

# 컴퓨터공학 분야 설계교육 방법론

김상진

한국기술교육대학교 컴퓨터공학부

## Design Education Methodology in Computer Science and Engineering

Sangjin, Kim

School of Computer Science and Engineering, Korea University of Technology and Education

### ABSTRACT

Commonly it is perceived that it is difficult to teach engineering design and satisfy accreditation criteria in computer science and engineering related majors. However, since engineering design in these disciplines are a product design, it is more adaptable than other majors which are process design oriented. This paper shows that although there are subtle differences with conventional engineering design, engineering design education in computer science and engineering can be effectively done. This paper concentrates on how engineering design related curriculum can be constructed, what should be considered when designing such curriculum, and how engineering design can be taught in individual design courses(excluding introductory and capstone courses) based on case experience.

**Keywords:** Engineering Design, Computer Science and Engineering, Accreditation, Design Courses

### 1. 서 론

공학(engineering)은 과학을 응용하여 문제를 해결하는 것이고 설계(design)는 지식의 창의적 표현이라고 할 수 있다. 따라서 공학설계(engineering design)는 기술을 이용하여 어떤 기능을 하는 제품이나 제품의 구성요소를 만드는 과정을 말하며(Dym et al., 2005), 이 과정에는 과학적 요소, 공학적 요소, 예술적 요소가 모두 포함된다. 즉, 공학설계는 과학적 분석과 지식의 합성을 통해 삶의 질을 향상시킬 수 있는 것을 개발하는 것이고, 그 형상화 과정에서는 예술적 요소도 포함된다. 따라서 우리가 공학을 전공하는 것은 이와 같은 공학설계 과정에 참여하기 위함이고, 이를 위해 공학을 전공할 때 충분한 설계 경험을 얻을 수 있어야 한다. 이것은 모든 공학분야에 동일하게 적용할 수 있는 것이다. 공학 분야에 따라 세부적인 차이가 있을 수 있지만 큰 개념에서는 차이를 찾기는 쉽지 않다. 이 때문에 공학교육인증에서도 공학 분야와 상관없이 설계 교육을 공히 매우 강조하고 있다.

공학설계는 크게 제품 설계(product design)와 절차(공정) 설계(process design)로 구분된다. 공학인증기준에 사용된 개

념이나 용어는 전통적인 기계공학 분야 설계에서 가지고 온 것이며, 이 분야는 제품 설계에 해당된다. 컴퓨터공학도 제품 설계에 해당되기 때문에 오히려 화학공학, 재료공학 등 절차 설계에 해당하는 공학 분야와 달리 공학인증기준에 제시된 설계 개념을 적용하기에 큰 무리가 없다. 그럼에도 불구하고 많이 다르다고 주장하는 경우도 있으며, 이것이 컴퓨터공학이 다른 공학과 분리된 인증기준을 가지게 된 여러 이유 중 하나로 간주되고 있다.

하지만 실제 컴퓨터공학 분야는 다른 어떤 공학 분야보다도 설계교육을 효과적으로 할 수 있는 학문분야이다. 이 논문에서는 이 주장을 뒷받침하기 위해 컴퓨터공학 분야설계 교육의 특징과 다른 공학 분야와의 차이점을 분석하여 제시한다. 또한 여러 대학의 컴퓨터공학 분야 설계 교육 현황을 조사하고 요소설계 교과와 종합설계 교과를 운영한 경험을 바탕으로 컴퓨터공학 분야의 설계교육 방법론을 제시한다. 특히, 전공분야와 독립적으로 운영이 가능한 기초입문설계와 한국공학교육인증원의 노력으로 많이 정착된 종합설계 교과를 제외한 컴퓨터공학에서 요소설계 교과를 어떻게 운영하고 전체적인 설계 이수체계를 어떻게 구성하는 것이 적절한지 제시한다.

