

<http://dx.doi.org/10.17703/JCCT.2019.5.1.211>

JCCT 2019-2-25

## 컴퓨터 비전공자의 효과적인 소프트웨어 프로젝트 수행을 위한 교수자-학습자 피드백 방법에 관한 연구

### A Study on Teacher-learner Feedback Method for Effective Software Project Execution of Non-Computer Major Students

정혜욱\*

Hye-Wuk Jung\*

**요약** 대학에서의 학기말 프로젝트 작업은 한 학기동안 학습한 내용을 기반으로 학생 스스로 주제 선정, 계획, 결과 도출 과정을 진행해 가는 학습자 중심의 학습 방법이다. 소프트웨어 관련 교과목의 학기말 프로젝트 작업의 경우 해당 프로그래밍 언어에 대한 다양한 기법을 학습 한 후 창의적인 프로그램 개발 과정을 통해 결과물을 완성하게 된다. 그러나 교양과목으로 소프트웨어 교과목을 수강하는 컴퓨터 비전공자는 프로그래밍 언어를 이해하는데 많은 어려움을 느끼고 있기 때문에 학생들이 프로젝트 수행을 원활하게 진행 할 수 있도록 유도하는 교수자의 피드백이 필요하다. 따라서 본 연구에서는 컴퓨터 비전공자의 학기말 프로그래밍 교과목에 대한 프로젝트 수행과정에 적용 할 수 있는 교수자-학습자간의 토론을 통한 피드백 방법을 제안하고, 실제 프로젝트 작업과정에 적용하여 진행과정 및 결과물 분석을 통해 의미 있는 결과를 확인하였다.

**주요어** : 소프트웨어 교육, 비전공자, 소프트웨어 융합, 컴퓨터 프로그래밍

**Abstract** The term project executed at the university is a learner-centered learning method in which students select their topics, draw up their plans, and produce results by themselves based on the content they have learned during the semester. Through the term-end project of the subjects relating software, students learn various techniques for the programming language and produce the outcomes of their project by the creative program development process. However, non-computer majors who take software course as liberal arts subjects have difficulty in understanding the programming language, so it is necessary to provide feedback from their professor for encouraging students in carrying out their projects smoothly. Therefore, a feedback method by the discussions between a professor and learners that can be applied to the term-end project of programming subject for the non-computer majors is proposed. The proposed method was apply to the actual term-end projects and the meaningful results were confirmed through the analysis of the project processes and outcomes.

**Key words** : Software Education, Non-Majors, Software Convergence, Computer Programming

---

\*정혜원, 경기대학교 융합교양대학 교양학부  
접수일: 2019년 1월 3일, 수정완료일: 2019년 1월 24일  
게재확정일: 2019년 1월 31일

Received: January 3, 2019 / Revised: January 24, 2018  
Accepted: January 31, 2019  
\*Corresponding Author: wukj@kyonggi.ac.kr  
Dept. of College of Liberal Arts and Interdisciplinary  
Studies, kyonggi University, Korea

## 1. 서 론

전 세계적으로 컴퓨터 공학 이외의 비전공 분야에 IT 기술을 접목한 융복합 시스템 및 서비스가 꾸준히 증가하고 있다. 이러한 시점에서 프로그램 개발자를 목표로 하는 전공자뿐만 아니라 비전공 학생들도 소프트웨어를 이용한 새로운 가치를 창출 할 수 있는 능력을 준비해야지만 빠르게 변화되는 사회에 적응 할 수 있다 [1][2].

프로그래밍 작업은 실생활에 필요한 시스템을 자동화하거나 여러 정보를 가공하여 반복적이고 복잡한 문제를 풀기위한 용도 등으로 다양하게 활용 할 수 있다 [3]. 일반적인 프로그래밍 처리절차는 어떤 주어진 문제를 정의하고 문제를 풀기위한 알고리즘을 작성 후 수집 및 가공된 데이터를 이용하여 알고리즘을 검토해보고 마지막으로 프로그래밍 작업을 한다.

