

ISSUE REPORT | 2024.08.19 IS-179

생성형 AI에 의한 소프트웨어 개발자 업무 영향 분석

Analyzing the impact of generative AI on Software developer work

진회승, 윤보성, 신승윤

이 보고서는 「과학기술정보통신부 정보통신진흥기금」에서 지원받아 제작한 것으로
과학기술정보통신부의 공식의견과 다를 수 있습니다.
이 보고서는 "진회승·윤보성·신승윤. (2023). 생성형 AI에 대응한 SW인재양성 정책방향 연구, SPRI
연구보고서"의 내용 중 일부를 발췌하여 작성되었습니다.
이 보고서의 내용은 연구진의 개인 견해이며, 본 보고서와 관련한 의문 사항 또는 수정·보완할
필요가 있는 경우에는 아래 연락처로 연락해 주시기 바랍니다.

소프트웨어정책연구소 SW기반정책·인재연구실
진회승 책임연구원 hschin@spri.kr

CONTENT

I. 생성형 AI 시대 도래 P.1

1. 생성형 AI 활용 현황 및 전망
2. 생성형 AI로 인한 개발 관련 주요 이슈

II. 개발 프로세스별 소프트웨어 개발자 업무 변화 P.5

1. SW 개발 프로세스 및 개발도구 시장
2. SW 코딩 방법 및 자동화 변화상
3. SW 개발 프로세스별 생성형 AI 활용 및 영향 분석

III. 소프트웨어 개발자 직무 변화 P.15

IV. 소프트웨어 개발자 양성 정책 제언 P.20

[별첨] O*NET 개발자 관련 직무 정의 P.22

요약문

본고는 생성형 AI로 인한 소프트웨어 개발 프로세스별 개발 업무 변화를 살펴보고, 이것이 소프트웨어(Software, 이하 SW) 개발자 직무에 어떠한 영향을 주는지와 그 시사점에 대해 논하려고 한다.

생성형 AI란, 텍스트, 오디오, 이미지, 동영상 형태의 새로운 콘텐츠를 생성하도록 설계된 인공지능 모델이다. 생성형 AI라는 화두를 세상에 내놓은 OpenAI사의 ChatGPT는 지금까지 출시된 애플리케이션 중 가장 빠른 속도로 이용자 수가 늘어나고 있다. 이러한 경향은 생성형 AI의 활용이 업무 효율성을 향상할 것으로 기대되기 때문이며, 특히 개발환경의 변화로 인해 개발자의 역할 또한 변화할 것으로 전망된다.

SW 개발은 SW 요구분석, 설계, 구현 및 시험 등의 복잡한 프로세스로 구성되어 있다. 또한 고객사의 개발 및 운영 환경, 성능 및 보안 요구 수준, 유지보수성 등 다양한 요소를 고려해야 하는 정형화하기 어려운 업무이기 때문에 생성형 AI가 개발자 업무에 주는 영향은 예상만큼 크지 않았다.

SW개발 단계 중 SW 구현과 SW 시험 단계에는 생성형 AI 활용이 가능하다. 생성형 AI를 이용하면 SW 구현 단계에서는 코드 생성, 코드 자동완성, 주석 작성, 코드 번역, 리버스 엔지니어링 등이 가능하다. SW 시험은 시간이 많이 소요되고 반복적인 업무가 많아, 생성형 AI 활용에 가장 효과적인 단계이다. 반면에 고객과의 커뮤니케이션이 중요한 SW 요구분석 단계와 최신 기술을 포함한 복잡한 기술을 연계하여 수행해야 하는 SW 설계 단계는 생성형 AI의 활용이 쉽지 않다.

전반적으로 생성형 AI 개발도구의 활용은 개발자의 생산성 향상에 효과가 있으나, 개발 단계, 개발하는 제품이나 서비스의 종류, 개발자의 수준 등에 따라 생성형 AI의 활용 방법 및 효과의 차이가 큰 것으로 분석되었다. 개발도구로서 생성형 AI의 활용은 초급개발자보다 생성형 AI 환각에 의한 잘못된 정보를 빠르게 판단할 수 있는 중급개발자의 생산성을 더 높이는 것으로 확인되었다. 반면에 초급개발자는 생성형 AI를 프로그램 언어 학습에 유용하게 활용할 수 있으며, 초급 개발자 수준이 빠르게 높아질 것으로 예측된다.

SW개발자의 작업에서 생성형 AI의 영향을 받는 작업은 일부분이다. 개발자 직무를 수행하기 위해서는 데이터 분석, SW 및 DB 설계, 시스템 결정 및 성능 개선 등 개발 관련 기술 업무 이외에도, 동료와 업무 협의, 이슈 해결 등의 소프트스킬이 필요한 작업이 있어 개발자 업무와 개발 생산성에 대한 생성형 AI의 영향 정도를 정량적으로 파악하기는 어렵다.

생성형 AI 시대에 대비하기 위해 개발자 수준에 따른 세분화된 생성형 AI 활용 전략 마련, 직무별로 상이한 생성형 AI의 영향을 고려한 개발자 양성 규모 검토가 필요하다. 또한 생성형 AI를 활용하면서 직면할 수 있는 저작권 문제, 프라이버시 문제, 모델 오남용 문제에 대해 적절히 대응할 수 있도록 기본적인 AI 윤리 교육에 대한 강화가 시급하다.

Executive Summary

In this article, we'll take a look at how generative AI is changing the development process and discuss the implications for developer jobs.

Generative AI is an AI model designed to generate new content in the form of text, audio, image and video. The use of generative AI is growing at a rapid pace. This is because it is expected to improve work efficiency, and the role of developers is also expected to change due to changes in the development environment.

SW development consists of a complex process of SW requirements analysis, design, implementation, and testing. In addition, the impact of generative AI on developers' work is not as large as expected because it is a difficult task to formalize that requires consideration of various factors such as the customer's development environment, performance and security requirements, and maintainability.

Generative AI can be used for SW implementation and SW testing during the SW development phase. On the other hand, the SW requirement analysis phase, which requires communication with customers, and the SW design phase, which involves complex technologies including the latest technologies, are not easy to use generative AI.

Overall, the use of generative AI development tools is effective in improving developers' productivity, but there are large differences in how generative AI is used and its effectiveness depending on the development stage, the type of product or service being developed, and the level of developers.

Only a small part of a SW developer's work is affected by generative AI. In addition to technical tasks related to development, such as data analysis, SW and DB design, system performance improvement, developers also need soft skills, such as collaborating with colleagues, so it is difficult to quantify the extent of the impact of generative AI on developer work and development productivity.

In order to prepare for the era of generative AI, it is necessary to prepare a strategy for using generative AI that is categorized by developer level and review the scale of developer training considering the impact of generative AI by job function. It is also urgent to strengthen basic AI ethics education so that developers can respond appropriately to copyright, privacy, and model misuse issues that may arise while utilizing generative AI.

I. 생성형 AI 시대의 도래

1. 생성형 AI 활용 현황 및 전망

- 생성형 AI란, 텍스트, 오디오, 이미지 또는 동영상 형태의 새로운 콘텐츠를 생성하도록 설계된 인공지능 모델
 - * 가트너는 생성형 AI를 ‘콘텐츠 및 사물에 대한 데이터를 통해 학습하고, 그것을 사용하여 창조적이며 현실적인 완전히 새로운 아웃풋을 만들어 내는 기계학습 방법’으로 정의
 - 생성형 AI 시스템은 일반적으로 신경망의 일종인 딥러닝 기술을 사용하여 방대한 양의 데이터를 분석하고 패턴을 학습하고, 이를 기반으로 입력 데이터와 유사한 새로운 콘텐츠를 생성
- 국내 직장인의 경우 약 40%가 생성형 AI 사용경험이 있으며, 이 중 약 절반은 업무에 활용 중¹⁾
 - 생성형 AI 유료 사용 경험자 비율은 32.7%이며, 12.6%는 현재 유료 사용
 - 생성형 AI 이용 용도는 단순 호기심(64.3%)이 가장 높기는 하나, 업무에 활용한다는 응답도 절반가량(47.1%)을 차지
- 생성형 AI라는 화두를 세상에 내놓은 OpenAI사의 ChatGPT는 전무한 기록으로 사용자 확산 중이며, 생성형 AI 세계시장 규모는 2023년 448억 9,000만 달러에서 2030년에는 2,070억 달러 규모로 성장할 것으로 전망²⁾
 - * ChatGPT는 출시 5일 만에 이용자가 100만 명이 넘었고 40일 만에 1,000만 명 가입, 또 두 달 만에 1억 명 이상의 이용자 기록(100만 명 이용자 확보까지 Netflix는 3년 이상, Facebook은 10개월, Apple iPhone은 두 달 이상 소요)
 - 산업 전반의 워크플로 현대화 및 텍스트→이미지→비디오로의 온라인 커뮤니케이션 형태 진화 등은 생성형 AI의 수요를 증가시키며, 생성형 AI 애플리케이션

1) 김경은 (2024.4), “생성형 AI 이용현황 및 노동 대체 가능성에 대한 이용자 인식조사”, 소프트웨어중심사회 (2023년 12월 4~8일까지, 수도권 및 6개 광역시 거주 20~50대 직장인 1,038명을 대상으로 온라인 설문조사 수행)

2) Statista Market Insights (2023.8), 생성형 AI - 세계 기준(Generative AI - Worldwide). 2023년 9월 기준
<https://www.statista.com/outlook/tmo/artificial-intelligence/generative-ai/worldwide#market-size>

