

소프트웨어 교육 강화를 위한 졸업프로젝트 수행 방안

금오공과대학교 임은기*

1. 서론

소프트웨어 교육 분야의 선진 외국 대학에서는 졸업프로젝트의 수행이 정착 단계에 이르렀으며, 최근에는 졸업프로젝트의 효율적인 운영을 위한 다양한 실험 결과를 활발하게 발표하고 있다[1][2]. 국내에서도 졸업프로젝트(Capstone Project)는 컴퓨터-소프트웨어 교육에 있어 필수 교과목으로 인식되고 있으며, 소프트웨어 교육 강화를 위한 최근의 연구보고서는 졸업프로젝트를 반드시 이수하도록 권고하고 있다[3].¹⁾

실용적인 소프트웨어 교육을 강화하기 위하여 졸업프로젝트의 수행이 매우 중요함에도 불구하고 국내의 많은 대학에서는 졸업프로젝트의 운영이 활성화되지 못하고 있는 실정이며, 이는 졸업프로젝트를 실제로 운영하기 위한 구체적인 방법에 대하여 적절한 해결책을 찾지 못하고 있기 때문으로 판단된다.

본 논문에서는 졸업프로젝트를 수행하고 있는 국내의 대학의 운영 사례에 대한 조사 연구를 바탕으로 하여 졸업프로젝트의 수행과 관련하여 중요한 쟁점사항을 도출하고, 도출된 항목에 대하여 국내 대학의 실정에 맞는 대안을 제시하고자 한다.

2. 졸업프로젝트 운영과 관련된 쟁점사항 도출

2.1 국내외 대학의 졸업프로젝트 운영 현황

졸업프로젝트의 실천 방안을 찾기 위하여 국내의 14개 대학과 외국의 9개 대학에 대하여 현재 운영중인 소프트웨어 실습교육 프로그램의 사례를 조사한 결과, 외국 대학에서는 졸업프로젝트의 운영이 정착

단계에 도달하였으며, 인턴십 프로그램의 운영을 강화하는 추세에 있다. 특히, Nice Sophia Antipolis 대학의 산학일체형 인턴십 프로그램이나 Waterloo 대학의 Co-op 프로그램은 대학의 프로젝트 수행 능력을 산업체에서 인정한 결과로서 국내 대학의 발전 방향에 대한 중요한 참조가 될 것으로 판단된다.

국내의 경우에는 졸업프로젝트를 도입하는 단계로서 졸업프로젝트를 체계적으로 운영해 본 경험이 절대적으로 부족하여 Ad-hoc한 방법으로 프로젝트를 수행하고 있는 실정이며, 프로젝트를 수행하기에 적합하지 않은 교과과정이나 프로젝트를 지도할 수 있는 숙련된 지도요원의 부족 등으로 인하여 졸업프로젝트를 정착시키기가 쉽지 않은 상황이다. 인턴십 프로그램에 대해서는 산업체의 인식이 매우 부정적이어서 대학에서 인턴십 프로그램을 도입하기가 단기적으로는 어려운 것으로 추정되며, 졸업프로젝트가 충분히 정착된 이후에 인턴십 프로그램의 도입을 검토하는 것이 타당할 것으로 판단된다.

2.2 졸업프로젝트 운영 관련 쟁점사항

졸업프로젝트를 운영하는 국내외 대학에 대한 사례 분석을 통하여 졸업프로젝트를 위한 별도의 교과목 설치여부, 프로젝트에서 수행할 과제의 선정 방법, 수업 진행 방법, 평가에 활용하는 자료, 그리고 프로젝트 수행 결과의 관리 방법 등이 대학별로 매우 다양함을 발견할 수 있다. 이러한 분석 결과와 국내 대학이 처한 현실을 고려할 때 졸업프로젝트의 운영상 중요한 쟁점사항은 다음과 같이 요약할 수 있다.

