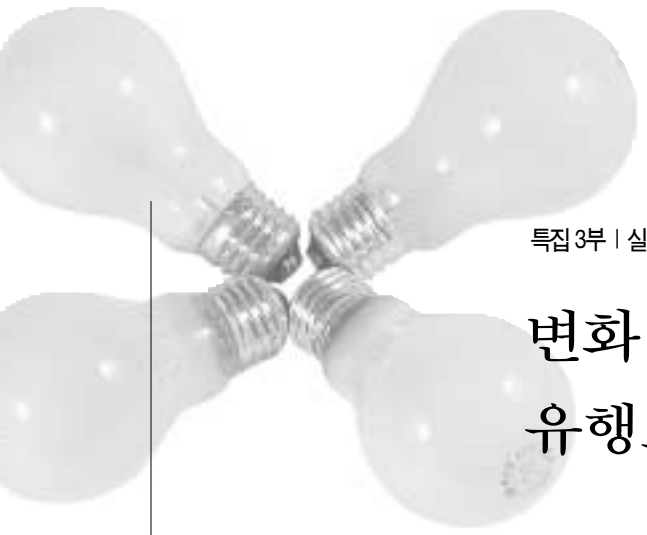




특집 3부 | 실수와 혼란을 뛰어넘어 1

변화에 적응하려면 유행보다 ‘기본’이 우선

이 글은 'Theory Into Practice' 교육 시스템을 적용한 장기간의 IT 교육을 진행하면서 겪었던 일들을 이야기 형식으로 꾸민 것이다. 처음 교육생들을 모집하기 위한 과정 발표부터 최종 프로젝트를 마치고 수료생을 배출하기까지의 크고 작은 에피소드를 통해 IT 교육의 현실과 문제점을 알아보았다. 이를 통해 제대로 된 IT 전문가를 양성하기 위해서는 수강생과 교육기관 모두 어떤 준비가 필요한지 생각해 보자.

김승권 seal@bluette.com

블루엿 인터내셔널의 연구원으로 개발 환경과 온라인 교육 시스템을 결합한 소프트웨어를 설계하고 있다. 모바일 환경의 정보 보안과 내장형 시스템 설계를 위한 HW/SW Codesign 기술에 관심을 갖고 있으며, 다국어 처리 시스템, 통합물류 시스템, 실시간 모니터링 시스템 등 다수의 프로젝트 개발에 참여했다. 현재 성공적인 개발 작업의 토대가 '국영수'와 철학임을 깨닫고, 구루 근처에 가보기 위한 인생 경험 생기와 책임기에 열심이다.

김상훈 desperado@tmic.tit.ac.kr

동명정보기술원의 선임연구원으로 실용적인 소프트웨어 개발 기술에 관심이 많다. 이론을 실제 업무에 적용시켜 소프트웨어 공학 기술이나 설계 기법이 실무에 제대로 적용되는 프로그래밍 기법을 만드는 것이 개발자로서의 목표. 현재 정보기술원에서 전문 개발자 양성과 인증을 위한 교육 시스템 'Theory Into Practice'를 설계하고 있으며, 다수의 국책 프로젝트에 참여했다.

필 자는 스노우캣이란 이름의 작은 고양이를 좋아한다. 여자 친구의 소개로 지난해부터 알게된 이 친구는 www.snowcat.co.kr이란 웹 사이트에 등장하는 고양이 캐릭터다. 병이다 싶을 정도로 혼자 뒹굴기를 좋아하는 스노우캣이 공개하는 일기를 매일같이 훑쳐보면서, 필자는 무척이나 외로움을 많이 타는 이 고양이를 조금씩 알아가고 있다. 갑자기 스노우캣 이야기를 하게 된 이유는 한 신문에서 읽었던 기사 내용 때문이다. 그 기사는 20세기가 충직하고 근면함을 상징하는 개의 시대였다면, 21세기는 빠른 시대의 변화에 잘 적응하는 약삭빠른 고양이의 시대가 될 것이라고 전망했다. 그리고 21세기에 접어들면서 점점 더 빠르게 변화하고 있는 IT 분야에도 비슷하게 적용될 것이라는 것이다. 다만 앞에서 언급한 개의 미덕이 가치를 잃는 것이 아니라, 거기에 고양이의 적응력이 더해져야 한다는 부담감의 차이가 있을 뿐이다. 이 시대에 필요한 개발자로서의 소양

은 무엇일까? IT 교육기관에 종사하고 있는 필자로서는 필자가 속한 기관에서 배출된 개발자들이 이러한 시대의 변화 속에서 성공적으로 살아 남도록 하기 위해서 제대로 된 교육 과정에 대해 진지하게 고민하지 않을 수 없었다.

수료 후 100% 취업 보장?

필자가 속한 교육기관은 매년 새 과정이 개설될 때마다 과정 발표회란 것을 통해 교육을 받고자 하는 사람들에게 과정의 취지와 특징 등에 대해 설명해주는 시간을 갖는다. 과정 발표회에는 나름대로의 기대와 의지를 갖고 모여든 많은 학생들로 늘 북적이기 마련이다. 예비 수강생들에게 과정을 제대로 설명하기 위해 밤새 준비한 프리젠테이션 자료를 모두 발표하고 나서 Q&A 시간을 가진다. 새로 만든 교육 과정을 처음 소개하는 자리였기 때문에 다들 기대 반 두려움 반으로 그 시간을 맞이했다.