이선 시장은 확대될 것으로 전망

- AI·딥러닝 발전, 창의적 앱·콘텐츠 생성 증가, 클라우드 저장소 혁신으로 인한 쉬운 데이터 접근과 함께 고차원 비주얼과 상호작용, 더 현실성 있는 AI 기술을 활용한 메타버스의 도입 등으로 앞으로 생성형 AI가 시장을 견인할 것으로 전망

□ 생성형 AI에 대한 다양한 성장 동인은 앞으로 많은 사업 기회를 제공할 것임³⁾

- 생성형 AI는 비즈니스 모델을 혁신하고 통신, 의학 등 다양한 산업 분야의 문제를 해결할 수 있어, 여러 분야에서 인공지능 통합 시스템의 사용이 확대될 것
- 생성형 AI는 산업 전반에 걸친 워크플로의 현대화를 촉진하고, 업무 효율성을 지속적으로 향상할 것으로 예상

2. 생성형 AI로 인한 소프트웨어 개발 관련 주요 이슈

□ 생성형 AI의 확산으로 기존 업무가 영향을 받고, 특히 개발환경의 변화로 인해 소프트웨어 개발자의 역할 또한 변화할 것으로 전망

- 생성형 AI의 도입으로, 미국 노동력의 약 80%, 업무의 최소 10%가 영향을 받을 수 있고,⁴⁾ 생성형 AI 관련 직업 증가 중⁵⁾
- 이에 따라 소프트웨어 개발 업무, 특히 코드 생성도 생성형 AI의 영향을 많이 받을 것으로 예상되며, 이러한 개발환경 변화는 소프트웨어 개발 생산성 향상과 함께 소프트웨어 개발자의 역할을 변화시킬 것으로 예상됨

□ 생성형 AI는 다양한 업무기능의 생산성 향상에 영향을 미치나, 소프트웨어 엔지니어링

3) Allied Market Research(2023.7), Generative AI Market Size, Share, Competitive Landscape and Trend Analysis Report by Component (Software, Service), by Technology (Generative Adversarial Networks (GANs), Transformer, Variational Autoencoder (VAE), Diffusion Networks, Retrieval Augmented Generation), by End User (Media and Entertainment, BFSI, IT and Telecom, Healthcare, Automotive and Transportation, Others): Global Opportunity Analysis and Industry Forecast, 2023-2032

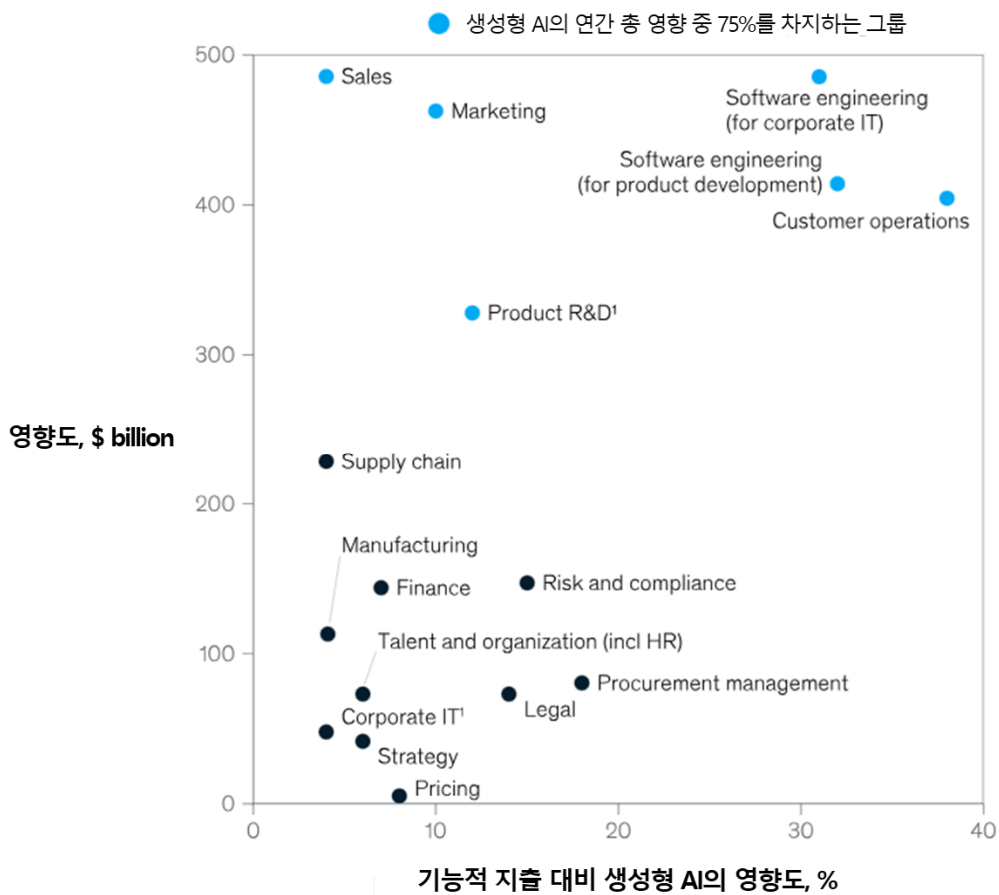
4) Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130.

5) Reuter (2023.6.23.), US-based generative AI job postings up 20% in May, Indeed data show

과 같은 개발 업무에서의 생산성 향상 변화는 작업 복잡성과 활용 대상의 수준 등에 따라 효과가 다르게 나타날 수 있음

- 생성형 AI가 생산성 향상에 영향을 많이 미치는 비즈니스의 기능에는 고객 운영, 마케팅 및 영업, 소프트웨어 엔지니어링, 연구개발이 있으며 이는 생성형 AI 사용을 통한 총 연간 가치의 약 75%를 차지

[그림 1] 생성형 AI의 비즈니스 업무기능별 영향력



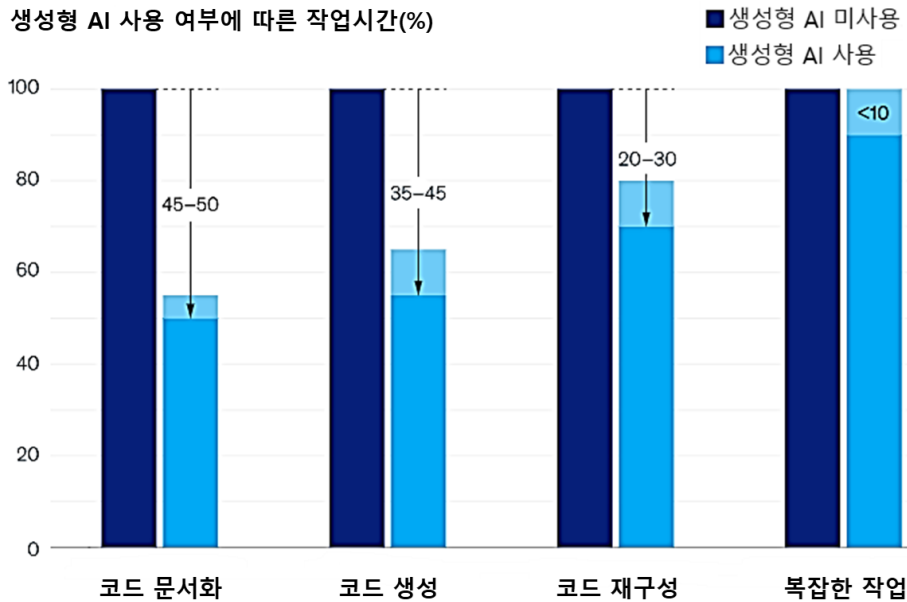
* 출처 : Mckinsey(2023.6), The economic potential of generative AI: The next productivity frontier

- 생성형 AI의 사용은 소프트웨어 개발 작업 속도를 향상시키지만, 그 효과는 소프트웨어 개발자 경험과 개발 난이도 등에 따라 차이가 있음⁶⁾
 - 코드 문서화에는 절반 가까이 속도 향상이 발생하였고, 새로운 코딩작업에는 약 40%, 기존 코드의 재구성·최적화 작업에는 약 25%의 시간을 절감

6) Mckinsey(2023.6), Unleashing developer productivity with generative AI

- 단, 개발 복잡도가 높다고 인지하는 작업은 10% 미만의 시간 절약 효과가 나타났으며 초급자는 오히려 작업 시간이 7%~10% 증가하는 것으로 측정됨

