

2026 AI 코딩 현황 보고서

업계 인사이트

하노버 리서치(Hanover Research)와 뉴렐릭은 소프트웨어 및 애플리케이션 개발에 생성형 AI 또는 에이전틱 AI를 사용하는 조직의 관리자급 이상 기술 리더 200명을 대상으로 설문조사를 실시했습니다.

업계의 최신 벤치마크가 될 조사 결과는 소프트웨어 및 애플리케이션 개발에 어떻게 AI가 사용되고 있는지를 명확하게 보여줍니다. 마이크로소프트(Microsoft)에서는 현재 코드베이스의 30%를 AI가 작성하고 있으며(CNBC), 세일즈포스(Salesforce)의 CEO 마크 베니오프(Marc Benioff)는 현재 워크로드의 30%-50%를 AI 에이전트가 처리하고 있다고 말합니다(Bloomberg). 메타(Meta)는 2026년까지 AI 사용률 50%를 목표로 한다고 밝혔습니다(Mashable). 깃허브 코파일럿(GitHub Copilot)의 텔레메트리에 따르면 전체 사용자 기반에서 AI 코딩의 점유율이 평균 46%를 보이고 있습니다(Larridin). 최근 구글(Google)이 자그마치 코드의 75%를 AI로 생성했다고 발표했다는 사실은 특히 주목할만합니다(Business Insider).

설문에 참여한 기술 리더의 3분의 2(67%)는 조직의 주간 코드 결과물 중 51%-75%를 현재 AI가 생성하거나 대폭 리팩터링하고 있다고 답했습니다. 이제 많은 조직들은 AI가 작성한 코드를 기반으로 운영되고 있습니다.

이 보고서에서는 이러한 변화로 야기되는 여러 가지 결과를 세부적으로 살펴봅니다. 그중 일부는 긍정적입니다. 기능 배포 속도가 빨라지고 배포 빈도도 증가했으며, 대다수의 리더들은 AI가 생성한 코드가 사람이 작성한 코드보다 품질이 우수하다고 평가합니다.

일부는 부정적입니다. 운영 인시던트, 기술 부채, 마이크로서비스의 난립, 시니어 개발자의 재작업 시간이 모두 증가했습니다.

또한 AI가 생성한 코드로 인해 발생하는 가장 흔한 4가지 운영 장애 유형(통합 실패, 규정 준수 문제, 데이터 무결성 문제, 보안 취약점)은 지난 6개월 동안 각각 약 10개 중 3개 조직에 영향을 미쳤습니다.

흥미로운 결과

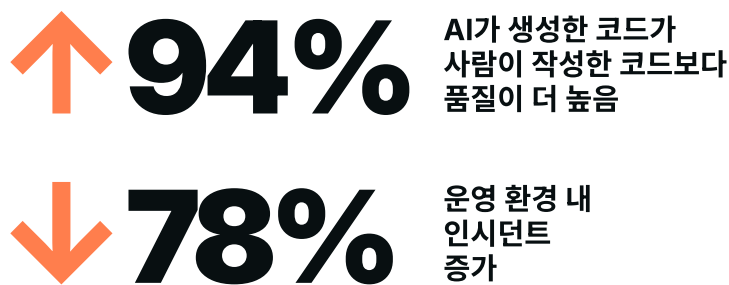
두 상황 간의 간극

AI가 생성한 코드는 코드 검토 단계에서는 품질이 더 우수하다고 평가받습니다. 하지만 운영 환경에 배포된 후에는 결과가 눈에 띄게 저조합니다. 이 보고서는 바로 이러한 간극에 대해 알아봅니다. 이는 2026년 오피버빌리티를 위한 전략적 기회가 되어줄 것입니다.

주요 결과

AI가 생성한 코드는 검토 단계 평가와 운영 환경 단계 평가 사이에 간극이 존재합니다.

이것이 AI로 지원되는 시대의 핵심적인 모순입니다. 코드 검토 단계에서는 94%가 AI가 생성한 코드가 사람이 작성한 코드보다 품질이 높다고 답했지만, 배포 후에는 78%가 더 많은 인시던트가 발생한다고 답했습니다. 86%는 시니어 개발자의 문제 해결 작업이 증가했다고 말했으며, 74%는 AI가 생성한 코드의 최소 25%에 대대적인 재작업이 필요하다고 답했습니다. 또한 82%는 지난 6개월 동안 이와 관련된 운영 장애를 최소 한 번 이상 겪었습니다.



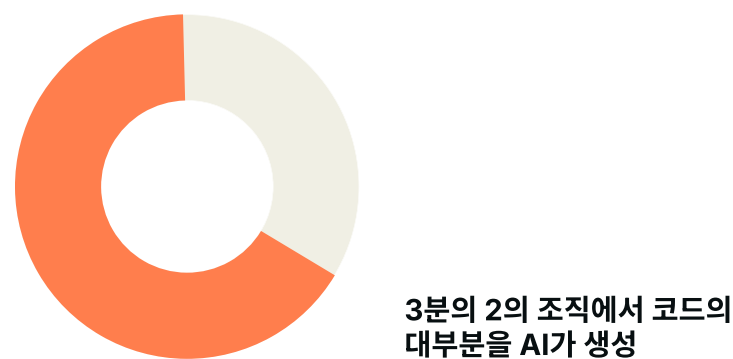
AI로 생성된 결과물을 검증 없이 신뢰하면 향후 운영 단계에서 위험할 수 있습니다.

바이브 코딩 시대의 핵심적인 모순은 개발 수명주기 초기 단계에 AI를 과도하게 신뢰해 버린 데서 비롯됩니다. 거의 3분의 2(62%)에 해당하는 기술 리더들은 팀이 AI가 생성한 코드를 신뢰하여 한 줄씩 수동으로 검증하지 않고 그대로 배포한다고 답했습니다. 이러한 광범위한 맹신은 초기 검토 단계에서 코드 품질이 왜 실제보다 인위적으로 높게 평가되고, 실제 운영 환경에 배포되자마자 왜 곧바로 무너져 내리는지를 잘 설명해 줍니다.



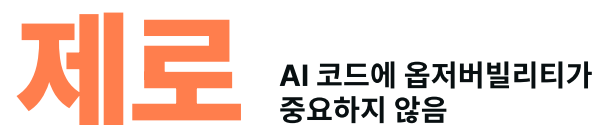
바이브 코딩의 주류화되어 운영에 도입되었고 구글의 볼륨 곡선에 근접하고 있습니다.

