

디지털 트랜스포메이션을 위한 프로그래밍 교육 - 무엇을 어떻게 가르쳐야 하는가?

박수진*

<목 차>

- I. 서론
- II. 국내외 기술경영 관련 (전문)대학원 커리큘럼 비교를 통한 IT 관련 교육의 동향
- III. 서강대학교 기술경영전문대학원의 IT 관련 커리큘럼 리노베이션 사례
- IV. 결론 및 향후 계획

국문초록 : 디지털 트랜스포메이션은 기업의 경영 전략적 관점에서의 조직, 프로세스, 비즈니스 모델, 커뮤니케이션의 광범위하고 근본적 변화를 요구하고 있으나, 성공적인 트랜스포메이션의 핵심 선제조건 중 하나는 구성원의 IT(Information Technology) 역량이다. 공학적 배경 지식 없이 기술경영을 전공하는 학생들은 비교적 단기간에 경영관리 능력과 IT 역량을 동시에 키워야 하는 난제에 처해 있다. 이러한 이유로 효과적인 IT 역량 교육 방법은 기술경영학 분야의 교육 이슈 중 하나로 부각되고 있다. 본 논문은 디지털 트랜스포메이션의 흐름을 주도할 수 있는 학생들을 양성하기 위한 IT 역량 교육은 “무엇”을 “어떻게” 가르쳐야 할 것인가라는 질문에 대한 답을 찾기 위한 하나의 사례로서, 서강대학교 기술경영 전문대학원의 IT 관련 커리큘럼 리노베이션의 과정과 현재까지 교육과정 실행 결과를 제시한다. 특히, IT 역량 교육의 출발점이자 진입장벽이 되는 기초 프로그래밍 교육 과정의 피드백 결과에 대한 충분한 논의를 통해, 기술경영대학원 학생들을 위한 효율적인 IT 교육의 운영 방향에 대한 선행 참조사례를 제공하는 데 본 논문의 목적이 있다.

주제어 : 기술경영 교육, 디지털 트랜스포메이션, 비전공자 프로그래밍 교육

* 서강대학교 기술경영전문대학원 조교수(psjdream@sogang.ac.kr)

Programming Education for Digital Transformation - What to Teach and How?

Soojin Park

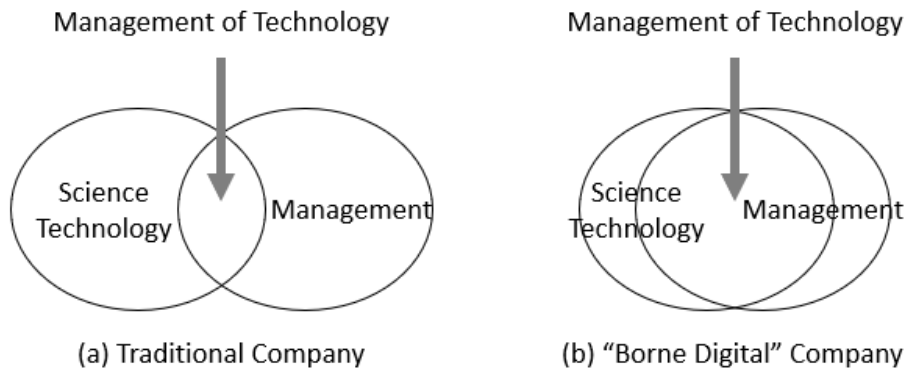
Abstract : Digital transformation demands significant and foundational alterations in an organization's structure, processes, business strategy, and communication from a managerial perspective. However, one of the crucial prerequisites for a successful transition is the digital proficiency of its members. Students studying Management of Technology without a prior background in engineering face the difficult challenge of improving both their business management skills and digital competency within a limited time frame. For this reason, effective IT competency education methods are emerging as one of the educational issues in technology management. This paper showcases the IT-focused curriculum revision process of Sogang University's Graduate School of Management of Technology, along with the outcomes of its implementation thus far, as a demonstration of addressing the questions of "what to teach" and "how to teach" for students who could potentially become leaders in guiding the digital transformation. The purpose of this paper, specifically, is to offer a prior reference for the operation direction of effective programming education for students in Management of Technology graduate schools through a discussion on the feedback results of the required fundamental programming course which acts as both the starting point and entry hurdle.

Key Words : Management of Technology education, digital transformation,
programming education for non-majors

I. 서론

1987년 National Research Council에 의해 제안되었던 기술경영학의 태동기에 기술경영학을 포지셔닝하는 하나의 확고한 관점은 <그림1>에 표기된 바와 같이, 과학기술과 경영의 합병자(merger) 역할을 담당하는 학문 분야라는 것이었다. 비슷한 시기에 “기술경영학에서 무엇을 교육해야 하는가”라는 질문의 답으로, Chanaron, J. and Jolly, D. (1999)는 “기존의 공학과 경영학의 단과대학에서 제공되는 극도로 전문화된 교육을 제외한, 경영대학의 충분히 기술지향적인 학문과 공과대학의 충분한 경영교육을 바탕으로 한 통합적이고 체계적인 접근”을 요구하였다. 그러나 “누구를 대상으로 기술경영학을 교육하여야 하는가”에 대해서는 Badawy (1998)는 그 대상자를 “기술자(technologists)”의 범주로 한정하였다.

지난 20 여년간 페이스북, 구글, 알리바바와 같은 ‘태생적인 디지털’ 기업들의 등장과 성공으로, 기술경영학의 태동기에 연구자들의 해답은 수정되어야 할 소지가 있어 보인다. 전통적인 형태의 기업에서는 <그림1>의 (a)가 도식화하는 바와 같이, 과학기술과 경영의 고유 범위가 명확했으며 그 접점 영역은 제한적이었다. 반면, ‘태생적인 디지털’ 기업의 경우엔 <그림1>의 (b)가 보이는 바와 같이, 과학기술과 경영의 영역이 불분명할 뿐만 아니라, 대부분의 경영 이슈가 과학기술의 이슈와 중첩된 양상을 보인다.



<그림 1> 디지털 기업의 등장에 따른 기술경영의 영역 변화

기업 환경의 변화는 과학기술과 경영의 접점에서 합병자로서의 역할을 담당하는 기술경영학이 포괄하는 영역이 확장되고 있음을 의미한다. 뿐만 아니라, COVID 19 팬데믹

위기는 디지털 기술을 통해 전략과 조직 방식을 혁신적이고 경쟁력 있게 전환해야 할 필요성을 가속화시켰다(여영준, 2020). 이러한 배경에서 디지털 트랜스포메이션이라는 새로운 패러다임 역시 조직의 유기적인 혁신을 요구하고 있으며, 성공적인 디지털 조직으로의 전환을 위해서는 유연한 디지털 기술 활용 능력이 필수 불가결한 요건 중 하나이다.

이와 같은 시대적 요구사항과는 상반되는 현상은 국내 기술경영 전문대학원에 진학하는 학생들 중 “기술자”의 비중은 감소하고 있다는 점이다. 결국, 국내 기술경영 전문대학원의 커리큘럼을 통해, “기술자”가 아닌 학생들을 대상으로 점점 더 비중이 확장되는 과학기술 영역의 역량을 키워야 하는 도전적인 상황에 직면하고 있다. 과학기술 역량, 그 중에서도 IT 역량은 결국엔 프로그래밍의 결과물인 디지털 서비스/제품의 형태로 표출된다. 물론, IT 역량을 갖추는 것이 곧 프로그래머가 됨을 의미하는 것은 아니다. 그러나, 기업 구성원의 근원적인 프로그래밍 원리에 대한 이해 정도는 최종 디지털 서비스/제품 품질에 직간접적으로 영향을 미치는 요소가 된다. 결과적으로, 디지털 트랜스포메이션이라는 새로운 사회적 패러다임을 주도하는 졸업생 양성의 필요 조건으로, 기술경영의 학제 내에서 선행 경험이 적은 학생들의 IT 역량을 배가시킬 것이 요구되어지고 있다.

본 논문은 Chanaron, J. and Jolly, D. (1999)가 언급한 “경영대학의 충분히 기술 지향적인 학문과 공과대학의 충분한 경영교육을 바탕으로 한 통합적이고 체계적인 접근”을 디지털 트랜스포메이션의 흐름에 부합하도록 커리큘럼 상에 어떻게 반영해 내야 할 것인가라는 질문에 답을 찾기 위한 하나의 사례를 소개하고자 한다. 서강대학교 기술경영 전문대학원(이하 서강 MOT로 지칭)은 2016년 이래로 6년여에 걸쳐, 기초 프로그래밍에서부터 인공지능 관련 기술까지의 IT 교육을 시행해 왔다. 2022년 1학기부터 그간의 IT 교육 경험에서 겪은 시행착오를 바탕으로, 새로운 커리큘럼을 설계하여 실행 중인 사례 소개를 통해 다음과 같은 두 가지 이슈에 대해 논의하고자 한다.

