

## SW 비전공 대학생을 위한 디자인 씽킹 기반 코딩교육에서의 학습경험 분석

신윤희<sup>1</sup>·정효정<sup>1\*</sup>·송종숙<sup>2</sup>

<sup>1</sup>단국대학교 교양교육대학

<sup>2</sup>아주대학교 다산학부대학

## Analysis of Learning Experience in Design Thinking-Based Coding Education for SW Non-Major College Students

Yoonhee Shin<sup>1</sup>·Hyojung Jung<sup>1\*</sup>·Jongsuk Song<sup>2</sup>

<sup>1</sup>College of General Education, Dankook University, Gyeonggi-do, 16890, Korea

<sup>2</sup>Dasan University College, Ajou University, San S, Woncheon-Dong, Yeongtong-gu, Suwon 443-749, Korea

### [요 약]

본 연구는 디자인 씽킹에 기반을 둔 코딩교육의 성과를 확인하고, 발전 방향을 도출하기 위하여 학습자들의 경험을 구체적으로 규명하기 위한 목적으로 진행되었다. 이를 위하여 경기도 소재 D 대학교의 교양필수 교과목인 ‘창의적 사고와 코딩’에서 디자인 씽킹을 접목한 코딩교육을 실시하였으며, 성찰일지 분석을 통하여 학습자들이 무엇을 배웠고, 어떠한 어려움을 경험하였는지를 분석하였다. 연구 결과, 디자인 씽킹 기반 코딩교육의 성과는 인지적 차원, 정의적 차원, 실천적 차원에서 구분해볼 수 있었다. 또한 학생들은 문제 발견, 아이디어 도출, 협력 및 소통, 코딩 구현 단계에 대한 어려움을 크게 느끼는 것으로 나타났다. 본 연구에서는 이러한 연구 결과를 토대로 향후 디자인 씽킹을 접목한 코딩교육에서 고려해야 할 시사점을 제시하였다.

### [Abstract]

The purpose of this study was to clarify the outcomes and difficulties of learners who participated in coding education based on design thinking. To do this, we conducted a coding education combining design thinking with 'creative thinking and coding' which is a core curriculum of D university in Gyeonggi - do. Through analyzing the reflection journal, we analyzed what learners learned and what difficulties they experienced. As a result of the study, outcomes recognized by learners can be divided into the cognitive, affective, and practical dimension. In addition, we were able to identify problems, difficulties in problem finding, ideation, communication, and implementing coding. Based on the results of this study, we suggested strategies to consider coding education.

**색인어** : 디자인 씽킹, 코딩교육, 비전공자, 학습경험, 창의적 문제 해결

**Key word** : Design thinking, Coding education, Non-major college students, Learning experience, Creative Problem-Solving

<http://dx.doi.org/10.9728/dcs.2019.20.4.759>



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 21 March 2019 ; Revised 11 April 2019

Accepted 26 April 2019

\*Corresponding Author; Hyojung Jung

Tel: 82-31-8005-3972

E-mail: [hyojung.jung@dankook.ac.kr](mailto:hyojung.jung@dankook.ac.kr)

## I. 서론

최근 창의성 향상과 코딩교육에 대한 중요성이 부각되면서 정부 차원의 지원이 확대되고 교양교과목에 코딩교육을 필수 편성하는 대학교가 증가하고 있다. 코딩교육에의 접근성을 높이는 것은 다양한 전공의 학생들에게 디지털 리터러시를 높여주는 효과를 기대할 수 있으나, 어떠한 교육경험을 제공하는가에 따라 그 성패는 좌우될 수 있기 때문에, 학습자들의 니즈와 경험을 고려한 교육의 설계가 요구된다[1]. 이러한 측면에서 코딩교육에서 학생들의 경험에 대한 연구가 이루어지고 있다.

코딩교육 과정과 같은 컴퓨팅사고 교육에서 공통적으로 나타나는 초보학습자의 어려움을 살펴보면 다음과 같다. 먼저 새로운 코딩언어에 대한 낯섦에서 오는 어려움, 알고리즘 및 구문 명령어 등 개념이해에 대한 어려움, 프로그래밍 과정에 대한 지루함, 프로그래밍이 어렵다는 인식, 그리고 학교교육의 제한된 학습시간 등이 포함된다[2], [3]. 유사하게 Burbaite, Drasute, & Stuijks(2018)연구에서는 코딩교육의 어려움을 내용자체 및 기술적 측면의 어려움, 인지처리 과정의 복잡성으로 분류하였다. 코딩과정에서는 여러 개념과 원리들을 먼저 이해하고 이를 습득하여 새로운 문제에 적용하는 과정까지를 요구하기 때문이다[4]. 또한 초보학습자의 경우 내용자체가 생소하고 어렵기 때문에 오류가 발생하거나 모르는 내용이 나올 경우 학습동기가 저하되는 경우가 많다[4]. 한편, 김수환(2015) 연구에서는 Jenkins(2002)의 제안을 토대로 컴퓨팅사고 교육과정에서의 학습자의 어려움을 극복하는 요소로 재미, 흥미, 난이도를 설정하고 각 주별 수업내용에서의 요소를 측정하였다[5]. 연구 결과 컴퓨팅 사고와 긍정적 상관관계가 높은 변인으로는 컴퓨터 프로그래밍 능력이나 난이도보다는 스스로 잘하고 있다고 느낀 프로그래밍 자신감 및 흥미인 것으로 나타났다. 이러한 결과는 창의성에서도 자기효능감 및 동기 등의 요인이 중요한 상관관계를 보여주듯이 컴퓨팅사고 또한 인지적 영역 뿐 아니라 정의적 영역을 함께 고려하여 교수전략을 설계해야한다는 시사점을 준다.

이와 관련하여 윤옥한(2017) 연구에서는 기존의 수업방식에 변화가 필요하다고 하였으며, 학습자가 제공하는 상황 맥락적인 정보를 교수설계에 반영할 필요가 있다고 하였다[6]. 최근 PBL, 플립러닝 등의 교육방법을 접목시켜 융·복합 교육을 운영하기 위한 다양한 수업모델 및 발전방안에 대한 연구가 수행되고 있으나[7]-[9], 여전히 융·복합교육 지원전략에 대한 이론에 대한 고찰과 이를 검증하기 위한 실증연구는 부족한 상황이다. 특히 비전공자 학생들에게 확대되는 창의성 향상을 위한 소프트웨어 교육을 효과적으로 운영하기 위해 학생들이 무엇을 어려워하는지를 분석하고 구체적으로 어떤 과정에서 교수지원이 요구되는지에 대한 분석이 필요하다.