현재 3분의 2의 조직은 51%-75%의 코드를 AI로 생성하고 있고, 88%는 바이브 코딩을 공식적으로 운영 정책에 포함시켰습니다. AI 사용을 비운영 환경으로 제한한 조직은 단 5%뿐이며, 금지한 곳은 전혀 없습니다.



AI가 생성하는 코드에는 옹저버빌리티가 필수적이며, 텔레메트리도 프롬프트 입력을 통해 출력 결과에 포함되고 있습니다.

96%의 리더들은 AI로 생성된 코드에서 옹저버빌리티가 매우 또는 대단히 중요하다고 생각하며, 약간 중요하거나 중요하지 않다고答한 비율은 0%입니다. 이제 78%의 팀들은 생성된 코드에 텔레메트리(로그, 트레이스, 고객 메트릭)를 포함하도록 AI 프롬프트에 입력합니다.

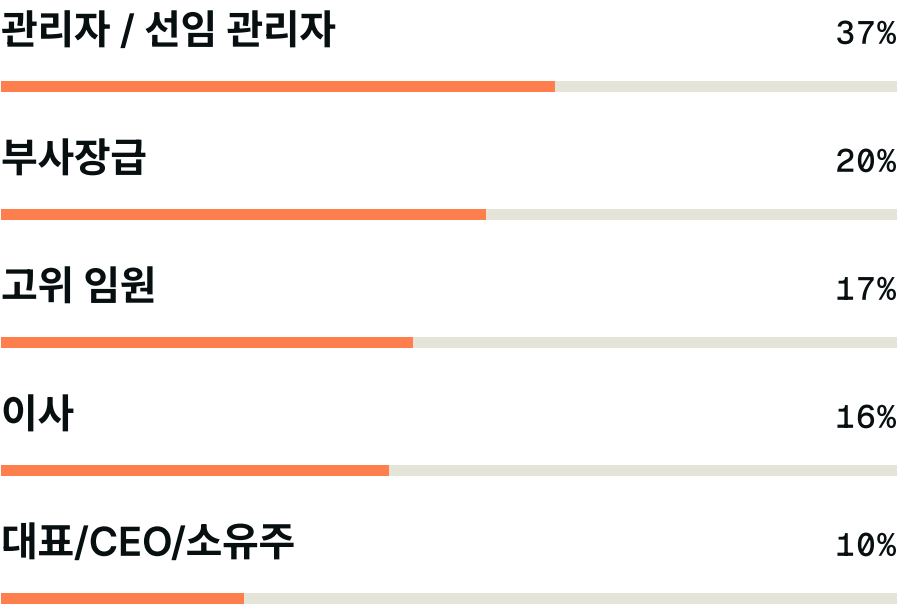


조사방식

2026년 뉴렐릭은 하노버 리서치(Hanover Research)에 온라인 설문조사를 의뢰했습니다. 조사 대상은 소프트웨어 개발에 생성형 AI를 사용하는 조직에서 근무하는 정규직 직원으로, 미국에 기반을 둔 IT 및 엔지니어링 분야 기술 의사결정권자 200명을 포함합니다. 모든 응답자는 관리자급 이상이며, 모두 소프트웨어 구매에 대한 실질적인 권한을 가지고 있습니다(65%는 주요 의사결정권자, 34%는 공동 의사결정권자).

이 보고서의 수치는 n=200을 기준으로 한 백분율로 표시됩니다. 본문의 주요 통계는 분포에서 가장 유용한 정보를 제공하는 세그먼트를 참조하며, 차트는 각 데이터 포인트의 이면에 있는 전체 분포를 보여줍니다.

응답자의 직급



산업 범위



매출 기준 대부분 중견기업 및 대기업

92%

500명 이상의 직원 보유

18%

연 매출 \$2억 5천만 - 4억 9천 9백만

37%

연 매출 \$5억 - 9억 9천 9백만

33%

연간 수익 \$10억 이상

뉴노멀

AI 코딩은 샌드박스 환경의
임시방편이 아니라 소프트웨어
개발 수명주기에 통합되었습니다.

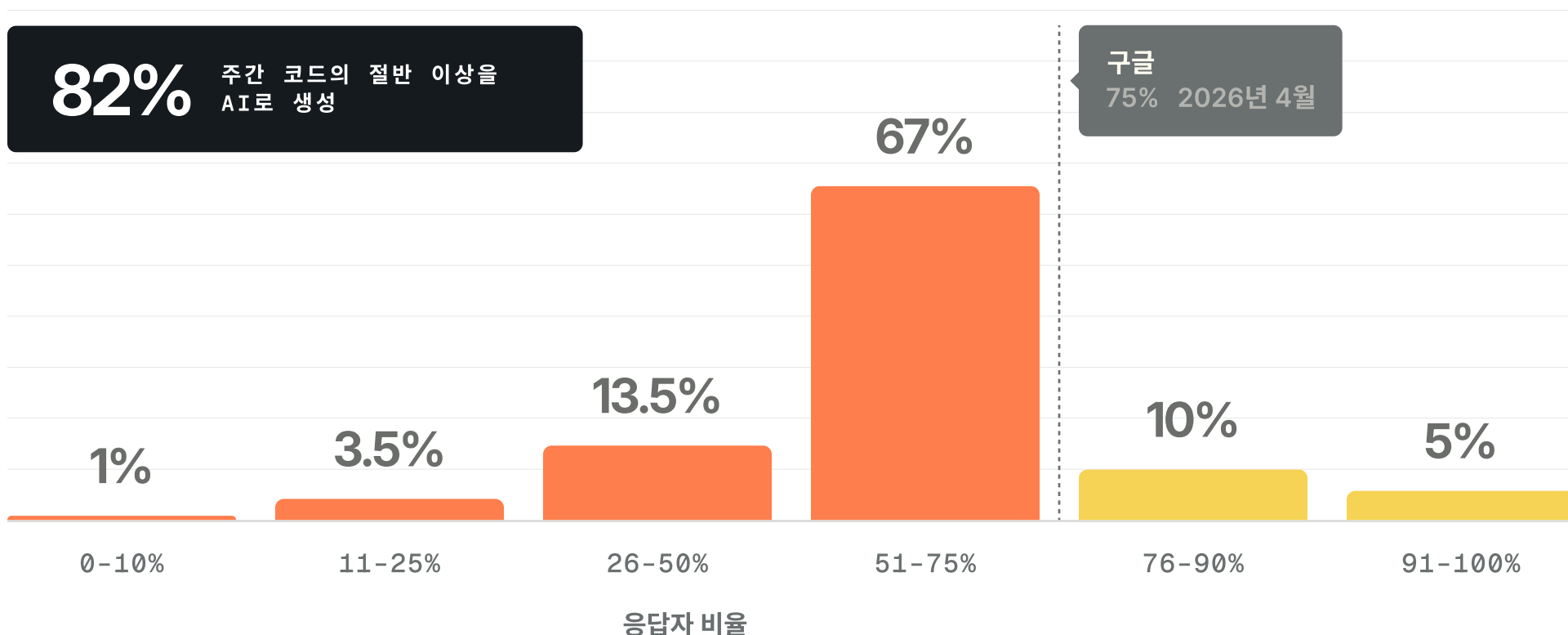


이제 AI로 지원되는 코딩이 표준으로 자리 잡았습니다.