기존 문헌에는 이 주제와 관련된 논문을 찾아보기 힘들다. 국내 공학교육연구 논문지나 국제 JEE 논문지에는 일반적인

Received February 23, 2015; Revised July 20, 2015

Accepted July 21, 2015

† Corresponding Author: sangjin@koreatech.ac.kr

설계 교육 관련 논문은 다수 있지만 컴퓨터공학 분야에 특화된 설계교육 관련 논문은 아직 없다. ACM SIGCSE에서 주최하는 학술대회인 경우에는 컴퓨터공학 분야 종합설계 관련 논문들은 다수 발표된 적이 있지만 컴퓨터공학 분야 관련 전체적인 설계 교육에 관한 논문은 아직 없다. IEEE에서 주최하는 각종 공학교육 학술대회의 경우에도 유사하다.

따라서 이와 같은 연구결과를 제시함으로써 컴퓨터공학 분야를 포함하여 각 공학 전공분야별 설계교육 방법론에 대한 연구가 활성화되기를 기대한다. 특히, 컴퓨터공학 분야의 설계교육에 대한 잘 못된 인식을 전환하는데 기여하고자 한다.

이 논문의 구성은 다음과 같다. 2장에서는 공학분야에서 일반적 설계교육과 공학인증기준(한국공학교육인증원, 2014a)에서 설계교육과 관련된 요구사항을 살펴보고, 3장에서는 이를 토대로 컴퓨터공학 분야에서 전반적 설계교육 방법론을 살펴본다. 이 때 여러 대학의 현황을 조사하여 제시하며, 이를 통해 컴퓨터공학 분야에서 설계 교육 과정을 어떻게 구성하는 것이 바람직한 것인지 제시한다. 4장에서는 컴퓨터공학에서 요소설계 교과 운영방법을 사례를 바탕으로 제시한다. 끝으로 5장에서 결론과 향후 연구방향을 제시한다.

## II. 공학분야 설계교육

### 1. 공학인증기준과 설계교육

공학인증 세부기준 중 교과과정에서 강조하고 있는 것 중 하나가 설계 교육이다. 설계란 수학, 기초과학, 전산학과 공학주제 영역의 이론을 기초로 하여 설계 절차나 설계의 결과물이 목표로 하는 기능과 성능을 포함한 현실적 제한조건을 만족하는 시스템이나 시스템의 일부를 고안하는 전 과정(예, 문제의 이해, 창의적인 아이디어 탐구, 수행 계획 수립, 개념 설계, 구체 설계 등 점진적 상세화의 단계)에 있어서 사용 가능한 자원을 최적으로 활용될 수 있게 하는 반복적인 의사결정 과정을 말한다(한국공학교육인증원, 2014c). 설계과정을 구성하는 설계요소에는 목표와 기준의 설정, 합성, 분석, 제작, 시험, 평가, 그리고 결과 도출 등이 포함된다. 또한 설계과정에서 다양한 현실적 제한조건을 고려하여야 한다. 이것은 산업 현장에서 실제 제품을 설계할 때 다양한 현실적 제한조건을 고려하여 설계되기 때문에 이를 경험할 수 있도록 하기 위한 것이다. 현실적인 제한조건이란, 예를 들어, 경제, 환경, 사회, 윤리, 미학, 보건 및 안전, 생산성과 내구성, 산업표준 등 설계의 절차나 설계의 결과물에 반드시 적용되어야 하는 제한 조건을 의미한다. 현실적 제한조건은 종종 제품 정의에 포함되는 제한조건과 혼

동되는 경우가 있다.

KEC2005 마지막 개정(한국공학교육인증원, 2012a)부터 설계학점을 공통기준에서 전공분야별 인증기준으로 옮겨, 각 전공분야 특성에 맞게 최소이수 설계학점을 정의할 수 있도록 하였다. 보통 현재 12학점 이상을 이수하도록 요구하고 있다.

### 2. 설계교과의 분류

설계 교과는 설계 교육 내용에 따라 기초입문설계, 요소설계, 종합설계로 분류할 수 있으며, 교과의 설계학점에 따라 부분설계와 순수설계로 분류할 수 있다.

