

‘국내 컴퓨터과학/공학의 위상제고’ 좌담회

서울대학교 | 이광근*

□ 사 회 : 이광근(서울대 교수)

□ 참 석(가나다순) :

김재우(한국마이크로소프트 플랫폼전략 자문)
민경오(LG 소프트웨어 센터 센터장)
박성우(포항공대 교수)
서진욱(서울대 교수)
이성환(고려대 교수)
이욱세(한양대 교수)
황규영(KAIST 교수)

□ 일시 및 장소 : 2009. 1. 20. 18:00-21:00.

서울대학교 호암교수회관

- ▶ **이광근**: 오늘 모여 주셔서 대단히 감사하다. 정보과학회지 2월호 특집이 국내 컴퓨터과학/공학 분야의 위상제고이다. 이에 대한 좌담회를 위해서 여러 분들을 모시게 되었다. 국내에서 컴퓨터 분야 인기가 예전 같지는 않다. 유럽에서는 항상 꾸준했던 것 같고, 미국에서는 컴퓨터 분야의 인기가 바닥을 치고 올라오고 있는 것 같다. 유독 국내 대학의 경우만 컴퓨터 분야의 인기가 생명력이 약한 듯하다. 국내 대학에서 어떻게 무엇을 해 나가야 이 문제를 해결 할 수 있을지 자유롭게 말씀 부탁한다.
- ▶ **민경오**: 우리 분야에서 성공한 케이스를 많이 발굴하고 소개해야 할 것 같다. 그런 분들이 찾아보면 꽤 있다. 교수님, 사업가 등 잘되신 분들을 찾아서 좀 밝은 면을 보여줘야 한다.
- ▶ **김재우**: 글로벌 기업에 진출하신 엔지니어들이 현재 어떤 경로로 일하시고 계신지, 어떻게 진출을 하셨는지 이야기를 들려주면 학생들한테는 좋은 인식을 심어줄 것 같다.
- ▶ **민경오**: 그리고 CS 분야의 성공한 역할 모델들을 찾아서 기사화하는 등 많이 알려야 한다. 너무 부정적인 것이 부각되고 있는 게 사실이다. 실리콘 벨리

에도 한국분 중에서 역할 모델이 될 만한 분들이 찾아보면 많이 있다.

- ▶ **황규영**: 여러 가지가 연결된 고리이다. 하나가 안 되니까 다른 것도 안 되고. 체인을 어디서 끊어야 하는가? 나는 대학 저년차(1,2학년) 프로그램을 교육을 아주 잘해야 한다고 본다. 그것의 사전 작업으로 초중고 컴퓨터 교육도 잘 시켜야 한다. 기업체에서는 컴퓨터 전공한 사람을 찾고 있는데 사람이 별로 없다고 하고, 학교는 학생을 많이 길러내지만, 막상 산업이 부족해서 보낼 데가 없다고 한다. 산업체에서 보면 실력이 없다고 한다. 잘하는 사람이 많아야 우리 분야가 성장하는데 잘하는 사람이 적으니 우리 분야가 성장이 안 된다. “프로그램 잘한다”는 것은 단순한 코딩 스킬이 아니다. 그리고, 롤 모델을 보면 모두 외국에 있다. 롤 모델도 별로 없고 인터스트리도 취약하므로 학생이 잘 안 온다. 학생들이 이야기 들어보면, CS가 어렵다, 커리어 패스가 안 보인다, 등의 이유로 전산 쪽에 안 온다. 인터스트리가 왜 취약한가? 정부에서 아무리 기업에 투자해야 효과는 크지 않을 것이다. 버튼 업으로 올라가야 하는데, 톱 다운으로 내려오므로 프로젝트를 쥐도 못해내면 아무 소용이 없다. 밑에서 할 수 있는 것을 정부에서 지원해줘야 한다고 본다. 정부가 저학년 프로그램과 초중고 프로그램에 투자해야 한다고 생각한다. 교사 양성 잘하도록 하고, 실력이 좋은 사람을 기르는 데 투자해야한다. 학교에선 취직을 못 시키고, 회사에서는 또 뽑을 만한 사람이 없다고 한다. 그러므로 교육의 갭을 메꾸지 못하면 이 문제 안 풀린다.
- ▶ **이성환**: 우리가 지나치게 인력배출을 많이 한다. 수요에 비해 공급이 많다. 많다보니 양질이 만들어지지 않는다. 정규 컴퓨터 과정을 전공한 학생이나 그렇지 않은 학생이나 기업에서 일 시켜보면 별 차가 없다는 거다.

* 종신회원, E-mail : kwang@cse.snu.ac.kr

- ▶ **황규영:** 결국은 학교에서 가르쳐 내보낸 스킬이 단순 스킬이라는 거다. 단순 프로그래밍 스킬로는 부가가치가 안 생긴다. 미국에서 프로그램 잘하는 사람들은, 다른 분야 사람들이 아무리 흉내 내려 해도 안 되지 않나? 학교에서 그 정도로 가르쳐야 한다는 것이다. 지금 대학교 과정에서 프로그램 못하는 학생들은 졸업할 때까지 못한다. 그런 학생들은 산업체 나가서도 잘 못한다. 대학에서 저학년 프로그램만 잘 해결해놔도 상당히 많은 부분이 풀릴 것 같다.
- ▶ **이성환:** 고등학교 교육과정의 만약 컴퓨터관련 과목의 시간 수가 늘어난다고 하면 좋은데 현실은 쉽지 않다. 수학이나 과학은 1시간씩 늘고 컴퓨터는 아직도 선택으로 남아 있다. 정보올림피아드 나간 소수의 아이들은 정말 잘하는 아이이지만, 나머지는 잘 모른다.
- ▶ **이광근:** 이런 면도 있는 것 같다. 들어온 학생들 잘 좀 가르쳐서 좋은 인재로 교육시키고, 새로운 산업 만들도록 준비시키고 하는 것도 중요하지만, 애초부터 고등학생들에게 CS 이미지가 덜 매력적으로 보여진 것 같다. 그것이 두렵다. 전교 탑 수준의 아이들 중에서도 한 둘은 그래도 컴퓨터 사이언스를 해야겠다, 라는 아이들이 나와야 하는데 CS는 한때는 그랬다가 덜해진 쪽으로 되는 것 같다.
- ▶ **황규영:** CS의 이미지가 “코딩”하는 것으로 보는 게 문제다. “코딩”과 “프로그래밍”은 다르다. 모르는 학생들은 전산과는 코딩만 죽어라 하는 줄 안다. 회사에서도 잘못 이해한다. 코딩과 프로그램을 구분 못한다. 프로그램은 다 잘하는데 설계는 잘 못한다며 외국에서 들여온다는 말을 들었다. 바르게 이야기하면 코딩은 잘하는 데 프로그래밍을 잘 못한다는 평이다. “프로그램은 잘하는 것”은 설계까지 잘하는 모든 과정을 뜻한다. 이 말은 즉, 학교에서 코딩만 가르친다는 것 아닌가. 그러니까 학생들이 재미가 없어한다. 프로그래밍은 큰 시스템을 디자인하고 설계하는 거고, 기술적인 아키텍처 기술을 활용하여 작품을 만드는 것이다. 학교에서 설계 능력을 가르쳐야 한다. 프로그래밍을 아티스틱하게 하고 즐기게 만들고 자기가 원하는 기능을 가지는 소프트웨어를 만들 큰 설계를 갖출 능력을 가르쳐주면 재미를 느낀다. 회사에서도 이것을 필요로 한다.
- ▶ **이성환:** 교육과정 개설에 문제와 연결되는데 소프트웨어 엔지니어링을 강조해서 교육해야 한다. 그런데

현실은 SE 한 과목과 프로그래밍 한 과목 정도다. 현실은 막상 교과과정 변경할 때 SE 분야의 교과목 4~5목 과목을 강조하자고 하면, 난리가 난다. 다른 분야에서 난리가 난다.