67%의 리더는 조직이 매주 AI로 생성하는 코드가 전체 코드의 51%-75%에 달한다고 말했습니다.

주간 AI로 생성하거나 대폭 리팩터링한 코드의 비율

N=200



시사점

구글이 AI 코딩 비율이 75%라고 발표했을 당시, 대부분은 이상 수치가 아니냐는 의견이었습니다. 이 연구 조사의 데이터는 이를 재조명합니다. 이제 51%-75%의 비율은 미국 기술 리더들에게 보편적인 수치가 되었습니다. 대부분의 응답자의 경우, AI가 코드를 생성하고 사람은 이를 검토합니다.

이러한 상황에서는 운영 인시던트에 대응하는 엔지니어가 해당 인시던트를 유발한 코드를 작성했을 확률은 통계적으로 낮습니다. “시스템을 이해한다”는 말의 의미가 바뀐 것입니다.

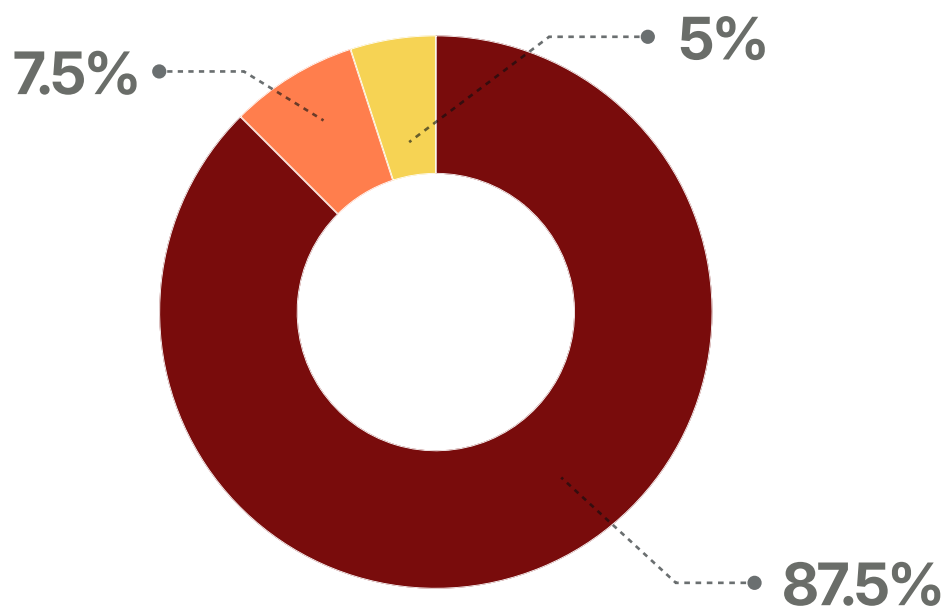
이제는 소스 코드를 읽는 것이 시스템을 이해하는 주된 방법이 아닙니다. 너무 많은 코드가 너무 빠르게, 그리고 너무 다양한 프롬프트에 의해 생성되고 있기 때문입니다. 런타임 증거(코드가 실제로 수행한 작업, 입력값, 실행 순서, 다운스트림 영향)가 더 신뢰할 수 있는 신호가 되었습니다. 오픈버빌리티 플랫폼을 “실제로 일어나고 있는 일”에 대한 기록 시스템으로 간주하는 엔지니어링 조직은 코드 검토를 이해 계층으로 의존하는 조직보다 훨씬 더 원활하게 확장할 수 있습니다.

바이브 코딩이 운영 정책 기준을 통과했습니다.

바이브 코딩을 비운영 작업에만 사용하도록 제한한 조직은 5%에 불과하며, 이를 전면적으로 금지하는 조직은 전혀 없습니다. 나머지 95%의 조직은 공식적(87.5%) 또는 비공식적(7.5%)으로 운영 서비스에 이를 허용하고 있습니다.

바이브 코딩에 대한 조직 정책

N=200



95% 운영 환경에서 바이브 코딩 허용

운영 환경에서 공식적으로 허용

운영 환경에서 비공식적으로 허용

비운영 환경에서만 허용

시사점

바이브 코딩은 이제 부수적인 관행이 아니며, 개인의 생산성 향상을 위한 수단이나 샌드박스 한정용도 아닙니다. 설문조사에 참여한 대부분의 조직에서, 바이브 코딩은 이미 소프트웨어 개발 수명주기(SDLC)에 포함되어 있습니다. AI로 생성된 코드는 시니어 엔지니어가 작성한 코드와 동일한 SLA와 장애 시간을 적용 받으며, 동일한 운영 환경, 동일한 수익 창출 서비스, 동일한 고객 대면 엔드포인트에 적용되고 있습니다.

바이브 코딩이 운영 정책에 포함되면, 누가 작성했는지에 관계없이 관련된 모든 거버넌스 체계가 동일하게 적용되어야 합니다.

코드 검토, 변경 관리, 배포 안전장치, 인시던트 대응 및 감사 로깅 데이터에 따르면, 대부분의 조직은 관련 통제 기능을 일관되게 강화하지 않은 채 이러한 관행을 허용하고 있습니다.

사람이 작성한 코드에 이미 적용하고 있는 릴리스 상태, 변경 추적 및 SLO 모니터링 원칙을 AI 생성 코드에도 동일하게 확장하여 이러한 간극을 조기에 해소하는 리더는 바이브 코딩 서비스를 별도의 카테고리로 취급하는 리더보다 운영 상 충격을 더 원만하게 흡수할 수 있을 것입니다.

AI가 생성한 코드가 사람이 작성한 코드보다 더 높은 평가를 받습니다.

61%의 리더는 AI가 생성한 코드가 사람이 작성한 코드보다 “품질이 약간 더 높다”고 답했고, 또 다른 33%는 “훨씬 더 높다”고 답했습니다.