기초입문설계에서는 설계에 대한 기본적인 개념 및 전반적인 수행 절차에 대한 교육과 더불어 창의력을 기르기 위한 교육으로 구성되어야 하며, 종합설계에서는 저학년에서 배운 지식과 기술을 기초로 하고 개별적인 설계 교과목으로부터의 설계 경험을 아우르는 교육으로 구성되어야 한다. 요소설계 교과의 경우 그 정의가 다소 모호하다. 하지만 요소설계에서 요소는 설계 구성요소를 말하는 것이기 때문에 일반적으로 각 개별 설계 구성요소에 초점을 두어 종합설계 이전에 설계교육을 진행하는 교과를 말한다. 2014년 자체평가보고서 작성양식(한국공학교육인증원, 2014c)부터는 요소설계라는 용어 대신에 개별 설계 교과라는 용어를 사용하고 있으며, 현재는 두 용어를 같이 사용하고 있다. 기초입문설계와 종합설계를 제외하고 설계학점이 있는 교과는 모두 요소설계 교과로 분류할 수 있으며, 종합설계 이전에 충분한 설계 경험을 할 수 있도록 제공해야 하는 교과이다.

부분설계란 해당 교과에 배정된 설계 학점이 해당 교과의 전체 학점보다 작은 경우이고, 순수설계란 해당 교과의 전체 학점과 교과의 배정된 설계학점이 같은 경우를 말한다. 즉, 순수설계 교과에서 설계교육만 진행되는 교과를 말한다.

보통 기초입문설계와 종합설계는 순수설계로 운영되고, 요소설계는 부분설계로 운영된다. 물론 요소설계도 순수설계로 운영될 수 있으며, 각 학년별로 학년에 맞는 순수설계 교과를 요소설계 교과로 개설하여 운영하는 것이 교육적 측면에서 효과적이다. 그 이유에 대해서는 3장 6절에서 자세히 논한다. 공학인증 기준에서 요구하는 설계학점 최소이수학점이 18학점에서 12학점을 축소된 이후, 프로그램들은 개설하는 요소설계 교과의 수를 많이 줄이고 있다.

### 3. 설계이수체계

설계 이수체계는 공학분야와 상관없이 1학년에 기초입문설

계 교과, 2, 3학년에 몇 개의 요소설계 교과, 4학년에 종합설계 교과를 편성하여 구성하는 것이 일반적이다. 종합설계 교과의 경우에는 프로그램마다 차이가 있지만 3학년 2학기부터 또는 4학년 1학기부터 2학기 연계된 교과로 운영하는 경우도 많다. 설계 이수체계에 대한 엄격한 평가를 시행한 초창기에는 기초 입문설계 교과보다 앞서 수강한 요소설계 교과와 종합설계 교과보다 늦게 수강한 요소설계 교과의 설계학점은 인정하지 않았다.

공학인증에서는 교과과정 이수체계가 적절하게 수립되어 공식적인 자료에 공개되어 있어야 하며, 이수체계 준수를 보장하는 규정이 실질적으로 시행되고 있어야 한다(한국공학교육인증원, 2013a). 이수체계 준수를 보장하는 규정의 시행이란 선수 교과목을 이수하지 않은 경우에는 후수 교과목을 이수하지 못하도록 하는 것을 말한다.

한국공학교육인증원은 지금까지 교과과정 이수체계에 대한 평가에 있어 설계 이수체계에 대해 많은 강조를 하였다. 2012년 EAC 중점점검사항에는 기초설계로 시작하여 종합설계로 완성되는 설계교과목 체계가 있어야 한다고 제시되었으며, 해당년도 판정가이드에는 설계를 포함하는 공학주제 교과목간 선수-후수 관계의 명시가 적절치 않거나 종합설계 교과목을 정점으로 하는 이수체계가 구성되어 있지 않으면 이수체계가 적절하지 않은 것으로 규정하고 있다. 최근 판정 가이드에서는 설계 이수체계와 관련하여 적절하지 않은 예로 1) 기초설계 교과목이 다른 설계과목의 선수과목으로 지정되지 않거나, 2) 종합설계 교과목에 다른 설계과목이 선수과목으로 지정되지 않거나를 제시하고 있다(한국공학교육인증원, 2014c).

