



## 21세기 IT 산업을 위한 소프트웨어공학 인력 양성

한국과학기술원 배두환\*

### 1. 서 론

소프트웨어산업은 21세기의 유망 산업중의 하나일 뿐만 아니라, 국가의 경쟁력 제고 차원에서도 중요한 역할이 기대되며 향후 모든 산업 분야에 미치는 영향력이 매우 큰 잠재력이 매우 큰 산업이다. 특히 우리처럼 천연 자원이 없고 대신에 많은 인적 자원을 가지고 있는 나라에서는 더욱이 그 가치가 중요하게 느껴진다.

소프트웨어 개발의 3대 요소로 자원(Resources), 기술(Technology)과 프로세스(Process)를 꼽을 수 있는데, 이 중에서도 가장 중요한 요소는 역시 인적 자원이라 할 수 있다. 우리나라의 경우 비록 지난 10년 간 세계 소프트웨어 산업에서 차지하는 비중이 1% 근처에 머물고 있지만 아직도 무한한 잠재력을 지니고 있다고 생각된다. 그 이유는 엄청난 교육열과 함께 매년 많은 IT 인력이 배출되고 있기 때문이다.

국내의 소프트웨어 개발 실태를 살펴보면 선진국에 비해서 생산성은 약 3배 정도 높은 수준이며 품질 수준은 결함율이 적게는 2배에서 많게는 6배까지 정도라고 한다. 생산성, 품질 이외에도 국내의 소프트웨어 기술 축적의 정도를 살펴보면 대부분의 소프트웨어 개발이 프로그래밍 기술에 의존하고 있는 실정이며 고도의 기술과 경험을 요구하는 대규모의 소프트웨어 개발을 위해서 꼭 필요한 분석 및 설계 전문가들이 턱없이 부족하고 특히 우수한 프로젝트 관리 능력의 부재로 인하여 부가 가치가 높은 소프트웨어의 개발을 힘겨워 하고 있는 것이 우리의 현실이다.

지금까지의 IT 교육은 저변 확대를 위한 인력 양성에 많은 투자가 이루어졌다고 할 수 있으며 앞으로는 고급 IT 인력의 양성을 위한 투자를 지속적으로

하여야 한다. 대학에서도 고급 소프트웨어공학 인력 육성을 위한 소프트웨어공학 교육의 활성화가 요구되며 이렇게 배출된 인력이 장기적으로 다양한 산업 분야에서 고급 IT 인력으로 발전할 수 있는 교육 프로그램을 마련하여 고급 소프트웨어공학 인력의 육성 기틀을 마련하는 것이 중요하다고 생각된다.

본 고에서는 21세기 IT산업 발전을 위해 필요한 고급 소프트웨어공학 인력 양성에 대해서 살펴보고자 한다. 2장에서는 소프트웨어공학의 정의 및 간단한 소개와 인력 양성을 위한 새로운 조류를 알아본다. 3장에서는 소프트웨어공학 전문 인력으로서 요구되는 능력과 소프트웨어공학 전문 지식과 기술 분류를 소개하고, 4장에서는 국외 대학들의 소프트웨어공학 전문 학사, 석사 프로그램들을 간단히 분석 소개한다. 5장에서는 국내에 소프트웨어공학 학사 학위 프로그램을 도입 시에 우선적으로 필요하다고 생각되는 교과 과정에 대한 제안을 하고 6장에서 결론을 맺는다.

### 2. 소프트웨어공학 개요 및 인력 양성 프로그램 현황

#### 2.1 소프트웨어공학 정의

소프트웨어공학은 IEEE Computer Society의 정의에 의하면 다음과 같다 "The application of a systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is, the application of engineering to software" 즉, 소프트웨어의 개발, 운영 및 유지 보수에 체계적이고, 훈련이 잘되고 정량적인 접근 방법을 적용하는 것을 의미하며, 이는 곧 소프트웨어에

\* 중신회원

공학적으로 접근하는 것을 말한다. 1960년 말에 NATO 컨퍼런스에서 소프트웨어 위기(Software Crisis)라는 용어가 처음 언급된 후에 소프트웨어 개발에 공학적인 접근 방법의 필요성이 제기 되면서 소프트웨어공학이라는 용어가 처음 소개되었으며 이후로 약간씩의 수정이 가해졌지만 소프트웨어공학의 정의는 대체로 유사한 내용을 담고 있다.