AI가 생성한 코드와 사람이 작성한 코드의 인지되는 품질



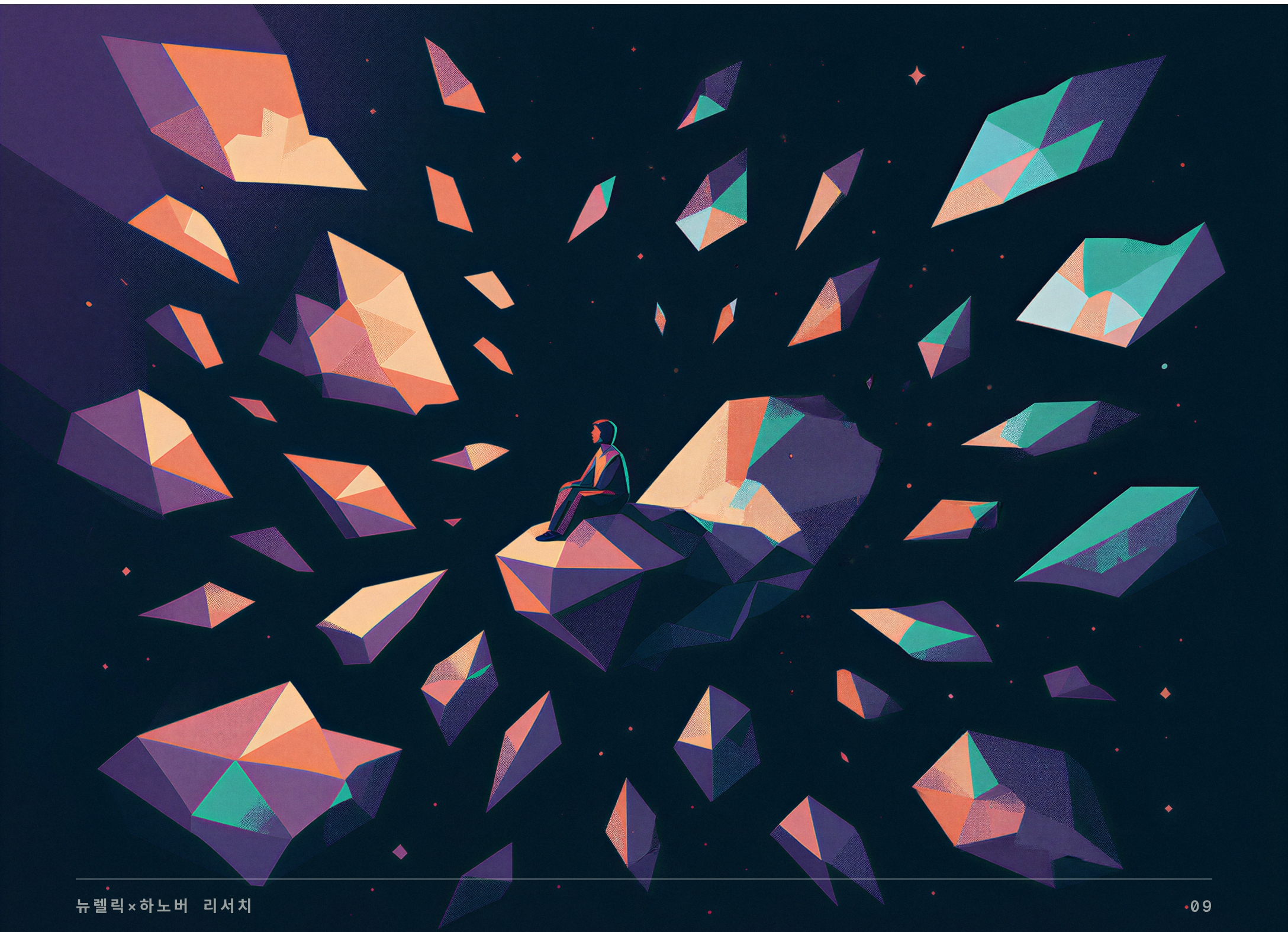
시사점

AI로 지원되는 코딩의 가장 큰 이점은 대부분의 엔지니어링 조직이 품질을 측정하는 지점인 코드 검토 단계에서 기대에 부응한다는 것입니다. 제출 시 패턴이 더 깔끔하고, 스타일이 일관되게 유지되며, 눈에 띄는 버그가 적습니다. 그러한 인식은 실재하며 광범위하게 퍼져 있습니다. 조직이 바이트 코딩 도입을 계속해서 가속화하는 이유이기도 합니다.

문제는 이러한 품질 평가가 인시던트 대응 시점이 아니라 코드 검토 시점에 내려진 판단이라는 점입니다. 읽기 좋은 코드가 반드시 잘 작동하는 코드는 아니며, 이 둘을 추적하는 메트릭은 전혀 다릅니다. 설문조사의 데이터는 동일한 코드가 실제 트래픽 환경에서 실제 종속성에 기반해 실제 고객을 대상으로 실행될 때 어떤 일이 발생하는지를 보여줍니다.

런타임 현실

AI로 코드 생성 속도는
빨라졌지만, 시스템의 신뢰성을
유지하는 데 드는 비용과 인력은
더 증가했습니다.



운영 부담이 동반 상승했습니다.

지난 12개월 동안 86%의 조직에서 시니어 개발자의 재작업 시간이 증가했습니다. 같은 기간 동안 4분의 3이 넘는 조직에서 마이크로서비스 확산, 운영 인시던트 및 기술 부채가 모두 증가했습니다.

상당한 재작업이 필요한 AI 생성 코드의 비율

N=200

환경 내 마이크로서비스/함수 볼륨



시니어 개발자가 코드 수정에 소비한 시간



운영 인시던트 수



기술 부채



감소

증가

시사점

위의 네 가지 메트릭은 AI 지원 개발에서 가장 흔하게 발생하는 비용이며, 개발 속도와 인지 품질이 향상된 12개월 동안 일제히 변동했습니다. 속도를 향상하는 비용은 낮아졌지만, 신뢰성은 다른 문제입니다. 속도를 향상한 조직들은 다양한 운영 부담을 감수하고 있습니다.

이는 AI 지원 시대의 핵심 딜레마이며, 데이터로 증명된 사실입니다. 속도를 늦추는 것이 올바른 전략적 대응이 아닙니다. 속도 향상은 실질적인 문제이며 수익에 그대로 반영되기 때문입니다.

