

미래 공학자 만들기, 소프트웨어 초등교육으로부터



최소영

서울대곡초등학교 교사
nicein@snu.ac.kr

서울대학교 교육공학 박사수로
서울교육대학교 생활과학교육 석사
관심분야: 창의·융합 교육, 교수설계

소프트웨어 교육의 시작

4차 산업혁명은 고성능 프로세서, 초고속 네트워크, 소프트웨어 기술의 융·복합이 강조되는 소프트웨어 중심 사회로의 전환을 의미한다. 알파고의 등장은 우리 사회 전반에 많은 변화가 일어날 것이라는 미래에 대한 예측과 더불어 소프트웨어 기술에 대한 관심을 불러일으키고 그 중요성을 실감하게 만들었다. 예컨대, 우리가 잘 알고 있는 무인자동차 기술의 선두주자는 완성차 업체가 아닌 정보통신기술 기업인 구글(google)인 것과 같이 소프트웨어는 융·복합 산업의 핵심으로 소프트웨어 기술이 국가 경쟁력의 중심으로 자리 잡고 있다.

소프트웨어 기술이 갖는 의미와 중요성은 초·중등 교육과정에도 반영되고 있다. 영국에서는 2014년부터 초·중등 교육과정에 ‘컴퓨팅’ 교육을 필수화하였고, 미국은

K-12 컴퓨터과학 표준에 ‘컴퓨팅 사고력(Computational Thinking)’을 명시하였다. 프랑스는 소프트웨어 교육을 중학교 정규과목으로 개편하였으며, 에스토니아는 2015년부터, 핀란드는 2016년부터 모든 초등학생을 대상으로 소프트웨어 교육을 실시하고 있다.

우리 정부는 소프트웨어 중심 사회로의 패러다임 변화에 대응하고자 2014년 「소프트웨어 중심사회 실현 전략」을 수립하여 발표하고, 교육부에서는 「초·중등 소프트웨어 교육 활성화 방안」과 「2015년 문·이과 통합형 교육과정」을 통해 초·중·고등학교에서의 체계적인 소프트웨어 교육의 도입을 위한 방안을 마련하였다. 학교급별 소프트웨어 교육의 내용을 살펴보면, 초등학교에서는 실과 교과의 17시간을 소프트웨어 기초 소양을 중심으로 문제해결 과정, 알고리즘 및 프로그래밍 체험, 저작권 보호 등 정

표 1. 2015년 교육과정 개정에 따른 소프트웨어 교육 주요 개편 사항

구분	과목명	주요 내용
초등학교	실과(17시간)	- 문제해결과정, 알고리즘, 프로그래밍 체험 - 저작권 보호 등 정보윤리의식 함양
중학교	정보(34시간, 필수)	- 컴퓨팅 사고력 기반 문제해결 교육 - 알고리즘, 프로그래밍 개발
고등학교	정보(일반선택 과목)	- 다양한 분야와 융합한 알고리즘 설계, 프로그램 개발 - 기초과목, 실무과목으로 개편하여 국가직무능력표준(NCS) 기반 교육과정 구성

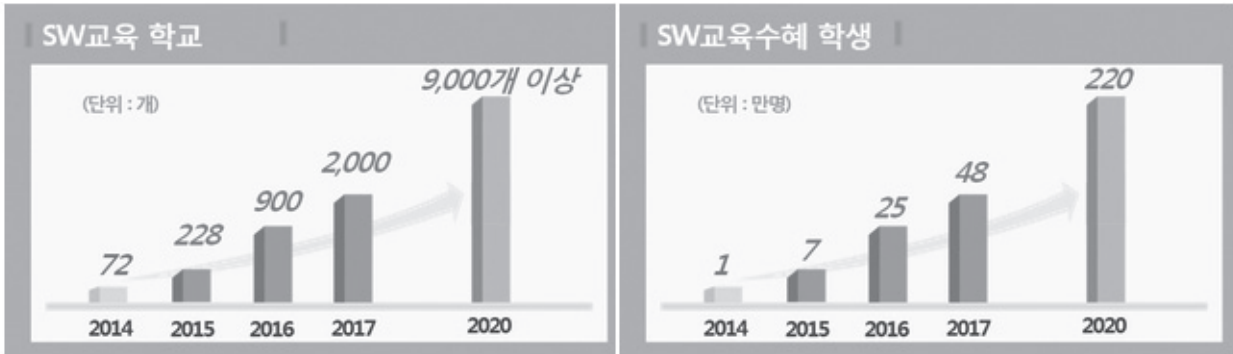


그림 1. 교육부와 미래창조과학부의 소프트웨어교육 운영 학교 및 수혜학생 전망

보 윤리 의식 함양에 대해 가르치게 된다. 중학교에서는 전문적인 소프트웨어 교육을 위해 ‘정보’과목을 의무화하여 컴퓨팅 사고 기반 문제해결 교육과 알고리즘 및 프로그래밍 개발 등을 주요 내용으로 총 34시간을 교육하도록 하였다. 고등학교에서는 심화선택이던 ‘정보’과목을 일반 선택으로 전환하고 다양한 분야와 융합한 알고리즘의 설계 및 프로그램 개발 중심의 교육을 실시하도록 하였다(〈표 1〉 참고).