소프트웨어공학 교육을 통하여 양성코자하는 소프트웨어 엔지니어와, 기존의 프로그래머의 차이를 살펴보면 프로그래머는 주로 개인이 혼자서 감당할 수 있는 크기의 소프트웨어 또는 그 일부를 프로그래밍 언어, 자료 구조, 알고리즘 등의 전산 지식에 기반하여 개발하는 개발자를 일컫는다면, 소프트웨어 엔지니어는 여러 개발자들이 한 팀이 되어 개발하는 소프트웨어의 요구 사항을 분석하고, 설계 및 시험, 프로젝트 관리 등을 핵심 역량으로 하는 전문 인력들을 가리킨다(소프트웨어 엔지니어에는 프로그래머의 역할도 포함된다고 할 수 있다).

## 2.2 소프트웨어공학 인력 양성을 위한 국내외의 새로운 조류

현재 개발 실무에서 전문적으로 분석 및 설계, 시험을 전문으로 할 수 있는 개발자들은 그 수가 매우 부족하며, 많은 경우에는 이러한 전문 인력의 필요성조차 인식하지 못하고 있는 것이 우리의 현실이다. 하지만 소프트웨어공학 전문 인력에 대한 요구가 국내에서도 점차 높아져 가고 있으며 다양한 인력 양성 프로그램이 현재 추진 또는 계획 단계에 있다. 소프트웨어공학 인력 양성 관련하여 세계 각 국에서도 그 필요성의 인식이 점차 확산되어 가고 있으며 관련된 새로운 다음의 두 조류를 소개한다.

- 소프트웨어 공학 전문 교육 프로그램 개설
- 소프트웨어 엔지니어 인증(Certification)

소프트웨어공학 전문 교육 프로그램의 경우 국외를 중심으로 많은 대학들이 벌써 학사/석사 또는 박사 과정을 개설하여 운영하고 있다. 한 편, 이를 지원하기 위해 IEEE Computer Society에서는(초기에는 ACM과 공동으로 추진하다 현재는 ACM은 참여하고 있지 않음) SoftWare Engineering Body of Knowledge(SWEBOK)를 미국 및 유럽의 학계, 연구소 및 산업체의 전문가 의견을 종합하여 작성 중에

있다[1]. 국내의 경우 몇몇 대학에서 소프트웨어공학 과를 개설하여 운영하기도 하였으며 또 현재 개설 중인 대학도 있으나 그 교과 과정의 내용은 소프트웨어공학과와 전문 인력 양성과는 다소 거리가 있다고 생각된다. 향후 2002년부터 정보통신부 산하의 한국정보통신대학원대학교에서 소프트웨어공학 전문 석사 과정을 개설할 준비를 하고 있다. 또한 정보통신부에서는 2000년부터 소프트웨어공학 전문 인력을 양성하기 위하여 미국의 카네기멜론 대학의 소프트웨어공학전문석사과정에 연간 40명의 학계, 연구소 및 산업체 인력을 파견 교육시키고 있다.

소프트웨어 엔지니어 인증 프로그램인 IEEE Certified Software Engineering Professional(CSEP)은 자격증 제도는 아니지만 소프트웨어 엔지니어로서의 역할을 적절히 수행하는데 필요하다고 생각하는 기본적인 지식의 습득 정도를 확인하는 인증 프로그램이다[2]. 2001년에 미국, 캐나다, 브라질, 헝가리, 일본, 중국, 인도, 아일랜드 등에서 실시할 계획이며, 시험 과목은 다음과 같이 구성되어 있다.

- 소프트웨어 형상관리, 구현, 설계, 시험, 유지보수, 품질 ;
- 소프트웨어공학 도구, 방법, 프로세스와 관리 ;
- 소프트웨어 요구 공학 ;
- professionalism 및 engineering economics

또한 다음과 같은 조건을 만족해야 시험에 응시할 수 있는 자격이 주어진다.

- 학사 학위 이상 또는 동등한 자로서 최근 4년 간 2년 이상의 소프트웨어공학 실무 경험자
- 시험 분야 중 적어도 6개 분야 이상에서 9000시간 이상의 경력자

인증 제도의 사후 관리로 시험 합격 후 3년까지만 인증이 유효하며 매 3년마다 재교육을 통하여 인증 유효 기간을 연장할 수 있게끔 되어 있다. 그 외에 캐나다, 호주 등지에서도 소프트웨어 전문 인력을 양성하기 위하여 대학을 중심으로 인증 프로그램을 진행하고 있다.

