



## 향후의 자동차 소프트웨어 개발은 어떤 방식일까요?

지난 수년 동안 자동차 산업은 빠르고 쉽게 업데이트할 수 있는 소프트웨어를 통해 더 많은 기능을 사용할 수 있는 소프트웨어 정의 차량(Software Defined Vehicle)으로 진화하였습니다.

오늘날의 차량에는 수천만 라인 이상의 소프트웨어가 동작하고 있지만 아직 기초 단계일 뿐입니다. 연결성, 자율주행 및 사용자의 환경개선(UX, 사용자 경험) 부문의 전반적인 변혁은 지속적으로 진행 중이며 각 부문의 혁신에는 복잡한 소프트웨어의 실행을 필요로 합니다.

따라서 자동차산업 전체에서 가장 높은 우선 순위는 효율적인 소프트웨어의 개발, 테스트 및 업데이트할 수 있도록 뛰어난 툴, 프로세스 및 개발환경을 개발자에게 제공하는 것입니다.

이에 대한 해답은 CI/CD(지속적인 통합과 배포) 방식의 도입이지만 추가적으로, 실시간으로 글로벌 협업이 가능하고 자동차 애플리케이션이 요구하는 엄격하고 광범위한 테스트가 가능하도록 충분히 강력한 클라우드 네이티브 개발 및 통합 개발환경 플랫폼이 필요합니다.

### 더 효과적인 접근법

소프트웨어가 새로운 수준의 안전, 편안함 및 편리함으로 차량을 발전시키면서 개발자는 이제 기존의 시스템보다 더 현대적이고 뛰어난 소프트웨어 개발 환경을 필요로 합니다.

과거에는 소프트웨어 개발이 융통성이 부족하고 속도가 느린 '폭포수' 개발 방식을 따르고 고도로 단편화된 툴 체인을 사용했습니다. 개발은 이전 단계를 마쳐야만 다음 과정을 시작하는 별개의 단계를 거쳤습니다. 툴 체인의 일부를 다음 단계로 전환하는 방식으로 대부분의 개발 프로세스가 수동이었습니다. 이러한 개발 방식에서는 다음 버전의 소프트웨어가 배포되기 까지 6~8개월이 소요되었습니다.

오늘날 자동차 분야에서 소프트웨어는 하드웨어로부터 점차적으로 추상화되고 있으며 개발자에게는 이러한 구조적인 변화의 이점을 활용할 기회가 주어지고 있습니다. 개발자들은 현대적 애자일 방식과 DevOps 방식을 사용하여 하드웨어 변경 또는 기타 물리적 업데이트에 구애받지 않고 훨씬 더 빠르고 자유롭게 개발 일정에 맞게 소프트웨어를 업데이트할 수 있습니다. 이러한 개발 방식은 이전에 경험할 수 없었던 수준의 개발 속도, 확장성, 품질 및 보안을 제공합니다.

지속적인 통합(CI)에서 소프트웨어를 구축하는 작업은 자동화되고, 한 부분의 출력이 다음 부분의 입력이 되므로 소프트웨어 생산의 모든 단계가 'CI 체인'에 포함됩니다. 지속적인 배포(CD)는 소프트웨어 새 버전 배포를 현장 자동화합니다. 이제 자동차에서 CI/CD 뿐만 아니라 연속 테스트(CT)가 가능하지만, 새로운 첨단 어플리케이션은 실제 차량이나 테스트 벤치에서 복잡한 시뮬레이션을 통해 검증되어야 하기 때문에 테스트는 자동차 산업에서 중요한 과제가 되었습니다. 예를 들어, 고속도로에서 저속 차량을 추월하거나 차량, 보행자 및 다른 장애물이 있는 복잡한 도시 환경을 운행하는 것과 같은 동작을 자동으로 실행하기 위해 여러 레이더와 카메라에서 전달된 입력을 분석하는 테스트 소프트웨어의 복잡성을 떠올려 보십시오.

