

[특집]탄소중립 선언이 자동차 업계에 가져올 임팩트와 확장되는 서비스 시장

변화하는 자동차 제조 높아지는 소프트웨어의 중요성과 넘어야할 장벽

야마모토 타카시(山本隆史)

타카하시 츠카사(高橋 主)

CONTENTS

- I. 새로운 자동차 제조 : 자동차 업계에서 소프트웨어 개발 기능이 중요한 이유
- II. 소프트웨어 기점의 자동차 제조의 본질
- III. 새로운 자동차 제조 실현을 위해 일본계 자동차 OEM에 부과된 과제 : 두가지 장벽이 존재/
개발 장벽, 비즈니스 모델 장벽
- IV. 새로운 자동차 제조 이행을 위한 방향성 확립

요약

1. 새로운 자동차 제조가 요구되는 배경으로, 자동차 개발 기능에서 소프트웨어의 중요성과 소프트웨어 개발을 중심으로 한 아키텍처의 근본적인 재검토를 들 수 있다.
2. 자동차 제조에 대해 완전히 다른 사고를 하는 GAFA와 테슬라의 대두로 인해 자동차 업계에 지각 변동이 일어나며 소프트웨어를 중심으로 한 자동차 제조가 가속화되기 시작했다.
3. 소프트웨어 중심 세계에서는 소프트웨어의 중요성이 커져 자동차와 일상생활, 사회와의 융합이 급속히 진행된다. 또한, 많은 새로운 가치가 창출되고 탄소중립 사회를 향한 가치도 창출된다. 자동차 제조에서 핵심은 「소프트웨어 개발의 내화/내재화」이며 선행중인 자동차 업체는 이미 전문 체제를 정비·강화하고 있다.
4. 새로운 자동차 제조로 이행하는데 있어 애자일 개발로의 전환과 비즈니스 모델의 업그레이드가 일본계 자동차 OEM의 장벽으로 작용한다.
5. 「코어·논코어 영역 분배」의 개념을 응용해 워터폴과 애자일 개발의 기능을 적절히 구분하는 것이 새로운 자동차 제조로 이행하는 데 있어서 중요하다.
6. CASE 시대에 자동차의 비즈니스 모델은 OPEX형으로 이행된다. 이때 밸류체인에서 프래그먼트화된 스택홀더(이해관계자)를 끌어들이어 기존 틀에 얽매이지 않는 비즈니스 체계를 구축하는 것이 중요하다.
7. 새로운 자동차 제조에서 소프트웨어와 하드웨어의 균형을 적절히 맞추면서 개발 자원을 단계적으로 소프트웨어 쪽으로 이행하는 것이 향후의 전략상 중요하다. 그 과정에서 대담한 사고와 과거에 얽매이지 않는 마인드셋으로 바꾸는 일도 필요할 것이다.

I. 새로운 자동차 제조 : 자동차 업계에서 소프트웨어 개발이 중요한 이유

2020년 3월 토요타자동차의 토요타 아키오 사장은 스마트시티 플랫폼 조성을 위해 NTT와 업무 자본 제휴를 발표한 기자회견에서 소프트웨어를 중심으로 한 개발 프로세스의 중요성을 밝혔다. 토요타자동차의 e-Palette구상이나 Woven City등 최근 미디어의 주목을 모은 프로젝트의 배경에는 소프트웨어를 중심으로 한 새로운 자동차 제조에 대한 사고방식이 있다. 하드웨어 개발에 힘을 쏟은 기존의 자동차 업계에서 소프트웨어를 중심으로 한 자동차 제조가 일으킨 영향은 크고, 각국 주요 자동차 OEM은 그 대응책 마련에 서두르고 있다.

일반적으로 애자일(Agile) 방식으로 개발하는 소프트웨어 개발 속도는 하드웨어에 비해 매우 빨라서 하드웨어와 일체형으로 개발을 진행하면 소프트웨어의 강점인 빠른 개발 속도를 저해하는 제약으로 작용한다. 그래서 최근 자동차 업계에서는 「하드웨어와 소프트웨어의 분리」라는 개념이 보급되기 시작했다. 일반적인 개념으로는 소프트웨어와 하드웨어를 분리해서 개발함으로써 소프트웨어가 가져다 주는 신속한 개발과 이로 인해 향상된 기능이 하드웨어의 제약으로 저해되는 것을 방지할 목적으로 이용된다.

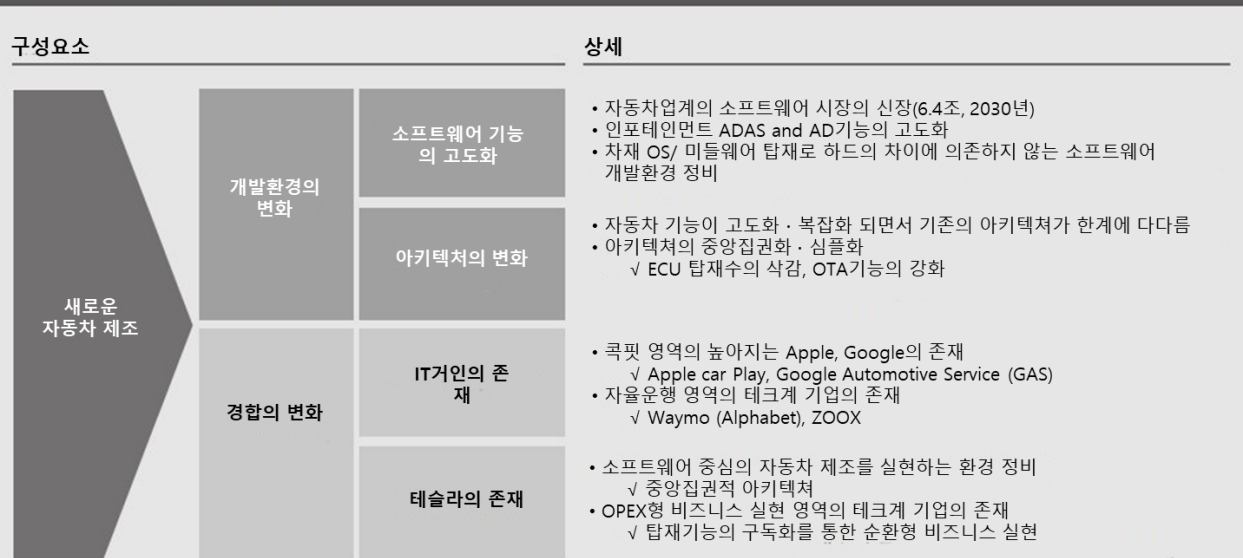
그러나 소프트웨어를 중심으로 한 자동차 제조를 본질적으로 이해하기 위해서는 「하드웨어와 소프트웨어의 분리」라는 문맥에서만 살펴본다면 어딘지 불안하다. 특히 하드웨어 중에서도 자동차는 복잡성과 기능성, 나아가 안전성의 담보가 가장 엄격하게 요구되는 제품 매체로 단순히 「하드웨어와 소프트웨어의 분리」라는 개념이 그대로 적용되는 일은 드물다. 실제로는 개발 기능 이외에도 사용자 측의 니즈나 급격한 변화가 초래하고 있는 경쟁 상황 등 다양한 요소가 얹혀 큰 파도가 되어 자동차 업계에 몰려들고 있다. 아래에서 소프트웨어를 중심으로 한 「새로운 자동차 제조」의 본질을 몇 가지 요소로 분석하고 소개하겠다.

1. 새로운 자동차 제조 개요

소프트웨어를 중심으로 한 새로운 자동차 제조는 크게 두 가지 항목(개발 환경의 변화, 경쟁의 변화)으로 구성된다(그림 1).

첫번째 요소인 「개발 환경의 변화」는 두 개의 상세 항목(소프트웨어 기능의 고도화, 아키텍처의 변화)로 나뉜다. 「소프트웨어 기능의 고도화」는 콕핏 주변 인포테인먼트^{주1} 기능과 ADAS (Advanced Driver-Assistance Systems: 첨단 운전자 보조시스템), AD(Automated Driving: 자율주행) 등의 발전으로 자동차 전체에 탑재되는

그림1 새로운 자동차 제조의 구성요소



출처) 각종 공개정보를 토대로 작성

소프트웨어 기능의 고도화·다양화를 불러일으켰다. 그 결과 2030년까지 자동차 개발에서 소프트웨어 시장은 비약적으로 성장(약 6.4조 엔)할 것으로 예측된다.

더욱이 탑재 소프트웨어·시스템 양이 비대화되고 복잡화됨으로써 자동차의 아키텍처 그 자체에 대한 재검토가 요구되고 있다(「아키텍처의 변화」).

두번째 항목인 「경쟁의 변화」는 Google이나 Apple과 같은 「IT 거대 기업의 존재」와 최근 비약적으로 전기 자동차의 판매 대수를 늘린 「테슬라의 존재」로 요소가 나뉜다. Google과 Apple은 그들의 강점인 소비자용 데이터, 디바이스 비즈니스를 통해 확보한 고객 접점을 레버리지해서 인포테인먼트 기능으로 자동차 산업에 본격적으로 진출하기 시작하였다.

세계 최대 EV 업체인 테슬라는 전기 자동차를 보급하기 위해 소프트웨어 중심의 자동차 개발 방법을 채택해 사용자에게 가치를 계속해서 제공하는 체계를 구축했다. 그 결과 기존의 자동차 판매형에 머물지 않고 탑재 기능의 구독화로 OPEX형 비즈니스 확립을 실현하고 있다.

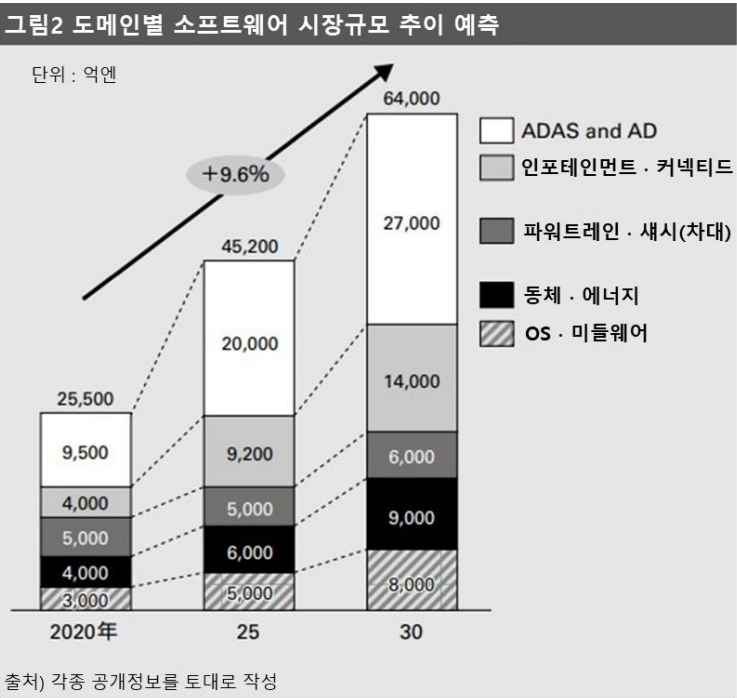
이후, 상기 구성 요소를 구체적으로 설명하겠다.

2. 개발 환경의 변화 : 소프트웨어 시장의 추이와 아키텍처의 변화

개발 환경의 변화를 이해하기 위해 우선 최근 큰 폭으로 확대된 자동차 관련 소프트웨어 시장의 전체상을 살펴볼 필요가 있다(그림2).