- ▶ **황규영:** “프로그래밍” 교육이 SE 교과과정에 국한되는 것이 아니라 CS의 기초 분야를 더 뜻한다. 설계, 구성, 논리 등등.
- ▶ **이광근:** 지금까지 논의가 졸업후 좋은 산업을 만들어서 훌륭한 산학 사이클을 만들어 내기 위해서 학교에서 할 것에 대해 이야기했다. 그리고 또 학교에서 할 것이, 후속 학문 세대를 제대로 키우는 거다. 학교에서 좋은 사람을 키워야 하는데 고등학교 때부터 좋은 학생들은 다른 분야로 가고, 컴퓨터 쪽은 조금 낮은 학생들이 오므로 우리분야의 미래가 조금 걱정이다. 출중한 몇 명이 계속 오고 있는지. 컴퓨터과학과 공학을 지적으로 매력있고 도전할 만한 분야라고 포장할 필요가 있다.
- ▶ **박성우:** 회사에서는 똑똑한 사람들 안 온다. 코더들만 온다. 불만이고, 대학에서는 똑똑한 학생들 안 온다. 말한다. 대학원에서는 공부 좀 잘하는 학생 왔으면 한다, 말하고. 정부에서는 우리나라 소프트웨어가 중요하다고 하는데, 소프트웨어 시장은 큰데, 소프트웨어 잘 안된다고 한다. 이런 만성적인 문제에 해결책은 하나가 아닐 거라 생각한다. 큰 생태계 같은 거다. 악순환이 있는거다.
- ▶ **황규영:** 그 악순환을 가장 먼저 끊어야 할 게, 초중고와 대학 저년차 프로그래밍 교육이라고 생각한다.
- ▶ **박성우:** 정부가 소프트웨어 중요하다는 것을 별로 인정하지 않는다. 아직까지 소프트웨어를 지식산업의 핵심으로 홍보나 투자를 안한다. 대부분 홍보 프로그램은 천문학, 우주, 바이오 류다. 프로그램 짜는 것 등은 홍보가 안 되고, 고등학교까지 교육이 없고, 선생님도 자질이 안 되는 것 같다. 고등학교까지 컴퓨터 분야관련 제대로 된 교육이나 홍보가 없다.
- ▶ **황규영:** 고등학교에 컴퓨터 가르치는 교사가 컴퓨터 교육학과 출신이 아니라 비전문가인 경우가 많다. 비전문가가 책만 가지고 가르치니, 문제가 없을 수 없다.
- ▶ **민경오:** 오히려 중고등과정에서는 컴퓨터를 안 가르치는 것이 낫다. 오히려 지루한 분야구나, 생각한다. 어릴 때 선생님 영향을 얼마나 많이 받나.
- ▶ **박성우:** 저는 황교수님 말씀처럼 프로그래밍 교육

뿐만 아니라, 학부 교육 전체가 완전히 바뀌어야 한다고 본다. 물리학과 졸업한 학생이 학부 3학년 가도 듣는다.

▶ **이광근:** 제 학부 3학년 프로그래밍 언어강의를 법대 1학년이 듣더라. 성적은 나빴지만, 만만할 것이라고 생각했나 보다.

▶ **박성우:** 반면 CS졸업한 사람이 물리과 가서 들으려면 못 듣는다. 그래서 우리가 진입장벽도 낮다고 생각하는데, 그게 결국은 학부 교육이 20년 전이랑 거의 비슷하기 때문이라고 본다. 1, 2학년 프로그래밍 과목은 대부분 C나 자바를 이용해서 한다.

▶ **이광근:** 그러면 코더를 만드는 모습으로 가기 쉽다.

▶ **박성우:** 그래서, 포항공대는 1학년 1학기 때부터 ML로 가르친다. 거기에서 1등하는 학생들 코딩 한번도 안 해봤다. 이런저런 학원에서 중고시절에 코딩 배웠던 학생들이 아니다. 개네들은 고등학교 때부터 자바 C 프로그래밍 다 짜본 아이들이다. 그 학생들이 오히려 불이익이다. 황교수님이 이야기하신 기초적이고 총체적인 내용을 신식으로 가르치기에 ML 등의 언어를 사용하는 것에 좋은 효과를 보았다. 학생들이나 주변 교수들의 반발이 많았다. 들어보지도 못한 언어이고.

▶ **이광근:** 국내에서만 들어보지 않은 언어다. 미국에서는 기초과정을 ML로 시작하는 경우가 많다. 좋은 학교일 수록 더욱 그렇다.

▶ **김재우:** 미국 마이크로소프트 본사 행사에 Professional Developer's Conference가 있다. 2-3일 동안 집중적으로 개발자들을 교육하는 매우 밀도있는 프로그램이다. 그런데 전체 세션의 1/3이 함수형 언어 프로그래밍이다. 이러한 언어가 실제에서도 대세다. 우린 뭔가 학교에서도 늦춰져 있다. 마이크로소프트에서 나오는 소프트웨어 개발 툴킷에도 함수형 프로그래밍 환경이 들어간다.

▶ **민경오:** 그렇게 어려서부터 재미있고 멋있고 챌링지하다는 걸 알려줘야 하는데. 가만히 두면 오히려 나옴데, 우리는 어릴때부터 비전문가 등을 통해 가르치니, CS가 지루하고 어렵다는 인식을 심어주는 역효과가 있다. 교육이 큰 문제다.

▶ **박성우:** 학부 프로그램 3학년 과정생도 너무 모른다. C프로그래밍 언어 배우면 처음엔 재미있어하다가, 복잡한 거 나오면 좌절하고, 살아남은 학생들이 또 두 학기 지나면 또 좌절한다. 교육전체가 밑에서부

터 바뀌어야 하는데, 제일 처음 시작해야 하는 게 프로그래밍 교육이고, 제가 함수형 프로그래밍 언어 전공이라서가 아니라 교육적인 면에서는 그런 언어를 사용해서 CS전공과정을 만드는 것이 훨씬 낫 것 같다. 이러한 언어를 쓰면 처음부터 생각하고 디자인하는 데만 집중하게 된다. 그 다음 과정에서 C 프로그래밍을 하게되면 제대로 준비된 학생들이 된다. 그리고 학생들이 이런 언어를 쓰면 프로그램 규모도 다른 언어로는 엄두를 못내는 크기의 것을 쉽게 즐겁게 해내게 된다. 학생들이 아주 좋아한다.

▶ **황규영:** 박교수님 방법을 똑똑한 학생들은 잘 할 것 같다. 중간층은 잘하지 못할 것 같다. 우리는 중간층도 흡수해야 한다.

▶ **박성우:** 저 개인적으로는 C언어나 자바보다 훨씬 효율이 훨씬 높다고 확신한다. 어려운 언어가 아니고 ML이 오히려 더 쉬워서 중간층 학생들이 더 잘 배운다.

▶ **김재우:** 제 경우는 학생들이 C를 쓰면 그 언어가 어렵다고 한다. 그래서 한 학기 실습에서 할 수 있는 프로그래밍이 매우 적은 것만 가능하게 된다. 그런데 ML은 언어 자체는 너무 간단해서 가르칠 게 없다. 그래서 2주 지나면 곧바로 큰 소프트웨어 개발이나 도전적인 문제풀이에 들어간다. 황교수님 생각하시는 프로그래밍 교육이 깊이 생각하고 논리적으로 단단해지고 설계가 잘 된 소프트웨어를 개발하는 것인데, C는 그것을 방해한다.