## 3. 소프트웨어공학 전문 인력의 역할 및 지식

소프트웨어 전문 인력의 역할은 소프트웨어 개발 및 유지보수 활동에서 생산성 및 품질을 확보하는 것

이다. 과거의 주먹구구식의 소프트웨어 개발로는 무한 경쟁시대에서 경쟁력 있는 제품을 적시에 만들고 또 적절한 유지 보수 활동을 지원하기 매우 어려운 실정이다. 특히 전 산업 분야에서 소프트웨어의 중요성이 부각되고 소프트웨어의 비중이 점차 커져가고 있어 소프트웨어 개발 및 유지 보수 활동에 필요한 지식과 기술을 보유한 전문 인력의 필요성이 요청된다.

### 3.1 소프트웨어공학 전문 인력으로서 요구되는 능력

전문 소프트웨어 엔지니어로서 요구되는 역할은 소프트웨어 개발 전반에 걸쳐 다양한 능력이 요구되지만 이들 중에서 프로그래머의 역할을 제외한 요구 능력을 살펴보면 다음과 같다.

- 의사 소통 능력: 고객 및 개발자들과의 의사 교환을 원활히 할 수 있는 능력
- 요구 사항 분석: 고객의 애매 모호한 요구 사항을 명확히 파악하여 정확한 명세를 도출할 수 있는 능력
- 설계 능력: 다양한 설계 기법을 활용하여 소프트웨어 architecture, 상세 설계 명세를 도출할 수 있는 능력
- 시험 능력: 소프트웨어 개발 프로젝트에서 산출되는 각종 산출물의 정적인 시험 및 동적인 시험을 수행할 수 있는 능력
- 프로젝트 관리 능력: 프로젝트의 매니저로서 프로젝트 기획, 자원 및 일정 관리, 품질보증 및 프로세스 관리 및 향상 능력

### 3.2 소프트웨어공학 기술 및 지식

소프트웨어공학 관련 기술 및 지식은 다양한 분류가 가능하겠지만 SWEBOK의 분류를 간략히 소개하기로 한다. SWEBOK에서는 소프트웨어공학 전체를 10개 기술 분야로 대 분류하고 각 기술별로 필요한 상세 지식을 다음과 같이 정의하고 있다[1].

1. 소프트웨어 요구 사항
  - 요구 공학 프로세스
  - 요구 추출
  - 요구 분석
  - 요구 명세
  - 요구 검증

- 요구 관리
- 2. 소프트웨어 설계
  - 소프트웨어 설계 기본 개념
  - 소프트웨어 설계 주요 이슈
  - 소프트웨어 구조
  - 소프트웨어 설계 품질 분석 및 평가
  - 소프트웨어 설계 표기법
  - 소프트웨어 설계 전략 및 기법
- 3. 소프트웨어 구현
  - 복잡도 감소
  - 다양성 예측
  - 검증을 위한 구조
  - 표준의 사용
  - 언어학적 구현 기법
  - 정형 구현 기법
  - 시각적 구현 기법
- 4. 소프트웨어 시험
  - 시험의 기본 개념 및 정의
  - 시험 단계(level)
  - 시험 기법
  - 시험 관련 측정
  - 시험 프로세스의 관리
- 5. 소프트웨어 유지 보수
  - 기본 개념
  - 유지보수 프로세스
  - 유지보수의 주요 이슈
  - 유지보수 기법
- 6. 소프트웨어형상관리
  - 소프트웨어형상관리 프로세스
  - 소프트웨어형상 추출
  - 소프트웨어형상 컨트롤
  - 소프트웨어형상 상태 관리
  - 소프트웨어형상 감사
  - 소프트웨어 배포 관리 및 배달
- 7. 소프트웨어공학 관리
  - 조직 관리
  - 프로세스/프로젝트 관리
  - 소프트웨어공학 측정
- 8. 소프트웨어공학 프로세스
  - 소프트웨어공학 프로세스 개념
  - 프로세스 인프라
  - 프로세스 측정
  - 프로세스 정의

- 정량적인 프로세스 분석
- 프로세스 구현 및 수정
- 9 소프트웨어공학 도구 및 방법
  - 각 단계 별 소프트웨어 도구
  - 방법 :경험적 방법, 시제품 제작 방법, 기타 방법 이슈
- 10.소프트웨어 품질
  - 소프트웨어 품질 개념
  - 정의 및 품질 기획
  - 정적인 품질 검증 기법
  - 기타 기법 지원
  - 시험 품질
  - 결함 발견 기법
  - 소프트웨어 품질 보증을 위한 측정

#### 4. 국외 대학의 소프트웨어공학 관련 학위 프로그램 소개

앞서 언급한 바와 같이 국외의 소프트웨어 산업 선진국들의 경우 다양한 소프트웨어공학 관련 학위 과정을 개설하고 있다. 이 둘 중에서 대표적인 석사 학위 및 학사 학위 과정 각각 2개씩을 간략히 살펴본다.