- 졸업프로젝트 과목의 목표 설정
- 졸업프로젝트를 위한 과목 설치 방법
- 졸업프로젝트 과목의 세부 운영 방안

* 중신회원

- 프로젝트 지도요원의 확보 방안
- 프로젝트 수행과제 관리 체계
- 졸업프로젝트의 소프트웨어공학적 수행 방안

2.3 졸업프로젝트 운영을 위한 선결 과제

졸업프로젝트를 통하여 성취하고자 하는 교육 목표를 달성하기 위해서는 2가지의 중요한 문제가 해결되어야 한다. 첫째, 프로젝트를 수행할 학생들의 프로그래밍 기술이 일정 수준에 도달하여야 한다. C, C++, Java, Visual Basic 등의 기본적인 프로그래밍 언어를 사용할 수 있는 능력을 구비해야 하며, 소단위의 문제를 분석하여 해결할 수 있어야 하고, 구현 언어를 위한 프로그래밍 환경에 숙달되어야 한다. 이러한 프로그래밍 기술을 습득하게 하기 위해서는 교과과정에 포함된 적절한 과목들에서 프로그래밍 과제를 부과하고 과제 검사를 철저하게 할 필요가 있다.

둘째, 학생들이 소프트웨어공학에 관한 지식을 습득하여야 한다. 기본적으로 소프트웨어를 개발하는 프로세스에 대하여 이해하고 있어야 하며, 구조적 방법론[4] 및 객체지향 방법론[5]에 기초한 단계별 산출물을 작성할 수 있는 능력을 갖추어야 하고, 팀 구성원들 사이에 의사 소통을 원활하게 할 수 있어야 한다. 이를 위해서는 최소한 2학기의 소프트웨어공학 강의를 필요하며, 여건이 허락되는 경우에는 Term Project를 병행함으로써 더욱 높은 효과를 달성할 수 있다.

3. 졸업프로젝트 운영 방안

졸업프로젝트를 도입하여 운영하고자 하는 대학에서는 사전에 철저한 준비를 해야 하며, 특히 2장에서 제시한 쟁점사항에 관해서는 교수진의 합의를 통하여 기본적인 운영 방침을 결정하여야 한다. 본 장에서는 졸업프로젝트의 도입을 준비하는 과정에서 참조할 수 있도록 실천 가능한 운영 방안을 제시한다.

3.1 졸업프로젝트의 목표 설정

졸업프로젝트를 도입하기에 앞서 학생들에게 졸업프로젝트의 수행을 통하여 성취하고자 하는 교육 목표를 명확히 할 필요가 있다. 일반적으로 설정 가능한 목표는 개별적인 교과목에서 이론 강의를 통하여 습득한 전공지식을 하나로 통합하여 체계화시키

고, 소프트웨어를 개발하는 전 과정에 익숙해지도록 하여 각 단계에서 생산되는 산출물을 작성하고 적용 가능한 개발 도구를 직접 사용해 보는 기회를 제공하며, 팀 구성원으로서 역할을 담당하여 임무를 수행하는 경험을 쌓도록 하는 것이다.

어떤 목표를 설정하느냐에 따라 졸업프로젝트 과목의 설계는 달라져야 하므로 교육 대상인 학생들의 기술 수준을 우선적으로 고려하여 무리한 목표를 설정하지 않도록 해야 하며 일단 설정한 교육 목표는 확실하게 성취하는 것이 중요하다. 또한 프로젝트의 수행 환경, 개발 도구의 확보 상황, 프로젝트 지도요원의 확보, 수행할 과제의 발굴 방법 등에 대하여 충분한 고려가 있어야 한다.

3.2 졸업프로젝트를 위한 과목 설치 방법

졸업프로젝트를 수행시키는 방법은 대학의 상황에 따라 다양한 선택이 가능하겠으나 국내의 실정을 감안하면 3가지의 가능한 대안을 들 수 있다. 첫째, 별도의 교과목을 개설하지 않고 졸업 작품을 필수로 요구함으로써 프로젝트를 수행시키는 방법으로 실행이 간편한 반면, 학생을 지도할 수 있는 기회가 없어 기대하는 효과의 달성 가능성이 희박하다. 둘째, 기존의 교과목에서 프로젝트를 병행시키는 방법으로 교과목을 담당하는 교수 개인의 노력으로 시행할 수 있으며 부분적인 효과는 달성할 수 있으나, 학생들의 수업 부담이 과다한 문제점이 있다.

