

난제를 푸는 사람들: 한국 엔터프라이즈 AX 현장에서 실험한 FDE [Forward Deployed Engineer]

· 황현태 | FDE 스타트업 스페이스와이 대표 | hello@dio.so



1. 들어가며

최근 국내 IT 및 엔터프라이즈 업계의 화두는 단연 ‘팔란티어(Palantir)’와 그들의 핵심 인력인 ‘FDE(Forward Deployed Engineer)’다. 바야흐로 FDE 전성시대라 해도 과언이 아니다. 최근 정부 차원에서 관련 정책 간담회가 열릴 정도로 뜨거운 감자가 되었고, 국내 주요 기업들 역시 앞다투어 “FDE를 몇백 명 단위로 대거 채용하겠다”며 그야말로 ‘핫’하다. 글로벌 AI 시장을 휩쓴 팔란티어의 성공 방정식이 이제 한국의 AI 전환(AX) 시장에서도 거스를 수 없는 대세가 된 것이다.

하지만 언론 보도나 리서치 결과만으로는 FDE의 진짜 의미를 알 수 없다. 기업들은 생성형 AI에 막대한 투자를 했지만 성과는 기대에 못 미친다. MIT NANDA 연구진은 2025년 보고서에서 조직의 95%가 생성형 AI 투자에서 측정 가능한 손익 효과를 얻지 못했다고 진단했다. 조직에 AI 모델이나 도구를 그저 던져준다고 해서 기업이 바뀌지는 않는다는 사실을 이제 모두가 체감하고 있다.

필자는 최근 한국 대기업 현장 최전선에 투입되어 FDE 방식의 일하기를 직접 실험했다. 어떤 조직은 에이전트를 4,000개씩 만들면서 AX에 성공했다고 하고, 어떤 조직은 만들어둔 4,000개 에이전트를 다시 4개로 좁히면서 정리(Sweeping)하고 집중해서 확산시키자고 한다. 필자는 이 간극의 정체가 ‘문제의 난이도’라고 본다. 이미 문서로 정리되어 있고, 절차가 분명하고, 정답의 모양이 보이는 문제는 앞으로 AI가 차례로 가져간다. 몇 달 전만 해도 사람이 붙어야 했던 일을 지금은 범용 에이전트가 해내고, 그 경계선은 몇 주 단위로 밀려난다. 쉬운 문제를 빠르게 처리하는 능력은 더 이상 희소하지 않다. 남는 것은 밖에서는 보이지도 않고, 정의부터 새로 해야 하고, 조직을 함께 움직여야 겨우 풀리는 문제들, 곧 난제(難題)다.

그 난제를 붙잡는 사람을 조직화하는 방식으로 주목받는 것이 FDE다. 필자가 이끄는 스페이스와이는 2025년 하반기부터 국내에서 비교적 이른 시기에 FDE 모델을 실험했다. 현재는 LG전자, 키움증권 등 10여 곳의 엔터프라이즈와 AX 파트너십을 맺고 FDE 그룹을 운영하고 있다.

이 글은 한국 엔터프라이즈 AX의 가장 치열한 현장에서, FDE라는 이름으로 직접 난제와 부딪히고 깨지며 얻어낸 가장 날것의 기록이자 진짜 현장의 이야기다. 이 글은 FDE가 어디서 온 개념이고, 한국에서 왜 자꾸 오해받으며, 지금 현장에서 무엇이 벌어지고 있는지를 차례로 짚은 뒤, 결국 이 직무의 본질이 무엇인지에 대한 필자의 답을 담았다.

2. FDE는 어디서 왔나: 팔란티어와 ‘전방 배치’의 20년

‘전방 배치(forward deployed)’는 원래 군사 용어다. 병력과 장비를 후방 기지에 두지 않고 작전이 벌어지는 최전선에 미리 내보내 두는 것을 뜻한다. 이 말을 소프트웨어 회사의 직무 이름으로 끌어다 쓴 곳이 2003년 설립된 미국의 데이터 분석 기업 팔란티어(Palantir)다.