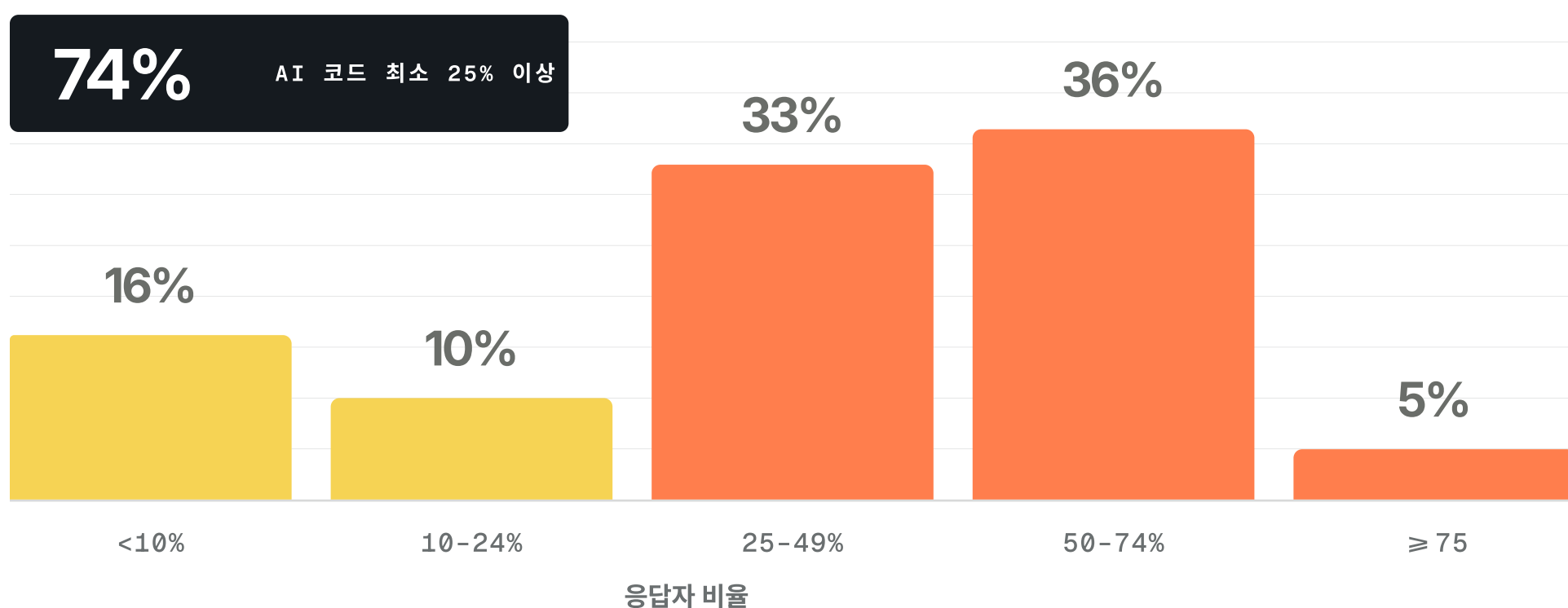
기능 배포 속도가 빨라진 덕분에 63%의 조직은 수익이 소폭 증가한 것으로 나타났습니다. 올바른 대응은 AI 도입으로 발생하는 운영 오버헤드를 예산에 명시적으로 반영하는 것입니다. 달리 말하면, 개발 속도와 배포 횟수만 따지는 리소스 계획에 장애 발생 빈도, 복구 시간, 기술 부채 해결 비용, 그리고 시니어 개발자가 코드를 다시 고치는 시간까지 전부 계산에 넣어야 한다는 뜻입니다. 이처럼 AI 도입의 득과 실을 경영진에게 투명하게 보고하는 조직이야말로, 어디서 속도를 더 내야 하고 어디서 잠시 멈춰 시스템을 강화해야 할지 훨씬 더 현명한 결정을 내릴 수 있습니다.

AI가 생성한 코드로 인한 재작업 증가

41%의 리더는 불충분한 컨텍스트, 불완전한 데이터 또는 잘못된 가정으로 인해 AI가 생성한 코드의 절반 이상에 상당한 재작업이 필요하다고 답했습니다. 74%는 최소 4분의 1이 그렇다고 답했습니다.

상당한 재작업이 필요한 AI 생성 코드의 비율

N=200



시사점

바이브 코딩의 사후 수습 비용은 상당합니다. AI가 코드의 대부분을 생성하는 조직에서조차, 그렇게 생성된 코드의 상당 부분이 다시 수정을 거칩니다. 단순한 다듬기 작업이 아니라 누락된 컨텍스트, 환각 인터페이스 또는 데이터에 대한 잘못된 가정 때문입니다. 이러한 재작업은 더 높은 가치를 창출하는 문제에 집중해야 할 시니어 개발자들이 수행하는 경우가 많습니다.

AI 코딩 툴이 런타임에 시스템이 실제로 어떻게 동작하는지 확인하지 않은 채코드를 생성한다는 데 근본적인 원인이 있습니다.

AI는 소스는 읽을 수 있지만, 트레이스는 읽을 수 없습니다. 운영 텔레메트리, 실제 쿼리 패턴 및 실제 의존성 그래프를 개발 루프로 다시 연결하여 이러한 간극을 해소하기 시작한 조직들에서 재작업률이 가장 낮게 나타납니다. 런타임 데이터와 코드 생성 간의 루프를 완성하는 것은 향후 2년 동안 가장 중요한 옴저버빌리티 관행이 될 가능성이 높으며, 운영 부담을 가장 직접적으로 줄이는 방법이기도 합니다.

조직들은 장애 모드의 영향을 체감하고 있습니다.

지난 6개월 동안 AI가 생성한 코드와 관련된 문제가 없었다고 보고한 조직은 19%에 불과합니다. 나머지 82%는 평균 2개 이상의 운영 문제를 겪었습니다.



시사하는 사항

6개월 동안 4가지의 뚜렷한 장애 유형(통합 장애 31%, 규정 준수/거버넌스 문제 30%, 데이터 무결성 문제 29%, 새로운 보안 취약점 도입 28%)은 각각 10개 조직 중 약 3개 조직에 영향을 미쳤습니다. 과거 사례를 보면 사람이 작성하고 동료 검토를 거친 코드는 기본 장애율을 유지해 왔지만, AI가 생성한 코드는 이보다 약 1.7배 더 많은 치명적인 런타임 오류를 발생시키는 것으로 나타났습니다. AI가 생성한 코드가 운영 환경에서 장애를 일으키는 방식은 결코 한 가지 정형화된 형태로 나타나지 않습니다. 여러 미세한 결함들이 동시 다발적으로 장애를 유발합니다. 이러한 문제는 널리 퍼져 있어서, 최근 반년 사이에 대부분의 조직이 최소 한 번쯤은 치명적인 장애를 수습한 경험이 있을 정도입니다.

