

AI-First(6+1) 운영 프로그램이 비전공 학습자의 프로젝트 완수율, 설계 효능감, 인지부하에 미치는 영향: 직업훈련 실무형 프로젝트 사례

유 순 복¹ · 배 현 영^{2*}

¹한국기술교육대학교 IT융합과대학경영산업대학원 AI융합교육학과 석사과정

²한국기술교육대학교 IT융합과대학경영산업대학원 AI융합교육학과 겸임교수

Effects of an AI-First (6+1) Program on Project Completion, Design Self-Efficacy, and Cognitive Load in Non-Major Vocational Training

Soon-Bog Yoo¹ · Hyun-Young Bae^{2*}

¹M.S. in AI Convergence Education, Graduate School of IT Convergence and Industrial Management, Korea University of Technology and Education (KOREATECH), Cheonan 31253, Korea

²Adjunct Professor, Department of AI Convergence Education, Graduate School of IT Convergence and Industrial Management, Korea University of Technology and Education (KOREATECH), Cheonan 31253, Korea

[요 약]

본 연구는 직업능력개발훈련의 실무형 웹개발 교육에서 AI-First 운영원칙을 적용한 1개월 추가 프로젝트(6+1)가 비전공 학습자의 프로젝트 수행 성과와 심리 요인에 미치는 영향을 검증한다. 비동등 통제집단 사전-사후 설계로 A 직업훈련기관 ‘풀스택 자바 개발자 과정’ 학습자 60명(대조군 30명, 실험군 30명)을 분석하였다. 두 집단은 동일한 6개월 커리큘럼을 이수하였고, 실험군은 추가 1개월 동안 단계화된 프로젝트 절차와 AI 활용 규칙, 체크리스트·테스트 기반 산출물 검증을 적용하여 프로젝트를 수행하였다. 종속변수는 프로젝트 완수율(Pass/Fail), 설계 효능감, 인지부하(NASA-TLX)이며 전공 계열과 사전 코딩 경험을 조절변수로 포함하였다. 분석 결과, 실험군은 대조군보다 완수율이 유의하게 높았고 설계 효능감 증가폭이 더 컸으며 인지부하 감소폭도 더 크게 나타났다. 전공 계열의 조절효과는 유의하지 않았으나 사전 코딩 경험은 설계 효능감과 인지부하에서 일부 조절효과가 확인되었다. 본 결과는 AI-First(6+1) 복합 처치의 프로그램 효과로 해석한다.

[Abstract]

This study evaluated a one-month AI-First (6+1) add-on project in vocational web-development training for non-majors. Using a nonequivalent control-group pre-post design, 60 trainees (control n = 30; experimental n = 30) in a full-stack Java program were analyzed. After the same six-month core curriculum, the experimental group completed an additional one-month project using staged workflow, AI-use rules, and checklist-based verification. Outcomes included project completion (pass/fail), self-efficacy, and NASA-TLX workload. Major field and prior coding experience were tested as moderators. The experimental group achieved higher completion, greater self-efficacy gains, and lower workload. Major field was not significant, but prior coding experience moderated self-efficacy and workload. Findings reflect program-level effects of the bundled AI-First (6+1) intervention.

색인어 : AI-First 교수법, 비전공 학습자, 프로젝트 완수율, 설계 효능감, 인지부하

Keyword : AI-First Pedagogy, Non-Major Learners, Project Completion, Design Self-Efficacy, Cognitive Load

<http://dx.doi.org/10.9728/dcs.2026.27.3.751>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 05 February 2026; **Revised** 05 March 2026

Accepted 16 March 2026

***Corresponding Author; Hyun-Young Bae**

E-mail: youg989@daum.net

I. 서론

대규모언어모델(LLM) 기반 생성형 인공지능(이하 생성형 AI)은 설명·예시·피드백 제공을 넘어 코드 생성·수정·디버깅 지원까지 가능해지면서 학습 지원 도구로 빠르게 확산되고 있다. 메타분석 연구는 ChatGPT 활용이 학습 성과뿐 아니라 학습 지각 및 고차 사고에도 긍정적 효과를 보일 수 있음을 보고하였다[1]. 또한 최근 생성형 AI 교육 연구는 학습성과, 학습자 인식, 만족도, 사용 경험을 중심으로 그 영향을 검토하는 경향을 보여 왔다[1]-[4]. 그러나 생성형 AI의 응답에는 오류 가능성이 상존하므로, 학습자가 산출물을 점검·검증·수정하도록 유도하는 교수-학습 설계가 병행되지 않으면 오개념 형성이나 오류 전파로 이어질 수 있다. 따라서 생성형 AI를 교육에 통합할 때에는 단순한 사용 권장에 그치지 않고, 오류 탐지와 검증을 포함한 운영 원리를 명시적으로 설계할 필요가 있다.

이러한 필요성은 단기간에 실무 역량을 요구하는 직업능력 개발훈련의 프로그래밍·웹개발 교육에서 더욱 두드러진다. 실무형 프로젝트는 요구사항 분석-설계-구현-디버깅-검증이 연속적으로 결합되는 복합 과업이며, 비전공 학습자는 학습 초기의 진입장벽과 시행착오 누적의 위험을 경험하기 쉽다[5]-[7]. 특히 “무엇을, 어떻게 검증할 것인가”에 대한 기준이 불명확하면 오류 수정과 설계 의사결정의 부담이 가중될 수 있다. 따라서 비전공 학습자의 프로젝트 완수를 지원하기 위해서는 과업 수행 과정을 단계화하고, 산출물 기준과 검증 절차를 구조화하는 접근이 요구된다.

그러나 기존 생성형 AI 교육 연구는 주로 학습성과, 학습자 인식, 만족도, 사용 경험 등을 중심으로 효과를 검토해 왔으며[1]-[4], 비전공 직업훈련의 실무형 프로젝트 맥락에서 프로젝트 완수와 같은 객관적 수행지표와 설계 효능감, 인지부하를 함께 분석한 연구는 상대적으로 제한적이다. 특히 프로그래밍 학습에서는 높은 실패율과 중도탈락 가능성이 반복적으로 보고되어 왔고[5],[6], 비전공 학습자의 경우 과제 수행 과정에서 경험하는 인지적 부담과 자기효능감이 학습 지속과 성과에 영향을 미칠 수 있음이 제시되어 왔다[8],[9]. 이러한 점에서 생성형 AI의 활용 효과를 단순한 인식이나 만족도 차원이 아니라, 실제 프로젝트 수행 성과와 심리적 학습 요인을 함께 고려하여 검토할 필요가 있다. 따라서 본 연구는 비전공 직업훈련 맥락에서 운영 규칙과 산출물 검증 절차를 구조화한 AI-First(6+1) 운영 프로그램의 효과를 프로젝트 완수율, 설계 효능감, 인지부하 측면에서 실증적으로 검증하고자 한다.

본 연구는 비동등 통제집단 사전-사후 설계를 적용한다. 대조군(전통 수업 6개월 집단)과 실험군은 동일 기관의 서로 다른 기수로 구성되며, 두 집단 모두 동일한 6개월 핵심 커리큘럼을 이수한다. 다만 실험군은 과정 종료 후 1개월 동안 AI-First 운영모형을 적용한 추가 프로젝트를 수행한다. 따라서 본 연구에서 검증하는 처치는 “AI-First 운영모형 적용 + 1개월 추가 프로젝트 수행”이 결합된 복합 처치(6+1)이며,

이하 이를 AI-First(6+1) 운영 프로그램이라 한다. 코호트 차이는 동일 운영 조건 유지, 사전 동질성 검증, 사전점수 통제 등을 통해 완화하고자 한다. 프로젝트 수행 성과는 필수 기능 체크리스트 충족 여부에 따른 프로젝트 완수율(Pass/Fail)로 평가하며, 심리적 학습 장벽은 설계 효능감과 인지부하(NASA-TLX)로 측정한다. 또한 효과의 이질성을 탐색적으로 검토하기 위해 전공 계열과 사전 코딩 경험 수준을 조절변수로 포함한다. 다만 본 연구의 처치는 복합 처치이므로, AI-First 운영모형의 단독효과를 분리하여 주장하지 않으며, 결과 해석은 AI-First(6+1) 운영 프로그램에 대한 프로그램 평가 범위 내에서 제한한다.

이상의 목적을 달성하기 위해 본 연구는 다음과 같은 가설을 설정한다.

- H1. AI-First(6+1) 운영 프로그램 집단은 전통 수업(6개월) 집단보다 프로젝트 완수율이 유의하게 높을 것이다.
- H2. 사전 설계 효능감을 통제했을 때, AI-First(6+1) 운영 프로그램 집단의 사후 설계 효능감은 전통 수업(6개월) 집단보다 유의하게 높을 것이다.
- H3. 사전 인지부하를 통제했을 때, AI-First(6+1) 운영 프로그램 집단의 사후 인지부하는 전통 수업(6개월) 집단보다 유의하게 낮을 것이다.
- H4. 전공 계열은 AI-First(6+1) 운영 프로그램의 효과를 조절할 것이다.
 - H4a. 프로젝트 완수율에 대한 AI-First(6+1) 운영 프로그램 효과는 전공 계열에 따라 유의한 차이를 보일 것이다.
 - H4b. 설계 효능감에 대한 AI-First(6+1) 운영 프로그램 효과는 전공 계열에 따라 유의한 차이를 보일 것이다.
 - H4c. 인지부하에 대한 AI-First(6+1) 운영 프로그램 효과는 전공 계열에 따라 유의한 차이를 보일 것이다.
- H5. 사전 코딩 경험 수준은 AI-First(6+1) 운영 프로그램의 효과를 조절할 것이다.
 - H5a. 프로젝트 완수율에 대한 AI-First(6+1) 운영 프로그램 효과는 코딩 경험 수준에 따라 유의한 차이를 보일 것이다.
 - H5b. 설계 효능감에 대한 AI-First(6+1) 운영 프로그램 효과는 코딩 경험 수준에 따라 유의한 차이를 보일 것이다.
 - H5c. 인지부하에 대한 AI-First(6+1) 운영 프로그램 효과는 코딩 경험 수준에 따라 유의한 차이를 보일 것이다.