▶ **박성우:** ML을 사용했을 때 학생들 피드백을 보면, 중독된 학생들이 있다. 즐기고 좋아한다.

▶ **이광근:** 그나마 다행인 건 서울대와 포항공대에서 이렇게 진행하고 있다는 것이다. 한양대도 그렇게 하는 것으로 아는데.

▶ **이욱세:** 1학년 C를 없애려고 하는데, 학부 거라 마음대로 못 없애고 있다가 이번에 없애면서 첫 언어는 자바고, 1학년 C고, 2학년 1학기가 Java이고 2학년 2학기에 ML로 간다.

▶ **황규영:** 우리는 자바를 1학년 때 가르쳤는데 결과가 안 좋았다.

▶ **이욱세:** 자바가 C보다는 단순한데 객체지향 개념이 문제다. 자바도 너무 잡다한 게 많다.

▶ **김재우:** 실무쪽에서 이야기하는 거 보면 학교에서 가르치는 자바가 별로 도움이 안 된다. 제 생각에는 오히려 학교에서는 철저하게 이론 중심을 가르쳐

주었으면 한다. 실제 일을 할 때 사이언스 백그라운드 강한 사람은 잘 적응한다. 그러나 프로그래밍 스킬 셋 중심으로 특정 언어에 대해서 배우면 수명이 바로 끝난다. 씨떡을 데가 없다.

- ▶ **황규영:** CS분야를 법대나 의대같이 대학원과정만 가지는 것도 의미 있는게 아닐까 싶다. 물리, 수학을 전공한 학생들을 주로 받는.
- ▶ **이광근:** 사실 그런 이야기를 하시는 분들이 있다.
- ▶ **민경오:** 모든 곳에 소프트웨어가 필요하니까 프로그래밍이 절대기술이 아니고 모든 사람들이 다 알아야 하는 것으로 인식하는 사람들도 많다. 바이오하는 사람도 소프트웨어 배워야 하고 물리하는 사람도 소프트웨어 알아야 하고 소프트웨어는 모든 학문에 다 필요하다.
- ▶ **황규영:** 그러나, 모든 분야에서 그것을 가르치는 것에 한계가 있을 것이다. 가르칠 시간이나 질과 양의 규모에 있어서.
- ▶ **이성환:** 일반 대학에서 타과 학생대상의 교양 전산이 있다. 영문학과에도 있고 심리학과에도 있고. 그런데 이제는 더 이상 그 과목을 컴퓨터전공학과에 협조를 의뢰하지 않는다. 자기네 대학원생 중에서 프로그램 좀 하는 학생들을 강사로 쓴다. 이것이 프로그래밍이라는 것에 대한 인식을 극명하게 보여주는 예다. 그러한 생각이 짧은 시간에 바뀔까? 어렵다고 생각한다. 물리학과 생물학과 심리학과 입장에서 전산학은 무엇인가? 소프트웨어는 복잡한 설계 방법을 알 필요가 없다고 생각한다. 그런 사람들을 이해시키지 못하고 체계를 잡지 못하면 곤란할 것 같다. 우리가 물리학이나 생물학을 못 가르친다. 그들은 프로그래밍을 가르친다. 뭔가 우리 분야의 위상이 잘못 되었다.
- ▶ **황규영:** 그게 바로 다른 과에서 자기 학생들 대상으로 가르치는 것은 “코딩” 수준인 것이다. 우리는 코딩 수준이 아니고 프로그래밍 수준을 가르쳐야 하는데 그게 안 되었어서 차별이 없어져 버린 것이다.
- ▶ **이성환:** 좀더 원론적으로 말씀드리면, 고등학교 과정도 교과 과정을 개선해야 하는 게 큰 숙제다. 정부가 할 때만 기다리는 건 문제고, 학회나 교수 중심이 되어서 정부에 우리가 적극적으로 요구해야 한다. 두 번째, 대학에서의 교과과정도 개선되어야 한다. 현실 문제는 4학년에 너무 대학원 위주의 세분화된 과목을 많이 가르치다보니 실제로 2~3학년의

기초과목들을 충실히 가르칠 여지가 없다. 4학년 과목 다 없애고 2~3학년 기초과목을 좀더 세분화시키고 확장 심화해야 한다. 저학년일 때 프로그램들을 다양화시키고, 흥미유발이 되게 잘게 잘라서 해야 한다. 4학년 교과목을 통해서 대학원 랩 소개하려는 욕심 버려야한다. 이걸 하지 않으면 위기는 계속 된다.

- ▶ **황규영:** 그 말에 전적으로 동의한다. 외국에서는 굉장히 단순하지만 깊이있게 학부과정을 익힌다. 우리가 학부 때 너무 많은 과목을 얇게 가르치고 만다.
- ▶ **이성환:** 저학년 때의 과목들을 통해서 학생들이 안착할 수 있도록 과목을 설계하여야 한다. 지금은 교수들의 전공 욕심에서 3~4학년 과목이 너무 짙다. 그러면 2~3학년에 여유가 없어진다.
- ▶ **박성우:** CS학부 교육이 근본적으로 제가 18년전 학부 다닐 때 과정과 거의 같다. 어느 과목이 필요하다 아니다를 떠나서, 예를 들어, 제가 학부때 디지털 실험을 하면서 땀질을 했는데, 지금도 학생들은 똑같은 디지털실험을 땀질하면서 배운다. 이 분야가 중요한 경우도 있지만 우선순위의 문제라고 보는데, 그런 과목은 우선순위가 떨어져야한다고 본다. 또, 제가 프로그래밍 언어 전공이라서, 컴파일러 과목을 예를 들면, 예전에 파서 만드는 것으로 배웠는데 지금도 비슷하게 파서를 만드는 것으로 배운다. 예전에는 파서 기술이 첨단기술이라서 그것을 배우면 컴파일러의 핫 이슈를 배우는 것이었지만 지금은 아니다. 컴파일러과목은 학부 4학년때 대형 소프트웨어 설계 능력을 키울 수 있는 몇 안되는 과목 중에 하나인데, 현재 엉뚱한 것 하면서 시간을 날려버린다. 그래서, 컴퓨터분야가 많이 바뀐 대세에 맞게 과목 내용도 많이 바뀌어야 한다. 그리고 결정적으로 물리학과 학생들이 CS 학부를 들 수 없을 만큼 수준이 높아져야 한다고 생각한다. 그래서, 2~3학년 과목에서 군더더기 과목, 20년 전부터 계속 같은 내용인 과목은 정비해야 한다. 학생들이 분야를 오해하게 되고 흥미를 잃게된다.
- ▶ **김재우:** 저는 교과과정을 좀 크게 봤으면 한다. 학교 안만 보지 말고 바깥도 관찰해주셨으면 한다. CS의 교과과정의 타겟이 T자형으로 퍼져야 한다. T자 가로로는 각 분야에 있는 도메인 전문가들이 나름대로 프로그래밍을 다 하고 있다. 문제는 이런 사람들이 사실은 프로그래밍의 많은 전문지식을 알고 있어야 한다는 거다. 아까 다른 분야 사람들은

코딩 수준만 한다고 했는데, 다른 도메인 전문가들이 소프트웨어를 다룰 때 필요한 것이 프로그래밍인 경우도 많다. 그런데 이런 레벨은 각 도메인에서 아무도 안 가르친다. 그런 분야의 전문가들에게도 제대로 프로그래밍 할 수 있는 교육과정이 있어야 한다. T자 밑으로는 산업에서 이쪽 전공분야가 아닌데도 굉장히 CS내용을 자발적으로 차근차근 공부하는 학교바깥의 커뮤니티가 있다. 서로 개별적으로 공부해오고 차곡차곡 학습하는 공동체가 있다. 실체가 있는건 아니고 블로깅 등을 하면서 영향을 준다. 이 사람들이 특정한 도메인의 소프트웨어개발자들에게 많은 영향을 준다. 모르지만 숨어있는 소프트웨어 리더라 할 수 있다. 이러한 사람들 특성이 굉장히 전통적인 CS를 학습하고, 동시에 산업을 적용하느라 애쓴다. 이러한 사람들을 위해서 학교에서도 신경을 써서 교육하는 교과과정도 필요하다. 이런 사람들을 제도권에서 신경써주는.