##### 4.1 카네기 멜론 대학의 소프트웨어공학 석사 학위 과정

카네기 멜론 대학의 소프트웨어공학 석사 학위 과정은 첨단 소프트웨어공학 전문 지식을 교육하는 대표적인 과정으로서 우리에게도 정보통신부의 연수 프로그램을 통하여 비교적 잘 알려져 있는 프로그램이다[3].

###### 4.1.1 학과 구성 및 교육 목표

전산학부(School of Computer Science)와 소프트웨어 공학 연구소(Software Engineering Institute)에서 공동 운영한다. 과정의 교육 목표는 첨단 소프트웨어공학 전공 전문가 양성에 있다.

###### 4.1.2 특징 및 배경

최소한 2년 이상의 실무 경력자들에게 입학 허가가 주어지며 전체 취득 학점 중에서 30% 전공 필수, 40% 소프트웨어 개발 스튜디오 및 30% 전공 선택 과목들로 구성되어 있다. 다른 대학의 유사 교육 과정과 비교하여 매우 선진화된 소프트웨어공학 교육

과정으로 고급 소프트웨어공학 인력 양성에 초점이 맞추어져 있다고 할 수 있으며 다음의 네 가지 주제를 중심으로 교육 과정이 이루어져 있다.

- Engineering design and system architecture: Software architecture, design trade-offs, tools/environment, exploiting domain knowledge
- Technical management and evolution of large, long-lived software: design for reuse, risk analysis, configuration management, product validation
- Mathematical modeling, abstraction and reasoning: formal methods, software system analysis, testing, reasoning about system design
- Human resource management: project planning, risk assessment, cost estimation, process improvements

여러 학기에 걸쳐 팀 단위로 진행되는 소프트웨어 개발 스튜디오를 전체 학점의 40% 가량 할당하고 있다. 이 팀 프로젝트는 학교 밖의 사용자를 대상으로 소프트웨어를 만들며 학교에서 배운 여러 가지 지식을 실제 적용하고 그 결과를 평가받는 형식으로 진행되고 있다.

###### 4.1.3 과정 구성

기본 필수 교과 과정은 다음과 같으며, 다양한 선택 과목을 수강할 수 있다.

분야	교과과목	주요 교과내용
필수	Models of Software Systems	소프트웨어 아키텍처 및 모델링의 원리 및 기법
	Methods of S/W Development	소프트웨어 개발 기법
	Managing S/W Development	소프트웨어 개발을 관리 전반에 걸친 주제 소개 및 프로젝트 관리 기법
	Analysis of S/W Artifacts	소프트웨어 및 관련 산출물들의 분석
	Architectures of S/W Systems	소프트웨어 시스템의 아키텍처 모델링 및 분석 기법 소개

##### 4.2 Monmouth 대학의 소프트웨어공학 석사 과정

#### 4.2.1 학부 구성 및 교육 목표

미국 대학들 중에서 소프트웨어공학과를 전산학과로부터 독립적으로 신설하여 소프트웨어공학 학사 및 석사 과정을 개설한 대학 중의 하나이며, 여기서는 소프트웨어공학 석사 학위 과정에 대한 커리큘럼을 조사하였다[4].

교육 목표는 소프트웨어 개발 팀의 우수한 일원이 되는데 필요한 기술과 지식을 교육시키는 것이다.

#### 4.2.2 특징 및 배경

대부분의 수업들이 직장인을 위하여 야간에 진행되고 있다. 전체 이수 학점은 36학점이며 이 중 15학점의 필수 과목과 15학점의 선택 과목 및 졸업 요구 조건으로 Capstone requirement라는 6학점의 프로젝트 또는 논문 연구 중에서 택일하도록 되어 있다.