II. 이론적 배경

2-1 생성형 AI와 프로그래밍 학습 지원

생성형 AI, 특히 대규모 언어모델(LLM) 기반 도구는 학습자의 질문에 대해 설명과 예시를 제공하고, 코드 생성·수정·

리팩토링 및 디버깅을 지원함으로써 프로그래밍 학습 과정에서 학습 지원 도구로의 활용이 확대되고 있다. 51개 실험 및 준실험 연구를 대상으로 한 메타분석은 ChatGPT 활용이 학습 성과에서 유의한 개선을 보였고, 학습 지각 및 고차 사고에서도 긍정적 효과가 보고되었음을 제시한다[1]. 또한 프로그래밍 교육 맥락에서 생성형 AI(예: ChatGPT) 통합을 검토한 일부 연구는 생성형 AI 통합이 학습자의 과제 수행 경험과 사용성을 개선하는 동시에 인지적 부담을 완화할 가능성을 시사한다[10]. 더 나아가 생성형 AI를 단순한 ‘도구’가 아니라 ‘협업 파트너’로 위치시키는 접근은 프로그래밍 불안 감소와 학습 동기 및 수행 향상과 관련된 차이를 보고한 바 있다[11]. 이러한 논의는 생성형 AI가 초보 학습자의 문제 해결 과정에서 즉각적 피드백과 오류 진단 지원을 제공함으로써 학습자 경험을 개선할 수 있음을 보여준다.

다만 생성형 AI 기반 학습 지원이 일관되게 긍정적 결과로 이어진다고 단정하기는 어렵다. 교육적 적용에서는 학습자가 AI의 제안을 그대로 수용하기보다, 산출물의 타당성을 확인하고 수정하는 활동을 수행하도록 설계할 필요가 있다. 즉 AI 활용을 허용하는 수준을 넘어, 수업 목표와 과제 단계에 맞춘 AI 활용 규칙과 산출물 검증 절차를 포함하는 학습 활동의 구조화가 요구된다. 특히 비전공 학습자가 설계 의사결정과 오류 진단에서 부담을 크게 경험할 수 있다는 점을 고려하면, 생성형 AI 활용을 과제 단계별로 구조화하고 결과물 검증을 학습 과정에 포함하는 교수·학습 설계는 학습 장벽 완화에 중요한 역할을 할 수 있다.

2-2 비전공자 프로그래밍 학습 장벽

비전공 학습자는 프로그래밍 학습에서 개념 이해(추상화·흐름 제어 등)와 구문 오류 해결, 디버깅, 과제 요구사항 해석 등 다양한 인지 과정을 동시에 처리해야 하므로 높은 인지적 부담을 경험하기 쉽다. 인지부하 이론은 작업기억의 제한으로 인해 과제 요구가 복잡해질수록 학습에 사용할 수 있는 인지 자원이 소진되어 학습 효율이 저하될 수 있음을 설명한다[8]. 또한 적절한 지원이 제공되지 않을 경우 문제 해결 과정이 스키마 형성에 충분히 기여하지 못하고 실패 경험을 누적시켜 학습 지속을 저해할 가능성이 있음을 시사한다[8]. 인지부하 관점에서 학습자의 부담을 관리하는 교수 전략은 문제 해결 수행과 전이에 영향을 미칠 수 있으며[12], 이는 복합 과업을 다루는 학습 환경에서 학습 지원 구조의 필요성을 뒷받침한다.

이러한 장벽은 직업능력개발훈련에서 수행하는 실무형 프로젝트 맥락에서 더욱 두드러질 수 있다. 프로젝트 기반 과제는 요구사항 분석, 설계, 구현, 디버깅, 검증이 연속적으로 결합되는 경우가 많으며, 학습자는 ‘무엇을 구현해야 하는가’뿐 아니라 ‘어떤 구조로 연결해야 하는가’까지 함께 해결해야 한다. 이 과정에서 인지적 부담이 증가하면 불안과 좌절이 강화될 수 있고, 이는 학습 지속과 성취에 부정적 영향을 줄 수 있다. 실제로 입문 프로그래밍에서의 높은 실패율은 반복적으로

보고되어 왔으며[5], 성인 학습자를 대상으로 한 프로그래밍 e-learning 과정에서도 중도탈락과 그 영향 요인이 주요 문제로 제시된다[6]. 국내 부트캠프형 SW·AI 교육을 대상으로 한 연구에서도 학습 이탈 방지 요인으로 동료 관계, 수행 성취감, 운영자와의 관계 등이 도출되어 학습 지속을 뒷받침하는 지원 구조의 중요성을 시사한다[7].

한편 협업 기반 학습 설계의 효과를 다룬 짝 프로그래밍 연구는 학습자의 자신감, 산출물 품질, 학습 유지 측면에서 긍정적 영향을 보고한 바 있다[13]. 이는 비전공 학습자의 장벽을 낮추고 학습 지속을 높이기 위해 학습 지원 구조가 중요함을 뒷받침한다. 따라서 생성형 AI를 협업 파트너로 통합하되, 학습 활동을 단계화하고 검증 절차를 포함해 구조화하는 교수·학습 설계는 비전공 학습자의 부담을 완화하고 지속을 촉진하는 방향으로 기능할 가능성이 있다[11],[13],[14]. 요컨대 비전공 학습자의 높은 인지적 부담과 반복되는 오류 수정 경험은 프로젝트 완수의 어려움으로 연결될 수 있으며, 수행 과정에서의 자기신념(효능감) 저하와 주관적 부담(인지부하) 증가를 동반할 가능성이 있다.

2-3 핵심 개념 및 연구변수의 조작적 정의

1) AI-First 운영모형 및 AI-First(6+1) 운영 프로그램

본 연구에서 AI-First 운영모형은 생성형 AI 사용을 단순히 허용하는 수준을 넘어, 문제 해결 절차를 요구사항 분석-설계-구현-디버깅-검증으로 단계화하고, 각 단계에 AI 활용 규칙과 산출물 검증 절차를 명시적으로 포함하도록 설계된 운영모형을 의미한다. 즉 생성형 AI를 학습 과정의 ‘협업 파트너’로 활용하되, 학습자가 수행해야 할 활동을 단계별로 구조화하고 각 단계에서 산출물의 타당성을 점검하도록 설계한다는 점에 특징이 있다. 특히 검증 단계에서는 체크리스트 및 테스트(시나리오/통합 테스트 등)를 통해 결과물을 확인하고, 검증 결과를 반영한 수정과 리팩토링을 수행하도록 한다.

한편 본 연구에서 실증적으로 검증하는 처치는 AI-First 운영모형을 적용한 1개월 추가 프로젝트 수행을 포함하는 AI-First(6+1) 운영 프로그램이다. 즉, 본 연구의 처치는 AI-First 운영모형 적용과 1개월 추가 프로젝트 기간이 결합된 복합 처치이다. 따라서 본 연구는 AI-First 운영모형의 단독효과를 분리하여 주장하지 않으며, 결과 해석은 AI-First(6+1) 운영 프로그램에 대한 프로그램 평가의 범위 내에서 제한한다.

2) 프로젝트 완수율

프로젝트 완수율은 프로젝트 수행 성과를 나타내는 객관적 지표로서, 실무형 과제에서 요구되는 핵심 기능 구현의 달성 여부를 반영한다. 프로그래밍 학습에서 성취 수준의 격차와 실제 경험은 학습 지속 및 성과와 밀접하게 관련될 수 있으며[5], 협업 기반 학습 설계가 산출물 품질과 학습 유지에 긍정적 영향을 줄 수 있다는 논의[13]를 고려할 때 ‘프로젝트의 완료/미완료’는 학습 성과를 요약하는 핵심 성과 지표로 활용될

수 있다. 본 연구에서 프로젝트 완수율은 필수 기능 체크리스트 충족 여부에 따라 프로젝트 완료(Pass) 또는 미완료(Fail)로 판정한 비율로 조작화한다. 이에 본 연구는 AI-First(6+1) 운영 프로그램이 실무형 프로젝트 산출물 완성에 미치는 영향을 검증하기 위한 성과 지표로 프로젝트 완수율을 설정한다.

3) 설계 효능감

효능감은 특정 과제를 성공적으로 수행할 수 있다는 개인의 신념으로, 학습 동기와 학습 지속, 수행에 영향을 미치는 핵심 변인으로 다뤄져 왔다[9]. 컴퓨터 활용 맥락에서는 컴퓨터 자기효능감이 실제 사용과 정서적 반응 및 수행과 관련될 수 있다는 근거가 제시되었고[15], 프로그래밍 맥락에서도 프로그래밍 자기효능감을 측정하기 위한 척도가 개발·타당화되어 학습 성과 예측에 활용되어 왔다[16]. 또한 설계 수행에 대한 자기효능감은 공학 설계 과제 맥락에서 측정 도구가 제안·검증되어 왔으며[17], ‘설계’ 활동이 학습자의 자기신념과 밀접하게 연결될 수 있음을 시사한다.

본 연구는 이러한 선행 논의를 바탕으로, 실무형 프로젝트 맥락에서 요구사항을 구조화하고 구성요소 간 관계를 설계하며, 오류 상황을 진단하고 수정 방향으로 연결하는 설계 활동을 수행할 수 있다는 자기신념을 설계 효능감으로 정의한다. 즉 설계 효능감은 구현 이전 단계에서의 구조화·연결·진단 중심의 설계 수행 가능성에 대한 신념을 의미하며, AI-First(6+1) 운영 프로그램이 학습자의 심리적 학습 장벽에 미치는 영향을 설명하는 핵심 변인으로 설정한다.