이러한 프로그래밍 과정으로 부터 문제 해결 능력, 창의적 사고력, 의사 결정 능력을 키울 수 있게 되기 때문에 컴퓨팅 사고력(Computational Thinking)이 키워지게 된다. 컴퓨팅 사고력이라는 용어는 1980년 미국 MIT의 시모어 페퍼트(Seymour Papert) 교수가 처음 사용하였으며, 프로그래밍 행위를 인간과 컴퓨터의 대화로 정의하였다[4]. 즉, 언어가 의사소통뿐만 아니라 사고의 내용이나 결정에 중요한 역할을 한다는 것이었다. 2006년 지넷 윙(Jeanette M. Wing)이 모든 사람들이 컴퓨팅 사고력을 기본적으로 갖추어야 한다고 주장하면서 부터 컴퓨팅 사고력에 대한 관심이 증가되기 시작했다[5]. 컴퓨팅 사고력은 컴퓨터 과학의 이론, 기술, 도구를 활용하여 현실의 복잡하고 어려운 문제를 해결하는 사고방식으로 복잡한 문제를 단순화하고 정보 분석, 알고리즘 작성 등을 통하여 복잡한 문제를 논리적, 효율적으로 해결하는 능력을 말한다[6]. 컴퓨팅 사고력의 주요 과정은 추상화, 자동화로 나누어 처리된다. 추상화 과정은 현실의 문제를 이해, 분석, 해결 가능한 상태로 만들기 위해 구조화 및 모델링 하는 작업이고, 자동화 과정은 실제로 동작하여 문제를 해결 할 수 있는 자동화 방법을 설계하여 프로그램을 구현 후 시뮬레이션 프로그램의 테스트 및 추론을 실시하는 과정으로 처리하게 된다[6].

이와 같이 프로그래밍 과정은 컴퓨팅 사고력을 기반으로 진행되기 때문에 소프트웨어 교육을 통해 소프트

웨어 기초 소양을 함양하고, 다양한 학문 분야의 문제 해결을 위하여 컴퓨팅 사고력을 적용하는 경험은 문제를 효율적으로 해결하는데 도움이 된다.

따라서 국내·외 대학에서는 비전공자를 대상으로 프로그래밍의 기본지식에 대한 학습과 프로젝트 제작과정을 통해 소프트웨어를 이해하고 응용 및 활용 할 수 있는 방법에 대하여 교육을 시행하고 있다.

국외대학의 경우를 살펴보면, 하버드대학교에서는 ‘CS50’ 이라는 교과목을 통해 기본적인 컴퓨터 이론과 C, JavaScript 등 다양한 프로그래밍 언어를 익히고 프로젝트 수행하며 경제, 보안, 게임과 같은 실세계 문제에 적용하여 문제해결의 경험을 할 수 있도록 한다. 또한, 최종 프로젝트 결과물의 전시회를 개최하여 서로 다른 분야의 연구내용을 참고 할 수 있는 기회도 제공한다[7]. 컬럼비아대학교의 ‘Computing in Context’ 과정은 Python 프로그래밍 언어를 6~7주 동안 학습 한 후 프로젝트 작업을 통해 자신의 전공분야(사회과학, 인문학 등)와 관련된 문제해결을 위한 추상화 및 자동화과정을 수행 할 수 있게 한다[8]. 버클리대학교에서는 스크래치(Scratch)언어 기반의 스냅!(Snap!)이라는 소프트웨어를 이용한 프로그래밍 기법 학습과정인 ‘컴퓨터 과학’ 교과목을 개설하여 비전공 학생들의 소프트웨어 교육을 시행하고 있다. 이과정은 기본적인 프로그래밍 원리 학습과 함께 학생들이 다양한 아이디어(Idea) 창출 할 수 있도록 관심 있는 주제를 학생 스스로 선택하여 프로젝트를 진행해보는 내용을 포함하고 있다[9]. 이와 같이 국외대학들의 비전공자들에 대한 소프트웨어 교육 방향은 컴퓨터 및 프로그래밍에 대해 학습한 내용을 기반으로 프로젝트 작업을 통해 자신의 전공에 적용 및 응용하는 능력을 키울 수 있게 교육하고 있음을 알 수 있다.