셋째, 졸업프로젝트를 위한 별도의 교과목을 개설하여 프로젝트를 지도하는 방법으로 궁극적으로 대부분의 대학이 가야할 방향으로 판단되나, 개설할 교과목의 학점 및 수업시간, 개설 학년 및 학기, 이수구분, 성적처리 방법에 관해서는 사전에 충분히 검토하여 교수 전원의 합의가 이루어져야 한다.

3.3 졸업프로젝트 과목의 세부 운영 방안

졸업프로젝트를 위한 교과목은 담당교수에 따라 다양한 방법으로 운영할 수 있으나, 졸업프로젝트를 처음으로 도입하는 대학에서 쉽게 적용할 수 있는 절차로서 다음과 같은 방안을 제안한다.

- 1단계 : 과목 담당교수가 수업의 골격을 구상하고 진행 과정에서 필요한 서식을 준비
- 2단계 : 수업 진행 방법과 프로젝트 수행 절차를

학생들에게 설명

- 3단계 : 프로젝트를 수행할 팀을 구성
- 4단계 : 팀별로 수행할 과제를 선정하고 지도요원을 배정
- 5단계 : 프로젝트를 예정된 일정에 맞추어 수행하고 단계별 산출물에 대하여 중간 평가
- 6단계 : 프로젝트 수행 결과에 대한 최종 평가
- 7단계 : 완료된 프로젝트에 대한 산출물을 정리하고, 관리체계에 따라 사후 처리

3.3.1 수업 준비

졸업프로젝트 과목을 담당하는 교수는 수업을 위하여 다음 사항을 준비한다.

- 프로젝트 지도요원의 역량을 고려하여 개발 대상 소프트웨어의 분류 또는 Domain을 지정하거나 졸업프로젝트 관련 사이트를 활용하여 학생들이 수행할 과제의 범위를 개략적으로 결정
- 프로젝트팀의 최대 및 최소인원을 결정하고, 예상되는 팀의 수에 따라 필요한 프로젝트 지도요원의 소요를 산정
- 프로젝트를 수행할 때 적용할 개발방법론을 선택
- 선택한 개발방법론에 맞게 단계별 일정 계획을 수립
- 진행 단계별로 생산할 산출물의 종류 및 양식과 각각의 산출물에 대한 평가 방법을 결정
- 학생들이 프로젝트를 수행하는 과정에서 참조할 프로젝트 수행계획서를 작성
- 프로젝트를 수행할 때 학생들이 사용할 개발 환경 및 개발 도구를 준비

3.3.2 프로젝트 수행 절차 설명

학기가 시작되면 담당교수는 프로젝트 수행 절차 및 수업 진행 방법에 관한 다음의 사항을 상세하게 설명하여, 학생들로 하여금 프로젝트의 수행과 수업의 진행을 적절히 조화시켜 졸업프로젝트를 통한 교육 효과를 극대화하도록 지도한다.

- 프로젝트를 수행할 때 적용할 개발방법론
- 개발방법론의 진행 단계별로 작성할 산출물
- 각각의 산출물에 대한 평가방법
- 각 단계의 마감 시한이 명시된 프로젝트 일정 계획
- 강의, 실습 및 토의, 발표 및 평가의 진행 방법 및 일정
- 프로젝트팀 구성원의 최대 인원 및 최소 인원

- 팀을 구성할 때 지켜야할 제약 조건
- 팀별로 수행할 과제의 선정 방법 및 절차

3.3.3 프로젝트팀 구성

프로젝트팀의 인원수는 선정되는 과제의 규모에 따라 다소 차이가 있을 수 있으나 3-5명이 적절하며, 팀별로 구성원 수의 편차가 심하지 않도록 조정한다. 팀의 구성은 학생의 자원에 맡기는 것이 바람직하나, 팀을 이끌어 나갈 수 있는 능력을 갖춘 학생이 팀별로 1명 이상 반드시 포함되도록 지도하여야 하며, 학생들의 관심 분야에 따라 팀을 구성하여 운영하는 것도 상당한 효과를 거둘 수 있다.