프로젝트의 경우 3-5명의 팀 중심의 소프트웨어 개발 실무를 다루고 있으며 각종 CASE도구를 이용하여 소프트웨어 개발, 시험 및 프로젝트 관리, 개발 비용 산정, 위험 관리 행사 관리 등등 다양한 실무 관련 기술을 습득 실습하는 과목이다. 한편 일정 수준의 전산학 지식이 부족한 학생들을 위하여 다음과 같은 준비 교과목(Preparatory Courses)을 제공하고 있다.

- 프로그램 개발
- 알고리즘
- 운영 체제
- 소프트웨어공학의 수학적 기초
- 소프트웨어공학 원리

#### 4.2.3 과정의 구성

기본 필수 과정은 다음과 같다.

분야	교과과목	주요 교과내용
필수	소프트웨어공학 프로세스	소프트웨어 개발 프로세스에 관한 제반 지식과 기법을 소개: CMM, ISO9000, PSP 등등
	소프트웨어 시스템 설계	소프트웨어 설계 원리 및 기법: 설계 원리, 객체 지향 기법 가능 중심 기법 등등
	소프트웨어 정형 기법	소프트웨어 개발에 사용되는 정형 기법 소개: 정형 명세 언어, 정형 검증 기법
	소프트웨어 시스템 요구 사항	소프트웨어 개발의 초기 단계인 요구 사항 도출 및 분석 기법 소개
	소프트웨어 시험 및 품질	소프트웨어 시험 기법 및 품질 관련 이슈를 소개

### 4.3 University of Texas, Dallas의 소프트웨어공학 학사 과정

#### 4.3.1 학과 구성 및 교육 목표

학부의 명칭이 Erik Jonsson School of Engineering and Computer Science로 되어 있으며 전산학 프로그래밍과는 별도로 Electrical Engineering, Telecommunication 및 Software Engineering으로 각각 특화된 프로그램을 운영하고 있다[5]. 교육 목표는 공학 프로그램의 경우 탄탄한 수학 및 과학 분야에 기반을 두고 복잡한 문제를 해결할 수 있는 능력의 배양과 새로운 기술의 적용 능력을 강조하고 있다.

#### 4.3.2 특징 및 배경

소프트웨어공학 학위 과정의 교과목들 중에서 상당 부분이 기존의 전산학 학위 과정의 교과목을 활용하고 있으며, 순수한 소프트웨어공학 과목은 다음에 열거한 5 과목 정도이다.

- 소프트웨어공학의 수학적 기초
- 소프트웨어 요구 사항
- 소프트웨어 아키텍처
- 소프트웨어 시험, 검증 및 품질 보증
- 소프트웨어공학 프로젝트

이 외에 기존의 전산학 과목들 중에 소프트웨어공학과 관련되는 과목으로는 Professional Technical Communication이라는 과목으로 실무에서 필요한 의사 소통에 관한 내용을 교양 및 전공 필수 과목으로 강조하고 있다.

#### 4.3.3 과정의 구성

전체 커리큘럼은 123학점으로 구성되어 있으며, 42학점의 교양 필수, 60학점의 전공 필수(기초 과목 20학점, 전공 필수 31학점, 전공 선택 9학점) 및 선택 21학점으로 나뉘어져 있으며 이들의 구성은 다음과 같다.

- 전공 기초 커리큘럼 목록(20학점)

분야	교과과목	주요 교과내용
전공 기초	Calculus 1	기존의 과목과 유사함
	Calculus 11	"
	Linear Algebra	"
	Computer Science 1	"
	Discrete Math. 1	"
	Discrete Math. 11	"
	Computer Science 11	"
Mathematical Foundations of Software Engineering	소프트웨어공학에 필요한 수학적 기초 지식	

- 전공 필수 커리큘럼 목록(31학점)

분야	교과과목	주요 교과내용
전공 필수	Probability and Statistics in CS	기존의 과목과 유사함
	Algorithm Analysis and Data Structure	"
	Software Engineering	"
	Professional Technical Communication	엔지니어들 사이의 의사소통 및 엔지니어와 고객 사이의 의사소통
	Computer Architecture	기존의 과목과 유사함
	Computer Architecture Lab.	"
	Operating Systems	"
	Software Requirements	소프트웨어 요구 사항 추출, 분석 기법
	Software Architecture	소프트웨어 설계의 핵심, 소프트웨어 구조 기술 및 분석
	Software Testing, V&V and Quality Assurance	소프트웨어 시험 원리, 기법 및 품질 관련 이슈를 다룬다.
Software Engineering Project	소프트웨어공학 기법을 활용하여 프로젝트 수행	

