

EXECUTIVE SUMMARY

AI 코딩 도구의 개발자 채택률은 90%를 넘었습니다. "**Vibe Coding**"이라는 신조어가 등장한 지 채 1년이 되지 않았는데, 이미 그 말 자체가 옛말이 되어버렸습니다 — AI로 코딩하는 것이 더 이상 새로운 방식이 아니라 기본값으로 자리잡는 흐름입니다. 2026년 2월, AI는 "**대답하는 도구**"에서 "**일하는 도구**"로 전환되었으며, 시장은 이미 반응하고 있습니다. 에이전트와 모델은 **독립 변수**이며, 두 레이어를 함께 고려하면 보다 정확한 판단이 가능합니다. 이 리포트는 바깥이 어디까지 와 있는지를 살펴보고 체험 전 기준선을 형성하기 위한 것이며, 사외 AI Agent Hands-on과 세트로 설계되었습니다.

목 차

01 리포트 작성 배경

06 역할의 변화

02 시장 반응 — SaaSpocalypse

07 코딩을 넘어 — 동시다발적 확산

03 업계 기준선

08 기술적 가능성과 현실 제약의 분리

04 핵심 개념 — Model ≠ Agent

09 과제

05 패러다임 전환

10 향후 논의가 필요한 질문들

A 업계 데이터 상세

C 업계 리더 발언 종합

B 에이전트 시장 현황

D 참고 자료

01

리포트 작성 배경

본 자료의 업계 현황과 방향 판단은 작성자 개인의 의견이 아니라, Jensen Huang(NVIDIA CEO), Andrej Karpathy(OpenAI 공동창립자), Dario Amodei(Anthropic CEO), Andrew Ng(Google Brain 공동창립자), Demis Hassabis(Google DeepMind CEO), Addy Osmani(Google Cloud AI) 등 AI 업계 최전선 인물들의 공개 발언과 Anthropic·Google·GitHub 등이 발행한 공식 보고서를 기반으로 구성되었습니다. 주요 발언 원문은 부록 C에 수록되어 있습니다.

1881년, 조선에서는 서양 문물을 배척해야 한다는 위정척사운동이 거세게 일고 있었습니다. 같은 시기, 조정은 조사시찰단 62명을 **비밀리에** 일본에 파견했습니다. 이 시찰단은 일본의 육군성·해군성·내무성·공장·병원·학교를 직접 걸어 다니며 목격했습니다. 그들이 귀국 후 제출한 보고서에는 전신선이 전국을 가로지르고, 증기선이 항구를 오가며, 근대식 군대가 규율 있게 훈련하는 일본의 실상이 담겨 있었습니다. 3개월간 직접 시찰하고 돌아온 이들은 하나같이 근대신제도의 도입이 시급하다고 보고했습니다.

우리는 미래에서 과거를 보기 때문에 결과를 알고 있지만, 당시 조선의 입장에서 진짜 갈림길은 수용이냐 거부냐가 아니라, 실체를 직접 확인했느냐 아니냐의 차이였습니다. 시찰단이 남긴 진짜 교훈은, 수용이든 거부든 판단 전에 직접 확인하는 과정을 거쳤다는 점입니다.

이 리포트는 2026년 2월 현시점, 바깥이 어디까지 와 있는지를 함께 살펴보기 위한 것이며, 사외 AI Agent Hands-on과 세트로 설계되었습니다.

02

시장 반응

SaaSpocalypse (SaaS + Apocalypse)

사실 "AI Agent"라는 개념 자체는 이제 그렇게 새롭지 않습니다. 그동안 발표와 데모는 끊이지 않았고, 뉴스 헤드라인도 꾸준히 이어져 왔습니다. 그러나 뉴스에서 보여주는 것과 달리, 실제 업무에 적용하기에는 사용성이 미흡하거나 비용 문제가 뒤따랐습니다. 그런데 2026년 2월, 분위기가 달라졌습니다. 1인 개발자가 만든 오픈소스 에이전트 OpenClaw가 업계를 뒤흔들었고, Anthropic의 Claude Cowork/Code, OpenAI의 Codex가 일반 사용자에게 잇따라 출시되면서, 실제로 일을 해내는 수준의 에이전트가 사람들 손에 쥐어지기 시작했습니다.

2.1 "Something Big is Happening"

6년간 AI 스타트업을 운영해온 Matt Shumer(OthersideAI CEO)는 2월 11일 Fortune에 "Something Big is Happening"이라는 기고를 올렸고, 현재 수 천 만 회 이상 조회되었습니다. 그는 이 시점을 "2020년 2월"에 비유합니다. 해외에서 바이러스가 퍼지고 있다는 소식이 들렸지만, 화장지를 비축한다는 사람을 보면 과민반응이라고 생각했던 그때. 그리고 3주 만에 세상 전체가 바뀌었습니다. 그는 지금이 바로 "과장으로 보이는" 단계라고 말합니다.

그가 전하는 현실은 이렇습니다. 원하는 것을 평문으로 설명하고 4시간 후 돌아오면, 수만 줄의 코드가 작성되어 있고, AI가 스스로 앱을 열어 버튼을 클릭하며 테스트한 뒤, 자체 기준을 충족했을 때만 "준비되었습니다"라고 보고합니다. 그리고 테스트해보면 대부분 완벽합니다. 그는 이것이 월요일 하루의 일과였다고 적었습니다.

이런 경험을 직접 해본 사람들은 자연스럽게 상상력이 확장됩니다. "이게 법률 리서치도 하네?" "CRM(고객관계관리) 입력도 자동으로 되네?" — 그리고 곧바로 다음 질문이 따라옵니다. "그러면 지금 쓰고 있는 이 소프트웨어의 라이선스가, 계속 이만큼 필요한가?"