전 세계적으로 창의·혁신 역량이 중요해지고 체계적인 프로세스를 기반으로 아이디어를 실제 산출물로 만들어보는 디자인 씽킹 모델을 활용하는 사례가 증가하고 있으나, 여전히 창의

성에 대한 인식 및 변인간의 관계에 대한 연구가 주를 이루고 있어 구체적인 창의성 향상을 위한 지원전략과 이를 토대로 하는 실험연구가 요구되는 실정이다[10]. 서응교(2017)는 플립러닝과 디자인 씽킹에 기반을 둔 코딩교육을 시도한 바 있는데, 연구 결과 학습자들의 문제해결력과 창의적 잠재력에 있어 긍정적인 효과가 나타났으며, 교과목 만족도에 있어서도 4.00 이상의 높은 수준을 보여주었다[11]. 비전공자들의 경우 코딩교육에 대한 학습동기가 떨어지고, 학습과정에서 지나친 어려움을 경험하게 되어 오히려 부정적인 인식을 얻게 될 가능성이 있는데[12], 디자인 씽킹을 접목한 코딩교육은 비전공자들에게 협업의 가치와 문제를 발견하고 해결하는 문제해결의 과정을 통해 소프트웨어 교육의 의미를 체감할 수 있도록 돕는 것으로 나타난다.

창의력 기반 코딩교육이 확산되고 효과적인 융·복합수업 전략 연구가 절실히 요구되는 시점에, 본 연구는 비전공자들을 위한 디자인 씽킹 기반 코딩교육에서 학습자들이 얻게 된 성과와 경험한 어려움을 분석하고 이를 토대로 교수지원 방법에 대한 구체적인 방안을 마련하는 데 연구의 목적이 있다. 이에 이 연구를 통하여 비전공 대학생 대상의 디자인 씽킹 기반 코딩교육을 적용하고 학생들이 인식한 교육 성과와 어려움을 양적·질적으로 분석하고자 한다. 이는 4차 산업혁명 시대를 앞두고 다양한 전공의 학생들에게 적용될 수 있는 창의융합형 코딩교육의 방향성에 대하여 제시하는 데 중요한 단서를 제공해줄 수 있을 것으로 본다. 본 연구의 구체적인 연구문제는 다음과 같다.

첫째, 비전공 대학생을 위한 디자인 씽킹 기반 코딩교육에서 나타난 교육 효과는 무엇인가?

둘째, 비전공 대학생을 위한 디자인 씽킹 기반 코딩교육에서 학생들이 경험한 어려움은 무엇인가?

## II. 연구 방법

### 2.1. 연구대상 및 배경

본 연구는 경기도 소재 D대학교 1학년 학생을 대상으로 한 2학점의 교양필수 교과목인 <창의적사고와 코딩>을 수강한 학생을 대상으로 이루어졌다. 수업에 앞서 디자인 씽킹을 접목한 코딩교육의 실효성을 확인하고 효과성을 높이기 위한 목적에서 연구를 수행할 것임을 학습자들에게 밝히고 동의를 구하였다. 학생들은 성적에 대한 불이익 없이 자유롭게 연구에 참여할 수 있음을 고지하였으며, 최종적으로 연구에 동참한 학생들은 115명에 해당하였다. 대상자는 남자 49명(42.6%), 여자 66명(57.4%)로 구성되었으며, 소속 학과는 상경대학 경영학부 43명(37.4%), 사회과학대학 커뮤니케이션학부 35명(30.4%), 상경대학 무역학과 18명(15.7%), 경제학과 15명(13.0%), 그리고 행정학과 및 기타 4명(3.5%)이었다. 학생들은 3-4명이 팀을 이루어 프로젝트기반의 수업에 참여하였으며, 수업 종료시점까지 팀

변경 없이 진행되었다.

## 2.2. 연구대상 교과목의 특징 및 운영 과정

본 연구에서는 가장 대중적으로 활용되고 있는 디자인 씽킹 프로세스인 아이데오(IDEO)의 5단계 모델을 접목하였다. 이 모델은 ‘공감하기-문제 정의하기-아이디어 만들기-시제품 만들기-발전시키기’ 단계를 포함하며, 전통적인 창의적 문제해결 모형에서의 ‘관심영역 발견-문제 발견-문제해결 발견-해결책 수행’과 절차 및 내용이 유사하여 창의적 문제해결모형으로 효과적으로 활용되고 있다[11], [12]. 이는 창의적 문제해결력과 협력적 소통능력 함양을 목적으로 학생들이 단순히 소프트웨어 지식을 습득하는 것이 아니라 사고력 함양에 목적을 둔다는 데 의의가 있다. 전체 15주 수업 중 총 7주에 걸쳐 디자인 씽킹에 기반을 둔 코딩교육이 적용되었으며, 주차별 주요 내용은 다음 <표 1>에 제시된 바와 같다.

표 1. 디자인 씽킹 기반 코딩교육 수업모형

Table 1. Design Thinking-based Coding Education Class Model

Week	Stages	Main contents
1	Orientation	Design Thinking Lecture Team Building
2	Empathize	Find the inconvenience you want to solve in your life Observe where the inconvenience occurs, interview with the user
3	Define	Persona, development of travel map Find the root cause of discomfort
4	Ideate	Brainstorming activities to derive a variety of ideas Selecting ideas that can be solved by software among derived ideas
5-6	Prototype	Design menu structure and screen structure Create key menus and screen models using reference application templates
7-8	Test	Implement key functions by using example code (change screen configuration, change data, etc.) Using sample code to refine the details (button additions, language extensions, etc.)
9	Summary	Final presentation and feedback Evaluation Activity