▶ **이성환:** 한편은 졸업생들을 취업시키는 데 위기이다. IT산업이 잘 나갈 것이라 믿고 싶지만 현실은 낙관적이지 않다. 졸업생들의 취업을 잘 시키기 위한 관점에서 보면, 4학년 2학기 과목은 예를 들어 모두 없애도 된다고 본다. 대신에 타 산업분야와 융합되는 것을 준비시키는 과목을 가르치는 게 좋다고 본다. 예를 들어, 조선, 자동차, 항공, 건축, 바이오 분야에서 컴퓨터 소프트웨어기술이 어떻게 스며들지 어떤 길이 있는지 배울 수 있는 과목이 좋지 않을까 한다. 4학년 2학기에 대학원 랩과 연계된 세세한 과목을 제공하는 것 보다는 더 필요하다고 본다.

▶ **이광근:** 지금 우리가 계속 이런 이야기가 나오는 걸 보면, 결국은 우리 분야가 아직은 다른 분야에 비해서 별거 없는 분야가 아닌가 한다. 기껏 40년 50년 정도 된 분야다. 다른 분야와 비교하면 참 초보 분야다. 앞으로 별개 있을 것 같은 분야임은 명확한데, 그래서 그렇게 성장할 분야인데, 학교가 미리 CS는 이것이다라는 명확한 슬로건을 갖고 뭔가 그쪽으로 향해 가는 것이 필요하다. 예를들어 생물학은 “Science of Life”이고 생물학의 응용이 “인간 수명의 연장”이라고 할 수 있듯이, 최근에 들은 이야기인데, CS는 “Science of Intelligence”이고 CS의 응용은 “인간 지능의 확장”이다라는 슬로건을 만들 수 있다. 이런 슬로건 아래 열심히 멋있게 하다 보면 이에 흥미를 느끼는 학생들을 모을 수 있고 그러면 좋은 학생들도 기웃거릴 것이다.

▶ **민경오:** 그러한 슬로건에 고개를 끄덕이도록 잘 준비해야 한다. 회사안에서 소프트웨어 센터를 맡고 나서 워싱턴에 보고를 드리면, 무슨 말인지 모르겠다고 한다. 어려운 벽을 느낀다. 물론 외부에서도 우리를 이해하는 게 필요하지만, 우리 자신도 우리가 하는 일을 잘 표현할 수 있도록 노력해야 한다.

▶ **이광근:** 우리분야를 더욱 흥미진진하고 그랜드하게 이야기할 수 있는, 우리 분야가 충분히 그런 내용과 미래가 있다고 본다. 그런 슬로건을 만들어서 교육과 연구와 산업화의 방향을 맞추는 일이 필요하다.

▶ **박성우:** 법대생이 들어와서 수업을 듣는 거, 제 4학년 과목에 타학과 학생이 들어와 A+를 받았다고 하면. 자존심 상할 것 같다. 대학 교수쯤이면 학부 심화 과목을 가르칠 때 큰 시스템을 설계하는 과목 자체를 만들 수 있다고 생각한다. 아주 도전적인 과목이 되도록 할 수 있다. 2~3학년 때 핵심이 되는 과목을 듣고, 4학년 때 대형의 시스템을 구성하는 깊이있는 과목들이 포진하면 타과 학생이 만만하게 볼 수 없고 본과 학생들도 재미있어할 것이다. 졸업을 해서 나가도 이미 많은 것을 해봤기 때문에 산업체에서 필요로 하는 시스템 아키텍트성 교육을 받고 나가는 것이기도 하다.

▶ **김재우:** 질 좋은 걸 가르쳐야 하지만 지금 커리큘럼 자체는 그리 나쁘지 않아 보인다. 한 가지 불만은 1학년에서 4학년까지 엮을 수 있는 스토리가 없다는 거다. 소프트웨어의 처음부터 끝까지 산업에 나가기 전에 경험을 해봐야 하는 데 그것이 어렵고, 두 번째로는 교육적인 면에서 단편 단편 기술들을 엮어줘야 한다. 개별적인 가치를 도저히 학생 스스로가 해결할 수 없고 경험 많은 교수님들이 가이드 해주지 않으면 어렵다. 그러나, 저는 선부르게 실무 교육을 하겠다고 해서, 실무에 쓰는 프로그램 언어를 가르치는 건 반대한다.

▶ **황규영:** 실무 교육을 하자는 게 아니라 프로그램 구성과 디자인 그리고 구현에 필요한 기본기를 철저히 가르치자는 것. 프로그래밍을 아트로서 아키텍처로서 가르치자는 것이다.

▶ **민경오:** 사회적인 계몽 활동이라고 할까. 우리 분야의 중요성을 외부에서 잘 모른다. 우리의 산업을 쉽게 이해할 수 있도록 사회적으로 환기 시켜야 한다. 역할 모델들이 좀 많아야 하겠고, 신문 기사 등을 통해서 이 분야의 중요성이나 매력을 전달하는 게 부족하다.

- ▶ **이성환:** 학교마다 커리큘럼이 너무 다르다. 이게 문제인거 같다. 스스로 정리가 안 된 것임을 나타내 주는 것이다. 이 교수 말씀대로 우리 분야가 성숙하지 않기 때문이다.
- ▶ **김재우:** 좋은 학교는 그래도 기반이 잡혔으나, 다른 학교들은 실무교육을 한다는 기치아래 중구난방이다. 어디에 쓰라고 가르치는지 잘 모르겠다.
- ▶ **서진욱:** 한국에 이제 막 돌아와서 이번 겨울학기 컴퓨터 개론을 가르치는데, 과목을 어떻게 가르치는지 다른 분의 강의록을 봤다. 많은 경우 옛날에 제가 배웠던 것을 그대로 가르치고 계셨다. 그런데 어떤 분의 강의록을 보니 참 인상적이었다. 누구나 알 수 있는 로직 게이트 등을 상위에서 우선 가르치고, 그것들을 조합을 해서 연산회로를 만들어가는 것을 간단하게 상위에서 가르치고, 거기에서 간단한 CPU를 구성하는 것을 가르치시더라. 이게 5시간이면 충분히 커버가 되고 학생들도 쉽게 따라오더라. 이렇게 배우고 나서 자바나 C를 배우면 분명히 머릿속에 컴퓨터에 소프트웨어가 작동하는 그림을 명확히 그릴 수 있고 좋은 것 같다.
- ▶ **황규영:** 옛날에는 디지털 회로 디자인이 대단한 일이었다. 그러나 이제는 비교적 쉽게 한다. 그러므로 상위개념을 이해할 수 있을 정도만 가르치면 좋을 것 같다. 이 과정을 상위에서 요약하고 축약해서 커버하는 과목들을 만들어야 할 것 같다. 그리고 그 위의 내용을 심도 있게 가르치고, 예전 방식으로 전깃줄 쫓는 실험을 하는 과정은 피하는 게 좋겠다.
- ▶ **이성환:** 이런 논의를 통해서 정리 좀 했으면 하는 게 교과과정을 틈틈이 몇 년마다 한번씩 고민했으면 한다. 서울대 카이스트 포항공대 등은 일반 과목이 많지 않다. 필수 위주로 되어 있는데 그 이외의 학교는 선택과목이 굉장히 많고 세분화 돼 있고 정리가 안 돼 있다. 어떤 식으로든 체계화하고, 공유하는 것이 필요하다.
- ▶ **이광근:** 그것을 위해서 국내 교과서를 잘 만들어 몇 십년에 걸쳐 시리즈로 나오게 하고, 그것들이 쌓이고 하면 참 좋을 것이다. 그런 운동을 일으켰으면 좋겠다.
- ▶ **이욱세:** ACM 가이드라인이 잘 마련된 점은 CS, OE 등 다양한 종류의 과정을 잘 구분해서 정비한 느낌이다.
- ▶ **박성우:** ACM의 가이드라인이 전 세계적으로 좋은 방향을 토픽관점에서 방향을 보여주는 것 같다. 그

리고, 저는 개인적으로는 4학년 과목들이 함부로 대할 수 없는 과목으로 만들었으면 한다.