### 필요한 단계 변화

BCG(Boston Consulting Group)는 "내연 기관에서 전기 자동차(EV)로의 전환으로 이미 업계를 흔들기에 충분했지만, 소프트웨어를 통한 혁신은 시장의 흐름을 통째로 바꾸거나 판도를 뒤집어 놓을 만한 결정적인 역할을 했다"라고 썼습니다. "모빌리티 업계의 유례없는 변동의 중심에서 자동차 제조업체는 시장 접근을 가속화하고 고객의 요구에 더 빠르게 반응해야 합니다. ... 기존의 방식으로 작업하는 것을 고수하는 업체는 이러한 새로운 환경에서 경쟁하기 위해 필요한 변화를 이루지 못한다". BCG에 의하면 애자일은 시장 진출 속도, 위험 감소 및 더 뛰어난 제품을 포함한 여러 이점을 제공합니다.

이상적인 접근 방식은 전 세계의 임의의 테스트 센터내의 테스트 벤치에서 어떠한 테스트라도 실행할 수 있는 클라우드 기반 중앙 집중식 플랫폼을 통해 개발자가 개발 및 테스트 프로세스의 병목 현상을 제거하는 것입니다. 테스트에 더 현실적인 상황을 반영하기 위해 SIL(Software-in-the-Loop), HIL(hardware-in-the-loop) 또는 VIL(Vehicle-in-the-loop)을 사용할 수 있습니다. 이 접근 방식을 통해 현재 가장 큰 난제를 적절한 규모의 글로벌 자원을 활용하여 해결할 수 있습니다.

**기존 방식의 문제점**

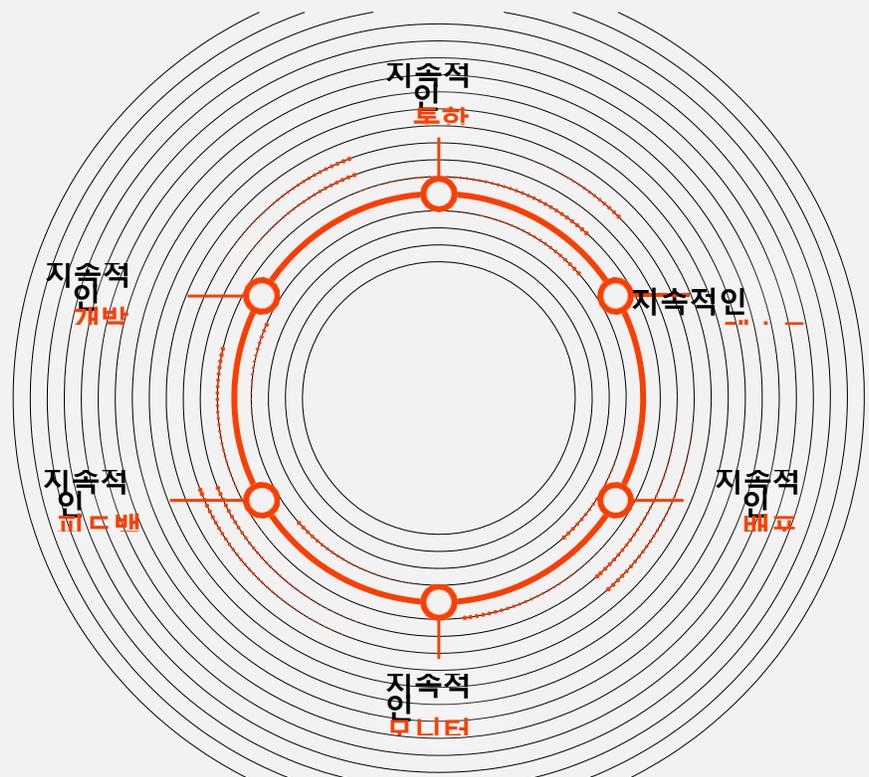
개발자가 하루에 수백만 줄의 코드를 작성하고 있을 경우 기존의 소프트웨어 개발, 테스트, 통합 및 배포에 대한 접근 방식은 지속 가능한 방식이 아닙니다. HIL 테스트의 요구 조건은 CI 과정에서 코드의 변경이 소프트웨어에 통합되면서 CI 시스템은 물리적으로 HIL 벤치 옆에 있어야 한다는 의미입니다. 이 접근 방식에는 다음과 같은 여러 단점이 있습니다.