먼저, 공감하기(Empathize) 단계에는 ‘사용자 스토리’ 활동을 통해 삶의 맥락 속에서 대상(고객, 사용자)을 관찰 및 인터뷰함으로써 사용자의 경험과 시각에 대한 충분히 이해하는 상호 작용 과정을 거친다. 두 번째, 문제정의(Define) 단계로 ‘사용자 특징정의’ 및 사용자 경험분석’ 활동을 통해 사용자스토리 활동에서 발견한 자료를 종합하고 사용자의 경험 및 감정곡선을 그려봄으로써, 주어진 문제를 분명하게 이해하고 간결하게 서술한다. 세 번째, 발상(Ideate)단계를 위해 실제 앱 화면 및 구조도를 분석하고 구상한 아이디어를 피오리를 활용해 모형으로

제작해보는 ‘모형제작’ 활동을 하며 팀원들과 함께 문제를 정립하고 이를 해결하기 위한 다양한 아이디어를 공유하는 브레인스토밍을 실시하여, 가능한 폭넓게 아이디어를 도출해내는데 중점을 둔다. 이 때 발산, 수렴적 측면의 창의적 해결과정이 나올 수 있는 과정이므로 학습자 간 충분한 피드백 활동을 지원하도록 한다. 마지막으로 프로토타입(Prototype)과 적용(Test) 단계를 위해 피오리 코딩 프로그램을 지원한다. 앞서 논의된 추상적인 아이디어를 실질적이고 구체적으로 시각화시키는 단계로서 시제품 생성의 단계이다. 무엇보다도 이러한 과정이 한번에 끝나는 것이 아니라, 최상의 결과를 얻어 낼 때까지 빠르게 반복하여 가장 효과적으로 시제품을 개선하고 만족스러운 결과를 도출할 수 있다. 디자인 씽킹 각 단계에서 분석적 사고와 직관적 사고를 적절히 조화롭게 사용하면서 문제 해결해 나간다. 사용자중심 어플리케이션 개발 프로그램의 장점으로는 먼저, 대학생이 일상생활에서 흔히 활용하는 어플리케이션을 통해 문제를 정의하고 아이디어를 구상하는 과정에서 학습자의 동기부여를 촉진하고 발산적 사고를 유도할 수 있다는 이점을 지닌다. 또한 소프트웨어에 익숙하지 않은 학습자가 SAP에서 개발한 피오리라는 레퍼런스 어플리케이션 형식의 앱 개발 프로그램을 통해 코딩을 처음 접하지만 로봇이나 블록 형태를 활용하는 것보다 직접적인 코딩언어나 정교한 오류수정 과정을 거칠 수 있다는 점에서 이점이 있다[11]. 그러나 이러한 이점이 교수자의 역량에 크게 구애받지 않고 한정된 수업시간 내 효과적으로 추진되기 위해서는 해당 프로그램을 통해 성취되는 구체적인 창의역량에 대한 분석과 사고과정에 대한 분석이 필요하며 이를 토대로 지속적인 교수전략 모색과 수업모형의 정교화를 위한 연구가 요구된다.

## 2.3. 자료의 수집 및 분석방법

본 연구를 위하여 수집한 자료는 설문과 성찰일지이다. 먼저 설문을 통해서 디자인 씽킹 단계별로 경험한 인지부하 수준을 리커트(Likert) 5점 척도로 표현하도록 하였다. 인지부하는 개인이 학습 과정에서 경험하는 인지적 어려움을 의미하며, 본 연구에서는 프로젝트를 수행하는 동안 발생한 인지부하의 수준을 자가진단의 형태로 측정하였다. 정효정, 김혜원(2012)의 연구에서 제안한 협력적 문제해결 상황에서의 인지부하 측정을 위한 문항 중 본 연구의 맥락과 맞는 문항을 선별하여 총 6문항, 리커트 5점 척도로 구성하였다[13]. 구체적인 문항의 내용은 다음 <표 2>에 제시한 내용과 같다. 전체 단계별 변화 추이를 분석하기 위하여 반복측정분산분석(Repeated Measured ANOVA)을 실시하였다.

또한 성찰일지에서는 전체 학습 단계별로 무엇을 배웠고, 어떠한 어려움에 직면하였는지를 기술하도록 안내하였다. 성찰일지의 내용은 일정한 기준과 의미의 공통성 여부에 따라 영역별로 분류하는 귀납적 내용 분석(Inductive Content Analysis) 방법을 통해 범주화하였다. 공통적인 개념이나 상위 개념으로 묶어 ‘1차 소영역’으로 통합하였고, 다시 1차 소영역들을 논리적

으로 맥락을 같이 하거나 상통하는 주제로 묶어 ‘2차 중영역’으로 묶은 후 ‘3차 대영역’으로 범주화하였다. 주요 키워드별로 내용을 분류한 후, 분류된 요인에 대한 빈도를 산출하였다. 신뢰도 확보를 위하여 세 명의 연구자가 함께 요인을 분류하였으며, 평정자간 일치도 값을 산출하였다. 분산분석 결과 ICC(Intraclass Correlation Coefficient) 값은 .952로 매우 신뢰로운 수준이었고, 이는 통계적으로 유의하였다( $p=.000$ ).

표 2. 인지부하 측정을 위한 설문 내용

Table 2. Questionnaire for cognitive load measurement

Types	Survey item	
Cognitive load	1. I felt exhausted after completing the assignment task.	
Difficulty of Instruction	2. The guidance on the task was easy to understand.	
Satisfaction on Process	3. The process by which our team solved the tasks was straightforward.	Not at all (1)
Satisfaction on Collaboration	4. Collaboration with team members was satisfactory.	-
Satisfaction on Outcome	5. I think I solved the task successfully.	Very agree (5)
Engagement	6. I was able to participate in the class happily without knowing the time.	

### III. 연구 결과

#### 3.1 인지부하 발생 수준

디자인 씽킹 단계별 인지부하 발생 수준에 대한 각 문항별 측정결과는 다음과 같다.