자동차 업계에서 소프트웨어 시장은 2030년에는 약 6.4조 엔 규모에 이를 것으로 예상되며, 20년부터 10년간 2배 이상으로 확대될 것으로 보인다. 분야별로 보면 ADAS and AD 영역(도메인)이 신장률과 시장 규모가 가장 크고 자동차 소프트웨어 시장의 약 절반을 차지하고 있다. 자율 주행이나 어드밴스 드라이브 시스템은 자동차 소프트웨어 개발 중에서도 아주 복잡한데, 통합해야 할 기능이 많고 소프트웨어로 제어하는 수가 필연적으로 많아 지기 때문이다.

다음은 커넥티드 서비스로 대표되는 인포테인먼트이다. 이것은 차량용 계기판이나 콕핏을 통해서 운전자에게 다양한 서비스를 제공하는 데 핵심이 되는 도메인이다. 이 영역으로 스마트폰 컴퓨터의 패권자인 Apple과 Google이 각각 「Apple Car Play」, 「Google Automotive Service」를 전개해서 시장을 견인하기 시작했고, 각국의 주요 자동차 OEM도 인포테인먼트 영역에서의 차별화, 기능 향상을 꾀하며 새로운 사용자 획득에 힘을 쏟고 있다.



마지막으로 주목해야 할 도메인은 OS/미들웨어이다. 소프트웨어 전체 중에서 구성비는 적지만, 필자는 자동차에서 OS/미들웨어가 수행하는 기능이 상당히 커질 것으로 추측한다. 기능 향상으로 인해 개발이 복잡해지면서 컴포넌트별 개별 관리에서 ECU^{주2}를 통합하고 도메인별 또는 주요 기능별 관리로 이행해야 한다. 따라서 하드웨어의 개수나 복잡성을 최대한 배제하여 중앙집중형 원스톱으로 관리되는 것이 바람직하며, OS/미들웨어를 도입함으로써 단번에 그 콘셉트를 실현할 수 있을 것이다.

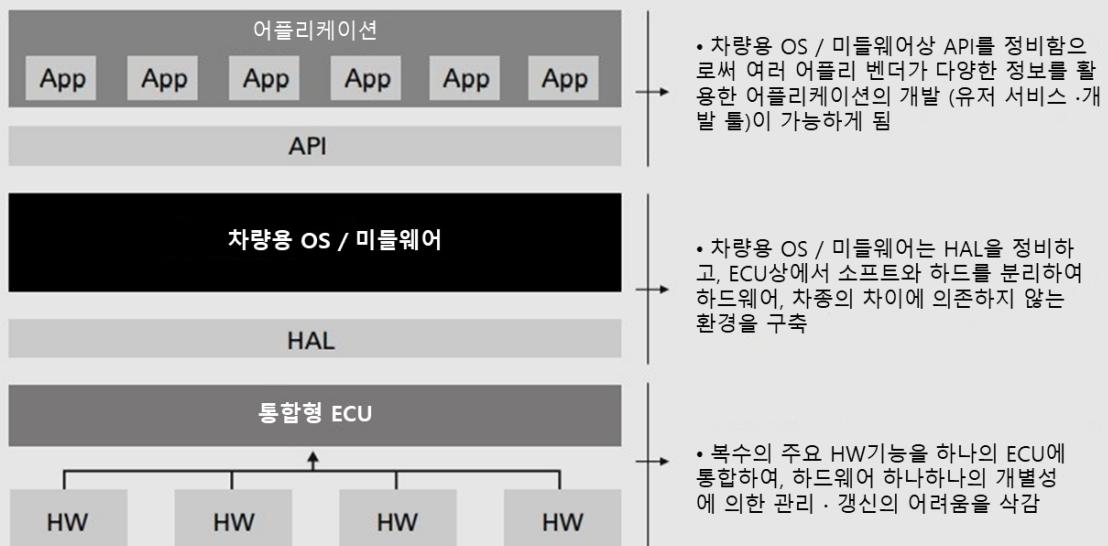
OS/미들웨어는 각 하드웨어 제품이 갖는 개별성이나 복잡성에 영향을 받지 않고 소프트웨어 개발을 실현하기 때문에 통상 HAL^{주3}을 가지는 경우가 많다. 즉 「하드웨어와 소프트웨어의 분리」를 실현할 강력한 도구가 되며 개발 효율화로 이어질 것이다. 조금 추상적으로 말하면 기존에는 자동차 도메인별로 몇 십대나 되는 PC가 탑재(ECU)되었는데 기능 향상을 위해서는 각각의 PC별로 소프트웨어를 갱신해야 했다. PC에 이상이 생겨 변경이 필요한 경우 동시에 소프트웨어도 갱신하는 등 두 차례의 수고가 발생하였다. 이러한 현상이 모든 차량용 PC에 적용된다고 하면 그 노력은 헤아릴 수가 없다. 그래서 개개의 PC를 강력한 장치로 성능이 높은 하나의 PC에 집약화해서(통합 ECU)그 PC에 OS/미들웨어를 탑재한다. 그러면 기존에는 각각의 PC별로

소프트웨어를 갱신했던 것을 OS/미들웨어에서 필요한 기능만을 일괄 갱신·변경이 가능해져 소프트웨어 개발 효율이 현격히 올라간다. 또 HAL을 도입하고 있어서 개념상으로는 어떤 차종이든(하드웨어 제약을 받지 않는다) OS상에서 자동차 도메인·기능별로 필요한 소프트웨어의 개발·갱신이 가능해진다.

이러한 OS/미들웨어를 내재화 개발하려는 추세는 최근 들어 활발해졌다. 자동차 OEM을 살펴보면, 폭스바겐은「vw.OS」, 다임러는「MB.OS」, 토요타자동차는「Arene OS(아린 OS)」콘셉트를 책정하고 이미 개발에 착수하였다. 그들은 OS/미들웨어와 그 위에서 작동하는 소프트웨어 툴을 개발해서 그 OS의 외판을 노리고 있으며 모든 차종에서 소프트웨어의 개발이 가능한 세계를 추구하고 있다. 또한 IT 거대 기업 Google도 「Google Automotive OS」를 전개해 이미 몇몇 자동차 OEM과 파트너십을 구축하였다(그림3).

토요타자동차와 다임러, 폭스바겐이 차량용 OS/미들웨어의 내재화에 발을 들인 배경에는 자동차 아키텍처의 대폭적인 변화가 있다. 현재 통상의 엔진 차량에서는 1대당 약 70개 이상의 ECU를 탑재하고 있는데, 최근에는 주행·멈춤·방향 전환 같은 기본 성능에 더하여 최신 ADAS나 크루즈 컨트롤, 자율주행 어시스트 시스템 등 급변하는 시대에 맞춰 기능이 추가되었다.

그림3 차량용 OS / 미들웨어 중심의 아키텍처



출처) 일본경제신문 니케이크로스틱 합동 취재반 「자동차거인 VS디지털 일인자」 (니케이BP, 2021년) 을 참고로 작성

기존에는 이처럼 매년 추가되는 최신 기능과 탑재된 하드웨어가 짝을 이루는 식으로 ECU를 추가 탑재했으나 그렇게 하면 그때마다 아키텍처가 복잡해져 이른바 스파게티형 구조가 되어버린다. ECU별로 소프트웨어 수정이나 동작 성능을 확인해야 해서 ECU 벤더가 하드웨어 일부를 변경하는 것만으로도 개발상 상당한 조정을 소프트웨어와 하드웨어 관점 양쪽에서 실시해야만 한다.

ECU와 이에 연동하는 하드웨어의 개수가 늘어날 때마다 개발 공수가 대폭 늘어나는 현재의 아키텍처는 한계에 다다랐다. 이것은 가령 파워트레인이 EV이고 탑재된 하드웨어 부품이 줄었다고 해도 같은 상황이다.

이런 흐름 속에서 자동차 OEM 업체는 현재 도메인 아키텍처 변경을 서둘러 추진하고 있다. 이것은 파워트레인, 새시(차대), ADAS and AD를 영역(도메인)별로 통합적으로 제어하는 방식이다. 그렇게 하게 되면 ECU의 탑재 수가 줄어들며 개발 효율화에 크게 공헌한다. 단, 도메인 간의 제어나 통합성 면에서는 과제가 남는다.

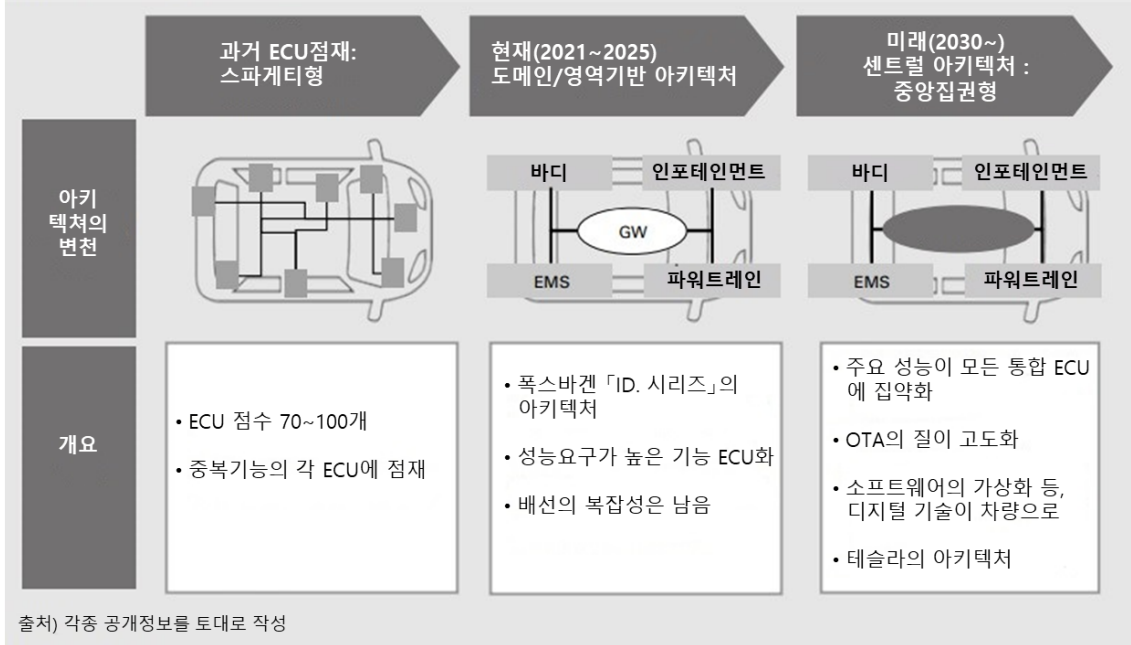
마지막 방식으로 도메인을 통합하는 중앙집중형을 들 수 있다. 이것은 소수의 고성능 ECU로 차량 전체를 제어하는 아키텍처인데 자동차 OEM 중에서는 테슬라만이 채용하고 있다.

테슬라는 고성능 통합 ECU로 차량 전체를 제어하는데 세 개의 Body Controller라고 불리는 차량 제어용 ECU를 통합 ECU 밑에 연결하였다. 이렇게 통합 ECU로 차량 전체 제어와 자동차 하드웨어 부분 제어의 연동성을 확보하였고, 이 Body Controller로 도메인 간 차량 하드웨어 부분을 일괄 제어하고 있다.

