

컴퓨터교육학회 논문지 2026년 제29권 제5호
<https://doi.org/10.32431/kace.2026.29.5.006>



바이브코딩(Vibe Coding)을 활용한 교육용 웹 앱 개발 연구 : 설계 기반 연구 방법론 적용*

Developing Educational Web Applications through Generative AI-Based Vibe Coding: Applying Design-Based Research Methodology

윤명희[†] · 박수홍^{††}
 Myunghee Yoon[†] · Suhong Park^{††}

요약

본 연구는 2025년 Andrej Karpathy가 제안한 바이브코딩(Vibe Coding) 접근법을 활용하여 비전공자가 교육용 웹 앱을 개발할 수 있는 가능성을 탐색하고, 그 과정과 결과를 보고한다. 생성형 AI 기반 개발 도구인 Lovable, Claude Code, Supabase를 활용하여 지도 기반 협업 도구 '핀 캔버스(Pin Canvas)'를 설계기반연구(Design-Based Research, DBR) 방법론에 따라 개발하였다. 개발된 앱은 커스텀 캔버스 배경 설정, 핀 배치 및 관리, 레이어 시스템, 드로잉 도구, 실시간 협업 등의 기능을 갖추었으며, 중학교 현장에 적용하여 교육적 활용 가능성을 탐색하였다. 교사 인터뷰 분석 결과, 핀 캔버스는 기존 상용 도구로 충족되지 않았던 교육적 요구를 반영한 맞춤형 도구로서 수업 활동의 효과와 효율을 높이는 데 기여한 것으로 나타났다. 본 연구는 바이브코딩이 소프트웨어 개발의 진입 장벽을 낮추고, 교사가 교육 현장의 필요에 맞춘 맞춤형 도구를 직접 개발할 수 있는 '교육 기술 개발의 민주화'를 실현할 수 있음을 보여준다. 이는 AI 기반 소프트웨어 개발 도구의 교육적 활용 가능성과 비전공자 대상 SW 교육에 시사점을 제공한다.

주제어 바이브코딩, 생성형 AI, 설계기반연구, 교육용 앱 개발, 노코드/로우코드, 소프트웨어 교육

ABSTRACT

This study explores the possibility of non-programmers developing educational web applications using the Vibe Coding approach proposed by Andrej Karpathy in 2025, and reports on the process and outcomes. Using generative AI-based development tools including Lovable, Claude Code, and Supabase, a map-based collaboration tool called 'Pin Canvas' was developed following the Design-Based Research (DBR) methodology. The developed application features custom canvas background settings, pin placement and management, a layer system, drawing tools, and real-time collaboration capabilities, and its educational applicability was explored through implementation in a middle school setting. Teacher interview analysis indicated that Pin Canvas, as a customized tool reflecting educational needs unmet by existing commercial tools, teachers reported that it contributed to enhancing the effectiveness and efficiency of classroom activities. This study demonstrates that Vibe Coding can lower the entry barriers to software development and enable teachers to directly develop customized tools tailored to the needs of educational settings, thereby realizing the 'democratization of educational technology development.' This provides implications for the educational application of AI-based software development tools and software education for non-programmers.

Keywords Vibe Coding, Generative AI, Design-Based Research, Educational App Development, No-Code/Low-Code, Software Education

†중신회원 부산대학교 대학원 교육과미디어융합
전공 박사수로
 ††정회원 부산대학교 사범대학 교육학과 교수(교
신자)
 논문투고 2025년 12월 22일
 심사완료 2026년 03월 30일
 게재확정 2026년 04월 03일
 발행일자 2026년 05월 20일

* 이 논문은 2021년 대한민국 교육부와 한국연구재단의
지원을 받아 수행된 연구임
 (NRF-2021S1A5C2A03086965)

1. 서론

디지털 기술의 급속한 발전은 교육 현장에 새로운 가능성과 도전을 동시에 제시하고 있다. 특히 2022 개정 교육과정의 도입과 함께 학교자율시간의 확대는 각 학교가 지역의 특성과 학생의 필요에 맞춘 교육과정을 설계하고 운영할 수 있는 여지를 넓혔다[1]. 그러나 이러한 자율성의 확대는 동시에 교육과정 운영에 필요한 적절한 교육 기술 도구의 부재라는 현실적 문제를 야기하기도 한다.

전통적으로 교육용 소프트웨어의 개발은 전문 개발자의 영역으로 여겨져 왔으며, 교사들은 기존에 개발된 도구 중에서 선택하거나 적응해야 하는 수동적 위치에 있었다. 이러한 상황에서 2025년 2월, OpenAI의 공동 창업자이자 Tesla의 전 AI 책임자인 Andrej Karpathy가 제안한 '바이브코딩(Vibe Coding)'이라는 새로운 개념은 소프트웨어 개발의 패러다임에 근본적인 변화를 예고하였다.

Karpathy[2]는 바이브코딩을 "완전히 감(vibes)에 몸을 맡기고, 기하급수적 성장을 받아들이며, 코드가 존재한다는 사실 자체를 잊는" 새로운 코딩 방식으로 정의하였다. 이 접근법에서 개발자는 대형 언어 모델(Large Language Model, LLM)에게 자연어로 원하는 기능을 설명하고, AI가 생성한 코드를 검토하지 않고도 기능적인 소프트웨어를 구축할 수 있다. 바이브코딩은 2025년 Collins Dictionary의 '올해의 단어'로 선정되었으며[3], Y Combinator의 2025년 겨울 배치 스타트업 중 25%가 95% 이상 AI로 생성된 코드베이스를 보유하고 있는 것으로 보고되었다[4].

바이브코딩의 등장은 교육 기술 개발의 '민주화(democratization)'를 촉진할 수 있는 잠재력을 가진다. Gadde[5]는 생성형 AI가 자연어 표현과 기능적 소프트웨어 생성 사이의 간극을 연결함으로써 접근성 패러다임을 근본적으로 변화시킬 수 있다고 주장하였다. 이는 교사가 현장의 요구에 맞춘 맞춤형 교육 도구를 직접 개발할 수 있는 가능성을 열어준다.

본 연구는 울산광역시 울주군 S중학교의 '지역사회와 계시민' 교과 운영을 지원하기 위해 바이브코딩 접근법을 활용하여 '핀 캔버스(Pin Canvas)' 앱을 개발한 과정과 결과를 보고한다. 연구의 목적은 첫째, 바이브코딩을 활용한 교육용 앱 개발의 실제 과정을 설계기반연구(Design-Based Research, DBR) 방법론에 따라 체계적으로 기술하고, 둘째, 이를 통해 교육 기술 개발의 민주화 가능성과 시사점을 탐색하는 것이다.

2. 이론적 배경

2.1 바이브코딩(Vibe Coding)

바이브코딩은 2025년 2월 Andrej Karpathy가 소셜 미디어 플랫폼 X(구 Twitter)에 게시한 글에서 처음 소개된 개념이다. Karpathy[2]는 "LLM(예: Cursor Composer와

Sonnet)이 너무 좋아져서 가능해진 새로운 종류의 코딩"이라고 설명하며, 자신이 "키보드를 거의 만지지 않고" 음성으로 AI에게 지시하여 코드를 생성하는 방식으로 작업한다고 밝혔다. 그는 "항상 '모두 수락'을 누르고, 더 이상 diff를 읽지 않으며, 오류 메시지가 나오면 코멘트 없이 복사하여 붙여넣으면 대개 문제가 해결된다"고 설명하였다.

Sarkar와 Gordon[6]은 바이브코딩에 대한 최초의 실증 연구를 수행하여 8시간 이상의 바이브코딩 세션 영상을 분석하였다. 연구 결과, 바이브코딩은 개발자가 AI에게 프롬프트를 제공하고, 생성된 코드를 빠르게 스캔하며, 애플리케이션 테스트를 통해 평가하고, 수동으로 편집하는 반복적 목표 충족 주기를 따르는 것으로 나타났다. 이들은 바이브코딩이 프로그래밍 전문성의 필요를 없애는 것이 아니라, 맥락 관리, 빠른 코드 평가, AI 기반과 수동 코드 조작 사이의 전환 시점 결정 등으로 재분배한다고 주장하였다.

Ge 등[7]은 1,000개 이상의 연구 논문을 분석한 종합 서베이를 통해 바이브코딩 생태계 전반을 검토하였다. 이들은 바이브코딩을 제약된 마르코프 결정 과정(Constrained Markov Decision Process)을 통해 형식화하고, 무제한 자동화, 반복적 대화 협업, 계획 주도, 테스트 주도, 맥락 강화 모델 등 다섯 가지 개발 모델로 분류하였다. 바이브코딩에 활용되는 주요 도구로는 Cursor, GitHub Copilot, Claude Code, Lovable, Replit 등이 있으며, 이러한 도구들은 Claude, OpenAI의 Codex, Gemini, LLaMA 3 등의 AI 모델을 기반으로 한다.

그러나 바이브코딩에 대한 비판적 시각도 존재한다. Fawzy 등[8]은 101개의 실무자 출처에서 518개의 행동 사례를 분석한 체계적 문헌 검토를 통해, 바이브코더들이 속도도와 접근성에 의해 동기부여되어 빠른 '즉각적 성공과 몰입'을 경험하지만, 대부분이 품질 보증(QA) 관행을 간과하고 테스트를 건너뛰거나 AI 도구의 출력을 수정 없이 의존한다고 보고하였다. 또한 AI 과학자 Andrew Ng는 2025년 5월 LangChain Interrupt 컨퍼런스에서 바이브코딩이라는 용어가 소프트웨어 엔지니어들이 "그냥 감에 따라" 작업한다고 오해하게 만들 수 있다고 지적하며, AI를 활용한 코딩은 실제로 "깊이 있는 지적 작업(deeply intellectual exercise)"이라고 강조하였다[9]. 이러한 우려에도 불구하고, Karpathy[2] 자신도 바이브코딩이 "버릴 수 있는 주말 프로젝트"에 적합하다고 언급하였으며, 교육용 프로토타입이나 소규모 도구 개발에는 효과적인 접근법이 될 수 있다.

2.2 설계기반연구(Design-Based Research)

설계기반연구(Design-Based Research, DBR)는 21세기 초에 발전한 연구 방법론으로, 연구와 실제 사이의 간극을 효과적으로 연결할 수 있는 실용적 방법론으로 주목받았다[10]. Brown[11]과 Collins[12]가 처음 제안한 이 방법론은 복잡한 교육 환경에서 개입(intervention)을 설계, 개발, 평가하는 체계적 접근법을 제공한다. 이후 The Design-Based Research Collective[13]는 DBR을 교육적 탐구의

새로운 패러다임으로 제시하며 연구자와 실천가의 협력적 파트너십을 강조하였고, Barab과 Squire[14]는 맥락 의존적 이론 개발의 중요성을 역설하며 DBR의 방법론적 기초를 정립하였다. 국내에서도 강정찬과 이상수[15]가 DBR을 수업 개선을 위한 현장연구방법으로 소개하며 그 개념과 특징, 설계원리 및 연구모형을 체계적으로 정리하였다.