국내에서는 소프트웨어 교육에 대한 관심이 높아지면서 대학교에서 컴퓨터 비전공자를 대상으로 하는 다양한 소프트웨어 교육과정을 도입하고 있다. 연세대학교에서는 2017년도부터 비전공자를 위한 필수교양 과목으로 소프트웨어 영역을 신설하고 소프트웨어 기초 과목으로 ‘컴퓨팅적 사고와 SW프로그래밍’을 개설하였다. 이 수업을 수강 후 학생들이 컴퓨터 전반에 대해 이해하고 프로젝트 작업을 통해 스스로 문제의 정의부터 해결까지의 과정을 경험하며 컴퓨팅적 사고 함양과 일상생활에 적용 할 수 있는 응용력을 배양하는 것을

목표로 한다[10]. 2015년에 소프트웨어 중심대학으로 선정된 성균관대학교는 소프트웨어 기초교육을 위하여 성균SW교육원을 설립하고 모든 신입생에게 컴퓨팅 사고력 중심으로 편성된 교과목(컴퓨팅과 기초SW, 컴퓨팅사고와 SW코딩, 문제해결과 알고리즘)을 단계별로 학습 후 소규모 프로젝트를 진행하도록 교육시키고 있다[11].

이와 같이 국내·외 대학에서는 컴퓨터 비전공자들에게 컴퓨터를 이해하고 프로그래밍 작업을 경험 할 수 있는 커리큘럼을 개발하여 학생들이 다양한 경험을 할 수 있는 기회를 제공하고 있다. 특히, 프로그래밍 교과목에서 학기 후반부에 시행하는 프로젝트 작업은 학생 스스로 학습한 프로그래밍 기법을 실생활 문제와 연계하여 구현해봄으로써 컴퓨터에 대한 응용력을 키울 수 있게 하고 있다.

그러나 컴퓨터를 전공하지 않는 비전공 신입생의 경우 프로그래밍 원리의 이해와 작성방법에 대하여 어려워하기 때문에 프로젝트 작업에 대해서도 많은 부담감을 느끼게 되고 작업내용의 응용범위나 완성도가 낮아질 수 있다. 따라서 컴퓨터 비전공자가 소프트웨어 교과목과 관련된 프로젝트를 원활하게 수행 할 수 있게 유도해주는 교수자의 피드백 방법이 필요하다.

이에 본 연구에서는 컴퓨터 비전공자를 대상으로 하는 소프트웨어 수업에서 프로젝트 작업을 위한 교수자의 피드백 방법을 설계하고, 학생들의 학기말 프로젝트 지도에 적용 한 후 프로젝트 결과 내용을 분석하였다. 그리고 프로젝트 수행 후 학습자가 체감하는 프로그래밍 능력의 변화와 프로젝트 수행과정 중 학생들이 느꼈던 어려운 점에 대한 의견을 살펴봄으로써, 향후 컴퓨터 비전공자들의 프로젝트 작업에 대한 지도방안을 제시하고자 한다.

## II. 연구방법

본 연구에서는 한국교육개발원에서 개발한 5가지 CT(Computational Thinking)학습 모델 중 시연중심(DMM, Demonstration Modeling Making)모델과 개발중심모델(DDD, Discovery Design, Development)을 참고하여 그림 1과 같은 학습모델을 설계하였다[12].

한 학기 동안 진행되는 수업은 이론학습 및 실습 모듈과 프로젝트 진행모듈로 나누어 구성된다. 이론학습

및 실습 모듈에서는 1주~11동안 기초이론을 학습하고 예제 및 응용문제를 실습한다. 프로젝트 진행모듈은 12주~15주 까지 진행되며 프로젝트의 계획, 토론, 개발과정을 거쳐 결과물을 완성하며 마무리 된다.