대부분의 사람들은 AI의 무료 버전을 사용하고 있는데, 무료 버전은 유료 사용자가 접근하는 것보다 **1년 이상 뒤쳐져** 있습니다. **무료 ChatGPT로 AI를 판단하는 것은 피처폰으로 스마트폰의 현재를 평가하는 것과 같습니다.** 대중의 인식과 현실 사이의 간극이 지금 가장 위험한 요소입니다 — 이 간극이 준비를 막고 있기 때문입니다.

— Matt Shumer, "Something Big is Happening"

2.2 SaaSpocalypse — 시장의 반응

Shumer의 기고보다 앞서, 시장은 이미 반응하고 있었습니다. 직접적 촉매는 Anthropic의 **Claude Cowork**였습니다. 1월 23일 연구 프리뷰로 먼저 공개된 이 에이전트 도구가, 2월 3일 **법률 업무 자동화 플러그인**을 추가 출시하면서 시장에 충격파를 보냈습니다. 투자자들은 법률 테크 플랫폼뿐 아니라 SaaS 섹터 전반에 대한 우려를 표출했고, 이날 하루에만 소프트웨어 섹터에서 약 2,850억 달러의 시가총액이 증발했습니다.

Salesforce, Adobe, ServiceNow 등 대표적인 SaaS 기업들은 2026년 들어 유독 큰 폭의 하락세를 보이고 있으며, 소프트웨어 섹터 전체에서 **1조 달러 이상**의 시가총액이 소실되었습니다. 이것이 **SaaSpocalypse**입니다. 그 배경에는, 20년간 SaaS의 매출 공식이었던 **더 많은 직원 = 더 많은 소프트웨어 라이선스**라는 등식이 흔들릴 수 있다는 판단이 있습니다. AI 에이전트 하나가 10명의 업무를 수행할 수 있다면, 10개의 소프트웨어 시트(seat)가 더 이상 필요하지 않기 때문입니다.

주가는 미래를 과대평가하기도, 과소평가하기도 합니다. 실제로 일부 애널리스트들은 이 매도 세가 과도하다고 평가하고 있으며, 이 수치 자체가 예언은 아닙니다. 다만 이 사건이 보여주는 것은, AI Agent가 더 이상 데모나 컨셉이 아니라 사람들이 직접 쓰고 판단을 내리는 대상이 되었다는 점입니다.

업계 기준선

지금 어디까지 와 있는가

3.1 채택률과 산출물

AI 코딩 도구의 개발자 채택률은 90%를 넘었습니다. Anthropic CEO Dario Amodei는 "우리 엔지니어 중에는 더 이상 직접 코드를 작성하지 않는 사람이 있다. 모델이 코드를 쓰고, 사람은 편집한다"고 밝혔습니다. 높은 채택률을 보이는 조직에서는 생산성이 두 배 이상 향상되었다는 보고가 나오고 있습니다. AI 코딩 도구는 이미 업계 표준으로 자리잡고 있으며, 채택 지연이 경쟁력 리스크로 연결될 가능성이 업계에서 지속적으로 논의되고 있습니다.

90%+

개발자 AI 도구
채택률

90%

AI 코드 작성 비율
(Amodei 전망)

110%+

높은 채택 조직
생산성 향상률

4.4h

Staff+ 엔지니어
주당 시간 절감

이 변화는 설문이 아니라 실제 산출물에서도 확인됩니다. FT(Financial Times) 분석에 따르면, 2022~2023년 거의 제로 성장이었던 지표들이 2024년 말부터 가파르게 상승하고 있습니다.

The past year has seen an explosion in coding productivity

Year-on-year growth in various indicators of coding output



Sources: Domain Name Industry Brief; SensorTower; GitHub. GitHub data is quarterly pushes

FT graphic: John Burn-Murdoch / @jburnmurdoch

©FT

출처: FT / John Burn-Murdoch (2026.2)

3.2 변화의 속도

- 2023 코드 자동완성
- 2024 채팅 기반 코드 생성
- 2025 에이전트 – 수십 분~수시간 자율 작업
- 2026 (현재) 멀티에이전트 협업 – 수시간~수일 자율 작업, IDE 통합
- 2027~28 (예측) 모듈 단위 자율 개발

에이전트가 단독 작업을 넘어 팀 단위 병렬 작업으로 진화하는 변곡점에 서 있습니다. 이 변화의 속도와 방향에 대해서는 AI 업계 최전선의 인물들이 놀랄 만큼 일치된 시각을 보이고 있습니다. 의견이 갈리는 지점은 **변화가 올 것인가**가 아니라 **얼마나 빨리, 어디까지 올 것인가**입니다.

수주의 간격이 의미 있는 차이를 만들 수 있는 속도라는 점은, 이 변화를 바라볼 때 참고할 만한 맥락입니다.

붉은 여왕 효과(Red Queen Effect)라는 개념이 있습니다. 『거울 나라의 앤리스』에서 붉은 여왕이 앤리스에게 한 말에서 유래한 것으로, "이곳에서는 같은 자리에 머물기 위해서라도 있는 힘껏 달려야 한다"라는 원리입니다. 주변이 빠르게 움직이는 환경에서는, 멈춰 있는 것이 곧 뒤처지는 것이 됩니다.

AI 업계의 변화 속도가 이 구조와 닮아 있습니다. 속도보다 방향이 중요한 시점도 있지만, 지금은 보폭 자체를 만들어가는 시점이기도 합니다 — 조직 차원에서도, 각자가 AI 도구 활용 감각을 쌓아가는 개인 차원에서도.