1. 융합학문 분야인 기술경영학의 범주 내에서, 프로그래밍 교육은 디지털 트랜스포메이션을 주도하는 인재 양성을 위한 필요조건인가?
2. 프로그래밍으로부터 출발하는 디지털 트랜스포메이션 트랙의 성공을 위해 기술경영 전문대학원의 교육 기간과 자원의 제한을 극복하기 위해 필요한 전략은 무엇인가?

위의 질문에 대한 답을 도출하기 위한, 본 논문의 구성은 다음과 같다. 서강 MOT의 커리큘럼 리노베이션(renovation) 사례를 설명하기에 앞서, 2장에서 국내외 주요 MOT 관련 (전문)대학(원)의 IT 관련 개설과목 구성 현황 및 특징을 살펴본다. 3장에서는 서강 MOT에서 설정한 IT 관련 과목 커리큘럼 리노베이션의 전략이 무엇이었으며, 설정된

전략에 따라 구성된 새로운 커리큘럼의 특징을 소개하고, 마지막으로, 기초 프로그래밍 과목을 수강한 학생들의 피드백과 설문 결과를 통해, 리노베이션된 커리큘럼의 효용성을 논의한다. 4장에서는 사례분석 결과를 토대로 기술경영학의 범주에서 디지털 관련 교육의 운영 방향에 대한 제언을 제시한다.

Ⅱ. 국내외 기술경영 관련 (전문)대학원 커리큘럼 비교를 통한 IT 관련 교육의 동향

1. 국내 기술경영 관련 (전문)대학원 커리큘럼 비교

서강대학교 MOT에서는 커리큘럼 리노베이션에 앞서, 국내 주요 기술경영(전문)대학원들의 커리큘럼에 포함된 IT 관련 과목들을 조사하였다. 컴퓨터 과학 분야의 세부 전공 과목들 중에서 IT 관련 과목으로 범위를 제한한 이유는, 2000년대 이후부터, 이미 기술경영학에서 경영의 대상이 되는 기술 중 가장 많은 부분을 차지하는 기술 분야가 전기, 컴퓨터 등과 관련한 정보 기술 분야(Nambisan & Wilemon, 2003)이기 때문이다. <표 1>은 2022년 1월 기준, 각 대학원의 홈페이지에 게재된 커리큘럼을 참조하여 수집된 개설과목 현황이다. <표 1>에 발췌한 과목들은 그 명칭만으로 판단하여 IT 관련 과목으로 선택된 것으로, 개설된 과목의 교육 내용을 세밀히 분석하여 IT 관련 과목으로 선택된 것은 아니다. 따라서, 과목명이 아닌 세부 교과 내용에 대한 접근이 가능했다면, 더 많은 과목들이 해당될 수도 있음을 밝힌다.

고려대학교 기술경영전문대학원의 커리큘럼에서는 기술금융 트랙에서 핀테크 및 블록체인 분야에 특화된 과목들이 다수 개설되어 있는 점이 특징적이다. 서울대학교 데이터사이언스 대학원의 경우에는, 거의 일반적인 전산학과와 유사하게 다양한 자료구조(Data Structure) 과목들로 커리큘럼이 구성되어 있었다. <표 1>에 나열된 과목들은 자료구조 관련 기초 과목을 선수하지 않은 학생들을 위해, 데이터 분석의 기초역량을 함양하는 과정인 기초교과목 트랙에 속하는 과목들만을 발췌한 것이다. 서강대학교 기술경영 전문대학원은 기존 커리큘럼 상에도 인공지능(Artificial Intelligence: AI) 및 프로그래밍과 직접적으로 관련된 다수의 과목들이 개설되어 있음을 확인할 수 있다.

〈표 1〉 국내 주요 기술경영전문대학원 커리큘럼 비교

기술경영전문대학원명	IT 관련 주요 개설과목
고려대학교 기술경영전문대학원	<ul style="list-style-type: none"> - 프로젝트 관리 - 시스템 공학 - 딥러닝의 이해 - 산업과 인공지능의 이해 - 인공지능을 위한 머신러닝 - 플랫폼 전략과 핀테크 - 블록체인과 투자 - 블록체인 사업화 - 블록체인 트랜스포메이션
서울대학교 데이터사이언스대학원	<ul style="list-style-type: none"> - 데이터사이언스를 위한 확률과 통계 - 데이터사이언스를 위한 컴퓨팅 - 데이터사이언스 원론 - 데이터사이언스원리와 응용
서강대학교 기술경영전문대학원	<ul style="list-style-type: none"> - 기술경영을 위한 프로그래밍 실습 - 빅데이터 분석을 위한 컴퓨팅 실습 - 빅데이터 컴퓨팅 프로젝트 - 인공지능 I - AI기반 산학연계 프로젝트(인공지능 II) - 산학연계 프로젝트 I - 의사결정 지원을 위한 인공지능의 활용 - 핀테크의 성장과 미래
성균관대학교 기술경영전문대학원	<ul style="list-style-type: none"> - 기술정보시스템
KAIST 기술경영학부 대학원	<ul style="list-style-type: none"> - 전산응용 개론 - 게임 이론과 응용
한양대학교 기술경영전문대학원	<ul style="list-style-type: none"> - 제조지능화 - 디자인 기술경영 - 인공지능과 고객형디자인 - 비즈니스 정보 시각화 - 디지털 문화와 조직 - 헬스케어 이노베이션 - 비즈니스 AI - 머신러닝과 기술예측 - 디지털 서비스 혁신

성균관대학교 기술경영 전문대학원에는 교과목 명칭으로만 확인할 수 있는 IT 관련 과목으로는 기술정보시스템, 한 과목이 개설되어 있었다. KAIST 기술경영학부 대학원의

경우에는 기본 과목인 전산응용개론 과목과 게임이론과 응용 과목이 개설되어 있다. 마지막으로, 한양대학교 기술경영 전문대학원의 경우엔, 기술디자인(Technology Design: TD) 트랙을 별도로 운영하며 사용자 경험(User Experience: UX) 디자인 및 예술, 헬스케어 분야와의 융합 과목들을 개설하고, 그 외에 머신러닝과 관련한 과목들도 개설되어 있다.

국내 주요 기술경영 (전문)대학원의 커리큘럼(2022년 1월 기준)을 수집/분석한 결과는 다음과 같이 요약할 수 있다. 첫째, 기술경영학의 범위 내에서 IT와 관련한 필수 과목 정립은 되지 않은 상태로 판단된다. 둘째, 각 대학원이 속한대학이 이미 선점하고 있는 IT 분야에서의 선도적인 분야가 각 기술경영 대학원의 커리큘럼에 반영되는 경향을 보인다. 셋째, 외부에 오픈된 과목 명칭을 통해 실제적인 프로그래밍 교육 과목 개설이 확인되는 대학(원)은 서강대 기술경영대학원과 서울대 데이터사이언스 대학으로 다소 한정적이다.

2. 국외 기술경영 관련 대학원 커리큘럼 비교

기술경영학의 태동 시기부터 학제적 기틀을 닦아 온 미국과 영국의 선도적인 대학들에서 현재 개설되고 있는 IT 관련 과목을 조사하여, 대학별 커리큘럼의 특징을 비교하였다. 먼저, UC Santa Barbara의 Technology Management program의 커리큘럼을 살펴보면, “프로그래밍 기초(Programming Fundamentals)”와 “객체 지향 설계 및 프로그래밍(Object-Oriented Design and Programming)” 등의 기초 프로그래밍 과목부터 데이터 구조, 알고리즘 및 소프트웨어 설계 원칙을 포함하여 컴퓨터 프로그래밍 개념 및 기술에 대한 기초 과목들이 다수 포함되어 있다. 뿐만 아니라, 소프트웨어 개발 방법론, 프로젝트 관리, 컴퓨터 프로그래밍 분야의 신기술과 같은 주제도 커리큘럼에 포함되어 있다.

Massachusetts Institute of Technology (MIT) Sloan School of Management에서는 별도의 Technology and Operations Management (TOM) 프로그램을 운영하고 있다. TOM 프로그램에서 학생들은 소프트웨어 엔지니어링, 정보 기술 및 데이터 분석과 같은 주제를 다루는 과목들을 선택하여 수강할 수 있다. MIT TOM 프로그램의 커리큘럼 상의 특징은 소프트웨어 개발 및 설계의 기초를 다루는 “소프트웨어 공학(Software Engineering)”, “데이터 과학 및 분석(Data Science and Analytics)”, “소프트웨어 프로젝트 관리(Software Project Management)” 등의 과목들이 포함되어 있다는 점이다.

UC Berkeley Haas School of Business의 MOT 프로그램 커리큘럼은 MIT와 마찬가지로, “소프트웨어 공학 개론(Introduction to Software Engineering)”, “데이터 과학 및 분석(Data Science and Analytics)” 등의 소프트웨어 개발의 기초를 다지는 과목들이 포함되어 있다. 그 외에도 특징적인 점은 다양한 소프트웨어의 도메인 군별로 프로그래밍 능력을 배양할 수 있는 실습 과목인 “데이터베이스 시스템(Database Systems)”, “웹 개발 방법(Web Development)”, “모바일 어플리케이션 개발방법(Mobile Application Development)” 등의 과목이 개설되어 있다는 점이다.