[그림 2] 생성형 AI의 사용 여부에 따른 개발업무별 작업 시간 절감 효과



* 출처 : Mckinsey(2023.6), Unleashing developer productivity with generative AI

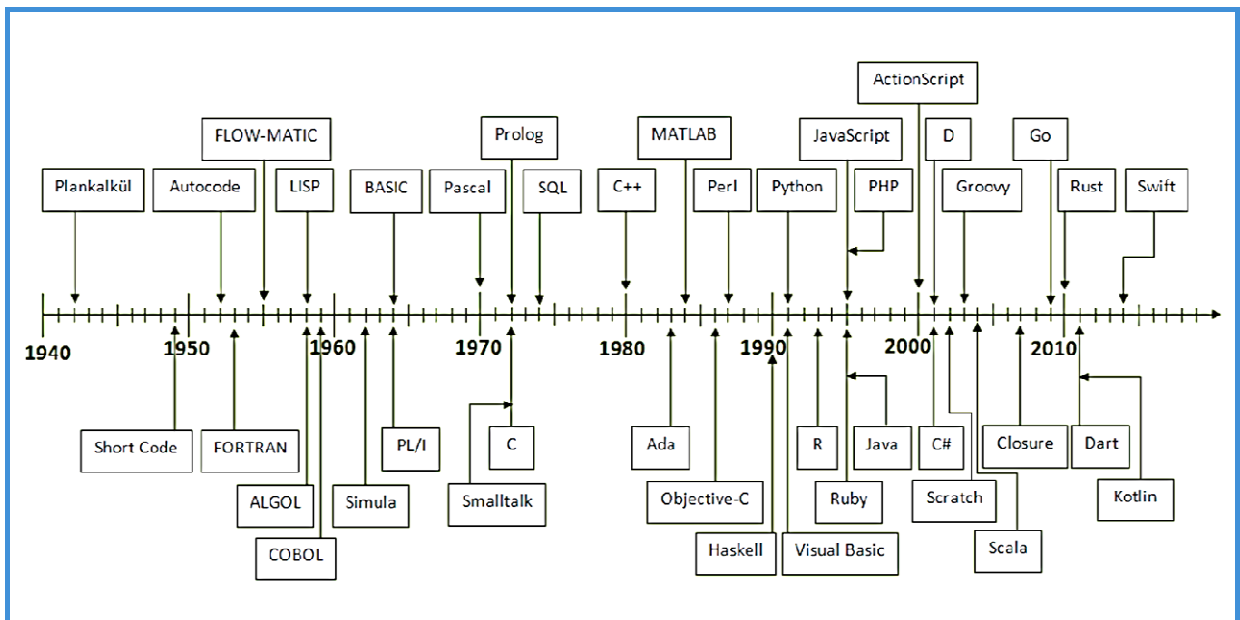
- 따라서 개발 숙련도가 낮거나 루틴한 업무를 수행하는 소프트웨어 개발자에게는 생성형 AI가 위협이 될 수 있으며, 현재로서는 일정 수준의 개발 숙련도를 갖추고 생성형 AI를 활용하는 것이 더 효과적일 수 있다고 추정할 수 있음
- 본고는 이상의 이슈를 고려하여 생성형 AI로 인한 개발 프로세스별 업무 변화를 살펴보고, 이것이 개발자 직무에 어떠한 영향을 주는지와 그 시사점에 대해 논함
 - 생성형 AI의 기술 발전과 활용은 개발 생산성 향상에 지속적 영향을 줄 것으로 예상되며, 그 과정에서 코딩 생성 관련 개발 환경이 변화하고 있어 이러한 변화가 개발 업무에 어떻게 영향을 주는지에 대해 검토함
 - 단, 생성형 AI가 작업에 미치는 영향이 업무 단계와 대상 등에 따라 다를 수 있어, 개발자 직무변화에 대해 보다 심층적으로 살펴보고 이에 대한 사회·경제적 영향, 개발자 대응 방안, 대학과 정부의 인재양성 방향에 대해 논의함

II. 개발 프로세스별 소프트웨어 개발자 업무 변화

1. 소프트웨어 개발 프로세스 및 개발도구 시장

- 소프트웨어(Software, 이하 SW) 개발은 SW 요구분석, 설계, 구현 및 시험 등의 복잡한 프로세스로 구성되어 있으며, 고객사의 SW개발·운영 환경, 성능·보안 요구 수준, 유지보수성 등 다양한 요소를 고려해야 하는 정형화하기 어려운 업무임
- SW 코딩과 관련이 많은 SW 구현 단계는 SW 상세 설계 내역을 이용하여 단위 SW 및 데이터베이스에 대한 코딩을 수행하고 소스 코드를 생성함
 - SW 소스코드를 작성하기 전에 알고리즘 개발이 선행되어야 하며, 알고리즘을 기반으로 프로그래밍 언어를 이용하여 코딩을 수행함
 - 프로그램 언어인 C/C++, 자바, Python 등은 HW, SW 아키텍처, SW 서비스에 따라 빠르게 변화하여 소프트웨어 개발자는 지속적 학습이 필요함
 - 다른 개발자들의 코드를 검토하여, 코드 오류, 최적화 등의 문제를 검토하는 중·고급 개발자들의 작업은 고도의 분석 및 판단 능력이 필요함

[그림 3] 프로그램 언어의 시대별 변천



* 출처 : Pramodbablad(2019), History Of Programming Languages

- SW 개발은 코딩을 포함하는 SW 구현 단계 이외에도 SW 요구분석, SW 설계, SW 시험 등의 여러 단계의 작업이 필요함
 - SW 요구분석은 개발하고자 하는 SW의 요구사항에 대한 분석, 검토 및 평가를 수행하고 우선순위를 설정하는 작업으로, 사용자 요구사항에 부합되는지 검토하고 확정하는 사용자와의 협의가 중요함
 - SW 구성요소를 식별하고 이들 간의 관계를 정의하는 구조설계와 알고리즘, 데이터 구조, 인터페이스를 설계하는 상세설계는 중상급 이상 개발자의 역할임
 - 시험 단계는 테스트 계획과 테스트 케이스를 작성하고, 단위·통합·성능·사용자 테스트 등의 일련의 테스트가 포함되어 있음
- SW 개발 지원 도구 시장은 MS, Atlassian, IBM, GitHub, Broadcom이 주도하며 2022년 기준 11.8% 증가하여 186억 달러에 이르며⁷⁾, 5천만 명이 사용하는 개발자 네트워크 플랫폼인 GitHub의 점유율이 높아지고 있음
- SW 구현 부분의 도구가 59%로 가장 큰 시장 점유율을 차지하고 있으며, 요구사항 분석, 설계 부분 도구가 25%의 시장을 점유함

[그림 4] SW 개발도구 시장 구분

Application Development			
Plan	Create	Verify	Other AD
<ul style="list-style-type: none"> ● Application Development PaaS ● Enterprise Agile Planning Tools <ul style="list-style-type: none"> - Frameworks ● Requirements Definition and Management ● Product Roadmapping Tools ● Conversational User Interfaces ● Design <ul style="list-style-type: none"> - Continuous Experience - Event-Driven Programming Models - Mediated API - API Economy - Database Design - Object-Oriented Analysis and Design - User Experience Tools ● Other Application Development Cycle Management ● Post-Scrum Methodologies 	<ul style="list-style-type: none"> ● Build <ul style="list-style-type: none"> - Event-Driven Web - Agile Development - Behavior-Driven Development - Hypothesis-Driven Development - Mobile AD Platforms ● Code <ul style="list-style-type: none"> - Java Platform AD Tools - Microsoft .NET Platform AD Tools - Language-Oriented Development Environments - Dynamic Web Application Tools - Intelligent Device Development Tools - Public Web APIs - Reactive Programming ● Configure <ul style="list-style-type: none"> - Application Portfolio Management 	<ul style="list-style-type: none"> ● Automated Testing ● Cloud Testing ● DevOps Application Testing ● Quality Engineering ● Service-Oriented Architecture Testing ● Test Data Management 	<ul style="list-style-type: none"> ● Conversational User Interfaces ● Gaming Tools

* 출처 : Gartner(2021), Market Share Analysis: Application Development Software, Worldwide, 2021

7) Laurie Wurster et al. (2023), Market Share: Application Development, Worldwide, 2022, Gartner (2023.6.16.)

2. SW 코딩 방법 및 자동화 변화상

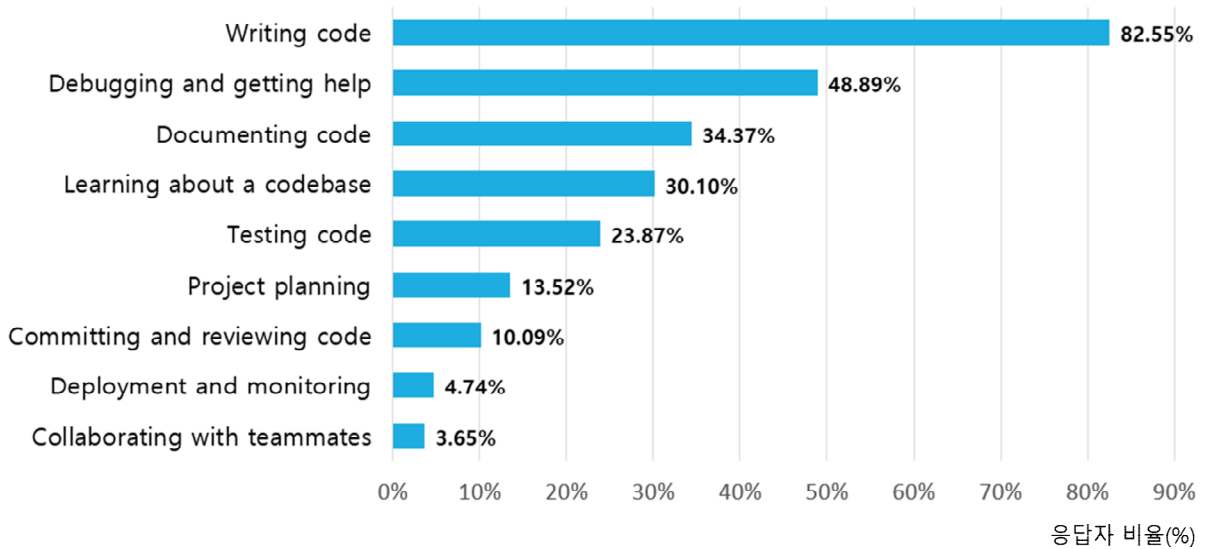
- 개발공수 감소, 프로그램 오류 감소, SW 유지보수 비용 절감 등을 위해 구조적 방법론, 객체지향 방법론, MDD(모델 주도 방법) 등이 개발되고, 코딩 자동화 도구가 활용됨
 - MDD를 활용하여 일정 부분 코딩을 자동화할 수 있으나, MDD 도구 부족, 초기 도입 비용, 학습 비용 등의 요인으로 MDD의 활성화가 어려움⁸⁾
 - 코딩작업을 최소화하거나 없앤 로우코드·노코드(LCNC) 도구를 이용하여 SW 개발전문가가 아닌 다른 도메인의 전문가가 SW 개발이 가능함
 - 가트너는 로우코드 시장은 2021년부터 연평균 성장률 19.2%로 2026년까지 445억 달러의 매출을 전망하며, 2026년까지 IT 부서에 속하지 않은 개발자의 80%는 LCNC를 사용하게 될 것이라고 예상함⁹⁾
- SW 개발 프로세스에 AI를 활용하려는 시도가 증가하고 있으며, 가트너는 2년에서 5년 이내에 AI를 소프트웨어 공학에 적용(AI-Augmented Software Engineering) 할 것이라고 예측함
 - Stackoverflow의 개발자 대상 설문에 의하면 전체 응답자의 77%가 AI 개발 도구에 대해 호의적이거나 매우 호의적이었으며, 전문개발자는 44%, 코딩을 배우는 사람들은 55%가 사용하고 있음¹⁰⁾
 - AI 개발도구는 생산성 향상(33%), 학습 속도 향상(25%), 코딩 효율성(25%) 등의 장점이 있으나, 정확성에 대해서는 소프트웨어 개발자 58%가 부정적인 의견을 표출함
 - AI를 SW 개발 단계별로 가장 많이 사용하는 부분은 코드를 작성하고, 디버깅하고, 프로그램 관련 문서를 작성하는 부분임