3.3 컴파일 언어·임베디드 영역

웹 서비스 중심 기업에서 먼저 폭발적으로 확산된 AI 코딩 도구가, 컴파일 언어·네이티브 개발 환경으로 빠르게 확장되고 있습니다. 임베디드·펌웨어 영역은 아직 개화 초기이지만, 이미 무시할 수 없는 수준의 변화가 관측되고 있습니다.

AI가 AI를 만든다. OpenAI는 GPT-5.3-Codex가 자기 자신의 개발에 사용된 최초의 모델이라고 밝혔습니다. 초기 버전이 자체 학습 파이프라인의 디버깅, 배포 관리, 테스트 결과 진단에 활용되었습니다. AI가 AI를 만드는 재귀적 구조가 이론이 아닌 실제 제품 개발에서 실현되고 있는 것입니다.

AI가 컴파일러 영역에 진입하고 있다. LLVM과 GCC 커뮤니티는 AI 도구 기여 정책을 공식 수립하는 단계에 있으며, 컴파일러 — 소프트웨어의 가장 깊은 기반 — 에서도 AI의 역할이 확대되고 있다는 신호가 나타나고 있습니다.

임베디드 전용 AI 도구의 등장. Embedder와 같은 AI 펌웨어 엔지니어링 플랫폼이 등장하여, C/C++·Rust에 대한 실시간 컴파일 진단, MCU 드라이버 자동 생성, 플래싱 및 디버깅을 지원하기 시작했습니다. AI Agent가 C/C++ 코드 분석, 리팩토링, 테스트 생성에서 이미 유의미한 성능을 보이고 있으며, 확산 속도는 가속되고 있습니다.

적용 속도와 경로는 분야별로 다를 수 있습니다. 다만 이 흐름이 웹 서비스에서 시작해 캠 파일러·펌웨어 영역까지 확장되고 있다는 점은, 우리 맥락에서도 참고해볼 만한 지점입니다.

04

핵심 개념

Model ≠ Agent

4.1 구조

모델과 에이전트는 독립 변수입니다. SSD에 비유하면 모델은 NAND이고, 에이전트는 Solution에 해당합니다.

역할	SSD 비유	ANTHROPIC	GOOGLE	OPENAI
회사	Samsung	Anthropic	Google	OpenAI
브랜드	Samsung SSD	Claude	Gemini	GPT
모델 (제품 군)	990 PRO / 870 EVO / PM9A3	Opus / Sonnet / Haiku	Gemini 3 Pro / Flash	GPT-5.3-Codex
에이전트 (구동 시스 템)	Firmware	Claude Code, Cowork	Antigravity, Gemini in Workspace	Codex (클라우드 + 데스크톱)

NAND가 아무리 좋아도 Solution이 못 살리면 성능이 안 나오고, Solution이 아무리 정교해도 NAND 성능이 받쳐주지 않으면 한계가 있습니다. 마찬가지로, "Claude Code가 Antigravity보다 낫다"는 실제로 Claude Code + Opus 4.6 조합과 Antigravity + Gemini 3 Pro 조합의 비교입니다.

에이전트와 모델은 독립 변수이며, 어느 한쪽만으로 전체 성능을 설명하기는 어렵습니다. 이 구분이 명확해지면 도구 선정·벤치마크·투자 판단에서 좀 더 정밀한 논의가 가능해집니다.

이 독립성을 보여주는 최근 사례가 있습니다. OpenClaw는 1인 개발자가 만든 오픈소스 에이전트 프레임워크로, 모델 자체가 아니라 모델을 오픈소스로 제공하는 레이어에 해당합니다. OpenAI와 Meta가 이 개발자의 영입을 두고 경쟁했고, 결국 OpenAI에 합류한 것으로 보도되었습니다(2026.2.15). 모델을 만드는 기업들이 에이전트 레이어의 역량을 별도로 확보하려 한다는 점은, 두 변수가 서로 대체 불가능한 독립적 가치를 갖고 있음을 시사합니다.

4.2 성능 공식

$$\text{코딩 성능} = \text{Model} \times \text{Agent(Harness)} \times \text{Pilot}$$

세 변수 중 하나라도 0이면 전체가 0입니다.

* Agent Harness: 모델의 도구·메모리·상태를 관리하며 자율 실행을 가능하게 하는 운영 계층

최고의 NAND를 탑재해도 Firmware가 부족하면 SSD 성능이 안 나오듯, 최고의 모델을 써도 에이전트가 부족하면 성능이 안 나오고, 둘 다 갖춰도 Pilot이 무엇을 만들어야 하는지 정의하지 못하면 의미가 없습니다.

05

패러다임 전환

5.1 무엇이 달라졌나

챗봇 (CHATGPT 웹, CLAUDE 웹)		에이전트 (CLAUDE CODE, CODEX, COWORK)
환경 인식	사용자가 복사한 텍스트만 봄	프로젝트 전체 파일시스템·업무 도구 접근
실행 능력	텍스트 출력만	파일 생성/수정, 터미널 실행, 외부 도구 연동
작업 범위	1회 질의-응답	다단계 자율 실행 (수시간~수일)
오류 대응	사용자가 예러를 복사해서 재질문	자체 예러 감지 → 수정 → 재시도
비유	전화 상담원	파견된 시니어 개발자

SaaSpocalypse를 촉발한 것이 바로 이 전환입니다. AI가 **대답하는 도구**에서 **일하는 도구**로 바뀌면서, 법률 리서치, CRM(고객관계관리) 관리, 데이터 분석 같은 전문 업무를 자율 수행하기 시작했습니다. 시장은 **소프트웨어를 사용하는 인간**이 아니라 **일을 수행하는 소프트웨어**의 시대 가능성에 반응한 것입니다.