Havard Business School의 MOT 프로그램은 UC Berkeley Haas School of Business의 MOT 프로그램과 유사한 구성으로, “소프트웨어 공학(Software Engineering)”, “데이터 분석(Data Analytics)” “정보기술(Information Technology)”, “데이터베이스 시스템(Database Systems)”, “웹 개발방법(Web Development)”, “모바일 어플리케이션 개발방법(Mobile Application Development)”, 그리고 “데이터 과학(Data Science)” 등의 다양한 컴퓨터 프로그래밍과 관련된 주제의 과목들이 포함되어 있다.

앞서 살펴본 미국의 대학들은 MOT 과정을 MBA(Master of Business Administration) 또는 전문 석사 학위 프로그램 내의 별도 섹션의 프로그램으로 운영하고 있는 반면, 영국의 University of Cambridge, Imperial College London, London Business School, University of Sussex 등에서는 대부분이 독립적인 석사 학위 프로그램 형태로 제공되고 있다. 이러한 학제의 차이점은 커리큘럼 구성에 있어서도 차이점으로 나타난다. 영국 대학의 MOT 프로그램은 엔지니어링, 컴퓨터 공학, 정보 기술 과정은 물론 비즈니스 및 관리 과정을 함께 융합하는 형태의 보다 포괄적인 커리큘럼을 운영하고 있다.

이와 같은 커리큘럼 구성의 특징을 보여주는 사례로, University of Sussex의 SPRU(Science Policy Research Unit) 과정에 포함된 과목들을 살펴보면, “사회 과학자를 위한 프로그래밍(Programming for Social Scientists)”, “데이터 과학과 기계학습(Data Science and Machine Learning)”, “디지털 시대의 정량적 연구방법 (Quantitative Research Methods in the Digital Age)” 과 같은 과목은 그 명칭 면에서도 IT 기술과 그 기술이 활용될 문제 분야를 포괄하고 있는 양상을 보인다. 특히 “데이터 과학과 기계학습” 과목의 경우, 기계학습 알고리즘을 사용하여 사회 과학 및 공공 정책 문제를 해석하고 예측하는 모델을 구축하는 방법을 다루는 등 일반적인 기술보다는 비즈니스 문제 해결을 위한 기술을 교육하는 데 초점이 맞추어져 있다.

국내의 대학의 MOT 과정 커리큘럼 비교 결과가 시사하는 바는 다음과 같다. 국내보다 50여년 정도 먼저 MOT가 독립적인 학제로 자리 잡은 미국과 영국 주요 대학의 커리

컬럼은 IT 관련 교육에 있어서, 어느 특정한 부분에 국한되지 않은, 기초 과목에서부터 특정한 도메인의 문제 해결과 결합한 형태의 응용 과목에 이르기까지 상당히 넓은 범위를 아우르고 있다. 이러한 점은 본 논문의 서두에 던진 “기초 프로그래밍 교육이 디지털 트랜스포메이션을 주도하는 인재 양성을 위한 필요조건인가” 라는 질문에 대한 간접적인 답이 될 수 있다. 국내 MOT 관련 대학(원)의 IT 관련 커리큘럼은 참여 교수진의 구성에 따라, 대학별로 IT 교육의 일부분에 다소 치우침 현상이 있는 것으로 판단되며, 실질적인 프로그래밍 스킬 습득을 위한 기초 과목이 상당히 부족한 실정이다. 국내에 MOT 대학원이 설립된 기간이 짧았던 것을 고려하면, 이러한 폭넓은 IT 관련 커리큘럼을 수행하기에는 인적 인프라가 부족한 한계가 있다. 그러나, 인적 인프라가 두터워지기 이전이라 하더라도, 대학 내의 가용한 기존 교수진의 역량을 최대한 발휘함으로써 충분한 IT 관련 기술들을 교육하기 위한 장기적인 전략 수립이 필요한 시점이다.

Ⅲ. 서강대학교 기술경영전문대학원의 IT 관련 커리큘럼 리노베이션 사례

서강 MOT는 6년여 간의 디지털 교육의 경험과 시행착오, 새로운 패러다임인 디지털 트랜스포메이션의 시대적 요구 등을 반영하여 2022년 1학기 신입생을 대상으로 새로운 IT 관련 커리큘럼을 설계하여 적용하고 있다. 본 장에서는 IT 관련 커리큘럼 리노베이션의 전략과 새롭게 개편된 커리큘럼을 소개하고, 학생들로부터의 실제적인 피드백 자료 분석을 통해, 부분적이거나, 커리큘럼 리노베이션의 효과에 대해 논의하고자 한다.

1. IT 커리큘럼 리노베이션의 출발점

서강 MOT는 개인의 학부 전공 및 현업에서의 전문 분야 특성에 따라 요구되는 학습량의 차이를 고려하여, 인공지능 트랙을 선택한 학생들만을 대상으로 기초 프로그래밍 교육을 진행해 왔다. 기초 프로그래밍 교육 대상을 인공지능 트랙 전공자에게만 국한시켜 이수토록 한 것은 IT 기술을 활용한 문제해결을 요구하는 후속 과목의 교육 효과를 반감시키는 문제를 초래하였다. 예를 들어, IBM의 인공지능 솔루션들을 활용하여 새로운 비즈니스 모델을 설계하고 구현해야 하는 “AI 기반 산학연계 프로젝트“ 과목의 경우, 프로그

래밍을 통한 구현이 가능한 팀의 결과물과 그렇지 못한 팀의 결과물의 현저한 격차가 발생하는 문제점이 반복적으로 지적되어 왔다. 디지털 트랜스포메이션 환경에서의 문제해결 능력이란, IT 기술을 경영할 수 있는 능력과 직결되며, IT 기술의 기반을 이해하기 위해서는 컴퓨팅 사고력의 함양이 필수불가결하다. 컴퓨팅 사고력의 함양은 결국엔 기초 프로그래밍 학습과 직접적으로 연관되어 있다(한옥영·김재현, 2017).

2015년 국내 소프트웨어 중심대학 사업(소프트웨어 중심대학 사업협의회, 연도미상)이 시행된 이후, 국내 주요 대학에서는 기초 프로그래밍 과목을 전공을 불문하고 필수 과목으로 지정하여 교육하고 있다. 소프트웨어 중심 대학 사업 시행 과정을 통해 축적된 교육 경험으로부터, 비전공자 대상의 기초 프로그래밍 교육에 있어서 비전공 학생들이 느끼는 학습 과정 중의 어려움의 정도나 원인 등을 분석한 결과에 따르면, 비전공자 중 87.0%에 프로그래밍 학습에 어려움을 토로하고 있었으며(오미자, 2017), 그 원인으로는 ‘변수와 리스트의 학습, 아이디어를 생각하고 구현하는 과정, 적합한 명령어의 사용’ 등이 해당된다(김수환, 2015). 또 다른 흥미로운 연구 결과 중 하나는 서주영(2017)의 연구 결과로, 이공계열 전공자와 인문계열 전공자 간의 프로그래밍 과목의 학습 성취도에는 실제 별다른 차이가 없으나, 체감난이도 측면에서는 인문계열 전공자가 느끼는 난이도가 상대적으로 높으며, 과제해결 등에 소요되는 시간 역시 인문계열 전공자의 경우 개인적인 차이가 크다는 분석 결과를 제시하고 있다. 결국, 학부 전공이 인문계열인 학생들에게 최초의 프로그래밍 기술 습득이라는 장애물의 문턱이 더 높게 작용하고 있음을 알 수 있는 연구 결과이다.

대학 신입생을 대상으로 전공을 불문한 기초 프로그래밍 교육을 실시하는 데 따르는 어려움과 비교했을 때, 서강 MOT 학생들에게 기초 프로그래밍 과목을 예외 적용 없이 필수 이수과목으로 지정하는 것은 몇 가지 추가적인 어려움이 뒤따른다. 첫째, 대학 신입생의 경우 전일제 학생들이지만, MOT 학생들의 대다수는 직장과 학업을 병행하는 상황에서 타 과목과 비교하여 많은 학습량을 요구하는 프로그래밍 수업에 대한 저항감이 크다. 둘째, 대학 신입생의 경우 학습 과정을 받아들이는 어려움에 있어 문/이과 전공 성향의 차이점 정도가 있을 뿐이지만, MOT 학생들은 그 외에도 다양한 연령대와 경력 기간에 따른 개인적 학업 능력 편차가 더 크다는 문제점이 있다. 이러한 문제점에도 불구하고, 미국과 영국 등의 MOT 관련 프로그램 구성에 당연한 듯 포함되어 있는 기초 프로그래밍 과목을 IT 역량 강화의 시작점으로 판단하여, 서강 MOT에서는 새로운 IT 관련 과목 커리큘럼 리노베이션의 출발점으로 기초 프로그래밍 실습 과목을 공통필수 이수과목으로 지정하여 모든 학생들이 수강하도록 결정하였다. 이러한 결정에 따른 학생들의 저항 정도 및 교육 효과는 3.4절에서 논의된다.