- ▶ **이성환:** 대학원에 가서 배워야 할 세부단위를 교수님의 욕심 때문에 4학년에 은근슬쩍 내려와 있는 경우가 있다. 이런 점은 피해야 한다고 본다.
- ▶ **박성우:** 4학년 과목들이 그렇게 세분화된 모습으로 대충하는 것은 아니어야한다. 제대로 깊게 가야 한다. 기본적으로 틀은 레벨 업을 해야 한다고 생각한다. 레벨업은 다양한 표면적인 내용을 다양하게 하는게 아니라 기본 기초 내용을 충실하고 깊게 가야 한다. 물리과학생이 우리과에 와서 쉽게 4학년 과목을 듣는다는 것이 단적으로 우리 분야 교과과정의 문제가 있음을 보이는 것이다.
- ▶ **서진욱:** 디지털 컴퓨터회로 같은 거는 1학년 때 쉽고 재밌게 가르치고, 점차 올라 갈수록 좀더 깊은 내용으로 깊이 가르치고, 차근차근 어렵게 가르치면서 심화시키면 4학년 과정에 와서는 철학과생들 같이 타과생들이 쉽게 듣지는 못하게 된다. 아주 중요한 과목은 두 세 과목으로 나누어서 제공하면서 기초적인 것과 심화된 내용을 분리해서 차근차근 듣도록 해야 한다. 외국 대학이 그렇게 한다.
- ▶ **김재우:** 중요한 과목과 응용과목의 경중이 대학과정에서 잘 정비가 안 된 듯하다.
- ▶ **황규영:** 애기나온대로 컴파일러나 디지털 과목 등은 상위레벨에서 축약해서 재미있게하고 그 다음 단계의 내용을 심화해야 할 것 같다.
- ▶ **박성우:** MIT에서는 보니까, 학부기초과목으로 두 세 개만 있어서 굉장히 단단히 가르치고, 나머지는 대규모 프로젝트를 수행하는 각 분야의 연구과목 같이 되어 있더라. 1학년 단위는 재밌게, 그리고 엄청난 고도의 사고를 필요로 하는 학문이라는 것을 소개하고, 2-3학년 단위는 들으면서 기본 공력을 쌓게 한 다음에, 4학년 단위는 깊이 있는 레벨을 만들어서 가르치면 그 사람들이 산업에 나가도 좋을 것이다.
- ▶ **황규영:** 그러니까 소프트웨어도 1만라인 2만라인을 해 봐야 설계도 할 줄 알고 프로그래머도 하게 되는데 학부과정에서는 그만큼 가르치기가 어렵다. 학부과정에서 만라인을 직접 처음부터 끝까지 하면 시간이 굉장히 많이 걸리니까 어떻게 말하자면 라이브러리를 도와주든가 이런 식으로 해가지고 시스템을 구성할 수 있는 훈련을 시켜야 할 것 같다.

MIT 같은 졸업생들은 눈치 딱 보면 프로그램을 굉장히 잘한다. 내가 볼 땐 CS 잘한다는 사람은 무조건 프로그램을 잘하는 사람들. 어디서 배우는지 모르는데 하여간 어디선가 그 능력을 배워가지고 온다.

- ▶ **이광근:** 기본적으로 똑똑해서 그런 거 아닐까. 똑똑한 학생을 전공하게 해야 하는지. 오늘 여러 가지 좋은 아이디어가 많이 나왔는데 아까 텍스트북 얘기한 것, 각각 excellent example들의 본보기를 보이고 그런 것들이 쌓이고, 널리 공유하고 퍼질 수 있게 하려면 책 쓰는 것밖에 없을 듯하다.
- ▶ **박성우:** 70년대에 물리학과 다니신 분들은 그때 내용 그대로 가르쳐도 되지 않나. 30년 동안 내용이 많이 바뀔 만큼 물리학이 발전했다고 생각하지 않는다. 수학은 거의 같을 거고, 근데 CS는 색깔이 항상 바뀐다.
- ▶ **이광근:** 색깔이 바뀌는 게 아니고 내 생각엔 우리 분야가 아직은 미개해서 그렇다. 물리로 치면 1800년대 말이 아닐지, 아니면 뉴턴이 나온 직후. 수학의 역사에서 재미있는 사실을 알게 되었다. 우리 분야를 대입해 비교해 볼 수 있는. 뉴턴이 미적분 방식의 계산법을 고안하고 나서 한동안 100년 200년 동안 극한으로 작은 것에 대한 정확한 이해 없이 미적분 방식을 여러 분야에서 열심히 대입하고 응용하고 신나게 갔다고 한다. 지금 우리가 꼭 그런 거 같다. CS의 많은 분야가 제대로 된 이해 없이 뭔가 되니까 우당탕 이것저것에 쓰고 산업화하고 난리치고 그러는 수준인 것 같다. 수학에서는 그러다가 Cauchy등이 극한에 대한 정확한 개념을 정리하는 작업을 하고 나서야 다시 한 단계 도약했다고 한다. CS의 지금 수준이 그렇게 낮은 것 같다.
- ▶ **김재우:** 어떤 분야는 관성이 있다. 새롭게 나타나는 융합점의 직업군들이 많이 생긴다. 그런데 내부의 관성은 CS, CE가 원래 가지고 있던 전통적으로 가지고 있던 공동체를 유지하려는 것이다. 제 생각에는 앞서 가시는 분들이 그것을 좀 깨뜨려 주시는 게 필요할 것 같다. 관성속에서 변화를 시키려면 잘 안될 것이다.
- ▶ **박성우:** 10년 20년 하던 스타일의 교과내용을 벗어나야 된다고 본다.
- ▶ **민경오:** 우리 분야가 학문으로서 자리를 잡으려면 공력이 좀 쌓여야 한다. 다른 학문에 비하면 우리 분야가 짧은 것이다. 100년 200년은 가야 여러 가지

진보도 많을텐데 아직은 비교적 짧다. 현재 우리가 이런 점도 감안을 해야 한다.