직접 사용해본 것과 설명을 들은 것 사이의 간극이 가장 크게 느껴지는 영역이기도 합니다. Hands-on을 함께 기획한 이유 중 하나입니다.

역할의 변화

"무엇을 만들 것인가"가 새로운 병목

6.1 구현 비용의 붕괴

AI 코딩 에이전트가 구현 비용을 극적으로 낮추면서, **무엇을 만들어야 하는지 정의하는 능력**이 새로운 병목이 되고 있습니다. 코드를 빠르게 짜는 것은 이제 AI가 대신할 수 있지만, 풀어야 할 문제를 발견하고 정의하는 것은 여전히 사람의 영역입니다. Jensen Huang의 표현대로, 코드 작성은 점점 값싼 상품이 되고 있고, 정말로 중요한 것은 어떤 질문을 할 수 있느냐입니다.

6.2 설계·기획 비중의 확대

일부 선도 조직에서는 구현 중심의 역할이 설계·기획 중심으로 재편되는 사례가 관측되고 있습니다. Karpathy가 정의한 **Agentic Engineering** — 에이전트의 레버리지를 확보하되 품질 타협은 없는 전문적 AI 기반 개발 — 이 여러 선도 조직에서 새로운 기준으로 채택되는 추세입니다.

Karpathy 본인의 워크플로우가 2025년 11~12월 사이 수동 80% → 에이전트 80%로 역전되었으며, 이를 "20년 프로그래밍 커리어에서 가장 큰 변화"라고 표현했습니다. 그는 LLM 에이전트 역량이 이 시기 "모종의 일관성 임계점"을 넘었으며, 점진적 개선이 아니라 상전이 (phase change)로 진단하고 있습니다.

단, 그조차 모델의 약점을 명확히 지적합니다. **잘못된 가정 위에 계속 구축, 명확화 질문 없음, 과도한 추상화, 아첨적 경향**. 이 점에서 Pilot의 역할은 줄어드는 게 아니라 오히려 달라지는 방향으로 보입니다.

6.3 PM:Engineer 비율의 역전

AI 업계의 대표적 석학인 Andrew Ng(Google Brain·Coursera 공동창립자)은 2025년 AI Startup School 강연에서 직접 관측한 사례를 공유했습니다. 엔지니어링 시간이 3주 → 1일로 압축되자, 제품 기획과 사용자 피드백 수집이 새로운 병목으로 부상했다는 것입니다. 전통적 PM:Engineer 비율 1:6~7에서, 그의 팀에서는 2:1(PM:Engineer) 배치 제안이 처음으로 나왔습니다.

이 제안이 정답인지는 아직 열린 질문이지만, 구현이 아니라 기획이 병목이 되는 시대가 시작되었다는 신호임은 분명합니다. Ng은 한 달에 코드베이스를 3번 완전히 재작성하는 팀의 사례를 소개하며, 아키텍처 선택이 더 이상 되돌릴 수 없는 결정이 아니게 되었다고 지적합니다.

6.4 새로운 병목 – 인지 노동의 증가

구현이 빨라졌다고 해서 모든 것이 편해진 것은 아닙니다. AI가 생성해내는 수많은 산출물에 대한 **판단과 평가** – 인지 노동의 영역은 오히려 크게 증가하고 있습니다.

Harvard Business Review는 2026년 2월 "AI Doesn't Reduce Work – It Intensifies It"이라는 연구를 발표했습니다. 200명 규모 테크 기업 근무자 40명을 8개월간 추적한 결과, AI 도입 후 직원들은 더 빠른 속도로, 더 넓은 범위의 업무를 수행했습니다. 문제는 **절약된 시간이 휴식이나 깊은 사고로 돌아가지 않고 즉시 다른 업무로 채워졌다**는 것입니다. 연구진은 이를 "**업무량 잠행 증가(workload creep)**"라 명명하며, 단기적 생산성 향상이 인지 피로와 의사 결정 품질 저하로 이어질 수 있다고 경고합니다.

개발 영역에서는 이 현상이 더 구체적으로 관측됩니다. METR(Model Evaluation & Threat Research)의 무작위 대조 시험(RCT)은 숙련된 오픈소스 개발자 16명을 대상으로, AI 도구 사용 시 실제 작업 완료 시간이 19% 더 길어졌다는 결과를 보고했습니다. 더 주목할 점은, 개발자 본인들은 AI가 자신을 20% 빠르게 해주었다고 체감했다는 것입니다. **인지적 착각과 실제 성과 사이의 괴리**가 존재하며, 이는 AI 생성물의 검토·수정·통합에 소요되는 비가시적 인지 노동이 과소평가되고 있음을 시사합니다.

Google의 DevOps 연구 프로그램인 **DORA**(DevOps Research and Assessment)의 2025년 보고서도 같은 구조를 확인합니다. 개인 단위 산출량(PR 수, 완료 태스크)은 증가했지만, **조직 수준의 소프트웨어 딜리버리 지표는 정체되어 있습니다.** DORA는 AI를 "**증폭기(amplifier)**"로 규정합니다 — 성숙한 엔지니어링 프랙티스를 갖춘 조직에서는 효과가 가속되지만, 기술 부채와 프로세스 병목이 있는 조직에서는 오히려 문제를 악화시킨다는 것입니다.