3.3.4 수행과제 선정 및 지도요원 배정

수행과제의 선정은 졸업프로젝트의 성패를 좌우하는 매우 중요한 문제이며, 프로젝트를 수행할 학생의 수준과 프로젝트를 진행할 지도요원의 역량을 반드시 고려해야 한다. 대학의 여건에 따라 다음에 제시한 방법이나 이들을 적절하게 조합한 다양한 방법을 적용할 수 있다.

- 팀별로 관심 있는 분야에서 자유롭게 채택하는 방법
- 개발 대상 소프트웨어의 분류나 Domain을 지정하는 방법
- 졸업프로젝트 관련 사이트를 활용하여 담당교수가 과제 목록을 제시하는 방법

첫 번째 방법은 학생들의 창의성을 최대한 발휘할 수 있도록 보장할 수 있으나, 지도요원에게 엄청난 부담을 줄 수 있으므로 뛰어난 지도요원을 확보한 경우가 아니면 적절한 방법으로 보기 어렵다. 두 번째 방법은 학생들이 창의성을 발휘할 수 있는 여지는 많으나, 학생들이 상당한 수준의 능력을 갖추고 있을 때 적용할 수 있어 졸업프로젝트를 도입한 초기 단계에서 적용하기에는 무리가 있으며, 팀별로 과제의 규모가 비슷하도록 조정하는 작업이 매우 어렵고, 과제를 선정하지 못하는 팀을 위한 준비가 필요하다. 세 번째 방법은 학생들의 수준에 적합한 과제 목록을 준비하는 일이 어려우나, 프로젝트를 무난하게 진행할 수 있는 장점이 있으며, 졸업프로젝트를 도입한 초기에 적용하기가 용이하다.

팀별로 수행할 과제가 선정되면 팀을 담당할 지도요원을 배정한다. 지도요원은 학과 교수, 별도의 전

담 지도요원, 산업체에서 초빙한 전문가, 박사과정 학생 등 대학의 사정에 따라 다양한 방법으로 구성되, 프로젝트팀의 수를 감안하여 충분히 확보해야 한다. 지도요원을 배정할 때 담당할 팀에서 수행할 과제와 지도요원의 전공 분야를 맞추어 효과적인 지도가 가능하도록 하며, 1명의 지도요원이 가능한 한 3팀을 초과하지 않도록 해야 한다.

3.3.5 프로젝트 수행 및 수업 진행

프로젝트의 진행에 따라 수행하는 활동을 요약하면 다음의 표 1과 같다.

표 1 프로젝트 진행에 따른 활동

활동	수행자	수행 일정
강의	담당교수	강의계획 일정
실습 및 토의	팀구성원, 지도요원	별도 일정
발표 및 평가	담당교수, 지도요원	프로젝트 수행 일정

강의는 학기초에 수립한 강의 일정계획에 따라 담당교수가 진행하며, 주로 프로젝트 수행 단계별로 수행해야 할 활동과 발표할 산출물의 작성 방법에 대하여 설명한다. 실습 및 토의는 팀 구성원들이 자율적으로 수행하되 프로젝트 수행 일정에 따른 마감 시간을 준수하도록 하며, 필요시 지도요원의 지도를 받는다. 지도요원은 담당하는 팀이 실습 및 토의를 진행하는 과정에서 기술적인 문제에 대하여 지도하며, 최소한 주 1회 이상 일정한 시간에 담당하는 팀과 만나는 것이 바람직하다. 발표 및 평가는 프로젝트 수행 일정에 따라 지정된 수업 시간에 이루어지며, 팀별로 프로젝트의 진행 상황과 작성한 산출물을 발표하고, 담당교수와 지도요원은 발표 내용을 평가하여 잘못된 부분은 수정한 후에 다시 발표하도록 지도한다.