팔란티어의 초기 고객은 정보기관과 수사기관이었다. CIA의 벤처 투자 조직인 인큐텔(In-Q-Tel)이 초기 투자자로 들어왔다. 문제는 이런 고객의 업무를 밖에서는 알아낼 방법이 없다는 점이었다. 무엇이 문제인지 자체가 기밀이고, 분석관이 실제로 어떤 순서로 무엇을 들여다보는지는 그 옆자리에 앉아 보지 않으면 알 수 없다. 요구사항을 받아 적을 수도 없고, 데모를 만들어 팔 수도 없었다. 그래서 팔란티어는 엔지니어를 고객의 책상 옆으로 보냈다. 팔 물건이 없어서 사람을 보낸 것이 아니라, 문제를 알아낼 방법이 그것밖에 없어서 보낸 것이다.

여기서 팔란티어 특유의 이원 구조가 나온다. 현장에 나가는 FDE와 본사에서 제품을 만드는 PD(Product Development) 엔지니어를 나누고, FDE가 현장에서 급하게 만들어 낸 것 가운데 여러 고객에게서 반복되는 것만 골라 제품으로 끌어올렸다. 현장은 문제의 원천이고, 본사는 그 문제를 일반화하는 곳이다. 훗날 최고기술책임자가 되는 삼 상카르(Shyam Sankar)가 이 모델을 정착시킨 인물로 꼽힌다.

이 방식은 오랫동안 비판받았다. 상장 전후로 팔란티어에는 “소프트웨어 회사가 아니라 컨설팅 회사”라는 꼬리표가 붙었다. 사람이 들어가는 만큼 매출총이익률이 낮았기 때문이다. 그러나 현장에서 길어 올린 패턴이 고담(Gotham)과 파운드리(Foundry)라는 제품으로 굳어지면서 이익률은 80% 안팎까지 올라갔다. 초기의 낮은 마진은 낭비가 아니라 해자(moat)를 사는 값이었다는 것이 지금의 평가다. 2023년 시작한 AIP 부트캠프는 이 모델 자체를 상품으로 만든 사례다. 제안서와 견적서 대신, 닷새 동안 고객의 실제 데이터로 돌아가는 무언가를 만들어 눈앞에 보여 준다.

FDE가 20년 만에 다시 주목받는 이유는 AI 모델 기업들이 똑같은 벽에 부딪혔기 때문이다. 모델은 벤치마크에서 인간을 앞서는데, 고객사 안으로 들어가면 좀처럼 돌아가지 않는다. 데이터가 어디에 어떤 상태로 있는지, 그 업무의 진짜 판단 기준이 무엇인지, 누가 무엇을 결재하는지를 모델이 알아낼 방법이 없다. 밖에서는 추론할 수 없는 정보를 얻으려면 안으로 들어가는 수밖에 없다. 실제로 2026년 상반기 OpenAI, Anthropic, Amazon, Microsoft가 잇달아 FDE 전담 법인이나 대규모 조직을 만들거나 키웠다. 해외 보도에 따르면 주요 AI 기업이 최근 1년간 FDE에 투입하기로 한 자금은 100억 달러에 가깝다. 운영 방식도 세 갈래다. OpenAI와 Anthropic은 외부 자본을 유치해 별도 법인을 세웠고, Microsoft와 Amazon은 자체 인력과 재원으로 조직을 운영하며, Google Cloud는 파트너 펀드를 조성했다. AI 모델 기업들이 B2B 시장에 AI를 제대로 안착시키는 역할에 본격적으로 돈을 쓰기 시작한 것이다.

그렇다면 FDE는 어떤 사람들인가. 팔란티어가 FDE를 뽑을 때 도메인 전문가를 찾지 않았다는 점은 시사적이다. 금융 고객을 맡을 사람에게 금융 경력을 요구하는 대신, 낯선 분야를 몇 주 만에 이해하고 모호한 상태를 견디며 스스로 다음에 할 일을 정하는 사람을 뽑았다. 도메인 지식은 고객에게서 빌릴 수 있지만 문제를 정의하는 능력은 빌릴 수 없기 때문이다. 정의된 문제에 완성도 높은 솔루션을 만드는 일반 엔지니어와 달리, FDE는 고객의 비즈니스 상황을 이해하고 모호한 문제를 정교하게 깎아 낸 뒤 해법을 고안하고 직접 코드로 구현한다. 필요하면 교육도 하고, 컨설팅도 하고, 조직문화를 바꾸는 일까지 한다. 한마디로 ‘컨설턴트와 엔지니어의 결합’이다.