AI가 빠르게 가져가는 것은 구현 노동 쪽입니다. 반면 **생성물을 판단하고, 맥락에 맞게 검증하고, 품질을 보증하는 인지 노동**은 줄어들기보다 성격이 달라지고 있습니다 — 그 비중도 함께.

6.5 숙련 엔지니어의 역할 증대와 주니어의 도전

인지 노동의 비중이 커질수록, **숙련된 시니어 엔지니어의 역할은 증대됩니다.** Addy Osmani가 지적한 대로, 시스템 설계·보안·성능에 대한 깊은 이해를 가진 엔지니어에게 AI는 "거대한 힘의 배울기"로 작용합니다. AI가 구현을 더 많이 처리할수록 아키텍처 사고와 보안 인식에 대한 프리미엄은 낮아지는 것이 아니라 오히려 높아집니다.

반면, 주니어 개발자들은 구조적으로 더 어려운 환경에 놓이고 있습니다. 2025년 LeadDev 조사에 따르면 엔지니어링 리더의 **54%**가 AI 코파일럿 덕분에 주니어 채용을 줄일 계획이라고 응답했습니다. AI가 주니어의 학습 기회였던 저위험 업무(UI 수정, 단위 테스트 등)를 자동화하면서, 성장의 사다리 자체가 흔들리고 있는 것입니다. 동시에 시니어 엔지니어의 코드 리뷰 부담은 가중되고 있습니다. AI 도구를 활용한 주니어가 더 많은 코드를 더 빠르게 생산하지만, 그 산출물의 품질을 검증하는 책임은 시니어에게 집중됩니다.

주니어의 진입 기회가 줄고, 시니어의 검증 부담이 늘어나는 경향은 업계 전반에서 관측되고 있습니다.

07

코딩을 넘어

동시다발적 확산

AI의 영향은 코딩 영역에 국한되지 않습니다. 같은 시기에 인접 산업에서 유사한 전환이 관측되고 있으며, 이 확산은 우리의 수요 구조와도 연결됩니다.

7.1 영상·콘텐츠 제작

2026년 2월 12일 ByteDance가 **Seedance 2.0**을 출시하자, MPA(미국영화협회), 디즈니, 파라마운트, SAG-AFTRA가 잇따라 법적 대응과 공식 규탄에 나섰습니다. 텍스트만 입력하면 대사·효과음·립싱크까지 완비된 영상이 나오는 수준이 되자, 할리우드가 즉각 반응한 것입니다. 특히 Seedance 2.0은 기존 AI 영상 도구와 달리 **상업 콘텐츠에 즉시 적용 가능한 수준의 퀄리티**로 업계를 놀라게 하고 있습니다. AI 영상 생성이 이론적 가능성에서 산업적 현실로 넘어왔음을 보여주는 사건입니다. ([예시 영상 →](#))

그동안 AI 영상 생성은 일부 얼리 어답터의 실험적 놀이이거나, 짧은 클립을 만들어 SNS에 공유하는 재미 요소에 가까웠습니다. 그러나 품질이 임계점을 넘으면서, 이 기술이 일반 사용자의 일상적 도구로 다가오는 것은 시간문제로 보입니다. 코딩 에이전트가 개발자의 업무 방식을 바꾸고 있듯, 영상 AI 역시 콘텐츠 제작의 문턱을 근본적으로 낮추는 방향으로 확산될 것입니다.

7.2 Science AI

2024년 노벨 화학상이 AI 단백질 구조 예측 연구(Google DeepMind의 AlphaFold 등)에 돌아간 데 이어, 2025~2026년에는 AI가 기존 지식의 정리를 넘어 **새로운 지식을 창출**하기 시작했습니다. GPT-5.2는 이론 물리학에서 새로운 공식을 도출하고 **12시간** 만에 자체 증명을 완성했으며, Gemini 3 Deep Think은 코딩 경진대회(Codeforces)에서 전 세계 상위 **8등** 수준을 기록했습니다.

7.3 우리와의 접점 – 메모리 수요의 구조적 전환

텍스트 LLM 시대에서 데이터센터의 메모리 수요는 선형적으로 증가해왔습니다. 모델이 커지고 사용자가 늘어도, 입력과 출력이 텍스트인 한 스토리지 부하의 증가는 예측 가능한 범위 안에 있었습니다. 이 구조가 근본적으로 바뀌고 있습니다.

단계	텍스트 LLM	영상·멀티모달 AI	배율
훈련 데이터	수십~수백 PB	수백~수천 PB	10배 +
추론 시 리소스	기준 1x	~10x	10배
생성물 크기	수 KB	수백 MB~수 GB	수 만 배
KV 캐시 (추론 중 임시 메모리)	수 GB	수백 GB~TB급	수 십 ~ 수 백 배

텍스트를 생성하는 것과 영상을 생성하는 것은 리소스 차원이 다릅니다. Seedance 2.0 같은 영상 생성 모델이 확산될수록, 훈련·추론·저장의 모든 단계에서 스토리지 소요가 폭증합니다. NVIDIA의 차세대 AI 플랫폼이 대규모 KV 캐시를 GPU HBM이 아닌 SSD 기반 플래시 스토리지로 오프로드하는 구조를 선택한 것은, 이 수요 곡선이 HBM만으로는 감당할 수 없는 규모로 진입하고 있다는 신호입니다.

LLM이 데이터센터의 메모리 수요를 선형적으로 키웠다면, 월드 모델(World Model – 물리 법칙·공간·시간을 이해하고 시뮬레이션하는 AI)과 영상 AI는 그것을 지수적으로 키우는 구조입니다. AI가 발전할수록 메모리 수요가 함께 증가하는 경향은 당분간 지속될 가능성이 높습니다.