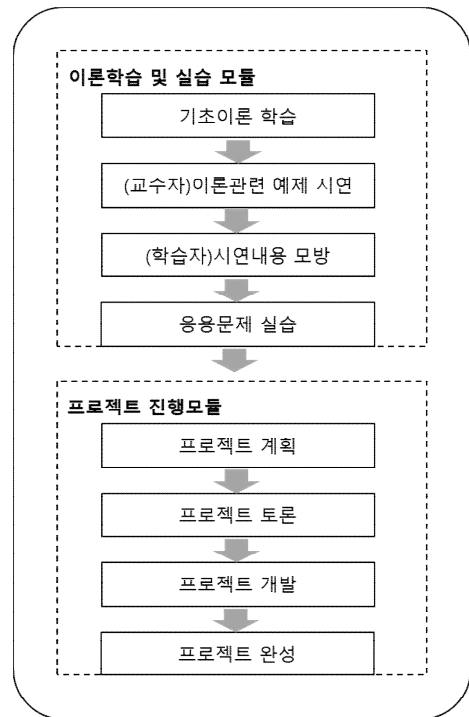


그림 1. 학습모델  
 Figure 1. Learning model

### 1. 프로젝트 진행모듈 설계

프로젝트 진행모듈은 그림 2와 같이 1단계~4단계로 구성하였다. 1단계~3단계 과정은 교수자-학습자 피드백 과정을 적용하여 진행하고 4단계에서는 학습자 중심으로 프로젝트 과정을 마무리 하게 된다.

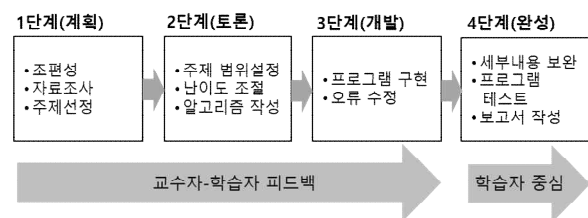


그림 2. 프로젝트 진행모듈 단계  
 Figure 2. Project progress module phase

1단계는 계획단계로 프로젝트 수행을 위한 조편성 후 자료조사를 통해 주제를 선정한다. 2단계에서는 프

로젝트 목적에 맞게 선정한 주제의 개발 범위 및 구현 기능의 난이도를 토론을 통해 조절하여 알고리즘을 작성한다. 3단계 개발과정에서는 이전단계에 계획한 알고리즘에 따라 입·출력부, 변수설정, 제어구조 및 함수기능 디자인 등 구체적인 내용을 프로그램으로 구현하고 실행해 보며 오류가 있는 경우 수정한다. 4단계에서는 프로젝트 내용의 완성도를 높이고 세부내용을 보완하며 프로그램을 테스트 한다. 보고서는 개발된 프로그램에 설명, 알고리즘, 수행소감, 참고문헌 항목에 대한 내용으로 작성한다.

2. 교수자-학습자 피드백 내용 및 진행방법

프로젝트 진행모듈의 1단계~3단계에서는 교수자-학습자간 피드백을 통해 토론을 진행한다. 이때, 교수는 표 1과 같은 피드백용 체크리스트 항목에 대하여 조별 프로젝트 진행내용을 확인하며 상황에 맞는 적절한 피드백을 제공한다.

프로젝트를 계획하는 1단계에서는 프로젝트 진행을 시작하기 전 조편성을 한다. 이때, 교수는 학습자의 관심사나 전공분야에 따라 조를 구성하고 상호보완적으로 협업하며 역할을 분담 할 수 있도록 피드백 해준다. 자료조사 항목은 주제선정을 위한 사전조사 작업으로 컴퓨터 프로그램을 이용한 프로젝트 작업의 사례를 찾아보게 한다. 주제선정 부분에서는 학습자의 관심분야나 전공지식과 관련된 융합기술에 대해 생각하며 해결해야 할 문제를 정의 할 수 있도록 교수가 피드백 해준다.

조별로 토론을 진행하는 2단계의 주제 범위 설정 항목에서는 개발 범위가 광범위한 경우와 너무 좁게 잡은 경우를 조율하며 추상적인 주제에 대해 구체화 할 수 있게 한다. 난이도 조절 항목에서는 학습한 내용 범위 내에서 실현가능한 기능으로 한정지어 기본 기능을 구현한 프로토타입 프로그램을 완성하게 하여 프로젝트 작업의 지연이나 완성도가 저하되지 않게 교수가 피드백 해준다. 알고리즘 작성부분에서는 개발 내용에 대하여 순서도를 이용하여 프로그램의 전체적인 내용을 구조화 시킬 수 있게 한다. 동일한 패턴이 반복되는 구조인 경우 부가적인 설명을 추가하여 함축적으로 정리 할 수 있게 한다.