- 단 몇 분 이내로 완료되어야 하는 시급한 상황에서도 하나의 빌드에 너무 많은 시간이 소요됩니다. 빌드가 길수록 자동화된 테스트를 할 시간이 더 짧아지기 때문에 품질은 물론 출시 속도에 영향을 미치는 심각한 병목 현상을 보입니다. 빌드가 너무 길 경우, 추가 테스트 또는 기능 개발에 값지게 사용되어야 할 시간을 소비하게 됩니다.
- 기존 인프라는 비용이 많이 들고 규모와 보안성 및 프로그램 간의 유연성이 부족합니다.
- 모든 프로그램은 재활용성이 없으며 기성 솔루션을 사용하지 않고 백지 상태에서 시작합니다.

- 일반적으로 통용되는 소프트웨어 방법론 없이는 '새도우 IT' 현상이 불가피하게 나타나는데 이는 종종 개발자들이 조직에서 승인되지 않은 자신이 선호하는 도구를 사용한다는 의미입니다. 이는 IT에 개발 프로세스의 가시성 부족으로 비효율적인 뿐만 아니라 보안상의 위험을 초래할 수 있습니다. 또한 서로 다른 도구를 사용할 경우 개발자가 상호간에 또는 고객과 협업하기가 더 어렵게 됩니다.
- 동일한 소프트웨어 프로그램을 지원하는 CI 체인은 HIL 테스트 벤치를 보유하고 있는 전 세계의 여러 기술 센터로 분산되어 있습니다. 표준 접근방식의 부재 시 기업은 클라우드 상에서 개발할 수 없으며 글로벌 표준화를 향한 일관성과 반복성을 달성할 수 없습니다.
- 모빌리티 소프트웨어 제품을 개발하는 팀 또는 공급업체가 증가함에 따른 통합 포인트의 증가는 품질, 납품 및 납품시간 측면에서 영향성이 증가하고 있습니다. 가장 좋은 방법은 코드를 초기에 자주 통합하는 것이지만 기존의 접근 방식은 개발 프로세스에서 유연성이 없는 게이트를 생성하여 다음 병합 게이트까지 테스트할 수 없는 문제가 발생합니다. 그 결과로 인하여 전체 프로젝트가 불필요하게 지연됩니다.

**지속적인 공급**

포괄적이고 지속적인 공급 방식을 통해 각 작업은 진행 중인 주기의 다음 작업을 알려줍니다.