DBR의 핵심 특징은 반복적 설계(iterative design), 맥락적 연구(contextual research), 이론과 실제의 통합에 있다. Wang과 Hannafin[16]은 DBR이 테크놀로지 강화 학습 환경의 설계와 연구에 특히 적합하다고 주장하였다. 이는 DBR이 실제 교육 문제를 해결하면서 동시에 이론적 원리를 도출할 수 있는 유연한 프레임워크를 제공하기 때문이다. McKenney와 Reeves[17]는 DBR의 주요 단계를 분석/탐색, 설계/구축, 평가/성찰의 반복적 순환으로 제시하였다.

최근 DBR은 교육공학 연구에서 널리 활용되고 있다. 김민정[18]은 DBR 방법론을 활용하여 대규모 수업을 위한 모바일 앱을 개발하고 적용한 사례를 보고하였으며, 강지혜와 박태정[19]은 2010년부터 2021년까지의 국내 설계·개발 연구 및 설계기반연구 86편을 분석하여, DBR이 교육공학 분야에서 다양한 교수학습 산출물 개발에 활발히 활용되고 있음을 확인하였다. 김성원과 이영준[20]은 설계기반연구를 통해 프로그래밍 기반 TPACK 교육 프로그램을 개발하여, DBR이 교사의 테크놀로지 통합 역량 개발에 효과적으로 적용될 수 있음을 보여주었다. Hoadley와 Campos[21]은 온라인 학습 환경에서 DBR이 교육적, 심리학적 연구 목표와 설계 노력의 균형을 맞추는 복잡한 도전을 해결하는데 독특하게 적합하다고 설명하였다. McHugh, Carroll, Connolly[22]는 로우코드/노코드(Low-Code/No-Code, LCNC) 플랫폼을 활용한 교육 도구 개발에 DBR을 적용한 연구에서 교사들이 디지털 기술을 향상시키는 데 개방적이지만, 기술에 대한 두려움이나 변화에 대한 저항이 장벽이 될 수 있음을 발견하였다. 이러한 연구들은 DBR이 새로운 교육 기술의 개발과 평가에 효과적인 방법론임을 입증한다.

2.3 교육 기술 개발의 민주화와 비전공자 SW 교육

교육 기술 개발의 민주화는 전통적으로 전문 개발자의 영역이었던 소프트웨어 개발을 비전문가도 접근할 수 있게 만드는 움직임을 의미한다. Reeves[23]는 교육공학 연구자들이 단순히 방법 A가 방법 B보다 나은지를 비교하는 연구 대신, 학습 환경의 실제적 문제에 대한 최적의 해결책을 개발하는 데 집중해야 한다고 주장하였다. 이러한 관점에서 로우코드/노코드 플랫폼과 AI 지원 개발 도구의 등장은 중요한 의미를 가진다.

국내에서도 비전공자를 위한 소프트웨어 교육에 대한 연구가 활발히 진행되어 왔다. 김은수와 박준석[24]은 앱 저작 도구를 활용한 교육용 앱 개발 연구를 통해 비전공자도 저작도구를 활용하여 교육용 앱을 개발할 수 있음을 보여주었다. 이명숙[25]은 생성형 AI를 활용한 소프트웨어 교육 수업 모델 연구에서 ChatGPT를 활용하여 비전공자 대학생들이

파이썬 프로그래밍을 학습하는 사례를 보고하였다. 이러한 연구들은 AI 기반 도구가 SW 교육의 진입 장벽을 낮출 수 있음을 시사한다.

Gambacorta 등[26]은 Ant Group의 CodeFuse를 활용한 현장 실험을 통해 LLM 기반 코딩 도구가 프로그래머의 생산성을 55% 향상시킨다는 것을 발견하였다. 특히 주목할 점은 생산성 향상이 주니어 개발자에게서 통계적으로 유의미하게 나타났으며, 시니어 개발자에 대한 영향은 상대적으로 적었다는 것이다. 이는 AI 지원 개발 도구가 경험미적은 개발자나 비전공자에게 더 큰 혜택을 제공할 수 있음을 시사한다. McHugh, Carroll, Connolly[22]의 연구에서도 시민 개발(Citizen Development) 접근법이 교사와 학생들에게 컴퓨터 과학의 측면을 소개하면서도 기술적 역량 개발에 대한 부담을 줄일 수 있는 기회를 제공한다고 보고하였고, Gadde[5]는 생성형 AI가 자연어 표현과 기능적 소프트웨어 생성 사이의 간극을 연결함으로써 접근성 패러다임을 근본적으로 변화시킬 수 있다고 주장하였다. 이는 교사가 현장의 요구에 맞춘 맞춤형 교육 도구를 직접 개발할 수 있는 가능성을 열어준다.

2.4 바이브코딩과 기존 개발 접근법의 비교

바이브코딩의 교육적 의미를 이해하기 위해서는 기존의 노코드(No-Code) 및 로우코드(Low-Code) 접근법과의 차별성을 명확히 할 필요가 있다. 노코드 플랫폼(예: Padlet, Google Sites)은 사전 설계된 템플릿과 드래그 앤 드롭 방식으로 소프트웨어를 구성하며, 로우코드 플랫폼(예: AppSheet, Glide)은 최소한의 코딩과 시각적 개발 환경을 결합한다. 바이브코딩은 이들과 본질적으로 다른 접근법으로, 자연어 대화를 통해 AI가 완전한 소스 코드를 생성하는 방식이다. 세 접근법의 주요 차이점은 Table 1과 같다.

Table 1. Comparison of No-Code, Low-Code, and Vibe Coding Approaches

Dimension	No-Code	Low-Code	Vibe Coding
Interaction Method	GUI drag-and-drop, template selection	Visual editor + partial code writing	Natural language prompts to AI
Customization Flexibility	Limited to pre-built components	Moderate; extensible with custom code	High; AI generates custom source code without template constraints
Output	Platform-dependent application	Platform-hosted or exportable app	Full source code (React, TypeScript, etc.)
Technical Barrier	Lowest; no coding knowledge required	Low-moderate; basic coding helpful	Low; requires prompt engineering and basic debugging

Dimension	No-Code	Low-Code	Vibe Coding
Iterative Modification	Constrained to platform's editing options	Possible within platform framework	Rapid via conversational prompts; code-level changes
Platform Dependency	High; vendor lock-in	Moderate; partial lock-in	Low; source code is portable
Risk / Limitation	Feature ceiling; cannot exceed platform capabilities	Complexity grows with customization	AI hallucination; code quality variability; debugging difficulty
Representative Tools	Padlet, Google Sites, Canva	AppSheet, Glide, PowerApps	Lovable, Cursor, Claude Code

교육적 맥락에서 바이브코딩의 핵심적 차별성은 세 가지이다. 첫째, 커스터마이징의 자유도이다. 노코드 플랫폼은 사전 설계된 기능 범위 내에서만 활용이 가능하지만, 바이브코딩은 교사의 자연어 요구사항을 AI가 완전한 소스 코드로 변환하므로 기존 도구로 충족할 수 없는 맥락 특화 기능의 구현이 가능하다. 본 연구에서 교사들이 요구한 6가지 핵심 기능을 단일 노코드 도구로 충족시킬 수 없었던 반면, 바이브코딩은 이를 통합적으로 구현할 수 있었다.

둘째, 반복적 수정의 신속성이다. DBR의 핵심 원리인 반복적 정교화를 실현하기 위해서는 사용자 피드백을 신속하게 반영할 수 있어야 한다. 노코드 플랫폼에서의 수정은 플랫폼이 허용하는 범위로 제한되지만, 바이브코딩에서는 자연어 프롬프트를 통해 코드 수준의 수정이 가능하므로 교사 피드백을 당일 또는 수일 내에 반영할 수 있었다.

셋째, 소스 코드의 소유와 이식성이다. 노코드 플랫폼은 벤더 종속(vendor lock-in)의 위험이 있어 서비스 중단 시 교육 도구의 지속성이 위협받는다. 바이브코딩은 완전한 소스 코드를 생성하므로 다른 환경으로의 이식이 가능하며, 이는 교육 기술의 장기적 지속가능성 측면에서 중요한 이점이다.

다만, 바이브코딩은 AI 환각(hallucination)으로 인한 코드 오류, 복잡한 기능에서의 예측 불가능한 동작, 비전문가의 디버깅 어려움 등의 한계를 가진다. 본 연구에서도 좌표 스케일링 문제, 드로잉 데이터 저장 실패 등의 기술적 문제가 발생하였으며, 이는 4.1절에서 상세히 기술한다.

3. 연구 방법

3.1 연구 설계 및 맥락

본 연구는 설계기반연구(DBR) 방법론을 적용하여 수행되었다. McKenney와 Reeves[17]의 DBR 모델에 따라 분석/탐색, 설계/구축, 평가/성찰의 반복적 단계를 거쳤다. 연구는 2025학년도 5월부터 12월까지 진행되었으며, 교사 인터뷰를 통한 요구 분석, 기존 도구 검토, 바이브코딩을 활용한 앱 개발, 실제 수업 적용 및 피드백 수집의 과정을 포함

하였다. 이 연구는 울산형 혁신학교인 서로나눔학교인 울산광역시 울주군 S중학교에서 진행되었다. S중학교는 2022년부터 덴마크 소재 학교와의 국제교류를 시작으로 세계시민교육의 가능성을 탐색해왔으며 2025학년도 2학기부터 교육감 승인 과목인 '지역사회와 세계시민'을 개설하여, 중학교 1학년을 대상으로 주당 2시간씩 운영하게 되었다. 이 교과는 2022 개정 교육과정의 학교자율시간을 활용한 것으로, 학생들이 지역에 대한 정보를 수집, 정리, 공유하면서 지역과 세계를 연결하는 학습을 수행하도록 설계되었다.

3.2 연구 참여자 및 데이터 수집

3.2.1 연구 참여자

본 연구의 참여자는 S중학교에서 '지역사회와 세계시민' 교과를 공동 운영한 교사 2인이다. 교사 A(국어 담당, 교직 경력 21년)는 해당 교과의 주 수업 담당자로서 핀 캔버스의 주요 피드백 제공자 역할을 수행하였으며, 교사 B(영어 담당, 교직 경력 11년)는 공동 수업 운영자로서 수업 적용 과정에서 피드백을 제공하였다. 두 교사 모두 프로그래밍 경험이 없는 비전문가이며, 서면 인터뷰 및 온라인 의사소통 기록을 통해 연구 데이터를 직접 제공하였다.