많은 대학의 경우 기초설계 교과와 종합설계 교과간에 필수 선후수 관계를 규정하고 있고, 요소설계의 선수로 기초설계, 종합설계의 선수로 몇몇 중요 요소설계를 규정하는 것이 일반적이다. 하지만 강제 선후수 도입에 대해 부담스러워 하는 대학의 경우에는 최소한의 선후수만 지정하는 경우도 많다.

#### 4. 설계교과 요건

공학인증 기준 판정가이드에 의하면 기초입문설계와 종합설계를 제외한 나머지 설계교과에서는 다음을 반드시 충족해야 설계교과로 인정받을 수 있다.

- 조건1. 설계 구성요소를 하나 이상 다루어야 함
- 조건2. 현실적 제한조건을 하나 이상 다루어야 함
- 조건3. 설계주제는 개방형 문제이어야 함
- 조건4. 설계학점에 부합하는 비중으로 설계 교육이 이루어

#### 저야 함

설계와 관련된 교육 비중은 다음에 의해 결정된다고 자체평가보고서 작성양식(한국공학교육인증원, 2013b)에 제시되어 있다.

- 전체 강의시간에 대한 비중
- 학생들이 수행하는 전체 학업 또는 과업에 대한 비중
- 프로그램이나 담당 교수가 합리적으로 정한 바에 따르는 비중

이것이 모호한 경우가 많기 때문에 많은 대학에서는 내규에 설계와 관련된 교육 비중을 어떻게 다루어야 하는지 규정하고 있다.

개별교과가 아니라 전체 교육과정 측면에서 검토해보면 종합설계 교과 이전에 이루어지는 설계교육에서 설계 구성요소와 현실적 제한조건에 대한 교육이 체계적이고 균형 있게 이루어져야 한다는 요구가 있다. 요소설계 교과의 수를 현재 많이 줄이고 있기 때문에 이 조건은 충족되기 매우 어렵다.

기초입문설계와 종합설계 교과목에서는 다음을 반드시 충족해야 기초입문설계, 종합설계 교과로 인정받을 수 있다.

- 조건1. 모든 설계 구성요소를 골고루 다루어야 함
- 조건2. 과제의 특성이 요구하는 현실적 제한조건을 모두 다루어야 함
- 조건3. 설계주제는 개방형 문제이어야 함
- 조건4. 의사소통 능력과 팀워크를 다루어야 함

이와 같은 측면에서 요소설계 교과에서는 의사소통과 팀워크를 다루지 않아도 된다는 것을 의미하지만 체계적 설계교육을 위해서는 요소설계 교과에서도 다루는 것이 필요하다.

### III. 컴퓨터공학 분야 설계교육

#### 1. 컴퓨터공학과 설계

컴퓨터공학의 경우 공학인증기준 또는 컴퓨터정보기술인증기준 중 하나를 선택하여 공학교육인증제를 도입하여 운영할 수 있다. 컴퓨터공학이 다른 공학분야와 달리 별도의 컴퓨터정보기술인증기준을 가지고 있는 여러 이유 중 하나가 설계교육이다. 이것은 KCC2010 인증기준(한국공학교육인증원, 2012)에서 설계 용어를 사용하지 않고 프로젝트라는 용어를 사용하고 있는 것만 보아도 이를 쉽게 인식할 수 있다. 물론 KCC2015 인증기준(한국공학교육인증원, 2014d)

을 새롭게 제정하면서 다시 설계 용어를 사용하고 있다. 컴퓨터공학 분야의 설계가 다른 공학 학문분야와 다른 점은 다음과 같다.