2015년 개정 교육과정이 적용되는 2018년이 다가오에 따라, 미래창조과학부에서는 작년 900여 학교에서 300개 학교를 추가하여 올해 1200개의 소프트웨어 교육 선도학교를 선정하기로 하였다. 또한 미래창조과학부와 정보통신기술진흥센터는 2015년 8개 소프트웨어 중점대학 선정을 시작으로 올해에는 20개 대학으로 확대 운영할 계획이다(〈그림 1〉 참고).

소프트웨어 교육의 본질

소프트웨어 교육은 무엇이며, 왜 중요한 것일까?

소프트웨어 교육의 목표는 학생들의 컴퓨팅 사고력 증진에 있다. 학자마다 컴퓨팅 사고력에 대한 정의가 다소 상이하지만, 미국의 컴퓨터과학을 가르치는 교사 모임(Computer Science Teachers Association, CSTA)에서는 컴퓨터과학 표준 전체에 스며들어 있는 추상화, 자동화, 분석에 중점을 둔 컴퓨터과학의 핵심 원리이며, 컴

퓨터과학과 모든 학문 분야를 연관하여 일상에서 발생할 수 있는 문제들을 컴퓨팅의 기본적 개념과 원리에 근거하여 분석하고 해법을 발전시키는 문제해결방법론의 하나로 정의하고 있다. 컴퓨팅 사고력은 학문 분야뿐만 아니라 일상의 문제를 해결하기 위해 컴퓨팅 시스템을 활용할 수 있는 능력과 문제해결을 위해 효율적, 절차적으로 사고할 수 있는 능력으로 소프트웨어 중심 사회에서 읽기, 쓰기, 셈하기의 3R과 더불어 학생들이 갖추어야 할 기본소양으로 이해되고 있다.

소프트웨어 교육을 통한 컴퓨팅 사고력 함양은 궁극적으로 창의·융합 인재 양성의 목적을 가진다. 컴퓨팅 사고는 복잡한 문제(ill-structured problem)의 증가와 소프트웨어가 일상의 삶에서 분리될 수 없는 사회에서 개인과 사회, 국가의 문제를 해결하는데 많은 영향력을 미친다. 이미 여러 분야에서 컴퓨팅 사고는 문제해결을 위해 새로이 고안한 기법을 가능하게 하며 혁신적인 결과를 이끌어내고 있다.

컴퓨팅 사고를 바탕으로 문제를 효율적, 효과적으로 해결할 수 있는 방법을 찾고 다양한 아이디어를 구안하고 적용하여 현실화하는 문제해결의 과정은 창의적 문제해결력을 향상시킬 수 있는 기회를 제공한다. 소프트웨어 교육을 통해 양성하고자 하는 창의·융합 인재는 컴퓨팅 사고를 기반으로 전문 지식, 타영역의 지식과 자원을 융합하여 독창적이며 유용한 산출물을 창출할 수 있는

사람이라고 할 수 있을 것이다.

초·중등 소프트웨어 교육과 공학교육

최근 미래교육의 방향을 제시하는 여러 연구들에서는 ‘교육의 연계성’을 강조하고 있다. 연구자들마다 연계의 대상과 의미는 다소 다르지만 서로 다른 요소들의 연계를 미래교육의 방향으로 제시하고 있다.

공학교육은 체계성을 강조하는 학문분야이다. 예컨대, 교육과정 측면에서 보면 공학교육은 전문교양, MSC, 전공기초 및 심화로 이어지는 일련의 연계성과 체계성을 바탕으로 전공 학문의 발전과 전문 인재의 양성을 목표로 하고 있다. 이러한 공학교육의 학문적 목표에 근거하여 초·중등 소프트웨어 교육의 의미를 두 가지 관점에서 논의하고자 한다.

첫째, 소프트웨어 교육은 초·중등 교육과 대학교육과의 연계가 가능하다. 초·중·고등학교를 대상으로 한 2015년 개정 교육과정은 초등학교부터 소프트웨어 교육을 필수화함에 따라 초등학교부터 대학교까지 소프트웨어 교육의 연계성을 갖게 되었다(그림 2참고). 대학에서 소프트웨어에 대한 기초지식과 역량을 갖춘 학생이 신입생으로 입학한다는 것은 교육의 효율성과 효과성에서 큰 강점을 가질 것이다. 대학은 소프트웨어의 역량을 갖춘 학생들에게 심화된 학습내용을 제공할 수 있으며 그만큼 대학 자체의 수월성을 높일 수 있는 기회를 확보할 수 있을 것이다. 또한 교육과정에 여유가 생겨 창업교육을 위한 시간을 확보할 수도 있다.