수업 적용 대상은 중학교 1학년 2개 학급 46명으로, 2025학년도 2학기 '지역사회와 세계시민' 교과 수강 학생 전원이다. 학생들은 4~5인 모둠으로 구성되어 8차시에 걸쳐 핀 캔버스를 활용한 수업에 참여하였다. 본 연구에서 교사와 학생의 역할은 구분된다. 교사 2인은 도구의 요구사항 제시, 수업 적용, 피드백 제공 등 DBR의 설계-평가 과정에 직접 참여한 연구 참여자(research participants)이다. 반면, 학생 46명은 개발된 도구를 활용한 수업의 학습자로서 교육적 개입(intervention)의 대상이며, 본 연구에서는 학생을 대상으로 한 직접적인 데이터 수집(사용성 평가, 학습 효과 측정 등)은 수행하지 않았다. 학생의 활동 양상 및 반응은 수업을 직접 운영한 교사의 관찰 보고를 통해 간접적으로 파악하였다. 이는 본 연구가 바이브코딩 기반 교육용 도구의 개발 과정과 교사의 활용 경험을 중심으로 한 DBR의 초기 사이클에 해당하며, 학습자 대상의 직접적 효과 검증은 후속 사이클의 과제로 설정하였기 때문이다.

본 연구는 연구자가 앱 개발자이자 연구자의 이중 역할을 수행하는 참여적 DBR의 특성을 가진다. 연구자의 이중 역할에 따른 잠재적 편향을 최소화하기 위해 다음과 같은 전략을 적용하였다. 첫째, 교사 인터뷰는 서면 형식으로 실시하여 연구자의 실시간 반응이 응답에 영향을 미치는 것을 방지하였다. 둘째, 인터뷰 질문은 개방형으로 구성하여 특정 방향의 응답을 유도하지 않도록 하였다. 셋째, 교사 인터뷰 결과, 온라인 의사소통 기록, 연구자 성찰일지의 세 가지 데이터 출처를 교차 비교하는 삼각검증을 수행하였다. 본 연구의 삼각검증은 연구자로부터 완전히 독립된 외부 데이터를 포함하는 것이 아니라, 동일 현상에 대해 서로 다른 관점과 시점의 데이터를 교차 확인하는 방법적 삼각검증

(methodological triangulation)의 성격을 가진다. 교사 인터뷰는 교육적 활용 경험에 대한 교사의 해석적 관점을, 온라인 의사소통 기록은 실시간으로 발생한 사건의 시간순 과정 데이터를, 연구자 성찰일지는 기술적 의사결정의 맥락과 근거를 각각 제공하며, 이를 교차 비교함으로써 특정 출처에 의존한 해석의 편향을 방지하였다. 넷째, 교사의 부정적 피드백(초기 버그에 대한 불만, 이중 작업의 불편함 등)을 연구 결과에 그대로 포함하여 긍정적 결과만을 선별적으로 보고하지 않았다.

3.2.2 데이터 수집 방법

DBR의 다중 관점 평가 원리에 따라 세 가지 출처의 데이터를 수집하였다.

첫째, 교사 인터뷰이다. 수업 종료 후 참여 교사 2인을 대상으로 서면 인터뷰와 후속 확인 인터뷰의 2단계로 실시하였다. 1단계에서는 교육적 활용 맥락, 학생 반응 및 교육적 효과(교사 관찰 기반), DBR 과정 경험, 바이브코딩 인식의 4개 영역에 걸쳐 12개 질문으로 구성된 반구조화된 서면 인터뷰를 실시하였으며, 교사 2인이 협의를 거쳐 공동으로 응답하였다. 2단계에서는 서면 응답의 내용을 확인·보완하기 위해 온라인 메신저를 통한 후속 인터뷰를 실시하였다. 두 교사는 동일 교과를 공동 운영하였으나, 핀 캔버스의 실제 수업 활용은 주로 교사 A가 담당하였으며 교사 B는 보조적 역할을 수행하였다. 이에 서면 인터뷰는 주 사용자인 교사 A의 경험을 중심으로, 교사 B가 이를 보완·확인하는 협의적 응답(collaborative response) 형태로 진행하였다. 이는 공동 수업 운영의 맥락에서 두 교사의 관찰이 상호보완적 성격을 가지며, 교사 B의 참여가 교사 A 응답의 확인(member checking) 기능을 수행한다는 판단에 기반한다.

둘째, 교사-연구자 간 온라인 의사소통 기록이다. 2025년 5월부터 12월까지 연구자와 교사 2인 사이에 메신저 플랫폼을 활용한 텍스트 기반 의사소통을 수행하였으며, 이 기록을 수집하였다. 온라인 의사소통 기록에는 기능 요청, 오류 보고, 수정 확인, 수업 중 관찰 사항 등이 포함되어 있으며, 피드백-수정 사이클의 과정 데이터(process data)로 활용되었다. 이러한 비동기적 텍스트 기반 의사소통은 수업 시간 중에도 즉각적인 문제 보고가 가능하다는 점에서 DBR의 현장 밀착적 연구에 적합한 데이터 수집 방식이다.

셋째, 연구자 성찰일지이다. 개발 전 과정에 걸쳐 연구자가 작성한 개발 일지를 수집하였다. 기술적 의사결정, AI 도구 활용 전략, 프롬프트 실패 사례와 대응, 교사 피드백에 대한 해석 등을 포함한다.

3.2.3 데이터 분석 방법

교사 인터뷰 데이터는 Braun과 Clarke[27]의 주제 분석(thematic analysis) 절차에 따라 분석하였다. 서면 인터뷰 응답 원문과 후속 인터뷰의 텍스트 기록을 통합하여 반복적으로 읽으며 초기 코드를 생성하고, 코드를 주제로 범주화한

뒤, 주제를 검토 및 명명하였다. 온라인 의사소통 기록은 시간순으로 정리하여 피드백-수정 사건(event)을 추출하고, 각 사건의 보고 일자, 문제 내용, 해결 일자, 소요 기간을 코딩하였다. 분석의 신뢰성 확보를 위해 세 가지 데이터 출처를 교차 비교하는 방법적 삼각검증(methodological triangulation)을 수행하였다. Carter 등[28]은 Denzin(1978)과 Patton(1999)이 제시한 삼각검증 유형을 정리하면서, 방법적 삼각검증이 동일한 현상에 대해 서로 다른 데이터 수집 방법을 적용하여 결과의 수렴(convergence), 상호보완(complementarity), 또는 차이(divergence)를 확인하는 전략이라고 설명하였다. 본 연구에서는 이에 따라 교사 인터뷰가 교육적 활용 경험에 대한 교사의 해석적 관점을, 온라인 의사소통 기록이 실시간으로 발생한 사건의 시간순 과정 데이터를, 연구자 성찰일지가 기술적 의사결정의 맥락과 근거를 각각 제공하는 것으로 설정하고, 주제 분석을 통해 도출된 각 주제(theme)에 대해 세 출처의 데이터가 해당 주제를 지지하는지, 보완적 정보를 제공하는지, 또는 상이한 관점을 보이는지를 체계적으로 검토하였다.

3.3 요구 및 기존 도구 분석

앱 개발에 앞서 교과 담당 교사들과의 심층 인터뷰를 통해 요구사항을 분석하였다. 인터뷰는 반구조화된 형식으로 진행되었으며, 현재 사용 중인 도구의 한계, 필요한 기능, 학생 활동 시나리오 등을 탐색하였다. 인터뷰 결과를 바탕으로 다음과 같은 핵심 기능 요구사항이 도출되었다.

첫째, 핀 배치 기능이다. 학생들이 지역의 특정 장소에 대한 정보를 디지털 지도 위에 핀으로 표시하고, 각 핀에 텍스트, 이미지, 링크 등의 콘텐츠를 첨부할 수 있어야 한다. 둘째, 캔버스 배경 커스텀 설정 기능이다. 일반적인 지도 외에도 학교에서 제작한 지역 지도, 역사 지도, 개념도 등을 배경으로 활용할 수 있어야 한다. 셋째, 레이어 기능이다. 각 항목별(예: 자연환경, 역사문화, 산업경제 등)로 핀을 분류하여 별도의 레이어로 관리할 수 있어야 한다. 넷째, 드로잉 기능이다. 지리적 요소(강, 산맥, 경계선 등)를 시각적으로 표시할 수 있는 그리기 도구가 필요하다. 다섯째, 공유 및 협업 기능이다. Padlet과 같이 로그인하지 않은 사용자에게도 공유하여 편집이 가능하도록 해야 한다. 여섯째, 핀 모양 커스텀 설정 기능이다. 각 카테고리나 항목에 맞게 핀의 모양이나 색상을 다르게 설정할 수 있어야 한다.

Table 2. Comparison of Existing Tools and Pin Canvas

Feature Requirement	Padlet	Google My Maps	Miro	Pin Canvas
Custom Canvas Background	×	×	○	●
Pin Placement on Map	△	●	△	●
Custom Pin Icons	×	○	×	●
Layer System	×	○	×	●
Drawing Tools	△	○	●	●
Non-Login Sharing	●	△	○	●
Real-Time Collaboration	●	×	●	●

(● Full support, ○ Partial, △ Limited, × Not supported)

Table 2.에서 확인되듯이, Padlet은 비로그인 공유와 실시간 협업에 강점이 있으나 지리적 맥락에서의 핀 배치와 레이어 기능이 부재하다. Google My Maps는 지도 기반 핀 배치에 특화되어 있으나 커스텀 캔버스 배경 설정이 불가능하고 실시간 협업이 지원되지 않는다. Miro는 무한 캔버스와 드로잉 도구를 제공하지만 지도 기반 작업에 최적화되어 있지 않다. 이러한 분석을 통해 교사들의 요구사항을 통합적으로 충족하는 새로운 도구의 개발이 필요하다는 결론에 도달하였으며, 이것이 바이브코딩을 활용한 핀 캔버스 개발의 직접적 동기가 되었다.

3.4 개발 도구 및 기술 스택

핀 캔버스의 개발은 바이브코딩 접근법을 활용하여 진행되었다. 개발에 사용된 주요 도구와 기술 스택은 Table 3.과 같다.

Table 3. Technology Stack Used for Pin Canvas Development

Category	Technology / Tool	Role
Front-end	React, TypeScript, Tailwind CSS	User interface implementation
Back-end / DB	Supabase (PostgreSQL)	Data storage, real-time synchronization, user authentication
AI Dev Tools	Lovable, Claude Code	Natural language-based code generation and modification
Deployment	Lovable Publish	One-click web hosting and deployment

전체 아키텍처 설계와 프론트엔드 개발에는 Lovable을 활용하였다. Lovable은 자연어 프롬프트를 기반으로 React 애플리케이션을 생성하는 AI 기반 개발 플랫폼으로, 전체적인 앱 구조와 사용자 인터페이스를 신속하게 프로토타이핑할 수 있게 해주었다. 데이터베이스는 Supabase를 활용하였다. Supabase는 오픈소스 Firebase 대안으로, PostgreSQL 기반의 실시간 데이터베이스와 사용자 인증 기능을 제공한다. 세부적인 기능 수정과 버그 수정에는 Claude Code를 활용하였다. Claude Code는 Anthropic의 Claude 모델을 기반으로 한 코드 생성 및 수정 도구로,

복잡한 기능 구현이나 에러 해결에 효과적이었다.