3.3.6 최종 평가

평가는 진행 단계별로 생산되는 산출물에 대한 중간 평가와 완성된 최종 시스템을 대상으로 하는 최종 평가로 구성된다. 중간 평가는 프로젝트 진행 단계별로 발표하는 산출물에 대하여 담당교수와 지도요원이 평가하여 그 결과를 성적 처리에 반영하며, 완성된 최종 시스템에 대해서는 담당교수를 포함하여 최소한 3명 이상의 교수가 심사하여 채점하고, 평균한 점수를 성적 처리에 반영하는 것이 좋다.

최종 평가는 계획된 일정에 따라 엄격하게 진행하여 졸업프로젝트의 효과를 극대화시켜야 하며, 졸업프로젝트가 졸업 필수 요건인 경우에는 심사 받는 기회를 여러 번 부여하여 탈락하는 팀을 극소화할 필요가 있다.

단계별 산출물과 최종 시스템을 평가할 때 적용할 평가 기준은 학기초에 학생들에게 충분히 설명함으로써 평가의 공정성을 확보할 필요가 있으며, 다음에 열거한 것은 최종 시스템에 대한 평가서 작성시 포함해야 할 대표적인 항목이다.

- 시스템의 기능이 사용자의 요구를 잘 반영하고 있는가?
- 시스템의 기능은 제대로 작동하는가?
- 시스템을 실무에 얼마나 효과적으로 적용할 수 있는가?
- 소프트웨어 개발 기술을 얼마나 많이 활용하였는가?
- 소프트웨어 개발 도구의 사용은 적절하였는가?
- 팀 구성원 사이에 협력은 잘 되었는가?
- 프로젝트 진행 과정에서의 문서화는 적절한가?
- 단계별로 생산한 산출물을 발표할 때 그 내용을 얼마나 잘 전달하는가?

최종 평가시 특별히 고려해야 할 것은 free rider에 대한 대책이다. 팀을 구성하여 프로젝트를 수행하는 경우에 free rider의 발생은 불가피하며, 이들에 대하여 적절한 대책을 수립하지 못하면 열심히 노력한 구성원이 불이익을 받을 수밖에 없다. 대학별로 다양한 대책을 선택할 수 있으나, 구성원 사이의 상호평가를 통하여 만족할 만한 결과를 얻을 수 있었다.

3.3.7 사후 관리

최종심사를 통과한 시스템은 작품전시회 등의 행사를 통하여 전시함으로써 학생들의 성취감을 고양시킬 수 있으며, 우수한 작품에 대해서는 어떤 형식으로든지 포상하여 학생들의 사기를 진작시킬 필요가 있다. 산업체의 관련 인사를 초빙하여 학생들과의 만남을 주선함으로써 취업 기회를 제공한다면 더욱 바람직할 것이다.

제출된 단계별 산출물 및 최종 시스템은 일정한 지침에 따라 관리하여 차후의 프로젝트 과제 선정에 도움이 되도록 하고, 졸업프로젝트 관련 사이트에도 등록하여 졸업프로젝트를 활성화하는데 기여하도록 할 필요가 있다.

3.4 프로젝트 지도요원의 확보 방안

졸업프로젝트를 진행하는 과정에서 프로젝트팀을 직접 지도하는 프로젝트 지도요원은 다음의 요건을 구비하여야 하며, 졸업프로젝트의 교육목표를 달성하기 위해서 우수한 지도요원을 확보가 필수적이다.

- 중규모 이상의 소프트웨어 개발 프로젝트를 수행한 경력
- 소프트웨어 개발 프로젝트에서 프로젝트 관리자의 역할을 담당한 경험
- 다양한 소프트웨어공학적 방법론에 숙달
- 개발 대상 소프트웨어에 따라 적절한 방법론을 선택할 수 있는 능력을 소유
- 다양한 영역에 대한 Domain Knowledge를 갖추고 있을 것
- 소프트웨어 개발에 필요한 전공 분야에 대하여 일정 수준 이상의 지식을 구비
- 학생을 지도한 경험이 있으면 더욱 바람직함

프로젝트 지도요원을 확보할 수 있는 방안으로는 아래와 같이 4가지를 들 수 있다.