4) 인지부하

인지부하는 과제 수행 과정에서 학습자가 경험하는 정신적 노력 및 작업 부담을 의미하며, 복잡한 문제 해결 과제에서 학습 효율을 설명하는 핵심 개념이다[8]. 본 연구에서는 인지부하를 프로젝트 수행 중 학습자가 지각한 주관적 작업부하로 조작화한다. 인지부하 측정에서는 과제 수행 후 학습자가 지각한 부담을 평가하는 주관적 평정 척도가 널리 활용되어 왔으며[18], 주관적 작업부하를 정신적 요구, 신체적 요구, 시간적 요구, 수행, 노력, 좌절감 등 다차원으로 측정하는 대표적 도구로 NASA-TLX(Task Load Index)가 제시되어 다양한 과제 환경에서 활용되어 왔다[19]. 이에 본 연구는 NASA-TLX를 사용하여 프로젝트 수행 과정의 주관적 작업부하를 측정하고, AI-First(6+1) 운영 프로그램이 인지부하를 완화하는지를 검증한다.

III. 연구방법

3-1 연구모형

본 연구는 AI-First 운영모형을 적용한 1개월 추가 프로젝트 운영(6+1개월), 즉 AI-First(6+1) 운영 프로그램이 비전공 학습자의 프로젝트 완수율, 설계 효능감, 인지부하(주관적

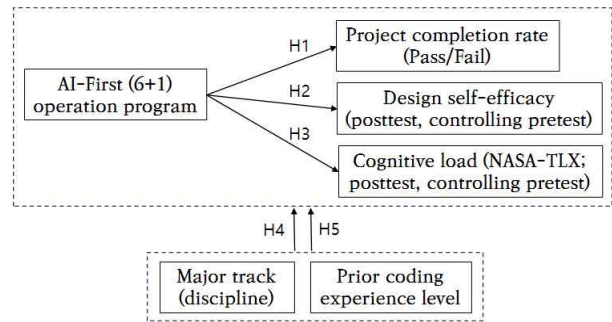


그림 1. 연구모형
Fig. 1. Research model

표 1. 연구 변수의 개념 정의 및 측정 개요
Table 1. Conceptual definition of research variables and measurement overview

Category	Item	Conceptual definition	Measurement/coding
Independent	AI-First (6+1) operational program	AI-integrated, stage-based instruction with rules & verification	Group (0=Control(6m), 1=AI-First(6+1))
Dependent	Project Completion	Project outcome based on required functions	Checklist Pass/Fail (Pass=1, Fail=0)
	Design Self-Efficacy	Confidence in design/system tasks	5-point Likert, 6 items (pre/post)
	Subjective workload	Perceived workload during project tasks	NASA-TLX (6 subscales), 5-point; mean score (pre/post)
Moderating	Major track	Major category	STEM=1, non-STEM=0
	Prior coding experience	Prior coding experience duration	≥3 months=1, <3 months=0

작업부하)에 미치는 영향을 검증한다. 또한 AI-First(6+1) 운영 프로그램의 효과가 학습자의 전공 계열과 사전 코딩 경험 수준에 따라 달라지는지(조절효과)를 함께 검토한다. 연구 모형은 그림 1에 제시하였고, 연구 변수의 개념 정의 및 측정 개요는 표 1과 같다.

3-2 연구설계 및 연구대상

1) 연구설계

본 연구는 직업능력개발훈련 맥락에서 무작위 배정이 어려운 점을 고려하여 비동등 통제집단 사전-사후 설계를 적용하였다. 사전검사는 설계 효능감과 인지부하(주관적 작업부하), 그리고 주요 배경변수(전공 계열, 사전 코딩 경험 등)를 포함하여 실시하였다.

대조군(Control)은 전통적 방식으로 운영된 6개월 풀스택 과정을 이수한 학습자이며, 실험군은 동일한 6개월 과정 이수 후 AI-First 운영모형에 따라 1개월 추가 운영(연장) 단계에서 프로젝트를 수행한 학습자로 구성하였다. 두 집단은 동일한 최종 프로젝트 요구사항과 동일한 평가 기준(부록)을 적용

받았다. 최종 프로젝트는 댓글 추천 기능 및 이중 프레임워크(Spring Boot-Flask) 연동 등 복합 기능 구현을 포함하는 실무형 과제로 구성되며, 대조군은 6개월 과정 내에서 전통적 방식으로 이를 수행한 반면 실험군은 동일 과제를 AI-First 운영 원칙(단계화·AI-use rules·산출물 검증 절차) 하에 1개월 연장하여 수행하였다. 실험집단의 추가 운영은 총 4주간 진행되었으며, 기관 운영표 기준 주 5일, 일 8시간으로 산정할 때 학습자 1인당 총 추가 투입시간은 160시간이었다. 따라서 본 연구의 처치는 AI-First 운영모형 적용과 1개월 추가 운영이 결합된 복합 처치(6+1개월)이다. 이에 따라 본 연구는 AI-First 운영모형의 단독효과와 추가 운영기간의 효과를 엄밀히 분리하여 검증하는 설계는 아니며, 결과는 AI-First(6+1) 운영 프로그램의 프로그램 수준 효과 범위 내에서 해석한다.

또한 대조군에서도 생성형 AI 사용 자체는 허용하였으나, 실험군과 달리 AI-use rules 및 산출물 검증 프로토콜은 제공하지 않았다. 생성형 AI 활용 정도는 사용빈도 문항으로 측정하여 분석에서 통제하였으며, 조작점점 문항을 통해 단계 기반 수행, 프롬프트 구조화, 오류 로그 및 재현 절차 정리, AI 출력 검증 여부 등 실제 활용 양상을 함께 확인하였다. 실험집단의 조작점점 총점 평균은 4.18(SD=0.52)로 나타났으며, 이는 AI-First 운영 원칙의 이행이 전반적으로 이루어졌음을 시사한다. 다만 본 연구의 AI 사용 통제는 시스템 로그 기반 직접 측정이 아니라 자기보고 문항에 근거한 부분 통제라는 점에서 해석상 한계를 가진다.

두 집단은 서로 다른 기수로 구성되므로(대조군: 2024년 하반기, 실험군: 2025년 상반기) 코호트 차이로 인한 역사효과 및 운영 변동을 최소화하기 위해 핵심 운영 조건을 가능한 범위에서 동일하게 유지하였다. 구체적으로 (1) 동일 기관의 동일 과정(풀스택 자바 개발자 양성 과정)과 동일한 6개월 핵심 커리큘럼, (2) 핵심 실습 환경(개발 도구·서버/DB·LMS 등)과 주요 과제 범위, (3) 프로젝트 완수율의 평가 기준(필수 기능 체크리스트) 및 평가 절차(집단 정보 배제 블라인드 판정)를 일관되게 적용하였다. 또한 사전 동질성 검정과 사전점수 통제를 통해 집단 간 초기 차이를 통계적으로 완화하였다.

독립변수는 수업 운영 조건(전통 수업(6개월) vs AI-First(6+1) 운영 프로그램)이며, 종속변수는 프로젝트 완수율(H1), 설계 효능감(H2), 인지부하(H3)로 설정하였다. 집단 간 비교의 타당성을 높이기 위해 사전 측정 변인을 활용하여 초기 수준을 확인하였다.

가설 검증은 종속변수의 자료 유형과 반복측정 설계를 고려하여 수행하였다. 프로젝트 완수율(H1)은 사후 시점의 이항 결과(Pass/Fail)이므로 집단 간 완수율 차이를 χ^2 검정으로 1차 비교하되, 필요 시 Fisher의 정확검정을 적용하였다. 이어 전공 계열 및 사전 코딩 경험 등 배경변수를 포함한 이항 로지스틱 회귀분석을 실시하여 집단 효과를 추가로 추정하였으며, 결과는 오즈비(OR)와 95% 신뢰구간으로 보고하였다.

설계 효능감(H2)과 인지부하(H3)는 사전-사후 반복측정 자료이므로 혼합설계(집단×시점) 분산분석을 주분석으로 적용하고, 집단×시점 상호작용을 중심으로 처치 효과를 검정하였다. 또한 결과의 강건성을 확인하기 위해 사후 점수를 종속변수로 하고 사전 점수를 공변량으로 투입한 공분산분석(ANCOVA)을 보조 분석(민감도 분석)으로 수행하였다.

조절효과(H4, H5)는 전공 계열 및 사전 코딩 경험 수준이 수업 운영 조건의 효과를 조절하는지 확인하기 위해 상호작용항을 포함한 모형으로 검정하였다. 완수율(H1)은 로지스틱 회귀모형에 집단×전공, 집단×코딩경험 상호작용항을 포함하여 분석하였다. 설계 효능감과 인지부하는 혼합설계 분산분석에서 집단×시점×전공 및 집단×시점×코딩경험 상호작용을 통해 조절효과를 검정하였다. 전공 계열은 STEM(이공계열)과 Non-STEM(타계열)으로 구분하고, 사전 코딩 경험은 3개월 미만과 3개월 이상으로 재범주화하여 분석하였다.

2) 연구대상

본 연구는 A 직업훈련기관에서 운영된 ‘풀스택 자바 개발자 양성 과정’ 수강생 중 연구 참여에 자발적으로 동의한 학습자 60명을 대상으로 하였다. 대조군은 2024년 하반기 동일 과정 수강생 30명, 실험군은 2025년 상반기 동일 과정 수강생 30명으로 구성하였다. 두 집단은 모두 동일한 6개월 핵심 커리큘럼을 이수하였으며, 두 기수의 최종 프로젝트 요구사항과 평가 기준(부록의 필수 기능 체크리스트 및 시나리오 기반 테스트)을 동일하게 적용하였다. 다만 실험군은 과정 종료 후 생성형 AI를 협업 파트너로 활용하는 AI-First 운영모형(단계화·AI-use rules·산출물 검증 절차) 하에 동일 프로젝트를 1개월 추가 운영 단계에서 수행하였다.

연구 참여는 자발적 동의에 기반하였으며, 설문은 익명으로 수집하였다. 수집된 자료는 연구 목적의 통계 분석에 한해 사용하였고, 개인을 식별할 수 있는 정보는 분석에 포함하지 않았다.