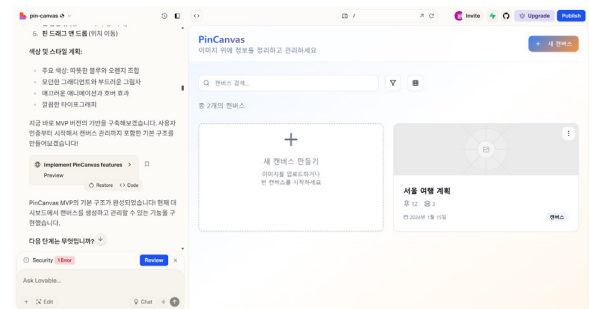
4. 연구 결과

4.1 DBR 반복 사이클에 따른 개발 과정

본 연구는 McKenney와 Reeves[17]의 DBR 모델에 따라 2회의 반복 사이클을 수행하였다. 각 사이클은 분석/탐색, 설계/구축, 평가/성찰의 세 단계로 구성되며, 사이클 간 전환 시 이전 사이클의 평가 결과가 다음 사이클의 분석 단계에 반영되는 순환적 구조를 따른다.

첫 번째 사이클(2025년 5월~9월)은 초기 프로토타입 개발을 목표로 수행되었다. 분석과 탐색 단계에서는 교과 담당 교사 2인과의 반구조화 인터뷰를 통해 6가지 핵심 기능 요구사항을 도출하고, 기존 도구(Padlet, Google My Maps, Miro)의 한계를 분석하였다. 교사들의 요구를 통합적으로 지원하는 단일 도구가 부재함을 확인하고, 바이브코딩을 활용한 신규 도구 개발을 결정하였다.

설계 및 구축 단계에서는 요구사항을 PRD(Product Requirement Document)로 문서화하고, 교육공학 전문가의 자문을 거쳐 내용 적절성을 검토하였다. Lovable을 활용하여 자연어 프롬프트 기반으로 React 애플리케이션의 기본 구조를 생성하고, Supabase를 연동하여 실시간 데이터 동기화와 비로그인 공유 기능을 구현하였다. Claude Code를 활용하여 레이어 필터링 로직, 커스텀 핀 아이콘 설정, 드로잉 좌표 변환 알고리즘 등 세부 기능을 개발하였다. 초기 프로토타입은 약 3주 만에 작동 가능한 상태로 완성되었다.

**Figure 1.** Generating the Basic App Structure Using Lovable

평가/성찰 단계에서는 2025년 2학기 수업 적용을 통해 실제 사용 맥락에서 프로토타입을 평가하였다. 교사들은 온라인 의사소통을 통해 수업 중 발견된 문제를 즉시 보고하였으며, 수업 적용 초기(9월)에 세 가지 주요 문제가 확인되었다. 가장 심각한 문제는 좌표 스케일링 오류로, 디바이스 화면 크기에 따라 핀의 위치가 달라지는 현상이었다. 교사 A는 "학생들이 PC에서 찍은 핀이 태블릿에서 보면 위치가 다르다"고 보고하였다(온라인 의사소통 기록, 2025.09.05). 또한 학생들이 그린 드로잉이 저장되지 않거나 새로고침 시

사라지는 데이터 저장 실패 문제가 발생하였으며(온라인 의사소통 기록, 2025.09.12), 모바일 환경에서 핀치 줌 인터랙션이 원활하지 않은 화면 조작 문제도 보고되었다. 이러한 문제들은 사이클 2의 분석 단계로 환류되어 개선 과제로 설정되었다.

이러한 기술적 문제들은 단순한 소프트웨어 결함을 넘어 교육적 맥락에서의 설계 과제를 드러내었다. 좌표 스케일링 오류는 교실에서 PC, 크롬북, 태블릿이 혼용되는 다중 디바이스 환경을 사전에 고려하지 못한 결과이며, 드로잉 데이터 저장 실패는 학생 학습 결과물의 영속성 확보라는 교육적 요구를 간과한 것이었다. 또한 핀치 줌 문제는 학생들이 실제로 사용하는 디바이스의 인터랙션 방식과 개발 환경 간의 괴리에서 비롯된 것으로, AI가 생성한 코드가 특정 환경에서만 테스트되는 경향이 있음을 보여주었다. 이처럼 사이클 1의 평가 결과는 기술적 수정 과제인 동시에, 바이브코딩 기반 교육용 앱 개발에서 고려해야 할 설계 원리의 필요성을 시사하였다.

이를 바탕으로 사이클 1에서 다음과 같은 잠정적 설계원리를 도출하였다. 첫째, 교육 현장에서 사용되는 다양한 디바이스에서 일관되게 작동하도록 설계해야 한다(DP3 초안). 둘째, 학습 결과물이 실시간으로 자동 저장되어 데이터 손실을 방지해야 한다(DP5 초안). 셋째, AI가 생성한 코드의 오류 가능성에 대비하여 버전 관리와 대안 전략을 사전에 준비해야 한다(DP7 초안). 이러한 잠정적 원리들은 사이클 2의 설계/구축 단계에서 구체적인 기술적 해결책으로 구현되었으며, 수업 적용을 통한 추가 평가를 거쳐 최종 설계원리로 정교화되었다.

두 번째 사이클(2025년 10월~12월)은 사이클 1에서 도출된 문제의 반복적 개선을 목표로 수행되었다. 분석/탐색 단계에서는 3가지 핵심 문제를 심각도와 수업 영향도를 기준으로 우선순위를 설정하였다. 좌표 스케일링 문제는 핀 배치라는 앱의 핵심 기능의 정확성에 직접 영향을 미치므로 가장 높은 우선순위로 분류되었다.

설계/구축 단계에서는 사이클 1에서 도출된 잠정적 설계원리를 기술적으로 구현하는 것을 목표로 하였다. DP3 초안(디바이스 독립적 설계)에 대응하여 Claude Code를 활용하여 좌표 시스템을 절대 좌표에서 상대 좌표(비율 기반)로 재설계하였고, DP5 초안(영속적 데이터)에 대응하여 드로잉 데이터의 실시간 자동 저장을 위한 Supabase 기반 메커니즘을 구현하였으며, 화면 조작 UX를 개선하였다. 이 과정에서 AI가 생성한 좌표 변환 코드가 기존 기능을 손상시키는 오류가 2건 발생하여 버전 롤백이 필요하였으며, 이는 DP7 초안(AI 오류 대비 설계)의 필요성을 실증적으로 확인시켜 주었다. 피드백 접수부터 수정 반영까지의 소요 시간은 평균 1~3일이었으며, 이는 바이브코딩의 자연어 프롬프트 기반 수정이 신속한 반복을 가능하게 한 결과이다.

평가/성찰 단계에서는 개선된 기능에 대한 교사 확인, 잔여 과제 도출, 그리고 설계원리의 정교화를 수행하였다. 교사들은 좌표 문제의 해결을 확인하였으며(온라인 의사소통 기록, 2025.10.08), 수업이 진행됨에 따라 기술적 문제 보고

가 감소하는 양상을 보였다. 3.2.3절에서 기술한 바와 같이 온라인 의사소통 기록에서 피드백-수정 사건을 추출하여 월별로 분류한 결과, 기술적 문제 보고 건수는 8월 2건, 9월 3건, 10월 1건, 11월 0건, 12월 0건으로, 사이클 2의 설계/구축 단계에서 핵심 문제가 해결된 이후 뚜렷한 감소 추세를 보였다. 이는 사이클 1에서 도출된 잠정적 설계원리에 기반한 기술적 개선이 실제 교육 현장에서 효과적으로 작동하였음을 시사한다. 학기 종료 후 실시한 교사 인터뷰를 통해 앱의 교육적 활용 경험과 잔여 개선 과제를 확인하였다. 동시 접속 시 로딩 지연, 일부 브라우저 호환성 문제 등은 후속 사이클의 개선 과제로 남겨두었다.

사이클 2의 평가 결과를 종합하여 잠정적 설계원리를 검증·정교화하는 동시에 추가적인 원리를 도출하였다. 사이클 1에서 도출된 DP3(디바이스 독립적 설계), DP5(영속적 데이터 아키텍처), DP7(AI 오류 대비 설계)은 사이클 2에서의 기술적 구현과 수업 적용을 통해 그 필요성이 실증적으로 확인되어 최종 설계원리로 확정되었다. 특히 DP3은 사이클 1에서 '다양한 디바이스에서 작동해야 한다'는 일반적 수준이었으나, 사이클 2에서 상대 좌표 시스템의 구현 경험을 거치며 'AI에게 코드 생성을 요청할 때 다중 디바이스 호환성을 프롬프트에 명시적으로 포함해야 한다'는 바이브코딩 맥락의 구체적인 지침으로 정교화되었다. 또한 8차시에 걸친 수업 적용과 교사 인터뷰 분석을 통해, 사이클 1에서는 명시적으로 인식되지 않았던 새로운 원리들이 추가로 도출되었다. 교사들의 요구가 곧 개발 프롬프트의 기초가 된다는 경험에서 DP1(교육자 주도 요구 분석)이, 메신저 기반의 즉각적 피드백-수정 순환 경험에서 DP2(신속한 피드백 루프)가, Lovable과 Claude Code의 상호 보완적 활용 경험에서 DP4(다중 AI 도구 전략)가, 46명의 비로그인 동시 접속 경험에서 DP6(낮은 진입 장벽의 협업)이 각각 도출되었다. 이러한 설계원리의 점진적 정교화 과정은 4.4절에서 상세히 기술한다.

두 사이클의 주요 활동과 결과를 정리하면 Table 4와 같다.

Table 4. Summary of DBR Iterative Cycles

Cycle	Phase	Key Activities	Outcomes
Cycle 1 (2025.05~09)	Analysis / Exploration	Teacher interviews; existing tool analysis	6 core requirements identified; no single tool meets all needs
	Design / Construction	PRD → Lovable + Supabase + Claude Code	Working prototype in ~3 weeks
	Evaluation / Reflection	Classroom use (Sep); teacher online communication feedback	3 critical issues identified; preliminary design principles (DP3, DP5, DP7 drafts) derived from educational analysis of issues

Cycle	Phase	Key Activities	Outcomes
Cycle 2 (2025.10~12)	Analysis / Exploration	Priority setting based on severity and classroom impact	Coordinate scaling = highest priority
	Design / Construction	Relative coordinate system; auto-save; UX improvements	Fixes deployed in 1~3 days per issue
	Evaluation / Reflection	Teacher confirmation; post-implementation teacher interviews	Preliminary DPs validated; 4 additional DPs derived (DP1, DP2, DP4, DP6); 7 final design principles established

4.2 핀 캔버스 주요 기능

개발된 핀 캔버스 앱의 주요 기능은 다음과 같다. 첫째, 커스텀 캔버스 기능이다(Figure 2). 사용자는 일반 지도 외에도 직접 제작한 이미지를 배경으로 설정할 수 있다. 이를 통해 지역의 역사 지도, 생태 지도, 개념도 등 다양한 형태의 배경을 활용할 수 있다.