3. 한국에도 ‘현장에 나가는 개발자’는 있었다

한국에서 FDE를 설명하면 옆에 아홉은 같은 질문이 돌아온다. “그거 SI랑 뭐가 다른데요?” 이 반사적인 반응은 무지에서 나온 오해가 아니라 역사에서 나온 기억이다.

1980년대 중반부터 대기업 그룹의 전산실이 잇따라 독립 법인으로 떨어져 나오면서 한국의 시스템 통합 산업이 만들어졌다. 1990년대 후반 Y2K 대응과 전사적 자원관리(ERP) 도입, 2000년대 전자정부 사업을 거치며 이 산업은 빠르게 커졌다. 수천 명이 등기부등본을 일일이 입력해 국가의 기록을 전산화해 낸 것도 이 시기의 일이다. 한국 소프트웨어 산업은 제품을 만들어 파는 방식보다 고객의 현장에 들어가 시스템을 지어 주는 방식으로 먼저 자랐다. 즉 ‘고객 현장에 나가는 개발자’는 한국에서 새로운 개념이 아니라 오히려 기본값이었다.

문제는 그 자리에 부여된 의미다. 일은 과업지시서로 시작해 검수로 끝났다. 값은 사람을 초급·중급·고급으로 나눠 한 달치 인건비(M/M)로 매겼다. 원청과 하청이 여러 단계로 겹쳤고, 현장에 실제로 앉아 있는 사람은 그 사슬의 아래쪽인 경우가 많았다. 이 구조에서 고객사 현장은 문제를 발견하러 가는 곳이 아니라, 이미 확정된 문제를 정해진 기간 안에 소화하러 가는 곳이었다. 2010년대에 제품을 직접 만드는 인터넷 기업들이 성장하면서 위계는 더 뚜렷해졌다. 뛰어난 엔지니어는 제품 회사에 남고 현장에는 시간 단위로 팔리는 인력이 간다는 인식이 자리 잡았다.

그래서 한국에서 “우리 회사에서 가장 뛰어난 엔지니어가 고객사에 나가 일한다”는 문장은 그 자체로 낯설게 들린다. 필자는 FDE와 기존 방식의 가장 큰 차이가 바로 여기에 있다고 본다. 기술이 아니라 누가 현장에 가느냐가 뒤집힌 것이다. 과거에는 가장 값이 싼 인력이 현장에 나갔지만, FDE 모델에서는 가장 비싼 사람이 나간다. 어떤 조직이 FDE를 제대로 하고 있는지 알아보는 가장 빠른 방법은 조직도가 아니라 이것이다. 그 회사에서 가장 뛰어난 사람이 지금 고객 앞에 앉아 있는가.

구분	정의된 문제를 푸는 방식	난제를 푸는 방식(FDE)
출발점	확정된 요구사항·과업지시서	모호한 문제와 목표 지표
현장에 가는 사람	단가가 맞는 인력	조직에서 가장 뛰어난 사람
견적계약	투입 인력(M/M) 기준	짧은 기간 내 성과 기준
일하는 방식	WBS에 따른 단계별 수행	관찰·구현·측정의 빠른 반복
종료 이후	유지보수 계약	고객사 내재화

차이는 계약서에서도 드러난다. 한 고객사 AX 담당자의 지적처럼, 투입 인력으로 값을 매기는 방식은 사람을 창의성이 아니라 투입과 산출의 관점에서 본다. 난제를 푸는 일에는 맞지 않는 자다. 어려운 문제를 세 명이 6주 만에 푼 것이 서른 명이 1년을 쓴 것보다 비싸야 정상이기 때문이다. 종료 시점의 목표도 반대다. 기존 방식은 유지보수 계약으로 이어지는 것이 성공이지만, FDE는 고객이 다시 우리를 부르지 않아도 되는 상태를 만드는 것이 성공이다.