2. IT 관련 커리큘럼 리노베이션 디자인 전략

기초 프로그래밍 실습 과목의 공통필수 과목 선정은 학생들이 느끼는 학업부담감을 최대한 낮추면서, 내부 교육 자원의 활용은 극대화시켜야 하는 과제를 부여하였다. 이러한 과제 해결을 위해, 서강 MOT는 몇 가지 커리큘럼 리노베이션을 위한 디자인 전략을 설정하였다.

2.1. 기초 프로그래밍 실습 언어: 파이썬(Python)

먼저, 기초 프로그래밍 교육에 사용되는 첫 번째 구현언어로는 파이썬이 선정되었다. 파이썬은 문법적인 제약도 비교적 적은 편이라는 점에서 프로그래밍 입문자들이 첫 번째 언어로 습득하기에 용이하다는 장점이 있다(이영석, 2018). 이와 같은 장점으로 인해, 2014년에 이미 컴퓨터 전공학과를 보유한 미국 대학 중 상위 69%가 파이썬을 프로그래밍을 요하는 수업의 구현언어로 선택하고 있었던 것으로 집계되었으며(Guo, P., 2014), 그 이후로도 여전히 최초 프로그래밍 언어로 가장 많이 선택되는 언어이다. 경영학과에서 많이 사용되는 언어로 R이 있지만, R과 비교하여 파이썬은 다양한 라이브러리 제공을 통해 빅데이터 분석이나 인공지능 분야에서의 범용성 측면에서 우위에 있다(Lasser et al., 2021). Java와 비교에서도, 첫 번째 프로그래밍 언어로 파이썬을 접하는 학생들이 더 높은 학습 동기와 자기효능감 인식을 가지고 있으며, 더 나은 학습 성취도를 얻는 것으로 입증되었다(Ling, et al., 2021). 이같은 점들을 고려하여, 서강 MOT 전체 학생을 대상으로 하는 기초 프로그래밍 실습 언어로는 파이썬이 선정되었다.

2.2. 명확한 선/후수 과목 설계 및 이수 유도

서강 MOT에서는 커리큘럼 리노베이션 이전에도 다수의 과목들이 IT 기술과 관련하여 개설되어 왔으나, 개설되는 과목의 숫자는 늘어났지만, 학생들이 마지막 학기에 수행하는 산학연계 프로젝트의 산출물 퀄리티는 그에 비례하지 않았다. IT 기술 관련 과목에 대한 선수과목 이수 요구가 권고에 그치는 정도에 머물렀기 때문에, 학생들이 선수과목을 이수하지 않고 그 이후 단계의 과목을 수강하는 사례가 상당히 많다는 것이 그 이유로 분석되었다. 인문사회 계열과는 달리 IT 관련 과목들은 이수과목의 선후가 뚜렷하다. 따라서, 요구되는 선수과목을 이수하지 않고 특정 과목을 수강할 경우, 학생들은 상당한 어려움을

겪게 되고, 학습성취도 역시 떨어지는 결과를 가져오게 된다. 비전공자의 경우, IT관련 전공자와는 달리 해당 과목의 교수자 외에는 학습에 도움을 줄 수 있는 동료가 없는 상황이 대부분이기 때문에 선/후수 과목을 무시한 임의적인 과목 수강 시퀀스는 학생들의 학습 노력에 대비한 효과를 저하시키는 요인으로 작용한다.

서강 MOT에서는 새로운 IT 관련 과목의 커리큘럼 설계에 있어서, 각 과목간의 선/후수 관계를 정확히 설계하고, 설정된 선수과목을 이수할 권고하는 것에 머무르지 않고, 수강신청 시스템에 반영하여 선수과목 미이수자의 후속과목 수강신청은 불가능하도록 설정하기로 결정하였다.

2.3. 개인 역량에 따라 차별화된 커리큘럼 패스 설계 및 제공

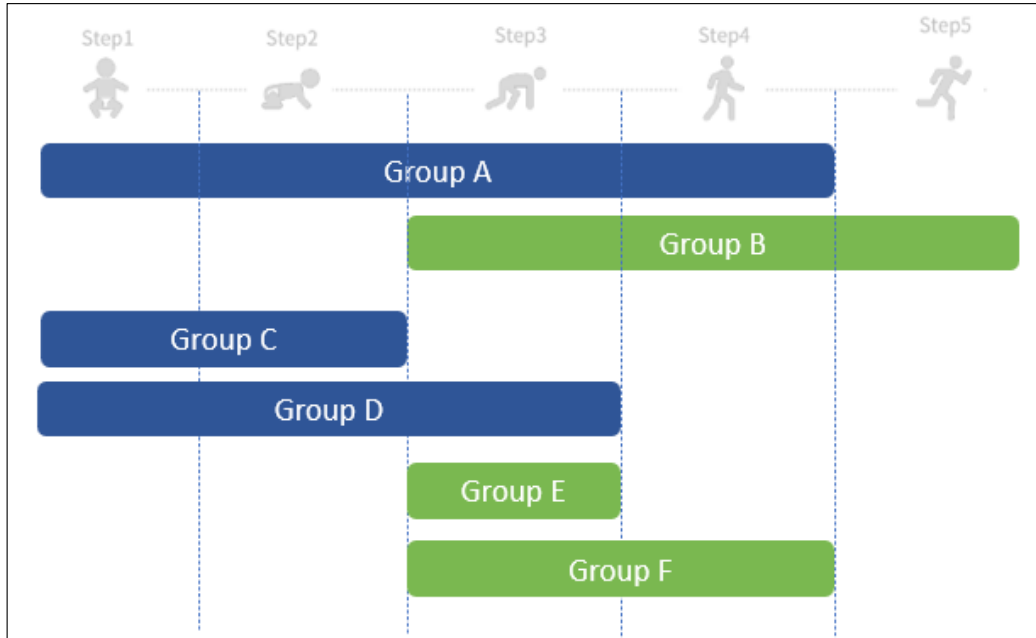
정확한 과목들 간의 선/후수 관계 설정 뿐만 아니라, 학생들의 IT 기술에 대한 개인적인 역량 차이를 반영한 차별화된 커리큘럼 패스를 설계하여 제공할 필요가 있다. 차별화된 커리큘럼 패스를 제공한다는 것은 학생들이 IT 관련 과목 이수 시퀀스에 각자가 진입하는 지점과 중단하는 지점에 대한 명확한 가이드라인을 제시함을 뜻한다.

<그림 2>에 도식화되어 있는 학생 그룹별 커리큘럼 패스는 그룹 A에서 그룹 F까지 총 여섯 개의 그룹이 존재하지만, 커리큘럼 패스의 진입점으로 구분하면 크게 두 개의 카테고리로 학생들을 분류할 수 있다. 첫 번째 그룹은 서강 MOT 입학 이전엔 기초 프로그래밍을 포함한 IT 기술 교육 경험이 전무한 학생들의 그룹이다. <그림 2>에 표시된 그룹 A, C, D에 속한 학생들로, 온라인 강의 형태로 제공되는 기초 프로그래밍에 대한 사전학습(pre-study) 단계인 스텝 1부터 커리큘럼을 이수해 나간다. 이 그룹에 속한 학생들은 개인의 학습성취도나 희망에 따라 필수 과목인 스텝 2과목까지만 이수할 수도 있고, 그 이후의 이어지는 과목들을 계속해서 수강할 수도 있다.

반면, 또 다른 그룹은 커리큘럼 진입 시점이 스텝 3부터 시작하는 그룹이다. 두 번째 그룹에 속하는 학생들은 주로 현업 개발자이거나, 개발자가 아니더라도 이전 학부 과정 등에서 이미 기초 프로그래밍 학습 경험이 있다. 이 그룹의 학생들은 기초 파이썬 프로그래밍 테스트를 통과할 경우, 스텝 2 과목을 수강하지 않고 스텝 3 과목 수강이 가능하다. 스텝 3 이후 과목의 수강은 개인의 선택에 따라 수강여부를 결정할 수 있다.

전체 커리큘럼은 총 다섯 개의 스텝으로 구성되며, 마지막 단계인 스텝 5는 AI 관련 솔루션을 기반으로 하는 산학연계프로젝트를 졸업요건으로 수행하는 단계이다. MOT 입학 이전까지 프로그래밍 경험이 전무한 그룹 A, C, D에 속한 학생일 경우, 3학기 만에

스스로 AI 솔루션을 구현하는 산학연계 프로젝트 수행한다는 것은 현실적으로 상당히 어려운 일이다. 따라서, 스텝 5의 수행은 커리큘럼 시작점이 스텝 3인 학생들만 선택하여 수행하도록 설계하였다.



<그림 2> 개인별 프로그래밍 역량 차이에 따른 차별화된 커리큘럼 패스

2.4. 오픈 콘텐츠 및 전문기업과의 협력관계를 활용한 내부 교육 자원 소요 최소화

기초 프로그래밍을 포함한 IT 관련 기술이 생소한 학생들을 3-4학기내에 비즈니스 솔루션을 구현하는 데 있어서, 능숙하게 IT 기술을 활용할 수 있는 단계까지 역량을 끌어 올리는 것은 학생들에게도 상당한 학습량을 요구하는 일이지만, 서강 MOT 내의 가용한 교육 자원만으로 필요한 과목을 모두 강의하는 것에도 분명 한계가 있다. 컴퓨터공학을 비롯한 공학계열 대학원이나 근래 들어 등장한 인공지능 대학원 등과 비교한다면, 자원의 한계는 더욱 극명하다. 그러나, MOT 대학원에서의 IT 교육의 목적은 IT 기술 그 자체라기 보다는 IT기술의 활용 능력 습득에 있다는 점에서 상대적인 교육 자원 소요 최소화 방법을 모색하였다.