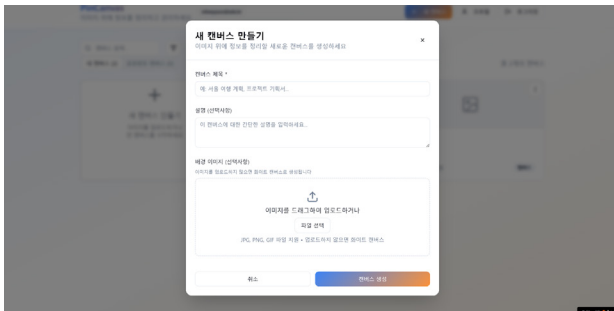


Figure 2. Custom Canvas Feature

둘째, 핀 배치 및 관리 기능이다(Figure 3). 캔버스의 원하는 위치를 클릭하여 핀을 추가하고, 각 핀에 제목, 설명, 이미지, 링크 등의 콘텐츠를 첨부할 수 있다. 핀의 모양과 색상도 커스터마이징이 가능하며, 커스텀 이미지를 첨부하여 사용자 맞춤형의 핀도 생성할 수 있다.

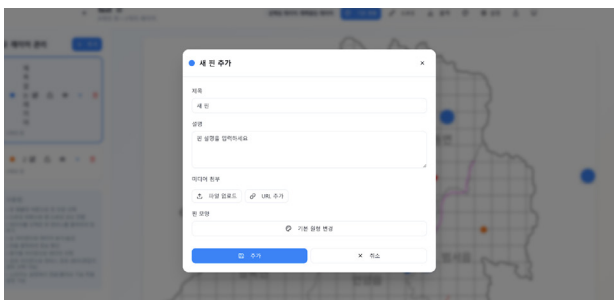


Figure 3. Pin Placement and Management Feature

셋째, 레이어 시스템이다(Figure 4). 여러 개의 레이어를 생성하여 핀을 카테고리별로 분류할 수 있다. 각 레이어의

표시/숨김을 토글하여 원하는 정보만 선택적으로 확인할 수 있다.

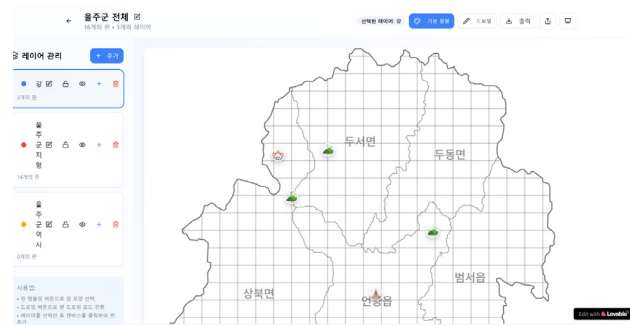


Figure 4. Layer System

넷째, 드로잉 도구이다(Figure 5). 자유 곡선, 직선, 도형 등을 그릴 수 있어 강, 경계선, 영역 등을 시각적으로 표현할 수 있다.

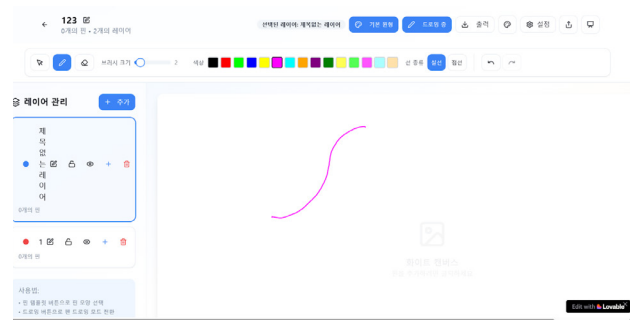


Figure 5. Drawing Tools

다섯째, 실시간 협업 및 공유 기능이다(Figure 6). 여러 사용자가 동시에 같은 캔버스에서 작업할 수 있으며, 변경 사항이 실시간으로 동기화되며, 고유한 공유 링크를 생성하여 비로그인 사용자도 캔버스를 보거나 편집할 수 있다.

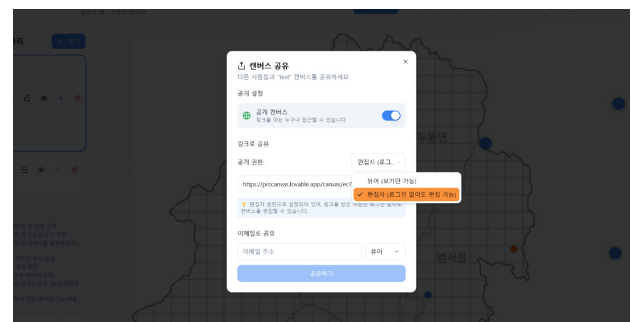


Figure 6. Real-Time Collaboration and Sharing Feature

4.3 수업 적용 결과

핀 캔버스는 2025학년도 2학기 S중학교의 '지역사회와 세계시민' 교과에서 1학년 2개 학급 46명을 대상으로 총 8차시(1-2차시 지역의 지형, 3-4차시 지역의 역사, 5-6차시 지

역의 산업, 7-8차시 지역의 교육과 문화)에 걸쳐 활용되었다. 수업 적용 이후 참여 교사 2인을 대상으로 서면 인터뷰 및 온라인 후속 인터뷰를 실시하였으며, 수집된 응답을 주제 분석(thematic analysis)을 통해 분석하였다. 분석 결과, 교육적 활용 맥락, 학생 반응 및 교육적 효과, DBR 과정 경험, 바이브코딩 인식의 4개 영역에서 다음과 같은 주제가 도출되었다.

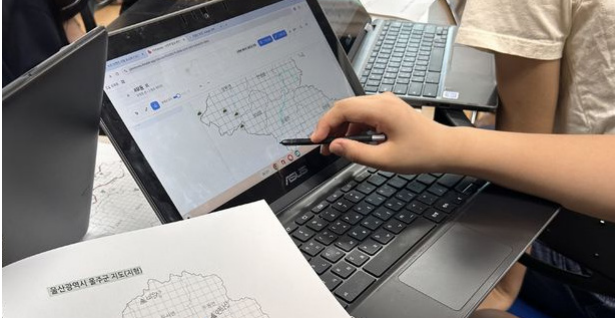


Figure 7. Actual Implementation in the Classroom

4.3.1 교육적 활용 맥락

교사들은 수업에서 핀 캔버스를 지역 탐구 결과의 누적적 기록 도구로 활용하였다. 구체적으로, 학생들은 지역의 백지도를 캔버스 배경으로 설정한 후 차시별 탐구 주제에 따라 해당 장소에 핀을 배치하고, 핀의 아이콘을 주제에 맞게 설정(예: 교육 기관은 '학교' 아이콘)한 뒤 설명을 추가하는 방식으로 작업하였다. 또한 차시별로 레이어를 구분하여 특정 주제(예: 지형 탐구 결과)만 선택적으로 표시하거나 숨길 수 있도록 활용하였다. 학생들은 4~5인 모둠으로 구성되어 실시간 협업 기능을 통해 동시에 작업하였으며, 모둠별로 핀 배치와 드로잉 기능을 자율적으로 선택하여 활용하였다.

교사들은 핀 캔버스 도입 이전에 Google My Maps와 Padlet을 검토하였으나, Google My Maps는 핀에 부가 설명이나 커스텀 아이콘을 첨부하기 어렵고, Padlet은 지도 기반 공동 작업에 한계가 있어 교과와 요구를 충족시키지 못하였다. 이에 비해 핀 캔버스는 커스텀 배경 설정, 핀 아이콘 커스터마이징, 레이어 분리, 드로잉, 실시간 협업 등 교사가 요청한 기능이 모두 구현되어 있어, 교사들은 "핀 캔버스의 특정 기능 한 가지만 유용하다고 말하기 어렵다"고 평가하였다.

4.3.2 학생 반응 및 교육적 효과

교사 관찰에 따르면, 학생들은 핀 캔버스의 사용법을 익히는 데 큰 어려움을 느끼지 않았으며 새로운 도구에 대한 두려움 없이 접근하였다. 다만 초기 버전에서 발생한 기술적 문제(지도 확대 시 작업물의 좌표 불일치, 작업 내용 미저장 등)로 인해 불편을 호소한 경우가 있었다. 이에 대해 교사들은 인쇄물 백지도를 병행 제공하여, 종이에 먼저 작업한 후 핀 캔버스로 옮기는 방식으로 대처하였다. 교사들

은 이 과정이 이중 작업이라는 점에서 아쉬웠으나, "한 번 작업한 것을 검토하고 디지털화하는 과정에서 정확한 위치를 다시 한번 확인하게 되었기 때문에 학습 목표 달성에 적절한 대응이었다"고 평가하였다.

버그 수정 이후 학생들의 활용 숙련도는 빠르게 향상되어 4~5차시경에는 능숙하게 사용할 수 있게 되었으며, 교사들은 현재 완성 버전으로 수업을 진행한다면 3차시만에도 능숙한 활용이 가능할 것으로 전망하였다. 문제 해결 이후에는 불편을 호소하는 경우 없이 수업 과제를 원활하게 수행하였다고 보고하였다. 특히 모둠별 실시간 협업 기능은 개인 과제로는 제시간에 완성하기 어려웠을 활동을 협업을 통해 완성도 높게 수행할 수 있게 하였으며, 교사들은 이 과정에서 학생들이 "협업의 소중함"을 인식하는 데 기여하였다고 보고하였다.

교사들은 후속 인터뷰에서 핀 캔버스의 교육적 기여에 대해 보다 구체적으로 언급하였다. 지역 백지도에 핀을 배치하는 활동을 통해 학생들의 지역에 대한 이해가 높아졌으며, 모둠원이 탐색 활동을 분담하고 학습 결과물을 함께 화면으로 확인하는 과정에서 모둠 활동의 효율성과 효과성이 향상되었다고 보고하였다. 또한 수업 주제에 따라 지형, 문화, 교육 등 레이어별로 활동 결과를 분리하여 정리할 수 있었던 점이 수업 효율성을 높이고 지역 탐색 활동을 효과적으로 마무리하는 데 기여하였다고 평가하였다. 교사들은 핀 캔버스가 없었다면 종이 지도를 모둠별로 배부하여 표기하는 아날로그 방식으로 수업을 진행해야 했을 것이며, 이 경우 다양한 레이어를 별도로 작업하거나 하나의 종이에 중첩하여 작성하는 한계가 있었을 것이라고 설명하였다. 아울러 교사들은 이 도구를 다음 학년도 수업에서도 지속적으로 사용하기를 희망한다고 밝혔다.