- ▶ **서진욱:** 제가 한국에 들어온 지 얼마 안됐는데, 내가 다니던 시절과 대학 커트라인 배치표를 살펴보니 많이 내려가 있다. 내부적인 문제도 있지만 사회적인 문제도 있는 듯하다. 우리는 캄캄한 데서 프로그램 하는 줄 안다. 만화나 글에 그런 이미지가 있는 것 같다. 그리고 직업안전성도 문제다. 부모한테서 듣고 졸업하고 나서 등으로 걱정하는 것 같다.
- ▶ **황규영:** 학생들이 우리분야를 밟새고 코딩하고 남는 것 없는 분야로 생각한다고 한다.
- ▶ **박성우:** 미국에서는 직업 만족도와 연봉 수준을 보면 밑의 30등이 의사, 2등이 대학교수, 1등이 소프트웨어 엔지니어이다. 구글이나 MS다니는 사람들이다. 사회적으로 미국에서는 창의로 승부해서 구글같은 성공을 할 수 있다는 문화가 있다. 미국의 탑 스쿨은 CS가 건재하다. 우리는 아무리 CS가 재밌다 해도, 왜 전자과를 가느냐? 삼성이라는 회사가 있고 커리어 패스가 있기 때문이다.
- ▶ **김재우:** 그런 면에서 교수님들께 부탁드리고 싶다. 가능한 일일 것이다. 우리나라처럼 소프트웨어 인터스트리를 쉽게 요약할 수 있는 나라도 없다. 우리나라는 소프트웨어 매출의 대부분은 공공과 금융에서 일어난다. 최대의 소프트웨어 발주처는 정부다. 그런데 정부가 이 중 잣대를 쓴다. 예를들어 똑같은 프로포절을 받으면 독일쪽과 거래한다. 독일회사에는 많은 돈을 요구한대로 지불한다. 그런데 국내 회사에는 같은 기술인데도 불구하고 엄청나게 값을 깎는다. 왜 이러는지 모르겠다. 오히려 국내 산업을 많이 쳐줘야 하는 거 아닌가. 한국사람들이 하는 것을 오히려 쳐주지 않는다. 평가에서 질이 떨어지는 것도 아닌데. 그래서 한국회사가 하는 일이 그런 외국회사 밑에 들어간다. 교수님들이 협회활동 등에서 의견 주실 때 이를 강하게 어필 해주셨으면 한다. 인터스트리가 안 크면 사람이 안 온다. 우리나라 시장은 크지 않다.
- ▶ **민경오:** 앞으로 지적 재산에 대한 가치를 인정해 줄 것이다. 만들지 못하는 것도 있고 인정해 주지 않는 것도 있는데 서비스에 대한 인정이 바뀌어야 한다. 서비스 질도 올라가야 한다. 소프트웨어 산업은 서비스 측면이 강한 건데 우리는 아직 3D 업종으로 취급이 되고 있어 문제다. 정책도 잘해야 하고.

- ▶ **이성환:** 사회적으로 시간이 필요할 것 같다. 대중의 인식을 바꾸려면 미국처럼 나이든 할아버지가 소프트웨어 설계하는 모습 보여주고, 이런 사회적인 분위기가 되어야 할 것 같다. 우리나라는 5~60대 프로그래머들이 액티브하게 활동하는 것을 아직까지는 못 보여주고 있지 않나. 20년 후면 그런 시절이 올수도 있을지 모르지만, LG나 삼성, 중소기업에서도 경험 많은 나이 든 프로그래머들이 활동할 수 있게 뒷받침해줘야 할 것이다. 우리나라 소프트웨어 산업이 미국과 다르게, 미국의 인구비례 CS 졸업생들이 우리나라보다 훨씬 작다. 미국 CS 졸업생들의 직장을 보면 우리와 다르게, 우리는 직장에서 졸업생들 일하는 거 보면, 저임금에 하부 코딩을 주로 하고 좀 괜찮은 배경을 가진 사람은 위에서 디자인만하고 하는 분위기가 존재하면 안된다. 존경받는 엔지니어 모습이 되게끔 이런 문화가 빨리 불식되지 않으면 우리나라 전산과에 대한 일반 대중의 평가는 좋아지지 않을 것이다. 이것을 최대한 빨리 바꿔나가는 게 우리 전산학과의 미래가 밝게 될 것이다.
- ▶ **이광근:** 좋은 책을 써야한다. 좋은 텍스트 북도 쓰고, IT관련 “체제비판” 책도 써야한다.
- ▶ **박성우:** 우리나라 CS가 왜 힘들까 가만 생각하면 나비 효과로 결국 SCI 논문 때문이라고 본다.
- ▶ **이광근:** SCI 논문 쓰는 것보다 좋은 책 쓰는 게 더 중요한 것일 수 있다.
- ▶ **박성우:** 대학 교수들을 너무 SCI 논문으로 평가한다. 저도 컴파일러 하나 만들려고 품잡고 있지만 SCI 논문 편수 압력 때문에 하다가 만다. 소프트웨어 시스템 잘 만들려면 5년 10년 계속 해야 하는데, SCI 페이퍼 1년에 몇 편 내려 하면 정말 하고 싶은 게 중단되고 자꾸 못하게 된다.
- ▶ **민경오:** 정부 정책이 중요하다. 도와주려고 나서는데 오히려 역효과를 내는 게 허다하다. SCI도 하나의 예고, 가만히 놔두는 것이 오히려 도와주는 것이다. 프랑스의 사회학자 기 소르망이 그런 말을 했다고 한다. 대한민국은 어떻게 정부가 어떤 사람이 풀지 선택을 하는가, 이상하다고 했다고 한다. 우리 정부가 현실하고 동떨어진 정책을 하는 것 같다.
- ▶ **박성우:** 학생들 자기 커리어 가지고 조언 받으러 오는데, 그게 사실 CS로 와라, 적극적으로 말 못하겠다.

- ▶ **이성환:** 나는 반반해라. 막연히 이야기한다. 하나로 밀기에는 좀 부담스럽다. 졸업생이 잘 될 것인지 내가 확신을 못하니까. 자신이 없다. 학생들도 그런 것을 잘 안다. 나는 예를들어 IT와 BT 다 하라고 한다. 전자과는 이것도 하고 저것도 하지만(코딩도 하고 칩 디자인도 하는 등 flexible하게 생각할 수 있지만) CS는 극단적으로 코딩만 한대라고 인식한다. 아이들 입장에서는 전자과가 전산에 비해서 나중에 할 수 있는 게 훨씬 많다고 생각한다.
- ▶ **서진욱:** 제가 아는 유명한 분중에 biology를 해서 박사를 받고, CS해서 또 박사학위를 받으신 분이 있는데, 둘 중에 어떤 게 재밌고 어떻게 더 쉽냐고 물었더니, 솔직히 biology가 훨씬 더 재밌다고 한다. 아직 못 푼 문제가 너무 많이 남아있고 도전할 만한 게 많다고 한다. 그럼 CS는 왜 또 했냐, 물었더니, 그 문제를 풀려면, 이 CS 지식이 없이는 전혀 불가능하다는 걸 느껴서 다시 CS를 했다고 한다. 배우는 건 어떤 게 쉽냐고 물었더니, 컴퓨터가 훨씬 쉽다고 한다. 완전히 모르는 입장에서 시작하면, 자기 입장에서는 컴퓨터가 진입장벽이 좀 낮다고 했다. 또 다른 분은 phd로 CS를 받고, master를 biology를 하신 분이 있는데 그 분은 CS가 훨씬 더 재밌었다고 한다. 그러면 biology를 왜 했을까, 싶어 물었더니 컴퓨터를 배웠는데 그것을 적용하고 싶은 부분을 찾은 게 biology여서 컴퓨터를 잘하는 사람이 하면 훨씬 더 잘 될 것 같아서 하게 됐다고 한다. MIT 보면 융합된 과목을 4학년에 넣어놓은 게 많다. 금융분야에서 어떤 펀드에 투자해야 돈이 더 되는지 실제로 최적화하는 경우를 가르치는 과목, 기상 예측하는 분야와도 융합해서 모델링하고 슈퍼 컴퓨터 돌리는 과목들도 있고, 우리분야가 적용할 데가 많은데 이렇듯 4학년 과목에 다른 분야를 융합하는 것이 좋을 것 같다.
- ▶ **이성환:** 우리 CS가 독자적인 고유한 영역을 차지해서 죽 나가야 한다고 생각한다. 그러나 현실은 과연 우리만의 identity를 가지고 좋은 학생들을 끌어모으고 잘 되게 할 수 있을지 자신이 없다. 이런 상황에서 타협할 수 있는 게 뭐냐면 우리의 코드를 가지되 학생들에게 유연성을 넓혀주는 것이라고 생각한다. 어차피 전통적인 CS만 해가지고는 취업 시장에서 우리 미래가 밝다고 생각하지 않는다. 기존의 다른 쪽의 마켓을 우리 학생들이 넘어가서 그 쪽을 접속하고 먹어야지만 우리 시장이 커질 것이라고 생각한다. 결국 그렇게 하려면 그 쪽 사람들

이 프로그램을 배워서 하기보다는 우리가 프로그램 스킬을 배워서 그쪽의 도메인 지식을 조금 조금씩 받아가면서 공격하는 게 훨씬 우리 분야를 키울 수 있는 게 아닌가 생각한다. 그런 측면에서 기본적으로 2~3학년 핵심적인 것 배울 때 4학년 때 주변적으로 연결할 수 있는 어떤 도메인 넬리지를 조금 배우도록 학부에서 조금 신경써주면 우리 전산과 학생들에게 큰 도움이 있을 것 같다. 연구 개념에 서라기보다는 취업 관점이다.