새로운 커리큘럼 실현에 따라 증가하는 교육 자원의 소요를 최소화하기 위해, 서강

MOT는 IT 교육분야의 오픈 강의 콘텐츠를 적극적으로 활용하기로 하였다. 특히, 기초 프로그래밍을 처음으로 접하는 학생들을 대상으로 구현언어문법에 대한 친숙도를 높이기 위해, 서강 MOT 입학 결정 이후 개강 이전까지의 기간동안 스스로 학습할 수 있는 온라인 강의 콘텐츠를 선정하여, 스텝 1의 사전교육 단계에서 이수하도록 하였다. 온라인 강의에 대한 수강 인증을 획득한 이후에 스텝 2 과목 수강이 가능하다. 코로나로 인한 팬데믹 상황을 겪으면서 학생들은 이미 온라인 강의에 대부분 적응되어 있는 상태라, 온라인 강의 콘텐츠 활용에 대한 거부감이 없다는 점이 온라인 강의 콘텐츠를 정규 커리큘럼 시퀀스에 포함시키는 데 있어서 주요한 요인으로 작용하였다.

구현언어 문법에 대한 친숙도를 높이기 위해 스텝 1에서 오픈 콘텐츠들을 활용하여 강의 소요를 감소시킨 반면, 실제 기업에서 활용하고 있는 솔루션들을 활용하여 새로운 비즈니스 모델을 구성해 보는 프로젝트 형태의 수업인 스텝 4 과목은 전문 기업의 콘텐츠를 활용하기로 하였다. 2017년 이후 협약관계가 체결되어 있는 IBM의 솔루션들을 활용하는 프로젝트 수업의 내용을 고도화하여 학생들이 실제 현업에서 활용 가능한 기술을 습득할 수 있도록 해당 과목을 재설계하였다.



<그림 3> 온라인 오픈 교육 콘텐츠 및 전문 기업과의 협력관계 활용을 통한 내부 교육자원 소요 최소화

3. IT 관련 과목 커리큘럼 재구성안

3.2절에서 설정한 리노베이션 전략에 따라 재구성된 서강 MOT의 IT 관련 과목의 커리큘럼은 아래 <표2>와 같다. 스텝 1은 학생들로 하여금 파이썬 언어 문법이 낯설지 않도록 준비시키는 단계로, 학기 시작 이전에 무료 오픈 강좌를 스스로 수강하여 완료했음을

인증서로 확인받도록 한다. 스텝 1의 교육 목적은 파이썬으로 작성된 프로그램을 어느 정도 읽어서 이해 가능한 상태로 만드는 데 있다.

스텝 2는 앞에서 설명한 바와 같이, 서강 MOT 학생 전원이 반드시 수강해야 할 필수 과목 중 하나로, 파이썬으로 스스로 프로그램을 작성할 수 있는 단계이다. 전임교원이 강의를 담당하며, 기초 파이썬을 활용한 다양한 문제 해결과 1:1 코드 리뷰 등을 통해 학생들은 주어진 문제해결 능력을 쌓아 나가게 된다.

스텝 3는 기초 파이썬 프로그래밍이 가능한 학생들을 대상으로, 빅데이터 분석에 필요한 다양한 라이브러리들의 활용 방법을 가르치고, 이를 기반으로 간단한 데이터 분석 프로젝트를 개별적으로 수행하는 "빅데이터 분석을 위한 컴퓨팅 실습" 과목이 개설된다. 이외에도 파이썬 기반의 기계학습을 위한 패키지 활용 방법과 딥러닝을 포함한 다양한 기계 학습 모델 등을 학습하는 "의사결정 지원을 위한 인공지능의 활용" 이 스텝 3에 속한 과목이다.

스텝 3까지를 모두 수강한 학생들은 파이썬을 활용하여 빅데이터를 분석하고, 경영 실무에서의 의사결정에 인공지능 모델을 활용할 수 있다. 이후 스텝 4에서는 상용 솔루션 중 하나인 IBM의 Watson이 제공하는 기능을 활용하여, 실제적인 산업계 데이터를 대상으로 한 인공지능을 활용한 비즈니스 모델을 설계하고, 그 중 일부를 구현하는 팀프로젝트를 수행한다.

<표 2> 서강 MOT의 재구성된 IT 관련 과목 이수 커리큘럼의 5 스텝

스텝	주요 교육내용
1	프로그래밍 언어 습득 단계: 파이썬으로 작성된 프로그램을 읽을 수 있는 단계 - 무료 Open Course를 활용한 Pre-Study: 변수, 문자열, 리스트(list),사전(dictionary), 튜플(tuple), 함수, 조건문, 반복문과파일입출력을 위한 파이썬 문법을 스스로 학습한 후 인증서
2	기초 프로그래밍 실습 단계: 파이썬으로 프로그램을 작성할 수 있는 단계 - 기술영역을 위한 프로그래밍 실습: 파이썬을 활용한 프로그래밍 문제 해결, 알고리즘을 구현하기 위한 코드 작성 및 코드 리뷰
3	데이터 분석 및 인공지능 활용 기법 습득 단계: 파이썬의 패키지 를 활용할 수 있는 단계 - 빅데이터 분석을 위한 컴퓨팅 실습: 파이썬 패키지들을 활용한 데이터 수집/분석/가시화/ 검증 - 의사결정 지원을 위한 인공지능의 활용: 파이썬의 기계학습 관련 패키지 활용 방법 실습
4	상용 솔루션을 활용한 비즈니스 모델 구현 단계: IBM 솔루션 활용 - 인공지능 I: IBM Watson Studio 사용법 교육(IBM Staff) - 인공지능 II: IBM Watson Studio를 활용한 비즈니스 모델 구축 및 시스템 구현
5	프로젝트 수행 단계: 습득한 IT 역량의 실전 적용 단계 - 산학연계 프로젝트 I: 4단계까지의 습득 기술을 활용하여, 실제적인 산업계 데이터 분석 및 인공지능 모델 구축 프로젝트 수행

마지막, 스텝 5에서는 학생들은 졸업요건의 하나로 수행하는 산학연계 프로젝트를 수행하게 된다. 모든 학생들이 산학연계 프로젝트를 인공지능이나 IT 기술 구현과 연관된 주제를 선정해야 하는 것은 아니다. 스텝 2 과목만 필수 이수과목으로 지정되어 있기 때문에, 학생들은 본인의 필요나 학업 목적에 따라 자유로이 그 이후 과목의 수강을 선택할 수 있다. 스텝 5에서의 산학연계 프로젝트 수행 학생 중, 스텝 4까지를 모두 이수한 학생들은 학습한 내용의 최종 실전 활용 사례로서 인공지능이나 IT 기술 구현과 관련된 산학연계 프로젝트를 교수 지도하에 수행하고, 그 결과를 심사받게 된다.

4. 필수 과목으로 지정된 기초 프로그래밍 과목 실행 이후의 피드백 분석

3.3절에 소개한 리노베이션된 커리큘럼은 현재까지 한 학기동안 적용되었다. 새롭게 설계된 커리큘럼의 전체 스텝이 모두 실행되지 않은 현재 시점에서, 서강 MOT의 커리큘럼 리노베이션의 효과를 논하기에는 무리가 있다. 리노베이션 된 커리큘럼의 핵심은 스텝 2 과목인 “기술경영을 위한 프로그래밍 실습”의 학생 전원 필수과목 지정이었다. 따라서, 본 절에서는 MOT 학생들의 학습 부담이 가장 증가할 수 있는 결정 사항인 기초 프로그래밍 과목을 전체 학생 대상으로 필수 과목 지정의 실행 효과를 가늠하고자, 2022년 1학기에 해당 과목을 수강한 서강 MOT 신입생 27명을 대상으로 한 익명의 설문 결과를 살펴본다. 설문 결과 분석을 통해, 학생들이 느끼는 프로그래밍 과목 수강의 어려움을 이를 극복하기 위한 시도들의 효용성에 대해 논의하고 있다.

4.1. 학생들의 학습 배경 파악

“기술경영을 위한 프로그래밍 실습” 수강생 27명의 학부 전공을 파악한 결과는 <표 3>과 같다. 컴퓨터공학 포함 공학 계열을 전공학 학생이 전체 인원의 약 45% 정도인 12명이었으며, 그 외 인문/자연 계열 전공 학생이 15명이었다. 프로그래밍의 경험치 측면에서는 전체 인원의 55% 정도의 인원인 15명의 학생은 프로그래밍 경험이 전무하였으며, 나머지 인원 중 파이썬 이외의 언어로 프로그래밍이 가능한 2명의 학생을 제외하면, 대부분의 학생들은 해당 과목 이수 이전에는 프로그래밍에 거의 익숙하지 않았던 것으로 파악되었다. 학생들의 전공 계열과 프로그래밍 경험치가 학습의 난이도와 어떤 관련이 있는지는 이후 설문 결과와의 교차 분석 결과를 통해 설명될 예정이다.