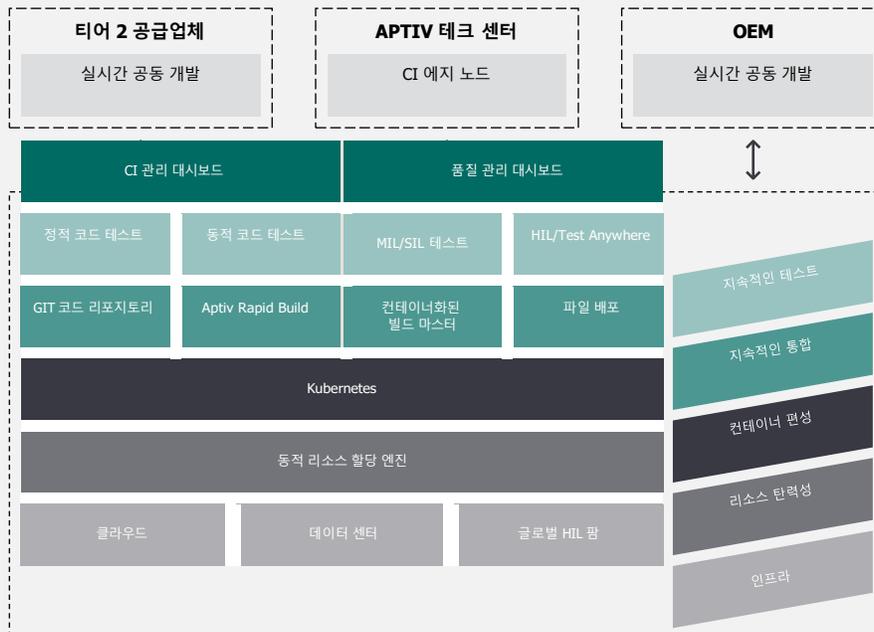
### 클라우드로 향하여

차량은 생명 안전 시스템이기 때문에 자동차 기업은 오랜 기간 동안 설계와 테스트에 대한 요구 사항을 강조하는 방식을 채택해 왔습니다. 자동차 기업들은 요구 사항 각각의 항목에 대한 검증을 통해 안전 및 법적 고려 사항이 올바르게 반영되었는지 확인합니다. 실제로 ISO 26262 기능 안전 표준은 기능에 대한 각각의 요구 사항을 테스트해야 할 필요성을 강조합니다. 간단히 말해서 제조업체는 양산 시점에 솔루션이 사양에 맞다는 것을 증명해야 하고, 양산 후 변경사항 적용 시 이전에 작동했던 모든 것이 여전히 올바르게 동작하는지를 확인해야 합니다.

지난 10년 동안 자동차 산업 애플리케이션의 소프트웨어 요구 사항의 수는 수백에서 수만으로 증가했습니다. 이러한 요구 사항 중 많은 부분이 차량에 탑재되는 것과 동일한 전자 제어 유닛(ECU) 하드웨어에서 실행되는 임베디드 소프트웨어로 테스트해야 합니다. 이 테스트는 반응 시간을 정확하게 예측할 수 있도록 실시간으로 진행되어야 합니다. 즉, 각 테스트 항목당 평균적으로 몇 분의 테스트 시간이 소요됨을 의미합니다. 그 외에도 소프트웨어는 종종 여러 ECU, 센서 및 기타 장치에 퍼져 있으며 서로 다른 회사의 여러 팀에서 개발될 수 있습니다.

### 지속적인 통합

차세대 지속적 통합은 프로그램 품질과 속도를 강조하는 방식을 통한 광범위한 세트의 클라우드 기반 도구를 요구합니다.



이러한 점은 동일한 프로그램을 개발하는 기업이 각기 다른 CI 체인을 사용하고, 지리적으로 여러 곳에 분산되어 있으며, 자체 HIL 테스트 벤치를 사용하게 합니다.

보다 효과적인 방식은 각 프로그램에 통합된 단일 툴체인을 만드는 것입니다. 이 툴체인은 글로벌하게 사용할 수 있는 클라우드 기반이어야 합니다. 또한 새로운 툴체인을 빠르고 쉽게 배포할 수 있는 메커니즘이 있어야 합니다.

클라우드 기반 툴체인은 CI 빌드 시간에서 병목 현상을 막을 수 있는 확장성을 가지고 있는데, 병목 현상은 테스트 시간을 부족하게 하고 개발을 지연시킬 수 있기에 중요한 부분입니다.