표 3. 단계별 인지부하 발생 수준에 대한 기술통계

Table 3. Descriptive statistics on level of cognitive load

	Empathize		Define		Ideate		Prototype		Test	
	M	SD	M	SD	M	SD	M	SD	M	SD
1	2.02	1.03	2.17	.99	2.44	1.01	2.79	1.10	3.39	1.20
2	3.99	.94	4.01	.90	4.02	.86	3.73	.99	3.61	1.00
3	4.10	.91	4.16	.82	4.11	.89	3.95	.97	3.68	1.10
4	4.19	.92	4.24	.86	4.25	.90	4.13	.95	4.04	1.02
5	4.27	.82	4.25	.83	4.27	.78	4.16	.83	3.89	.99
6	3.89	.95	3.82	.98	3.77	1.03	3.71	1.06	3.75	1.11

1번 문항 ‘수행과제를 마친 후 지치는 느낌을 받았는지’에 대한 분석 결과, 코딩구현( $M=3.39$ ,  $SD=1.204$ ), 모형제작( $M=2.79$ ,  $SD=1.096$ ), 아이디어 만들기( $M=2.44$ ,  $SD=1.009$ ), 문제정의( $M=2.17$ ,  $SD=.990$ ), 공감( $M=2.02$ ,  $SD=1.034$ ) 순서대로 높은 결과 값을 나타냈다. 각 단계별 차이에 대해 살펴보기 위해 일원 반복측정 분산분석을 수행한 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다( $F(4,$

$456)=57.10$ ,  $p<.00$ ,  $\eta_p^2=.33$ ). 사후검증의 결과에서도 5단계에서의 인지부하는 모두 유의한 차이가 있었다( $p<.05$ ). 구체적으로 공감과 문제정의 단계에서보다 아이디어 만들기, 모형제작 및 코딩구현 단계를 수행한 후 지치는 느낌을 많이 받았다는 것을 알 수 있었다( $p<.01$ ). 특히 코딩구현 단계가 다른 단계와 모두 높은 수준의 유의한 차이를 보여 코딩구현 단계를 수행한 후 가장 지치는 느낌을 받았음을 알 수 있었다( $p<.00$ ).

2번 문항 ‘수행과제에 대한 안내는 이해하기 쉬웠는지’에 대한 분석 결과, 아이디어 만들기( $M=4.02$ ,  $SD=.858$ ), 문제정의( $M=4.01$ ,  $SD=.903$ ), 공감( $M=3.99$ ,  $SD=.941$ ), 모형제작( $M=3.73$ ,  $SD=.994$ ), 코딩구현( $M=3.61$ ,  $SD=1.002$ ) 순서대로 높은 결과 값을 나타냈다. 각 단계별 차이에 대해 살펴보기 위해 일원 반복 측정 분산분석을 수행한 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다( $F(4, 456)=13.635$ ,  $p<.00$ ,  $\eta_p^2=.107$ ). 사후검증 결과에서는 모형제작과 코딩구현 단계와 공감, 문제정의, 및 아이디어 만들기 단계 사이에서의 유의한 차이가 나타났다( $p<.01$ ). 특히, 아이디어 만들기 단계와 코딩구현 단계 사이에서 유의성이 가장 높게 나타났다( $p=.00$ ). 즉, 학습자들은 공감, 문제정의, 아이디어 만들기 단계에서보다 모형 제작 및 코딩구현에서의 안내에 대해 이해하는 데 어려움을 겪었음을 확인할 수 있었다.

3번 문항 ‘우리 팀이 수행과제를 해결하는 과정은 문제없이 수월했다’에 대한 분석 결과, 문제정의( $M=4.16$ ,  $SD=.823$ ), 아이디어 만들기( $M=4.11$ ,  $SD=.892$ ), 공감( $M=4.10$ ,  $SD=.908$ ), 모형제작( $M=3.95$ ,  $SD=.972$ ), 코딩구현( $M=3.68$ ,  $SD=1.095$ ) 순서로 높은 점수를 나타냈다. 일원 반복측정 분산분석을 수행한 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다( $F(4, 456)=12.920$ ,  $p<.00$ ,  $\eta_p^2=.102$ ). 사후검증의 결과에서는 공감하기 단계와 코딩구현 단계 간에는 유의한 차이가 있었으나( $p<.01$ ), 문제정의, 아이디어 만들기, 모형제작 단계 간에는 유의한 차이가 없었다. 즉, 학습자들은 팀별로 과제를 수행하는 데 있어서 코딩구현 단계에서 가장 어려움을 느꼈음을 알 수 있었다.

4번 문항 ‘팀원과의 협력은 만족스러웠다’에 대한 분석 결과, 아이디어 만들기( $M=4.25$ ,  $SD=.904$ ), 문제정의( $M=4.24$ ,  $SD=.861$ ), 공감( $M=4.19$ ,  $SD=.916$ ), 모형제작( $M=4.13$ ,  $SD=.951$ ), 코딩구현( $M=4.04$ ,  $SD=1.021$ ) 순서대로 높은 점수를 나타냈다. 일원 반복측정 분산분석 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다( $F(4, 456)=3.936$ ,  $p<.01$ ,  $\eta_p^2=.03$ ). 사후검증의 결과에서는 코딩구현 단계와 다른 모든 단계에서 (공감, 문제정의, 아이디어 만들기 단계( $p<.00$ ), 모형제작 단계( $p<.05$ )) 유의한 차이가 나타났다. 문제정의 단계와 모형제작 단계 사이에서도 유의한 차이가 나타났다( $p<.05$ ). 이는 학습자들이 아이디어 만들기와 문제정의 단계에서 팀원과 협력하는 과정에 상대적으로 만족하는 것을 알 수 있었으며, 코딩구현 단계를 수행하는 데 있어서는 팀원과의 협력과정에 가장 만족도가 낮은 것으로 나타났다.

5번 문항 ‘수행과제를 성공적으로 해결했다고 생각한다’에

대한 분석 결과, 공감(M=4.27, SD=.820), 아이디어 만들기(M=4.27, SD=.776), 문제정의(M=4.25, SD=.825), 모형제작(M=4.16, SD=.833), 코딩구현(M=3.89, SD=.994) 순서로 높은 점수를 나타냈다. 일원 반복측정 분산분석을 수행한 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다(F(4, 456)=14.577, p<.00,  $\eta_p^2$ =.11). 사후검증의 결과로 이를 보다 구체적으로 살펴보면, 코딩구현 단계와 대부분의 다른 단계들(공감, 문제정의 및 아이디어 만들기) 사이에서 유의한 차이가 나타났으며(p<.00), 코딩구현 단계와 모형제작 사이에서도 유의한 차이가 나타났었다(p<.01). 이를 통해 학습자들이 과제를 수행한 후 성공적으로 해결했다고 생각하는 정도가 코딩구현 단계에서 가장 낮은 것을 알 수 있었다.