- ▶ **이광근:** 그렇다고 또 우리쪽의 identity를 깊이하는데 필요한 내용을 소홀히하면 안된다. 우리분야가 아직 미개하고 크게 진보할게 많은데 그런 미래를 준비 못하면 문제다.
- ▶ **면경오:** 대학원 과정은 다학제간의 융합내용을 해 볼 필요가 있을 것 같다.
- ▶ **이성환:** 그런데 현실적으로 대학원에서는 다른 것과 연결하기 좀 힘들다.
- ▶ **민경오:** 대학원생 뽑을 때 다른 분야 한 사람을 우선순위로 뽑아서 CS에 끌어들이는 것도 방법일 것이다.
- ▶ **이성환:** 대학원에서는 융합적인 리서치 프로그램이 있다. 그런데 내가 이야기하는 것은 대학원 차원이 아니라 학부 취업의 관점에서 볼 때의 관점이다.
- ▶ **민경오:** 내가 볼 때 학부만으로는 좀 힘들지 않나? 석사까지는 해야 그래도 제대로 인력수준이 되는 것 같다. 현재 LG만 해도 학부 졸업생은 잘 뽑지도 않지만, 뽑고 나도 이들이 인도나 중국이랑 비교할 때 그 정도의 평균 엔지니어 수준이라고 하면 앞으로는 경쟁이 점점 힘들어 질 거 같다. 우리 같은 경우는 고부가가 있는 일을 하지 않으면 인도나 중국 등에 인건비가 비교가 되니까 힘들어질 거 같다. 석사 정도는 해야 경쟁이 될 것이고 그것이 되려면 CS가 바탕이 되고 특정 도메인에서 할 수 있는 인력이 돼 줘야 나름대로 기여가 될 수 있을 것이다.
- ▶ **이성환:** 자동차 산업의 경우 기계과 나온 사람이 코딩을 배워서 자동차에 들어갈 코딩을 쉽게 만들 수 있을 거라 본다. 그러나 우리가 보기엔 전산과 나온 사람이 고급적인 소프트웨어 개발 기법을 훈련하고 배우고 그 후에 그 쪽 자동차 구조를 조금만 배우면 훨씬 더 고급스럽고 믿을만한 소프트웨어를 짤 수 있을 거라는 게 우리 생각 아닌가. 이런 우리 생각을 관철해야 하는데 아직도 시장에서는 기계과 출신이 코딩만 조금하면 충분히 잘할 수

있다고 믿는다. 그것이 우리가 깨뜨려야할 속제다.

- ▶ **민경오:** 내 생각은 규모의 문제다. 자동차에 들어가는 소프트웨어가 커지기 시작하면 기계전공하고 프로그래밍을 귀동냥한 인력보다는 프로그래밍 전공하고 기계 귀동냥한 사람들이 필요해 진다. 이럴 때가 오고 있고 그걸 준비해야 한다.
- ▶ **이성환:** 한 달 전쯤 디스플레이 엔지니어사람들과 이야기할 기회가 있었는데, 앞으로는 자기네 디스플레이 쪽 사람들도 실질적으로 소프트웨어 일이 굉장히 필요하고 실질적으로 그런 사람을 충원하려 하는데 막상 지원자가 없다고 한다. 디스플레이 회사에서 이렇듯 소프트웨어를 주장하는 것 보니까 우리 소프트웨어 분야도 인식이 바뀌긴 바뀌었다 싶다. 그런 시장도 우리 소프트웨어 분야가 적극적으로 접근을 해야 한다. 지금은 그쪽은 광학이나 기계 반도체 물리 마켓이라고만 생각한다. 거기에서도 우리 소프트웨어 수요가 있으므로 적극적으로 접근을 해야 한다.
- ▶ **김재우:** 중요한 얘기다. 자동차 산업은 사실상 소프트웨어 산업이다. 핵심적인 설계, 디바이스, 운전 자용 인터페이스 등 전체적으로 모두 소프트웨어다. 그걸 고루 덮어쓸 수 있는 인력을 충분히 마련해야 하는데 그것이 잘 안되니까 기존에 있던 산업에서 품질이 문제가 되고, 품질개선을 하려면 소프트웨어 공학적인 관점에서 모델링도 잘해야 하고 프로세스 굉장히 잘해야 하는데 그게 안 되는 거다.
- ▶ **민경오:** 국가적으로 미국 같은 경우도 소프트웨어가 발달하게 된 데에는 냉전시대 때 몇 만 명이 모여서 국방관련 대규모의 소프트웨어들을 개발한 것이 큰 힘이 되었다고 한다. 예를 들어 소프트웨어 과제를 예를들어 5천 명 정도가 필요한 주요 과제를 하면, 그거 중요하지 않다고 할 사람 아무도 없다. 그러다 실패를 하면 또 수업료를 내고 배울텐데, 예를들어 정부에서 국가적으로 대형 소프트웨어 과제를 몇 개 하면 소프트웨어 산업에서 큰 모멘텀이 되지 않을까 한다.
- ▶ **김재우:** 다른 분야와 융합되는 내용으로 교과를 마련해도 양쪽으로 다 좋을 것 같다. 예를 들어 컴파일러 만드는 과목도 특정분야에 언어를 위한 컴파일러를 만드는 내용으로 만들면 다른 분야에서도 배우기 좋고 우리 전공자들도 배우기 좋은 원천기술을 잘 익히게 될 것 같다.

▶ **이광근:** 이제 각자 정리말씀을 부탁드립니다.

▶ **민경오:** 우리 토픽이 어떻게 하면 컴퓨터 사이언스를 공부하는 사람이 많아지게끔 할 수 있을까? 잘하는 사람들이 많이 올까? 인데 제 생각에는 사회적으로 이 분야의 중요성이라든지, 이걸 해도 다른 분야에 비해 상대적으로 나쁜 게 아니고 괜찮다, 해볼 만하다, 라는 노력을 우리가 좀 해야 되지 않나 싶습니다. 너무 부정적인 이야기만 돌아다녀서, 안 그런 면도 많은데, 중요성에 대해서도 알려주고, 실제로 잘 된 사람도 자주 불러다 이야기 좀 듣게 해주고 그런 것이 필요할 것 같다. 교육도 물론이고,

▶ **박성우:** 이 문제가 정부, 대학, 대학원 교육, 산업이 네 가지가 서로 맞물려서 생기는 문제이기 때문에 문제의 시발점이 어디인지를 찾는 거는 어려울 거 같다. 그래서 문제를 해결 하려면 어떤 큰 조직에서 골병이 들었을 때 외부 강사 초청해가지고 문제를 해결할 수 없는 것처럼 문화 자체가 바뀌어야 되는데 그러기 위해서는 정부도 정책을 바꾸어야 하고 산업체에서도 똑똑한 사람이 들어왔을 때 활용했다가 새로운 사람 들어오면 버리고 이러는 것도 없어져야 한다. 학교에서도 SCI 실적만 강요하는 게 바뀌어야 한다. 그리고 지금 큰 생태계가 골병이 든 거라고 본다면 모두 바뀌어야겠지만 우리가 할 수 있는 것 중 하나가 학부 교육 자체를 많이 개선하는 것이라고 본다. 내용 자체도 바뀌고 그리고 그 중요성을 인식하고 교수님들이 강의할 때 얼마만큼의 열정을 보이냐가 가장 중요하다고 본다.