이와 같은 아키텍처의 변모 트렌드로서 ECU 탑재 수의 감소(이에 동반한 하드웨어 부분의 감소), 도메인별 또는 도메인 간의 통합 움직임으로 인한 아키텍처의 심플화를 들 수 있다. 이러한 배경에는 OTA^{주4} 기능 업데이트가 크게 연관되어 있다. 현재는 테슬라와 극히 일부 자동차 OEM에만 설치된 기능이지만 앞으로 OTA가 표준 탑재되어 각 기능이 무선으로 업데이트되면 아키텍처의 구조로 인해 OTA의 질이 크게 바뀔 것이다. 테슬라처럼 중앙집중식 구조로 탑재된 ECU 수가 적은 아키텍처라면 ECU 간의 호환성을 포함한 전자기판 전체의 동작 확인 등을 확인하는 수고도 덜 수 있게 된다. 이것은 소프트웨어를 갱신하고 OTA로 각 기능을 업데이트 시켜 사용자에게 그 가치를 전달하는 데 매우 중요한 요소이다.

반대로 기존의 아키텍처로 OTA를 실현하기란 거의 불가능하다. 소프트웨어 중심의 자동차 제

그림4 아키텍처의 변모



조 환경에서는 기능별로 소프트웨어를 계속해서 업데이트하고, 업데이트한 기능을 OTA로 자동차에 반영시키며, ECU를 통해 하드웨어 부분에 연동시킨다. 바로 이러한 부분을 실현하기 위해 아키텍처의 재검토가 지금 급속도로 진행되고 있다(그림4).

3. 경쟁의 변화 : IT패권자의 진입과 테슬라의 존재

본 장에서는 외부 환경의 변화로서 비자동차 OEM, 거대 IT 기업의 자동차시장 진입과 테슬라가 가져온 파급 효과에 대해서 설명하겠다.

최근 Alphabet산하의 Google, Waymo나 Zoox, 또는 실리콘밸리의 유니콘 스타트업 기업, 그리고 컨슈머 시장에서 우리에게 익숙한 Apple이 Apple Car를 개발하여 자동차 업계에 본격적으로 진출하는 등 비자동차 OEM인 IT 거대 기업의 자동차 업계 본격 진출을 암시하는 움직임이 심상치 않다.

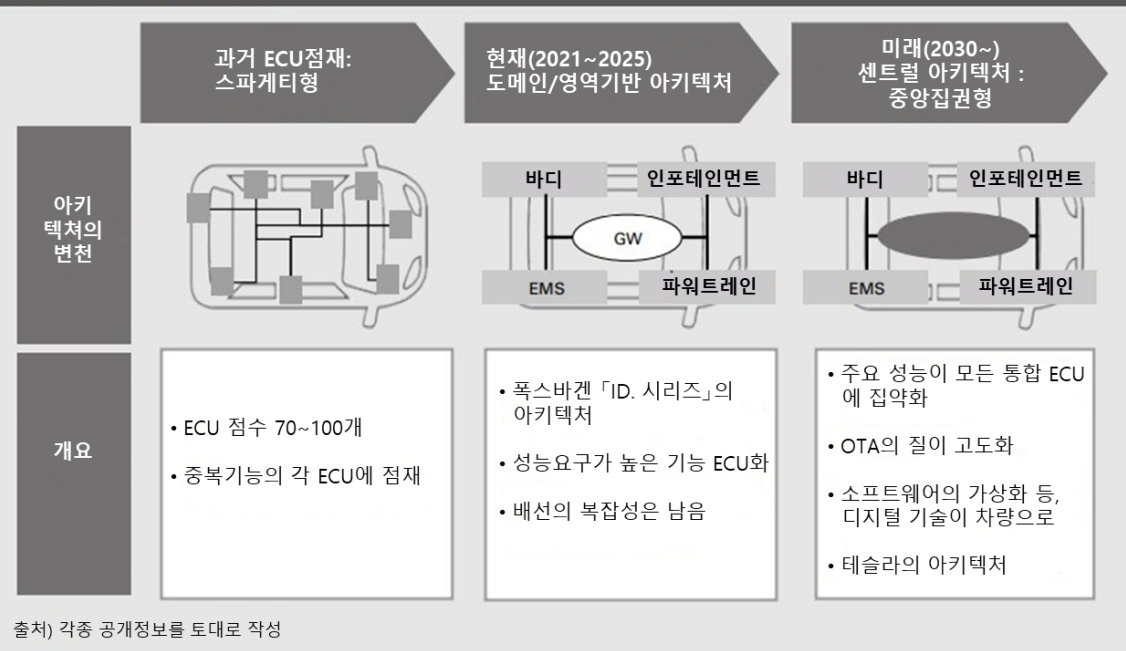
그들은 자동차 분야를 시장 규모의 크기·영향력, 자동차 사용자로부터 얻을 수 있는 정보·데이터가 가져올 잠재력 관점에서 주요 산업 중에서 마지막 프린티어라고 보고 있어 앞으로 치열한 경쟁이 예상된다. 더구나 자동차업계는 기존의 안전성과 시큐리티 담보의 관점에서 타업계에 비해 보수적인 산업이어서 변화의 속도도

IT 업계와 비교하면 느린 편이다. 그 틈을 노린 거대 IT 기업은 주로 스마트폰이나 PC를 비롯한 컨슈머 디바이스 시장에서의 성공 체험을 기반으로 자동차 업계로 영역을 확장, 나아가 제일 인자로서 신시장 창출을 호시탐탐 노리고 있다.

그들이 자동차업계에 본격적인 진출을 가속화하는 배경은 크게 두 가지 요소(자동차 커넥티드화의 발전과 자율주행 기술 개발에서 IT 패권자의 강점 응용)를 들 수 있다. 커넥티드화에 관련해 최근 차량 인포테인먼트의 충실화, 거기에서 파생하는 사용자 데이터를 서비스에 응용·활용하는 것이 하나의 트렌드가 되었다.

이 영역에서는 모바일 기기 패권자인 Apple과 Google이 진출하여 콕핏 영역에서 존재감을 높이고 있다. Apple은 「Apple Car Play」를 60개 이상의 자동차 브랜드, 600종 이상의 차종에 탑재해 iPhone에서 쓰던 기존 앱이나 기존 서비스를 콕핏으로 연계·활용할 수 있게 만들었다. Google은 GAS(Google Automotive Service)를 1,500만 대 이상의 차종에 이미 적용하였고, 「Google Maps」, 「Play Store」, 「Assistant」 등이 콕핏에 구현되었다. 그들은 디바이스 분야에서 키워온 강점인 「사용하기 쉽고 표준화된 UX」, 「애플리케이션의 높은 품질」, 「수집한 데이터를 활용한 기능의 업그레이드」 등 우선은 클라우드와 직접 연계할 수 있으므로 차량의 하드웨어

그림4 아키텍처의 변모



어 부분과 연계가 적은 멀티미디어, 콕핏 같은 영역에 초점을 맞추어 자동차 업계에 진출했다.

더욱이 Google은 Android Automotive OS (AAOS)라고 불리는 자동차 개발용 OS를 만들어 AAOS에서 각 자동차 OEM이 개별로 API를 설정해 HAL 설정으로 차량 제어 하드웨어와 연계해 개발할 수 있도록 했다. Google은 사용자와 직접적인 접점이 있는 콕핏을 시작으로 현재는 자동차 OEM의 개발자용 OS나 툴 제공 같은 영역에 손을 뻗고 있다. 이처럼 Google의 Tier1화가 현실성을 띠기 시작하며 자동차 OEM이나 기존 서플라이어 사이를 비집고 들어가려 하고 있다(그림 5).

다음으로 경쟁 변화의 큰 요인인 테슬라에 대해 설명하겠다. 테슬라는 두말할 필요 없는 유명한 EV 업체이지만, 소프트웨어 중심의 자동차 제조라는 시점에서 자동차 업체의 전통적인 사고방식에 한 획을 긋는 행보를 보였다. 앞서 말했듯이 테슬라는 최소한의 ECU로 구성된 중앙집중식 아키텍처를 채용해 OTA를 통해서 다양한 기능을 사용자에게 무선으로 연결하는 소프트웨어 중심의 자동차 제조를 실현하기 위해 환경 정비를 개발하고 나아가 비즈니스 모델의 관점에서 강화해 왔다.

개발 관점에서는 중앙집중식 아키텍처를 채용하고 최소한의 ECU와 하드웨어 제품을 탑재해 소프트웨어 업데이트로 OTA를 실현하기 쉬운 플랫폼을 정비해 왔다.

또 소프트웨어 기능을 충실히 하기 위한 테슬라의 의지는 하드웨어 부품에서도 엿보인다. 2018년 생산모델까지 테슬라는 자율 주행 제어의 SoC^{주5}영역에서 최첨단을 달리는 Mobileye나 NVIDIA 같은 서플라이어에 개발·제조를 위탁했다. 그러나 이 SoC 영역은 FSD^{주6}의 소프트웨어를 갱신할 때 가장 중요한 하드코어 부분이라고 인식을 전환해 이 영역의 개발 방침을 수정했고 19년부터는 SoC 개발의 완전 내재화로 방향을 틀었다. 이처럼 아키텍처의 심플화와 코어 하드웨어 영역의 완전한 내재화를 통해 하드웨어에서의 쓸데없는 소모를 철저하게 배제하고 소프트웨어 갱신으로 부가 가치를 추구하는 비즈니스 모델의 토대를 조성했다.

비즈니스 모델 관점에서 테슬라는 OTA로 소프트웨어를 업데이트할 때마다 사용자에게 과금하는 수익 모델을 만들어 냈다. 그림6에 제시되었듯이 테슬라는 판매 후 차량 모델별로 정기적으로 소프트웨어를 갱신하고 있다. 구체적으로는 4주에 1번, 각 기능의 메이저 버전업을 실시, 각 기능의 향상을 실현하고 있다. '2021.4'의 4는 해당년의 주차를 나타내고 도트(.) 이하의 번호가 계속되면 계속될수록 버그를 수정해 차종으로 마이너 릴리스를 롤아웃한 것을 의미하며, 반대로 주차 번호 이하의 도트 숫자 표기가 없는 경우는 메이저 릴리스한 뒤에 다른 차종이나 차량에 롤아웃하지 않은 것을 의미한다.

그림5 거대 IT 기업의 자동차 업계에서의 동향

		콕핏계	자율주행 기능계
아키텍처		어플리케이션, 미들웨어로 기본적으로 완결, 펌웨어 이하는 HAL 정의로 실현	센싱,인지, 어플리케이션, 아키텍처간의 긴밀한 통합이 필요
개발상 특징		OS 개발 로직이 완성되면, 어플리케이션 접합, HAL정의로 완결	각 레이어마다 독자의 고난도 개발을 진행, 그에 따른 인티그레이션도 복잡화
비즈니스 모델		OS를 무료로 전개하여 기존 서비스·어플리케이션에서 라이선스 수익으로 연결	OS로는 타사용 외판은 진행하지 않음
각사의 주요 동향	Google	서비스·어플리케이션에「GAS」를 전개. OS에 Google android automotive OS를 전개	별도 브랜드「Waymo」에서 자율주행 SW, Lidar 등의 센싱 기술을 개발
	Apple	서비스·어플리케이션에「Apple Car Play」를 전개	「드라이브AI」를 매수. 자율주행 SW 알고리즘 강화를 도모
	Amazon	서비스에「Alex Car play」를 전개	「Zoox」를 매수. 자율주행 SW 알고리즘 강화를 도모

어느 쪽이든 테슬라는 월 1회 이상의 페이스로 사용자에게 소프트웨어 업데이트를 준비해 필요에 따라 계약 서비스의 그레이드업이나 부품의 교환을 제공하고 있다. 이러한 소프트웨어의 갱신이 항상 사용자에게 과금을 유발하는 것은 아니지만, 사용자의 호기심을 부추겨 매력을 느낄 때마다 과금하게 만드는 소비자 심리를 꿰뚫어 본 시스템이다.