명명된 4가지 장애 유형은 저마다 뚜렷한 특징을 가지고 있습니다.

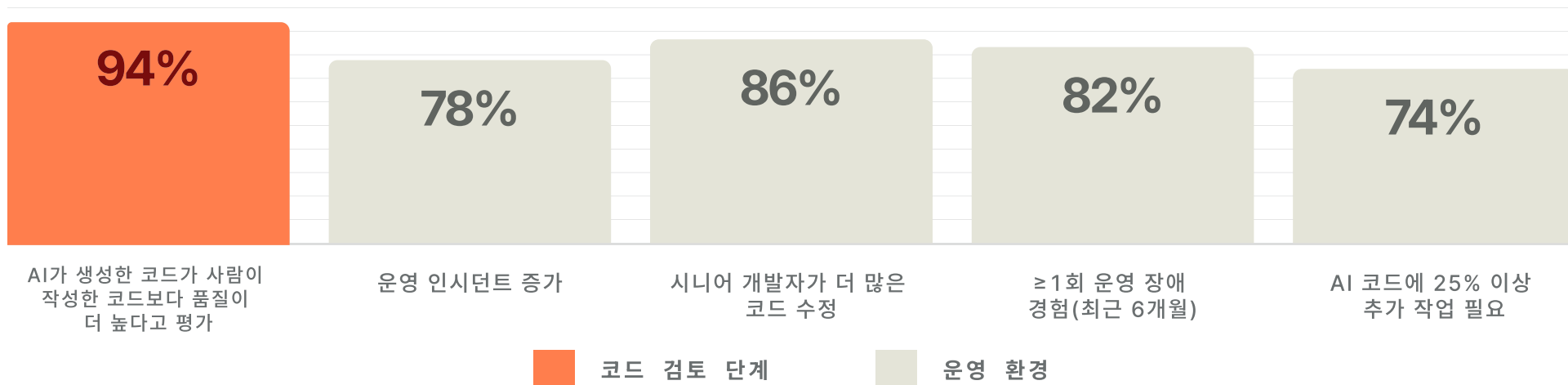
통합 장애는 스키마 드리프트, 계약 위반 및 서비스 간 오류율 증가로 나타납니다. 규정 준수 문제는 감사 로그 및 정책 위반으로 표출됩니다. 데이터 무결성 문제는 운영 텔레메트리에 서 중복, 드리프트, 예상치 못한 null 비율과 같은 이상 징후로 나타납니다. 보안 취약점은 트레이스 데이터, 인증 패턴 및 행동 이상 징후에서 드러납니다. 각 장애 유형을 옴저버빌리티 플랫폼의 특정 감지 패턴에 매핑한 조직은 사용자 불만이나 정기 감사에만 의존하는 조직보다 “AI가 생성한 실수가 운영에 도달”과 “실제 피해를 입히기 전에 발견” 사이의 간극을 훨씬 더 빠르게 좁힐 수 있습니다. 이것이 바로 통합된 텔레메트리 표면이 필요한 실질적인 이유이며, 뉴렐릭 및 이와 유사한 플랫폼에서 AI 코드 워크로드가 증가하는 이유 중 하나입니다.

AI로 지원되는 시대에는 모순이 존재합니다.

검토 단계에서 AI가 생성한 코드가 사람이 작성한 코드보다 품질이 더 높다고 평가한 조직들(94%)은 코드가 배포된 후 더 많은 인시던트(78%), 더 많은 재작업(74%가 코드의 25% 이상 재작업 필요), 시니어 개발자의 더 많은 긴급 대응(86%), 더 많은 운영 장애(지난 6개월 동안 82%)가 발생했다고 말했습니다.

검토에서는 좋아 보이지만 실제 운영에서는 성능이 낮은 AI 코드

N=200



시사하는 사항

이는 이 연구 조사에서 가장 중요하게 나타난 패턴입니다. 코드 검토를 할 때, AI 코드는 사람이 작성한 것보다 더 깔끔하고 일관되며 구조가 잘 짜인 것이 눈에 보입니다. 운영 환경에서는 이러한 코드가 더 자주, 더 다양한 방식으로 장애를 일으키며, 이를 수정하기 위해 시니어 개발자의 개입이 더 많이 필요해집니다. 두 환경의 상황은 모두 사실이지만 측정하는 대상이 서로 다릅니다. 검토 단계에서는 가독성과 표면적인 품질을 측정합니다. 운영 단계에서는 실제 부하, 실제 종속성, 실제 엣지 사례에서의 동작을 측정합니다. 이는 AI 코딩 툴이 코드를 생성할 때 전혀 파악할 수 없는 조건입니다.

이러한 모순에 대한 해답은 하나뿐입니다. AI가 생성한 코드가 검토에서는 우수해 보이지만 운영 환경에서는 성능이 떨어지는

이유는 AI가 사람인 검토자가 볼 수 있는 아티팩트(소스)를 기반으로 작업하며, 운영 환경에서만 생성할 수 있는 아티팩트(트레이스)는 파악하지 못하기 때문입니다.

이러한 간극을 좁히는 것이 향후 몇 년간 옴저버빌리티 계층의 전략적 과제입니다. AI로 지원되는 시대에서 성공하는 조직은 운영 텔레메트리를 단순한 사후 디버깅 도구가 아니라 개발 루프의 최우선 입력 요소, 즉 다음 프롬프트, 다음 코드 검토, 다음 배포 결정을 개선하는 피드백 신호로 취급하는 조직일 것입니다. 옴저버빌리티 분야가 존재하는 이유도 바로 이 때문이며, 이 연구 조사의 데이터는 이것이 왜 중요한지를 명확하게 보여줍니다.

AI 코딩에 대한 맹신은 운영 위기를 초래할 수 있습니다.

62%의 기술 리더들은 팀이 AI가 생성한 코드를 신뢰하여 한 줄씩 수동으로
검증하지 않고 그대로 배포한다고 답했습니다.