〈표 3〉 수강생들의 학습 배경 파악을 위한 설문 결과

1. 학부과정에서의 본인의 전공은 다음 중 어디에 해당되니까?	
1) 컴퓨터공학	2 명 (7.41%)
2) 컴퓨터공학 이외의 공학계열	10 명 (37.04%)
3) 자연계열	2 명 (7.41%)
4) 인문계열	13 명 (48.15%)
5) 예체능계열	0 명 (0%)
2. 본 과목을 이수하기 이전에 프로그래밍 경험치는 어느 정도였습니까?	
1) 파이썬 외의 다른 프로그래밍 언어를 사용하여 프로그래밍 가능했었습니다.	2 명 (7.41%)
2) 파이썬 외의 다른 프로그래밍 언어를 배워본 적은 있지만, 실제 프로그래밍이 가능하진 않았습니니다.	9 명 (33.33%)
3) 파이썬 프로그래밍을 잠시 배워본 적이 있었습니다.	1 명 (3.7%)
4) 프로그래밍은 처음 접해 보았습니다.	15 명 (55.56%)

4.2. 온라인 오픈 콘텐츠 활용의 효과

스텝 2 과목인 “기술경영을 위한 프로그래밍 실습” 수강을 위한 선수과목이었던 셀프 사전학습 과목이 본 강의의 이해도를 높이는 데에 도움이 되었는지를 묻는 질문에는, <표 4>가 보이는 바와 같이, 7명을 제외한 20명의 학생들(약 74%)은 그 정도에 차이가 있던 하지만, 도움이 되었다는 답변을 하였다.

〈표 4〉 프리스터디 과정의 효용성에 관한 설문 결과

3. 본 강의 이전에 이수했던 pre-study 과정이 본 강의를 이해하는데 도움이 되었습니까?	
1) 매우 도움이 되었습니다.	6 명 (22.22%)
2) 조금 도움이 된 것 같습니다.	14 명 (51.85%)
3) 글썄, 잘 모르겠습니다.	3 명 (11.11%)
4) 별로 도움이 되지 않았던 것 같습니다.	4 명 (14.81%)
5) 전혀 도움이 되지 않았던 것 같습니다.	0 명 (0%)

4.3. 기초 프로그래밍 수업에서의 도구 활용

SW 중심대학으로 선정된 주요대학의 비전공 신입생 대상의 프로그래밍 교육과 비교하여, MOT 학생들 대상의 기초 프로그래밍 수업이 직면하는 또 다른 도전 과제 중 하나는 모든 학생이 비전공자는 아니라는 점이다. 학습 배경에 관한 설문 결과가 말해주듯이, 많은 숫자는 아니지만, 수강 학생 중에는 이미 프로그래밍이 가능하거나, 프로그래밍 교육을 받은 경험이 있는 학생들도 포함되어 있다. 파이썬이 아닌 다른 언어를 사용하여 프로그래밍을 해본 경험이 있는 학생들과 프로그래밍 경험이 전무한 학생들 간에는 이해도 측면에서 큰 차이가 발생한다. 이러한 학생들의 이전 프로그래밍 경험 차이에 따른 이해도 차이를 최소화시키기 위해, 해당 과목 진행시 다음과 같은 도구들을 활용하였다.

1. 구현 언어의 문법보다는 전반적인 프로그래밍의 구조를 파악하기 용이하도록 전체 프로그램 로직을 도식화하는 Flowchart를 활용
2. 학생들 간의 이해도 차이를 상쇄시키기 위해, 주요 코드 패턴을 암기하도록 하고, 매 수업 시작 시간에 지난 시간에 배운 코드 패턴과 관련된 퀴즈 세션을 가짐
3. 본 과목 수강 이전에 프로그래밍 경험이 있는 학생이라 하더라도, 코드 품질을 고려한 프로그래밍에는 익숙하지 않다는 점에서 과제로 제시한 문제에 학생들이 작성한 코드들에 대한 1:1 리뷰 세션을 가짐

위와 같은 교육 도구 활용에 대한 학생들의 피드백은 아래 <표 5>와 같다. 프로그램 작성시 알고리즘을 도식화하여 이해를 돕고자 했던 Flowchart는 약 63%의 인원인 17명 가량의 학생들은 도움이 되었다고 응답하였고, 주요 패턴 코드의 암기가 문제 해결에 도움이 되었다고 응답한 학생의 숫자는 70% 가량에 해당되는 19명이었다. 코드 리뷰의 효용성에 대해서는 약 89%에 달하는 학생이 도움이 되었다고 답변했으며, 매 수업 시간에 가졌던 퀴즈 세션이 강의 내용을 이해하는 데 도움이 되었다고 답한 학생도 전체 70% 정도였다. 강의를 수강하는 것 이외에 가장 도움이 되었던 것은 스스로 과제 해결을 하는 것이었다고 답하였으며, 코드 리뷰 세션과 주요 코드 패턴 암기, 그리고 퀴즈 순으로 도움이 되었다고 설문에 답하였다.

석사과정생에게는 다소 어울리지 않는 학습방법일 수 있는 주요코드 패턴 암기나 매 수업시간 시작시의 퀴즈 세션, 그리고 1:1의 코드 리뷰 등에 대한 학생들의 피드백 결과가 긍정적이었던 점은 상당히 고무적이다. 실제 강의 외에 시도했던 몇 가지의 학습을 위한 도구들을 적용하지 않았던 이전 학기의 클래스와 비교하면 해당 학기 클래스의 기말고사 성적의 평균점은 약 15% 정도가 상승되었다. 본 저자의 강의 경험에 비춰볼 때, MOT

학생들은 컴퓨터공학 전공 학생들처럼 개발자 커뮤니티에 참여한다거나, 지인과의 소통을 통해 프로그래밍 능력을 향상시키기에는 상대적으로 취약한 부분이 있다. 결과적으로 학생들은 프로그래밍 과목 수강에 있어서, 교수자의 가이드에 일방적으로 의존하는 경향을 보이기 때문에, 석사 과정 학생이라고 하더라도 교수자가 끊임없이 동기부여를 하는 노력을 추가적으로 기울일 필요가 있다는 점을 시사하는 결과로 해석 가능하다.

〈표 5〉 수업에 활용한 학습 방법의 효용성에 관한 설문 결과

4. 초반에 활용한 Flowchart는 프로그래밍을 이해하는 데 도움이 되었습니까?	
1) 도움이 되었습니다.	17 명 (62.96%)
2) 도움이 되지 않았습니다.	3 명 (11.11%)
3) 글썄, 잘 모르겠습니다.	7 명 (25.93%)
5. 주요 코드 패턴을 암기하는 것이 새로운 프로그램 과제를 해결하는 데에 도움이 되었습니까?	
1) 도움이 되었습니다.	19 명 (70.37%)
2) 도움이 되지 않았습니다.	2 명 (7.41%)
3) 글썄, 잘 모르겠습니다.	6 명 (22.22%)
6. 프로그래밍 과제에 대한 학생들의 작성 코드를 일일이 수업시간에 review하는 것이 본인의 코드 품질 향상에 도움이 되었습니까?	
1) 도움이 되었습니다.	24 명 (88.89%)
2) 도움이 되지 않았습니다.	2 명 (7.41%)
3) 글썄, 잘 모르겠습니다.	1 명 (3.7%)
7. 거의 매수업 시작하기 전에 본 간단한 퀴즈들은 각 chapter의 내용을 이해 하는데 도움이 되었습니까?	
1) 도움이 되었습니다.	19 명 (70.37%)
2) 도움이 되지 않았습니다.	3 명 (11.11%)
3) 글썄 잘 모르겠습니다.	5 명 (18.52%)
8. 강의 이외에 다음 activity 중 가장 도움이 되었다고 생각되는 것은 무엇인가요?	
1) 과제(Exercise/Lab) 해결	13 명 (48.15%)
2) 퀴즈를 보기 위한 공부	3 명 (11.11%)
3) 주요 코드 암기	5 명 (18.52%)
4) 각 과제에 대한 code review session	6 명 (22.22%)
5) 기타	0 명 (0%)