4. 지금 한국에서는 무슨 일이 벌어지고 있나

한국에서 FDE가 빠르게 확산된 데는 세 가지 배경이 있다. 첫째, 팔란티어의 성장이 ‘K-팔란티어’라는 모델에 대한 관심을 불러일으켰다. 둘째, 2025년 하반기 추론 모델이 크게 발전하면서 “과거 10~20명이 하던 일을 3~5명이 할 수 있다”는 막연한 기대가 현실적인 계산이 되었다. 셋째, 엔터프라이즈들이 내부 시도만으로는 한계를 느끼고 외부의 동력을 걸어 보려 했다. 이 과정에서 AX 스타트업들이 FDE라는 개념을 들고 대기업 현장에 들어갔고, 대형 IT 서비스 기업과 클라우드 기업도 FDE 조직을 꾸리기 시작했다. 국내 FDE 채용 공고가 크게 늘었다는 이야기가 나오는 것도 이 때문이다.

국내 엔터프라이즈의 AX 흐름을 보면 FDE가 등장한 맥락이 더 분명해진다. 필자는 이를 네 세대로 나누어 본다. 1세대는 계열사마다 AX 전담팀을 두고 현업의 요청을 대신 풀어 주는 방식이었다. 과거 IT 팀은 1년에 몇 개의 과제를 해결하면 됐지만, AI는 작성하고 찾으면 과제가 수백 개씩 나온다. 전담팀에는 곧 과부하가 걸렸다. 2세대는 현업이 직접 할 수 있도록 교육에 집중했지만, 학습 성취도의 편차 때문에 한계가 있었다. 3세대는 인원은 적지만 AI 활용이 최적화된 스타트업을 불러들여 속도와 방법을 배우는 시도였다. FDE는 바로 이 3세대의 산물이다. 그리고 지금은 리더 자신이 AI 네이티브가 되어야 한다는 4세대로 넘어가고 있다. 4,000개의 에이전트도 결국 무엇을 만들지 말아야 하는지를 정하는 리더의 철학이 없었기 때문에 생긴 결과다.

확산 속도에 비해 개념은 아직 정리되지 않았다. 얼마 전 참석한 FDE 정책 간담회에는 8개 기업이 참여했는데, 회사마다 말하는 FDE의 정의가 달랐다. 누군가는 세일즈 엔지니어로, 누군가는 코딩할 줄 아는 컨설턴트로, 누군가는 고객사에 상주하며 프로젝트를 총괄하는 PM으로 본다. 이름만 바꿔 단 조직도 이미 적지 않다. 개발자의 직함을 FDE로 고쳐 쓴다고 FDE가 되지는 않는다.

5. 난제를 푸는 사람들: 엔지니어가 아니라 생각하는 사람들

그래서 FDE란 결국 무엇인가. 필자는 두 단어로 정리한다. ‘난제’와 ‘패턴화’다.

먼저 난제다. 앱이나 SaaS처럼 100만 명의 고객에게 하나의 공통 문제를 풀어 주던 범용 엔지니어링의 ‘쉬운 열매’는 이미 상당 부분 수확되었다. 남은 것은 엔터프라이즈 내부에 있어서 밖에서는 추론조차 할 수 없는 문제들이다. 무엇을 개선해야 하는지가 합의되어 있지 않고, 판단 기준은 실무자의 머릿속에만 있으며, 데이터는 시스템 다섯 곳에 흩어져 있고, 성과를 재는 지표부터 새로 만들어야 하는 문제. 이런 문제는 좋은 모델이나 잘 쓴 프롬프트로 풀리지 않는다. 누군가 현장에 들어가 문제를 다시 정의하고, 사람들을 설득하고, 코드로 증명해야 겨우 움직인다.