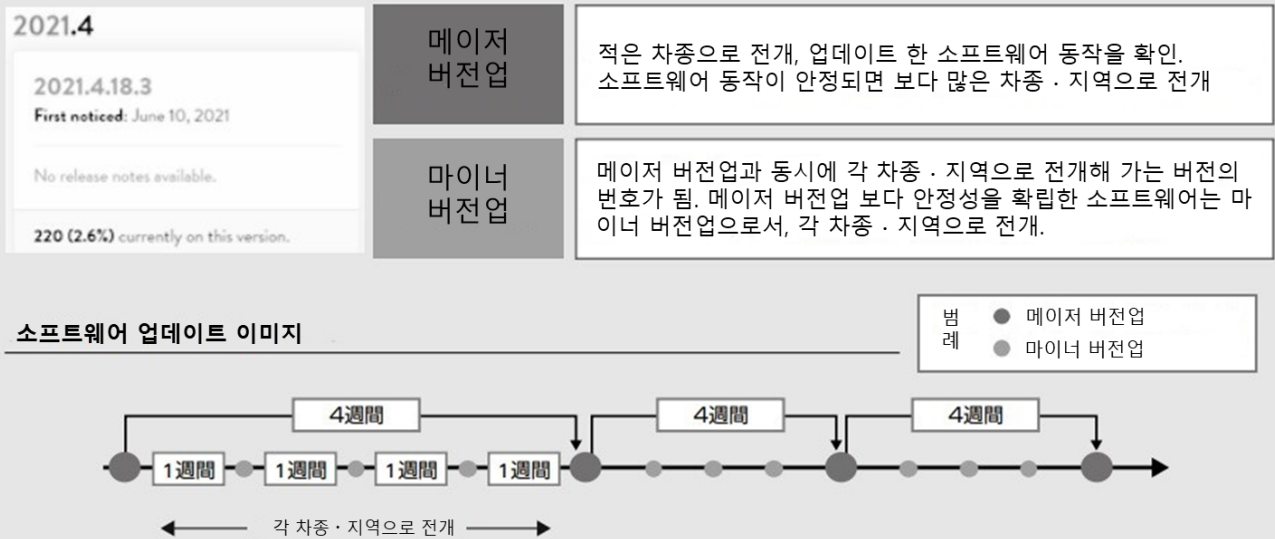
그 대표 사례로 불리는 기능이 FSD 업데이트 기능이다. 이 자율 주행 기능 소프트웨어 패키지는 이전 테슬라의 하이라인 모델인 S나 Y를 중심으로 구매 옵션 약 110만 엔의 추가 비용을 내야 탑재되었다. 그러나 현재 테슬라의 개발 양산 코스트가 내려가 모든 차종에 FSD 탑재가 가능해지며 2017년 7월부터는 옵션이 아니라 구독 서비스로 제공하기 시작했다. 현재 테슬라는 「Basic Autopilot to FSD capability」와 「Enhanced Autopilot to FSD capability」 두 종류의 옵션을 제공하는데, 각각 전자는 월 199달러, 후자는 월 99달러로 제공하고 있으며 정식으로 구독 서비스를 시작했다. 이 FSD 구독 서비스는 테슬라에게 크게 두 가지 의미가 있다. 첫 번째는 구독 서비스 도입에 따른 마스킹 효과로 FSD 탑재율이 높아지며 결과적으로 차량 판매 단가가 올라가게 된다. 원래는 차량을 구매할 때 100만 엔이 넘는 금액을 내야 옵션으로 들어갔던 기능이 월 100달러 정도로 서비스를

이용할 수 있게 되었다. 또 언제든지 취소할 수 있다는 구독의 이점으로 인해 결과적으로 원래 구매를 고려하지 않았던 유저층에게도 다가갔으며 테슬라는 그만큼 추가 수익을 확보하게 되었다.

또 다른 하나는 FSD의 구독 서비스 운용 개시를 계기로 본격적으로 「리컬링 비즈니스 모델 확립」의 길을 그려내기 시작했다는 점이다. 그림 7에서 제시했듯이 노무라종합연구소(NRI)의 독자적인 시뮬레이션에 따르면 FSD 탑재는 앞으로 구독형이 주류가 될 것이며, 2년 후에는 약 1,500억 엔(매출의 15~20%)이 구독을 통한 FSD 수익원이 될 전망이다. 이것은 종래 한 대의 차량 판매를 경쟁 원동력으로 삼았던 자동차 OEM에는 완전히 새로운 전략이며 앞으로 OPEX에서 이익을 내는 비즈니스 모델로의 전환을 의식하기 시작한 상징적인 이벤트이다. 이처럼 테슬라는 소프트웨어가 가져온 유저에 대한 지속적인 부가가치 제공을 경쟁력의 원천으로 삼아 아키텍처를 포함한 개발 기능 효율화나 리컬링 비즈니스 모델의 구축과 같은 요소를 정비해 왔다.

테슬라는 양산 차종 모델 3의 도입을 계기로 매년 순조롭게 판매 대수를 늘려왔다. 현재는 토요타자동차의 약 20분의 1에 불과하지만, 앞으로 동등 베이스에서 양산 체제를 확대하여 사용자의 저변이 넓어지면 토요타 자동차나 폭스

그림6 테슬라의 소프트웨어 업데이트



출처) 테슬라 공식 Web사이트 토대로 작성

바겐 정도로 대수가 늘어나지 않아도 앞서 서술한 OPEX형 비즈니스 모델 등으로 인해 자동차 OEM에는 충분한 위협이 될 수 있다. 이것은 Apple이 디바이스 점유율에서는 Android나 다른 PC 업체에 미치지 못하나 수익성이나 시가총액에서는 다른 기업들을 크게 웃도는 상황과도 중첩되는 부분이 있다. 앞으로 탄소중립이 중요해지면서 테슬라의 EV 전용 기업이라는 브랜드 가치가 소비자의 구매 동기를 유발할 가능성이 크다는 점도 주목해야 한다.

II. 소프트웨어 중심의 자동차 제조의 본질

1. 사용자에게 있어서 「소프트웨어 중심 자동차 제조」의 본질 : 새로운 가치 체험과 감동, 그리고 환경 가치의 창출

현재 자동차 개발은 하드웨어와 소프트웨어가 일체화되어 이루어지는 것이 일반적이다. IT가 급속하게 변화·진화를 거듭하는 지금, 소프트웨어 발전 속도가 하드웨어를 훨씬 앞서고 있어, 지금까지의 방식으로는 소프트웨어 발전을 따라가지 못하니 상품이 출시될 때는 세상에 뒤처지고 시대에 뒤처지는 일도 드물지 않다.

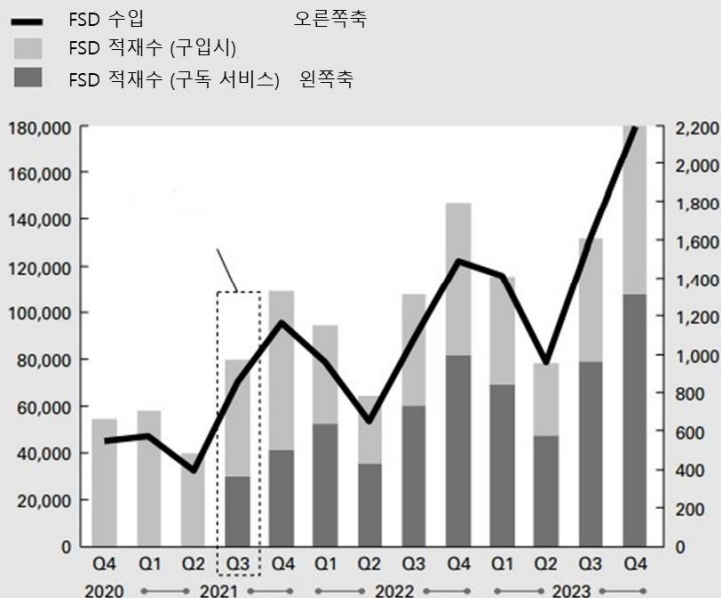
예를 들어 카 내비게이션 시스템은 이전에는 차

차량용 기기로 「카 내비게이션」이 주류를 차지하며 상품의 가치 향상을 선도해 왔으나 요즘은 스마트폰 앱이 그 자리를 대신 차지하며 빠르게 발전하는 IT를 차례차례 구현하면서 업데이트하는 것이 사용자에게는 당연한 일이 되었다.

스마트폰과 마찬가지로 요즘은 자동차에서도 소프트웨어가 상품력·상품 가치를 좌우하는 중요한 요소가 되었다. 상품이 다양한 정보와 연결, 연계되어서 상품 단위에는 없는 새로운 가치나 체험, 그리고 감동을 만들어 내는 것은 그야말로 소프트웨어 중심의 자동차 제조가 실현하는 새로운 세계관이다. 그것을 실현하는 것이 「소프트웨어」와 「커넥티드 기술」이다.

필자는 이전 자동차 열쇠를 스마트폰 앱으로 대체하는 「Digital KEY」사업 프로젝트에 참여한 적이 있다. 자동차 열쇠라는 하드웨어가 소프트웨어 중심의 자동차 제조라는 새로운 가치 시점으로 재정의된 것이다. 열쇠를 소프트웨어화하여 스마트폰 앱으로 만들어 자동차와 일상을 심리스(seamless)로 연결해 때에 따라 자동차 열쇠, 집 열쇠, 사무실 열쇠가 된다. 모든 사물의 열쇠가 될 수 있으며, 전달도 스마트폰으로 그 자리에서 즉시 간단하게 주고받을 수 있는, 지금까지 없었던 새로운 체험 가치이다. 이러한 실제 경험을 바탕으로 자동차가 일상과 연결되는데, 이른바 소프트웨어 중심의 자동차 제조를

그림7 테슬라의 OPEX형 비즈니스 모델로의 전환



출처) 각종 공개자료를 독자 분석하여 작성

1 FSD적재율의 큰폭 상승

- 구독 서비스 제공으로 140,000달러 이상하는 FSD 옵션의 심리적인 구입 허들이 낮아짐
- 상기의 결과 FSD 적재율·대수가 올라감

2 모든 규격에 FSD를 표준 탑재

- 구독 서비스 플랜에 따른 FSD 탑재율 향상으로 인해 FSD를 전 모델에 표준 탑재 개시
- 상기에 따라 하드 탑재·조달 코스트가 내려가고 대수당 OP%가 향상

3 신규규격의 가격은 인상

- FSD의 메이저 업데이트에 따른 탑재단가 인상 : 과거 5,000달러->6,000달러->7,000달러->8,000달러, 2020년 Q3부터 Q4에 걸쳐 8,000달러->10,000달러로 인상
- 지불총액이 올라도 버전업과 구독에 의한 마스킹 효과로 수요는 내려가지 않음

통해서 여전히 새로운 가치, 그리고 감동을 창출할 수 있다고 확신한다.