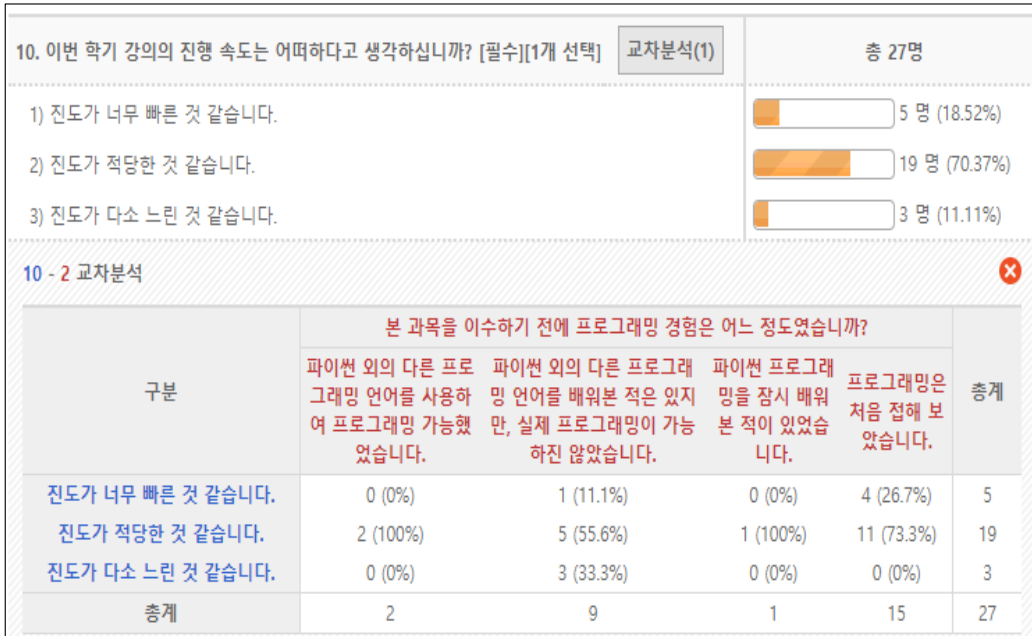
4.4. 개인별 학습 배경 차이에 의한 과제 해결 난이도 및 강의 이해도 격차 유무

앞서 언급한 바와 같이, MOT 학생들의 기초 프로그래밍이나 IT 기술에 대한 경험 정도가 모두가 전무한 것이 아니라, 개인적인 편차가 존재한다는 점이 MOT에서의 IT 관련 과목 커리큘럼 설계시 주요 고려사항 중 하나였다. <그림 4>는 학생 개인이 느끼는 과제 해결의 난이도와, 앞서 답했던 학생 개개인의 프로그래밍 경험치 간의 교차 분석 결과를 보여준다. 과제 난이도가 ‘너무 어렵다’ 라고 답한 학생은 전체의 약 44%, 그리고 ‘할만 합니다’ 라고 답한 학생은 나머지 56% 였다.

앞선 설문 항목에서 프로그래밍을 처음 접해 보았다고 답한 학생은 전체의 55% 였다. 그렇다면, 과제 난이도가 ‘너무 어렵다’고 답한 44%의 학생들은 모두 프로그래밍을 처음 접해 본 55%의 학생이었을까? 이 질문의 답을 <그림 4>의 하단에 정리된 교차분석 결과에서 확인할 수 있다. 결과는 예상 외로, 과제 난이도가 ‘매우 어렵다’ 라고 학생들이 느끼는 것과 해당 과목을 이수하기 이전의 프로그래밍 경험치와는 상관관계가 없었다. 프로그래밍 경험이 전무한 학생의 47%는 과제가 너무 어렵다고 느꼈지만, 프로그래밍 언어를 배운 경험이 있지만, 스스로 프로그래밍은 불가능한 상태인 학생의 56%도 과제가 너무 어렵다고 느끼고 있었다. 반면, 다른 프로그래밍 언어를 사용하여 프로그래밍이 가능한 2명의 학생은 모두 난이도 측면에서 할만하다는 답을 하고 있다.



<그림 4> 과제 난이도에 대한 설문 결과와 학생 개인의 학습 배경과의 교차분석 결과



<그림 5> 학습 이해도에 대한 설문 결과와 학생 개인의 학습 배경과의 교차분석 결과

학생 개인의 프로그래밍 경험치와 학습 난이도간의 관계를 살펴보기 위해, 강의 진행 속도를 어떻게 느꼈는지를 질문하였다. 설문 결과 약 70%의 학생들은 강의 진도는 적당하다고 답하였으며, 19% 가량의 학생들은 빠르다고 답하였다. 11%의 학생들은 강의 진도가 느리다고도 답하였다. 이런 설문결과를 앞의 질문과 마찬가지로, 개인의 프로그래밍 경험치 정도와 교차분석을 한 결과가 <그림 5>의 하단에 정리되어 있다. 프로그래밍을 전혀 경험하지 않은 학생의 약 27% 가량이 강의 진도가 빠르다, 다시 말해 강의 진도를 따라가기에 어려움이 있다는 답을 한 반면, 프로그래밍 수업을 들은 적이 있는 학생들의 경우엔 약 11% 가량의 학생들만이 강의 진도가 빠르다는 답을 하였다.

기초 프로그래밍 과목 수강 이전의 학생 개인의 프로그래밍 경험치와 과제 해결 난이도, 그리고 학습 난이도 간의 관계를 살펴 본 설문 결과에서 유추할 수 있는 사실은 다음과 같다.

이미 다른 구현 언어로 프로그래밍이 가능한 학생들이 새로운 언어로 프로그래밍 수업을 수강하는 데에는 과제 해결 난이도와 학습 난이도 측면에서의 어려움은 없었다.

다른 구현언어로 프로그래밍 수업을 수강한 경험은 있으나, 스스로 프로그래밍하는 것이 가능하지 않은 학생들은 새로운 언어의 문법 설명이 추가 되는 강의를 수강하는 데

있어서는 어려움이 없었지만, 과제 해결 능력에 있어서는 프로그래밍 경험이 전혀 없는 학생들과 마찬가지로 어려움을 겪고 있다. 즉, 구현언어의 구조나 문법을 조금 더 빨리 이해한다고 해서, 프로그래밍을 통한 문제 해결 능력이 향상되는 것을 의미하지는 않는다는 것을 시사한다.

4.5. 강의 진행 형태

마지막으로 수업 형태 선호도에 대한 설문 결과는 <표 6>과 같다. 실제 설문 대상이 된 과목은 온라인 실시간 수업 형태로 진행되었다. 이에 대한 학생들의 선호도를 묻는 질문에서, 4명의 학생을 제외하고는 온라인 수업 진행에 대한 긍정적인 답변을 하였다. 그러나, 향후 오프라인 강의 전환 시에, 동영상 강의를 활용한 Flipped Learning 방식의 수업 형태에 대해서는 찬반이 엇갈리는 답변을 하였다. 팬데믹을 거치면서 화상 회의 도구들을 활용한 실시간 양방향 소통에 익숙해진 학생들은 강의의 매체가 화상회의 도구가 되는 것에는 거부감이 없으나, 미리 녹화된 동영상 강의를 혼자서 수강하는 것에는 의견이 양분되는 반응을 보이고 있었다.

<표 6> 강의 형태에 대한 선호도 설문 결과

11. 온라인으로 수업을 진행하는 것에 대해 어떻게 생각하시나요?	
1) 오프라인 수업과 별 차이가 없습니다.	6 명 (22.22%)
2) 오프라인 수업보다 이해하기가 어렵습니다.	4 명 (14.81%)
3) 오프라인 수업보다 시간적인 면에서 더 효율적입니다.	17 명 (62.96%)
12. 향후 오프라인 강의로 전환되더라도, 일반적인 강의는 동영상 강의로 수업시간 이전에 수강한 후, 본 수업시간에는 질의 응답 및 exercise 문제 해결을 하는 방식의 강의진행(Flipped Learning)에 대해 어떻게 생각하시나요?	
1) Flipped Learning 방식이 더 좋을 것 같습니다.	12 명 (44.44%)
2) 그냥 본 강의 시간에 강의를 진행하는 것이 좋은 것 같습니다.	11 명 (40.74%)
3) 글썸, 잘 모르겠습니다.	4 명 (14.81%)

4.6 기초 프로그래밍 과목의 필수과목 선정에 대한 피드백

학생들에게 기초 프로그래밍 과목 수강을 전원 필수과목으로 선정한 결정에 대한 의견을 자유롭게 답하도록 설문을 진행하였다. 다음은 다수의 학생들의 의견을 정리한 결과이다.

- 막연히 필요하다고는 생각하였으나, 혼자 시작하기엔 어려웠던 프로그래밍을 필수 과목으로 수강한 것은 결과적으로는 좋은 기회였다.
- 수강하는 데 있어서 요구하는 학습량은 많은 과목이다.
- 퀴즈, 과제, 코드 패턴 암기 등이 어렵기는 하지만, 도움이 되었다.
- 스스로 프로그래밍 과제를 해결했을 때는 나름의 희열이 있다.

대다수의 학생들은 위와 같이 긍정적인 답변을 하였으나, 1~2명의 학생들은 IT 기술 습득을 위해 MOT 진학을 한 것이 아니라는 점에서, 다른 학생들에게 추천하고 싶지는 않다는 답변을 하기도 하였다. 소수의 의견이지만, 향후 커리큘럼 운영에 있어서 감안해야 할 의견으로 볼 수 있다. MOT 학생들 중 소수는 굳이 디지털 트랜스포메이션에 적응하여 새로운 솔루션을 창출해야 할 니즈를 가지고 있지 않은 경우들이 있다. 이런 학생들에게는 필수과목 이수를 우회할 수 있는 패스를 열어주는 것도 고려할 필요가 있다고 판단된다.