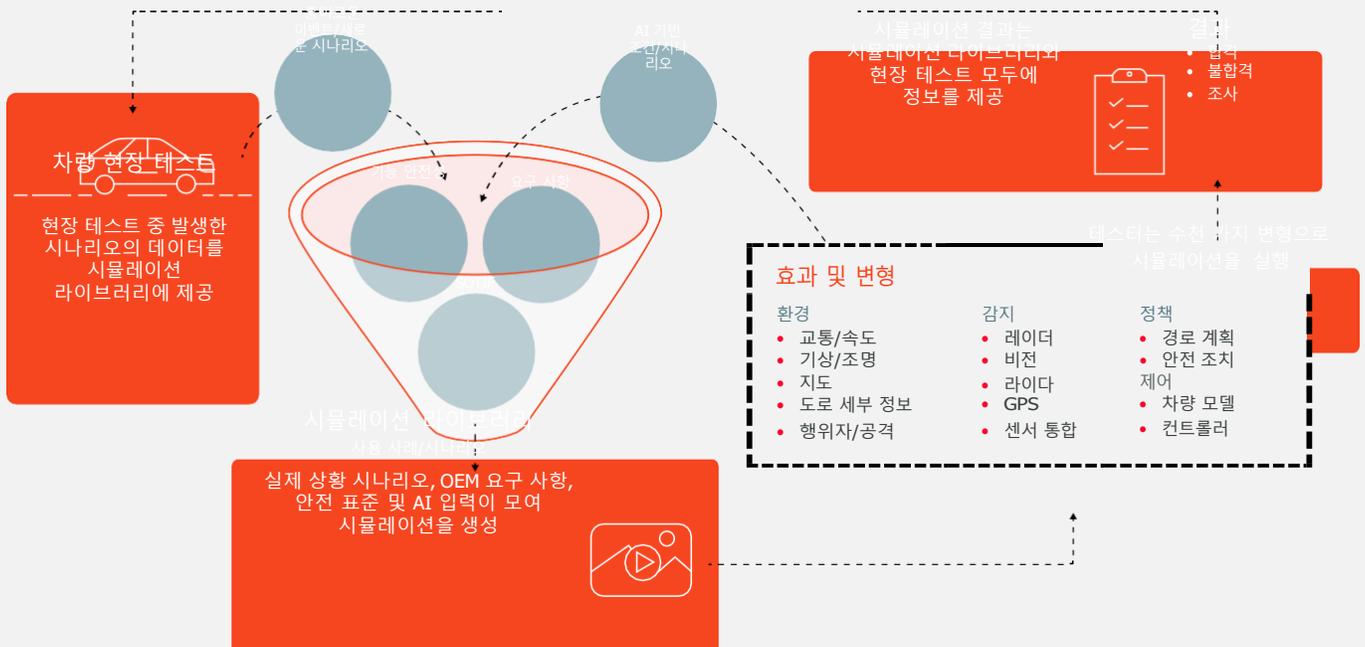
클라우드 기반의 접근 방식에서 코드에 적용된 업데이트는 개발 팀 내에서 공유되므로 모두가 프로젝트 상태에 대해 잘 알게 되기 때문에 개발이 원활하게 진행됩니다. 이를 통해 자동화 증대, 원격 문제 해결, 빠른 피드백, 명확한 팀 할당 및 분명한 제품 오너십이 가능합니다.

클라우드 기반 CI를 사용하는 것은 개발 팀에서 만든 새로운 코드가 더 큰 코드 기반에 자동으로 통합된다는 의미이며, OTA(Over-the-air)를 사용하는 클라우드 기반 CD는 성공적으로 빌드된 코드를 어디에서든 테스트 또는 프로덕션 환경에 자동으로 배포되게 합니다. 클라우드 접근 방식을 올바르게 사용하면 프로세스 전체에 걸쳐 재해 복구를 포함한 보안성도 구축할 수 있습니다. 또한 클라우드 호스팅되는 중앙 집중식 인프라는 매우 안전하고 탄력적인 CI 체인 전체에 대한 단일 관리 모니터링이 가능합니다.

### 지속적인 테스트

자동차 업계에서 지속적인 테스트는 물리적 세트와 소프트웨어 기반 시뮬레이션 간의 피드백 루프를 생성한다는 의미입니다.