프로그램을 개발하는 3단계에서는 기본적인 변수 선언에서부터 반복문 및 함수를 다양하게 사용하여 프로

그램의 간결성과 기능성 향상 시킬 수 있도록 피드백 해준다. 프로그램의 주요기능에는 주석처리를 하여 가독성 및 작업효율을 높일 수 있게 한다. 코딩작업 중 오류가 발생하는 경우 오류 메시지를 참고하여 수정 할 수 있게 유도하고 특정기능이 실행되지 않는 등 코딩내용의 복잡도가 높아 오류가 발생하는 경우 기능별로 분리하여 테스트 후 오류를 수정하고 다시 통합 할 수 있게 피드백 해준다.

표 1. 피드백용 체크리스트  
Table 1. Checklist for feedback

단계	항목	체크 리스트
1	조편성 및 자료조사	조구성원의 역할분담
		교재 또는 인터넷 검색을 통한 자료조사
	주제선정	관심분야에 대해 생각하기
		전공과 관련해서 자동화하고 싶은 항목 생각하기
2	주제 범위 설정	추상적인 주제에 대해 구체화하기
		주제의 범위 중에서 실현가능한 한계치 설정
	난이도 조절	학습한 내용, 작업시간을 고려하여 프로토타입 프로그램 구현을 위한 기본적인 기능으로 한정
		알고리즘 작성
3	프로그램 구현	변수 설정, 반복문 및 함수를 사용하여 프로그램의 간결성과 기능성 향상
		주석처리로 코딩내용의 가독성 높이기
	오류 수정	코딩작업 중 발생하는 오류 해결방법 제시
		오류 메시지 참고를 통한 수정방법 가이드

III. 연구결과

본 연구에서는 교양필수 과목으로 소프트웨어 교과목을 수강하는 컴퓨터 비전공 1학년 학습자를 대상으로 선정하였다. 표 2는 경기도에 있는 K대학에서 2018년도에 교양필수 과목으로 ‘소프트웨어기초’ 교과목을 수강하는 학생의 소속 및 분포를 나타낸다. 학습자들의 소프트웨어관련 교과목 수강 여부에 대한 조사결과 전체 282명 중 23명(8.15%)의 학생들이 중·고등학교 때 배웠거나 개인적으로 공부한 경험이 있었고, 이 학습 경험자들의 프로그래밍과 관련된 프로젝트 경험은 없는 것으로 나타났다.

2018년도 1학기에는 지식정보서비스 대학에 소속된

학생들에게 학습자 중심으로 프로젝트를 진행하게 하였고, 2학기에는 휴먼인재융합대학에 소속되어 있는 학습자에게 본 연구에서 제안한 교수자-학습자 피드백 방법을 적용하여 프로젝트를 진행하게 하였다.

표 2. 2018년도 수강학생 분포  
 Table 2. Distribution of students in 2018

학기	대학형태	소속	학생 수(명)
1	지식정보 서비스대학	법학과, 경상계열, 휴먼서비스학부, 행정학과, 경찰행정학과, 국제관계학과	152
2	휴먼인재 융합대학	유아교육과, 어문학 계열, 사학과, 문헌정보학과, 문예창작학과, 예체능 계열, 건축공학과	130

교과목에서 다른 프로그래밍 언어는 문법이 간결하고 가독성이 좋은 파이썬(Python)을 사용하였다[13].

1주~11주차의 강의진행 방식은 표 3과 같은 학습내용에 해당되는 이론 설명과 예제 실습을 병행하였고 응용문제를 스스로 작성하는 시간도 가졌다.

프로젝트 작업은 12주~14주 동안 교수자-학습자의 피드백을 적용하여 진행하고 15주차에는 학습자 중심으로 프로젝트를 완성하여 최종 결과물(개발 프로그램 파일, 보고서)을 제출하게 하였다.