4.3.3 DBR 과정에 대한 교사 경험

교사들은 핀 캔버스의 개발 과정에서 온라인 메신저를 통해 개발자와 지속적으로 소통하였으며, 이 방식이 "가장 빠른 소통 방법"으로 적절하였다고 평가하였다. 피드백 제공 시 개발자가 거의 즉시 응답하고 다음 수업 시작 전까지 버그를 수정해 주었으며, 수업 중 발생한 문제도 화면 캡처를 통해 즉시 전달하였다. 개발자가 수정본을 제공하면 교사가 테스트한 후 다음 수업에 적용하는 반복적 개선 과정이 이루어졌다.

교사들은 이러한 피드백-수정 과정이 기존의 상용 교육용 도구에서는 경험하기 어려운 것이었다고 평가하였다. 기존 상용 앱의 경우 개선이나 기능 추가를 요청하면 "팀에서 검토한 후 연락하겠다"는 응답이 일반적이며, 버그 수정에도 상당한 시간이 소요되고 기능 추가 요청은 거부되는 경우도 있다고 보고하였다. 이에 비해 핀 캔버스 개발 과정에서는 교사의 요구가 직접적으로 반영되었으며, 교사들은 "내가 원하는 수업에 최적화된 앱을 사용할 수 있다는 것은 교사로서 행운이었다"고 표현하였다. 초기 버전과 최종 버전의 차이에 대해서도 "마지막 버전은 원하는 기능이 모두 추

가되어 제대로 작동하였다"며 높은 만족감을 표현하였다. 또한 공동 수업 교사들의 평가에서 핀 캔버스를 상용화하면 수요가 있을 것이라는 의견이 나온 점은 개발된 도구의 교육적 가치를 간접적으로 시사한다.

4.3.4 바이브코딩 기반 개발에 대한 인식

교사들은 바이브코딩으로 개발된 핀 캔버스의 가장 큰 장점으로 교사의 요구에 최적화된 맞춤형 도구라는 점을 꼽았다. 기존 도구를 '선택하여 사용하는' 방식과 '필요에 맞게 만드는' 방식의 차이에 대해, 교사들은 "선택하는 것은 나의 요구를 모두 충족하기 어렵고, 필요에 맞게 만들면 필요한 모든 기능을 활용할 수 있게 된다"고 설명하였다. 이를 통해 수업 활동과 결과물을 체계적으로 누적할 수 있게 되었으며, "학습 목표 달성의 효과와 효율을 높일 수 있었다"고 평가하였다.

한편, 바이브코딩 기반 도구의 한계로는 플랫폼 의존성에 따른 문제(일정 기간 미접속 시 로그인 오류 등)가 지적되었으며, 전문 개발자의 상용 앱에 비해 초기 버그가 많았다는 점도 언급되었다. 다만 교사들은 상용 앱과 달리 즉각적인 업데이트와 문제 해결이 가능했다는 점을 장점으로 평가하였다.

교사들은 교육용 도구 개발에서 개발자와 교사의 협업이 핵심적이라고 강조하면서도, 이러한 협업이 확산되기 위해서는 개발자와 교사 간 긴밀한 소통 구조가 전제되어야 하며, 에듀테크 코치와 같은 학교 내 지원 체계가 도입된다면 접근성이 높아질 것이라고 제안하였다. 또한 교사가 바이브코딩의 구조를 이해하면 개발자에게 요구사항을 정확하게 전달하는 데 유리하며, AI 기술의 발전과 교사의 행정 업무 경감이 이루어진다면 교사 스스로 바이브코딩을 활용한 도구 개발도 가능할 것이라는 전망을 제시하였다. 핀 캔버스의 활용 가능성에 대해서는 지역 탐구 기반 수업 외에도 국어과(전국 문학관 지도), 미술과(세계 미술관 지도) 등 다양한 교과로 확장 가능하며, "기본적 기능을 이해한다면 교사들이 활용하는 방법은 무궁무진하다"고 평가하였다.

4.3.5 분석 결과의 삼각검증

4.3.1~4.3.4에서 보고한 주제 분석 결과에 대해, 3.2.3절에서 기술한 방법적 삼각검증 절차에 따라 세 데이터 출처 간 교차 확인을 수행하였다. 그 결과, 주요 발견에서 수렴, 상호보완, 차이의 세 가지 양상이 확인되었다.

첫째, 바이브코딩 기반 개발의 신속한 반복적 개선(4.3.3)에 관한 발견에서 세 출처의 수렴이 확인되었다. 교사들은 인터뷰에서 "개발자가 거의 즉시 응답하고 다음 수업 시작 전까지 버그를 수정해 주었다"고 보고하였는데, 이는 온라인 의사소통 기록의 피드백-수정 사건 분석에서 확인된 평균 1~3일의 수정 소요 시간(좌표 스케일링: 5일, 드로잉 저장: 3일)과 일치하였다. 연구자 성찰일지에서도 동일 기간의 자연어 프롬프트 기반 수정 작업이 기록되어 있어,

세 출처가 동일한 현상을 서로 다른 관점에서 지지하였다.

둘째, 초기 기술적 문제로 인한 교육 활동의 제약(4.3.2)에 관한 발견에서 출처 간 상호보완이 확인되었다. 교사들은 인터뷰에서 "좌표 불일치로 인해 인쇄물 백지도를 병행하는 이중 작업이 필요했다"고 보고하였으며, 온라인 의사소통 기록에서는 이 문제의 최초 보고 일자(2025.09.05)와 해결 확인 일자(2025.10.08)가 확인되었고, 연구자 성찰일지에서는 절대좌표에서 상대좌표로의 재설계라는 기술적 해결 전략이 기록되어 있었다. 세 출처는 각각 교육적 영향, 사건의 시간적 경과, 기술적 해결 과정이라는 상호보완적 정보를 제공하였다.

셋째, 출처 간 차이도 확인되었다. AI가 생성한 좌표 변환 코드가 기존 기능을 손상시켜 버전 롤백이 필요했던 2건의 사례는 연구자 성찰일지에서만 확인되었으며, 교사 인터뷰와 온라인 의사소통 기록에서는 보고되지 않았다. 이는 교사가 인식하지 못한 기술적 맥락을 연구자 성찰일지가 보완한 것으로, 각 출처가 고유한 관점의 정보를 제공하고 있음을 보여준다. 이러한 삼각검증 결과는 본 연구의 발견이 특정 단일 출처에 의존한 것이 아니라 복수의 데이터 출처를 통해 확인되었음을 시사한다.

4.4 설계 원리 도출

DBR의 핵심 산출물은 특정 맥락에서 도출되지만 다른 맥락에도 전이 가능한 설계원리(design principles)이다 [17]. 본 연구에서 설계원리는 단일 시점에서 일괄 도출된 것이 아니라, 2회의 반복 사이클을 거치며 점진적으로 정교화되었다. 사이클 1의 수업 적용 평가에서 3개의 잠정적 원리(DP3, DP5, DP7의 초안)가 도출되었으며, 사이클 2에서 이를 기술적으로 구현·검증하는 동시에 추가적인 4개 원리(DP1, DP2, DP4, DP6)를 도출하여 총 7개의 최종 설계원리를 확정하였다. 이러한 과정은 McKenney와 Reeves[17]가 강조한 DBR의 반복적 정교화 원리에 부합한다. 최종 설계원리는 Table 5와 같다.

Table 5. Design Principles for Vibe Coding–Based Educational App Development

No.	Design Principle	Description
DP1	Educator–Driven Requirements	App development should begin with systematic needs analysis through teacher interviews, not developer assumptions.
DP2	Rapid Feedback Loops	Establish real-time communication channels between developer and teachers for immediate feedback during classroom use.
DP3	Device–Agnostic Design	Educational apps must function consistently across diverse devices (PC, tablet, smartphone) used in classrooms.
DP4	Multi-Tool AI Strategy	Use complementary AI tools for different development tasks: one for rapid prototyping, another for precise debugging.
DP5	Persistent Data Architecture	All user-created content (pins, drawings, layers) must be auto-saved in real-time to prevent data loss.
DP6	Low-Barrier Collaboration	Enable participation without login requirements to reduce access barriers in classroom settings.
DP7	Proactive AI Error Management	Anticipate AI hallucination and code errors by maintaining version control and preparing fallback strategies before they occur.

각 설계원리의 도출 근거는 다음과 같다. DP1(교육자 주도 요구 분석)은 교사들이 제시한 6가지 핵심 기능 요구가 Padlet, Google My Maps, Miro 등 기존 단일 도구로 충족될 수 없었다는 분석 결과에서 도출되었다. 교육용 앱 개발이 기술 중심이 아닌 현장 요구 중심으로 출발해야 효과적임을 보여준다.

DP2(신속한 피드백 루프)는 온라인 의사소통 기록에서 확인되는 피드백-수정 사이클의 실제 소요 시간에 근거한다. 좌표 스케일링 문제는 교사의 최초 보고(9월 5일)부터 수정 완료(9월 10일)까지 5일, 드로잉 저장 문제는 보고(9월 12일)부터 해결(9월 15일)까지 3일이 소요되었다. 이러한 신속한 반복은 바이브코딩의 자연어 프롬프트 기반 수정이 가능했기 때문이며, 노코드 플랫폼에서는 플랫폼 업데이트를 기다려야 하는 수정 사항도 자체적으로 해결할 수 있었다.

DP3(디바이스 독립적 설계)은 사이클 1에서 발견된 가장 심각한 문제인 좌표 스케일링 오류에서 비롯되었다. PC에서 배치한 핀이 태블릿에서 다른 위치에 표시되는 문제는 교실 환경에서 다양한 디바이스가 혼용되는 현실을 간과한 결과였으며, 상대 좌표 시스템으로의 전환을 통해 해결되었다.

DP4(다중 AI 도구 전략)는 개발 과정에서 단일 AI 도구의 한계를 경험한 데서 도출되었다. Lovable은 전체 앱 구조와 UI를 신속하게 생성하는 데 효과적이었으나, 레이어 필터링 로직이나 좌표 변환 알고리즘 등 세부 로직의 정밀한 구현에는 한계를 보였다. Claude Code를 보완적으로 활용함으로써 이러한 한계를 극복할 수 있었다.