8) 진회승(2015), SW MDD 유용성 논의와 사례 분석

9) Varsha Mehta et al.(2022), Forecast Analysis: Low-Code Development Technologies, Worldwide, Gartner, 2022.10.27.

10) stackoverflow(2023), 2023 Developer Survey, 185국가 9만명 이상 개발자 설문, 전문 개발자 75%

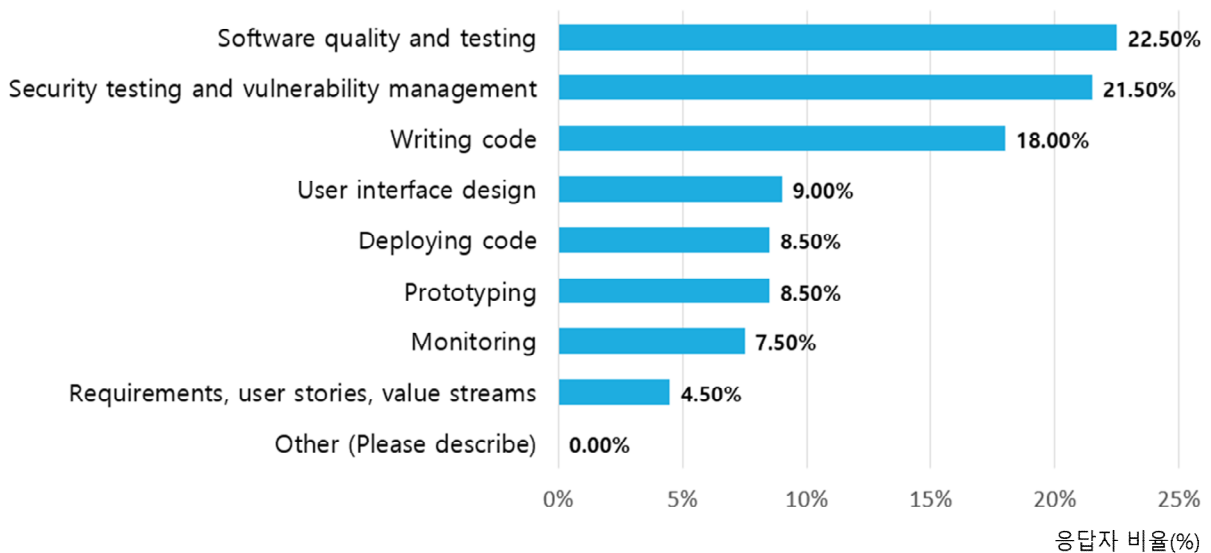
[그림 5] 소프트웨어 개발자들이 AI를 가장 많이 사용하는 개발 단계



* 출처 : Most popular uses of AI in the development workflow among developers worldwide as of 2023, StackOverflow, 2023

- 특히, 생성형 AI에 대해서는 소프트웨어 개발자 대상 설문조사에서 SW 품질관리 및 보안 테스트, 코드 작성 순으로 활용할 수 있다고 조사되었음

[그림 6] 생성형 AI의 활용이 많을 것으로 예측되는 소프트웨어 개발 작업



* 출처 : Melinda-Carol Ballou(2023), Generative AI Offers Greatest Expected Developer Benefits to Software Quality and Testing Followed by App Security Testing, IDC

3. 소프트웨어 개발 프로세스별 생성형 AI 활용 및 영향 분석

- (SW 요구분석) SW 요구분석은 고객과의 협의가 중요하고 소프트웨어 전문가와 도메인 전문가 간의 지식의 격차로 생성형 AI를 활용하기 쉽지 않은 작업이나, 생성형 AI의 언어 해석 기능을 활용하여 요구분석을 수행하려는 시도가 진행 중임
 - 생성형 AI 도구를 활용하여 소프트웨어 개발자와 도메인 전문가의 요구사항 관련 회의를 요약하고, 작성된 요구사항을 검증하고 좀 더 알기 쉽게 수정함
 - LLM(Large Language Model)의 환각 현상을 줄이기 위해 RAG(Retrieval-Augmented Generation) 기반 QA(질의응답) 시스템을 활용함
 - 생성형 AI는 특정 비즈니스의 트렌드, 비즈니스 모델, 관련 제도 등 일반적인 비즈니스 특성 조사에 활용할 수 있으며, 추출한 자료를 토대로 SW의 궁극적인 목표와 방향성을 결정하여 고객 요구사항 만족도 향상에 기여함
 - * 예를 들면 인터넷 금융 서비스에 필요한 기능을 생성형 AI에게 질의
- (SW 설계) 요구사항, 예제 및 메타모델에서 모델 합성을 자동화하여 설계를 자동화하는 방식이 고려되고 있으나, 생성형 AI 도구를 이용하여 시스템 아키텍처, 데이터 구조 등을 설계하는 데는 한계가 있음¹¹⁾
 - 생성형 AI가 구조 설계 단계를 전체적으로 수행하기는 어려우며, 부분적으로 활용하는 것도 제약사항이 많음
 - 설계 단계에서 여러 가지 가능성을 확인하는 차원에서 LLM에게 질의하거나, 활용할 기술에 대한 간단한 설명 및 사용 예시를 생성시켜 관련 기술 학습의 첫 단계로 활용할 수 있음
- (SW 구현) 개발 단계 중 생성형 AI를 가장 많이 활용하는 단계이며, 코드 생성, 코드 자동완성, 주석 작성, 코드 번역, 리버스 엔지니어링, 코드 리팩토링¹²⁾ 등이 가능하나, Hallucination, 저작권, 보안 등의 문제 이슈가 있음

11) K. Lano et al., Model Transformation Development Using Automated Requirements Analysis, Metamodel Matching, and Transformation by Example, ACM Transactions on Software Engineering and Methodology Vol. 31, No. 2, 2021.11.17.

12) 위키피디아, 소프트웨어 공학에서 '결과의 변경 없이 코드의 구조를 재조정함'을 뜻한다. 주로 가독성을

- 생성형 AI¹³⁾를 활용하여 소프트웨어 개발을 지원하는 도구를 개발자 통합개발환경(IDE, integrated development environment)에 연결하여 애플리케이션 코드를 자동으로 생성하고, 에러, 특히 프로그램 문법 오류 확인이 가능함

<표 1> SW 구현 시 활용되는 생성형 AI 도구

도구 이름	특징
GitHub Copilot	<ul style="list-style-type: none"> - Open AI의 GPT-3 이상의 알고리즘과 Github에 공개된 프로그램 언어와 프로젝트를 학습하여, Python, JavaScript, Go, Perl, PHP, Ruby, Swift, TypeScript, Shell 등 12개 이상 다양한 프로그래밍 언어로 자동 코딩 지원 - Github에 따르면 Copilot을 사용하는 개발자의 코딩 속도가 최대 55% 빨라졌으며, Copilot과 함께 작성된 코드의 경우 코드의 거의 절반이 자동 생성된다고 발표 - GitHub Copilot X는 GPT-4를 적용하였으며, Pull Request¹⁴⁾ 향상, 스크립트 및 셸 도구 지원, 의사코드 실행 같은 기능을 포함
CodeWhisperer	<ul style="list-style-type: none"> - Java, Python, JavaScript, TypeScript, C#의 5개의 프로그램 언어에 대해 활용 - 기존 코드를 기반으로 주석 및 코드를 보완하여 코드를 제안하며, 코드에 보안 문제가 있는지 검토
DeepCode	<ul style="list-style-type: none"> - 코드를 분석하여 찾기 어려운 미묘한 버그, 보안 취약점, 성능 문제를 감지 등 프로그램 오류를 확인하고 코드 리뷰를 하는 데 특화
Tabnine	<ul style="list-style-type: none"> - Python, Java, Node.js 등의 코드에 관한 Q&A, 코드 생성·리팩토링·검색·수정 및 설명 등 코드 구현에 특화된 도구 - 허가된 라이선스가 있는 오픈소스 소프트웨어에 대해서만 학습을 진행하여 라이선스 및 개인정보 보호 문제를 고려
ChatGPT	<ul style="list-style-type: none"> - 프로그램 코드 작성을 위해 특화된 도구는 아니나, 프로그래밍에 보조 도구로 활용 - 프롬프트 엔지니어링을 통해 코드 제안, 코드 리뷰 지원, 관련된 문서 작성 등에 활용