소프트웨어 중심의 자동차 제조는 탄소중립 사회 실현에도 공헌할 수 있다. 즉 자동차의 에너지 절약, 자원 절약의 추진이다. 예를 들어 HEV(하이브리드 차)/PHEV(플러그인 · 하이브리드 차)는 엔진과 모터를 번갈아 사용하며 제어를 고도화함으로써 더 나은 친환경 자동차로 발전하는 것이 가능하다. 주행 장소나 시간 등에 따라 실시간으로 엔진과 모터의 전환 제어를 수행하는 「지오펜스 기술」을 탑재함에 따라 한결 환경에 좋은 자동차가 된다. 여기서 중요한 것이 제어를 최적으로 수행하는 소프트웨어와 커넥티드 · 데이터이다.

토요타자동차가 2021년 8월 25일에 발표한 「토요타의 자동차 제조를 위한 신념과 미래의 도전」이라는 주제로 집행 임원 CPIO 야마모토 게이지가 발표한 프레젠테이션에 따르면 「HEV는 주행 시간의 절반은 엔진이 정지하고 있으며 그것이 PHEV에서는 80%나 된다(일본의 경우)」고 한다. 소프트웨어 및 커넥티드로 더 세심하게 연비 · 전비를 고도로 제어하면 다양한 자동차가 친환경에 이바지하게 된다. 또 커넥티드를 통해 자동차에서 모은 데이터를 해석하고 자동차의 상태를 진단하는 서비스는 단순히 안심하고 자동차에 계속 타게 만드는 것만이 아니라 자원 절약, 순환형 사회의 실현에 크게 공헌할 것으로 기대된다. 구체적으로는 진단 결과에 기반한 적절한 메인テナンス를 수행함으로써 자동차를 보다 오래 사용할 수 있으며 배터리의 리사이클성이나 리유스성을 높이는 것도 가능해진다.

하드웨어의 강점은 소프트웨어 중심의 자동차 제조를 추진하는 데에 있어 더욱더 활용될 것으로 보인다. 소프트웨어에서 기능이 추가되면서 하드웨어의 사용 기간은 더 길어지리라 예측된다. 이 경우에는 「내구성 향상」이나 「교환 부품 입수의 편의성」 그리고 「수리의 편의성」이 요구된다. 그야말로 소프트웨어 중심의 자동차 제조로 인해 일본 제조업의 강점이 다시금 발휘되며 경쟁력 있는 가치 창출로 이어지지 않을까.

어느 쪽이든 하드웨어 단위만으로는 실현하기 어려운 탄소중립 사회를 위한 자동차를 현실화

하는 데에도 이 같은 자동차 제조가 중요한 역할을 맡을 것이다.

이제까지의 내용을 토대로 사용자에게 소프트웨어 중심의 자동차 제조의 본질이란, 일상과 자동차가 연결되어 새로운 체험 가치와 감동이 생성되는 것이라고 말할 수 있다. 매력적인 서비스와 앱의 고도화 · 진보가 이루어지고 일상과 자동차가 무선으로 연결되며 자동차에서의 시간과 생활이 점점 즐겁고 쾌적해진다. 게다가 지구 환경에도 좋은 영향을 미치고 좀 더 기대되는 미래를 꿈꾸게 됨으로써 사용자에게 큰 임팩트를 안겨줄 것으로 기대한다.

2. 업체/개발 관점에서 「소프트웨어 중심 자동차 제조」의 본질 : 자동차와 최신 IT의 융합

기존의 자동차에 탑재된 「임베디드 소프트웨어」 개발은 대상이 되는 하드웨어가 정해져 있으며 그 성능이나 코스트 절감을 극대화하는 것이 중심이 되었다. 이른바 하드웨어를 전제로 한 소프트웨어 개발, 맞춤 개발이 당연시되었다. 그러나 자동차 성능이나 부가 가치 향상을 위한 대응과 함께 소프트웨어의 개발 규모, 예를 들어 커넥티드의 행수나 프로그래머 수가 폭발적으로 증가 · 확대되고 있는 것은 많은 기사나 문헌에서 드러난 바와 같다.

게다가 임베디드 소프트웨어 개발에서도 최근에는 기술 혁신으로 인해 툴류나 개발 환경에서 IT 활용이 이루어지고 있다. 그러나 AI나 블록체인과 같은 새로운 IT의 활용, 사이버 보안 대응이나 빈번하게 업데이트되는 소프트웨어의 활용, 나아가 네트워크 경유로 접속되는 서버나 클라우드 연계에 대한 대응 등과 같은 최첨단 소프트웨어와 서비스의 활용은 좀처럼 진척을 보이지 않았다. 지금까지 하드웨어를 중심으로 했던 제조업의 한계에 직면했다고 말할 수 있다.

이러한 흐름 속에서 소프트웨어 중심의 자동차 제조를 통해 제조업의 전환을 꾀하고, 더불어 하드웨어와 소프트웨어를 분리해 널리 이용되고 있는 IT나 클라우드 기술이 임베디드 소프트웨어 개발에 빠르게 적용되기를 기대하고 있다. 예를 들어 클라우드의 활용(컴퓨팅 자원의 동적 할당)이나 엣지와 클라우드의 연계 등 자동차

소프트웨어 개발에서도 최첨단 IT가 적용될 것으로 보인다. 또 소프트웨어 중심의 자동차 제조를 실현하기 위해서 중요한 부분이 비히클 OS라고 불리는 「차량용 소프트웨어 기반(차량용 OS)」이다. 토요타 그룹의 Woven Planet사가 개발을 추진하는 「Arene OS」나, 폭스바겐의 「vw.OS」 등이 대표적이다.

차량용 OS가 구현되면 하드웨어와 소프트웨어가 분리되어 소프트웨어 개발은 더욱더 효율적이고 생산적이 되며, 품질은 향상되고 개발 리드 타임의 단축을 꾀할 수 있다. Apple의 「iOS」나 Google의 「Android」 같은 스마트폰의 소프트웨어 기반(OS)처럼 전 세계의 개발자가 차량용 OS를 개발하는 것이 가능해졌다. 또 개발한 소프트웨어는 동일한 차량용 OS를 탑재한 차량이라면 기기나 하드웨어에 상관없이 사용할 수 있다. 그리고 스마트폰이나 클라우드 기술, 사이버 시큐리티 기술, AI 기술이나 FinTech 등 최신 IT를 차량용으로 적용함으로써 다양한 차량용 소프트웨어를 효율적으로 개발할 수 있다. 커넥티드나 OTA로 인해 소프트웨어는 항상 최신 상태로 업데이트되어 사용자는 늘 안심하고 이용할 수 있다. 결과적으로 많은 사용자를 가진 자동차 업체라면 더 큰 스케일 메리트나 파급 효과를 누릴 수 있다. 바야흐로 앞으로가 소프트웨어 중심의 자동차 제조의 진가가 발휘될 때이다.

업체나 개발 관점에서 소프트웨어 중심 자동차 제조의 본질이란 이 세상의 기술 혁신을 자동차로 받아들이는 것, 항상 최신 IT를 자동차에 적용하는 것이 아닐까. 이를 실현하려면 차량용 OS의 역할이 매우 중요하며, 개발 측에는 개발의 효율화뿐만 아니라 새로운 가치 창출과 같은 본래의 목적에 집중해야 한다. 자동차 업체가 경쟁해야 할 부분이 하드웨어에서 소프트웨어로 이행하는 시기가 이미 코앞으로 다가왔다.

3. 소프트웨어 중심 자동차 제조의 핵심은 「소프트웨어 개발의 내재화」

자동차 OEM이 차량용 OS 개발에 힘을 쏟는 것은 소프트웨어 개발이 방대하고 복잡해 자동차 제조의 보틀넥이 되었기 때문이다. 소프트웨

어 개발을 추진하는 기존의 사내 조직에서는 서플라이어나 소프트웨어 전업 벤더에 「외부 위탁」하는 케이스가 일반적이었다고 한다. 또 지금까지는 개발 코스트 절감이나 효율화를 중시한 나머지 구태여 사내에 실질적인 소프트웨어 개발 조직을 만들 필요가 없다는 생각이 강했다. 심지어 기술적인 실현성에 맞게 의사를 결정하고 가능한 개발을 그저 수행하는 데 그치는 자기를 형편에 맞춘 소프트웨어 개발이 추진된 케이스도 있다. 이런 식으로는 소프트웨어 중심 자동차 제조는 도저히 실현될 수 없다.

필자는 소프트웨어 중심의 자동차 제조를 실현하는 본질적인 요건은 「소프트웨어 개발의 내재화」라고 생각한다.

내재화란 소프트웨어를 포함한 제품의 기획부터 설계·개발, 운용까지 전부 사내 조직에서 수행하는 것이다. 앞서 다뤘던 소프트웨어를 중심으로 자동차를 만들겠다고 선언한 토요타자동차도 「원리 원칙을 철저히 지키며, 자동차 제조에서 중요한 요소는 먼저 스스로 해보고 내재화를 지향할 것」, 「소프트웨어나 커넥티드 기술도 내재화를 추진할 것」, 「토요타에는 자동차를 만들면서 오랜 시간 이어져 내려온 토요타의 기본자세」(토요타자동차)라는 말에서 알 수 있듯이 소프트웨어 개발 내재화의 중요성을 강하게 주장하고 있다. 개발 리드 타임이나 개발비에서 소프트웨어가 상당 부분을 차지하고 있는 현 상황에서 자동차 업체들이 소프트웨어 개발을 근본적으로 효율화하기 위해 차량용 OS를 내재화하고 경쟁력을 높이려는 노력은 매우 합리적이라고 생각한다.

소프트웨어 개발을 내재화 했을 때 가장 큰 이점은 속도이다. 불량이 발생했을 때나 사용자에게서 피드백이 오는 경우 등 사용자가 계속 이용하게 만들려면 신속한 대응이 매우 중요하다. 스마트폰이 이미 너무나 당연한 현실 속에서 신속한 대응은 이제는 숙명이라고 말해도 과언이 아니다. 이러한 상황에서 외부에 위탁하면 본질적인 것이 이해될 수 없고 오히려 기대에 어긋날 가능성마저 있다.

내재화의 또 다른 중요한 이점으로 「노하우의 축적」을 들 수 있다. 소프트웨어 개발은 가설과 검증의 반복으로, 특별한 방법이 있는 것이 아

니다. 수많은 실패를 겪으며 거기서 배우고 깨달음을 얻어서 작은 성공을 계기로 기회를 잡는다. 그렇게 얻은 배움과 깨달음 자체가 무엇과도 바꿀 수 없는 귀중한 지식이자 자산이다. 이것을 외부에 위탁해버리면 가장 중요한 지식이 사내에 쌓이지 않을 뿐더러 외부에 축적되어 버린다. 스펙이나 운용상 주의사항 등 그런 노하우는 공식적으로 문서화된다고 하지만 본래 진짜 노하우는 문장으로 남길 수 없는 부분에 있다.