6번 문항 ‘시간 가는 줄 모르고 즐겁게 수업에 참여할 수 있었다’에 대한 분석 결과, 공감(M=3.89, SD=.953), 문제정의(M=3.82, SD=.976), 아이디어 만들기(M=3.77, SD=1.026), 코딩구현(M=3.75, SD=1.107), 모형제작(M=3.71, SD=1.057) 순서대로 높은 점수를 나타냈다. 일원 반복측정 분산분석을 수행한 결과, 5단계 과정 사이에서 인지한 부하의 차이는 유의한 차이가 나타났다(F(4, 456)=2.699, p<.05,  $\eta_p^2$ =.02). 사후검증의 결과에서는 공감 단계와 모형제작 단계 사이에서만 유의한 차이가 나타났다(p<.05). 이는 학습자들이 대부분의 활동에서 과제를 수행하는 데 몰입한 것을 알 수 있었으나 모형제작 단계에서는 다소 몰입하는 데 어려움을 느꼈음을 알 수 있었다.

### 3.2 교육 효과에 대한 인식

성찰일지 분석을 통하여 디자인 씽킹에 기반을 둔 코딩교육의 교육효과에 대한 학생들의 인식을 확인하였다.

표 4. 교육효과에 대한 학생들의 인식

Table 4. Students' perception of educational effectiveness

Dimension	Keywords	Frequency (Ratio)
Cognitive	Knowledge acquisition	Coding development process Coding language 71(61.21)
	Motivation	Interest, Efficacy 22(18.97)
Affective	Perception	15(12.93)
	Belief	7(6.03)
	Attitude	Practical will 15(12.93)
Practical	Competency	Problem finding ability Problem-solving ability Ability to collaborate Coding Implementation capability 50(43.10)
Sum		116(100.00)

성찰일지 분석을 통하여 도출된 교육효과는 크게 인지적 차원, 정의적 차원, 실천적 차원을 구분해볼 수 있었다. 먼저 학습

자들은 인지적 차원으로 코딩, 코딩 도구, 디자인 씽킹과 관련된 지식을 습득할 수 있었다는 점에서 만족감을 드러냈다.

*코딩을 직접해보면서 간단하게나마 통합개발프로그램을 사용하는 방법과 코딩/소프트웨어에 관련된 단어들도 알게 되었다(경영학부, 여, 최OO).*

*코딩하는 법을 배우고 우리가 사용하는 어플리케이션이 어떤 구조로 돌아가는지 배웠다(커뮤니케이션학부, 남, 황OO).*

*코딩하는 방법을 알게 되었고 디자인 씽킹에 대해 알게 되었다. 처음으로 코딩을 배우게 되고 구현 단계까지 해본 과정 자체가 나에게겐 신기하고 새로웠다(경영학부, 남, 이OO).*

*이번 기회를 통해 코딩을 해볼 수 있어서 좋았습니다. 저 같은 경우에는 이 수업 전까지 코딩에서 대해서 몰랐는데 직접해보아서 이러한 영역을 알아간다는 데에 뿌듯함을 느꼈습니다. 또 코딩을 구현하기 전까지 많은 단계를 거쳐야 한다는 것을 깨달았습니다(경영학부, 남, 황OO).*

*아직 완벽하지는 않지만 코딩을 어느 정도 구현할 수 있었다. 초반에는 코딩이 사실 쓸모없는 것이라는 생각을 가지고 있었는데 유용하게 쓰이는 분야가 많다는 것을 알게 되었다(커뮤니케이션학부, 여, 한OO).*

*코딩에 대해 더 자세하게 알 수 있었다. 평소 내가 사용하는 어플리케이션의 구조, 흐름 등에 대해서도 이해할 수 있었고 복잡하지만 나도 실존하는 어플리케이션 같은 나만의 어플을 만들 수 있을 것 같다는 확신도 들었다(커뮤니케이션학부, 여, 조OO).*

정의적 차원은 동기 향상, 인식 변화 및 신념 변화를 경험하였다는 의견으로, 코딩/실습/협력학습에 대한 흥미를 얻었다는 점, 코딩/컴퓨터/앱 개발 및 문제해결/협력학습에 대한 인식이 긍정적으로 변화했다는 점, 코딩/컴퓨터 활용/자기주도학습에 대한 효능감이 높아졌다는 점 등을 긍정적인 경험으로 꼽았다.

*팀 프로젝트를 처음하게 되어 어떻게 시작해야 할지부터 막막했는데 팀끼리 원활하게 역할을 분담하고 맡은 바를 모두 성실하게 완성하여 혼자 과제를 수행하는 것보다 수월하게 과제를 끝낼 수 있었다. 팀 간의 상호작용이 매우 중요하다는 것을 배웠고 또한 한 번도 접해보지 못한 코딩에 대하여 기초적인 틀을 잡게 해준 귀중한 기회였다고 생각한다(경영학부, 남, 강OO).*

*개인적으로 이 수업을 듣기 전부터 코딩 & 프로그래밍 쪽*

에 관심이 있던 터라 흥미로웠고 특히 C, C++처럼 어려운 코딩 대신 누구나 쉽게 접근할 수 있는 프로그램으로 코딩을 배워서 조금 더 수월하게 코딩에 대한 흥미를 얻었다(무역학과, 여, 정OO).

코딩이 마냥 어렵기만 했는데 직접 코드를 다뤄보고 페이지도 만들어보니 마음의 벽을 허물 수 있었습니다. 컴퓨터 다루는 일이 어려울 줄만 알았는데 즐거워서 배움에 있어서 함부로 '어렵다. 난 못할거야' 하는 장벽을 스스로 세우지 말아야겠다는 점을 배웠다고 생각합니다(경영학부, 여, 권OO).

잘 다루지 못하는 컴퓨터를 이용해 코딩이라는 낯설고 복잡한 프로그램을 직접 배우고 제작하니, 뭐든지 열심히 하면 할 수 있다는 것을 깨달았고 매우 어려운 프로그램이었지만, 결과를 보니 뿌듯해서 컴퓨터를 잘못 다루는 나의 단점(?)을 조금이나마 극복한 것 같다(커뮤니케이션학부, 남, 김OO).