- 전공 선택 커리큘럼 목록(9학점)

분야	교과과목	주요 교과내용
전공 선택	Numerical Analysis	기존의 과목과 유사함
	Programming Language	"
	Database Systems	"
	Computer Graphics	"
	Artificial Intelligence	"
	Principles of UNIX	"
	Object Oriented Programming Systems	"
	Compiler Design	"
	Automata Theory	"
	Computer Networks	"
	Senior Hours in CS/SE	"
	Introduction to VLSI Design	"

4.4 호주의 RMIT(Royal Melbourne Institute of Technology)대학의 소프트웨어공학 학사 과정

4.4.1 학과 구성 및 교육 목표

School of Computer Science and Information Technology에서는 다음과 같은 다양한 학사 학위 프로그램을 운영하고 있으며 소프트웨어공학 학위 과정도 이들 중에 포함되어 있다[6].

- Bachelor of Applied Science in Computer Science
- Bachelor of Applied Science in Computer Science with Honours(연구직 track)
- Bachelor of Applied Science in Computing & Internet Technology
- Bachelor of Applied Science(Information Technology)
- Bachelor of Design(Multimedia Systems)
- Bachelor of Applied Science in Software Engineering

다음과 같은 능력을 갖는 엔지니어를 배출하는 것을 목표로 삼고 있다.

- 고품질의 대형 소프트웨어 시스템을 공학적 방법으로 개발할 수 있는 능력
- 급속히 변하는 컴퓨팅 환경에 적응하면서 소프트웨어를 개발할 수 있는 능력
- 전문 직업 의식

4.4.2 특징 및 배경

4년제 과정 중에 1년을(3학년) 산업체에서 소프트웨어 개발 실무를 직접 경험하는 것이 다른 과정들과 크게 다른 점이다. 1년 동안의 현장 경력을 요구하기 때문에 상대적으로 다른 유사 학위 과정에 비해 교양 과목 및 선택 과목의 수가 매우 적다. 또한 각 학년별로 수강 과목이 정해져 있어서 수강생들이 거의 같은 강의 일정에 따라 교과과목을 수강한다.

전산학 학위 과정의 특징은 Bachelor of Computer Science 와 Bachelor of Computer Science with Honours라는 별개의 학위 과정을 운영하고 있는데 전자는 산업체 인력 양성을 그 목적으로 하고 있으며, 후자는 연구원 양성을 그 목적으로 하고 있다.

4.4.3 과정의 구성

전체 커리큘럼은 380 credit points(3 credit point 가 약 1 학점으로 추정됨)로 구성되어 있으며, 학년별로 다음과 같이 교과 과정이 구성되어 있다.

- 1학년(96 credit points)

학년	교과과목	주요 교과내용
1학년 (92 credit pts.)	Context Curriculum Subject(12cr.)	선택 과목으로 추정
	Programming Principles 1A(12cr.)	기존의 과목과 유사함
	Programming Principles 1B(12cr.)	"
	Intro. to Info. Tech.(12cr.)	"
	Computer Systems 1(12cr.)	"
	Intro. to Database Systems(12cr.)	"
	Mathematics for Computing(12cr.)	"
Approved Elective(12cr.)	선택 과목	

- 2학년(96 credit points)

학년	교과과목	주요 교과내용
2학년 (96 credit pts.)	Programming Principles 2B(12cr.)	기존의 과목과 유사함
	Computing Theory(12cr.)	"
	Operating Systems(12cr.)	"
	Software Engineering 1(24cr.)	"
	OO Programming(12cr.)	"
	Foundations of S/W Eng.(12cr.)	"
Programming Principles (12cr.)	"	

- 3학년(96 credit points)

학년	교과과목	주요 교과내용
3학년 (96 credit pts.)	Approved Industry Experience	산업체 현장에서 실무 경험

- 4학년(96 credit points)

학년	교과과목	주요 교과내용
4학년 (96 credit pts.)	Context Curriculum Subject(4cr.)	선택 과목으로 추정됨
	Foundations of AI(12cr.)	기존의 과목과 유사함
	File Structures and DB Systems(12cr.)	"
	Data Communication and Networks(12cr.)	"
	Software Engineering Project (24cr.)	"
	Professional Practice(8cr.)	"
	Approved Computer Science Subject(12cr.)	전공 선택 과목
Approved Computer Science Subject(12cr.)	"	