실제로 토요타 자동차는 소프트웨어 개발의 내재화를 강력하게 추진하고 있다. Woven Planet 사, 토요타 커넥티드, 해외 개발 거점을 포함해 글로벌에서 3,000명 규모의 소프트웨어 개발 체제를 구축했고 그룹 전체 규모로는 1만 8,000명에 이른다고 한다. 지금까지와는 차원이 다른 새로운 단계로 진입을 시도하고 있는 참이다.

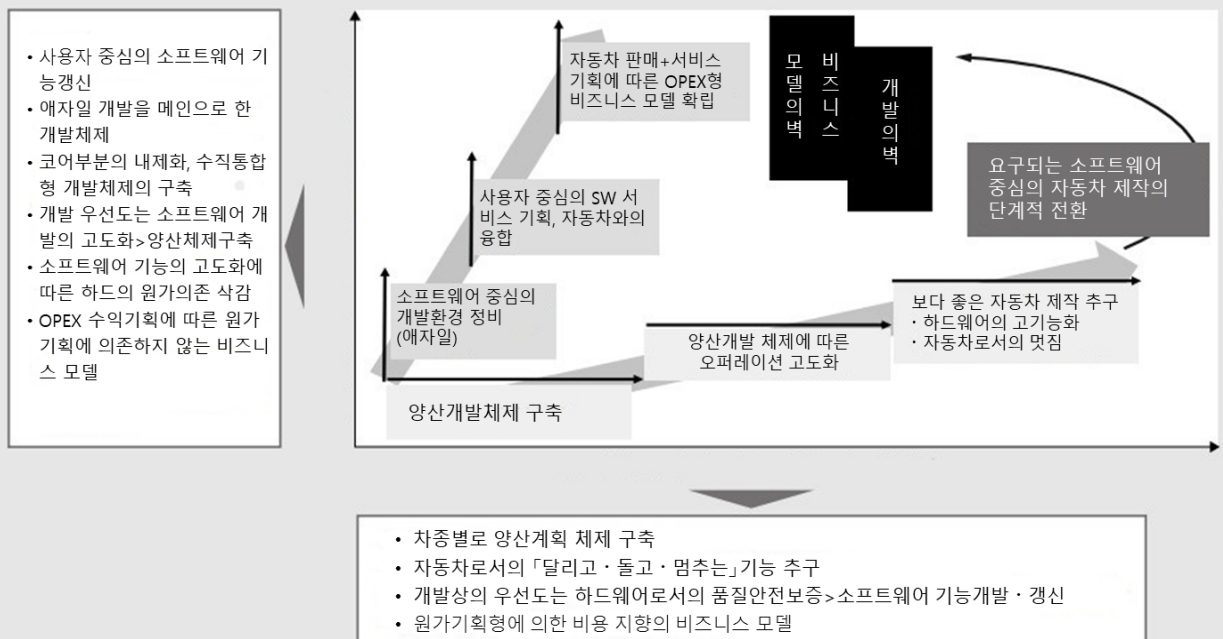
Ⅲ. 새로운 자동차 제조 실현을 위해 일본계 자동차 OEM에 부과된 과제 : 두 가지 장벽의 존재/개발 장벽, 비즈니스 모델 장벽

이번 장에서는 새로운 자동차 제조를 실현하는데 필요한 기능과 과제에 관해서 일본계 자동차 OEM의 현 상황을 짚어가며 해설하겠다.

일반적으로 일본계 자동차 OEM은 마케팅 콘셉트로서 안전 기능·품질을 최대한 높이면서 가격을 일정 규모로 제한하는 염가 양산형을 추구했다. 각사는 다양한 차종을 각국에서 각각의 세그먼트 니즈와 전략에 맞추어 전개해왔다. 그러면서 일본계 자동차 OEM은 차종별로 최적화를 꾀하고 개발 기획과 양산 체제를 그때그때 개선하며 그 기능을 업그레이드했다. 예를 들어 토요타자동차는 더 좋은 자동차 제조를 콘셉트로 내걸고 개발과 생산팀이 함께하는 체제를 구축하려고 노력했다. 이러한 움직임은 차종별로 더 좋은 자동차를 일정 규모를 가진 시장에 도입하는 데 매우 중요하며 일본계 자동차 OEM이 오랜 시간 키워온 경쟁 기반이기도 하다. 그림 8에서 제시한 것처럼 일본계 자동차 OEM은 차종별 개발 체제를 경쟁력의 주축으로 구축해 수년마다 재검토하면서 스탭업을 도모했다.

한편 테슬라 같은 신흥업체는 사용자 중심의 소프트웨어 개발로 인해 발생하는 부가가치가 중추이며, 이러한 부가가치를 볼륨존으로 규모를 확대하기 위한 양산 체제 구축에서 트라이&에러를 반복했다. 즉 테슬라는 소프트웨어를 중심으로 한 자동차 개발을 시발점으로 하는 한편, 일본계 자동차 OEM은 양산 체제의 확보를 중심으로 한 하드웨어를 시발점으로 하고 있어 양자의 전략에 대한 사고의 차이가 현저하다.

그림8 소프트웨어 중심의 자동차 제작과 하드웨어 중심의 자동차 제작의 차이



이런 상황 속에서 앞서 언급한 소프트웨어 중심 자동차 제조의 거센 파도로 인해 일본계 자동차 OEM은 전략 수정이 시급해졌다. 구체적으로는 엑스축에 있는 하드웨어 중심의 자동차 제조에서 와이축에 있는 사용자를 의식한 소프트웨어 중심의 자동차 제조로 이행하는 것이 요구되고 있다. 그렇다고 해서 인재를 포함해 지금까지의 경쟁 기반을 전부 대체해 무리하게 와이축으로 이행하는 것은 이제까지 오랜 기간 키워온 경쟁 기반의 관점에서 현실적이지 않다고 그들은 생각한다. 그래서 필자는 엑스축의 발전을 전통적인 베이스에서 추진하면서 서서히 중점을 소프트웨어로 옮기는 것이 중요하다고 제안한다. 다음은 차세대 자동차 제조로 단계적인 이행을 도모하는 데 있어 앞길을 가로막는 두 가지 장벽에 대해서 언급하도록 하겠다.

1. 첫 번째 장벽 : 애자일 개발로 이행

자동차 제조를 살펴보면 보디, 샤시 등의 최신 기술을 탑재한 선진 안전 기능 등 다양한 기능을 조직·부문별로 개발하고 최종적으로 개별 기능을 통합해 1대의 차량으로 조립된다. 기존의 자동차 제조 프로세스에서 대규모 모델 변경을 진행할 때는 3~5년이라는 중장기가 걸리며 개발 기획 제안에서 구현하는 하드웨어·소프트웨어의 시뮬레이션, 기획의 콘셉트를 시뮬레이션하는 CV(Concept Vehicle) 마일스톤을 설정하고 그에 맞추어 단계적으로 기능을 정비하였다. 그리고 최종적으로 각 공장에 생산을 위탁하는 양산 체제로 완전히 이행한다.

이러한 양산 프로세스까지 일련의 기간 동안 자동차 OEM은 일반적으로 워터폴 개발 방법을 고수했다. 워터폴 개발은 톱다운 방식으로 사전에 달성해야 할 차량 전체의 요건을 정의한 다음 「요건 정의 → 각 개발 기능·도메인 분해 → 각 개발팀·부문의 사양 결정 → 개발 구현」이라는 순서에 따라 단계적으로 수행된다. 워터폴 개발의 특징은 정해진 마일스톤까지 차량 전체의 사양을 결정할 필요가 있는데, 그것은 일반적으로 차량 판매 2~3년 전으로 거슬러 올라가야만 한다. 자동차 개발에서 자동차 OEM은 다양한 서플라이어에 RFQ^{주7)}를 발행하고 각 개발

기능을 다양한 도메인으로 위탁하기 때문에 일단 전체 일정이 정해지면 일정을 일부 변경·갱신하는 것은 거의 불가능하다. 이것은 앞으로도 워터폴 개발에서 간과할 수 없는 특징이라고 할 수 있다.

한편 GAFA를 비롯한 사용자 중심의 서비스를 잇달아 생성하는 조직은 애자일 개발 방법을 적용한다. 애자일 개발은 Agile(민첩한, 기민한)이라는 어원처럼 소단위에서 구현과 테스트를 반복하는 개발 방법이다. 또 그 중에 스크럼이라는 개발 프레임워크가 존재해 개발 테마별로 소수의 팀을 구성해 각 팀에서 주로 2주 단위로 트라이&에어를 거듭하면서 그 프로세스를 추진한다. 워터폴 개발과 달리 모니터링, 개선하는 기간이 짧은 주기여서 사용자의 니즈가 끊임없이 바뀌며 그때마다 방침을 전환하게 되는 모바 일이나 인터넷 소프트웨어 개발에 주로 이용되는 방법이다(그림9).

지금 이 워터폴과 애자일을 둘러싼 자동차 업계의 개발 프로세스 그 자체에 대한 재검토가 요구되고 있다. 과거의 하드웨어 중심 세계에서 완성되는 자동차 개발 프로세스였다면 워터폴 개발 방법으로 충분히 기능을 했다. 그러나 앞서 말한 것처럼 자동차의 장비 기능 특히 소프트웨어 면에서 탑재 기능의 고도화가 요구되면서 개발 프로세스가 복잡해지고 다공정을 거쳐 변화가 요구되는 지금의 자동차 개발 프로세스에서 워터폴 개발만으로는 대응이 어려워졌다.

그 주요 이유로 ① 차량 개발을 위해 수년 전에 전체 사양을 확정해야 하므로 몇 년 뒤에 사용자 니즈를 반영한 사양으로 정의하기 어려움, ② 한번 전체 일정이 확정되면 개발 프로세스가 불가역적으로 진행되기 때문에 개발 도중이나 시뮬레이션 단계에 부상한 문제에 대한 유연한 대응이 거의 불가능과 같은 요소를 들 수 있다. 이대로는 CASE 시대에 기능 부전이라고도 할 수 있는 문제가 반드시 발생할 것이다. 왜냐하면 10년 후의 자율 주행을 중심으로 한 커넥티드 서비스의 잠재적인 니즈를 현시점에 정확히 파악하는 것은 거의 불가능하기 때문이다. 그럼 모든 프로세스를 애자일로 전환하면 좋으나 하면 그렇게 간단한 문제가 아니다. 샤시나 스티어링, 액추에이터와 같은 하드웨어 부품은 몇년

이 지나도 대폭적인 사양 변경을 요구할 가능성이 작고 최종적으로 차량을 양산화하는 공정에서는 워터폴 개발 쪽이 압도적으로 관리하기 쉽기 때문이다.

CASE 시대에 소프트웨어 중심의 자동차 제도가 요구하는 기능에 대해서는 애자일 개발이 필요하지만, 거기서 문제는 애자일 개발과 워터폴 개발 두 가지 자동차 개발 프로세스를 얼마나 구분해서 쓸 수 있는가 하는 점이다. 이론적으로는 소프트웨어 기능과 이에 밀접하게 연관된 하드웨어는 애자일 개발을 중심으로 프로세스를 추진하고 소프트웨어 작용에 영향을 주지 않는 하드웨어는 워터폴 개발로 진행하면 된다. 그러나 여기에는 몇 가지 제약이 존재한다.

첫 번째는 애초에 자동차 OEM 중에 애자일 개발의 문화나 인재가 충분히 양성되어 있지 않고, 소프트웨어 기능 자체를 워터폴로 개발할 수밖에 없는 환경에 놓여있다는 점이다. 애자일 조직을 구성했어도 인재 · 매니지먼트 관점에서 조직으로서 충분히 기능하지 않을 수 있다는 사태가 우려된다.