▶ **이광근:** 저는 그런 것들을 잘 퍼뜨릴 수 있는 제일 좋은 방법은 좋은 내용들을 책으로 잘 마련하고, 연습 문제도 잘 만들고, 강의 슬라이드도 따로 잘 마련해서 출판을 하든 웹에 올리든 잘 퍼지게 하는 것이다.

▶ **박성우:** 4학년 졸업하고 나가는데 들은 과목도 없는 거 같고, 배운 것도 없는 거 같고, 이렇게 졸업하는 걸 보면 학부교육을 바꾸는 것이 정말 시급하다고 본다. 어느 특정 학교에 해당하는 문제가 아니라 전국 대학에 거의 모두 해당한다고 본다.

▶ **이옥세:** 지금까지 주로 나온 얘기들을 보면 교육과정 얘기인데 제가 생각할 때 교육과정을 탄탄히 하는 것도 중요하긴 한데, 근데 제가 볼 때 제일 먼저 해결되어야 될 것은 우리 전공자들이 대우를 잘 받게 해야 한다. 대우를 잘 받으려면 소프트웨어가 대우를 받아야한다. 소프트웨어를 돈 주고 사야 되고,

제값 받고 팔아야 된다. 어떻게 이게 될까는 생각해 봐도 잘 모르겠다. 불법 복제에 대해서는 굉장히 많이들 얘기한다. 그런데도 대학에서는 불법 복제하고 살고 있다. 그래서 제 수준에서 할 수 있는 노력은 제 값 받고 팔 수 있는 소프트웨어를 만드는 것이다. 이게 해결되면 그 다음에 여러 가지로 해결할 수 있을 거 같다. 그리고 밤샘 코딩 얘기가 나왔는데, 밤샘 코딩하는 것도 소프트웨어를 만들 때 이 얼마나 많은 노력이 들어가는데, 컴퓨터 쪽 전공 안 하신 분들은 이거 그냥 나오는 건 줄 알아요. 우리가 프로젝트 같은 걸 받을 때도 제값 받고 해야죠. 그럼 결국은 대우도 좋아지고 사회적인 인식도 좋아지지 않을까 싶은데, 솔루션은 잘 모르겠다.

▶ **이광근:** 소프트웨어가 제 값을 받으려면 남들이 쉽게 짜지 못하는 원천 기술이 있는 소프트웨어가 나와야 된다.

▶ **이옥세:** 그래서 품질이 좋은 소프트웨어와 품질이 나쁜 소프트웨어를 정확히 구분 차별해야 된다. 예를 들어 비누를 팔아도 좋은 비누랑 나쁜 비누랑 차별해서 팔린다. 나쁜 비누는 백 원이고 좋은 비누는 만 원 한다. 그러니까 소프트웨어도 차별이 되어야 되고 결국 그러면 소프트웨어 개발자들도 차별이 되어야한다. 그런데 그런 걸 어떻게 제도화해야 할지는, 그런 것까지는 잘 모르지만 이 방향으로 가야 한다.

▶ **민경오:** 그런 것들이 모두 시장에서 결정이 될 것이다. 우리가 아직 시장이 그 수준인 거다. 시장이 발전하는 과정에서 남들보다 일찍 눈이 떠서 좋은 품질의 것에 투자한 사람들은 살아남는다. 건전한 시장을 형성하는 게 중요하다. 그런 면에서는.

▶ **이광근:** 그렇게 될 때 학교에서는 이런 시장의 형성을 빨리 땡기거나 주어진 시간에 그런 걸 대비할 수 있는 인력을 만들어 내는 게 중요한 것 같다.

▶ **서진욱:** 컴퓨터 분야 자격증 중에 기술사 자격증이 있는데 이런 자격증 실제로 아무의미가 없다. 다른 분야는 예를들어 건축관련 기술사는 굉장히 중요하다.

▶ **이성환:** 결국은 우리 컴퓨터 분야의 학과가 잘되려면 가장 중요한 게 사회적으로 저기 가고 싶다, 중요하다라는 인식을 갖게 하는 것이 제일 중요하다. 그러려면 산업이 잘되어야 한다. 이렇게 하려면 우리 대학에서 할 수 있는 것은 교과과정을 나름대

로 잘 운영하는게 중요하다. 교과과정 어떻게 할 것이냐? 답은 없지만 기본적인 방향은 두 가지 정도라고 본다. 일단은 저학년 학생들한테 소프트웨어 설계하는 기법, 디자인하는 기법, 프로그래밍 등의 내용이 강조가 많이 되고 그런 과정들이 활발해졌으면 좋겠다. 또 프로그램 경험이 없는 학생들에게도 ML같은 언어를 통해서 접근하여 잘 준비시키듯이 등등. 그 다음에 고학년이 너무 대학원 단위의 수업을 하는 경향이 있다. 우리 선생님들 입장에서 내 분야를 많이 들었으면 하는 욕심에 고학년에 전공 분야 대학원 과정을 늘어놓는 것보다는 졸업하고 나간 학생들이 다양한 산업 분야에 가서 쉽게 적응하고 그 분야에서 전문성을 갖게 하기 위해서 그걸 연계해주는 과목들을 개설했으면 한다. 예를 들면 IT융합에 관련된 과목, 즉 건축이나 자동차나 핸드폰 등 몇 개 과목들에 대해서 4학년 2학기 과목에서는 그걸 연결시켜주는 커리큘럼이나 프로젝트들이 좀 부활이 되어서 우리나라 전체적으로 활성화되었으면 하는 게 제 욕심이다.

- ▶ **김재우:** 현실적으로 교과과정에 집중해서 보면, 설계나 모형화에 연습을 깊게 많이 시켜야 높고 비싼 직업군을 양성하게 하는 것이다.
- ▶ **서진욱:** 고학년에 바이오 쪽이든 물리든 화학이든 우리분야와 융합하여 프로젝트과목을 만들어줘야 한다. 그리고 학부 커리큘럼 내에서도 약간의 대형 소프트웨어 아키텍처 과목을 1학년처럼 쉽게 가르쳐주어 학생들에게 도움이 될 수 있도록 기초를 다질 수 있게 해주어야한다. 그리고 꼭 이야기하고 싶은 한 가지는 방학을 잘 이용하는 거다. 예를 들면 겨울방학, 여름방학 때 중고등학생을 불러서 프로그래밍 스쿨 같은 것을 열고, 또 내부적으로는 다음 학기에 대형 소프트웨어 구성 과목을 듣는 학생이 특정 언어 프로그래밍에 자신이 없다면 그런 학생들을 위한 언어 스킬을 익히게 하는 과목을 개설하는 것이다. 이런 보조 코스들을 방학때 짜임새 있게 배치를 해 주어서 따라 가고 싶은 욕심이 있고, 의욕이 있는 사람이 올라갈 수 있게 해야 한다.
- ▶ **이광근:** 두 시간 넘는 귀중한 시간 내 주어서 감사드린다. 좋은 말씀 감사드린다.