난제를 고집하는 것은 자존심의 문제가 아니라 생존의 문제다. 쉬운 문제는 정의상 AI가 가장 먼저 정복하는 영역이다. 오늘은 사람이 붙어야 겨우 되던 일이 다음 분기에는 도구 하나로 끝나는 일이 된다. 쉬운 문제를 많이 처리하는 것으로 존재를 증명하는 조직은 모델이 한 번 좋아질 때마다 존재 이유가 깎여 나간다. 반대로 난제를 푸는 사람의 가치는 AI가 좋아질수록 커진다. 도구가 강력해질수록, 그 도구를 어디에 겨눌 것인지 정하고 조직이 그 결정을 감당하게 만드는 일이 희소해지기 때문이다.

여기서 이 글에서 가장 강조하고 싶은 이야기가 나온다. FDE는 엔지니어라는 이름을 달고 있지만, 본질은 코드를 잘 쓰는 사람이 아니라 생각하는 사람이다. 코딩 실력은 필요조건일 뿐이다. 현장에 들어간 FDE의 시간은 대부분 코드 편집기 밖에서 쓰인다. 실무자 옆에 앉아 무엇을 보고 무엇을 넘기는지 관찰하고, “왜 그렇게 하시죠?”를 지치지 않고 되묻고, 서로 다른 부서가 같은 단어를 다른 뜻으로 쓰고 있다는 것을 발견하고, 이 업무의 성과를 무엇으로 재야 하는지를 고객과 다시 합의한다. 요청받은 것이 진짜 문제가 아니라고 판단되면 계약서에 적힌 일을 멈추고 다른 문제를 제안한다. 이것은 구현 노동이 아니라 사고 노동이다.

필자가 FDE를 채용할 때 창업이나 초기 스타트업 경험을 핵심 조건으로 보는 이유도 같다. 아무도 정의해 주지 않은 문제를 스스로 정의하고, 자원이 부족한 상태에서 순서를 정하고, 틀렸으면 빠르게 방향을 바꿔 본 경험. 그리고 모호함 앞에서도 결국 풀 수 있다고 믿는 낙관. 이런 것들은 알고리즘 문제 풀이 실력으로 대체되지 않는다.

생각하는 사람과 시키는 일을 하는 사람의 차이는 질문의 모양에서 드러난다. 예를 들어 지금 기업과 기관에는 ‘OO 업무 자동화 에이전트 구축’이라는 이름의 사업이 넘쳐 난다. 업무를 잘게 쪼갠 뒤 우선순위가 높은 것부터 하나씩 소프트웨어로 만들던 습관을 에이전트에 그대로 옮긴 것이다. 이는 정의하기 쉬운 문제만 골라 계속 푸는 일이고, 그래서 끝이 없다. 사무 업무는 무한하고 업무별 에이전트도 무한히 늘어난다. 에이전트 4,000개는 곧 유지보수 4,000개다. 게다가 범용 코딩 에이전트들은 몇 주 단위로 업데이트 되는데 사내 전용 에이전트가 이 속도를 따라가기는 어렵다. 난제를 푸는 사람은 질문을 바꾼다. “이 업무의 에이전트를 어떻게 만들까?”가 아니라 “범용 에이전트가 우리 회사의 업무를 안전하게 수행하려면 무엇을 갖춰야 할까?”를 묻는다. 기업이 꼭 잡아야 할 것은 개별 에이전트가 아니라 데이터 접근 채널, 보안, 평가 체계, 거버넌스다. 우리가 회계용 컴퓨터와 사무용 컴퓨터를 따로 만들지 않았던 것처럼, AI 에이전트는 애플리케이션이 아니라 컴퓨터에 가까운 개념으로 받아들여야 한다.



과제를 설계하는 방식에서도 마찬가지다. 실무자의 업무만 자동화하면 경영진은 “회사가 달라진 게 없다”고 느낀다. 경영진용 대시보드만 만들면 실무자는 “우리 일은 그대로”라고 느낀다. 그래서 FDE는 과제를 옆으로 넓히기보다 위로 세운다. 특정 업무의 자동화에서 출발해 그 결과가 보고되고 의사결정으로 이어지기까지, 전 직급을 관통하는 좁고 긴 워크플로우를 설계해야 조직 전체가 AX의 효능감을 느낄 수 있다. 이런 판단은 요구사항 목록 어디에도 적혀 있지 않다.