그 날 받았던 첫 번째 질문은 “여기는 수강 신청하면 사은품으로 뭘 주나요? 모니터나 PDA 주는 곳도 있던데…”였다. 사실 나름대로 문서걸이나 마우스 패드 같은 것을 나눠줄 것을 검토하고 있는 중이었지만, 요구하는 것과 너무 차이가 나서 놀랐다. 우리가 진행하는 교육 과정은 모두 1년 정도의 장기 과정이다. 자신의 짧은 인생 중에서 1년이나 되는 시간을 투자하러 온 사람이 그 1년이 어떠한 시간인지를 소개하는 자리에서 하는 질문치고는 다소 실망스러웠다. 그리고 그 뒤에 쏟아진 질문도 우리가 예상했던 질문과는 거리가 멀었다. 대부분 과정보다는 작년까지의 결과를 궁금해하는 것이었다. “여기 수료하면 취업은 잘 되나요?”, “자격증은 몇 개나 딸 수 있어요?” 등….

우리는 과정 발표회에서 학생들과 나눈 많은 대화를 통해, 많은 교육생들이 ‘날개 짓을 하지 않아도 바람을 타고 높이 날 수 있을 것’이라고 생각하는 것을 알 수 있었다. 과정을 단지 듣는 것만으로 그 분야의 전문가가 되고, 모두 다 좋은 곳에 취직하길 바라는 그들의 기대는 마치 교육을 마술로 착각하는 듯 했다. 또한 잡지·신문·영화 등 대중 매체를 통해 잘못된 IT 소식을 듣고 오는 경우가 많았으며, 특정 요소 기술에 지나칠 정도로 민감하게 반응했다. 심지어 “닷컴 기술이 시장을 장악할 것이 확실시되고 있어 곧 자바 개발자들은 소외될 것이 뻔한데 왜 자바 과정을 개설했느냐”며 확신을 갖고 따지듯 물어오는 사람도 있었다. 도대체 어디서 그런 이야기를 들었는지 궁금해서 물어봤더니 다른 교육 기관에서 주최한 닷넷 과정 설명회에서 들었다고 했다. 물론 그 학생이 설명을 잘못 이해한 부분도 있었겠지만, 교육생들이 갖고 있는 잘못된 생각의 상당 부분은 IT 교육기관에도 책임이 있다.

“수료 후 100% 취업 보장”, “XXX 자격증 100% 취득”
“이 과정을 들으면 여러분도 IT 전문가가 될 수 있습니다”
“XXX 기술을 알면 미래가 보장됩니다”
“최고의 전문가들이 여러분을 가르치게 됩니다”

현재 대부분의 IT 교육기관의 모집 광고를 보면 이와 같은 문구들을 쉽게 발견할 수 있다. 물론 필자가 속한 교육기관도 예외는 아니었다. 그나마 이것도 상당히 다듬은 결과인 걸 알면 놀라는 독자가 많을 것 같다. 물론 모든 광고 문구의 내용이 허황된 것만은 아니다. 하지만 분명 과대포장 광고는 교육생들로 하여금 잘못된 환상에 젖어들게 만들어 결과적으로 스스로 무엇을 해야 하는지에 대한 고민은 접어둔 채 무작정

교육 기관에 기대게 할 수 있다.

필자는 교육생들이 교육 기관의 광고를 나름대로 잘 받아들이기를 바란다. 그리고 무엇보다 취업이라는 단순한 목적보다는 기술을 제대로 배우고자 하는 의지로 가득한 교육생을 바란다. 80년대 민주화 운동으로 대학 수업이 제대로 이뤄지지 않던 시절에도 우리의 선배들은 다양한 컴퓨팅 이론을 제대로 활용하는 기술을 습득해서 지금 IT 산업의 핵심 인력으로 활동하고 있다. IT 전문학원의 교육은 허황된 기대로 가득 찬 사람에게 아무런 도움도 주지 못한다. 다만 노력하고자 하는 의지가 있는 사람이 가고자 하는 길을 좀 더 밝게 비춰줄 수 있을 뿐이다.

고기 잡는 법을 알려줘야

현재 IT 교육기관이 수행하고 있는 많은 교과 과정이 기술 열거형 단기 교육에 치우쳐 있는 것을 볼 수 있다. 고급 기술 습득과 거리가 먼 자격증을 취득하기 위한 과정이 많다. 이러한 과정은 최소한 몇 년의 실무 경험을 갖춘 재직자에게는 적합할지 모르지만, 초급 수준의 기술을 가진 전공자나 비전공자에게는 채 3~4년을 버텨내기 어려운 소모성 인력으로 전락할 위험성이 도사리는 과정이라 할 수 있다. 우리가 수행하던 교육 과정도 다른 기관과 마찬가지로 문제 해결 방법을 제대로 가르치지 못하고, 흥미 위주의 도구 교육에 치우쳐져 있었다.

지난 1999년 마침내 우리 팀은 완전히 새로운 교육 체계를 만들기 시작했다. 새 팀을 이룬 연구개발진의 첫 번째 과제였다. 처음 개발을 시작한 당시에는 블루엿 인터내셔널, 플



라스틱 소프트웨어, 이스텡 소프트웨어, 아이팁스 등 나름대로 한 가지 소프트웨어 기술을 자랑하는 여러 개의 기업이 한데 어우러져 일을 추진했지만, 일을 해나가는 과정에서 우여곡절이 많았다. 결국 독립 기업이던 아이팁스는 동명정보기술원의 연구개발팀으로 변했고, 현재 교육 시스템 개발과 운영은 블루엣 인터내셔널과 동명정보기술원이 공동으로 책임지고 있다.

교육 시스템의 이름은 우리의 개발 철학을 살려, ‘이론을 실제로(Theory Into Practice)’라고 정했다. 우리는 응용 기

술을 열거식으로 주입하는 교육이 어떤 결과를 가져오는지 잘 알고 있었기 때문에, 이론을 직접 가르치지는 못하더라도 이론이 자연스럽게 녹아있는 기술 교육이 필요하다고 생각했다. 교육 과정을 만들고 시행하면서 가장 힘들었던 부분은 제각기 다른 특성을 가진 플랫폼 의존적인 기술을 쓰면서, 어떻게 하면 플랫폼에 의존하지 않도록 키워내느냐 하는 것이었다. 우리가 키워낸 기술자들이 오랜 기간 빠르게 성장하도록 바탕을 마련해주기 위해서는 가장 먼저 해결하고 넘어가야만 하는 과제였기 때문이다.