#### 물리적 테스트를 위한 피드백



그 외에도 유연성을 통해서 개발자는 개발 주기의 후반부에 제품을 개선할 수 있게 됩니다. 향후의 소비자 욕구를 예측하는 대신 개발자는 지난 연도의 요구 사항을 정의하고 테스트하여 소비자의 기대에 더욱 근접한 시스템을 구축할 수 있습니다.

그 후 향후 몇 년 간 OTA 소프트웨어 업데이트를 배포하여 보증 비용을 줄이고, 소프트웨어는 지속적으로 소비자의 기대를 충족시킬 수 있습니다.

### 테스트가 열쇠

테스트는 자동차 산업에서 타이어 고무가 실제 도로를 만나는 지점입니다. 차량에 마일당 7~10 달러로 새로운 소프트웨어를 물리적으로 로드하고 모든 유형의 주행 조건에서 소프트웨어가 올바르게 작동하는지 확인하기 위해 필요한 대략 수십만 마일의 주행을 테스트하는 것은 시간 소모는 말할 필요도 없으며 비용도 매우 비쌉니다.

시뮬레이션 프로그램을 통해 기업은 비용을 절감하고 유연성과 반복성을 제공하기 위한 자동차 업계 소프트웨어를 제작할 수 있습니다. 게다가 시뮬레이션 경로를 통해 기업은 전체 제품을 완성한 후 테스트하고 다시 수정하고 처음부터 다시 테스트하는 대신 개발 중인 복잡한 솔루션의 코드 일부나 구성 요소를 테스트할 수 있습니다.

시뮬레이션 기술과 SIL 또는 HIL을 조합하여 사용하면 매일매일의 빌드를 테스트하고 검증할 수 있습니다. 그리고 멀티 스레드된 SIL 또는 HIL은 여러 테스트를 순차적이 아닌 동시에 진행하여 시간을 절약하고 효율성도 향상합니다.

SIL 테스트는 소프트웨어가 생성된 모델링 환경 내에서 모두 진행됩니다. SIL 테스트는 특별한 하드웨어가 필요하지 않다는 이점이 있으며(모든 노트북이나 다른 컴퓨팅 플랫폼에서 가상으로 테스트 가능) 설계 초기 단계에서 테스트하는 데 가장 적합합니다.

HIL 테스트에서는 시스템이 실제 차량과 연결되어 있다고 믿게끔 차량 및 ECU 환경 입력을 시뮬레이션합니다. HIL 벤치에는 관련된 모든 차량 구성요소가 포함됩니다. 시뮬레이터는 실제 카메라 및 레이더 시스템에 입력을 제공하고 테스트 중인 시스템에 신호를 차례로 전송하여 해당 입력에 올바르게 반응하는지 확인합니다.

예를 들어 테스트 스크립트는 차량이 빗속에서 60mph로 구부러진 길을 주행 중에 갑자기 알 수 없는 물체를 마주 한대거나 마주 오던 차가 방향을 틀어 중앙선을 침범하는 시나리오를 생성할 수 있습니다. HIL 테스트 벤치에 부착된 카메라 및 레이더가 이미지를 ECU에 보내면 테스트 중인 시스템이 실시간으로 데이터를 처리하여 취해야 할 조치를 결정합니다.

Aptiv 테스터는 이러한 접근 방식을 사용하여 레이더 및 카메라 입력을 포함한 실제 센서 시뮬레이션으로 매일 32만1900km(200,000마일)를 '주행하면서' 하루 100만 개 이상의 Object-level 시뮬레이션을 진행할 수 있습니다.

또한 테스터가 시뮬레이션을 통해 발생빈도가 현저히 적은 잠재적으로 위험한 사례를 빠르게 테스트할 수 있습니다. 일반적으로는 특정 주행 조건 또는 구성요소 문제를 재현하기 위해 누군가가 수백 또는 수천 마일을 운전해야 할 수 있습니다. 시뮬레이션은 온디맨드 테스트가 가능할 뿐만 아니라 특히 위험한 시나리오를 다시 시뮬레이션하여 다양한 소프트웨어 버전이 동일한 입력에 대해 어떻게 반응하는지 보여줍니다.