- 첫째, 컴퓨터공학은 소프트웨어공학이라는 설계 관련 세부 학문이 있는 몇 안되는 공학분야이다.
- 둘째, 기존 인증기준에서 제시된 설계 구성요소 관련 용어들이 컴퓨터공학 설계 과정 용어와 맞지 않는다.
- 셋째, 설계 결과로 순수 소프트웨어(하드웨어 제작 또는 특수 하드웨어를 활용하지 않는)를 만드는 경우에는 공학인증 기준에서 보통 제시되는 여러 현실적 제한조건을 고려하기 어렵다.
- 넷째, 소프트웨어 배포 환경의 급격한 변화로 대학 설계교육 과정에서 학생들이 개발한 시스템을 쉽게 배포하고 운영해 볼 수 있다.

이 장에서는 컴퓨터공학 분야의 설계교육을 설명하기 위해 다음 대학 학부(과)의 운영현황을 활용한다.

- 경북대학교(KNU) 컴퓨터학부
- 경희대학교(KHU) 컴퓨터공학과
- 동국대학교(Dongguk) 컴퓨터공학과
- 목포대학교(Mokpo) 컴퓨터공학과
- 서강대학교(Sogang) 컴퓨터공학과
- 서울과학기술대학교(SeoulTech) 컴퓨터공학과
- 숭실대학교(SSU) 컴퓨터학부
- 아주대학교(Ajou) 정보컴퓨터공학부
- 연세대학교(Yonsei) 컴퓨터과학과
- 이화여자대학교(Ewha) 컴퓨터전자공학부
- 충남대학교(CNU) 컴퓨터공학과
- 한국기술교육대학교(KoreaTech) 컴퓨터공학부
- 한국산업기술대학교(KPU) 컴퓨터공학과

위 13개 학부(과)는 모두 인증을 받은 프로그램을 운영하고 있으며, 한국기술교육대학교와 서울과학기술대학교를 제외하고는 모두 CAC로 인증을 받은 상태이다.

## 2. 설계 구성요소

컴퓨터공학 분야에서 설계 구성요소는 기존 많은 자료에서 제시된 목표와 기준의 설정, 합성, 분석, 제작, 시험, 평가, 그리고 결과 도출 대신에 소프트웨어공학의 전통 개발 과정인 폭포수 모델(waterfall model)(Winston, 1970)에 따라 요구사항 분석, 설계, 구현, 통합, 시험 및 디버깅, 배포, 유지보수를 설

계 구성요소로 정의하여 사용하는 것이 적절하다. 즉, 설계 구성요소는 각 종 공학인증 자료에서 제시되는 용어가 컴퓨터공학 분야와 맞지 않는 것일 뿐, 컴퓨터공학에 맞게 재정의하여 사용하면 문제될 것이 없다.

컴퓨터공학에서 폭포수 모델에 제시된 핵심 단계들을 설계 구성요소로 정의하여 사용하더라도 이 단계들의 순차적 진행을 강조할 필요는 없다. 하지만 한번쯤 순차적 진행을 엄격하게 지키도록 설계 과정을 진행하여 폭포수 모델의 단점을 이해하고 각 단계에서 수행하는 활동을 경험할 수 있도록 하면 교육적 효과가 크다.

또한 애자일(agile) 소프트웨어 개발(Martin, 2003)과 같은 반복적 개발(iterative development) 방법론들을 활용하여 설계를 진행하도록 하는 것이 필요하다. 더구나 이제는 구글 플레이, 애플 앱 스토어 등 오픈 마켓에 개발한 소프트웨어를 배포해보기 쉬우므로 배포, 유지보수, 운영 단계까지 교육과정에서 충분히 경험해 볼 수 있도록 할 수 있다.