두 번째 단어인 패턴화는 이 사고 노동이 한 번 쓰고 버려지지 않게 하는 장치다. 난제만 받는다고 해서 모든 고객의 문제를 매번 0에서 시작해서는 안 된다. 여러 고객을 거치며 쌓인 지식과 모범 사례로, 개별 고객이 스스로는 발견하지 못할 패턴을 찾아내야 한다. 팔란티어가 20년에 걸쳐 증명한 것처럼, 이렇게 축적된 패턴이 FDE 조직의 해자다.

6. 현장에서 배운 FDE 운영 원칙

FDE는 유행한다고 누구나 할 수 있는 일도, 어느 기업이나 바로 도입할 수 있는 일도 아니다. 지난 1년간 시행착오를 거치며 정리한 원칙은 다음과 같다.

- ① **난제만 받는다.** 한 고객사 팀장이 이렇게 말한 적이 있다. “쉽고 확실한 업무 자동화였으면 외부에 맡기지 않았다.” 문서로 정리된 업무를 자동화하는 일은 이미 시중의 도구와 고객사 내부 인력으로 충분하고, 그 범위는 해마다 넓어진다. FDE가 들어가야 할 곳은 실무자와 팀장의 머릿속에만 있는 암묵지를 건드려야 하는 문제, 모델 성능이 아니라 조직의 합의가 병목인 문제다. 진입 기준이 없으면 FDE 조직은 순식간에 ‘무엇이든 해 주는 외주팀’이 되고, 그때부터는 AI에게 잠식당하는 일만 남는다.
- ② **영웅이 아니라 2~3인 팟(pod)으로 운영한다.** FDE에 대한 가장 큰 오해는 ‘고독한 슈퍼맨’이라는 이미지다. 문제를 소유하는 메인 FDE를 중심에 두고, 현장을 조율하는 작업반장과 서포터가 함께 움직이는 소수 팀이 효과적이다. 상황에 따라 엔지니어링·리서치·커뮤니케이션 자원을 나누고, 역할을 일부러 겹치게 설계해야 한 사람이 힘든 주에도 동료가 바로 받쳐 줄 수 있다. AI로 한 사람의 생산성이 커질수록 프로젝트의 무게가 한 사람에게 쏠리기 쉽다. 팀 설계는 번아웃을 막는 장치이기도 하다.
- ③ **타임박싱(Time-boxing)을 한다.** FDE를 1~2년짜리 구축 사업처럼 쓰면 속도가 사라진다. 해외 기업들도 3~4주, 6~8주, 12~14주처럼 짧은 기간 안에 실제 운영 가치를 만드는 것을 목표로 한다. 필자는 3~6개월 안에서 문제를 설계하고 풀도록 기간을 정할 것을 권한다. 기업이 FDE에게 원하는 것은 결국 ‘속도감의 이식’이기 때문이다. 다만 프로덕션 수준의 품질 기준을 생략하지 않으려면 착수 전에 리스크를 모두 목록으로 만들어 두어야 한다.

- ④ **목표를 분명히 정하고 결과에 집착한다.** FDE에게 문제의 모호함은 피할 수 없지만, 기간과 목표까지 모호해서는 안 된다. 이번 과제의 목표가 전사 확산인지, 빠른 실험의 개수인지, 완성도인지를 먼저 합의해야 한다.
- ⑤ **끝은 내재화다.** FDE가 많은 것을 만들어 놓고 나가면 결국 녹슬지 않느냐는 우려가 늘 따른다. 필자는 FDE 컴퍼니의 최종 목표가 “고객이 유지보수를 위해 다시는 우리를 찾지 않아도 되는 상태”여야 한다고 생각한다. 외부 인력이 유지보수를 맡으면 업무의 변화가 제때 반영되지 않는다. 최신 업데이트를 하지 못한 ERP를 계속 써야 했던 과거가 되풀이되는 것이다. 그래서 고객사 안에 메인테이너를 키워야 한다. 주니어는 메인테이너 역할을 맡으며 도메인 지식을 쌓을 수 있고, 도메인 지식이 풍부한 시니어는 AI 도구로 자신의 업무 시스템을 직접 고칠 수 있다.