실천적 차원에서는 코딩/컴퓨터 활용에 대한 자신감을 얻게 되었고, 이는 향후 학습 참여 및 문제 해결에 대한 적극적인 태도를 갖게 하는 데 도움이 될 것이라는 의견을 나타내었다. 또한 분석 능력/탐구 능력/문제해결 능력/협력 및 대인관계 능력/유연한 사고능력/창의적 사고능력 등 다양한 측면에서의 역량이 증진되었다고 응답하였다.

일상에서 사용하는 앱을 분석해보고 부족한 점이나 불편했던 점을 찾아 코딩으로 직접 수정, 적용해보면서 창의적 사고 능력과 동시에 코딩에 대해 어느 정도 이해할 수 있었던 것 같다(경영학부, 여, 조OO).

코딩이라는 게 섬세하고 신경을 많이 기울여야 하는 구나를 느낄 수 있었고 코딩을 구현하다가 중간에 코드에 오류가 생겨 구현이 되지 않을 때 왜 되지 않는 건가 스스로 회의감에 빠져 고민할 수 있는 시간을 가질 수 있었다. 끝까지 구현하려고 참을성을 가지고 하다 보니 나는 '무언가에 대한 집념과' 집중력 그리고 '인내의 자세를 가질 수 있었다(경제학과, 여, 임OO).

일상생활과 관련된 문제들을 생각해보고 이에 대한 대응 방안이나 해결책등을 팀원들과 함께 모색해나가는 과정에서 협동심, 책임감 등을 기를 수 있었다. 뿐만 아니라 컴퓨터 프로그램을 통해 직접 코딩을 구현해내고 그 결과물을 보는 것도 흥미로웠고 유익한 경험이 된 것 같다(무역학과, 남, 장OO).

앱을 사용할 때 불편한 점이 있으면 잠깐 투덜대고 그냥 사용하였는데, 이 수업시간에는 불편한 점이 있으면 그걸 어떻게 고칠 수 있는지에 대한 해결방안을 생각함으로써 문제해

결능력을 기를 수 있었다. 또한 코딩의 두려움이 사라졌다(경제학과, 여, 김OO).

직접 어플리케이션을 분석해보고 문제점을 찾고 개선점까지 찾아내는 과정에서 분석능력과 문제해결능력을 키운 것 같습니다. 또한 직접 피오리 프로그램을 만들어보면서 하나의 앱에도 복잡한 체계와 무척 많은 입력 값이 들어간다는 걸 배웠고 앞으로 어떤 앱을 사용할 때 분석하면서 보게 될 것 같습니다(경영학과, 여, 김OO).

### 3.3 학생들이 직면한 어려움

성찰일지 분석을 통하여 디자인 씽킹에 기반을 둔 코딩교육에서 학생들이 직면하는 어려움의 내용을 확인하였다.

표 5. 디자인 씽킹 기반 코딩교육에서 학생들이 경험한 어려움  
Table 5. Difficulties experienced by students in design thinking-based coding education

Type	Keywords	Frequency (Ratio)
Instruction related	Familiarity	Low familiarity to the subject itself Familiarity with using computers and coding programs 11(10.28)
	Lack of understanding	Lack of understanding of project contents Lack of understanding about learning contents 11(10.28)
Task related	Problem definition	Finding a problem 4(3.74)
	Ideation	Creating an idea 7(6.54)
	Coding	The overall process of coding implementation Identify and resolve coding errors 8(7.48)
		Coding Principle Application 4(3.74)
Extra	Team related	Communication problems between team members Role sharing problem among team members 14(13.08)
	Environmental issues	Lack of computers and lab space 16(14.95)
Sum		107(100.00)

성찰일지 분석을 통하여 도출된 어려움의 유형은 크게 교수적 안내 관련, 과제 관련, 기타 요소로 구분해볼 수 있었다. 먼저 교수적 안내와 관련하여, 학습자들은 내용 및 코딩언어 등에 대한 친숙도가 낮은 것을 학습초기에 경험하는 어려움으로 제시하였다. 또한 수행해야 하는 과제의 방향성과 내용에 대한 이해

부족에서 어려움을 경험하였음을 토로하였다.

처음에 피오리 프로그램을 만들면서 정말 어렵다고 생각했지만 비밀번호를 까먹어서 처음부터 다시 만들어본 결과 무척 재밌다는 생각을 했습니다. 첫 번째 만들 때는 3시간이 걸렸지만 두 번째로 만들 때는 1시간 이내로 걸렸습니다. 반복적인 학습, 노력을 통해 어려움을 극복할 수 있었습니다(경영학과, 여, 김OO).

과제로 무엇을 제출해야 하는지 헷갈렸었는데 교수님께서 한 번 더 설명을 해주셔서 이해할 수 있었다(커뮤니케이션학부, 여, 조OO).

가끔 코딩에 익숙하지 않고 뒤에 앉다보니 수업내용을 놓쳐 헤매는 경우가 있었는데 이를 주변 친구들과 함께 찾아내면서 극복했던 것 같다(경영학부, 여, 정OO).

과제 관련 요소로는 디자인 씽킹의 프로세스에 따라 문제를 찾고 문제의 본질을 규명하는 것의 어려움, 아이디어를 도출하는 과정에서 창의적인 아이디어를 제안하는 것, 프로토타입을 개발하고 수정/보완하는 과정에서 에러, 코딩 구현의 어려움, 코딩 원리 적용의 어려움 등을 경험하는 것으로 나타났다.

계속 업데이트 되는 어플 환경에서 불편한 점을 찾기가 힘들었다. 마지막에 스스로 코딩을 하는 것에 있어서 가장 어렵고 항상 프로그램(피오리) 자체인 오류도 많아 힘들었던 기억이 있다. 이에 수업시간에 놓친 부분은 책과 교수님 Band 자료를 통해 스스로 따라가고 익혀나갔다(커뮤니케이션학부, 여, 도OO).

늘 편하게 쓰는 앱에서 불편한 점을 몇 개씩 찾아내고 아이디어를 만드는 게 어려웠다. 하지만 계속해서 아이디어를 내보고 한 가지를 두고 깊게 생각하니까 잘 찾을 수 있게 되었다(경영학부, 여, 이OO).