만화방이 아저씨들로 북적거리는 이유

올해 새로 개설한 과정의 교육생을 선발하기 위해 면접을 치르던 중에 있었던 일이다. 한 학생이 낸 지원서에 적힌 자격증 기재란에는 MCSE · SCJP · OCP · MCDBA 등 흔히 얘기하는 이 분야의 유명한 자격증이란 자격증은 다 적혀 있었다. 그리고 짧은 하지만 회사에서 실무 경험도 어느 정도 갖고 있는 학생이었다. 필자는 그 학생이 왜 우리 과정을 신청했는지 궁금해서 물어보았다. 그리곤 곧 씩씩하게 웃을 수밖에 없었다. 그 학생의 대답은 이러했다.

“저는 이 분야의 비전공자인데 친구들이 권해서 MCSE를 취득했습니다. 처음이라 어떻게 시작해야 할지 몰라서 집 근처의 작은 학원을 다녔는데, 꼭 대입 학원 다닐 때처럼 문제 풀고 답 듣고 하는 몇 달의 과정을 통해 쉽게 자격증을 딸 수 있었죠. 그 뒤로는 학원비가 부담스러워 후배들과 스터디 그룹을 결성해 각종 자격증에 도전했는데 그 때마다 흔히 ‘족보’라고 불리는 브레인 덤프의 도움을 얻어 잘 알지는 못했지만 어떻게 자격증을 딸 수 있었습니다. 작년에 졸업하자마자 자격증 덕분에 비전공자인데도 쉽게 취직을 할 수 있었는데요, 회사에 가보니 제 실력으로 할 수 있는 일이 별로 없더군요. 그래서 이번에는 제대로 공부를 해보자 결심하고, 회사를 그만두고 이 과정에 등록하게 됐습니다. 사실은 회사에서 일 못한다고 찼습니다(웃음)”

만일 그 학생이 처음 학원에 갔을 때 누군가 현 업계에서 자격증의 실체를 조금이라도 이해할 수 있게 소개해 줬더라면 2년이나 지난 이 순간 그 학생이 다시 교육 기관을 찾아지 않았을 수도 있었을 것이다. 어쩌면 지금 이 순간에도 빨리 자격증을 취득하고 싶어하는 사람들은 밤을 새워 족보와 답을 외우는 어리석은 짓을 하고 있을지 모른다. 그렇게 해서 취득한 자격증은 마치 전쟁터 근처에도 가보지 못한 사람이 집 근처 가게에서 구입해 붙인 훈장과도 같다. 처음에는 그 가짜 훈장 덕분에 사는 것이 편해질 지도 모른다. 그러나 결국 전쟁터에 가보지 못한 사람은 그 사실이 탄로날 수밖에 없다. 너도나도 돈으로 만든 훈장을 달고 다니면 누가 그 훈장을 보고 존경의 뜻을 표시하겠는가.

물론 자격증이 중요하지 않다거나, 별로 쓸모가 없기 때문에 취득하지 말라는 것은 절대 아니다. 오랜 기간 준비해 비싼 수업료를 내고 취득한 자격증이 정말 사회에서 그 가치를 인정받을 수 있도록 모두 힘을 합쳐야 한다는 뜻이다. 정도를 걷는다는 것이 분명 어려운 일이긴 하지만, 정도를 걷지 않았을 때 치러야 할 대가에 비하면 아무 것도 아닐 수 있다.

참고로 필자의 옛날 얘기를 하나 하자면, 대학 2학년 시절에 자료구조론이라는 과목을 수강한 적이 있다. ‘Fundamentals of Data Structures in Pascal’이란 원서를 교재로 사용했는데, 그때는 원서가 솔직히 부담스러웠다. 그 과목은 시험 문제가 바뀌지 않기로 유명했는데, 그래서인지 유난히 전년도 시험 문제를 구해 시험 준비를 하는 친구들이 많았다. 선배로부터 그 과목이 얼마나 중요한지를 귀에 못이 박히도록 들은 필자는 시간이 얼마나 걸리든 간에 그 책을 끝까지 탐독하기로 했다. 시험이 작년과 거의 비슷하게 출제되어 상대적으로 점수가 낮았던 필자는 B학점을 받았다. 하지만 그 이후부터 지금까지 그때 공부했던 내용은 큰 도움이 되고 있으며, 당시 필자의 선택을 조금도 후회하지 않는다. 오히려 4학년 인공지능 수업 시간에 당시 A+을 받았노라고 좋아하던 몇몇 선배가 개발한 소스 코드가 자료구조를 잘못 사용해 형편없이 구현됐다고 교수로부터 혼나는 것을 보고 그 때의 선택이 옳았음을 절감했다.

나이를 먹어가면서 ‘정의는 항상 이긴다’는 만화속 내용이 거짓말이라고 단정지었는데 요즘 들어 생각이 다시 바뀌고 있다. 시간이 지날수록 힘들었지만 제대로 하려고 노력한 일들이 좋은 결실을 맺게 되는 경험을 자주 하기 때문이다. 필자는 ‘정의가 항상 이긴다’는 만화책 속의 진리를 다시 한번 확인할 때마다 기본 좋게 만화방에서 하루를 보낸다. 요즘 만화방에 들러보면 옛날과는 달리 필자 또래의 아저씨들로 가득차 있는 이유가 그것 때문일지도 모르겠다.

처음부터 다시...