* 출처 : Michele Rosen(IDC, 2023), Generative AI Will Have the Greatest Impact on Software Development and Design in the Short Term, Martin Heller(2023), Review: CodeWhisperer, Bard, and Copilot X, Swimm Team, AI Code Review: How It Works and 3 Tools You Should Know, Christof Ebert and Panos Louridas(IEEE Software, 2023), Generative AI for Software Practitioners에서 SPRI 분석

높이고 유지보수를 편하게 한다. 버그를 없애거나 새로운 기능을 추가하는 행위는 아님, 2023.9.12. 조회

13) GitHub Copilot, ChatGPT, AWS CodeWhisperer, Snyk DeepCode 및 Tabnine 등이 가장 많이 활용되며, 이외에도 Akamas, CodeT5, Diffblue, Google, IBM watsonx Code Assistant, SeaLights, Sedai 등의 AI를 이용하여 소프트웨어 개발을 지원하는 도구가 존재

14) Github에서 사용자가 원격 저장소에 Push하여 새로운 사항이 적용됐을 경우, 다른 사용자에게 푸쉬된 상황을 알리고, 변경 사항을 코드 리뷰 받는 기능

- GitHub Copilot, AWS CodeWhisperer 등은 코드 생성에 특화된 도구이며, ChatGPT는 코드 생성을 위한 아이디어와 기초 정보를 생성하는 데 특화되어 있는 등 각각의 도구는 사용 목적이 다름
- 최근 데이터를 반영하지 못하는 생성형 AI의 기술적 한계로 새로운 라이브러리가 많이 출시되고, 빠른 주기로 업데이트되는 등 변화가 많은 부분에는 활용이 어려움
- 자동으로 생성되는 코드는 유효하지 않은 코드, 요구사항에 맞지 않는 코드, 보안 취약성, 라이선스 침해 등의 문제 소지가 있음
- 주석 작성, 다른 프로그래밍 언어로 변환 등의 작업은 시간이 많이 소요될 뿐만 아니라 소프트웨어 개발자에게 아주 지루한 작업으로, 생성형 AI를 활용하는 것은 개발 효율성을 높이고, 소프트웨어 개발자 만족도를 높임
- 프로그래밍 언어가 자연어와 같은 특성을 가지므로 생성형 AI 도구는 코드를 분석하여 주석을 만들거나, 코드 번역 등이 가능함
 - * 발전소, 국방 등 안전에 필수적인 분야에서 신뢰할 수 있는 SW의 상당수는 전문개발자가 부족한 코볼 등 구식 언어를 활용하고 있어 생성형 AI 도구를 코드의 작동방식을 설명하고 이를 번역함으로써 시스템 유지보수에 활용할 수 있음¹⁵⁾
- (SW 시험) 테스트는 시간이 많이 소요되고 반복적인 업무가 많아 생성형 AI를 이용하면 가장 소프트웨어 개발 효율성을 높일 수 있는 단계임
 - AI 및 생성형 AI를 사용하여 테스트케이스 우선순위 지정, 자동테스트, 테스트케이스 생성, 테스트 코드 작성, 테스트 데이터 생성 등에 활용이 가능함
 - 생성형 AI는 자연어 형태의 고객 요구사항이나 비즈니스 로직을 시험하기 위한 테스트 시나리오 작성이 가능함
 - 생성형 AI를 이용하여 테스트 결과에 대한 결함 분류, 결함 예측을 포함한 요약 및 분석보고서를 작성이 가능함

15) Christof Ebert and Panos Louridas(2023),Generative AI for Software Practitioners, IEEE Software Volume: 40 Issue: 4, 2023.7. - 8.

[그림 7] 인공지능/기계 학습(AI/ML)을 활용하는 시험(테스트) 분야



* 출처 : DevOps Practices, Tooling and Perceptions Survey (IDC# US49379723, Jan 2023)

- * 테스트 자동화, 데이터 생성, 시나리오 작성 등을 지원하는 AI 테스트 도구는 AccelQ, Avo automation, Functionize, Katalon, Keysight, Parasoft 등이 있음 (가트너, 2022)
- SW 시험 단계에서 가장 유용한 작업은 단일 기능에 대한 테스트 코드를 자동 생성하는 것이며, 전체 비즈니스 로직에 대한 테스트는 불가함
 - 생성형 AI를 이용하여 테스트케이스의 우선순위 지정은 가능하나, 가장 크리티컬한 테스트 케이스를 선정해 주는 것은 생성형 AI를 활용하기 어려움
 - 소규모의 기존 테스트 데이터를 활용하여 생성형 AI가 새로운 테스트 데이터를 생산함으로써 테스트 데이터의 부족 문제를 해결할 수 있으나, 편향된 훈련 데이터를 생성할 가능성이 있음
- (개발 지원 작업) 직접적인 소프트웨어 개발 이외에도 설명서 작성, 오류메시지 설명, 초급개발자의 프로그램 언어 학습 등에 생성형 AI를 활용할 수 있음
 - LLM이 인간의 언어와 프로그램 언어를 구분하지 않는다는 점에서, 언어 번역 기능과 유사한 작업인 소스코드 설명서 작성에 활용 가능
 - 초급개발자가 많은 시간을 할애하는 개발도구의 오류메시지 해석에 생성형 AI를 활용할 수 있으나, 오류메시지에 대한 설명이 정확하지는 않다는 점에 주의가 필요함
- (개발 생산성) 전반적으로 생성형 AI 개발도구를 활용하면 개발자의 생산성이 향상되는 것으로 분석되나, 이는 개발 단계, 개발하는 제품이나 서비스의 종류, 개발자의 수준 등에 따라 생성형 AI의 활용 방법 및 효과의 차이가 큼

- 개발 프로세스 중 생성형 AI를 활용하여 주목할 만한 생산성 향상이 보이는 부분은 SW 구현 단계 및 SW 시험 단계의 일부 작업임
- SI 사업, 인터넷서비스 기업 등 SW산업 분류에 따라 생성형 AI를 활용하는 분야와 방법이 다르며, 생산성 향상 정도가 상이할 수 있으나 정확한 측정은 어려움
 - 인터넷서비스의 경우는 기존 시스템의 기능을 추가하거나 신규 비즈니스를 기획할 때, 이미 인터넷에 있는 자료 및 비슷한 사례 조사를 통한 정보 수집에 생성형 AI를 활용할 수 있음
 - SI의 경우와 같이 고객의 요구사항이 다양하며 비즈니스 로직이 복잡하고, 시스템에 적용해야 하는 응용 프레임워크, 개발 표준, UI 표준 등의 회사 표준이 중요한 경우는 생성형 AI 도구의 활용에 제약이 큼
 - * 단, SI 사업에서 복잡하고 중요한 시스템 업그레이드 시 새로운 시스템 설계에 필요한 기존 시스템의 기능을 분석하는 데 생성형 AI를 유용하게 사용 가능
- 소프트웨어 개발자 수준에 대한 학계 및 업체의 통일된 개념 정의가 미흡하며 생성형 AI 출현 이후 개발자 역할 및 수준에 대한 논란이 많아졌으나, 본고는 아래 정의를 바탕으로 수준별 업무 변화 및 생산성 향상에 대해 정리함

<표 2> 소프트웨어 개발자 수준 정의

수준	정의 및 특징	생성형 AI 활용 형태
초급	<ul style="list-style-type: none"> - 기술과 개발 언어 문맥 이해를 기반으로 업무 소통 및 단일 기능 구현 가능 - 업무 지시가 필요하며 필요 기술 및 라이브러리에 대한 지식과 활용 미숙 	<ul style="list-style-type: none"> - 상급개발자 도움 최소화 및 자기학습 속도 향상
중급	<ul style="list-style-type: none"> - 트렌디한 업계 기술과 최신 라이브러리에 대한 지식을 기반으로 업무에 필요한 기술 스택을 찾아냄 - 각종 요구사항 반영 및 요청에 대한 납기 준수 가능 - 솔루션 개선책 제시 가능 - 코드 리뷰 및 주니어 리딩 가능 	<ul style="list-style-type: none"> - 주로 개발도구를 활용한 코딩 초안 작성 등에 생성형 AI(코파일럿 등)를 활용함으로써 업무 효율성 향상 - 코드 리뷰 및 코드 검색을 통한 기존 지식 환기 등에 생성형 AI를 활용함으로써 업무 효율성 향상
고급	<ul style="list-style-type: none"> - 솔루션 및 서비스 전체 구조 파악과 대부분의 문제 해결 가능 - 팀 운영을 통해 전체 프로젝트를 총괄 리딩 	<ul style="list-style-type: none"> - 기술 조합 및 설계에 대한 아이디어 소스를 얻는 데 생성형 AI를 활용 - 새로운 모델을 제안하고 생성형 AI를 발전시킬 수 있음