기술적 가능성과 현실 제약의 분리

AI Agent의 역량을 판단할 때, **기술적으로 가능한가와 우리 환경에서 적용 가능한가는** 별개의 질문입니다. 이 두 가지를 분리하면 보다 정확한 판단이 가능해집니다.

8.1 세 가지 영역

A

기술적으로 가능하고 적용도 가능

지금 당장 우리 환경에서 실행할 수 있는 것

예: 사외망 PoC 환경에서의 AI 코드 분석·리팩토링·테스트 생성

B

기술적으로 가능하지만 제약으로 적용 불가

기술은 있지만 보안·인프라·프로세스가 막는 것

예: Agent의 JIRA/Confluence 연동, 사내망 프론티어 모델 직접 사용

C

기술적으로 아직 불가능

현재 AI 기술 자체의 한계

예: 100% 신뢰할 수 있는 자율 아키텍처 설계, 도메인 완전 이해

실무에서 의미 있는 논의가 모이는 지점은 주로 **B**입니다. "안 된다"고 판단된 것 중 상당수가 기술 한계(C)가 아닌 환경 제약(B)이기도 합니다. 물론 제약이라고 해서 쉽게 해결되는 건 아닙니다. 다만 한계는 기술 성숙을 기다리는 영역이고, 제약은 논의의 여지가 있는 영역입니다 — 이 구분이 있으면 우선순위 대화가 조금 더 수월해집니다.

8.2 우리 환경에의 적용

제약 유형	현재 상태	해결 경로
사내망 보안	프론티어 AI 직접 사용 불가	사외망 별도망 구축 (진행 중)
코드 보안 레벨	외부 서비스에 코드 노출 우려	코드별 보안 레벨 재점검 및 코드 분류 (진행 중)
인프라·형상관리	기존 프로세스와 미연동	CI/CD 통합 파일럿 (계획)
SOP·프로세스	AI 산출물을 직접 확인하는 구조	품질 게이트·리뷰 기준 재설계 (설계 필요)
조직 인식	체감 부재로 관망	← 이 리포트와 Hands-on의 목적

사외 AI Agent Hands-on이 사외망에서 진행되는 이유가 여기에 있습니다. 영역 B의 제약을 일시적으로 제거한 환경에서 영역 A의 가능성을 직접 확인하는 것입니다. 기술적으로 무엇이 되는지를 먼저 확인하면, 우리 환경에서 어떤 제약을 어떤 순서로 풀어갈지에 대한 논의로 자연스럽게 이어질 수 있습니다.

09

과제

현재 사외 SOTA(State-of-the-Art) 서비스에 대한 PoC가 진행 중입니다. 동시에, 조직 차원에서 실질적으로 활용하려면 함께 풀어야 할 과제들이 뒤를 잇고 있습니다.

내부 시스템과의 통합. Agent가 JIRA, Confluence, Bitbucket 등 사내 서비스와 연동될 때 비로소 이슈 관리, 문서화, 코드 리뷰, 형상관리를 포함한 전체 개발 워크플로우가 재구성될 수 있습니다. 기술적으로는 이미 실현 가능한 영역이지만(영역 B), 사내 보안 정책과 인프라 환경 안에서 어떻게 적용할 수 있는지를 판단해야 합니다.

일하는 방법의 변화. Agent 도입은 단순한 도구 교체가 아니라 업무 프로세스 자체의 변화를 수반합니다. 코드 리뷰 기준, 품질 게이트, 일정 산정 방식, 산출물 검증 체계 등 기존 SOP 전반에 대한 재검토가 수반될 수 있습니다.

CI/CD 파이프라인 통합. AI가 생성한 코드가 기존 빌드·테스트·배포 파이프라인에 안정적으로 통합되려면, 파일럿 프로젝트를 통한 검증과 단계적 확대가 필요합니다.

리소스와 비용. 사외 프론티어 서비스 사용에 따른 라이선스 비용, 사외망 인프라 확장, 보안 검토 인력, 교육 리소스 등 조직 차원의 투자 판단이 수반됩니다.

가능한 것의 범위가 먼저 그려지면, 제약의 무게를 가늠하는 것도 한결 수월해집니다.

향후 논의가 필요한 질문들

아래 질문들은 이 자리에서 즉답을 구하기 위한 것이 아닙니다. 사외 AI Agent Hands-on을 통해 직접 체험한 후 각자의 도메인 단위로 다양한 의견을 도출하는 것을 목표로 합니다.

영역	질문
준비	AI 시대에 우리 조직은 무엇을 준비해야 하는가? 어떤 SOP를 새롭게 구축해야 하는가?
자율화	현재 업무 중 AI Agent에게 위임할 수 있는 영역은 어디인가?
제약 하의 실행	사내망·보안·프로세스 제약 하에서 지금 당장 할 수 있는 것은 무엇인가?
가능성과 제약의 분리	"안 된다"고 판단한 것 중, 기술적 한계(C)인가 환경적 제약(B)인가?
역량 전환	구성원에게 요구되는 역량이 어떻게 달라지며, 어떤 순서로 전환할 것인가?

들어서 아는 것이 아니라 해봤으니 안다고 답할 수 있을 때, 조직 차원의 의사결정 논의가
실질적으로 이루어집니다.

이 리포트는 그 논의를 시작하기 위한 보조 자료이며, 아래의 사외 AI Agent Hands-on에서
직접 체험한 뒤 각자의 도메인에서 위 질문들을 함께 논의하는 자리로 이어집니다.