아이디어를 내는 것에서 어려움을 겪었다. 우리가 직접 아이디어를 내서 그 문제를 해결해 나가야 되는 것이다 보니, 조원 각자가 내는 의견을 다 들어보고 선정을 하는 것에서 어려움을 겪었다(경제학과, 여, 이OO).

문제점을 해결하기 위한 아이디어가 잘 떠오르지 않아 힘들었다. 그래도 작은 변화를 주어 문제점을 해결하기 위해 노력했다. 또 코딩 구현과정에서 코드를 입력한 후 test를 할 때 오류가 나타나면, 원인을 몰라 답답하기도 했다. 그 때마다 처음부터 다시 하거나 잘못 입력한 문자를 찾으면서 오류를 해결했다(경영학부, 남, 황OO).

코딩 과정이 이해하면 쉬운 부분인데, 이해가 되지 않아서

순서대로 보면서 하다보니까 시간도 오래 걸리고 오류도 발생했었다. 안 되는 부분을 찾아서 그 부분이 왜 안 되는지를 파악하고, 이해하려고 노력했다(커뮤니케이션학부, 남, 김OO).

배우지 않은 코딩 구현은 하고 싶어도 쉽게 할 수 없었다. 코딩 활용 방법 PDF 파일을 응용하고, 팀원과 같이 의견을 나누며 구현했다(경영학부, 여, 조OO).

코딩을 구현해내는 과정에서 따라가지 못해서 뒤쳐진 경우도 있었고, 에러가 뜬다거나 컴퓨터 작동이 더디다거나 렉을 먹어 처음부터 다시 해야 하는 경우 등의 어려움도 있었다. 하지만 다시 계속 도전해보고자 하는 의지와 교수님, 그리고 주변 동기들의 도움으로 이 어려움들을 잘 이겨낼 수 있었다(무역학과, 남, 정OO).

기타 요소로는 팀원들과의 소통 문제, 역할분담 등에서 어려움을 경험하였고, 환경적 요소로는 보유 컴퓨터나 실습환경의 문제 등을 제시하였다.

우리 조는 가위바위보나 제비뽑기를 통해 원하는 역할을 수행하여 마찰이 없었다. 하지만 다른 조에서 역할 분담 때문에 어려움에 직면할 수 있다고 생각한다. 이러한 장벽을 극복하기 위해서는 팀 내의 원활한 소통과 양보 정신이 필요하다(경영학과, 남, 강OO).

코딩 구현할 때 노트북이 1개여서 많은 시간이 걸렸고, 막상 할 수 있는 사람은 한명이어서 미안했습니다. 옆에서 '이건 아닐까?'라는 식의 의견도 나누고 그 한명이 묵묵히 도와줬기 때문에 극복할 수 있었습니다(커뮤니케이션학과, 여, 원OO).

#### IV. 결론 및 제언

디자인 씽킹에 기반을 둔 코딩교육은 협력을 통해 창의적으로 문제를 해결해나가는 경험, 코딩을 통해 프로토타입의 완성도를 높여나가는 경험을 통해 비전공자들이 코딩의 의미와 가치를 체감할 수 있도록 지원하는 힘을 갖고 있다. 이는 4차 산업혁명 시대의 대응을 위해 대학의 코딩교육이 지향하고 있는 융복합 교육의 방향성과도 잘 맞물린다고 볼 수 있다. 그러나 한 학기라는 시간은 기본적인 컴퓨팅 지식을 쌓기에도 부족한 시간인데, 디자인 씽킹의 전반적인 과정을 이해하고 익숙하지 않은 코딩 과제까지 수행해내는 것은 교수자와 학생 모두에게 결코 쉽지 않은 도전이 될 수 있다. 이에 본 연구에서는 디자인 씽킹에 기반을 둔 코딩교육이 과연 어떠한 교육효과를 거두고 있는지, 학생들은 어떠한 어려움에 직면하게 되며 어떻게 이를

해결하고 있는지를 점검하고자 하였다.

연구 결과, 학생들은 교육의 전체적인 과정 중 공감, 문제정의, 아이디어 만들기 단계에서는 비교적 어렵지 않게 교육에 참여하였으나, 코딩을 본격적으로 수행하는 과정에서 이전 단계에 비하여 높은 수준의 인지부하를 경험하는 것으로 나타났다. 아무래도 코딩 프로그램 및 활동에 대한 사전지식 및 경험이 없고 코딩 과정에서 다양한 어려움과 문제를 경험하게 되면서 자연스럽게 나타난 현상이었다. 물론 문제 정의 및 아이디어 만들기 단계에서도 기존에는 눈여겨보지 않았던 어플리케이션의 문제를 찾고, 문제를 구체화하고, 이를 해결하기 위한 다양한 아이디어를 고안하는 과정에 대한 어려움을 토로하기도 하였다. 학생들은 이를 극복하기 위하여 교수자가 제공한 추가 자료를 적극적으로 활용하였으며, 반복연습 및 팀원들과의 협력적 활동을 통해 코딩구현 단계를 극복해나가는 모습을 보였다. 인지부하 분석 결과 산출물에 대한 만족도 및 협력에 대한 만족감이 교육을 진행하면서 점차 높아지는 것을 확인할 수 있었는데, 학생들은 문제 해결의 과정에서 디자인 씽킹이 기반을 두고 있는 집단 창의성(Collective Creativity)이 코딩교육에서 직면하게 되는 어려움을 극복하는 데 기여하고 있음을 유추할 수 있었다. 또한 어려움을 해결하는 과정에서 소프트웨어 개발의 과정에서 나타나는 협력적 문제해결의 중요성을 체감하게 된 것도 또 다른 차원의 성과라고 볼 수 있겠다. 나정은(2017)은 비전공자 대상의 코딩교육에서 단계별 학습 성취를 맛보게 하는 것의 중요성, 코딩교육을 통해 컴퓨팅 기술이 문제해결의 도구로 활용될 수 있도록 경험하는 것의 중요성을 강조한 바 있다[14]. 디자인 씽킹은 코딩교육이 이러한 효과를 거두는 데 기여할 수 있음을 가능하게 한다.