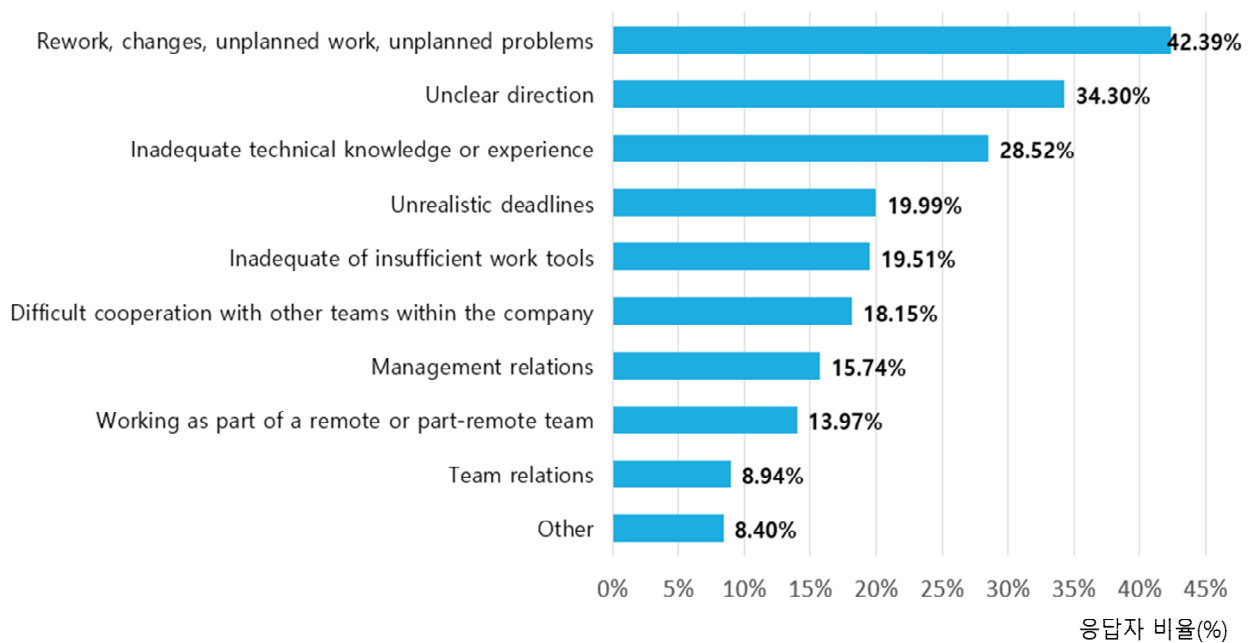
* 출처 : 문헌 검토, 전문가 자문을 통해 연구진이 정리

- 개발도구로서 생성형 AI의 활용은 초급개발자보다 생성형 AI 환각에 의한 잘못된 정보를 빠르게 판단할 수 있는 중급개발자의 생산성을 더 높임
 - 초급개발자는 생성형 AI를 프로그램 언어 학습에 유용하게 활용할 수 있으며, 개발자 수준이 빠르게 높아질 것으로 예측됨
 - 중고급 이상의 소프트웨어 개발자가 되기 위해서 컴퓨터공학의 기본적인 개념*을 알고 활용할 수 있는 능력이 필요하며, 이는 특히 생성형 AI가 생산하는 지식을 선별할 수 있는 기본적 역량임
 - * 운영체제(Operating System), 데이터 구조(Data Structure), 프로그램 언어(Program Language) 등
- (문제점) 생성형 AI의 사용은 신뢰성, 보안 등의 문제에 노출될 수 있으며, 개발 단계별 작업의 일부 단순 업무에만 생성형 AI 활용이 가능함
- 생성형 AI의 대부분 학습 데이터는 오픈소스와 같은 개방형 도메인에서 가져온 것으로, 최신 정보 부족, 신뢰성, 보안, 저작권 등 문제가 있음
 - 신뢰성 문제는 학습용 데이터에 프로그램 오류, 보안 취약성, 편향된 코드 등을 포함한 자료로 기인하여 발생함
 - 생성형 AI 도구의 활용은 기업의 내부 보안코드나 데이터의 유출 가능성이 있으며, 이로 인해 국가별, 기업별로 생성형 AI 활용을 제한하는 사례가 늘고 있음
 - SW를 통합하고 고객의 승인을 받는 단계에서는 개별 코드 생성만으로는 해결할 수 없는 문제들이 많기 때문에 통합적 지식이 있고 종합적 사고가 가능한 전문가 개입이 필요하여, 생성형 AI 개발도구 활용이 어려움
 - 생성형 AI 산출물에 대한 검증 과정을 생략할 경우, 소프트웨어의 오류는 물론 비즈니스 로직에도 문제가 생길 수 있으며, 복잡한 코드는 검증하는데 시간이 더 소요되어 활용성이 낮음
 - 초급개발자의 경우는 생성된 자료의 검증 과정에 시간이 많이 소요되어 오히려 생성형 AI의 사용이 생산성을 감소시킬 수 있으며, 생성형 AI가 제안한 코드의 의존성이 높아져 개발 역량 향상에 부정적인 영향을 미칠 가능성이 있음

III. 소프트웨어 개발자 직무변화

- 소프트웨어 개발자는 개발 이외에 많은 작업(Task)을 하고 있으며, 개발자 업무와 개발 생산성에 대한 생성형 AI의 영향 정도를 정량적으로 파악하기는 어려움
 - 개발자 직무를 수행하기 위해서는 데이터 분석, SW 및 DB 설계, 시스템 결정 및 성능 개선 등 개발 관련 기술 업무 이외에도, 프로젝트 관리, 동료와 업무 협의, 이슈 해결 등의 소프트스킬이 필요한 작업이 있음
 - 생성형 AI 출현으로 일부분 해소된 지식 습득의 어려움이나 개발도구의 부족 문제 이외에 개발자들이 업무에서 어려움을 느끼는 부분인 업무 변경, 정확하지 않은 업무 지시 등은 생성형 AI를 활용하기 어려움

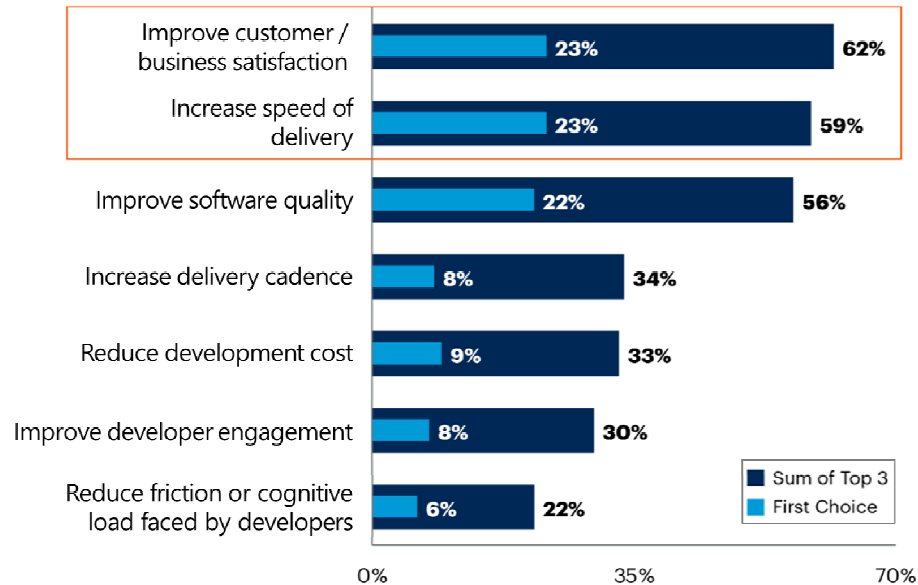
[그림 8] 소프트웨어 개발자 업무에서 어려운 점



* 출처 : CodinGame(2023), The State of Tech Hiring in 2023, 2022년 131개국 개발자 98개국 채용 담당자 18,200명 대상 조사

- Gartner(2023)는 SW 엔지니어링 리더의 개발 생산성 목표는 고객 만족도 향상, 개발 속도, SW 품질 순이라고 발표하였으며, 고객 승인 후에 제품이 완료됨으로 SW 개발 업무에 고객만족도의 중요성이 큼

[그림 9] 개발자 생산성 향상 목표



* 출처 : Hema Nair et al.(2023), Quick Answer: How Can Generative AI Tools Speed Up Software Delivery?, Gartner, 2023.6.8.

- SW 개발과 관련된 작업(Task)을 분석한 결과 컴퓨터 프로그래머, SW개발자, 웹개발자¹⁶⁾가 SW 개발과 직접적 관련이 있는 직무이며, 각 직무의 작업에서 생성형 AI의 영향을 받는 작업은 일부분임
 - 각 직업의 작업은 컴퓨터 프로그래머 17개, SW개발자 19개, 웹개발자 29개이며, 개발자 직무는 요구사항·분석 문서화, SW 설계, 구현, 테스트 등 SW 코딩을 포함한 SW 개발 업무 이외에 작업이 다수 포함됨
 - SW 연구, HW 엔지니어와 협력하여 시스템 통합, 웹사이트 인프라 개발, 웹사이트 성능 향상, 서버 프로세스 최적화 등 여러 작업 수행이 필요함
 - 개발 작업 중에도 생성형 AI에 영향을 가장 많이 받는 작업인 코드 작성, 테스트도 그 활용도가 한정적이며, 요구분석과 설계 단계의 작업은 영향도가 미비함
 - 구현 단계는 부분적으로 생성형 AI의 영향을 받으며, 복잡한 알고리즘 구현이

16) O*NET의 직업 중분류(O*NET-SOC 2019) 중 Computer Occupations (15-12**, 컴퓨터 직군) 조사, 본고는 개발에 직접적인 관련이 있는 컴퓨터 프로그래머(Computer Programmers), 소프트웨어개발자(Software Developers), 웹개발자(Web Programmers) 대상 분석