Hands-on 개요

구분	사내 교육	사외 HANDS-ON
목적	업계 현황·핵심 개념 정리· 시연·토론	AI Agent 직접 체험
시간	2시간	4시간 (반나절)
장소	SW개발팀 회의실	동탄에듀센터 (사외망)
체험 내용	Agent 동작 시연, 기술 한계 vs 환경 제약 토론	Claude Cowork (업무 자동화), Claude Code (코드 분석·리팩토링), 관심 업무 자유 체험
기대 산출물	조직별 AI 적용 가능 영역 1~2개 식별, 되는 것 / 막혀 있는 것 / 아직 안 되는 것 구분 기 준 형성	

대상 범위·일정·회차는 별도 협의 예정입니다. 사외망에서 진행하는 이유는 환경 제약을 일시적으로 제거한 상태에서 기술적 가능성의 범위를 확인하기 위함입니다.

마치며

이 리포트는 AI Agent를 활용하여 작성되었습니다. 자료 수집과 인사이트 도출은 NotebookLM, 관련 리포트·사례 조사는 Claude Cowork, HTML 문서 구성은 Claude Code로 처리했습니다. 작성자가 직접 타이핑한 본문은 없으며, 역할은 구조 설계, 팩트체크, 최종 검증이었습니다. 수백 건의 해외 리포트·뉴스·기술 문서를 참조했는데, Agent가 초안을 만들수록 사람의 역할은 작성에서 검증으로 이동했습니다 — DORA에서 실제로 조사한 내용인지, FT에 기고된 실제 뉴스인지, Fortune지에 실제 등록된 기사인지를 확인하는 것이 Pilot의 일이었습니다. 본문에서 다른 인지 노동의 증가를 직접 체감한 과정이기도 했습니다.

이 문서의 워크플로우를 Skill로 복제하면, 다음에 같은 종류의 리포트를 작성할 때 소요 시간은 수분의 1로 줄어들 수 있습니다. 리서치에 가까운 문서이기에 가능한 방식이고, 모든 종류의 문서에 적용할 수 있는 것은 아닙니다. 다만, 이전이라면 한 사람이 시도조차 어려웠을 범위의 작업이 가능해졌다는 것 — 이것 역시 본문에서 다룬 변화의 일부입니다.

A

업계 데이터 상세

본문 3.1의 산출물 지표(FT 분석)는 변화의 방향과 추세를 보여주기 위한 것이며, 아래는 채택률과 생산성에 대한 정량 데이터입니다.

지표	수치	출처
개발자 AI 도구 채택률	90~91%	2025 DORA Report, DX Q4 Report
AI 코드 작성 비율 전망	3~6개 월 내 90%	Dario Amodei, Anthropic CEO (2025.3)
Staff+ 엔지니어 주당 시간 절감	4.4시간	DX Q4 2025 Report
80~100% 채택 조직 생산성 향상	110%+	Jellyfish-OpenAI 공동연구

B

에이전트 시장 현황

2026. 2

본문 4.1의 비유를 기준으로, 모델(NAND)과 에이전트(Firmware)를 분리하여 정리합니다.

주요 플랫폼별 모델 × 에이전트 조합

회사	모델	에이전트	아키텍처	가격
Anthropic	Opus 4.6 / Sonnet 4.5 / Haiku 4.5	Claude Code	터미널 기반, Agent Teams (병렬)	\$100~200/ 월
		Claude Cowork	데스크톱 앱, 파일시스템·브라우저 제어	
OpenAI	GPT-5.3-Codex	Codex	클라우드 + macOS 데스크톱, 멀티에이전트 병렬 실행	구독제
Google	Gemini 3 Pro / Flash	Antigravity	IDE 기반, Jules (GitHub 연동)	무료 Preview
		Gemini in Workspace	Gmail·Docs·Sheets 통합	

IDE 통합 에이전트

특정 모델에 종속되지 않고 여러 모델을 교체하여 사용할 수 있는 범용 에이전트입니다.

도구	특징	가격
Cursor 2.0	최대 8 에이전트 병렬, Plan Mode	~\$200 / 월
Windsurf	Arena Mode(모델 비교), 멀티에이전트	~\$60 / 월
Kimi Code	오픈소스, Agent Swarm	무료

같은 NAND를 써도 Firmware에 따라 SSD 성능이 달라지듯, 같은 모델이라도 어떤 에이전트 위에서 구동하느냐에 따라 실제 작업 성능이 크게 달라집니다. 역으로, 같은 에이전트라도 어떤 모델과 조합하느냐에 따라 결과가 달라집니다. **도구 선정 시 모델이나 에이전트 어느 한쪽만 비교하는 것은 불완전한 평가입니다.**

C

업계 리더 발언 종합

이 변화의 속도와 방향에 대해서는 AI 업계 최전선의 인물들이 놀랄 만큼 일치된 시각을 보이고 있습니다. 아래는 본문에서 인용·참조된 발언을 중심으로 원문을 정리한 것입니다.

Jensen Huang

NVIDIA CEO, 2025 — 본문 6.1, 7.3 참조

"코드를 작성하는 것은 본질적으로 타이핑이고, 타이핑은 점점 값싼 상품이 되고 있다. 정말로 중요한 것은 답이 아니라, 어떤 질문을 할 수 있느냐다."

"소프트웨어 엔지니어의 목적은 코딩이 아니다. 알려진 문제를 풀고, 아직 발견되지 않은 문제를 찾는 것이다."