연구대상자의 사전 특성은 남성 35명(58.3%), 여성 25명(41.7%)이었고, 연령은 26-30세 29명(48.3%)이 가장 많았다. 전공은 Non-STEM 45명(75.0%), STEM 15명(25.0%)이었으며, 사전 코딩 경험은 3개월 미만 43명(71.7%), 3개월 이상 17명(28.3%)이었다. GPT 사용(주당 사용일수)은 5일 이상 41명(68.3%), 0-4일 19명(31.7%)로 나타났다. χ^2 검정 결과 성별($p=1.000$), 연령($p=0.960$), 전공($p=1.000$), 코딩 경험($p=1.000$), GPT 사용($p=1.000$)에서 집단 간 분포 차이는 유의하지 않았다(표 2). 또한 생성형 AI 사용 정도는 사용빈도 문항으로 측정하여 민감도 분석에서 공변량으로 통제하였다.

표 2. 연구 대상자의 인구통계학적 특성(N=60)

Table 2. Demographic characteristics of research participants (N=60)

Variable	Category	Control (n=30)	Experimental (n=30)	Total (N=60)	%	p(χ^2)
Gender	Male	17	18	35	58.3	1.000
	Female	13	12	25	41.7	
Age	20-25	10	11	21	35.0	0.960
	26-30	15	14	29	48.3	
	≥31	5	5	10	16.7	
Academic Major	STEM	8	7	15	25.0	1.000
	Non-STEM	22	23	45	75.0	
Coding Experience	<3 months	21	22	43	71.7	1.000
	≥3 months	9	8	17	28.3	
GPT Use	≥5 days/week	20	21	41	68.3	1.000
	0-4 days/week	10	9	19	31.7	

3-3 AI-First 교수법 처치 및 수업 운영

1) AI-First 교수법의 개념 및 단계별 설계

본 연구에서 실험집단에 적용한 AI-First 운영모형은 생성형 AI 활용을 단순 허용하는 수준을 넘어, 문제 해결 과정을 요구사항 분석(Req)-설계(Design)-구현(Build)-디버깅(Debug)-검증(Verify)으로 단계화하고 각 단계에 AI 활용 규칙과 산출물 검증 절차(체크리스트·시나리오 기반 테스트 및 리뷰)를 포함하도록 학습 활동을 구조화한 운영모형이다. 학습자는 생성형 AI의 제안을 초안으로 활용하되, 오류 재현(로그·증거)-수정-재검증의 반복적(Iterative) 검증을 수행하도록 안내받는다. AI-First 운영모형의 단계별 운영은 다음과 같다.

- (1) 요구사항 분석(Req): 기능 요구를 목록화·우선순위화한 뒤, AI를 활용하여 사용자 시나리오 및 요구사항 명세를 도출·정교화한다.
- (2) 설계(Design): 데이터 흐름과 API 구조를 도식화하고 책임 분리를 명확히 한 후, AI를 통해 누락 요소 및 예외 상황을 점검한다.
- (3) 구현(Build): 모듈 단위 구현을 수행하되, AI가 생성한 코드 초안의 요구사항 적합성을 자체 점검하고 실행·테스트 결과를 근거로 기록한다.
- (4) 디버깅(Debug): 오류 로그와 재현 절차를 문서화한 뒤, AI로 원인 후보 및 수정 방안을 도출하고 수정 후 재검증을 반복한다.
- (5) 검증(Verify): 필수 기능 체크리스트 및 시나리오 기반 테스트로 완성도를 확인하고, 결과를 반영해 수정·리팩토링 및 문서화를 수행한다.

2) 수업 운영 및 처치 내용

실험집단은 7개월 과정 중 마지막 1개월을 AI-First 운영모형을 적용한 추가 운영(연장) 단계로 운영하였다. 해당 기

간 동안 실험집단은 생성형 AI를 협업 파트너로 활용하여 Req-Design-Build-Debug-Verify의 단계 기반 절차(stage-based workflow)에 따라 과업을 수행하였으며, 단계별 AI 활용 규칙(AI-use rules)과 산출물 검증(artifact verification; 체크리스트·시나리오 기반 테스트 및 리뷰)을 포함하도록 안내받았다. 특히 오류 처리 과정에서는 오류 로그와 재현 절차를 기반으로 consult AI → fix → re-verify(반복적 검증) 루틴을 적용하였다. 교수자는 단계별 산출물과 검증 기록을 점검하고 피드백을 제공하였다.

대조군에서도 생성형 AI 사용은 허용되었으나, AI-First 운영모형에서 제공한 단계별 AI-use rules 및 산출물 검증 프로토콜은 제공하지 않았다.

두 집단의 최종 프로젝트 요구사항과 연구 평가 기준(부록의 필수 기능 체크리스트 및 시나리오 기반 테스트)은 동일하게 적용하였다. 즉, 집단 간 차이는 (1) AI-First 운영 원칙(단계화·AI-use rules·산출물 검증 프로토콜 제공 여부)과 (2) 1개월 추가 운영(연장) 단계의 유무에 있다. 두 집단 간 수업 운영의 차이는 표 3과 같다.

또한 본 연구의 처치는 AI-First 운영모형 적용과 1개월 추가 운영이 결합된 복합 처치(6+1개월)이므로, 결과 해석은 AI-First(6+1) 운영 프로그램의 효과 범위 내에서 제한한다. AI-First 운영 원칙의 적용 여부는 조작점검 문항을 통해 확인하였다.

표 3. 수업 운영 비교

Table 3. Instructional method comparison

Category	Control (6 months)	AI-First (6+1 months)
Operating principle	Instructor feedback + self-directed work	Stage-based workflow (Req-Design-Build-Debug-Verify) with AI-use rules and artifact verification
Extended period	None	+1-month extension phase under AI-First principles
Error handling	Search/community + feedback	consult AI → fix → re-verify (iterative)
Verification practice (during development)	Basic functional checks	Checklist + scenario-based tests
Outcome assessment (study evaluation)	Same required-feature checklist + scenario-based tests (Appendix; blind rating)	Same required-feature checklist + scenario-based tests (Appendix; blind rating)

3) 주차별 AI-First 처치 상세

1개월 추가 기간 동안 실험집단에 적용된 처치는 AI-First 운영모형을 적용한 추가 프로젝트 운영(AI-First(6+1) 운영 프로그램)이다. 본 처치는 문제 해결 절차를 요구사항 분석-설계-구현-디버깅-검증으로 단계화하고, 각 단계에서 생성형 AI를 협업 파트너로 활용하되 AI 활용 규칙과 산출물 검증 절

차(체크리스트/테스트/리뷰)를 포함하도록 설계하였다. 학습자 배경 특성(전공 계열, 사전 코딩 경험)의 이질성을 고려하여 동일한 AI-First 운영 가이드라인을 모든 학습자에게 제공하였으며, 이해를 돕기 위한 예시 및 템플릿은 상황에 따라 선택적으로 활용하도록 안내하였다.

• 1주차(프롬프트 기초 및 요구사항 정의)

프로젝트 아이디어를 기능 요구사항으로 구체화하고, AI 질의 구조(목표-제약-입력/출력-예외)를 학습한다. 기능 목록(필수/선택)과 사용자 시나리오를 작성한 뒤, AI를 활용해 요구사항 명세 초안을 생성·수정한다.

주요 산출물: 기능 목록(필수/선택), 사용자 시나리오, 요구사항 명세(수정 이력 포함)

• 2주차(아키텍처 설계 및 구현 초안 생성)

구성요소(프론트-백-API-DB-인증/권한) 관계를 도식화하고 DB 스키마 및 API 설계안을 작성한다. 이후 AI를 통해 누락 요소(예외 처리, 권한, 오류 응답 등)를 점검하고, 모듈 단위 구현을 시작한다.

주요 산출물: 구조도/흐름도, DB 스키마, API 정의서, 모듈별 구현 계획 및 코드 초안

• 3주차(시스템 통합 및 AI 협업 디버깅)

React-Spring Boot-Flask 구성요소를 통합하고 연동 오류를 재현·기록한다. 학습자는 오류 로그, 재현 절차, 기대 출력 등을 정리해 AI에 원인 후보와 수정 방안을 질의하고, 수정 후 재검증을 반복한다.

주요 산출물: 오류 로그(재현 절차 포함), 수정 내역, 재검증 결과, 리팩토링 기록

• 4주차(리팩토링 및 검증·내면화)

핵심 기능 구현 후 AI를 활용해 리팩토링 후보(중복 제거, 책임 분리, 예외 처리 강화 등)를 도출하고, 변경 사항은 체크리스트 및 테스트(시나리오/통합 테스트 등)로 검증한다. 최종 산출물은 README 및 설계 요약 문서로 정리한다.

주요 산출물: 체크리스트 충족 결과, 테스트 결과(시나리오 포함), 리팩토링 내역, 최종 문서(README/설계 요약)

예를 들어, 시스템 통합 단계에서 일부 학습자는 React-Spring Boot-Flask 연동 과정에서 오류가 발생했을 때 AI가 제시한 코드를 그대로 적용하기보다, 오류 로그와 재현 절차, 기대 출력값을 먼저 정리한 후 AI에 원인 후보와 수정 방안을 질의하도록 안내받았다. 이후 수정된 결과는 체크리스트와 시나리오 기반 테스트를 통해 다시 검증하였고, 필요 시 리팩토링과 문서화를 반복하였다. 이러한 과정은 AI 사용 자체보다 오류 재현-수정-재검증의 구조화된 협업 절차가 실제 프로젝트 수행에 적용되었음을 보여준다.

3-4 측정도구 및 자료수집

본 연구는 AI-First(6+1) 운영 프로그램의 효과를 검증하기 위해 프로젝트 완수율(H1), 설계 효능감(H2), 인지부하(H3)를 측정하였으며, 전공 계열(H4)과 사전 코딩 경험(H5)을 조절변수로 포함하였다.