본 연구를 통해 도출한 학생들이 경험하는 어려움에 대한 분석 결과를 토대로 비전공자 대상 코딩교육에서 염두에 두어야 할 몇 가지 요소를 제안하자면 다음과 같다. 첫째, 코딩을 실행하는 단계에서의 어려움을 조절하기 위하여, 코딩 원리 및 기본 기능에 대한 이해를 돕고 반복연습을 지원할 필요가 있다. 둘째, 학습자가 사전학습 혹은 사후학습을 자율적으로 진행할 수 있도록 추가자료 활용 가이드가 필요하다. 이와 관련하여 본 연구에서도 플립러닝 및 SNS(네이버 밴드)를 활용한 학습지원을 진행한 바 있다. 이는 학생들이 직면한 어려움을 지속적이고 직접적으로 지원하는 데 도움이 된 것으로 나타났다. 향후 간단한 코딩 활동을 연습해볼 수 있도록 안내하는 쿠키 제공 등을 고려해보는 것도 좋겠다. 또한 아이디어 만들기 단계에서 구현의 어려움을 염두에 두고 다양한 아이디어를 제안하지 못하는 상황을 고려하여, 자신이 제안하고자 하는 아이디어를 그대로 구현하지 못하더라도 모형제작-구현과정의 반복(피드백)을 통해 자신의 아이디어가 점점 구체화되고 발전해나가는 것을 경험할 수 있도록 활동을 설계하고 지원해야 할 것이다. 코딩을 활용한 창의성교육 프로그램에서 궁극적인 목적은 코딩 자체의 지식 습득이 아닌 학습자의 인지 및 정서적 측면의 긍정적 향상을 통한 창의적 문제해결력을 촉진하는 데 있기 때문이다. 또한 디자인 씽킹 활동 수행과정동안 창의적인 아이디어에 대한 평가와

코딩 구현완성도에 대한 평가가 균형 있게 이루어질 수 있도록 구체적인 평가 준거를 제시할 필요가 있다. 이 외에도 공감 및 아이디어 만들기 단계에 비해 모형제작 및 구현 단계에서는 협업에 어려움을 경험하는 점을 고려하여 향후 협업 소프트웨어 도구를 활용하는 방안이 시도될 필요가 있다.

## 참고문헌

- [1] W. S. Kim, "A Study on the Recognition of Freshman on Computational Thinking as Essential Course," *Culture and Convergence*, Vol. 39, No. 6, pp. 141-170, December 2017.
- [2] J. W. Choi and Y. J. Lee, "The Analysis of Learners' Difficulties in Programming Learning," *The Journal of Korean Association of Computer Education*, Vol. 17, No. 5, pp. 89-98, September 2014.
- [3] T. Jenkins, "On the difficulty of learning to program," in *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Loughborough, pp.53-58, 2002.
- [4] R. Burbaitė, V. Drašutė, and V. Štūkys, "Integration of computational thinking skills in STEM-driven computer science education," in *2018 IEEE Global Engineering Education Conference*, Tenerife, pp.1824-1832, 2018.
- [5] S. H. Kim, "Analysis of Non-Computer Majors' Difficulties in Computational Thinking Education," *The Journal of Korean Association of Computer Education*, Vol. 18, No. 3, pp. 49-57, May 2015.
- [6] O. H. Yoon, "A Study of the Instructional Systems Design Model for STEAM Education: Focus on Design Thinking," *Korean Journal of General Education*, Vol. 11, No. 1, pp. 443-474, February 2017.
- [7] S. Y. Pi, "A Study on Coding Education of Non-Computer Majors for IT Convergence Education," *Journal of Digital Convergence*, Vol. 14, No. 10, pp. 1-8, October 2016.
- [8] K. S. Oh, E. K. Suh, and H. J. Chung, "A Study on Development of Educational Contents about Combining Computational Thinking with Design Thinking," *Journal of Digital Convergence*, Vol. 16, No. 5, pp. 65-73, May 2018.
- [9] S. J. Kim and D. E. Cho, "A Study on Learning Model for Effective Coding Education," *Journal of the Korea Convergence Society*, Vol. 1, No. 1, pp. 7-12, February 2018.
- [10] Y. S. Song and K. H. Lee, "An Analysis of the Effect of Creativity Education on College Students' Creativity and Cognitive Learning Competency," *Global Creative*



*Leader*, Vol. 2, No. 2, pp. 73-88, December, 2012.

- [11] E. K. Suh, "Development of Creative Thinking and Coding Course Method on Design Thinking Using Flipped Learning," *Journal of Learner-centered Curriculum and Instruction*, Vol. 17, No. 16, pp. 173-199, August 2017.
- [12] H. S. Shin, E. K. Suh, K. S. Oh, H. J. Chung, and S. H. Park, "A study on the development of software education program-based on design thinking for non-majors SW basic education," in *Proceedings of 2018 Annual Conference of the Korean Association of Computer Education*, Daegu, pp. 125-128, 2018.
- [13] H. J. Jung and H. W. Kim, "An Exploratory Validation for the Constructs of Collaboration Load," *Journal of Educational Technology*, Vol. 28, No. 3, pp. 619-640, September 2012.
- [14] J. E. Nah, "Software Education Needs Analysis in Liberal Arts," *Korean Journal of General Education*, Vol. 11, No. 3, pp. 63-89, June 2017.



**신윤희(Yoonhee Shin)**

2015년 : 한양대학교 대학원 (교육학 석사)

2018년 : 한양대학교 대학원 (일반대학원 교육학 박사-교수설계 및 이러닝)

2019년~현 재: 단국대학교 교양교육대학 초빙교수

※관심분야 : 교수설계, E-Learning, 소프트웨어교육(Software Education)



**정호정(Hyojung Jung)**

2006년 : 한양대학교 대학원 (교육학 석사)

2010년 : 한양대학교 대학원 (일반대학원 교육학 박사-교수설계 및 이러닝)

2013년~현 재: 단국대학교 교양교육대학 조교수

※관심분야 : E-learning, MOOC(Massive Open Online Course), 성인교육(Adult Learning)



**송종숙(Jongsuk Song)**

2011년 : 에든버러대학교 Moray House School of Education (교육학 석사)

2017년 : 한양대학교 대학원 (일반대학원 교육학 박사-교수설계 및 이러닝)

2019년~현 재: 아주대학교 다산학부대학 강사

※관심분야 : 교수설계, E-Learning, 정보격차(Digital Divide)