- 산업체에서 소프트웨어 개발 경험이 풍부한 인력을 겸임교수로 임용하거나 산학협력 프로그램을 통하여 일정기간 근무하도록 하는 방안
- 대학 교수를 산업체에 파견하여 소프트웨어 개발 경험을 쌓도록 하거나 별도의 프로젝트 지도요원 양성 프로그램을 통하여 훈련시키는 방안
- 대학에서 별도의 예산을 확보하여 프로젝트 지도를 담당할 전임자를 채용하는 방안
- 학과 내에서 프로젝트 수행 경험이 있는 박사과정 학생을 활용하는 방안

위에서 열거한 방법은 대학이 처한 여건에 따라 가능할 수도 있으나, 대부분의 대학에서는 학과 소속 교수가 지도요원의 역할을 담당해야 할 가능성이 매우 높다. 따라서 졸업프로젝트를 도입하여 운영하기 위해서는 교수 전원의 합의가 대단히 중요하다.

3.5 프로젝트 수행과제 관리체계

학생이 수행할 과제를 쉽게 발굴할 수 있도록 하고, 담당 교수의 수업준비 부담을 경감시키기 위하여 완성된 과제를 체계적으로 관리할 필요가 있다. 수행

과제를 효과적으로 관리하기 위해서는 대학별로 독자적인 관리체계를 수립해야 하고, 대학간에 정보공유를 위한 졸업프로젝트 포털 사이트를 운영함으로써 시너지 효과를 유발시킬 수 있다.

졸업프로젝트를 운영하는 대학에서는 수행한 모든 과제를 표준화된 형식에 따라 제목, 요약, 단계별 산출물 및 최종 시스템을 등록하고, 등록된 과제는 프로젝트를 수행할 때 자유로이 참고할 수 있도록 공개하도록 한다. 학과의 책임자는 수행과제의 관리책임자를 임명하고, 과제의 등록 및 조회 절차를 제정하여 준수하도록 한다.

졸업프로젝트 포털 사이트는 참여 대학이 공동으로 운영하여 정보공유의 장으로 활용하고, 대학별로 수행한 과제 및 수행 방법을 등록하도록 유도하고, 산업체 및 전문 교육기관의 참여를 적극 권장하여 국가적인 차원에서 소프트웨어 인력양성에 기여하도록 할 필요가 있다.

4. 졸업프로젝트의 소프트웨어공학적 수행 방안

4.1 소프트웨어공학적 프로젝트 수행의 필요성

졸업프로젝트의 수행에 소프트웨어공학을 적용해야 하는 것은 산업체에서 소프트웨어공학에 숙달된 전문기술자를 필요로 하고 있으며, 앞으로도 이러한 추세는 가속화될 것으로 예상되기 때문이다. 개발 대상 소프트웨어는 점차 대형화되는 추세에 있어, 문제를 분석하여 시스템의 기능, 범위, 제약사항 등의 정확한 요구사항을 도출할 수 있는 기술과 요구사항으로부터 소프트웨어 시스템의 전체적인 구조와 입출력, 인터페이스를 설계하는 기술을 갖추고 있는 전문 기술자에 대한 수요 역시 증가하고 있다. 산업체에서 필요로 하는 인력을 양성하기 위해서 단순한 프로그래밍 skill 중심의 실습교육만으로는 부족하며, 팀 중심의 소프트웨어공학 실습을 통하여 다양한 소프트웨어 개발 경험을 쌓도록 할 필요가 있다.