프로젝트 완수율은 교육과정 종료 시 제출된 최종 산출물을 대상으로, 사전에 정의한 필수 기능 체크리스트 충족 여부에 따라 평가하였다. 완수(Pass)는 필수 기능을 충족하고 실행·시연이 가능한 상태로 정의하였으며, 2인의 평가자가 집단 정보를 배제한 블라인드 상태에서 독립 판정하였다. 불일치가 발생한 경우 합의를 통해 판정을 확정하였고, 최종 판정은 이분형(Pass/Fail)으로 기록하였다.

설계 효능감은 실무형 프로젝트 맥락에서 요구사항을 구조화하고 구성요소 간 관계를 설계하며 오류를 진단·수정 방향으로 연결할 수 있다는 자기신념으로 정의하였다. 총 6문항을 5점 리커트 척도로 측정하였고, 문항 평균 점수를 분석에 사용하였다.

인지부하(주관적 작업부하)는 NASA-TLX(Task Load Index)를 활용하였다. NASA-TLX는 정신적 요구, 신체적 요구, 시간적 요구, 수행(Performance), 노력, 좌절감의 6요인을 평가하는 도구이며[19], 본 연구에서는 각 요인을 5점 척도로 평정하였다. 6요인 점수의 평균을 종합지표로 사용하였고, 가중치 절차는 적용하지 않았다(점수가 높을수록 인지부하가 높음).

전공 계열(H4)은 사전 설문을 통해 STEM(이공계열)과 Non-STEM(타계열)으로 코딩하였다. 사전 코딩 경험(H5)은 교육 참여 전 코딩 학습·실습 경험 기간(개월 수)으로 측정하였으며, 분석에서는 3개월 미만(<3 months)과 3개월 이상(≥ 3 months)으로 재범주화하였다.

자료수집은 사전-사후 절차에 따라 진행하였다. 사전 설문은 교육과정 시작 시점(입과 시점)에, 사후 설문은 최종 프로젝트 종료 시점에 실시하여 설계 효능감과 인지부하를 측정하였다. 사전 인지부하는 입과 시점에 실시한 진단 과제(미니 과제/프로젝트) 수행 직후 NASA-TLX로 측정하였다. 프로젝트 완수율은 최종 산출물의 체크리스트 기반 평가로 산출하였고, 실험집단의 처치 적용 여부를 확인하기 위해 조작점 검 문항을 사후에 추가로 수집하였다. 본 연구의 사전 측정은 입과 시점에 실시되었으므로, 1개월 추가 운영 단계 직전의 중간 측정은 포함하지 않는다. 따라서 본 연구의 효과 추정은 AI-First 운영 원칙과 1개월 추가 운영이 결합된 AI-First(6+1) 운영 프로그램 단위의 효과로 해석하며, 구성요소별(운영 원칙 vs 추가 학습시간) 단독효과를 분리하여 주장하지 않는다.

3-5 자료분석 방법

설계 효능감 척도와 인지부하(NASA-TLX) 문항의 신뢰

도를 확인하기 위해 Cronbach's α 를 산출하였으며, 교육·심리 연구 관례에 따라 α 값이 .70 이상일 때 수용 가능한 수준으로 판단하였다. NASA-TLX는 다차원 작업부하 척도이므로, 본 연구에서는 6요인 평균 점수를 종합지표로 사용한다는 전제하에 내적 일관성을 참고 지표로 확인하였다. 프로젝트 완수 여부(Pass/Fail)는 2인의 평가자가 집단 정보를 배제한 블라인드 상태에서 독립 판정하였으므로 Cohen's κ 를 산출하여 평가자 일치도를 확인하였다.

처치 이전 집단 간 초기 수준을 확인하기 위해 연속형 변인(설계 효능감, 인지부하)의 사전 점수에 대해 독립표본 t-검정을 실시하였다. 성별, 연령, 전공 계열(STEM/Non-STEM), 코딩 경험(3개월 미만/3개월 이상), GPT 사용빈도(주 5일 이상 vs 0-4일) 등 범주형 변인에 대해서는 χ^2 검정을 적용하되, 기대빈도가 5 미만인 셀이 존재하는 경우 Fisher의 정확검정을 사용하였다.

가설 검증은 종속변수의 자료 유형과 반복측정 설계를 고려하여 수행하였다. 프로젝트 완수율(H1)은 사후 시점의 이항 결과(Pass/Fail)이므로 집단 간 완수율 차이는 χ^2 (또는 Fisher) 검정으로 1차 비교하고, 전공 계열 및 사전 코딩 경험 등 배경변수를 포함한 이항 로지스틱 회귀분석으로 집단 효과를 추가 추정하였다. 로지스틱 회귀모형에서는 다중공선성과 모형 적합도 등 기본 진단을 점검하였고, 결과는 오즈비(OR)와 95% 신뢰구간으로 보고하였다. 또한 통제집단의 생성형 AI 사용 가능성(오염)을 고려하여 GPT 사용빈도(주 5일 이상 vs 0-4일)를 추가 공변량으로 포함한 민감도 분석 로지스틱 회귀모형을 추가로 추정하였다.

설계 효능감(H2)과 인지부하(H3)는 사전-사후 반복측정 자료이므로 반복측정 분산분석을 주분석으로 적용하였다. 구체적으로 집단(전통 수업(6개월) vs AI-First(6+1) 운영 프로그램), 시점(사전/사후), 그리고 집단×시점 상호작용을 중심으로 처치 효과를 검증하였다. 측정 시점이 2수준(사전/사후)이므로 구형성(sphericity) 가정은 문제되지 않으며, 분석에 앞서 이상치 및 정규성 등 기본 가정을 점검하였다. 효과 크기는 부분 $\eta^2(np^2)$ 로 보고하였다.

또한 결과의 강건성을 확인하기 위해 사후 점수를 종속변수로 하고 사전 점수를 공변량으로 투입한 공분산분석(ANCOVA)을 보조 분석(민감도 분석)으로 수행하였다. ANCOVA 적용에 앞서 회귀기울기 동질성(집단×사전점수 상호작용)을 검증하여 집단 간 공변량-종속변수 관계가 동일한지 확인하였고, 등분산성(Levene의 검정) 등 주요 가정을 점검하였다. 동일한 취지에서, ANCOVA 보조 분석에서는 사전 점수 외에 GPT 사용빈도를 추가 공변량으로 투입한 모형을 함께 검토하였다.

조절효과(H4, H5)는 전공 계열 및 사전 코딩 경험이 수업 운영 조건의 효과를 조절하는지 확인하기 위해 상호작용항을 포함한 모형으로 검증하였다. 완수율(H1)은 로지스틱 회귀모형에 집단×전공, 집단×코딩경험 상호작용항을 포함하여 조절효과를 검증하였고, 설계 효능감과 인지부하는 반복측정 분

산분석에서 집단×시점×전공 및 집단×시점×코딩경험 상호작용을 통해 조절효과를 검증하였다. 모든 검정은 양측검정으로 수행하였고, 통계적 유의수준은 $p<.05$ 로 설정하였다.

IV. 연구결과

4-1 기초 분석 및 측정도구 신뢰도

본 절에서는 본격적인 가설 검증에 앞서 집단 간 사전 특성의 동질성을 점검하고, 측정도구의 신뢰도 및 산출물 평가의 평가자 일치도를 확인하였다. 먼저 연구대상자의 인구통계학적 특성은 표 2에 제시하였다. 성별, 연령, 전공 계열(STEM/Non-STEM), 사전 코딩 경험(<3개월/≥3개월), GPT 사용빈도(≥5일/주 vs 0-4일/주) 등 범주형 변인에 대한 집단 간 분포 차이는 χ^2 검정(기대빈도 5 미만 시 Fisher의 정확검정)으로 확인하였으며, 전반적으로 두 집단 간 유의한 차이가 나타나지 않아($ps>.05$) 집단의 사전 분포는 통계적으로 유사한 것으로 판단하였다(표 2).

다음으로 처치 이전의 초기 수준을 확인하기 위해 연속형 변인인 설계 효능감과 인지부하의 사전 점수에 대해 독립표본 t-검정을 실시하였다. 그 결과, 설계 효능감 사전 점수는 집단 간 유의한 차이가 나타나지 않았고($t(58)=0.42, p=.676$), 인지부하 사전 점수 역시 집단 간 유의한 차이가 나타나지 않았다($t(58)=-0.36, p=.721$). 이에 따라 두 집단은 주요 종속변수의 사전 수준에서 통계적으로 유사한 상태에서 출발한 것으로 보고, 이후 분석에서는 사전-사후 반복측정 설계에 기반한 집단×시점 상호작용을 중심으로 처치 효과를 검증하였다.

측정도구의 신뢰도는 설계 효능감 6문항과 인지부하(NASA-TLX) 6문항에 대해 Cronbach's α 를 산출하여 확인하였다. 그 결과 설계 효능감은 $\alpha=.89$, 인지부하는 $\alpha=.81$ 로 나타나 수용 가능한 내적 일관성을 확보한 것으로 판단하였다(표 4). 또한 프로젝트 완수 여부(Pass/Fail)는 체크리스트 기준에 따라 평가자(강사 및 외부 전문가)가 교차 평가하였으므로, 평가자 일치도를 Cohen's κ 로 산출하였다. 그 결과 $\kappa=.90$ 으로 나타나 판정의 일관성이 높은 것으로 확인되었다(표 4).

아울러 실험집단에서 AI-First 운영 원칙이 실제로 적용되었는지를 확인하기 위해 조작점검 문항을 사후에 수집하였다. 조작점검 평균은 $M=4.18, SD=0.52$ 로 나타나 실험집단에서 단계화된 수행(요구사항 분석-설계-구현-디버깅-검증), 프롬프트 구조화, 로그-재현 기반 디버깅, 체크리스트·테스트 기반

표 4. 측정도구 신뢰도 및 평가자 일치도

Table 4. Reliability and inter-rater agreement

Measure	N	Statistic	Value
Design Self-Efficacy	6	Cronbach's α	0.89
Subjective workload (NASA-TLX)	6	Cronbach's α^*	0.81
Project Completion (Pass/Fail)	-	Cohen's κ	0.90

검증이 전반적으로 이루어졌음을 시사한다.