- 소프트웨어공학 전공 선택 과목들로는 다음과 같은 과목들이 개설되고 있다(각 12 credit points).  
병렬 프로그래밍(Concurrent Programming)  
컴퓨터 테크닉(Computer Techniques)  
소프트웨어 프로젝트 관리(Software project Management)  
소프트웨어공학 2(Software Engineering 2)  
실시간 및 병렬 시스템(Real-Time and Concurrent Systems)  
검증(Verification and Validation)  
소프트웨어 개발 및 유지 보수(Software Creation and Maintenance)

## 5. 소프트웨어공학 학사 과정의 교과 과정 제안

국내에서도 향후 소프트웨어공학 전문 인력의 수요가 증가할 것으로 예상되며 국내의 전산학 관련학과에서 기존의 교과 과정을 수용하면서 소프트웨어공학 전문 인력을 양성할 수 있는 소프트웨어공학 학위 교과 과정을 제안하고자 한다. 앞서 살펴 본 국외의 소프트웨어공학 학사 과정을 참고하여 작성하였으며 필요시 각 학교의 특수성을 고려하여 특화된 교과 과정을 운영하는 것이 바람직하여 보인다.

### 5.1 교과 과정 개요 및 특징

전산학과 교과 과정을 활용하되 일부 과목의 추가를 통하여 쉽게 소프트웨어공학 학위 과정을 만들 수 있다고 판단된다. 위의 국외 대학 프로그램의 경우 University of Texas, Dallas의 교과 과정과 유사한 프로그램 형태가 가장 적절한 사례로 여겨진다. 소프트웨어공학 분야에서 추가되는 과목은 SWEBOK에서 추천하는 소프트웨어공학 분야의 필요한 지식을 중점적으로 교육하도록 유도하며 이러한 교육의 결과를 팀 기반의 대형 프로젝트를 통하여 실습하도록 반드시 소프트웨어공학 프로젝트 과목이 함께 개설될 것을 적극 권장한다. 한 편 원만한 실습을 위해서 실습실의 장비 및 필요한 CASE 도구의 사용을 적극 권장하며 적절한 시설 및 도구에 대한 지원 및 우수한 교수 요원의 확보가 전제 조건으로 필요하다고 생각된다.

### 5.2 교과 과정의 구성

분야	교과과목	주요 교과내용
필수	소프트웨어공학 개론	소프트웨어공학의 정의, 필요성, 원리, 품질, 개발 프로세스 모델, 프로젝트 관리, 유지 보수, 형상 관리, 비용 산정 모델, 구현 기술, 국제 표준 등 기본적인 지식
	소프트웨어 분석 및 설계	요구 사항 분석 및 명세/검증, 설계 개념, 소프트웨어구조 및 아키텍처, 소프트웨어 설계 품질 분석 및 평가, 소프트웨어 설계 기법
	소프트웨어 시험 및 품질	시험 기본 개념, 시험 level 및 시험 기법, 시험 관련 측정, 소프트웨어품질 개념, SQA(Software Quality Assurance), 품질 분석 metrics, 소프트웨어 프로세스 개념, 정의 및 구조, 정량적인 프로세스 분석 및 측정, 프로세스 향상 기법 소개
	소프트웨어공학 프로젝트	CASE 도구를 이용하여 위의 필수 과목 수강에서 획득한 지식을 활용하여 팀 프로젝트를 실시(1학기 또는 2학기)

위의 4개 과목은 필요에 따라서는 소프트웨어 분석 및 설계를 2개의 과목으로, 또 소프트웨어 시험 및 품질도 시험 및 품질의 2개의 과목으로 운영하는 것도 가능하여 보인다. 현재 위의 4개 과목을 학년별로 나누면 3학년에서는 소프트웨어공학 개론, 소프트웨어 분석 및 설계, 4학년에서는 소프트웨어 시험 및 품질과 소프트웨어공학 프로젝트 과목을 교육하는 것이 바람직하여 보인다. 단 프로젝트를 강화하는 경우 4학년 1, 2학기를 통해 실시할 수 있으며 이 경우 시험 및 품질 과목과 병행하여 제공할 수 있다고 생각된다.