이런 목표로 연구하던 우리 팀이 주로 참고한 자료는 외국 대학에서 쓰는 소위 잘 만든 교재였다. 어느 정도 집작은 하고 있었지만 그 수준에 정말 놀라지 않을 수 없었다. 몇 권의 교재만으로도 우리 기술 교육 수준이 얼마나 뒤떨어져 있으며, 또한 얼마나 방향을 잘못 잡아가고 있는지 실감할 수 있었다. 예를 들어 어떤 학교의 교재는 우리와 다를 바 없이 자바를 쓰고 있었다. 그러나 그 내용은 우리가 흔히 보던 것과는 완전히 달랐다. 정말 프로그래밍을 가르치는 것이 무엇이고, 자바라는 언어만을 가르치는 것과는 얼마나 다른지 정확하게 보여주고 있는 자료였다. 같은 기술을 쓰면서도 이렇게 차이가 날 수 있다는 것이 놀라울 뿐이었다. 어쨌거나, 우리가 내린 결론은 ‘처음부터 아예 다시 가르쳐야 한다’는 것이었다. 가르치는 우리부터가 생각을 아예 바꿔야 할 수준이었기 때문에 밑바탕이 되는 교육을 더하지 않고서는 방법이 없다고 판단한 것이다. 최종 교과 과정은 다섯 단계가 되었고, 2년쯤이면 좋을 과정을 현실성이 없다는 이유로 1년으로 최적화했다.

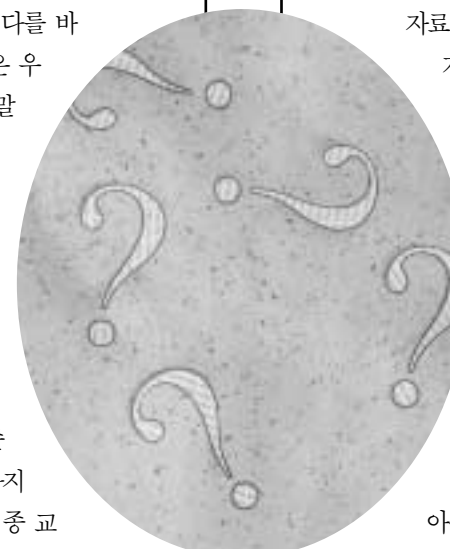
당연히 첫 번째 시도에서 많은 시행착오가 있었다. 특히 가르치는 사람에게는 힘든 한 해였다. 정해진 로드맵에 따라 시행하는 각 과정은 한정된 시간 안에 교육 효과를 내도록 최적화되어 있었는데, 이런 당초 의도가 오히려 좋지 않은 결과를 유발하는 원인이 됐다. 이런 과정은 일별 · 주별로 시행되는 과정간의 연계성이 생명이다. 그러나 외부에서 오는 강사는 아무리 토론과 공동 학습을 통해 노력해도 개발진의 의도를 알리도 없고 책임감도 덜하다. 더욱이 전체 과정을 모르고 특정 기술만 집중해서 가르치는 경우 학생들은 학습 동기를 잃어버리게 된다. 우리가 만든 과정이 스스로도 많은 학습을 통해 얻어낸 것이라는 점을 간과하고 있었던 것이다. 결국 그 모든 실수는 우리 팀원들 하나 하나의 부담이 되었다. 그리고 그 결과 개발진들은 짐과 여관을 점점 구분하기 힘들게 되었다.

몇 년 동안 우리는 여러 교육 기관에서 자바를 수개월간 학습하고 온 사람들을 많이 보아왔다. 때때로 3개월간 하루 4시간의 교육으로 프로젝트까지 마치고 왔다는 얘기를 들으면 걱정부터 앞선다. 만약 배운 게 잘못됐다면 새로 가르치는 것은 고사하고 알던 것부터 바로잡아야 하기 때문이다. 알다시피 새로운 것을 얻는 것보다 애매하게 알고 있던 것을 바로잡는 게 더 힘든 법이다. 결국 우리는 이런 저런 이유 때문에 스

스로 교재를 개발하고자 했다. 시중에서 우리 틀에 맞는 교재를 찾기도 힘들지만 외국어 서적의 부담까지 떠 안게 하지는 말자는 뜻에서 시도했던 일이었다. 좋은 시도였으나 고생은 이루 말할 수가 없었다. 특히 닷넷 과정에서 C# 교재는 거의 모험에 가까웠다. 더욱이 자바 과정과 완전히 동일한 예제를 만들어내야 했는데 언어의 의미마저 정확하게 알 수 있는 자료가 없었기 때문에 일일이 실험을 통해 결과를 기록하는 수밖에 없었다. 또, 언어의 스펙과 다른 결과를 내는 기능이 있을 때는 그게 버그인지, 기능을 아직 만들지 않아서인지 그 이유까지 규명해야 했다(이후 닷넷의 베타판이 다시 나왔을 때 우리는 다시 한번 야단법석을 떨어야 했다) 많은 교육 자료를 얻어낼 수 있었고, 연구진 자체 교육 성과도 높아져 전혀 성과가 없는 것은 아니었지만, 안타깝게도 연구팀의 고생은 고생으로 끝났다. 결국 연구팀은 약속을 지키지 못하고 그나마 잘 쓴 번역서 몇 권을 주는 것으로 아픈 마음을 접어야만 했다. 오랜 회한과 반성 끝에 연구팀은 아무래도 외서를 그대로 쓰는 게 낫다는 결정을 했다. 이 판단이 또 어떤 결과를 나올 지는 알 수 없지만, 최소한 지킬 수 없는 약속을 하는 것보다는 마음이 덜 아프기 때문이다. 교재 연구를 하는 중에 한 연구원이 하던 말이 생각난다. “아니~무슨 기초 과정 예제 하나를 만드는 게, J2EE 블루프린트 디자인 패턴 연구하는 것보다 더 어렵지(그러기에 응용보다는 사실 바탕을 제대로 아는 게 더 힘든 법이다)”