두 번째 제약 요건은 애자일과 워터폴 개발 기간의 차이를 들 수 있다. 앞서 말한 바와 같이 애자일 개발은 2주 단위로 스크럼을 돌리면서

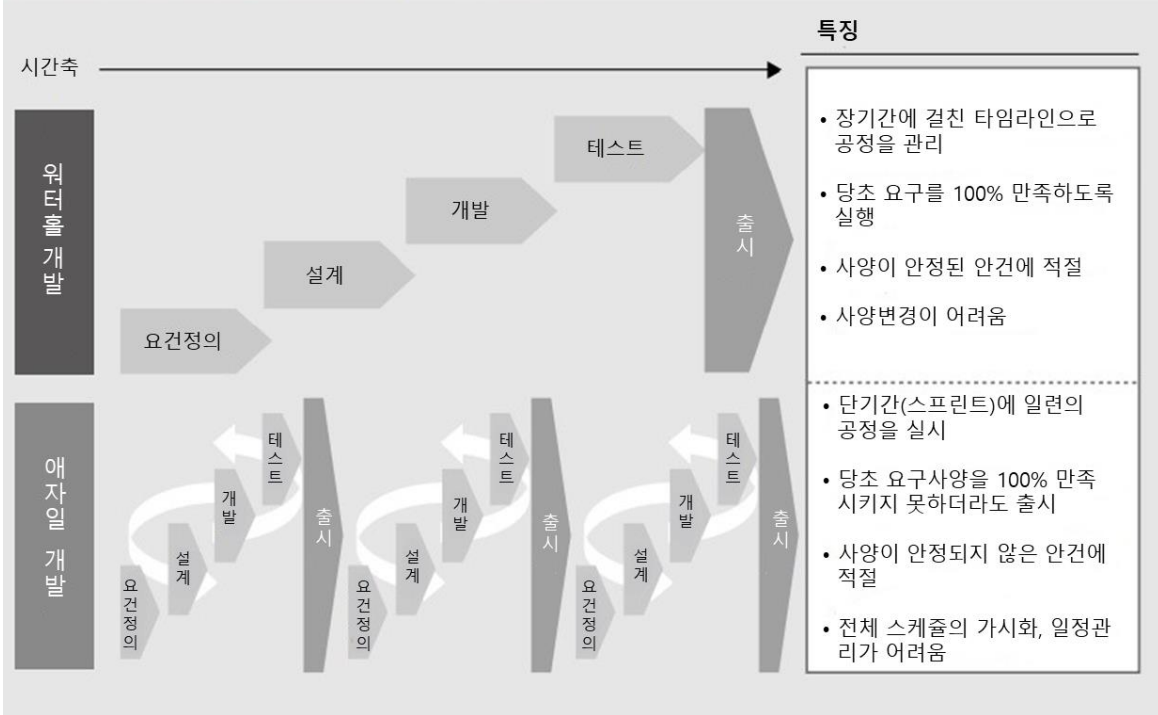
영속적으로 그 프로세스를 진척시켜 필요에 따라 대폭적인 프로세스 수정도 이루어진다. 한편 워터폴 개발은 수년이라는 개발 기간 동안 미리 설정된 마일스톤에 대해 개발 구현을 진행한다. 차량의 하드웨어 파트를 워터폴로 추진한다고 했을 때 애자일로 개발한 기능을 언제 어느 시점에 넣어야 할지 같은 양자 간의 일정, 마일스톤의 조정이 매우 어렵다. 이것은 자동차에 탑재하는 하드웨어 수가 매우 많고, 안전성, 품질 관리와 같은 관점에서 양산 일정에 놓기까지 절대로 변경할 수 없는 몇몇 마일스톤이 존재하기 때문이다.

이런 상황에서 소프트웨어 관련 도메인을 애자일 개발 중심의 프로세스로 이행하면서, 소프트웨어 개발의 영향을 받지 않는 기존 하드웨어 파트 개발 사이에서, 애자일 개발과 워터폴 개발을 적절히 구분해 적용할 수 있을지가 앞으로 큰 과제로서 자동차 OEM을 압박하게 될 수도 있다.

2. 두 번째 장벽 : OPEX형 비즈니스 모델로의 이행

다음 장벽으로 기존의 자동차 판매와 비즈니스 모델의 차이를 들 수 있다. 이것도 개발과 마찬가지로

그림9 워터폴 개발과 애자일 개발의 차이



가지로 소프트웨어 중심의 자동차 제조와 기존의 자동차 판매의 사고방식이 매우 큰 차이를 보이는 분야이다.

자동차 OEM에는 원래 차종별로 개발·엔지니어링팀이 존재해 해당 차종의 판매 가격을 어느 수준으로 설정하고 각 지역에서 몇 대를 판매할지와 같은 지표의 설정을 최우선 과제로 삼았다. 이것은 앞서 말한 바처럼 저렴하면서 고품질의 차량을 대량 생산하는 것은 일본계 자동차 OEM이 뛰어나므로 CASE 도래 이전에는 이러한 지표 아래에서 각 프로세스 오퍼레이션을 개선해 나감으로써 어느 정도 점유율을 구축할 수 있었다. 또 각 차종의 대수를 확대하려면 가격 설정이 굉장히 중요한데 각사들은 얼마나 원가를 낮게 설정할 수 있을지 원가 기획에 주력했다.

그러나 CASE 시대와 함께 테슬라의 대두로 인해 자동차 업계에서 판매 중심 비즈니스 모델이 맥을 못 추고 있다. CASE 시대에는 자동차 밸류체인의 저변이 판매 후에도 확장되며 판매 후의 차량을 주축으로 각 기능을 업데이트하며 서비스로서 비용을 회수하는 이른바 리컬링 비즈니스 모델로 이행하고 있다.

앞서 말했듯이 테슬라는 판매 후 차량에 대해 OTA로 기능을 업데이트하면서 구독형 OPEX 비즈니스 라인으로 수익을 늘리고 있다.

그러나 일본계 자동차 OEM이 테슬라처럼 OPEX에서 수익 라인을 늘리는 것은 쉽지 않다. 그 이유로서 ① OPEX로 수익 기획에 착수하고 있지 않으며 판매 후의 유스케이스를 포함한 비즈니스 모델 기획이 결여되어 있는 점, ② 미래의 잠재적인 니즈가 예측되는 OTA 업데이트를 기획하려면 ECU를 비롯한 탑재할 하드웨어의 스펙을 올릴 수밖에 없으며 그 결과 원가가 오른다는 딜레마에 직면하는 점을 들 수 있다.

①에 관련해 실제로 자동차 OEM 각사는 신차 차종의 커넥티드화를 서두르고 있다. 토요타자동차는 「T-Connect」를, 닛산 자동차는 「Nissan Connect」를 전개하는 등 각사 커넥티드 서비스의 라인업을 정비하고 인포테인먼트 영역을 중심으로 오퍼레이터 서비스나 부분적인 결제 서비스의 구현에 착수하고 있다. 그러나 이는 어떤 의미에서 신차 구매 후의 서포트라는 틀에서

크게 벗어나지 않으며 OTA를 통해 자동차 일부가 업데이트되는 것으로 그 기능을 구독으로 체감하는 참신함은 테슬라에 비해 뒤떨어진다. 또 이 기획은 어디까지나 신차 차종을 위해 추가 설치된 기능의 연장선상에 머무르며, 테슬라나 Apple처럼 과거 모델을 포함한 모든 차종과 디바이스에서 자사의 플랫폼을 통해 회수하는 모델과는 크게 다르다.

통신 회사와 연계하거나 클라우드 기반 구축 등 환경 정비도 중요하지만, 여기서 더 중요한 것은 사용자의 니즈를 얼마나 흡수해 서비스 기획에 반영하는가 이다. 이러한 의미에서는 사용자와 직접적으로 접점을 가지는 자동차 OEM의 직판 회사나 딜러 등을 독립시키지 않고 자동차를 중심으로 한 서비스 구성에 끌어들이는 것이 중요하다.

앞으로 MaaS가 활발해지면서 동일한 움직임이 비자동차 업계의 플레이어에도 발생할 것이다. 이러한 변화를 가속함으로써 앞으로 구현되는 OTA에 관련해 어떤 기능이 OTA를 통해 업그레이드되어야 사용자가 원하는(앞으로 요구할 가능성이 큰) 니즈에 대응할 수 있는지와 같은 질문에 훨씬 정확한 추측이 가능해질 것이다.

나아가 커넥티드 서비스와 OTA로 차량 기능 업그레이드를 실행하기에 앞서, 당장 차량 단가 상승과 미래적인 기대 수익 사이에서 어느 것을 우선할지 딜레마에 빠진다. CASE 시대에는 사용자 니즈의 고도화, 다양화가 예측되며 수년~5년 후 그들의 니즈를 정확하게 파악한 다음에 자동차의 기능을 구현하는 것은 어렵다. 잠재적인 니즈에 대응해야 하는 소프트웨어 개발진은 애자일 개발을 추진해 각 기능의 신속한 업그레이드를 진행해야 하므로 OTA 업데이트는 일종의 전제 조건이 된다.

한편 기존의 자동차 개발 시각으로는 잠재적인 니즈를 충족하기 위해 OTA에 대응하면 ECU를 비롯한 하드웨어 스펙의 기능 향상이 필요한데 결과적으로 코스트 상승을 피할 수 없다. 따라서 기존의 자동차 개발 관점에서는 「지금, 당장 니즈가 없는데 불투명한 미래를 위해서 무리하게 하드웨어 스펙을 올려 차량 단가가 상승하면 지금까지 키워온 사용자 기반을 잃을 수 있다」

한편 소프트웨어 개발진으로부터 「애자일 개발은 몇 년 뒤 사용자의 잠재적인 니즈에 대응하기 위해서다. 몇 년 뒤 OTA 업데이트에 대응하지 못하면 그거야말로 미래의 사용자 기반을 잃을 수 있다」라는 불안의 목소리도 나오고 있다. 「현재의 니즈에 대응 vs 미래의 잠재적인 니즈에 대응」과 같은 구도의 딜레마가 발생할 가능성이 크다.

특히 염가 양산형으로 각 차종의 라인업을 정비하고 있는 일본계 자동차 OEM에게 이러한 딜레마는 굉장한 부담이 된다. 다임러나 폭스바겐의 입장에서는 메르세데스 벤츠나 아우디, 포르쉐 같은 차량 단가가 약 700만~1,000만 엔 이상의 프리미엄 세그먼트에서 각각 브랜드를 갖추고 있어서, 기존 사용자에게 가격 상승은 일본계 자동차 OEM의 사용자만큼 심각한 문제가 아니다. 오히려 최신 기능이 사용자 차량에 반영되는 사용자 가치 쪽이 앞서 나온 리스크를 뛰어넘을 가능성이 크다. 토요타 자동차 LEXUS와 일부 차종, 지역 브랜드(닛산 자동차의 북미 브랜드 INFINITI)에서만 프리미엄 세그먼트를 보유한 일본계 자동차 OEM은 통상 라인업으로 어떻게 이러한 딜레마를 해결할지 결단이 촉구된다. 그리고 그것은 OPEX 비즈니스 모델의 구축 계획을 세움으로써 그 해결책을 찾을 수 있을 것이다.