7. 제언

수요기업에게. 첫째, FDE 도입 전에 실제로 FDE를 써 본 기업의 담당자와 먼저 이야기해 보기를 권한다. 둘째, 앞의 사례에서 보듯 데이터 접근 권한과 보안 절차를 미리 마련하고, 현업과 FDE 사이를 잇는 조율자를 초기에 배치해야 한다. 셋째, 계약에는 내재화 과제를 반드시 하나 이상 넣어야 한다. 모델 성능이 좋아지고 직원들의 AI 활용 수준이 높아져도 외부 FDE가 여전히 필요한 이유는 따로 있다. 이미 짜여 있는 조직과 기존 인력만으로는 도전적인 일정과 난이도에 스스로 뛰어들기 어렵기 때문이다. FDE는 그 ‘외부 동력’이 되어야 하고, 동력이 빠진 뒤에도 조직이 굴러가게 만들어야 한다.

정부와 공공 부문에게. 공공 서비스는 전 국민이 사용자인, 가장 큰 서비스다. 정부의 인공지능 행동계획이 AI 도입을 ‘조직 운영 체계의 재설계’로 규정하고, 수년이 걸리는 기존 사업 절차와 구축·유지보수를 분리한 용역 방식이 AI의 속도에 맞지 않는다고 스스로 진단한 것은 매우 반가운 일이다. 문제는 실행이다. 기관마다 ‘AI 에이전트 구축 용역’이 따로 발주되면 공공 부문도 유지보수 지옥을 되풀이할 수 있다.

소수 정예의 ‘공공 FDE’를 제안한다. 우수한 인재를 선발해 민간 수준의 보상과 실질적인 권한·책임을 주고, 기관 현장에 깊이 들어가 핵심 AX 과제를 직접 풀게 하는 방식이다. 미국의 USDS(United States Digital Service)나 영국의 GDS(Government Digital Service)처럼, 기술 인재를 정부 안으로 불러들여 공공 서비스를 바꾼 전례도 있다. 애국심과 실력을 겸비한 젊은 엔지니어는 많다. 제대로 된 기회와 대우가 없을 뿐이다.

8. 맺으며

FDE는 “기존 조직보다 나아야” 존재 이유가 있는 직업이다. 부담스러울 만큼 어렵거나 많은 문제를 풀고, 부족한 도메인 지식은 고객에게서 빌리며, 매일 ‘와우 포인트’를 만들어야 한다. 쉬운 문제를 붙잡고 있으면 그 존재 이유는 모델이 한 번 좋아질 때마다 줄어든다. 그러니 결국 FDE라는 이름이 남을지 아닐지는 중요하지 않다. 조직 안에 난제를 자기 것으로 끌어안고 생각하는 사람이 몇 명이나 있는지가 그 기업의 AX 수준이 된다.

AI가 쉬운 문제를 하나씩 정복해 갈수록 사람에게 남는 일은 줄어드는 것이 아니라 어려워진다. 필자는 이것을 비관이 아니라 초대장으로 읽는다. 한국은 원래 새로운 기술을 누구보다 빠르게 받아들여 현장에 적용하고 발전시켜 온 나라다. 수천 명이 등기부등본을 일일이 입력하며 전산화를 이뤄 냈던 그 실행력이 지금 AI 전환의 마지막 1마일에 필요하다. 다만 이번에 필요한 것은 많은 손이 아니라, 아무도 정의해 주지 않은 문제 앞에서 먼저 손을 드는 소수의 사람이다. FDE는 그런 사람들을 담는 새로운 그릇이 될 수 있다.

참고문헌

- MIT NANDA (2025). The GenAI Divide: State of AI in Business 2025.
- Andreessen Horowitz (2025). “Trading Margin for Moat: Why the Forward Deployed Engineer Is the Hottest Job in Startups.”
- Palantir Technologies. “Forward Deployed Engineering” / AIP Bootcamp 소개 자료.
- 국가인공지능전략위원회. 「대한민국 인공지능 행동계획」.
- Jaipuria, T. “Build the Agent or Power the Agent?”