모든 개발자 과정에 ‘유닉스’를 넣게된 이유

옛날 한 교수님께서서는 컴퓨터가 없어 연습장에 코드를 작성하고, 한참 동안 줄을 서서 천공 카드를 뚫고 나서야 겨우 그 코드를 테스트할 수 있었던 시절의 프로그래밍 환경을 얘기 해주시면서, “어떻게 그렇게 공부한 자기보다 몇 배는 더 좋은 환경에 있는 네가 더 프로그래밍을 못할 수 있냐”며 심하게 야단을 쳤던 적이 있다. 공교롭게도 박사 과정에 다니고 있을 무렵 후배들에게 강의를 하면서 비슷한 말을 하고 있는 필자를 발견할 수 있었다. “PC란 개념조차 익숙지 않았던 시절에는 선배가 방 만들어 준다는 말에 유닉스 계정 하나를 발급해주는 지도 모르고, 아무런 의심도 없이 이 학교는 학생들에게 방도 주네 하면서 이사집을 싸들고 오는 사람들도 많았다. 그런데 너희는...”으로 시작하는 잔소리에 후배들도 내가





교수님으로부터 얘기를 들던 당시처럼 호랑이 담배피던 시절 얘기라며 웃었을지도 모르겠다.

지금 생각해보면 후배들의 기본 실력이 떨어지기 시작하는 시기가 유닉스 시스템이 점점 사라져가고 윈도우 환경이 득세하면서 편리하게 기술을 익힐 수 있게 된 시기와 묘하게도 일치한다. 물론 후배들의 실력이 떨어진 모든 이유가 단지 이 때문이라고는 볼 수 없다. 그보다는 갑자기 컴퓨팅의 응용 분야가 확대되고, 개발 생산성이 높아지면서 새로운 응용 기술들이 정신차릴 수 없을 정도로 쏟아져 나왔기 때문이라고 해야 옳다. 하지만 기술을 배운다는 것이 매일 매일 스스로 문제를 해결해가며 생활 속에서 축적해나가야 하는 과정이라면 지금의 학습 환경은 그렇게 도움을 주지 못하는 것이 사실이다. 유닉스 시스템은 단순한 상품 가치를 넘어서는 의미가 있다. 누군가도 그랬듯이 유닉스 시스템은 마치 조립되지 않은 스텔스기 부품들이 비행장에 널려 있는 것과 같은 환경이다. 이 환경에서 무언가를 제대로 작업하자면 사용자 스스로가 자기에 맞는 환경을 만들어 쓰지 않고는 버틸 수 없다. 물론 최근에는 여러 가지 유닉스 변형이 나와서 윈도우 OS 만큼 간단하게 설치하고 운영할 수 있지만 기본 서비스를 제외한 다른 환경을 얻고자 하면 그 작업이 그리 만만하지 않다. 이런 환경에서 문서를 만들고, 프로그래밍을 하면서 얻게 되는 교육 효과는 무시할 만한 것이 못된다. 가장 큰 부분은 역시 매일 매일 스스로 문제 해결을 통해 작업 환경을 구축해가면서 기술자로서의 기본 소양을 자연스럽게 익힐 수 있다는 점이다. 물론 윈도우 환경에서 이런 교육 효과를 얻을 수 없다고 말한다면, 다소 억지 주장일 수 있다. 그러나, 오랜 역사에서 살아 남은 기술은 분명 세련되고 편리한 환경이 주

지 못하는 교육 효과가 있다. 연구팀은 닷넷 플랫폼의 소프트웨어 기술을 배우고자 하는 사람이라도 유닉스 시스템을 쓰면서 더 많은 것을 얻어갈 수 있다고 믿는다. 이것이 우리가 각 과정에서 쓰는 플랫폼에 상관없이 유닉스 시스템을 기본으로 가르치는 이유다.

프로젝트 과정에서 발생하는 문제

교육 과정 마지막에는 4개월 이상의 장기 프로젝트 과목이 배정되어 있다. 물론 프로젝트 과정은 어느 과정에서도 빠질 수 없는 부분이지만, 수년 전만 하더라도 강사들에게 가장 편한 시간이었고 스스로를 재충전하는 유일한 휴식시간으로 활용됐다. 당시에 교육받던 학생들은 스스로 구현할 프로젝트를 결정하고, 자체 그룹 스터디나 세미나를 통해 대부분의 문제를 해결했기 때문에 가끔씩 기술의 향방이나 구현 기술의 문제점에 대해 조언을 해주는 정도만이 강사가 할 일이었다. 하지만 이와 같은 프로젝트는 시간이 지나면서 많은 문제점을 드러냈다. 일단 시장의 상황이 바뀌었다. 당시만 하더라도 국내 정보기술 시장에서 요구하는 인력은 대단한 기술을 가진 경력자가 아니었다. 그래서 학생 스스로의 노력만으로도 시장이 요구하는 제품을 만들어내는 데는 무리없는 교육이었고, 프로젝트 교육 과정을 바꿔야 한다는 위기 의식이 없었다. 그러나 많은 인력이 배출되면서 고급 기술자에 대한 요구가 높아져만 갔다. 더욱이 지금의 인터넷 비즈니스 환경은 단순한 정보 전달의 수단이었던 초기 인터넷 환경을 벗어나서 한 기업의 정보 기술력을 모두 집결해놓은 대규모 소프트웨어 시스템으로 이뤄져 있다. 이와 같은 환경에서는 한두 가지 요소 기술을 쓸만하게 학습했다고 해서 소프트웨어 기술자로 대우받지 못한다. 여러 기술을 종합해 섞어 써야 하는 것은 물론이고 계속 변화하는 기술도 빠르게 익혀가며 기존의 컴퓨팅 환경에 곧바로 응용할 수 있는 능력이 필요하다. 이런 요구를 만족하는 기술자를 키워내기 위해서 가장 중요한 교육은 역시 프로젝트 과정이다. 특히 시스템의 규모가 점점 커지면서 개발자간의 기술적인 교류 및 공동 개발 작업 체계를 익히는 것이 무엇보다 중요한 교육 과제가 됐다.