줄 단위 검증 없이 AI 코드 신뢰

N=200

62%

자주 또는 항상

시사하는 사항

바이브 코딩 시대의 핵심적인 모순은 개발 수명주기 초기 단계에 AI를 과도하게 신뢰해 버린 데서 비롯될 수 있습니다. 이러한 광범위한 맹신은 초기 검토 단계에서 코드 품질이 왜 실제보다 인위적으로 높게 평가되는지, 그러면서도 실제 운영 환경에 배포되자마자 왜 곧바로 무너져 내리는지를 잘 설명해 줍니다.

대규모 언어 모델(LLM)은 격리된 최적의 조건에서 완벽하게 작동하는 코드를 생성하는 데 탁월합니다. 그러나 엣지 사례, 동시성 사각지대, 오래된 API의 지원 중단, 복잡한 상태 변화를 고려하지 못하는 경우가 빈번합니다.

검증되지 않은 상태로 배포될 경우, 이러한 구조적 누락은 실제 사용자 트래픽이 발생하기 전까지 겉으로 드러나지 않은 채 숨어 있습니다.

검사 없이 AI가 생성한 코드를 운영 환경에 적용하면 국지적인 운영 인시던트가 대폭 증가할 수도 있습니다. 사이트 안정성 엔지니어링(SRE) 팀과 데브옵스(DevOps) 팀은 점점 더 "AI 코드 청소부" 역할을 강요받고 있으며, 검증되지 않은 기계 출력물을 분류하고 리팩터링하는 데에만 실제 근무시간의 최대 3분의 1을 허비하고 있습니다. 이로 인해 본연의 업무인 로드맵 가속화에 차질을 빚습니다.

미래의 프레임워크

기술 리더들은 AI가 생성한 코드를 관리할 때 오피저버빌리티가 절대적으로 필요하다는 데 만장일치로 동의합니다.

이제 옹저버빌리티는 필수 요소가 되었습니다.

기술 리더의 39%는 AI가 생성한 코드로 작업할 때 옹저버빌리티를 “대단히 중요”하다고 평가했으며, 이는 단일 응답 중 가장 많이 선택된 답변입니다. 또 다른 57%는 “매우 중요”하다고 답했습니다. “약간 중요”하거나 “중요하지 않다”고 평가한 응답자는 단 한 명도 없었습니다.

AI가 생성한 코드로 작업할 때 옹저버빌리티의 중요성

N=200



시사하는 사항

AI가 생성한 코드에 옹저버빌리티가 필수적인지에 대한 논쟁은 사실상 끝났습니다. 하위 2개 항목을 선택한 응답자가 단 한 명도 없다는 것은, 이제 옹저버빌리티에 '투자를 해야 할까 말까'를 고민할 단계는 지났다는 뜻입니다. 대신 '어떤 플랫폼으로 표준화할 것인가'가 진짜 문제입니다.

중요도 점수가 이토록 상위권에만 몰려 있다는 건, 이제 전략의 초점이 새로 '구매하는 것'에서 하나로 '통합하는 것'으로 옮겨 가야 함을 보여줍니다. 이번 조사에 참여한 대부분의 기업은 이미 옹저버빌리티 시스템을 갖추고 있으며, 심지어 두 개 이상의 툴을 사용하는 곳도 많습니다.

이처럼 거의 완벽한 합의가 이루어졌을 때 가장 큰 지렛대 효과를 얻는 리더는, 옹저버빌리티와 AIOps 전체를 아우르는 단일 플랫폼으로 표준화하는 리더입니다. 그렇지 않으면 당직 엔지니어가 야간 장애 발생 시 어떤 툴도 완전히 신뢰할 수 없는 파편화된 환경이 만들어질 뿐입니다.

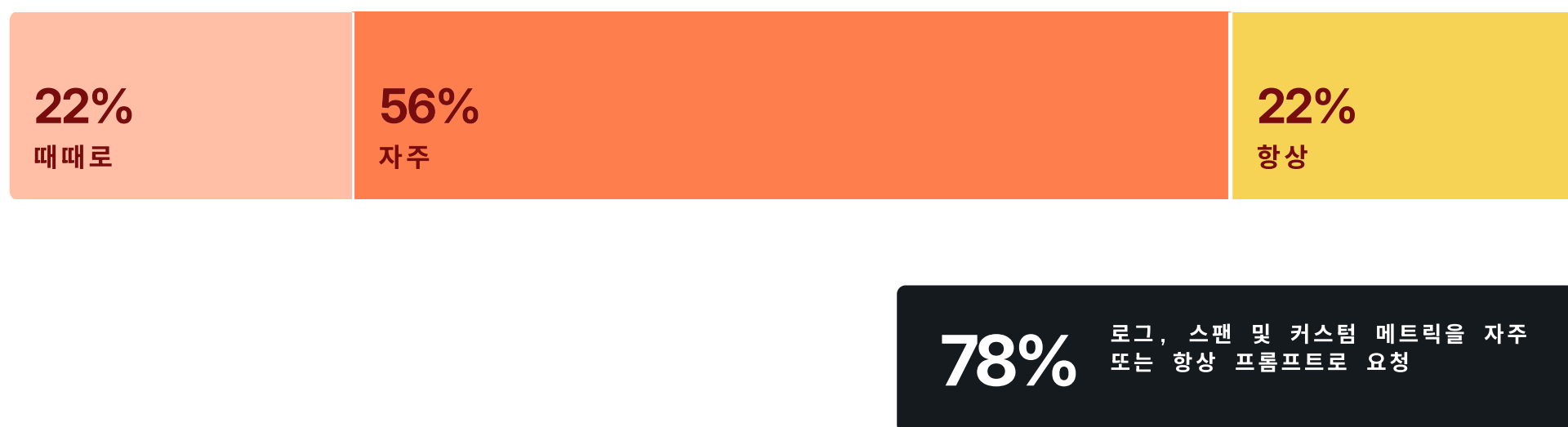
4가지 핵심 신호를 하나의 데이터 모델과 하나의 쿼리 화면으로 통합한 뉴렐릭 같은 플랫폼들이야말로, 바로 지금 같은 '통합의 타이밍'에 가장 강력한 대안이 될 수 있습니다.

텔레메트리가 AI의 프롬프트에 입력되고 있습니다.

리더의 56%는 팀에서 코드를 생성할 때 특정 텔레메트리(로그, 트레이스, 커스텀 메트릭)를 포함하도록 AI에 “자주” 프롬프트를 입력한다고 응답했습니다. 또 다른 22%는 “항상” 그렇게 한다고 응답했습니다. 드물게 하거나 전혀 하지 않는다고 응답한 경우는 없었습니다.