4.2 소프트웨어공학 실습교육의 목표 설정

소프트웨어공학 실습교육의 목표는 소프트웨어를 개발하는 과정에서 필요한 지식과 기술을 숙달하여 어떤 과제가 주어지든 예측된 비용으로 정해진 기간

안에 좋은 품질의 소프트웨어를 생산해낼 수 있는 고급 기술자를 양성함에 있으며, 이러한 교육 목표를 달성하기 위하여 수행하는 제반 활동은 다음의 사항을 고려하여 추진되어야 한다.

- 소프트웨어공학에서 중요시하는 3가지 주제인 개발 프로세스, 개발 방법론, 그리고 개발 도구에 대하여 실습할 수 있는 틀을 제공해야 함
- 소프트웨어가 점차 대형화하는 추세에 부응하기 위해서 개인이 수행하는 프로젝트가 아니라, 팀 중심의 중형 이상의 프로젝트를 경험할 수 있도록 함
- 소프트웨어공학에서 다루는 방대한 지식과 기술을 실습교육의 틀 안에 모두 포함시킬 수는 없으므로, 완전한 능력을 갖춘 전문가를 양성하는 교육 프로그램을 추구해서는 안되며, 장차 전문가로 성장할 인력을 육성한다는 장기적 시각에서 실습교육을 추진해야 함
- 이론 강의를 위주로 하던 기존의 교육 방식에서 과감히 탈피하여 모든 학생이 소프트웨어공학의 이론을 직접 수행해 볼 수 있는 방향으로 실습교육을 운영해야 함
- 소프트웨어의 개발 단계별로 적용 가능한 자동화 도구의 사용은 적극적으로 권장해야 함
- 학과 교수의 전공 분야와 대학이 처한 현실적인 여건을 충분히 고려하여 실천 가능한 교육 프로그램을 준비해야 함

4.3 소프트웨어공학에 기초한 졸업프로젝트 수행 방안

소프트웨어공학적 프로젝트의 수행을 위하여 가장 먼저 해야 할 것은 실습교육에서 다룰 대상의 선정이다. 다음에 열거한 분야가 대표적인 실습 대상이나, 어떤 것을 선택하여 학생들에게 가르칠 것인가는 실습교육을 계획하고 추진함에 있어 중요하고도 어려운 문제이다. 대학에 따라 서로 다른 결론에 도달할 수 있으나, 열거한 분야 모두를 포함하기보다는 대학의 실정에 따라 적절한 몇 개의 분야를 선택하여 시작하고 졸업프로젝트 과목의 운영이 정착되는 정도에 따라 점차적으로 분야를 확대해 나갈 것을 권장한다. 초기에는 요구수집 및 분석, 설계, 구현, 테스트를 대상 분야로 선정하는 것이 무리가 없을 것으로 판단된다.

- 모델링 : UML 기반 도구 사용, 코드 자동 생성
- 요구 공학
- 시스템 설계
- 코딩 및 시험
- 유지보수, 재공학, 역공학
- 프로젝트 관리
- 소프트웨어 품질
- 소프트웨어 프로세스 개선

실습교육의 대상 분야가 선정되면 실습교육의 발전 단계를 설정한다. 소프트웨어공학 실습교육을 실시할 때 처음부터 완벽한 교육 프로그램을 운영하는 것은 거의 불가능하며, 다음과 같은 실습교육의 단계에 따라 점진적으로 발전시켜 나가도록 노력할 필요가 있다.

- 1단계 : 순차적 개발 실습을 하는 초급 단계로서 폭포수 모델 방식으로 진행하되, 필요시 반복적 또는 점증적 모델 방식으로 진화할 수 있음
- 2단계 : 유지보수 실습을 수행하는 단계로서 다른 팀에서 개발한 소프트웨어에 포함된 오류를 수정하거나 새로운 요구를 추가하여 발전시켜 나가는 실습을 함
- 3단계 : 역공학 단계로서 대상 시스템을 분석하여 시스템의 구성요소와 그들 사이의 관계를 파악하여 시스템을 다른 방법으로 표현하거나 보다 높은 수준의 추상을 도출해내는 실습을 수행함

실습 단계를 설정한 후에는 각 단계별로 적용할 실습교육 방법을 구상하여 프로젝트의 수행을 위한 구체적인 방안을 준비한다. 다음은 초급 과정인 1단계의 순차적 개발 실습교육을 위한 수행 방안이며, 2단계 및 3단계의 실습교육 방법에 대해서는 추후의 연구과제로 남긴다.