4-2 기술통계 분석

표 5는 집단별 사전-사후 기술통계를 제시한다. 설계 효능감의 사전 평균은 대조군 2.84(SD=0.54), AI-First(6+1) 운영 프로그램 집단 2.88(SD=0.55)로 유사하였으며, 사후에는 대조군 3.20(SD=0.57), AI-First(6+1) 집단 3.78(SD=0.52)로 나타나 AI-First(6+1) 집단의 증가 폭이 상대적으로 크게 관찰되었다. 인지부하(주관적 작업부하; NASA-TLX) 역시 사전 평균이 대조군 3.58(SD=0.62), AI-First(6+1) 집단 3.54(SD=0.64)로 유사하였고, 사후에는 대조군 3.34(SD=0.59)로 소폭 감소한 반면 AI-First(6+1) 집단은 2.76(SD=0.58)로 더 큰 감소 폭이 관찰되었다. 프로젝트 완수율은 대조군 60.0%(18/30), AI-First(6+1) 집단 83.3%(25/30)로 AI-First(6+1) 집단이 더 높게 나타났다.

요약하면, 기술통계 수준에서 AI-First(6+1) 운영 프로그램 집단은 대조군에 비해 설계 효능감은 더 크게 증가하고 인지부하는 더 크게 감소하는 경향을 보였으며, 프로젝트 완수율도 더 높게 나타났다. 이러한 차이의 통계적 유의성은 다음 절의 가설 검증에서 확인하였다.

표 5. 주요 변수 기술통계(집단별 사전/사후) 및 프로젝트 완수율
Table 5. Descriptive statistics of key variables (pre/post) and project completion rate

Outcome	Group	Pre M (SD)	Post M(SD)
Design Self-Efficacy	Control (n=30)	2.84 (0.54)	3.20 (0.57)
	AI-First(6+1) (n=30)	2.88 (0.55)	3.78 (0.52)
Cognitive Load (NASA-TLX)	Control (n=30)	3.58 (0.62)	3.34 (0.59)
	AI-First(6+1) (n=30)	3.54 (0.64)	2.76 (0.58)
Outcome	Group	Pass n(%)	Fail n(%)
Project Completion	Control (n=30)	18 (60.0)	12 (40.0)
	AI-First(6+1) (n=30)	25 (83.3)	5 (16.7)

4-3 가설 검증 결과

1) 프로젝트 완수율(H1)

프로젝트 완수율(H1)은 사후 시점의 이항 결과(Pass/Fail)이므로, 주분석으로 교차표를 통해 집단 간 완수율 차이를 비교하고 χ^2 검정을 실시하였다. 그 결과, 대조군의 완수율은 60.0%(18/30)인 반면 AI-First(6+1) 집단의 완수율은 83.3%(25/30)로 나타났으며, 집단 간 차이는 통계적으로 유의하였다($\chi^2(1)=4.02$, $p=.045$). 완수율의 절대 차이는 23.3%p였다. 보조 분석으로 전공 계열과 코딩 경험을 포함한 이항 로지스틱 회귀분석(Model 1)을 실시한 결과, 집단 효과가 유의하게 나타났고 AI-First(6+1) 집단의 완수 가능성은

대조군 대비 약 3.62배 높은 것으로 추정되었다(OR=3.62, 95% CI [1.10, 11.90], $p=.034$). 또한 통제집단의 생성형 AI 사용 가능성(오염)을 고려하여 GPT 사용빈도를 추가 공변량으로 투입한 민감도 분석(Model 2)에서도 집단 효과는 유사한 수준으로 유지되었다(OR=3.40, 95% CI [1.01, 11.45], $p=.048$). 따라서 주분석과 보조 분석 결과를 종합하면, AI-First(6+1) 운영 프로그램은 프로젝트 완수율을 유의하게 향상시키는 것으로 나타났으며 H1은 지지되었다(표 6).

표 6. 집단별 프로젝트 완수율(H1)

Table 6. Group differences in project completion (H1)

Group	Pass n(%)	Fail n(%)	$\chi^2(1)$	p
Control (n=30)	18 (60.0)	12 (40.0)	4.02	.045
AI-First(6+1) (n=30)	25 (83.3)	5 (16.7)		

표 7. 이항 로지스틱 회귀분석 결과

Table 7. Group differences in project completion (H1) and logistic regression results

Predictor	Model 1 OR	Model 1 95% CI	Model 1 p	Model 2 OR	Model 2 95% CI	Model 2 p
Group (AI-First(6+1)=1)	3.62	[1.10, 11.90]	.034	3.40	[1.01, 11.45]	.048
Academic Major (STEM=1)	1.35	[0.40, 4.52]	.624	1.22	[0.34, 4.39]	.766
Coding Experience (≥ 3 months=1)	1.58	[0.50, 4.99]	.437	1.46	[0.44, 4.86]	.539
GPT Use Frequency (≥ 5 days/week=1)	-	-	-	1.28	[0.35, 4.72]	.707

Note. OR=odds ratio; CI=95% confidence interval. Model 1 includes Group, Major, and Coding; Model 2 adds GPT use frequency to Model 1 (sensitivity model). Coding: Group(AI-First=1), Major(STEM=1), Coding(≥ 3 months=1), GPT(≥ 5 days/week=1).

2) 설계 효능감(H2)

설계 효능감(H2)은 사전-사후 반복측정 자료이므로 집단(전통/AI-First(6+1))과 시점(사전/사후), 그리고 집단 \times 시점 상호작용을 포함한 모형으로 처치 효과를 검정하였다. 분석 결과, 설계 효능감의 시점 주효과가 유의하였고($F(1,58)=54.12$, $p<.001$, $\eta^2=.48$), 핵심 관심 효과인 집단 \times 시점 상호작용도 유의하였다($F(1,58)=17.36$, $p<.001$, $\eta^2=.23$). 이는 두 집단 모두 설계 효능감이 향상되었으나 AI-First(6+1) 집단의 향상 폭이 대조군보다 더 컸음을 의미한다(표 5). 추가로 사후 설계 효능감을 종속변수로 하고 사전 점수를 공변량으로 통제한 ANCOVA에서도 집단 효과가 유의하였다

($F(1,57)=19.02, p<.001, \eta^2=.25$). 따라서 AI-First(6+ 1) 운영 프로그램 집단은 대조군에 비해 사후 설계 효능감이 유의하게 높게 나타났으며, H2는 지지되었다.

3) 인지부하(H3)

인지부하(H3) 역시 사전-사후 반복측정 자료이므로 동일한 방식으로 처치 효과를 검증하였다. 분석 결과, 인지부하의 시점 주효과가 유의하여($F(1,58)=32.40, p<.001, \eta^2=.36$) 사후 인지부하가 사전보다 전반적으로 감소하는 경향이 확인되었고, 집단×시점 상호작용도 유의하였다($F(1,58)=18.92, p<.001, \eta^2=.25$). 이는 두 집단 모두 인지부하가 감소하였으나 AI-First(6+ 1) 집단의 감소 폭이 대조군보다 더 컸음을 의미한다(표 5). 또한 사후 인지부하를 종속변수로 하고 사전 점수를 공변량으로 통제된 ANCOVA에서도 집단 효과가 유의하였다($F(1,57)=16.11, p<.001, \eta^2=.22$). 따라서 AI-First(6+ 1) 운영 프로그램 집단은 대조군에 비해 사후 인지부하가 유의하게 낮게 나타났으며, H3는 지지되었다.

표 8. 설계 효능감(H2) 및 인지부하(H3) 처치 효과(집단×시점 상호작용) 결과

Table 8. Treatment effects for design self-efficacy (H2) and cognitive load (H3): Time × group interaction

Outcome	Effect	F(df1, df2)	p	η^2
Design Self-Efficacy	Time	$F(1,58)=54.12$	<.001	.48
	Time × Group	$F(1,58)=17.36$	<.001	.23
Cognitive Load (NASA-TLX)	Time	$F(1,58)=32.40$	<.001	.36
	Time × Group	$F(1,58)=18.92$	<.001	.25

4-4 조절효과 분석

본 절에서는 AI-First(6+ 1) 운영 프로그램의 효과가 학습자 배경 특성에 따라 달라지는지 검토하기 위해 전공 계열(H4)과 사전 코딩 경험(H5)의 조절효과를 분석하였다. 설계 효능감과 인지부하(주관적 작업부하; NASA-TLX)는 반복측정 분산분석 모형에서 집단×시점×조절변수(3원 상호작용)를 중심으로 검증하였고, 프로젝트 완수율(Pass/Fail)은 이항로지스틱 회귀모형에 집단×조절변수 상호작용항을 포함하여 검증하였다. 유의한 조절효과가 확인된 경우에 한해 하위집단별 단순효과를 추가로 제시하였다(표 9).

1) 전공 계열의 조절효과(H4)

전공 계열(STEM/non-STEM)이 AI-First(6+ 1) 운영 프로그램의 효과를 조절하는지 확인하기 위해 집단×시점×전공의 3원 상호작용을 검증하였다. 그 결과, 설계 효능감의 3원 상호작용은 유의하지 않았고($F(1,56)=0.64, p=.427, \eta^2=.01$), 인지부하(주관적 작업부하) 역시 유의하지 않았다($F(1,56)=0.51, p=.479, \eta^2=.01$). 프로젝트 완수율에 대해서도 로지스틱 회귀모형에서 집단×전공 상호작용항은 유의

하지 않았다($OR=1.12, 95\% CI [0.25, 5.09], p=.887$). 즉, AI-First(6+ 1) 운영 프로그램의 효과는 전공 계열에 따라 통계적으로 유의한 차이를 보이지 않았으며, H4는 지지되지 않았다(표 9).