팀이 생성된 코드에 텔레메트리를 포함하도록 AI에 프롬프트를 입력하는 빈도

N = 200



시사하는 사항

이제 엔지니어들은 코드를 작성하고 나서 뒤늦게 옵저버빌리티를 어떻게 할지 고민하지 않습니다. AI에게 명령을 내리는 '프롬프트' 단계에서부터 옵저버빌리티를 포함해줄 것을 요구합니다. 텔레메트리는 AI가 정의하는 “완료”의 일부가 되고 있는 것입니다.

어떤 로그를 남기고, 어떤 트레이스를 수행하며, 어떤 알림을 전송할지에 대한 결정이 SRE 백로그에서 벗어나 개발자의 프롬프트로 이동하고 있습니다. 이에 따른 두 가지 시사점은 다음과 같습니다. 첫째, 조직의 텔레메트리 품질은 이제 IT 엔지니어가 사용하는 프롬프트와 템플릿에 크게 좌우됩니다.

둘째, 특정 방식을 지향하고 스키마를 인식하는 규칙 (OpenTelemetry 시맨틱 규칙, 구조화된 로깅, 일관된 스펀 속성)을 표준화하는 조직은 각 개발자가 자신만의 형식을 프롬프트에 입력하도록 허용하는 조직보다 훨씬 더 명확한 신호 대 노이즈 비율을 얻게 됩니다. Cursor, Copilot, Claude Code 및 ChatGPT에서 표준을 구축하는 조직이 사실상 이 카테고리의 다음 단계를 위한 옵저버빌리티 표준을 구축하게 됩니다.

결론

세 가지 조사 결과는 엔지니어링 조직의 2026년 남은 기간에 대한 방향을 제시합니다.

개발 속도는 현실이며 그 대가는 런타임에 치러지고 있습니다.

더 빠른 릴리스, 더 잦은 배포, 향상된 검토 단계의 품질은 모두 약속한 대로 성과를 내주고 있습니다. 그에 따른 비용은 인시던트, 재작업, 무분별한 확산, 시니어 개발자의 시간 소모로 나타나고 있습니다. 이러한 득실의 양면을 모두 고려하여 계획을 수립하는 리더는 어디에 추진력을 더하고 어디를 보강해야 할지 더 나은 결정을 내릴 수 있습니다.

우려와 통제 간의 간극은 이 연구 조사에서 도출된 가장 실용적인 결과입니다.

리더들은 AI가 생성한 코드의 위험성에 대해 당연히 우려하고 있습니다. 그러나 실제로 배포된 통제 수단은 아직 이러한 우려 수준에 미치지 못합니다. 보안 스캔, 배포 가드레일, 거버넌스 정책, 런타임 오피저버빌리티 전반에서 다층적인 안전 장치를 마련하여 이러한 간극을 가장 빠르게 해소하는 조직이 허용할 수 없는 위험에 직면하지 않고 엔터프라이즈 SLA 수준에서 AI 비중이 높은 코드베이스를 운영할 수 있을 것입니다.

검토 단계의 품질과 실제 운영 환경 간의 괴리야말로 오피저버빌리티가 진가를 발휘하는 부분입니다.

AI 코딩 툴은 소스를 보고, 트레이스는 보지 못합니다. 이러한 루프가 닫힐 때까지, AI가 생성한 코드는 검토 단계에서 계속 좋은 평가를 받으면서도 운영 환경에서는 정작 팀을 당혹스럽게 만들 것입니다. 오피저버빌리티 플랫폼은 AI가 생성한 코드가 실제로 어떻게 돌아가고 있는지 정확히 기록하는 단 하나의 기준이 되어주고, 그 데이터를 다시 개발 과정에 반영해 줍니다. 이러한 플랫폼은 AI 협업 시대에 개발 팀이 작업하는 방식의 중심에 자리하게 될 것입니다. 뉴렐릭은 정확히 이러한 선순환 구조를 중심으로 만들어진 플랫폼이며, 이것이 바로 뉴렐릭이라는 계층에 최우선으로 투자해야 하는 이유입니다.

뉴렐릭이 이번 연구 결과를 발표한 이유는 이 보고서에 담긴 해답들이 지금 모든 개발 조직이 치열하게 고민하고 있는 숙제이기 때문입니다. 데이터는 AI가 코드를 생성하는 시대가 이미 시작되었고, 이로 인해 실제 운영 과정에서 겪는 문제들이 현실로 나타나고 있음을 보여줍니다. 그리고 이제 오피저버빌리티는 이러한 부작용과 문제들을 해결하고 관리하는 핵심 계층이 되었습니다.

하노버 소개

하노버 리서치(Hanover Research)는 정책제 구독 모델로 맞춤형 정량 및 정성적 인사이트를 제공하는 글로벌 시장 조사 및 분석 전문 기업입니다. 2003년에 설립되었으며, 현재 200명 이상의 데이터 분석가, 통계학자, 설문 조사 방법론 전문가들을 보유하고 있습니다. 기업, 의료, 교육 부문 전반의 전략적·운영적·재무적 의사결정을 지원하기 위해 1차 시장 조사, 예측 데이터 모델링, 그리고 세그먼트 분석을 전문적으로 수행하고 있습니다.

뉴렐릭 소개

뉴렐릭 인텔리전트 오피버빌리티 플랫폼은 소프트웨어 스택 전반의 텔레메트리 데이터를 수집, 표준화 및 분석합니다. 메트릭, 이벤트, 로그, 트레이스를 단일 텔레메트리 데이터 플랫폼으로 통합함으로써, 시스템 엔지니어링을 사후 장애 대응에서 선제적 성능 관리로 전환합니다. 아디다스 런타스틱(Adidas Runtastic), 미국 적십자사(American Red Cross), 도미노피자(Domino's), 고투 그룹(GoTo Group), 라이언에어(Ryanair), 탑골프(Topgolf), 윌리엄 힐(William Hill) 등의 글로벌 기업들이 인프라 건전성을 모니터링하고 애플리케이션 성능을 최적화하기 위해 뉴렐릭을 도입한 이유도 이 때문입니다.

보고서 소개

이 보고서의 데이터는 뉴렐릭의 의뢰를 받아 하노버 리서치가 수행한 2026년 온라인 설문조사 결과를 바탕으로 합니다. 조사 대상은 소프트웨어 개발에 생성형 AI를 활용하는 기업의 IT 및 엔지니어링 부문 직원이며, 미국에 기반을 둔 기술 의사결정권자 200명이 포함됩니다.

뉴렐릭 플랫폼에 대해 자세히 보기