### 클라우드 기반 CI/CD/CT의 이점

개발 표준 및 애자일 개발 원칙을 기반으로 하는 글로벌 통합 클라우드 네이티브 아키텍처는 전형적인 병목 현상을 없애고 자동차 산업 관련 고급 소프트웨어를 개발하는 회사에게 핵심적인 이점이 됩니다. 다음과 같은 이점이 있습니다.

- **성능:** 전 세계적으로 사용 가능한 클라우드 기반 아키텍처는 세계 어디에서나 테스트 벤치를 중앙에서 원격으로 관리할 수 있습니다. 그렇기 때문에 확장성과 유연성이 증가합니다.
- **속도:** 자동화는 빌드 시간을 최대 80%까지 단축시킬 수 있습니다. 회사는 컨테이너 기반 아키텍처 및 코드 병합 병목 제거 접근 방식을 통해 병목 현상을 없앨 수 있습니다. 변경된 부분만 재구축 및 테스트하여 60% 더 빠르게 배포할 수 있습니다.
- **공동 개발:** 여러 파트너가 서로의 코드를 실시간으로 테스트할 수 있으면 공급업체가 코드를 커밋한 당일에 모두가 통합 문제를 찾고 수정하고 테스트할 수 있게 됩니다. 그 결과 OEM의 복잡한 소프트웨어 지원 기능을 위험도 낮게 고품질로 빠르게 제공할 수 있을뿐만 아니라 생산 후에도 유지 관리 및 개선을 비용 효율적이고 간단하게 할 수 있습니다.
- **품질:** 자동차 산업에서 애자일 개발에 대한 전형적인 과제인 원격 SIL/HIL 테스트 벤치와의 완전한 통합으로 개발자가 장소에 구애받지 않고 동시에 제한 없이 테스트할 수 있기 때문에 제품 품질이 향상됩니다.
- **투명성:** 팀 전체는 프로그램 규모, 복잡성 또는 위치에 구애받지 않고 모든 소프트웨어 개발 체인의 최신 상태에서 완전한 가시성을 얻게 됩니다. CI의 모범적인 사례를 실현하고 지원할 수 있으며 품질에 영향을 초래하기 전에 문제를 식별할 수 있습니다. 이는 또한 보안 정책을 시행하는 것을 단순화합니다.

### 데이터 마이닝

Aptiv는 고객 프로젝트당 400,000~100만 마일의 데이터를 운영하고 기록합니다. 이 가공되지 않은 정보와 시나리오 데이터의 방대한 데이터베이스는 전 세계에서 다양한 시간과 기후 조건에 따른 모든 환경에서의 레이더 및 카메라 데이터를 포함합니다. 제품 개발자는 로그 파일을 분석하여 Aptiv 센서, 제어 장치 및 OEM 차량이 전체 시스템으로써 상호 작용하는 방식을 관찰할 수 있으며, 이 데이터를 사용하여 추가 개발 정보를 제공할 수 있습니다.

### APTIV의 CI 틀체인

Aptiv는 CI 소프트웨어 개발 틀체인에서 클라우드 기반 확장/축소 기술을 사용하여 여러 테스트를 병렬식으로 진행할 수 있습니다. 이는 개발 초기 단계의 요구사항보다 릴리스 날짜에 가깝게 정의된 요구사항을 구현할 수 있도록 하는 새로운 작업 방식을 지원합니다. 또한 이를 통해 동일한 툴 세트를 사용하여 동일한 대시보드에서 실시간으로 작업하는 모든 Aptiv, OEM 및 타사 개발자가 긴밀하게 협업할 수 있습니다.