우리는 학생들이 과정에서 배운 것을 충분히 활용할 만큼 수준있는 프로젝트의 주제를 스스로 결정하지 못한다는 사실을 잘 알고 있었다. 또한 팀 별로 프로젝트 주제가 다르다 보니 각 팀간의 기술 교류가 원활하게 이뤄지지 못했다. 같은 일을 하는 구성원간의 대화는 기술 그 자체보다 중요하다. 서로 거리낌없이 서로가 가진 기술의 장단점을 지적해주고 어떻게 하면 더 나은 방법으로 작업을 진행할 것인지 끝없이 토

론하며 함께 문제점을 발견해 가는 분위기가 프로젝트 교육의 성패를 좌우한다고 할 수 있다.

외국의 어떤 교수는 자신이 한 교육이 학생들에게 효과가 있었는지를 확인하기 위해 커피 마시는 시간에 어떤 대화를 나누는지 항상 체크했다고 한다. 만족스럽게 교육이 진행된 날은 커피 마시는 시간에 학생들끼리 그날 수업에 있었던 얘기를 하면서 자연스럽게 관심이 이어지지만, 그렇지 않은 날은 스포츠 이야기, 여자 친구 이야기와 같은 잡담만 하게 된다는 사실을 알았기 때문이라고 한다. 그 당시의 학생들이 쉬는 시간에 하는 얘기는 마치 수업이 제대로 진행되지 않은 날 나누는 얘기와 같았다. 공통된 관심사를 갖지 못하다 보니, 프로젝트 진행과 관계된 얘기를 서로 나눌 수 없었던 것이다. 또한 가장 큰 관심사가 수료 후의 취업이다 보니, 과정을 마치기 전인데도 취업을 위해 면접을 보거나 미리 취업이 되어 나가는 학생들로 인해 프로젝트 팀이 해체되는 경우도 많았다. 심할 경우 7명의 프로젝트 팀원 중에서 한 명만 남아도 저지 프로젝트를 진행할 수 없는 경우도 있었다. 무엇보다 가장 큰 문제는 학생들 중에 실무를 경험한 사람이 거의 없어서, 어떤 식으로 문제를 정의하고 분석해야 하는지 그리고 그 문제 해결을 위해 어떻게 여러 명의 동료들이 협력해 작업해야 하는지에 대한 방법을 전혀 알지 못했다. 그러다 보니 팀원들간에 잦은 불화로 프로젝트를 망치는 경우가 많았고, 문제 분석에 너무 많은 일정을 잡아 급기야 설계를 다 해놓고도 완성을 못하는 경우도 있었다.



이런 저런 문제점으로 연구팀은 지난해에 프로젝트를 학생 스스로가 아닌 교육 개발팀에서 통합 관리하는 방식을 취했다. 각 팀들과 일일이 날마다 면담하고, 잘못된 것을 지적하며 어느 해보다도 힘들게 프로젝트 과정 교육을 진행했다. 또 프로젝트 과정 중에 특강 및 세미나 형태로 계속해서 추가 기술 교육을 시행했다. 이 때문에 연구팀은 스스로를 기다들을 수 있는 중요한 시간을 희생한 것이지만 그만큼 가치가 충분하다고 생각했다. 그러나 그 생각은 크게 잘못된 것이었다. 오히려 예전보다 학생들의 프로젝트 성과는 기대에 미치지 못했다. 물론 몇몇 우수한 팀은 연구팀의 집중적인 관리로 충분히 좋은 결과를 이끌어 낸 것도 사실이지만, 나머지가 예년 수준에 미치지 못했다는 데 문제가 있었다. 전체적인 성과로 볼 때 연구팀이 기획한 프로젝트는 노력에도 불구하고 실패가 입증된 셈이다.

결국 교육은 선생이 아닌 학생이 부지런해져야 성공한다는 사실을 알게 됐다. 올해 연구팀은 이런 실패를 거울삼아 두 가지 방법의 장단점을 살릴 수 있는 새로운 프로젝트 교육 진행 방법을 놓고 고민하고 있다. 새로운 방법의 핵심은 프로젝트 과정에 앞서서 기초 교육을 실시할 때부터 정해진 팀간의 자발적 협력 관계를 유도하는 데 있다. 이러한 협력 관계가 장기간에 걸쳐 구축되면 프로젝트 교육에 들어가서 기술 개발 자체에 더욱 집중할 수 있을 것으로 예상한다.

"나는 우리 분야에서 이론과 실재를 떼어놓는 것은 부자연스러울 뿐 아니라 매우 해로운 것이라고 생각해 왔다. 전산분야의 실용적인 작업들은(하드웨어와 소프트웨어를 막론하고) 무슨 문제를 어떻게 다루고 있는지에 대한 근본적인 이해가 없이 부지런하게만 건설된 것들이기 때문에 위태롭고 어설픈 성과들이 대부분이다. 이론 분야에서 이루어놓은 성과들도 진짜 컴퓨팅 현실로부터 너무나 초연해 있었던 까닭에 현실의 열매를 키워내는 비옥한 토양이 되지 못한 경우가 대부분이다. 우리의 '프로그램 연구 그룹'의 주요 목표는 이러한 이론과 실재의 불통이 발생할 수 없는 공부 풍토를 건설하는 것이다(출처: <http://ropas.kaist.ac.kr/~kwang/quote/strachey.html>)".