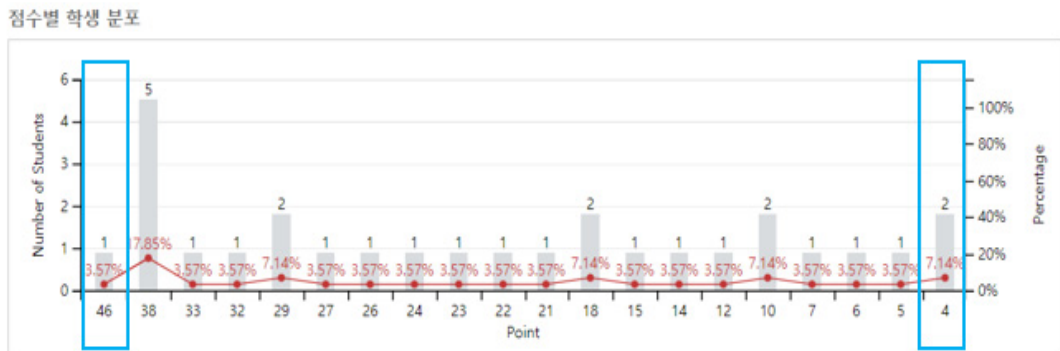
IV. 결론 및 향후 계획

변화하는 기업환경과 디지털 트랜스포메이션이라는 새로운 패러다임으로의 흐름에 따른 기술경영학의 커리리지 변화로부터 기인한 교육 커리큘럼의 변화, 그 중에서도 특히 IT 역량 교육을 위한 커리큘럼의 변경의 필요성을 논의하고, 하나의 선행 사례로서, 서강 MOT의 IT 역량 교육을 위한 커리큘럼 리노베이션의 시작점과 디자인 과정, 그리고 리노베이션된 커리큘럼에 포함된 일부 과목에 대한 학생들로부터의 피드백 등을 소개하였다. 서강 MOT의 커리큘럼 리노베이션 과정을 통해 찾은, 서두에 제시한, 두 가지 질문에 답은 아래와 같이 요약 가능하다.

1. 융합학문 분야인 기술경영학의 범주 내에서, 프로그래밍 교육은 디지털 트랜스포메이션을 주도하는 인재 양성을 위한 필요조건인가?
 - 리노베이션된 커리큘럼을 실행한 기간이 2학기에 불과한 현재 시점에서, 본 질문에 대한 단정적인 답을 제시하는 것은 시기상조로 판단된다. 그러나, 향후 실질적인 문제 해결 결과를 보여야 하는 프로젝트 수업에서 학생들이 작성하는 산출물들의 품질 향상 정도와 전체 커리큘럼을 이수한 학생들로부터의 피드백 결과 분석 등을 통해, 프로그래밍 교유의 효과에 대한 정성/정량적인 분석이 가능할 것으로 기대한다.
2. 프로그래밍으로부터 출발하는 디지털 트랜스포메이션 트랙의 성공을 위해 기술경영전문대학원의 교육 기간과 자원의 제한을 극복하기 위해 필요한 전략은 무엇인가?
 - 적극적인 오픈소스의 활용과 솔루션 기업과의 전략적인 협업을 통해, 내부 인력 자원의 부족과 MOT 과정의 물리적인 IT 교육 시간의 부족을 어느 정도 극복할 수 있다. 오픈 소스의 활용 방식은 두 가지 측면으로 나뉠 수 있다. 학생들의 실습 진행에 있어서 오픈소스를 활용함으로써, 프로그래밍에 대한 난이도 부담을 덜 수 있다. 또 한 가지 오픈소스 활용은 서강 MOT의 사례에서 보이듯이, 기존 인터넷에 오픈되어 있는 무료강좌를 사전학습에 활용하는 것이다. 프로그래밍은 단방향 티칭만으로 습득할 수 없는 만큼, 양질의 오픈 강좌들을 발굴하여 활용함으로써 MOT 내부 교수진은 좀 더 융합교과목을 개발하고 교육하는 데에 집중할 수 있다. 현재 프로그래밍 관련 과목을 교수진이 담당하고 있으나, 어느 정도 커리큘럼이 안정화되는 기간을 거친 후에는 기초 프로그래밍 과목은 전면적으로 오픈 강좌로 대신하고, 내부 교수진들은 좀더 IT 기술과 경영을 융합한 과목의

개발과 교육에 집중할 계획을 서강 MOT에서도 수립 중이다. 오픈 강좌로 해결하기 힘든 주제에 대해서는 외부 솔루션 기업 혹은 교육업체와의 협업을 통한 교육 품질의 향상도 대안이 될 수 있다.

기술경영 전공 학생들 대상의 IT 역량 교육에 있어서의 난제들은 여전히 존재한다. 본 저자의 교육 경험에 비추어, 향후 해결해 나가야 할 난제 중 하나는 IT 역량 관련 과목에서의 학업성취도의 표준편차가 여전히 매우 크다는 점이다. <그림 6>은 설문 대상이었던 “기술경영을 위한 프로그래밍 실습” 과목의 기말고사 점수별 학생 분포를 보여준다. 여러 가지 학습 도구의 활용으로 평균점의 향상은 있었으나, 46점 만점 시험에서 4점에서 46점에 이르기까지, 거의 동일한 점수에 각각 1명씩의 학생이 배치된 듯한 분포 그래프는 상당히 여러 학기에 걸쳐서 반복적으로 나타나는 기초 프로그래밍 과목 점수분포표의 전형이라고 볼 수 있다. 대부분의 클래스에서 최고점을 받는 학생과 최저점을 받는 학생은 모두 프로그래밍 경험이 전무한 학생들이라는 점도 특징적이다.



<그림 6> “기술경영을 위한 프로그래밍 실습” 과목 기말고사 점수별 학생 분포

이러한 현상은 본 저자의 교육 경험에 비추어 봤을 때, MOT 학생이 아니더라도 비전공자 학생을 대상으로 하는 대부분의 IT 역량 교육 과목에서 반복적으로 발견된다. 앞으로의 향후 지속적인 분석과 연구를 통해, MOT 전공 학생들 혹은 비전공자 학생들 대상의 IT 역량 관련 과목의 학업성취도 표준편차를 줄일 수 있는 방안을 강구해 나갈 계획이다.

참고문헌

(1) 국내문헌

- 이영석 (2018), “컴퓨터 비전공자를 위한파이썬 기반 소프트웨어 교육 모델”, 한국융합학회논문지, 제9권 제3호, pp. 73-78.
- 오미자 (2017), “스크래치 프로그램을 활용한 프로그래밍 교육에 대한 비전공자의 인식 연구”, 컴퓨터교육학회 논문지, 제20권 제1호, pp. 1-11.
- 한옥영·김재현 (2017), “비전공자 소프트웨어 교육을 통한 컴퓨팅 사고력 향상에 대한 연구”, 한국컴퓨터교육학회 학술발표대회논문집, 제21권 제1호, pp. 139-141.
- 서주영 (2017), “SW융합인재 양성을 위한 비전공자 프로그래밍 학습에 관한 사례 연구”, 디지털 융복합연구, 제15권 제7호, pp. 123-132.
- 김수환 (2015), “Computational Thinking 교육에서 나타난 컴퓨터 비전공 학습자들의 어려움 분석”, 컴퓨터교육학회 논문지, 제18권 제3호, pp. 49-57.
- 여영준 (2020), “포스트 코로나 시대 기술변화와 혁신정책 방향성 재정립: 창조적 학습사회 전환을 중심으로”, 기술혁신연구, 제28권 제4호, pp. 153-165.
- 소프트웨어 중심대학 협의회(연도미상), SW중심대학, <https://www.swuniv.kr/>.

(2) 국외문헌

- Chanaron, J. and Jolly, D. (1999), “Technological management: expanding the perspective of management of technology”, *Management Decision*, Vol. 37, No. 8, pp. 613 - 621.
- Badawy, M. K. (1998), “Technology Management Education: Alternative Models”, *California Management Review*, Vol. 40, No. 4, pp. 94 - 116.
- Guo, P. (2014), “Python is Now the Most Popular Introductory Teaching Language at Top U. S. Universities”, *BLOG@CACM*, <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>.
- Ling, H. C., Hsiao, K. L. and Hsu, W. C. (2021), “Can Students’ Computer Programming Learning Motivation and Effectiveness Be Enhanced by Learning Python Language? A Multi-Group Analysis”, *Frontiers in Psychology*, Vol. 11, pp. 1-7.
- Lasser, J., Manik, D., Silbersdorff, A., Säfken, B. and Kneib, T. (2021), “Introductory data science across disciplines, using Python, case studies, and industry consulting projects”, *Teaching Statistics*, Vol. 43, S190-S200.
- Nambisan, S. and Wilemon, D. (2003), “A global study of graduate management of technology programs”, *Technovation*, Vol. 23, No. 12, pp. 949 - 962.

□ 투고일: 2022.12.02. / 수정일: 2023.02.12. / 게재확정일: 2023.05.09.