DP5(영속적 데이터 아키텍처)는 사이클 1에서 드로잉 데이터가 저장되지 않아 학생들이 작업을 반복해야 했던 문제

에서 도출되었다. 학습 결과물의 손실은 학생들의 학습 동기와 수업 효율성에 직접적 영향을 미치며, 사이클 2에서 자동 저장 메커니즘을 구현하여 해결하였다.

DP6(낮은 진입 장벽의 협업)은 46명의 중학교 1학년 학생이 개별 계정 생성 없이 공유 링크만으로 앱에 접근하여 협업할 수 있었던 경험에 근거한다. 중학교 환경에서 개별 계정 생성은 시간 소모, 비밀번호 분실 등의 현실적 장벽을 수반하며, 비로그인 접근은 이를 효과적으로 우회하는 전략이었다.

DP7(AI 오류 대비 설계)은 AI가 생성한 좌표 변환 코드에서 발생한 오류 경험에서 도출되었다. 2건의 사례에서 AI가 생성한 코드가 기존 기능을 손상시켜 버전 롤백이 필요하였다. 이는 Fawzy 등[8]이 지적한 바이브코딩의 품질 보증 문제와 일치하며, 버전 관리와 대안적 프롬프팅 전략의 사전 준비가 필요함을 시사한다.

이러한 설계원리 중 DP2(신속한 피드백 루프)와 DP4(다중 AI 도구 전략)는 바이브코딩 특유의 원리로, 전통적 개발 방법론이나 노코드 접근법에서는 도출되기 어려운 것이다. 한편, 나머지 설계원리(DP1, DP3, DP5, DP6, DP7)는 교육용 앱 개발에 일반적으로 적용 가능한 원리이나, 바이브코딩 맥락에서 특수한 의미를 갖는다. DP1(교육자 주도 요구 분석)은 바이브코딩이 자연어 기반으로 작동하므로, 교사의 요구가 곧 개발 프롬프트의 기초가 된다는 점에서 전통적 개발보다 요구 분석의 정밀성이 개발 결과물에 직접적으로 영향을 미친다. DP3(디바이스 독립적 설계)과 DP5(영속적 데이터 아키텍처)는 AI가 생성한 코드가 특정 환경에서만 테스트되는 경향이 있어 다중 디바이스 검증과 데이터 영속성이 사전에 프롬프트에 명시되어야 함을 시사한다. DP6(낮은 진입 장벽의 협업)은 바이브코딩의 빠른 프로토타이핑 특성상 인증 시스템과 같은 복잡한 기능을 의도적으로 배제하는 설계 전략이 유효함을 보여준다. DP7(AI 오류 대비 설계)은 바이브코딩에 내재된 AI 환각 문제에 대한 직접적 대응으로, 전통적 개발에서는 요구되지 않는 바이브코딩 고유의 원리이다. 이 원리들은 본 연구의 특정 맥락에서 도출되었으나, 바이브코딩을 활용하여 교육용 앱을 개발하고자 하는 다른 연구자와 교사에게 실천적 지침을 제공할 수 있을 것이다.

5. 결론 및 제언

5.1 결론

본 연구는 바이브코딩 접근법을 활용하여 지역특화 교육 과정을 지원하는 '핀 캔버스' 앱을 개발하고 중학교 현장에 적용한 과정과 결과를 DBR 방법론에 따라 보고하였다. 연구 결과를 연구 문제별로 종합하면 다음과 같다.

첫째, 바이브코딩을 활용한 교육용 웹 앱 개발은 DBR의 반복적 설계 원리에 따라 효과적으로 수행될 수 있었다. 2회의 설계-개발-평가 사이클을 통해 교사의 요구사항이 단

계적으로 반영되었으며, 특히 온라인 메시지를 활용한 교사-개발자 간 즉각적 피드백 체계는 신속한 반복적 정교화를 가능하게 하였다. 교사들은 기존 상용 도구에서는 "팀에서 검토한 후 연락하겠다"는 응답이 일반적인 반면, 바이브코딩 기반 개발에서는 피드백이 거의 즉시 반영되었다고 보고하였다. 이 과정에서 도출된 7개의 설계원리는 향후 유사한 바이브코딩 기반 교육용 도구 개발에 참고할 수 있는 실천적 지침을 제공한다.

둘째, 바이브코딩 접근법은 교육 기술 개발의 민주화 가능성을 실증적으로 보여주었다. 교사들은 기존 도구를 '선택하여 사용하는' 방식이 교육적 요구를 일부만 충족시키는 반면, 바이브코딩을 통해 필요에 맞게 개발된 도구는 필요한 모든 기능을 활용할 수 있게 한다고 평가하였다. 실제로 핀 캔버스는 Google My Maps나 Padlet 등 기존 도구로는 충족되지 않았던 커스텀 배경, 레이어 분리, 핀 아이콘 커스터마이징 등의 기능을 통합적으로 제공하였으며, 교사들은 이를 통해 학습 목표 달성의 효과와 효율을 높일 수 있었다고 보고하였다. 또한 교사들은 한 학기 동안의 연속적인 탐구 작업을 수행하는 데 핀 캔버스가 핵심적 역할을 하였으며, 이 도구 없이는 연속적인 작업이 어려웠을 것이라고 평가하였다.

다만, 교사들은 이러한 개발 방식의 확산을 위해서는 개발자와 교사 간의 긴밀한 소통 구조가 전제되어야 하며, 에듀테크 코치와 같은 학교 내 지원 체계의 도입이 필요하다고 제안하였다. 이는 바이브코딩의 교육적 활용이 도구의 기술적 발전뿐 아니라 이를 지원하는 제도적 환경의 조성을 함께 요구함을 시사한다.

5.2 연구의 의의

본 연구의 의의는 다음과 같다. 첫째, 학술적 측면에서 바이브코딩이라는 새로운 소프트웨어 개발 접근법을 교육공학 연구의 맥락에서 DBR 방법론을 적용하여 체계적으로 검토하였다. 특히 개발 과정의 반복적 사이클과 설계원리를 명시적으로 도출함으로써, 바이브코딩 기반 교육용 도구 개발의 방법론적 기초를 제공하였다. 둘째, 실천적 측면에서 교사가 교육용 도구의 소비자에서 개발 참여자로 전환될 수 있는 구체적인 과정과 조건을 제시하였다. 교사 인터뷰 결과는 교사-개발자 협업 모델이 교사의 교육적 요구를 충실히 반영한 맞춤형 도구 개발에 효과적임을 보여주었다. 셋째, 교육 형평성의 관점에서 농어촌 소규모 학교도 바이브코딩을 활용한 맞춤형 도구 개발을 통해 지역 특성을 반영한 교육과정을 운영할 수 있는 가능성을 제시하였다.

5.3 한계 및 후속 연구 제언

본 연구에는 다음과 같은 한계가 있다. 첫째, 교육적 효과 분석이 교사 인터뷰에 기반한 질적 데이터에 한정되어 있어, 학습자를 대상으로 한 직접적인 효과 검증이 이루어지지 못하였다. 특히 46명 학생의 학습 경험에 대한 데이터가

교사의 관찰 보고에 한정되어 있어, 학습자 관점의 직접적인 평가가 포함되지 않았다. 학생의 학습 성과, 협업 역량, 지역에 대한 관심도 변화 등을 측정하는 양적 연구가 후속 사이클에서 수행될 필요가 있다. 둘째, 바이브코딩으로 개발된 앱의 장기적인 유지보수와 확장성에 대한 검토가 필요하다. 교사 인터뷰에서도 플랫폼 의존성에 따른 접속 문제 등이 한계로 지적되었으며, AI가 생성한 코드의 품질과 보안은 지속적인 관심이 필요한 영역이다. 셋째, 본 연구는 단일 학교의 사례에 기반하므로 일반화에 한계가 있으며, 다양한 교과, 학교급, 지역 맥락에서의 적용 가능성을 탐색하는 후속 연구가 필요하다.

후속 연구를 위한 제언은 다음과 같다. 첫째, 핀 캔버스의 교육적 효과를 학습자 대상 사전-사후 설계를 통해 검증하는 준실험 연구가 필요하다. 둘째, 바이브코딩을 활용한 교육용 도구 개발에 관한 교사 연수 프로그램의 개발 및 효과 연구가 필요하다. 교사들이 바이브코딩의 구조를 이해하면 개발자와의 소통이 보다 효과적으로 이루어질 수 있다는 인터뷰 결과는 이러한 연수의 필요성을 뒷받침한다. 셋째, 교사들이 제안한 다양한 교과 활용 사례(국어과 문학관 지도, 미술과 세계 미술관 지도 등)를 실제로 구현하고 적용하는 연구를 통해 핀 캔버스의 교과 간 확장 가능성을 검토할 필요가 있다. 넷째, 연구자가 개발자와 연구자의 이중 역할을 수행하였으며, 교사 인터뷰가 개별 독립 응답이 아닌 협의적 응답으로 수집된 점은 본 연구의 방법론적 한계이다. 또한 삼각검증에 활용된 세 가지 데이터 출처(교사 인터뷰, 온라인 의사소통 기록, 연구자 성찰일지)는 각각 서로 다른 관점의 데이터를 제공하였으나, 세 출처 모두에 연구자가 직접 간접적으로 관여하고 있어 연구자로부터 완전히 독립적인 데이터 출처가 포함되지 않았다. 후속 연구에서는 제3자에 의한 인터뷰 수행, 개별 교사의 독립적 응답 수집, 학생 대상 직접 데이터 수집 등을 통해 연구의 객관성을 강화할 필요가 있다.

결론적으로, 바이브코딩은 교육 기술 개발의 민주화를 실현할 수 있는 유망한 접근법이다. 본 연구의 교사 인터뷰 결과가 보여주듯이, 교사의 요구에 최적화된 맞춤형 도구를 신속하게 개발하고 반복적으로 개선할 수 있다는 점은 교육 혁신의 새로운 가능성을 열어준다. 그러나 이 가능성을 실현하기 위해서는 바이브코딩 도구의 발전, 교사 역량 강화, 개발자-교사 협업 체계 구축 등 다양한 조건이 함께 갖추어져야 할 것이다.