표 3. 파이썬 프로그래밍 강의 범위 및 내용  
 Table 3. Python programming lesson scope and content

학습범위	학습내용
파이썬 기초	변수의 이해, 입출력 함수
연산자, 자료	산술, 나머지, 지수연산자, 자료유형, 리스트
수식 활용	연산자 우선순위, 관계/논리 연산자
제어문	if/if-else문, 조건식, 블록
반복문	while/for문, break
함수	함수의 정의 및 호출

### 1. 프로젝트 내용에 대한 창의성, 기능성, 완성도 측면 평가

본 수업에서 진행하는 프로젝트의 학습 목표는 아래와 같다.

- 실생활이나 학생들의 전공분야와 연관된 내용으로 프로그램을 설계하고 직접 작성하며 창의적인 사고력을 키운다.

- 한 학기동안 배운 내용을 다양하게 적용하여 컴퓨터 프로그래밍에 대한 기초지식을 활용해본다.

이와 같은 학습목표에 준하여 프로젝트 내용에 대한 평가 및 분석은 객관성과 타당성을 위해 인문, 사회, 예체능, 컴퓨터 교육 분야의 전문가 4명의 논의를 통해 진행하였고 평가항목은 다음과 같다.

- 창의성 : 실용성 등을 고려한 참신한 아이디어의 포함 여부
- 기능성 : 수업시간에 배운 내용을 다양하게 적용하였는지 여부
- 완성도 : 프로그램의 모든 항목에 대한 정상적인 동작 여부

그림 3은 프로젝트 수행 시 피드백을 미적용(2018학년 1학기)한 경우와 피드백을 적용(2018학년 2학기)한 경우의 결과물에 대한 평가결과이다. 각 항목을 10점 만점기준으로 평가한 점수의 평균을 비교해보면 창의성, 기능성, 완성도 부분에서 교수자의 피드백을 적용하여 프로젝트를 수행한 경우에 평가점수가 상승된 것을 알 수 있다. 특히, 기능성 부분에서는 동일한 조건문 형식으로 구성하거나 반복된 패턴의구조인 경우 피드백을 통해 특정기능 부분을 함수로 모듈화하고 다수 개의 데이터 입력에 리스트를 이용 할 수 있게 하는 등 수업시간에 배운 내용을 다양하게 활용 할 수 있도록 피드백이 적용되었기 때문에 피드백을 받지 않은 경우에 비해 수업시간에 배운 내용을 다양하게 활용한 결과물이 도출된 것으로 판단된다. 완성도 부분에서도 미완성 결과물이 있었던 피드백을 미적용 했을 때와는 달리 구현 범위 조정, 충분한 테스트 작업 등 교수자-학습자의 피드백을 적용한 경우가 모든 학습자들이 완성된 상태로 프로젝트를 마무리한 것을 확인 할 수 있다.

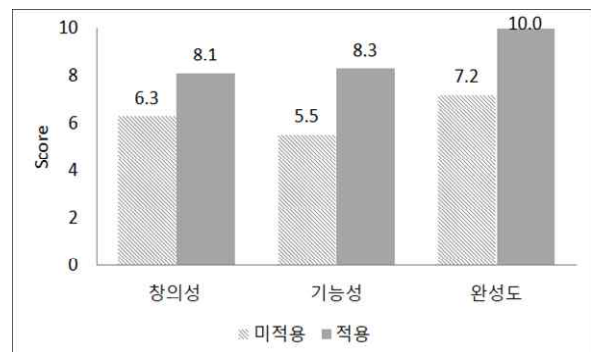


그림 3. 프로젝트 내용의 평가결과  
 Figure 3. Evaluation result of project contents

2. 프로젝트 수행 후 프로그래밍 작업에 대한 학습자 의견

기말 프로젝트 수행 후 프로그래밍 작업에 대한 학습자 의견에 대해 조사하였고, 질문 항목은 아래와 같이 2가지로 구분하여 질의 하였다.