결과는 놀랍습니다. 오늘날 플랫폼은 매일 1,000만 줄의 코드를 구축하고 테스트하여 속도와 품질을 월등하게 향상시켰습니다. 예를 들어, 첨단 안전 시스템의 전체 빌드 시간은 Aptiv의 도구 체인을 적용하기 전에는 12시간 이상이었고, 현재 업계 표준 빌드 기술에 기반하였지만 다음 빌드를 시작하기 전에 하나의 빌드 전체가 완료되어야만 했습니다.

플랫폼은 평균적으로 70%의 빌드 시간을 단축합니다. 이렇게 자동차 CI에서 커다란 병목 현상을 제거하여 모든 빌드에서 테스트 세트를 광범위하게 완전히 수행하고 품질을 비약적으로 발전시킬 수 있습니다. 이는 코드 병합 게이트로 알려진 지속적인 테스트에서 중요한 부분을 지원하며, 모든 코드 체크인에서 자동화된 테스트의 전체 수행으로 이어지는 전체 빌드로 구성됩니다.

자동차 솔루션이 대폭적으로 복잡해지면서 정적 인프라에 기반한 기존 CI 체인의 능력을 상회하게 되었습니다. Aptiv의 CD 플랫폼은 완전히 새로운 스마트 빌드와 탄력적 인프라 기술을 포함한 클라우드 네이티브 접근 방식을 채택하여 이러한 과제를 해결합니다. 이러한 기술을 통해 플랫폼은 수십에서 수백 개의 동시 빌드로 간단히 확장할 수 있으며 최적의 상태로 코드를 구축하고 테스트하기 위한 적절한 양의 컴퓨팅을 자동으로 프로비저닝합니다.

새로운 CI 체인은 병목 현상 없이 몇 분 안에 배포할 수 있습니다. 하나로 통합된 모니터링 관리 기능을 통해 보안 및 재해 복구가 내장되어 전 세계의 모든 CI 체인에 대한 가시성을 확보할 수 있습니다. 원클릭 방식으로 업데이트를 배포할 수 있으며 업데이트 내용은 모든 스프린트에 통합됩니다. 또한 틀체인은 ASPICE 및 TISAX 3를 지원하며 요구 사항 및 프로세스 규정 준수에 대한 전적인 추적성을 제공합니다.

**결과:** 개발 비용 절감, 시장 출시 속도 향상, 유연성 향상, Aptiv, OEM 및 타사 간의 실시간 공동 개발.

## 자동화의 능력

과거에 개발자는 소프트웨어를 수동으로 테스트했습니다. 소프트웨어 출시는 월 별로 있었으며, 테스트에는 그 정도의 기간이 필요했습니다. 하지만, 각 빌드마다 관련 코드 라인의 급증으로 이러한 방식을 지속하기 어려워졌습니다.

Aptiv는 주요 자동차 소프트웨어 프로젝트를 개발하고 테스트한 경험이 있으며 이에 따르는 과제를 이해하고 이 문제를 해결하기 위한 도구를 갖추게 되었습니다. 오늘날, CI/CT 파이프라인 덕분에 현재 96%의 테스트 프로세스가 자동화되었고, 테스트할 요구 조건이 증가함에도 테스트 실행 시간은 수일 내에 마무리할 수 있는 것으로 예측되고 있습니다.

우리는 지속적으로 자동화 수준을 올려 전체 빌드의 테스트를 하루 6~10시간으로 단축합니다.

**저자 정보****Brian Murphy**

엔지니어링 서비스 디렉터

Brian Murphy는 첨단 기술과 프로세스를 사용하여 제품 개발의 품질과 효율성을 향상하는 업무에 집중하고 있습니다. 그는 재무, 건강, 전기 통신 및 자동차 분야에서 25년 이상의 소프트웨어 개발, 아키텍처 설계 및 관리 경험을 가지고 있습니다. 높은 성과의 팀을 구축하고, 신속하게 비즈니스를 진행하고, 기대를 넘어선 혁신을 이루는 것은 Brian이 표준 작업으로 실천하는 세 가지 핵심 가치입니다.

[에서 자세히 알아보십시오](#)