실무 교육만이 능사 아니다

연구팀은 고급 기술을 어떻게 하면 좀 더 쉽게 받아들이도록 할 수 있을까를 놓고 항상 고민하지 않을 수 없었다. 이를테면 '자바의 기본 기술을 가르치면서 자연스럽게 좋은 설계란 어떤 것인가를 알게 하자' 라든가 '뻔하게 만들어 보는 예제라도 디자인 패턴이 녹아있게 한다' 든지 등... 하지만 문제는 이 분야 전공자와 비 전공자의 수강생이 섞여 있다는 점이다. 전공 비전공을 갈라야 옳다고 생각하는 게 상식이었지만

현실은 그렇지 않다. 일단 모집에 있어서 의지보다 교육 시장의 현실이 앞선다. 그리고 정보 기술이란 것이 응용 기술인 바에야 기술에 접목할 수 있는 다른 분야의 배경지식이 있다면 나중에 가서는 학습 성과가 오히려 올라 갈 가능성이 있기 때문에 뭐가 좋다고 단언하기도 힘들다. 알다시피 기업용 소프트웨어를 만드는 과정에서 겪게 되는 난제는 기술보다 업무 분석에 있고, 어차피 IT 교육기관에서 교육은 그 모든 문

제를 해결해주지 못한다. 그러므로 교육으로 할 수 있는 최고의 방법은 기본기를 제대로 갖추 수 있도록 이끌어 주는 것이라 할 수 있다. 더구나 4년간 관련 전공을 마친 학생들에게서 오히려 더 많은 문제를 발견하는 경우도 있다. 요사이 ‘실무’라는 바람이 불고, 우리도 ‘실무’ 기술을 가르친다는 걸로 홍보를 해왔지만 대학의 실무 교육은 오히려 IT 교육기관에서의 교육에 방해가 될 가능성이 있다. 여러 가지 유행하는 기

‘김쌤’의 C# 교육

연구팀에는 김쌤(김선생님의 경상도 사투리)으로 불리는 세 사람이 있다. 그 중 한 사람이 닷넷 교육을 진행하고 있는 데 작년 한 해 가장 처절한 경험을 한 사람이 아닐까 싶다. 지금부터 소개하는 얘기는 그 김쌤이 C#.In The Large 교육을 하면서 겪은 경험담이다.

가르치는 사람이라면 누구나 그렇듯이 나도 학생들이 감동받기를 원한다. 감동은 공감대가 이뤄졌다는 것을 뜻하고, 공감대를 이루는 교육은 효과가 높다. 그래서인지 때로 나는 그 감동을 강요하는 경향이 있고, 이 때문에 겪는 우여곡절이 많다. 작년 한해 나는 좌절과 보람 사이를 여러 번 넘나들어야 했다.

처음 수업에서 유닉스 기본 수업을 하며 vi와 같은 에디터가 정규표현이란 이론이 가져다주는 힘을 어떻게 잘 활용했는지 깨닫게 하고 싶었다. 그저 기능으로 익숙해지는 것만으로는 직성이 풀리지 않았기 때문이다. 당연히 이럴 때 나는 감동을 강요하는 버릇이 있다.

감동적이지 않습니까?
...
이걸 해보고 감동받지 않으면 인간이 아닐 진대.... 감동의 눈물이 막 솟음을
치지 않습니까?
...
감동의 물결이 파도를 치고 유닉스를 설계한 사람들의 숨소리가 느껴지지
않습니까?
...

그 다음 썰렁해진 분위기는 자세히 얘기하지 않아도 짐작할 수 있을 것이다. 이렇게 감동을 수차례나 유도했던 부분에서는 전혀 반응이 없더니, 윈도우 2000의 터미널 서비스 웹 클라이언트에서는 감동받는 것을 보고, 나는 스스로를 되돌아보지 않을 수 없었다. 좋은 선생이란 보이지 않는 것을 보이게 하는 것이라 배웠기 때문이다.

C#을 가르칠 때의 일이다. 디자인 패턴을 목이 터져라 설명하고 Visitor Pattern이 어떻게 문제를 해결하는지 감동 받기를 원했다. 디자인 패턴과 서브 타이핑의 상관성을 아무리 설명해도 돌아오는 반응

은 없었다. 더 황당한 것은 감동을 받은 척(?)하는 학생이다. 내용을 이해하고 감동받았다기 보다, 분위기에 휩싸여 열렬결에 그런 것뿐인데, 그 학생의 한 마디는 가히 살인적이다.

“선생님, 그래서 Visitor가 어떤 클래스예요?”

5주간 총 270여 시간의 교육이 끝나고 개인별 프로젝트를 진행할 시간이 됐다. 통신 게임을 만들어 보겠다는 학생들이 대부분이고, 한 두 명은 스프레드시트를 만들어 보고자 했다. 이 때쯤 나는 ‘감동의 효과’를 더 이상 기대하지 않고 있었다. ‘그저 돌아가는 프로그램만 만들어 주세요.’ 이게 내 솔직한 심정이었다. 통신 게임을 만드는 학생들의 진도는 잘 나갔다. 반면에 스프레드시트를 만드는 학생들은 감을 잡는데 만 2주 가까운 시간이 걸렸다. 그럭저럭 흥내를 내기 시작할 즈음 한 학생이 찾아와 질문했다. 자꾸 코드가 길어지지만 하는데 이래도 되는 것인지 모르겠다는 것이다. 보여준 코드에는 is 연산자가 가득했다. 순간 밀려오는 충격. 나는 다시 한번 마지막 ‘감동의 효과’를 기대하면서 힘을 냈다.