표 9. 조절효과(상호작용) 분석 결과

Table 9. Moderation analysis: Interaction effects

Moderator	Outcome	Interaction term	F(df1, df2) / OR	p	Effect size/ 95% CI
Academic Major	Design Self-Efficacy	Time×Group ×Major	$F(1,56)=0.64$.427	$\eta^2=.01$
	Subjective workload (NASA-TLX)	Time×Group ×Major	$F(1,56)=0.51$.479	$\eta^2=.01$
	Project Completion	Group×Major	$OR=1.12$.887	95% CI [0.25, 5.09]
Coding Experience	Design Self-Efficacy	Time×Group ×CodingExp	$F(1,56)=5.21$.026	$\eta^2=.09$
	Subjective workload (NASA-TLX)	Time×Group ×CodingExp	$F(1,56)=4.78$.033	$\eta^2=.08$
	Project Completion	Group×CodingExp	$OR=0.74$.678	95% CI [0.18, 3.02]

2) 사전 코딩 경험의 조절효과(H5)

사전 코딩 경험(<3개월/≥3개월)이 AI-First(6+ 1) 운영 프로그램의 효과를 조절하는지 확인하기 위해 집단×시점×코딩경험의 3원 상호작용을 검증하였다. 분석 결과, 설계 효능감에서는 3원 상호작용이 유의하였다($F(1,56)=5.21, p=.026, \eta^2=.09$). 인지부하(주관적 작업부하)에서도 3원 상호작용이 유의하게 나타났다($F(1,56)=4.78, p=.033, \eta^2=.08$). 반면 프로젝트 완수율에 대한 로지스틱 회귀모형에서 집단×코딩경험 상호작용항은 유의하지 않았다($OR=0.74, 95\% CI [0.18, 3.02], p=.678$)(표 8).

유의한 조절효과가 확인된 설계 효능감과 인지부하에 대해서는 하위집단별 단순효과를 추가로 검토하였다. 코딩 경험이 3개월 미만인 하위집단에서는 설계 효능감의 집단×시점 상호작용이 유의하여($F(1,41)=18.74, p<.001, \eta^2=.31$), AI-First(6+ 1) 집단의 증가 폭($\Delta=사후-사전$)이 대조군보다 크게 나타났다(대조군 $\Delta=0.30$, AI-First(6+ 1) $\Delta=1.00$). 인지부하(주관적 작업부하) 역시 3개월 미만 하위집단에서 집단×시점 상호작용이 유의하였고($F(1,41)=16.02, p<.001, \eta^2=.28$), AI-First(6+ 1) 집단의 감소 폭이 대조군보다 크게 나타났다(대조군 $\Delta=-0.20$, AI-First(6+ 1) $\Delta=-0.90$). 반면 코딩 경험이 3개월 이상인 하위집단에서는 설계 효능감($F(1,15)=1.12, p=.307$)과 인지부하($F(1,15)=0.84, p=.373$) 모두에서 집단×시점 상호작용이 유의하지 않았다.

종합하면, 사전 코딩 경험은 AI-First(6+ 1) 운영 프로그램의 효과를 설계 효능감과 인지부하(주관적 작업부하) 측면에서 조절하는 것으로 나타났으며, 특히 코딩 경험이 낮은(입

문자) 하위집단에서 변화 폭이 더 크게 관찰되었다. 다만 본 연구의 처치는 AI-First 운영 원칙 적용과 1개월 추가 프로젝트 수행이 결합된 복합 처치(6+1개월)이므로, 조절효과의 해석은 해당 프로그램 적용 범위 내에서 제한되어야 한다. 또한 하위집단 분석은 표본 수가 제한적이므로 결과 일반화에는 신중할 필요가 있다. 이에 따라 H5는 설계 효능감과 인지부하 측면에서 부분적으로 지지되었고, 완수율에서는 지지되지 않았다.

V. 결론 및 논의

본 연구는 직업능력개발훈련의 실무형 웹개발 프로젝트 맥락에서 AI-First 운영원칙을 적용한 1개월 추가 프로젝트 운영(6+1), 즉 AI-First(6+1) 운영 프로그램(이하 본 프로그램)이 비전공 학습자의 프로젝트 수행 성과와 학습 관련 심리요인에 미치는 영향을 검증하였다. 본 프로그램은 생성형 AI 사용 여부의 단순 비교가 아니라, 과업 단계의 구조화(요구사항 분석-설계-구현-디버깅-검증), 단계별 AI 활용 규칙, 체크리스트-테스트 기반 산출물 검증을 추가 1개월 프로젝트에 통합한 운영이다. 따라서 본 연구의 결과는 6+1 운영에 대한 프로그램 수준의 효과로 해석하며, 구성요소별 단독 효과로 분리해 단정하지 않는다.

가설 검증 결과, 실험군은 대조군 대비 프로젝트 완수율이 유의하게 높아 H1이 지지되었다. 설계 효능감은 실험군의 향상 폭이 더 크게 나타났고, 인지부하(주관적 작업부하; NASA-TLX)는 실험군의 감소 폭이 더 크게 관찰되어 H2와 H3도 지지되었다. 조절효과 분석에서는 전공 계열(STEM/non-STEM)의 조절효과가 유의하지 않아 H4는 지지되지 않았으나, 사전 코딩 경험은 설계 효능감과 인지부하에서 조절효과가 확인되어 H5는 부분적으로 지지되었다. 다만 조절효과는 표본 규모와 검정력, 다중 검정 가능성을 고려할 때 탐색적으로 해석하는 것이 타당하다.

이러한 결과는 생성형 AI의 교육적 효과가 “도구 사용” 자체보다 규칙·절차·검증을 포함한 운영의 구조화를 통해 강화될 수 있음을 시사한다. 완수율 향상은 단계화된 수행과 산출물 검증(체크리스트-테스트-리뷰)이 실무형 산출물의 완성으로 연결될 수 있음을 보여주며, 생성형 AI 활용의 학습 성과 개선 가능성을 보고한 선행연구[1],[10],[11]의 논의와도 맥락을 같이한다. 설계 효능감 향상은 수행 기준의 제시와 피드백의 가시화가 자기신념을 강화했을 가능성으로 설명할 수 있으며[9], 인지부하 감소는 표준화된 루틴이 불필요한 탐색과 시행착오를 줄여 부담을 완화했을 가능성으로 해석된다[8],[18].

조절효과 결과는 적용 전략 측면의 시사점을 제공한다. 전공 계열 조절효과의 비유는 본 표본에서 효과 차이가 확인되지 않았음을 의미하며, 더 큰 표본과 동시기 설계에서 재검

증이 필요하다. 반면 사전 코딩 경험의 조절효과는 사전 경험이 낮은 학습자에게 구조화된 비계의 효과가 상대적으로 크게 나타날 수 있음을 시사한다[14]. 후속 연구에서는 운영원칙과 추가 학습시간(또는 추가 프로젝트)을 분리한 요인설계와 로그-커밋-테스트 기록 등 과정 데이터 기반 분석을 통해 효과 메커니즘을 추가로 검증할 필요가 있다.

VI. 제 언

첫째, 직업훈련 현장에서 생성형 AI 도입은 “AI 사용 허용”을 넘어 단계별 운영 가이드라인으로 구조화할 필요가 있다. 단계별 질문 구조·산출물 형식·검증 기준을 명시하고, 요구사항 명세·구조도/흐름도·API 정의·오류 로그(재현 절차 포함)·테스트 결과·체크리스트 점검 기록을 핵심 산출물로 요구하는 운영이 수행 안정성과 학습 부담 완화에 기여할 수 있다. 핵심은 AI 사용 자체가 아니라 검증 가능한 산출물 생성과 검증-수정 루틴의 반복이다.

둘째, 초보 학습자에게는 “검증의 내면화”를 핵심 학습목표로 설정할 필요가 있다. AI 응답을 즉시 채택하기보다 실행·시연·테스트로 확인하고 수정하는 루틴을 반복하게 하며, 오류 로그 정리와 재현 절차 문서화, 체크리스트 기반 점검을 수업 운영의 기본 단위로 삼는 전략이 유효하다.

셋째, 평가 체계를 고도화할 필요가 있다. 완수율(Pass/Fail)뿐 아니라 코드 품질(예외 처리, 책임 분리, 중복 제거), 테스트 충족률, 문서화(README), 유지보수성 등 질적 지표를 단계적으로 포함하고, 특히 검증 기록의 충실도(테스트 설계의 적절성, 재현 절차의 명료성)를 평가 요소로 반영하는 방안을 고려할 수 있다.

넷째, 내적 타당도를 강화해야 한다. 서로 다른 기수 비교는 역사효과를 완전히 배제하기 어렵다. 동일 기수 내 무작위 배정 또는 매칭, 동시 운영 조건에서의 처치 구분을 검토할 필요가 있다.

다섯째, 과정 기반 데이터로 메커니즘을 검증할 필요가 있다. 프롬프트 로그, 오류 로그, 커밋 이력, 테스트 수행 기록 등 과정 데이터는 문제 해결 행동 변화가 성과·효능감·부담으로 연결되는 경로를 보다 설득력 있게 설명한다. 또한 통제집단의 AI 사용 가능성을 고려하여 AI 사용의 양과 방식(질문 유형, 검증 행동, 테스트 빈도 등)을 함께 추적할 필요가 있다.