필요한 코딩을 제외하고 일반적으로 인터넷에서 찾을 수 있는 단순 기능의 코드가 생성형 AI를 활용하여 생성 가능함

- SW 개발 관련 직무의 65개 작업을 분석¹⁷⁾한 결과, 생성형 AI의 영향을 받는 직무는 개발 프로세스와 직접적인 관련이 있는 코드 작성 및 수정, 테스트팅 등임
- SW 관련 직무의 작업 중 생성형 AI의 영향을 가장 많이 받는 업무는 코드 작성, 코드 리뷰, 테스트 관련 작업, 사용설명서 작성 등임

<표 3> SW 관련 직무의 작업 중 생성형 AI에 영향을 가장 많이 받는 업무

Task	의견
Write computer programming code	-비즈니스 로직과 밀접하게 연관이 없는 일반적 코드 자동 작성 -이미 업계에서 활용되는 코드로 현업의 다수 개발자가 활용
Develop computer or online applications	-소프트웨어 개발에 필요한 모듈을 작성하는 데 유용
Test software performance	-코드 리뷰, 디버깅이 가장 주요하게 사용되는 영역 -코드 리뷰는 AI가 객관적인 시각이 개발자 역량 향상에 도움
Prepare instruction manuals	-설명서 초안에 활용할 수 있으나, 사용설명서는 사용자 친화적으로 만들 필요가 있어 사람의 검토가 필수적
Develop testing routines	-정확한 정보가 주어진다면 시나리오와 테스트 코드 생성 가능

* 출처 : O*NET 데이터를 SPRI 연구진이 분석

- 부분적으로 생성형 AI의 영향을 받는 업무는 SW 수정을 통한 성능 향상, 요구사항·설계와 개발 절차 문서화, 정보 검색 등이 있으나, 활용이 가능한 기능과 결과가 제한적이며 프롬프트 엔지니어링이 필수

<표 4> SW 관련 직무의 작업 중 생성형 AI에 영향을 받는 업무

Task	의견
Modify software programs to improve performance	-비효율적인 코드를 수정하는데 제한적으로 활용할 수는 있으며, 성능에 영향을 끼치는 코드 등을 탐지하는데 유용
Document design or development procedures	-문서화에 대한 초안을 생성하거나 생성된 문서를 보완 시 활용 -문서화는 개발자의 시간이 많이 소요되는 문서화에 LLM을 활용하여 개발 생산성 향상

17) SW 관련 분야 교수, SW 관련 기업의 대표, 전문 개발자, 아키텍트 등 10 여명의 SW 전문가 분석 (2023.9. ~ 10.)

Task	의견
Document technical specifications or requirements	-시스템별 특수성으로 인해 사용에 제약이 있으나, 부분적으로 활용 가능
Apply mathematical principles or statistical approaches to solve problems in scientific or applied fields	-LLM은 많은 시간 누적된 내용(역사, 이론, 기술 등)에 대해 높은 수준의 응답(대체로 맞는)을 생성 -문제를 해결하기 위한 기본 프롬프트 작성은 사람의 영역
Conduct research to gain information about products or processes	-널리 알려진 프로세스에 대해서는 생성 결과가 유용하나, 전문적이거나 특정 제품에 대한 정보는 부족

* 출처 : O*NET 데이터를 SPRI 연구진이 분석

- 개발자 직무 중 생성형 AI의 영향을 가장 많이 받는 직무는 O*NET 직무의 작업을 분석한 결과¹⁸⁾ 개발자¹⁹⁾는 생성형 AI를 활용할 수 없는 많은 작업을 수행하고 있음
 - 코드 작성 및 수정, 주석 작성, 테스트 등은 부분적으로 생성형 AI 도구를 활용할 수 있으나, 새로운 프로그래밍 방법 활용은 생성형 AI 도구 활용이 어려움

<표 5> 생성형 AI의 영향을 받는 개발자 세부 작업

세부 작업
입력, 출력 및 논리적 작동을 설명하는 워크플로우 차트와 다이어그램을 준비하고 코딩
코드 주석을 작성하고 개발 내용을 문서로 작성
웹 어플리케이션, 웹사이트 코드 개발
소프트웨어를 실행하여 원하는 정보를 생성하는지 테스트
테스트 계획, 테스트 절차 또는 테스트 결과를 문서화
원하는 결과가 나오도록 프로그램을 적절히 변경하고 다시 확인하여 오류 수정
운영 효율성을 높이거나 새로운 요구사항에 적응하기 위해 기존 프로그램의 수정
기존 SW 오류 수정 및 성능 개선, 새로운 하드웨어에 맞게 조정
코드를 평가하여 코드가 유효한지, 적절하게 구조화되어 있는지, 업계 표준을 충족하는지, 브라우저, 장치 또는 운영 체제와 호환되는지 확인
최종 사용자를 안내하기 위한 지침이나 매뉴얼을 작성

18) SW 관련 분야 교수, SW 관련 기업의 대표, 전문 개발자, 아키텍트 등 10 여명의 SW 전문가 분석 (2023.9. ~ 10.)

19) 컴퓨터 프로그래머이며, SW개발자와 웹개발자는 O*NET 상의 분류로 국내에서 통용되는 직무와 다른 기준의 직무이므로, 국내 실정에 맞추어 개발자로 통일함

- 개발자의 업무 중 사용자 요구 사항 분석, SW 솔루션 개발, 하드웨어 엔지니어와 협력하여 시스템을 통합하는 등 코딩과 테스트 이외의 많은 작업은 생성형 AI 도구를 활용할 수 없음
 - 개발자는 고객, 경영진, 다른 부서와의 협력이 필요하며, 네트워크 담당자, 데이터 담당자, 시스템 담당자들과 소통하여 시스템 설계 및 문제 해결을 해야 함
 - 개발자는 <표 5>의 작업 이외에 시스템 분석, 구조설계, 네트워크 설계 등의 다양한 업무를 수행하기 때문에 생성형 AI의 영향을 받는 작업이 많지 않음

<표 6> 생성형 AI의 영향을 받지 않는 개발자 세부 작업

작업 (Task)
프로그래머, 설계자, 기술자 등을 감독하고 업무 할당
사용자 요구사항을 분석하여 기술 요구사항 결정하고, 경영진 또는 개발팀과 상의하여 요구사항의 우선순위를 정하고 솔루션을 선정
시스템 분석가, 프로그래머 등과 상의하여 시스템을 설계하고 프로젝트의 한계와 기능, 성능 요구 사항 및 인터페이스에 대한 정보 획득
프로젝트 상태, SW 설계 등 사항에 대해 고객 또는 다른 부서와 상담
컴퓨터 제조업체 및 다른 사용자와 협력하여 새로운 프로그래밍 방법 개발
네트워크 담당자 또는 웹사이트 호스팅 대행사와 소통하여 웹 사이트에 영향을 미치는 하드웨어 또는 소프트웨어 문제를 해결
데이터 처리 관리자에게 데이터 처리 제한 사항이나 기능에 대한 정보 획득
시스템 기능 및 요구사항 분석을 위해 데이터를 저장, 검색 및 조작
정보를 분석하여 새 시스템의 설치 또는 기존 시스템의 수정을 결정하고 계획, HW 구성을 결정하기 위해 보고서, 비용 또는 보안 요구 사항 등을 수집하고 평가
서버 부하, 대역폭, 데이터베이스 성능, 브라우저 및 디바이스와 같은 기술적 요소를 문서화
예산, 장비, 성능 요구 사항, 접근성 및 개인정보 보호를 포함한 법적 문제 등 기술적 고려 사항을 웹사이트 디자인 계획에 통합
웹 아키텍처 또는 기술에 대한 조사, 문서화, 평가 또는 대안 선택
물리적, 인터페이스, 논리적 또는 데이터 모델을 포함하는 웹 모델 생성
시스템 상호 작용 또는 시퀀스 다이어그램을 개발
제품 기능, 활동, 소프트웨어, 통신 프로토콜, 프로그래밍 언어, 운영 체제 등 웹 사이트 사양에 대한 명확하고 자세한 설명을 제공
저작 또는 스크립팅 언어, 콘텐츠 제작 도구, 관리 도구, 디지털 미디어를 사용하여 웹 사이트를 설계, 구축 또는 유지 관리
시스템 프로그래머로서 시스템 분석 및 프로그래밍
프로그래밍 교육 및 사용자에게 프로그램의 사용법과 기능에 대해 교육
시스템이 사양에 맞게 작동하는지 확인하기 위해 장비의 기능을 모니터링