## 3. 현실적 제한조건

공학인증 기준에서 제시하는 현실적 제한조건은 경제, 환경, 사회, 윤리, 미학, 보건 및 안전, 생산성과 내구성, 산업표준 등을 말한다. 컴퓨터공학의 경우 하드웨어 제작이 포함되지 않는 순수 소프트웨어 제품을 개발할 경우에는 보통 공학에서 사용되는 제한조건의 의미를 그대로 적용하기 힘든 측면이 있다. 하지만 아래에 제시된 것처럼 몇 가지 제한조건을 제외하고는 충분히 고려가 가능하다.

- 경제(economy): 경제성 관점에서 프로젝트 진행 여부를 결정하는 것. 주요 고려사항은 제품 개발에 소요되는 비용, 제품 가격(시장에서 경쟁력), 생산자와 소비자 입장에서 소유 비용 등임
  - 학문분야 상관없이 충분히 적용할 수 있는 조건
- 환경(environmental): 제품의 생산, 사용, 폐기가 환경에 미치는 영향을 고려하는 것. 이를 위해 재료의 선택, 기술의 선택 등이 달라질 수 있음
  - 하드웨어 요소가 포함되지 않을 경우에는 컴퓨터공학 분야에서는 고려하기 힘들
- 보건 및 안전(health and safety): 공학은 지식을 응용하여 인류의 삶의 질을 향상시키는 것이므로 개발된 제품이 일상적으로 사용될 때 어떤 해를 야기하면 안 됨. 안전한 제품을 만든다는 것은 제품 자체가 어떤 위험도 없어야 한다는 것을 의미하는 것은 아니며, 발생 가능한 위험이 수용

가능한 수준이어야 한다는 것을 말함. 모든 제품에는 수명이 있기 때문에 제품에서 발생될 수 있는 문제점(재료의 피로, 나쁜 설계, 환경 악화, 인간 조작 오류)을 설계자가 파악하고 있어야 함

- 하드웨어 요소가 포함되지 않을 경우에는 컴퓨터공학 분야에서는 고려하기 힘들
- 생산성(manufacturability): 실제 생산이 가능하도록 설계하는 것. 이 때 생산의 효율성, 신뢰성, 비용 측면을 고려해야 함. 예를 들어 나누어지는 parts의 수, 공통 재료와 parts의 활용 등을 고려하여 설계해야 한다는 것을 말함
  - 순수 소프트웨어 시스템의 경우에는 모듈화와 라이브러리 활용, 오픈소스 등에 대해 검토하는 것으로 바꾸어 정의할 필요가 있음
- 내구성(sustainability): 미래 세대의 필요를 위태롭게 하지 않으면서 현재의 필요를 충족할 수 있도록 가용한 자원을 사용하여 제품을 만들어야 한다는 측면과 제품 자체의 수명을 연장하기 위한 방안을 설계시 고려해야 함. 제품의 생산과 사용에 재생가능한 자원의 사용 여부와 제품의 수명이 다하였을 때 제품의 재사용 또는 재활용(recycle) 가능 여부는 반드시 고려되어야 함.
  - 순수 소프트웨어 시스템의 경우 제품의 수명 측면에서는 해당 제품이 얼마나 오랫동안 사용될 수 있는 제품 인지와 유지보수와 확장을 고려한 설계여부에 대해 검토하는 것으로 바꾸어 정의하여 사용할 필요가 있음
- 사회(social): 인간의 필요와 사회적 이슈를 해결하기 위한 제품을 설계해야 한다는 측면에서 제품의 사회적 영향을 고려해야 한다는 것을 말함.
  - 예) 이 제품이 소수자를 불쾌하게 하지 않는지, 게임에 포함된 폭력 수위가 적절한 것인지 등
  - 학문분야 상관없이 충분히 적용할 수 있는 조건
- 미학(aesthetics): 대상 사용자 층에게 어떻게 매력적으로 느끼게 해줄 것인지에 대한 검토를 말하며, 제품의 모양, 색깔, 감촉 등 모든 요소를 말하며 현재와 미래 트렌드를 고려하여야 함
  - 사용자 상호작용 관점에서 사용자 인터페이스 개발에 반드시 적용해야 하는 조건임
- 정치(political): 공학과 정치 활동 간의 상호작용을 이해하고 고려되어야 함
  - 예) 특정 성별, 특정 계층, 특정 인종의 부정적 측면을 주목시키는 제품
  - 학문분야 상관없이 사회라는 큰 범위 내에서 검토되어야 함.

