemens Digital Industries Software의 솔루션은 규모를 막론한 모든 기업이 조직에 혁신을 촉진할 새로운 인사이트, 기회, 자동화 수준을 제공하는 포괄적 디지털 트윈을 생성하고 활용할 수 있도록 지원합니다. Siemens Digital Industries Software 제품과 서비스에 대한 자세한 내용을 보려면 [sw.siemens.com/software](https://sw.siemens.com/software)를 방문하시거나 [LinkedIn](#), [Twitter](#), [Facebook](#), [Instagram](#) 계정을 팔로우 하십시오. Siemens Digital Industries Software – Where today meets tomorrow.

## [siemens.com/software](https://sw.siemens.com/software)

© 2019 Siemens. 관련 Siemens 상표 목록은 [여기](#)서 확인할 수 있습니다. 기타 모든 상표는 해당 소유자에 귀속됩니다.

81120-83172-C4-KO 12/20 LOC