## 6. 결론

국내의 소프트웨어 산업 발전을 위해서는 고급 소프트웨어공학 인력을 양성하는 것이 필요함은 앞서 언급한 바와 같으며, 이를 위해서는 우선 소프트웨어공학에 대한 인식의 오류를 바로 잡는 것이 급선무라고 할 수 있다. 21세기 디지털 시대에서는 많은 제품들이 하드웨어뿐만 아니라 소프트웨어에 의해 전체 제품의 품질이 영향을 받게 되며 소프트웨어의 역할이 점차 커져 가고 있으며, 주먹 구구 식의 소프트웨어 개발이 아닌 공학적인 접근 방법에 의한 체계

적인 소프트웨어 개발 및 관리를 하는 것이 경쟁력 향상과 직결된다는 것을 명확히 인식하여야 한다.

CSEP 프로그램의 소프트웨어 엔지니어 자격 조건에서 알 수 있듯이 고급 소프트웨어 엔지니어의 육성은 장기간의 투자를 필요로 한다. 단기간에 인력 양성을 목적으로 하는 계획보다는 장기적인 안목에서 고급 인력을 육성하기 위한 투자가 정부, 기업 연구소 및 학계에서 지속되어야 한다. 이를 위해 교육 기관에서는 각 학교의 특성을 살릴 수 있는 차별화된 소프트웨어 공학 인력 양성 교육 프로그램을 추진하는 것이 바람직하여 보인다. 소프트웨어 개발은 전산학과를 졸업해야만 할 수 있는 것은 아니며, 오히려 다양한 도메인 지식을 필요로 한다. 예를 들어 통신용 소프트웨어, 또는 금융 솔루션을 만들기 위해서는 각각 통신, 금융 등 해당 분야의 지식이 없이는 제대로 소프트웨어를 만들 수 없다. 따라서 소프트웨어공학 전문 과정 역시 전산학 전공자들에게만 기회를 줄 것이 아니라 다양한 전문 지식을 가진 사람들에게도 그 문호를 개방하여야 한다.

또한 연구 중심과 산업체 요원 중심의 교과 과정으로 별도의 프로그램을 만들어 보는 것도 검토가 필요하다고 판단된다. 국내의 경우 대부분의 대학이 연구 중심의 전산학 교과 과정을 유지하고 있는 것처럼 보이는데, 연구원 양성뿐만 아니라 산업체에서 필요로 하는 인력 양성에도 눈을 돌려야 할 것이다. 이를 위해서는 산업체의 요구 사항을 명확히 파악하여 소프트웨어공학 이론 및 실무 능력 배양을 위한 실질적인 노력이 절실히 요구된다. 한편 정부는 소프트웨어공학을 IT의 한 분야로서 상품화에 목적을 둔 연구 개발에 투자를 할 것이 아니라 국가적인 차원에서 전략 학문 분야로 육성할 수 있는 장기 계획이 수립이 기대된다.

## 참고문헌

- [1] Guide to Software Engineering Body of Knowledge, <http://www.swebok.org>
- [2] Certified Software Engineering Professional Program, <http://www.computer.org/certification>
- [3] Master of Software Engineering Program, Carnegie Mellon University, <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/mse/www/index.html>



- [4] Master of Software Engineering Program, Monmouth University, <http://www.monmouth.edu/%7Esegrad.segrad-cur.htm>
- [5] Bachelor of Software Engineering, University of Texas, Dallas, <http://www.cs.utdallas.edu/student/catalog/undergraduate01/ecs.html>
- [6] Bachelor of Applied Science in Software Engineering, RMIT University, <http://www.cs.rmit.edu.au/courses/undergraduate/se/seintro.shtm>



**배 두 환**

1980 서울대학교 공과대학 학사  
1987 University of Wisconsin · Milwaukee 전산학 석사  
1992 University of Florida 전산학 박사  
1992~1994 University of Florida 전산학과 조교수  
1995~현재 한국과학기술원 전자전산학과 전산학 전공 부교수  
관심분야: 객체 지향/컴포넌트 소프트웨어, 소프트웨어 프로세스 및 개선 기법

E-mail:bae@salmosa.kaist.ac.kr

**• 컴퓨터시스템연구회 2002 동계 워크샵 •**

- 일 자 : 2001년 1월 31일 ~ 2월 2일
- 장 소 : 강원도 피닉스파크
- 내 용 : 논문발표 등
- 주 최 : 컴퓨터시스템연구회
- 문 의 처 : 고려대학교 컴퓨터학과 유 혁 교수

Tel. 02-3290-3198