“이미 배웠던 것인데, 이렇게 구성해야 확장해도 문제가 없지. 이럴 때 쓰라고 Visitor Pattern을 설명했는데...”

사흘이 지난 뒤 한 통의 메일이 왔다.

“선생님이 강조하시던 감동의 실체를 알았습니다. 말씀하신대로 구조를 고쳐서 그대로 돌아가는 것이 너무 신기했습니다. 알고 있어서 쓰면서 감동을 느끼지 않으면 말짱 헛수고라는 것을 알았습니다. 고맙습니다.”

그러나 다음 한 줄도 빼놓지 않았다.

P.S : 하지만 감동을 강요하지 마십시오.

술을 제대로 알고 오지 못할 바에야, 대학에서 컴퓨터 과학의 기초를 튼튼하게 잡고와야 옳은데 이도 저도 아니어서 기술에 대한 선입관과 잘못된 습관을 그대로 들고 오는 경우가 있기 때문이다. 이런 수강생들의 경우, 설득하는 일이 교육 그 자체보다도 힘들다.

선입관이 주는 피해

유닉스 보안 과정이 그 전형이다. 관련 전공자가 거의 없었는데도 ‘해커’에 대한 헐리우드 영화식 발상은 학생들의 마음을 덮고 있었다. 우리가 만든 ‘유닉스 보안 과정’의 원래 명칭은 ‘유닉스 보안 솔루션 개발자 과정’이다. 누구의 실수인지 과정 이름이 잘못되는 어처구니 없는 실수를 한 것인데, 그 후유증이 생각보다 심각했다. 학생 대부분은 간단하게 말해서 ‘시스템을 뚫는 방법을 배우는 것’이 보안 기술 교육이라고 생각했고, 시스템 엔지니어가 되고 싶어 입학한 사람들이 많았다. 그러나 연구팀이 만든 교육은 소프트웨어 개발자 교육이고, 이전 10개월의 교육에 보안 이론과 기술을 엮어서 1년 과정으로 만들었을 뿐이었기 때문에, 연구팀은 학생들만 큼이나 혼란스러워 할 수밖에 없었다. 일차적인 실수는 학생들이 연구팀이 만든 교육과정의 세부 기술 교육 과정을 모두 꼼꼼히 훑어보고 왔을 거라 일방적으로 믿어버린 데 있다. 과정 설명회에서 충분히 설명한 걸로는 모자란다. 내용이야 어 떠하든 사람들은 겉모습에 끌리는 게 인지상정이다. 일부 학생의 경우 제목에 매료되어 온 사람도 있고 소프트웨어 기술은 간단히 훑어보고 마칠 거라 생각하고 온 사람들도 제법 있었다.

무엇이 이런 결과를 가져오고 말았을까? 설령 시스템 엔지니어 과정이었다고 하더라도 ‘뚫고 막는 기술’이 보안 기술이라는 등식은 다소 황당한 발상이다. 이런 빛나간 선입관이 교육에 주는 악 영향은 생각보다 심각했다. 현재 이 과정은 연구팀의 노력으로 제 자리를 잡았다. 그러나 그 동안의 갈등으로 생긴 피해를 어떻게 메우느냐가 중요한 난제로 남아있다.

시장의 흐름과 바른 교육 사이에서

정보 기술 시장은 유행을 타고 흘러간다(실제 산업에서 쓰는지 안 쓰는지는 아랑곳없다). 언제부터가 앞서지 않으면 쓰러진다는 강박관념으로 모든 게 앞으로 치닫고만 있는 것처럼 보인다. 그러나 기술의 본질 그 자체는 별로 변한 게 없기 때문에, 교육 기관은 언제나 겉질과 본질 사이에서 방황하기 쉽다.

우리는 지난해부터 닷넷 기술을 가르친다. 현재로서 닷넷은 산업에 뿌리내린 기술이라고 보기 어렵다. 그러나 교육기관 입장에서 곧 유행할 닷넷을 내세우면 얻는게 많기 때문에, 수익을 생각해야 하는 IT 교육기관에서는 그 유혹을 버리기가 어렵다. 이런 문제로 국내 분위기를 타지 못한 분야의 기술을 가르치는 것은 모험에 가깝다. 개설하고 싶어도 배우러 오는 사람이 적기 때문이다. 모바일, XML, 보안, J2EE, 닷넷, 리눅스 등의 글귀가 들어가지 않으면 침탄 기술이 아닌 것으로 받아들여지기 쉬운 것이다. 따라서 장기 교육에 투자할 가치가 없다고 생각하는 풍토가 만연해 있다. 시장을 따라 기술이 흘러가는 것은 어쩔 수 없다. 그러나 교육은 좀 혼란스러운 시장에서 한 걸음 떨어져 있어야 하지 않을까?

IT 경쟁력의 밑거름

올해 연구팀은 작년 한해동안 겪었던 여러 가지 실패를 밑천 삼아 세부 운영 방안을 추가한 새로운 교육을 다시 한번 시도하고 있다. ‘Theory Into Practice’라는 교육 시스템을 운영하면서 그동안 이론에 해당하는 교과 과정과 내용에만 신경을 썼지, 새로운 교육 과정에 맞는 완전히 다른 운영 방법이 필요하다는 사실을 깨달았다. 따라서 올해의 연구는 운영 기법의 체계를 마련하는 데 집중하고 있다. 이번에도 어쩌면 지난해와 같이 크고 작은 좌절을 맛보게 될지도 모른다. 하지만 올바른 교육을 하고자 하는 마음이 변하지 않는 한, 언젠가 이 교육 시스템이 조만간 완성될 수 있을 것이라고 믿는다. 그리고 이런 연구가 결실을 맺어서 우리 정보 기술 수준을 끌어올리는 밑거름이 되리라 믿는다. **황**

정리 : 강경수 elegy@sbmedia.co.kr