"모든 국가, 모든 산업, 모든 기업을 휩쓸 기술 혁명."(COMPUTEX 2025)

"공장을 가진 모든 기업은 미래에 두 개의 공장을 갖게 될 것 — 제품 생산을 위한 공장과 AI를 위한 공장."(GTC 2025)

Andrej Karpathy

OpenAI 공동창립자, 전 Tesla AI 총괄 — 본문 6.3 참조

2025년 초 "Vibe Coding" 용어 제시 → 2026년 1월 "Agentic Engineering"으로 전환 선언.

"99%의 시간 동안 직접 코드를 작성하지 않고, 에이전트들을 오케스트레이션하며 감독하는 것"

본인의 워크플로우가 2025년 11~12월 사이 수동 80% → 에이전트 80%로 역전. "20년 프로그래밍 커리어에서 가장 큰 변화". LLM 에이전트 역량이 "모종의 일관성 임계점"을 넘었으며, 점진적 개선이 아니라 상전이(phase change)로 진단.

단, 모델의 약점도 명확히 지적: 잘못된 가정 위에 계속 구축, 명확화 질문 없음, 과도한 추상화, 아첨적 경향.

Dario Amodei

Anthropic CEO — 업계 방향성의 타임라인 참조

2025.3 CFR 인터뷰: "우리 엔지니어 중에는 더 이상 직접 코드를 작성하지 않는 사람이 있다. 모델이 코드를 쓰고, 사람은 편집한다" / "3~6개월 내에 AI가 코드의 90%를 작성하는 세계에 도달할 것"

2025.1 에세이 "On DeepSeek and Export Controls": "거의 모든 것에서 거의 모든 인간보다 상당히 더 똑똑한 AI를 만드는 데에는 수백만 개의 칩과 수백억 달러가 필요하며, 이것이 실현되는 시점은 2026~2027년이 가장 유력하다"

2026.1 WEF 다보스: "6~12개월 내에 소프트웨어 엔지니어가 하는 대부분, 어쩌면 전부를 모델이 앤드투엔드로 수행할 수 있게 될 것"

Andrew Ng

Google Brain 공동창립자, DeepLearning.AI — 본문 6.2 참조

전통적 PM:Engineer 비율 1:6~7에서, 자신의 팀에서 2PM: 1Engineer 배치 제안이 최초로 등장.

"엔지니어링이 빨라지면서 제품 관리 업무가 점점 더 병목이 되고 있다."

한 달에 코드베이스를 3번 완전 재작성하는 팀 사례 — 코드 자체의 가치가 하락.

Demis Hassabis

Google DeepMind CEO, 노벨상 수상자 — 본문 3.2, 7.2 참조

"코딩이나 수학 같은 영역에서는 [AI 자기 개선 루프가] 확실히 작동할 수 있다. 결과물이 겸증 가능하기 때문이다."

2026년 에이전트 기반 시스템이 본격화하여 생산성을 근본적으로 변화시킬 것으로 전망.

Addy Osmani

Google Cloud AI, 전 Chrome 14년 — 본문 4.2, 6.3 참조

Agentic Engineering은 시니어 엔지니어에게 불균형적으로 큰 이점을 제공 — 시스템 설계·보안·성능에 대한 깊은 이해가 있는 사람이 AI를 "거대한 힘의 배율기(massive force multiplier)"로 활용.

"AI가 구현을 더 많이 처리할수록, 아키텍처 사고·보안 인식·시스템 설계에 대한 프리미엄은 낮아지는 것이 아니라 높아진다."

Ilya Sutskever

OpenAI 공동창립자, SSI CEO — AI 발전 속도의 균형 잡힌 시각 참조

"2020~2025년은 스케일링의 시대였다. 지금은 다시 연구의 시대로 돌아가고 있다. 다만 훨씬 큰 컴퓨터와 함께."

현재 모델은 벤치마크에서 인간을 넘지만, 실세계 일반화에서는 인간에 크게 못 미침.

D

참고 자료

자료	출처	본문 참조
2025 DORA Report	Google	3.1, 6.4, 부록 A
DX AI-assisted Engineering Q4 2025	DX	3.1, 부록 A
Jellyfish-OpenAI 공동연구	McKinsey 인터뷰	3.1, 부록 A
2026 Agentic Coding Trends Report	Anthropic	3.2
FT AI Impact on Software Output	John Burn-Murdoch (2026.2)	3.1
Matt Shumer — "Something Big is Happening"	Fortune (2026.2.11)	2.1
SaaSpocalypse 보도	Bloomberg, CNBC, Inc., Fortune 등 복수 매체 (2026.2)	2.2
OpenClaw / OpenAI 영입 보도	CNBC, TechCrunch, Bloomberg 등 복수 매체 (2026.2.15)	4.1
Andrew Ng — AI Startup School / Buildathon Keynote	DeepLearning.AI (2025)	6.2
Andrej Karpathy — Agentic Engineering (X post)	2026.1	6.3
Dario Amodei — CFR / WEF / 미 상원 증언	2025~2026	부록 C
Demis Hassabis — Axios AI+ Summit / Fortune	2025~2026	3.2, 7.2
Ilya Sutskever — Dwarkesh Podcast	2025.11	부록 C
Jensen Huang — No Priors / CES / COMPUTEX / GTC	NVIDIA, 2025	6.1, 7.3

자료	출처	본문 참조
Addy Osmani — Agentic Engineering / AI's 70% Problem	Google Cloud AI, 2025~2026	4.2, 6.3, 6.5
"AI Doesn't Reduce Work — It Intensifies It"	Harvard Business Review (2026.2)	6.4
METR — AI Impact on Experienced OS Dev Productivity (RCT)	METR (2025.7)	6.4
LeadDev — Junior Developer Hiring Survey	LeadDev (2025)	6.5