참고문헌

- [1] Ministry of Education. (2022). *2022 revised curriculum overview* (2022 개정 교육과정 총론). Ministry of Education.
- [2] Karpathy, A. (2025, February 2). *There's a new kind of coding I call "vibe coding"* [Post]. X. <https://x.com/karpathy/status/1886192184808149383>
- [3] Collins Dictionary. (2025, November 6). *Collins' Word of the Year 2025: AI meets authenticity as society shifts*. Collins Dictionary Language Blog. <https://blog.collinsdictionary.com/language-lovers/collins-word-of-the-year-2025/>
- [4] Mehta, I. (2025, March 6). *A quarter of startups in YC's current cohort have codebases that are almost entirely AI-generated*. TechCrunch. <https://techcrunch.com/2025/03/06/a-quarter-of-startups-in-ycs-current-cohort-have-codebases-that-are-almost-entirely-ai-generated/>
- [5] Gadde, A. (2025). Democratizing software engineering through generative AI and vibe coding: The evolution of no-code development. *Journal of Computer Science and Technology Studies*, 7(4), 556-572. <https://doi.org/10.32996/jcsts.2025.7.4.66>
- [6] Sarkar, A., & Gordon, A. D. (2025). Vibe coding: Programming through conversation with artificial intelligence. *arXiv*. <https://doi.org/10.48550/arXiv.2506.23253>
- [7] Ge, Y., Wang, Z., Chen, X., Zhang, T., Lu, S., Chen, B., Liu, X., Li, Y., Zhang, S., Liu, Y., Wu, Y., Shen, Y., Chen, W., Duan, N., & Lou, J. G. (2025). A survey of vibe coding with large language models. *arXiv*. <https://doi.org/10.48550/arXiv.2510.12399>
- [8] Fawzy, A., Tahir, A., & Tempero, E. (2025). Vibe coding in practice: Motivations, challenges, and a future outlook—A grey literature review. *arXiv*. <https://doi.org/10.48550/arXiv.2510.00328>
- [9] Lee, C. (2025, June 5). Andrew Ng says vibe coding is a bad name for a very real and exhausting job. *Business Insider*. <https://www.businessinsider.com/andrew-ng-vibe-coding-unfortunate-term-exhausting-job-2025-6>
- [10] Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher*, 41(1), 16-25. <https://doi.org/10.3102/0013189X11428813>
- [11] Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *The Journal of the Learning Sciences*, 2(2), 141-178. https://doi.org/10.1207/s15327809jls0202_2
- [12] Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 15-22). Springer. https://doi.org/10.1007/978-3-642-77750-9_2
- [13] The Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32(1), 5-8. <https://doi.org/10.3102/0013189X032001005>
- [14] Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences*, 13(1), 1-14. https://doi.org/10.1207/s15327809jls1301_1
- [15] Kang, J., & Lee, S. (2011). Design-based research (DBR) as field research method to improve instructions. *Korean Journal of Educational Methodology Studies*, 23(2), 323-354. <https://doi.org/10.17927/tkjems.2011.23.2.323>
- [16] Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5-23. <https://doi.org/10.1007/BF02504682>
- [17] McKenney, S., & Reeves, T. C. (2018). *Conducting educational design research* (2nd ed.). Routledge. <https://doi.org/10.4324/9781315105642>
- [18] Kim, M. (2014). Development and Implementation of Multi-Purpose Peer Learning Mobile App for Students' Participation in Large-Size classes. *Journal of Educational Information and Media*, 20(4), 547-570. <https://doi.org/10.15833/KAFEIAM.20.4.547>
- [19] Kang, J., & Park, T. (2022). Research trend analysis of design and development research and design-based research: A scoping review. *Journal of Educational Technology*, 38(3), 661-698. <https://doi.org/10.17232/KSET.38.3.661>
- [20] Kim, S., & Lee, Y. (2019). Development of programming-based TPACK educational program through design-based research. *Journal of the Korea Society of Computer and Information*, 24(10), 267-278. <https://doi.org/10.9708/jksci.2019.24.10.267>
- [21] Hoadley, C., & Campos, F. C. (2022). Design-based research: What it is and why it matters to studying online learning. *Educational Psychologist*, 57(3), 207-220. <https://doi.org/10.1080/00461520.2022.2079128>
- [22] McHugh, S., Carroll, N., & Connolly, C. (2023). Low-code and no-code in secondary education—Empowering teachers to embed citizen development in schools. *Computers in the Schools*, 1-26. <https://doi.org/10.1080/07380569.2023.2256729>
- [23] Reeves, T. C. (2006). Design research from a technology perspective. In J. van den Akker, K. Gravemeijer, S. McKenney, & N. Nieveen (Eds.), *Educational design research* (pp. 52-66). Routledge. <https://doi.org/10.4324/9780203088364-13>
- [24] Kim, E., & Park, J. (2012). A study on educational app development using app authoring tools. *Journal of Digital Convergence*, 10(5), 1-6.
- [25] Lee, M. (2024). Software Education Class Model using Generative AI - Focusing on ChatGPT. *Journal of Practical Engineering Education*, 16(3), 275-282. <https://doi.org/10.14702/JPEE.2024.275>
- [26] Gambacorta, L., Huang, H., Qiu, H., & Tan, J. (2024). *Generative AI and labour productivity: A field experiment on coding* (BIS Working Papers No. 1208). Bank for International Settlements. <https://www.bis.org/publ/work1208.htm>
- [27] Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-

101. <https://doi.org/10.1191/1478088706qp0630a>

- [28] Carter, N., Bryant-Lukosius, D., DiCenso, A., Blythe, J., & Neville, A. J. (2014). The use of triangulation in qualitative research. *Oncology Nursing Forum*, 41(5), 545-547. <https://doi.org/10.1188/14.ONF.545-547>



윤명희

- 2000년 부산대학교 물리학과(이학사)
- 2002년 부산대학교 물리학과(이학석사)
- 2025년 부산대학교 교육과미디어융합전공 (박사수료)
- 2024년~현재 글로벌사이버대학교 미디어콘텐츠 창작학과 특임교수

✚ 관심분야 : 생성형 AI 활용 교육, 바이브코딩, 설계기반 연구, 소프트웨어 교육

✉ mhee.yoon@gmail.com



박수홍

- 1984년 부산대학교 교육학과(교육학사)
- 1987년 부산대학교 교육측정평가전공 (교육학석사)
- 2001년 인디애나 대학교 교수체제공학 (철학박사)
- 2003년~현재 부산대학교 교육학과 교수

✚ 관심분야 : 앙트러프러너십, 인공지능 융합교육, 창작 교육

✉ suhongpark@pusan.ac.kr

부 록

〈표 1〉 노코드, 로우코드, 바이브코딩 접근법 비교

항목	노코드	로우코드	바이브코딩
상호작용 방식	GUI 드래그 앤 드롭, 템플릿 선택	비주얼 에디터 + 부분적 코드 작성	AI에게 자연어 프롬프트 전달
커스터마이징 유연성	사전 구축된 컴포넌트로 제한	보통: 커스텀 코드로 확장 가능	높음; AI가 템플릿 제약 없이 커스텀 소스 코드 생성
산출물	플랫폼 종속 애플리케이션	플랫폼 호스팅 또는 내보내기 가능한 앱	완전한 소스 코드 (React, TypeScript 등)
기술적 진입 장벽	최저; 코딩 지식 불필요	낮음~보통; 기초 코딩 지식 도움됨	낮음; 프롬프트 엔지니어링과 기본 디버깅 필요
반복적 수정	플랫폼의 편집 옵션으로 제한	플랫폼 프레임워크 내에서 가능	대화형 프롬프트로 신속하게 가능; 코드 수준 변경 가능
플랫폼 종속성	높음; 벤더 종속 (lock-in)	보통; 부분적 종속	낮음; 소스 코드 이식 가능
위험/제한 사항	기능 상한선; 플랫폼 역량 초과 불가	커스터마이징에 따라 복잡성 증가	AI 환각 (hallucination); 코드 품질 변동성; 디버깅 어려움
대표 도구	Padlet, Google Sites, Canva, Notion	AppSheet, Glide, PowerApps	Lovable, Cursor, Claude Code, Replit Agent

〈표 2〉 기존 도구와 핀캔버스 비교

기능 요구사항	Padlet	Google My Maps	Miro	Pin Canvas
커스텀 캔버스 배경	×	×	○	●
지도 기반 핀 배치	△	●	△	●
커스텀 핀 아이콘	×	○	×	●
레이어 시스템	×	○	×	●
드로잉 도구	△	○	●	●
비로그인 공유	●	△	○	●
실시간 협업	●	×	●	●

(● 완전지원, ○ 부분지원, △ 제한적, × 미지원)

〈표 3〉 핀 캔버스 개발에 사용된 기술스택

구분	기술/도구	역할
프론트엔드	React, TypeScript, Tailwind CSS	사용자 인터페이스 구현
백엔드/DB	Supabase (PostgreSQL)	데이터 저장, 실시간 동기화, 사용자 인증
AI 개발 도구	Lovable, Claude Code	자연어 기반 코드 생성 및 수정
배포	Lovable Publish	원클릭 웹 호스팅 및 배포

〈표 4〉 노코드, 로우코드, 바이브코딩 접근법 비교

사이클	단계	주요 활동	산출물
사이클 1 (2025.05~09)	분석/탐색	교사 인터뷰; 기존 도구 분석	6가지 핵심 요구사항 도출; 단일 도구로 충족 불가 확인
	설계/구축	PRD → Lovable + Supabase + Claude Code	약 3주 만에 작동 가능한 프로토타입 완성
	평가/성찰	수업 적용(9월); 교사 온라인 의사소통 피드백	3가지 핵심 문제 확인; 교육적 분석을 통한 잠정적 설계원리(DP3, DP5, DP7 초안) 도출
사이클 (2025.10~12)	분석/탐색	심각도 및 수업 영향도 기준 우선순위 설정	좌표 스케일링 = 최우선 과제
	설계/구축	상대 좌표 시스템; 자동 저장; UX 개선	이슈당 1~3일 내 수정 반영
	평가/성찰	교사 확인; 사후 교사 인터뷰	잠정적 설계원리 검증; 4개 추가 원리 도출(DP1, DP2, DP4, DP6); 7개 최종 설계원리 확정

〈표 5〉 바이브코딩 기반 교육용 앱 개발을 위한 설계원리

번호	설계원리	설명
DP1	교육자 주도 요구 분석	앱 개발은 개발자의 가정이 아닌, 교사 인터뷰를 통한 체계적 요구 분석에서 출발해야 한다.
DP2	신속한 피드백 루프	수업 중 즉각적인 피드백을 위해 개발자와 교사 간 실시간 의사소통 채널을 구축해야 한다.
DP3	디바이스 독립적 설계	교육용 앱은 교실에서 사용되는 다양한 디바이스(PC, 태블릿, 스마트폰)에서 일관되게 작동해야 한다.
DP4	다중 AI 도구 전략	서로 다른 개발 과제에 상호 보완적인 AI 도구를 활용한다: 하나는 빠른 프로토타이핑에, 다른 하나는 정밀한 디버깅에 사용한다.
DP5	영속적 데이터 아키텍처	사용자가 생성한 모든 콘텐츠(핀, 드로잉, 레이어)는 데이터 손실을 방지하기 위해 실시간으로 자동 저장되어야 한다.
DP6	낮은 진입 장벽의 협업	교실 환경에서의 접근 장벽을 줄이기 위해 로그인 없이 참여할 수 있도록 한다.
DP7	선제적 AI 오류 관리	AI 환각(hallucination)과 코드 오류에 대비하여 버전 관리를 유지하고 대안 전략을 사전에 준비한다.