- 프로그래밍 능력 변화에 대한 견해
- 프로젝트 수행과정에서 어렵다고 느낀 점

첫째, 프로젝트 수행 후에 느낀 본인의 프로그래밍 능력변화를 ‘향상’, ‘보통’, ‘미흡’으로 나누어 조사하였으며, 그 결과는 그림 4와 같다. 학생들이 자신의 프로그래밍 능력이 향상되었다고 느끼는 경우는 교수자-학습자 피드백을 적용한 경우가 40.0%로 미적용한 경우보다 9.1% 높게 나타났다. 반면, 보통이라는 의견은 피드백을 미적용 했을 때 5.5% 높게 나타났지만 미흡하다는 의견은 피드백을 적용 후 3.5% 감소하는 현상을 보였다. 즉, 학습자 중심으로 진행된 1학기에 비해 교수자-학습자 피드백 방법을 적용한 2학기에 학습자가 프로그래밍 능력이 상승되었다고 체감하는 비율이 높아진 것을 확인 할 수 있다. 이 결과를 보면, 교수자-학습자 피드백 방법을 적용하여 프로젝트를 수행했을 때 토론을 통해 작업내용을 체계화 및 구체화시키며 완성해 가는 과정에서 성취감이 상승되었음을 알 수 있다.

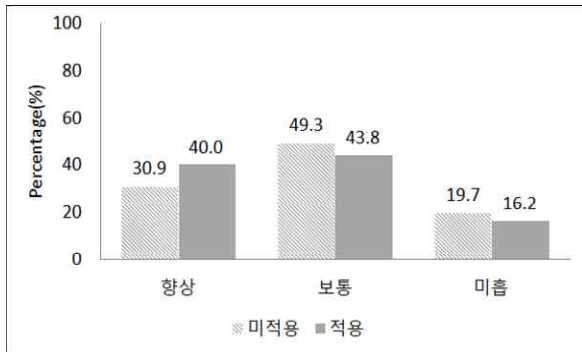


그림 4. 프로그래밍 능력 변화에 대한 견해  
Figure 4. View on programming ability changes

둘째, 프로젝트 수행 과정에서의 어려웠던 점에 대한 의견은 표 4와 같으며 각 항목이 중복으로 선택된 경우도 있었다. 주제 선정, 알고리즘 작성, 코딩 작업에 대하여 교수자-학습자 피드백 방법을 미적용한 경우가 적용했을 때 보다 많은 어려움을 느낀 것으로 답변하였다. 이 결과를 보면, 학습자가 직접 주제선정하고 설계 및 코딩하는 과정을 처음 접해왔기 때문에 학습자 중심

으로 진행했을 때보다 교수자의 피드백을 받으며 다양한 관점에서 주제를 선정하고 프로젝트 설계를 시도하면서 프로그래밍 작업을 진행해 나가는 것이 프로젝트 작업에 도움이 되었을 것으로 판단된다. 또한, 코딩 작업부분에서도 오류수정이나 구현하는 내용을 확장할 경우 교수자의 피드백을 통해 해결점을 찾을 수 있었기 때문에 교수자-학습자 피드백 방법이 코딩 작업에 긍정적인 영향을 미친 것으로 사료된다. 하지만 알고리즘 작성 항목에서는 교수자-학습자 피드백 방법을 적용한 경우에도 어려움을 느끼는 학습자의 수가 크게 낮아지지 않았다. 그 이유는 주제선정 후 알고리즘을 계획하는 단계에서 학습자가 너무 많은 문제를 동시에 고려하기 때문에 이를 알고리즘으로 표현하는 것이 복잡해서이다. 따라서 학습자가 정해진 시간 내에 완성할 수 있도록 학습자의 스타일이나 작업역량에 맞게 개발범위를 결정 후 프로그램 구조를 알고리즘화 할 수 있도록 지도하는 교수자의 피드백이 필요한 것으로 보여진다.