- 실습교육은 문서양식 등을 포함하는 문제의 기술부터 시작하여 요구사항 분석, 구조 설계, 상세 설계, 코딩 및 시험으로 진행하며, 최종 시스템과 개발 문서를 산출함으로써 종료
- 각 단계별로 강의 25%, 실습 및 토의 50%, 발표 및 평가 25% 정도의 비율로 구성
- 강의에서는 프로젝트를 진행하는 동안 학생이 수행해야 할 활동과 단계별로 생산해야 할 산출물

에 대하여 설명

- 실습 및 토의는 수업시간 외에 팀별로 수행하는 것을 원칙으로 하며, 프로젝트 일정 계획에 명시한 마감시한을 준수하도록 함
- 발표 및 평가는 팀별로 생산한 산출물을 발표하며, 담당교수 및 지도요원이 평가하고 지도함
- 프로젝트가 완료되면 개발 과정에서 생산한 모든 산출물을 정리하여 최종적으로 발표하고 최종 시스템은 테스트 데이터를 준비하여 시연하도록 함

5. 결론

본 연구에서는 실무 중심적 전문인력을 양성하기 위한 소프트웨어 교육에 있어 핵심적인 위치를 차지하는 졸업프로젝트를 활성화하기 위하여 국내외 대학의 운영 사례를 분석하고, 분석 결과를 바탕으로 졸업프로젝트의 운영상 중요한 쟁점사항을 도출하였으며, 국내 대학에 적용 가능한 졸업프로젝트의 실천적 운영방안과 프로젝트의 소프트웨어공학적 수행방안을 제시하였다.

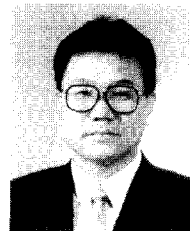
그러나, 국내 대학의 소프트웨어 관련 학과에서 졸업프로젝트를 활성화하지 못하는 것은 학과 내부, 대학 본부, 산업체 및 정부로부터의 저해 요인이 존재하기 때문이다. 이러한 요인을 극복하고 졸업프로젝트를 활성화함으로써 소프트웨어 교육을 강화하기 위해서는 교과과정의 획기적인 개편, 교수평가제도 및 학과평가제도의 개선, 프로젝트 지도요원의 정책적 양성, 정부의 각종 지원사업의 지속적 시행 등 수많은 과제들이 해결되어야 한다. 이와 같이 열악한 상황에도 불구하고 소프트웨어 교육 강화를 위한 대

학 교수들의 확고한 의지만 있다면 졸업프로젝트를 정착시키는 것은 가능하리라 확신한다.

참고문헌

- [1] B. Olsson, M. Berndtsson, B. Lundell, and J. Hansson, "Running Research-Oriented Final Year Projects for CS and IS Students," SIGCSE, Feb. 2003, pp. 19-23.
- [2] David A. Umphress, T. Dean Hendrix, and James H. Cross, "Software Process in the Classroom : TheCapstone Project Experience," IEEE Software, Sep./Oct. 2002, pp. 78-85.
- [3] 대학의 컴퓨터-소프트웨어 교육 강화 방안, 한국소프트웨어진흥원, 2001년 11월.
- [4] 최은만, 소프트웨어공학론, 사이텍미디어, 2001년.
- [5] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson, Object-Oriented Modeling and Design, Prentice Hall, 1991.

임은기



1977 서울대학교 수학과(학사)
 1988 한국과학기술원 전산학과(석사)
 1993 한국과학기술원 전산학과 박사과정 수료
 1989~현재 금오공과대학교 컴퓨터공학부 교수
 관심분야 : 데이터베이스 설계, 소프트웨어 개발 프로세스, 컴퓨터교육
 E mail : eklim@se.kumoh.ac.kr