부 록

표 10. 주요 변인별 설문 측정 지표

Table 10. Survey items and measurement indicators for key variables

Category	Survey items (measurement indicators)	Ref.
Design self-efficacy	DSE1. I can break down requirements into functional units and plan an implementation strategy.	[14]-[17]
	DSE2. I can explain and design the linkage among service components (frontend, backend, data/DB, external APIs).	
	DSE3. I can design and document data flow and API call flow.	
	DSE4. I can incorporate authentication/authorization requirements into my design.	
	DSE5. When errors occur, I can infer plausible causes and design a direction for fixing them.	
	DSE6. I can justify design decisions with evidence and propose alternatives when needed.	
Cognitive load (subjective workload)	TLX1 (Mental demand). How mentally demanding was the project task?	[10], [18], [19]
	TLX2 (Physical demand). How physically demanding was the project task?	
	TLX3 (Temporal demand). How much time pressure did you feel during the project task?	
	TLX4 (Performance difficulty). How difficult was it to achieve the project goals? (higher = more difficult)	
	TLX5 (Effort). How much effort did you need to complete the project task?	
	TLX6 (Frustration). How much frustration/stress did you experience during the project task?	
Manipulation check	MC1. I followed a stage-based workflow (Requirements-Design-Implementation-Debugging-Verification).	This study
	MC2. I structured prompts using a template (goal-constraints-input/output-exceptions) before asking the AI.	
	MC3. When errors occurred, I organized logs and reproduction steps before asking the AI for causes/fixes.	
	MC4. I did not blindly accept AI outputs; I tested/verified and revised them before applying.	
	MC5. I used a checklist and tests (including scenarios) to verify outputs.	
Moderators	MA1 (Academic major; measurement). <input type="checkbox"/> Humanities/Social Sciences <input type="checkbox"/> Arts/Physical Education <input type="checkbox"/> STEM (Engineering/Natural Sciences/CS/IT) <input type="checkbox"/> Other (____) → Analysis coding: STEM = 1; Non-STEM (all others) = 0	This study
	CE1 (Prior coding experience; measurement). <input type="checkbox"/> <3 months <input type="checkbox"/> 3-6 months <input type="checkbox"/> ≥6 months → Analysis coding: <3 months = 0; ≥3 months = 1 (merge 3-6 and ≥6)	

표 11. 프로젝트 완수율(Pass/Fail) 판정 기준 및 체크리스트

Table 11. Project completion (pass/fail) criteria and checklist

1. Pass/fail criteria		
A submission was classified as Pass when all required features (R1-R10) were marked "Yes" on the checklist and the system was executable and demonstrable without critical failures; otherwise, it was classified as Fail.		
2. Required-feature checklist		
ID	Required item	Yes/No
R1	Application runs and main page is accessible (no critical errors)	<input type="checkbox"/> Y <input type="checkbox"/> N
R2	Login and authorization work (protected access controlled)	<input type="checkbox"/> Y <input type="checkbox"/> N
R3	Core data CRUD works (Create/Read/Update/Delete)	<input type="checkbox"/> Y <input type="checkbox"/> N
R4	Frontend-backend API integration works for main flows	<input type="checkbox"/> Y <input type="checkbox"/> N
R5	DB persistence is verified (save/retrieve consistency)	<input type="checkbox"/> Y <input type="checkbox"/> N
R6	Comment feature works (create + retrieve/list)	<input type="checkbox"/> Y <input type="checkbox"/> N
R7	Comment recommendation works end-to-end (request → response → UI reflection)	<input type="checkbox"/> Y <input type="checkbox"/> N
R8	Spring Boot-Flask integration works (request/response verified)	<input type="checkbox"/> Y <input type="checkbox"/> N
R9	Error handling & validation are implemented (incl. integration failure case)	<input type="checkbox"/> Y <input type="checkbox"/> N
R10	End-to-end demo completed + basic documentation provided (README/run steps)	<input type="checkbox"/> Y <input type="checkbox"/> N
Final Decision: <input type="checkbox"/> Pass <input type="checkbox"/> Fail		
Notes (optional): _____		
3 Inter-rater Procedure		
Two raters independently judged Pass/Fail using the checklist; disagreements were resolved by consensus. Rater-specific evaluation sheets and evidence logs are available from the authors upon request.		

참고문헌

[1] J. Wang and W. Fan, "The Effect of ChatGPT on Students' Learning Performance, Learning Perception, and Higher-Order Thinking: Insights from a Meta-Analysis," *Humanities and Social Sciences Communications*, Vol. 12, 621, 2025. <https://doi.org/10.1057/s41599-025-04787-y>

[2] P. Suchanek and M. Kralova, "Generative Artificial Intelligence Expectations and Experiences in Management Education: ChatGPT Use and Student Satisfaction," *Journal of Innovation & Knowledge*, Vol. 10, No. 5, 100781, 2025. <https://doi.org/10.1016/j.jik.2025.100781>

[3] S. Rudra, P. Weber, T. Tillmanns, A. Salomão Filho, E. Wiersma, J. Dawitz, ... and S. Reynolds, "Use and Perception of Generative AI in Higher Education: Insights from the ERASMUS+ Project 'Teaching and Learning with Artificial Intelligence' (TaLAI)," in *Proceedings of the 17th*

- International Conference on Computer Supported Education (CSEDU 2025)*, Vol. 2, pp. 319-330, 2025. <https://doi.org/10.5220/0013360200003932>
- [4] S. S. Oyelere and K. Aruleba, "A Comparative Study of Student Perceptions on Generative AI in Programming Education Across Sub-Saharan Africa," *Computers and Education Open*, Vol. 8, 100245, 2025. <https://doi.org/10.1016/j.caeo.2025.100245>
- [5] C. Watson and F. W. B. Li, "Failure Rates in Introductory Programming Revisited," in *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education (ITiCSE '14)*, Uppsala, Sweden, pp. 39-44, 2014. <https://doi.org/10.1145/2591708.2591749>
- [6] A. Diržytė, A. Perminas, L. Kaminskis, G. Žebrauskas, Ž. Sederevičiūtė-Pačiauskienė, J. Šliogerienė, ... and I. Gajdosikienė, "Factors Contributing to Dropping Out of Adults' Programming E-Learning," *Heliyon*, Vol. 9, No. 12, e22113, 2023. <https://doi.org/10.1016/j.heliyon.2023.e22113>
- [7] J. Park and J. Kim, "A Qualitative Study of Bootcamp Perceptions and Dropout Prevention Factors: Focus on Stakeholders," *The Journal of Korean Association of Computer Education*, Vol. 27, No. 1, pp. 1-23, 2024. <https://doi.org/10.32431/kace.2024.27.1.001>
- [8] J. Sweller, "Cognitive Load During Problem Solving: Effects on Learning," *Cognitive Science*, Vol. 12, No. 2, pp. 257-285, 1988. https://doi.org/10.1207/s15516709cog1202_4
- [9] A. Bandura, *Self-Efficacy: The Exercise of Control*, New York, NY: W. H. Freeman, 1997.
- [10] G. Gutiérrez Carreón and R. López Escalera, "Evaluating Usability and Cognitive Load in Programming Education with ChatGPT Integration," in *Proceedings of the 6th International Computer Programming Education Conference (ICPEC 2025)*, Vol. 133, pp. 12:1-12:7, 2025. <https://doi.org/10.4230/OASlcs.ICPEC.2025.12>
- [11] G. Fan, D. Liu, R. Zhang, and L. Pan, "The Impact of AI-Assisted Pair Programming on Student Motivation, Programming Anxiety, Collaborative Learning, and Programming Performance: A Comparative Study with Traditional Pair Programming and Individual Approaches," *International Journal of STEM Education*, Vol. 12, 16, 2025. <https://doi.org/10.1186/s40594-025-00537-3>
- [12] F. G. W. C. Paas, "Training Strategies for Attaining Transfer of Problem-Solving Skill in Statistics: A Cognitive-Load Approach," *Journal of Educational Psychology*, Vol. 84, No. 4, pp. 429-434, 1992. <https://doi.org/10.1037/0022-0663.84.4.429>
- [13] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "Pair Programming Improves Student Retention, Confidence, and Program Quality," *Communications of the ACM*, Vol. 49, No. 8, pp. 90-95, 2006. <https://doi.org/10.1145/1145287.1145293>
- [14] D. Wood, J. S. Bruner, and G. Ross, "The Role of Tutoring in Problem Solving," *Journal of Child Psychology and Psychiatry*, Vol. 17, No. 2, pp. 89-100, 1976. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- [15] D. R. Compeau and C. A. Higgins, "Computer Self-Efficacy: Development of a Measure and Initial Test," *MIS Quarterly*, Vol. 19, No. 2, pp. 189-211, 1995. <https://doi.org/10.2307/249688>
- [16] V. Ramalingam and S. Wiedenbeck, "Development and Validation of Scores on a Computer Programming Self-Efficacy Scale," *Journal of Educational Computing Research*, Vol. 19, No. 4, pp. 367-381, 1998. <https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P>
- [17] A. R. Carberry, H.-S. Lee, and M. W. Ohland, "Measuring Engineering Design Self-Efficacy," *Journal of Engineering Education*, Vol. 99, No. 1, pp. 71-79, 2010. <https://doi.org/10.1002/j.2168-9830.2010.tb01043.x>
- [18] F. G. W. C. Paas, J. J. G. van Merriënboer, and J. J. Adam, "Measurement of Cognitive Load in Instructional Research," *Perceptual and Motor Skills*, Vol. 79, No. 1, pp. 419-430, 1994. <https://doi.org/10.2466/pms.1994.79.1.419>
- [19] S. G. Hart and L. E. Staveland, "Development of NASA-TLX: Results of Empirical and Theoretical Research," in *Human Mental Workload, Advances in Psychology*, Vol. 52, Amsterdam, the Netherlands: North-Holland, pp. 139-183, 1988. [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)

유순복(Soon-Bog Yoo)



2026년 : 한국기술교육대학교
IT융합과환경영산업대학원
AI융합교육학과 석사과정

2020년~현 재: 소프트웨어 개발 및 인공지능 교육 강사

2023년~현 재: 인공지능 기반 풀스택 개발자

※관심분야: 머신러닝·딥러닝 모델, 생성형 인공지능



배현영 (Hyun-Young Bae)

2024년 : 한국기술교육대학교
IT융합과학경영산업대학원
(공학석사 - AI융합교육학과)
2026년 : 서강대학교 가상융합전문대학
원 박사수료

2025년~현재 : 한국기술교육대학교
IT융합과학경영산업대학원 AI융합교육학과
겸임교수

※ 관심분야 : AI, 머신러닝, 메타버스, 디지털트윈 등