둘째, 창의·융합교육이 실현된다. 자동차 부품의 절반 가량이 전자장치로 이루어지는 시대에서 소프트웨어 교

육은 창의·융합교육의 DNA를 지니고 있다. 최근 공과대학에서는 소프트웨어학과를 신설하거나 개편하고 창의·융합교육의 실현을 위한 교육과정을 개발하고 있다. 공과대학에서의 소프트웨어 교육은 제반 산업의 맥락을 중심으로 기계, 전자, 디스플레이, 자동차, 의료 등 다양한 분야와의 실제적인 융합을 통해 창의·융합교육이 실현되도록 노력하고 있다. 대학에서 처음 융합교육을 실현하는데 가장 큰 걸림돌은 전공에 대한 이해도가 낮은 상황에서 다른 학문과 융합을 시도하기 어렵다는 것인데 초·중등 소프트웨어 교육을 통해 알고리즘을 이해하고 설계할 수 있는 학생들의 입학은 이러한 문제점을 해결하는데 도움이 될 것으로 보인다.

이와 같이, 소프트웨어 교육이 초중등에서 시작됨에 따라 공학교육의 역할과 기능은 더욱 확대될 것이다. 소프트웨어 교육이 초등교육을 시작으로 대학교육으로 이어지는 기반을 갖추었다는 것은 4차 산업 혁명을 대비한 창의·융합적 인재양성에 중추적인 역할과 기능을 수행할 것을 의미한다. 이처럼 소프트웨어 교육은 현재가 아닌 미래교육의 관점에서 볼 때, 교육의 연계성과 창의·융합교육에 시사점을 제공하고 있다.

소프트웨어 교육, 제2의 르네상스 시대를 열수 있을 것인가?

스티브 잡스의 인문학 열풍이후 4차 산업혁명 시대의 도래로 다시 창의·융합적 공학교육이 주목받고 있다. 이전에 공학교육과는 또 다른 공학교육의 사회적 요구가 밀려오고 있다. 그 중심에 소프트웨어 교육이 있으며, 소프트웨어 교육이 기대한 결과를 얻기 위해서는 몇 가지 전



그림 2. 교육부에서 발표한 초등학교에서 대학까지 소프트웨어 교육의 단계별 목표

제조조건이 필요하다.

첫째, 교육적 소프트인프라를 먼저 구축해야 한다. 교육은 단편적·단기간적인 운영으로는 성과를 보기 어렵다. 소프트웨어 중심의 창의·융합적 인재양성을 위해서는 교육과정의 개편과 더불어 유연한 학사제도를 운영할 수 있는 다양한 대학지원 제도 및 정책이 뒷받침되어야 한다. 이는 단순히 제정 중심의 지원이 아니라 대학 스스로 소프트웨어 교육을 위한 제도와 정책을 마련할 수 있는 기회를 제공하는 소프트인프라 구축을 의미한다.

둘째, 초·중·고 교육과정에서 대학 교육과정으로 이어지는 연계형 교육과정이 개발되어야 한다. 초·중등 교육에서 소프트웨어 교육을 정규화 함으로써 공과대학은 소프트웨어 전공지식의 깊이와 넓이를 모두 갖출 수 있는 시간적 여유를 마련하게 되었다. 이러한 이점을 활용하기 위해서 대학의 공학교육은 초·중등 교육과정에서 배운 소프트웨어 교육의 성과를 이어나갈 수 있는 교육과정을 개발하여야 한다. 이를 통해 공과 대학의 교육과정을 수월성 전략으로 특화시킬 수 있다.

셋째, 연계형 비교과 프로그램을 활성화해야 한다. 최근 소프트웨어 관련 경진대회나 소프트웨어 대학생 멘토링 프로그램, 초중등-대학교 SW Capstone Design Camp 등 연계형 프로그램이 활성화되고 있다. 이러한 다양한 비교과 프로그램은 공학교육의 다양화뿐만 아니라 소프트웨어 교육의 완성도를 높여 전문적인 소프트웨어 인재를 양성하는데 기여할 수 있을 것이다.

공학교육은 국가경쟁력을 높일 수 있는 전략이자 수단이다. 특히 창의·융합적 공학인재의 양성은 국가경쟁력 향상에 핵심적 요소이다. 초·중등 소프트웨어 교육에서 대학의 소프트웨어 중점대학으로 이어지는 일련의 소프트웨어 교육체계는 미래 지능정보사회를 준비하는 국가 교육정책의 방향을 의미한다.

클라우드 슈밥(Klaus Schwab)회장이 4차 산업혁명을 ‘모든 것이 연결되고, 더욱 지능적인 사회로의 진화’로 설명한 것처럼, 소프트웨어 교육은 초등교육에서 대학교육까지 교육의 연계성을 강조하고, 더욱 더 진화된 창의·융합적 공학인재 양성에 이바지하여야 할 것이다. 