IV. 생성형 AI 시대의 소프트웨어 개발자 양성을 위한 정책 제언

- **소프트웨어 개발자 수준(초·중·고급)에 따른 세분화된 생성형 AI 활용 전략 마련 필요**
 - 생성형 AI의 영향은 개발자 수준에 따라 차별적일 것으로 확인되는바, 각 수준별 개발자들이 그들의 역량 향상과 성과 제고에 생성형 AI를 전략적으로 활용할 수 있도록 지원 필요
 - 초급개발자에게는 생성형 AI 활용 방법 교육에 비중을 두는 것이 필요하며, 생성형 AI를 효과적으로 활용할 수 있는 초급개발자의 새로운 역할 범위를 형성하고 적응할 수 있도록 지속적 지원이 필요함
 - * 생성형 AI의 기능이 초급개발자 과업 일부를 대체할 수 있다 하더라도, 초급개발자의 역할은 다른 형태로라도 여전히 존재한다는 전문가 의견 존재(진회승 외, 2023)
 - 중·고급 개발자는 생성형 AI를 활용한 개발성과 향상이 예측되며, 생성형 AI에 대한 직접적 교육보다는 신기술에 대한 습득 기회* 제공에 초점을 맞추어 지원하는 것이 필요함
 - * 예시: 최신 기술 동향 세미나, 컨퍼런스 참여 지원, 네트워킹 기회 확대, 생성형 AI 개발도구 구독료 지원, 데이터 튜닝을 위한 API 사용료 지원 등
- **직무별로 상이한 생성형 AI의 영향을 고려하여 소프트웨어 개발자 양성 규모 검토 필요**
 - 향후 개발자 양성 사업 설계 시, 생성형 AI의 영향이 제한적일 것으로 예상되는 직무를 식별하여, 해당 직무를 담당하는 개발자 대한 교육과 양성 규모를 확대하는 것이 필요
 - 본 연구의 분석 결과, SW 개발단계 중 ‘구현 단계’는 생성형 AI의 영향이 클 것으로 확인되며, 특히 코드 작성, 애플리케이션 개발, 테스트 등의 업무에 영향이 클 것으로 확인됨
 - 반면, 커뮤니케이션의 중요도가 높은 ‘요구분석’과 통합적이고 복잡하고 다양한 기술을 구현해야 하는 ‘설계’ 단계는 작업 시 생성형 AI의 영향이 제한적일 것으로 확인됨

- 향후 생성형 AI로 인한 생산성 개선이 제한적이며, 커뮤니케이션 및 통합적 사고 역량이 요구되는 직무를 담당하는 인력양성에 정부의 지원을 보다 집중할 필요
- **개발자 양성 및 교육 도구로서 생성형 AI의 효과적 활용에 대한 고민 필요**
- 본 연구의 분석 결과, 생성형 AI는 프로그램 언어 학습, 코드 생성 등에 활용도가 높아 초급개발자들의 개발 지원에 활용될 여지가 큼
 - 특히, 생성형 AI는 초급개발자 양성을 상당 부분 담당하는 비정규 교육과정에서 교육의 질적 수준을 높일 수 있는 매우 좋은 활용 도구임
 - 수업 시 생성형 AI를 활용한다면, 초급개발자 교육생들이 많은 시간을 할애하는 프로그램 언어 학습, 코드 생성에 대한 습득 시간을 단축시킬 수 있을 것으로 판단됨
 - 초급개발자 양성 교육에 생성형 AI 활용 시, 기존의 단편적인 지식의 습득 위주의 교육에서 벗어나, 문제의 발굴과 해결에 대한 창의적인 접근 역량을 제고할 수 있는 교육에 더 집중할 수 있을 것으로 판단됨
 - 따라서, 정부에서 지원하는 개발자 인력양성사업의 교육 도구로서 생성형 AI의 효과적 활용방안에 대해 모색할 필요가 있음
- **소프트웨어 개발자의 책임성 확보를 위한 개발자 대상 AI 윤리 교육 강화 필요**
- 생성형 AI를 활용하면서 직면할 수 있는 저작권 문제, 프라이버시 문제, 모델 오남용 문제에 대해 적절히 대응할 수 있도록 기본적인 AI 윤리 교육에 대한 강화가 필요함
 - 커리큘럼에 AI 윤리 교육과정의 비중을 높이거나 필수로 포함하는 방안을 고려해 볼 수 있음
 - 나아가, 정부는 개발자들이 참고할 수 있는 생성형 AI 활용 윤리 가이드라인을 개발 및 배포하고, AI 윤리 준수에 대한 자율적 노력 체계를 마련할 필요가 있음

[별첨] O*NET 개발자 관련 직무 정의

□ Computer Programmers 정의

컴퓨터 애플리케이션을 실행할 수 있는 코드와 스크립트를 생성, 수정 및 테스트를 주로 수행하며, 소프트웨어 및 웹 개발자 또는 다른 개인이 작성한 사양을 바탕으로 작업함

유사 용어 : Computer Programmer, Internet Programmer, Java Programmer, Programmer, Web Applications Programmer, Web Programmer

□ Software Developers 정의

컴퓨터 및 네트워크 소프트웨어 또는 전문 유틸리티 프로그램을 연구, 설계, 개발을 주로 수행하며, 컴퓨터 과학, 공학 및 수학적 분석의 원리와 기술을 적용하여 사용자 요구사항을 분석하고 소프트웨어 솔루션을 개발함. 이해관계자와 협력업무 비중이 높으며 컴퓨터 하드웨어 엔지니어와 협력하여 하드웨어 및 소프트웨어 시스템을 통합하고 사양 및 성능 요구 사항을 개발하는 업무와 애플리케이션 영역 내에서 데이터베이스를 유지 관리하며 개별적으로 작업하거나 팀의 일원으로 데이터베이스 개발을 조율하는 업무를 수행함

유사 용어 : Application Developer, Application Integration Engineer, Infrastructure Engineer, Network Engineer, Software Architect, Software Developer, Software Development Engineer, Software Engineer, Systems Engineer

□ Web Developers 정의

웹관련 업무를 수행하는 직무를 통칭하며, 다음과 같은 업무가 포함됨

- 웹사이트, 웹 애플리케이션, 애플리케이션 데이터베이스, 대화형 웹 인터페이스를 개발
- 코드가 적절하게 구조화되어 있는지, 업계 표준을 충족하는지, 브라우저 및 디바이스와 호환되는지 평가
- 웹사이트 성능, 확장성, 서버 측 코드 및 프로세스 최적화
- 웹사이트 인프라를 개발하고 웹사이트를 다른 컴퓨터 애플리케이션과 통합

유사 용어 : Web Architect, Web Design Specialist, Web Developer, Webmaster

참고문헌

1. 참고문헌

- 김경은 (2024). “생성형 AI 이용현황 및 노동 대체 가능성에 대한 이용자 인식조사”, 소프트웨어중심사회.
- 진희승 (2015). “SW MDD 유용성 논의와 사례 분석”, 소프트웨어정책연구소.
- 진희승·윤보성·신승윤 (2023). “생성형 AI에 대응한 SW인재양성 정책 방향 연구”, 소프트웨어정책연구소 연구보고서.

2. 국외문헌

- Allied Market Research (2023.7). “Generative AI Market Size, Share, Competitive Landscape and Trend Analysis Report by Component (Software, Service), by Technology (Generative Adversarial Networks (GANs), Transformer, Variational Autoencoder (VAE), Diffusion Networks, Retrieval Augmented Generation), by End User (Media and Entertainment, BFSI, IT and Telecom, Healthcare, Automotive and Transportation, Others): Global Opportunity Analysis and Industry Forecast, 2023-2032”
- Chavi Mehta (2023.6.23.). “US-based generative AI job postings up 20% in May, Indeed data show”, Reuters.
- Christof Ebert and Panos Louridas (2023.). “Generative AI for Software Practitioners”, IEEE Software Volume: 40 Issue: 4, 2023.7. - 8.
- CodinGame(2023.). The State of Tech Hiring in 2023
- Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). “Gpts are gpts: An early look at the labor market impact potential of large language models”, arXiv preprint arXiv:2303.10130.
- Gartner (2021). “Market Share Analysis: Application Development Software, Worldwide, 2021”
- Gartner (2021). “Top Strategic Technology Trends for 2022”
- Hema Nair et al. (2023). “Quick Answer: How Can Generative AI Tools Speed Up Software Delivery?”, Gartner.
- K. Lano et al. (2021). “Model Transformation Development Using Automated Requirements Analysis, Metamodel Matching, and Transformation by Example”, ACM Transactions on Software Engineering and Methodology Vol. 31, No. 2
- Laurie Wurster et al. (2023). “Market Share: Application Development, Worldwide, 2022”, Gartner
- Mckinsey (2023). “The economic potential of generative AI: The next productivity frontier”
- Mckinsey (2023). “Unleashing developer productivity with generative AI”
- Michele Rosen (2023). “Generative AI Will Have the Greatest Impact on Software Development and Design in the Short Term”, IDC
- Melinda-Carol Ballou (2023). “Generative AI Offers Greatest Expected Developer Benefits to Software Quality and Testing Followed by App Security Testing”, IDC
- Varsha Mehta et al. (2022). “Forecast Analysis: Low-Code Development Technologies, Worldwide”, Gartner

3. 기 타

- DevOps Practices, Tooling and Perceptions Survey (IDC# US49379723, Jan 2023)
- Martin Heller (2023). “Review: CodeWhisperer, Bard, and Copilot X”, InfoWorld
- Pramodbablad (2023). “History Of Programming Languages”, LinkedIn
- Reuter (2023.6.23.). “US-based generative AI job postings up 20% in May, Indeed data show”
- StackOverFlow (2023). “2023 Developer Survey”
- StackOverFlow (2023). “Most popular uses of AI in the development workflow among developers worldwide as of 2023”
- Statista Market Insights (2023.8). 생성형 AI - 세계 기준(Generative AI - Worldwide). 2023년 9월 기준 (<https://www.statista.com/outlook/tmo/artificial-intelligence/generative-ai/worldwide#market-size>)

주 의

이 보고서는 소프트웨어정책연구소에서 수행한 연구보고서입니다.
이 보고서의 내용을 발표할 때에는 반드시
소프트웨어정책연구소에서 수행한 연구결과임을 밝혀야 합니다.



생성형 AI에 의한 개발자 업무 영향 분석

경기도 성남시 분당구 대왕판교로 712번길 22 글로벌 R&D 연구동(B)

Global R&D Center 4F 22 Daewangpangyo-ro 712beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do

www.spri.kr