IV. 새로운 자동차 제조로 이행하기 위해서 : 가야 할 방향성

지금까지 소프트웨어 중심 제조업이 가져올 세계관과 소프트웨어 중심 자동차 제조의 세계를 실현하기 위해 일본계 자동차 OEM이 넘어서야 할 과제에 관해서 설명했다. 마지막으로 지금까지 언급한 과제에 대해 각사가 취해야 할 방향성을 고찰하겠다.

1. 코어 · 논코어를 구분해라

첫 번째 장벽으로 애자일과 워터폴 개발 부서를 통합시켜 단일 차량을 개발할 때 매니지먼트하는 어려움에 대해 이야기했다. 이 문제는 기본

적으로는 내재화를 추진해야 하지만, 소프트웨어 관련 도메인이나 파트가 비대해지는 길목에 갑자기 모든 것을 내재화하고 전부 애자일 방식으로 관리하는 것은 보통 일이 아니다.

그래서 필자는 애자일 개발과 워터폴 개발의 통합 매니지먼트를 고려할 때 각사의 코어 · 논코어를 분리하는 것이 중요하다고 생각한다. 코어 · 논코어 영역의 공존은 자동차 OEM 각사의 특징에 따라 공존하는 포인트는 다르겠지만, 기본적으로 코어 영역에서는 내재화를 추진하고 논코어 영역에서는 외부 파트너를 적절히 활용한다는 점은 공통적이다. 다만 현실적으로 코어 영역이라고 해서 서플라이어를 포함한 외부 파트너를 아예 활용하지 않는다는지, 반대로 논코어라고 해서 서플라이어에 모든 일을 통째로 맡기라는 뜻은 아니다. 다양한 기능이 탑재되어 소프트웨어 개발이 복잡해지는 상황에서 전부 내재화하는 것은 불가능하다.

코어 · 논코어로 나누어 활용함으로써 각사의 강점에 따라 자원을 최적으로 분배하고 코어가 되는 영역에서는 자사의 그림력을 잘 살려 애자일 개발을 계속하고, 반대로 논코어 영역에서는 워터폴로 서플라이어를 포함한 외부 파트너를 적절히 활용하면서 개발을 추진하는 것이 중요하다. 코어 · 논코어에 따라 유연하게 애자일과 워터폴을 구분하여 사용하는 것이 소프트웨어 개발 기능에서 앞으로 점점 중요해질 것이다.

2. 프래그먼트화된 스택홀더를 통합하라

CASE 시대에는 자동차의 비즈니스 모델이 OPEX형으로 이행해 간다고 앞서 설명했다. 소프트웨어 기능이 개발 기능과 함께 고도화되면서 비즈니스에서 한층 중요해진 것은 사용자의 의견을 얼마나 잘 받아들여 개발에 반영해 더 나은 서비스를 생산하는 루프를 만들어 내느냐에 있다.

그때 단순히 데이터를 많이 추출해서 개발만 하면 되는 것은 아니다. 데이터 연계를 위해 환경을 정비하는 통신, 클라우드 영역에 머무르지 않고 자동차와 직접적으로 연관이 없다고 여겼던 플레이어도 끌어들여 그들에게서 받는 피드백을 더 중요시해야만 한다.

또 구체적인 이용 상황이나 고객의 특성, 사용자의 의견을 추출하면서 어떤 영역에서 어떤 부가 가치를 고객에게 제공하고 그 대가로 얼마에 서비스를 제공할지와 같은 밸류 프로포지션이 중요해진다. 자동차 OEM은 밸류체인인 고객과 접점이 있는 다양한 플레이어와 대등한 파트너링을 강화함으로써 기존의 자동차 제조·판매에서는 손에 넣을 수 없었던 노하우를 획득할 가능성이 커진다.

물론 그것은 자동차 OEM이 일방적으로 스택홀더(이해관계자)에게 요구하는 것이 아니라, 이해관계자의 니즈도 고려해 자동차 OEM만의 역량을 제공할 필요가 있다. 그 결과, 더욱 폭넓은 시장으로서 민감도가 좋아져 CX나 UX 영역의 이해나 향상으로 이어진다. 그렇게 되면 앞으로 고객 중심의 부가 가치가 높은 서비스를 만들 기회나 그것을 개발 프로세스에 반영시킬 기회가 늘어날 것이다.

또한 그러한 부가 가치를 높인 서비스를 널리 보급하기 위해서는 마인드셋의 개혁도 자동차 OEM에 요구된다. 지금까지는 보유한 볼륨존에 대해 기존의 니즈에 얼마나 맞출 수 있는지 하는 현시점의 비즈니스에 특화되어 있었다. 그러나 이제부터 OPEX형으로 이행하기 위해서는 현시점에는 상상하기 어려운 잠재적인 니즈에 유연하게 대처해야만 한다. 기존의 시장에 안주하려는 생각에서 벗어나 새로운 시장을 개척하려는 프런티어 정신으로 무장해야 한다.

그때 오히려 스몰 볼륨인 니치 세그먼트에서 승부를 보는 것도 하나의 방법이 될 수 있지 않을까. 예를 들어 코스트 문제로 최신 규격을 모두 갖춘 상태로 시장에 투입하는 것이 어렵다면 차량 단가가 높은 프리미엄 세그먼트에 프로토타입으로 투입하는 방법 등을 고려할 수 있다. 테슬라도 현재는 모델 3과 같은 양산 제품 라인이 확립되어 있지만, 그 기반이 되는 기술 노하우나 UI, UX를 높이는 실마리를 제공한 제품은 고급 차종인 Roadster, Model S이다.

신제품 라인업이나 세그먼트 중에서 코스트 논의에 얽매이지 않고 또 그 제약을 최대한 배제하면서 우선 각사가 바라는 세계관을 반영한 차종을 시장에 투입한다. 그 다음에 시장 규모를 키워야 하고 거기서 얻은 깨달음이나 배움을 바

탕으로 새로운 자원을 추가해 한층 더 발전한다. 이전 일본 제조업의 강점이었던 기존의 기술이나 상식에서 벗어나 새로운 것에 도전하는 「즐기는 자세」가 자동차 제조 개발 프로세스나 비즈니스 모델 창출에서 중요해질 것이다.

그때 토요타자동차의 LEXUS, 닛산자동차의 INFINITI 등 프리미엄 브랜드를 적절히 활용해 트라이&에러를 반복한다면 상기의 프로세스를 효과적·효율적으로 추진할 수 있지 않을까.

3. 마지막으로: 새로운 자동차 제조에서 중요한 것은 소프트웨어와 하드웨어의 밸런스

CASE 시대에 소프트웨어 개발의 중요성이 날로 커지는 한편, 기존의 사고와 전혀 다른 GAFA나 테슬라의 습격으로 자동차 업계에 지각 변동이 일어나기 시작했다. 소프트웨어 개발에서 압도적 우위를 자랑하는 그들이 향후 자동차 산업 진출에 더욱 박차를 가하는 상황 속에서 자동차 OEM은 대담하게 전략을 재검토해야 한다.

CASE 시대에 자동차 제조는 하드웨어 개발 시점만으로 완결될 수 없다는 점은 앞서 다루었다. 그러나 전부 애자일 방식으로 바꾸어 앱이나 소프트웨어 개발 회사처럼 소프트웨어 관점만으로 개발해도 자동차 제조는 성립하지 않는다. 중요한 것은 소프트웨어와 하드웨어 사이에서 균형을 잡으면서 그 힘의 배분을 어떻게 할 지이다. 소프트웨어를 중심으로 한 개발은 기존의 자동차 제조와는 전혀 다른 발상이 필요하고 때로는 지금까지 괜찮다고 여겨졌던 사고방식을 버려야만 하는 국면도 마주할 수 있다. 이러한 딜레마를 해소하고 앞으로 나아가려면, 지금이야말로 대담한 발상을 바탕으로 관계된 산업 내외의 스택홀더가 상호 존중하는 마음으로 원팀이 되어 개발 체제의 재검토·개선을 새로운 비즈니스 모델의 계획으로 책정하는 것이 중요하다.

일본계 자동차 OEM은 하드웨어를 기축으로 한 자동차 제조에서 창의적인 요소를 고민하면서 제조 기능을 업데이트하고 오퍼레이션을 효율화해 왔다. 그 과정에서 키운 경쟁 우위성을 유지하면서 소프트웨어에 앞으로는 더욱 힘을 쏟아야만 한다.

소프트웨어화의 거친 파도 속에서 과거에 얽매이지 않고 미래 지향의 마인드를 추구한다면 미래 자동차 제조에서도 일본계 자동차 OEM 각사는 세계 정상에 우뚝 설 수 있을 것이다.

주

1. 인포테인먼트 (In Vehicle Infotainment). 주로, 자동차(차재시스템에 사용되는 용어로 오락의 제공, 정보의 제공을 실현하는 시스템의 총칭, 또는 양쪽 기능의 제공을 실현하는 시스템을 통칭.
2. Electronic Control Unit의 약어. 주로 자동차에 제어되는 전자제어유닛을 가리킴
3. Hardware Abstraction Layer의 약어. 컴퓨터의 구성을 계층화한 모델 중, 소프트웨어측의 최하층에서 하드웨어별 사양의 차이를 흡수하고 공통 방식으로 동작할 수 있는 역할을 하는 계층
4. Over the air의 약자. 일반적으로 데이터의 송수신을 무선통신으로 실행하기 위한 기술을 나타냄. OTA를 활용함으로써 IoT기기의 소프트웨어 및 펌웨어의 업데이트의 프로그램을 무선 경유로 송수신 가능
5. System on a chip의 약어. 시스템을 구성하는 모든 기능을 하나의 반도체칩으로 실현하는 의미를 나타냄.
6. Full Self Drive의 약어. 테슬라의 완전 자동운전용 소프트웨어의 명칭
7. Request for Quotation 의 약어. 물품이나 서비스의 구입처, 그 후보가 되는 기업 등에 대해 가격 및 그 내역 등을 기입한 견적서를 작성하도록 의뢰하는 의뢰 문서

필자

야마모토 타카시(山本隆史)

NRI 글로벌 제조업 컨설팅부 컨설턴트

전문은 제조업계의 사업전략입안, 플래폼 전략, 구조개혁지원, 하드웨어 X 소프트웨어의 비즈니스모델 플랜 책정

타카하시 츠카사(高橋 主)

NRI 글로벌 제조업 컨설팅부 상급 컨설턴트

전문은 자동차 및 주변 업계의 중장기경영계획 책정, 사업전략 입안·추진, 서비스/디지털 기업으로의 기업 변혁 추진, 경영 기반으로서의 코퍼레이트 거버넌스, 리스크 매니지먼트 구축·추진 등

본 기사는 知的資産創造 2022년 2월호에서 발췌하여 한국어로 번역하였습니다.

문의사항은 노무라종합연구소 서울로 연락 바랍니다.

문의처 : inquiry@nri-seoul.com

홈페이지 www.nri-seoul.com 의 insight 메뉴에서 더 많은 기사를 볼 수 있습니다.

또한 知的資産創造 2022년 2월에 대한 전문 및 기사는 www.nri.com에서 열람 가능합니다.

본 기사의 무단 전재, 복제를 엄격히 금합니다. 모든 내용은 일본의 저작권법 및 국제조약에 따라 보호받고 있습니다.

Copyright © by Nomura Research Institute, Ltd. All rights reserved.