표 4. 프로젝트 수행과정에서 어렵다고 느낀 점  
Table 4. The difficulties in the process of carrying out project

항 목	미적용(명)	적용(명)
주제 선정	92	51
알고리즘 작성	70	66
코딩 작업	56	45

IV. 결 론

소프트웨어 교과목을 수강하는 학습자는 수업시간에 이론학습 및 실습을 통해 프로그래밍 기법을 익히고 프로젝트 작업을 통해 문제의 탐구에서부터 해결까지의 과정을 복합적으로 다루어 보며 잠재된 능력의 표출과 폭넓은 학습을 경험하게 된다. 컴퓨터 비전공자가 소프트웨어 프로젝트를 처음으로 수행을 하는 경우 학습자 중심으로 프로젝트를 진행하기에는 많은 어려움이 있기 때문에 학습자의 역량에 맞는 교수자의 피드백이 필요하다. 따라서 본 연구에서는 한 학기동안 학습한 내용을 기반으로 수행하는 소프트웨어 교과목의 프로젝트 작업에서 기존의 학습자 중심으로만 진행하였던 방식과는 달리 교수자-학습자 피드백 방법을 설계하여 적용 한 후 그 효과를 프로젝트 내용 평가와 학습자 의견 분석의 결과를 통해 제시하였다. 본 연구에서 제안한 교수자-학습자 피드백 방법을 적용하여 프로젝트를

수행한 결과 미적용 했을 때 보다 프로젝트 내용에 대한 창의성, 기능성, 완성도 측면의 평가점수가 상승된 것을 확인 할 수 있었다. 또한, 교수자-학습자 피드백 방법을 통해 프로젝트 수행 후 프로그래밍 작업에 대한 학습자 의견 조사에서도 프로그래밍 능력이 향상되었다는 견해가 증가되었고 아직 미흡하다는 의견은 낮아진 것으로 확인되었다. 그리고 프로젝트 수행과정에서 어렵다고 느끼는 주제 선정, 알고리즘 작성, 코딩 작업에서도 교수자의 피드백을 적용했을 때 도움이 되는 것을 알 수 있었다. 향후에는 학습자의 성향 및 전공 지식에 최적화된 맞춤형 피드백 방식을 개발하여 컴퓨터 비전공 학생들이 보다 효율적으로 소프트웨어 프로젝트 작업을 할 수 있게 하는 방법을 소개 할 계획이다.

## References

- [1] J. Y. Park & Y. H. Chang, "Study on Arduino Kit VR contents modularization based on virtualization technology in software education field," The Journal of the Convergence on Culture Technology (JCCT), Vol. 4, No. 3, pp. 293-298, 2018.  
DOI: <http://dx.doi.org/10.17703/JCCT.2018.4.3.293>
- [2] H. Jung, "Finding the Research Possibilities of Computer Technologies in Art Education," International Journal of Advanced Culture Technology, Vol. 6, No. 2, pp. 51-57, 2018.  
DOI: <http://dx.doi.org/10.17703/IJACT.2018.6.2.51>
- [3] B. E. Penpras, The Fourth Industrial Revolution and Higher Education, Palgrave Macmillan, pp. 207-229, 2018.
- [4] S. Papert, Mindstorms: Children, Computers, And Powerful Ideas, Basic Books Pub, 1980.
- [5] J. Wing, "Computational thinking," Communications of the ACM, Vol. 49, No. 3, pp.33-35, 2006.
- [6] Y. Tabesh, "Computational thinking," Olympiads in Informatics, Vol. 11, pp.65-70, 2017.
- [7] <https://cs50.harvard.edu>
- [8] <https://www.cs.columbia.edu/education/courses>
- [9] <https://bjc.berkeley.edu>
- [10] J. Nah, Computational thinking as basic liberal arts education, The 10th joint deliberation for professors and faculty members in charge of

liberal education, Korea National Institute for General Education, pp. 19-27, 2017.

- [11] S. Lee & J. Kim, "A Study of SW Education for Non-Majors in Sungkyun SW Education iNstitution(SSEN)," Proceedings of the Korea Association of Computer Education, pp. 107-109, 2017.
- [12] J. Kim et al., A Research on the Development of Teaching and Learning Models for SW Education, Korean Educational Development Institute, 2015.
- [13] <https://www.python.org/about>

※ 본 연구는 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2015R1D1A1A01061064)