

# 소프트웨어 공학의 미래

이동재

2025년 1월 10일

## Abstract

지난 4년간 언어 모델의 성능은 비약적으로 발전했고, 그 속도는 점차 빨라지고 있다. 최근 OpenAI의 o3 모델은 CodeForces에서 전 세계 개발자 중 175등에 달하는 점수를 얻어 큰 화제가 되었다. 코딩 능력만 보면 언어 모델은 이미 인간을 앞질렀다고 봐도 무방하다. 그렇다면, 앞으로 소프트웨어 공학에서 해결해야 할 과제는 무엇인가? 본 글에서는 현재 언어 모델이 지닌 한계점을 분석하고, 자동화된 소프트웨어 개발과 유지보수를 위해서는 어떤 기술이 필요할지에 대해 논의하고자 한다.

최근 인공지능 분야에서는 모델이 어떤 일을 아직 잘하지 못할 때, 단순히 ‘아직’일 뿐이고, 근시일 내에 해결될 문제라는 견해가 널리 퍼져 있다. 결국에는 스케일링을 통해서, 데이터, 훈련량, 추론 시간을 키우면 해결될 문제라는 생각이 널리 퍼져 있다. 이러한 접근 방식을 통해 인공지능은 현재 대부분의 지적 노동 분야에서 인간의 평균 수준을 뛰어넘었으며, 최근에는 박사 수준의 전문 영역까지도 위협하고 있다.

소프트웨어 공학도 예외가 아니다. 많은 개발자들이 ChatGPT, Copilot과 같은 AI 도구에 의존하고 있으며 이를 통해 개발 생산성을 크게 향상시켰다. 그렇다면 소프트웨어 공학에서 남아 있는 과제는 무엇일까? 정말로 모델이 모든 문제를 해결해줄 수 있을까?

그렇지 않다. 언어 모델이 지금까지 다뤄왔던 환경과 실제 소프트웨어 개발 환경의 가장 큰 차이점은 코드의 규모다. 현재의 언어 모델은 간단한 함수나 클래스, 혹은 단일 파일 수준의 작업을 주로 수행한다. 하지만 실제 소프트웨어 개발에서는 하나의 기능을 구현하기 위해 여러 파일에 걸친 작업이 필요하며, 기존 코드와의 호환성과 재사용성을 고려해야 한다. 현재의 언어 모델은 대규모 코드베이스를 다루거나 여러 파일을 동시에 처리하는 데 한계를 보인다.

이 문제를 해결하려면 모델이 잘 작동할 수 있는 기반 인프라를 구축하는 것이 필수적이다. 기존의 소프트웨어 엔지니어링과 프로그래밍 언어 기술은 인간 개발자를 위한 것이었다. 이제는 모델을 위한 소프트웨어 엔지니어링과 모델을 위한 프로그래밍 언어 기술이 필요하다. 인간과 언어 모델은 동작에 분명한 차이가 있으며, 이 점을 해결해줄 수 있는 기술을 우리가 개발해야 한다.