- 윤리(ethical): 다른 사람들과 상호작용할 때 올바른 행동의 표준을 제공하여 주는 행동 코드(예: 주장에 대한 정직, 뇌물 거부 등)를 인식하고 있어야 함
  - 윤리는 제품 자체에 대한 것보다는 개발과정에 대한 것임, 학문분야와 상관없이 충분히 적용할 수 있는 조건
- 법률(legality): 전문영역과 관련된 법률이나 정부 규제를 이해하여야 하며, 저작권, 특허와 관련 법률에 대한 이해도 필요함
  - 해당 제품을 개발할 때 해당 제품과 관련된 법 유무를 알아보는 것은 학문분야 상관없이 적용할 수 있는 조건이며, 컴퓨터공학 분야의 경우에는 특히 정보통신 관련 많은 법률이 있기 때문에 충분히 적용할 수 있는 조건임
- 산업표준(Standards): 해당 제품과 관련된 산업표준을 적용해야 함
  - 산업표준은 법률과 함께 검토가 가능한 것이며, 이 역시 학문 분야와 상관없이 적용이 가능한 점검임

따라서 환경, 안전, 생산성, 내구성을 제외하고는 컴퓨터공학 분야에서는 제품의 종류와 상관없이 충분히 활용할 수 있다. 생산성과 내구성도 컴퓨터공학에 맞게 재해석하면 사용이 가능하다. 그러므로 컴퓨터공학에서 현실적 제한조건을 설계 교육에 적용하기 어렵다는 것은 잘못된 인식에서 비롯된 것임을 알 수 있다.

이와 같은 현실적 제한조건은 제품을 분석, 설계할 때 고려되어야 하며, 최종 제품을 완성한 다음에도 영향평가를 해야 한다.

#### 4. 기초입문설계 교과 운영현황

12개 프로그램 중 11개 프로그램이 1학년에 개설하고 있으며, 1학년 2학기에 개설하는 프로그램은 그 중에 7개이다. 경북대 프로그램을 제외하고는 모두 순수설계 교과이며, 3학점으로 구성되어 있다. 기초입문설계는 전공분야와 독립적으로 운영될 수 있는 교과이며, 융합교육을 고려할 경우에는 루이지아나 테크처럼 입학한 전공분야와 상관없이, 기계주제, 전자/컴퓨터 주제, 재료주제를 모두 경험하도록 하는 형태의 운영도 충분히 고려할 필요가 있다. 특히, 다양한 전공학생들을 같은 분반에서 수강하게 하고 다학제팀을 구성하여 설계주제를 진행하면 고학년 융합교육에 큰 역할을 할 수 있다.

Table 1 Operation Status of Introductory Design Course

Univ.	Credit	Design Credit	Year	Semester
KNU	3	2	1	2
KHU	3	3	1	2
Dongguk	3	3	1	2
Mokpo	3	3	1	2
Sogang	3	3	1	1
SeoulTech	2	2	1	2
SSU	3	3	1	2
Ajou	3	3	1	1
Yonsei	3	3	1	2
Ewha	3	3	2	1
CNU	3	3	1	1
KoreaTech	3	3	1	2
KPU	3	2	1	2

Table 2 Operation Status of Capstone Design Course

Univ.	Credit	Design Credit	# of courses	Start Y-S	End Y-S
KNU	8	8	2	3-2	4-1
KHU	3	3	1	4-1	-
Dongguk	6	6	2	4-1	4-2
Mokpo	6	6	2	4-1	4-2
Sogang	6	5	2	4-1	4-2
SeoulTech	6	6	2	4-1	4-2
SSU	6	6	2	4-1	4-2
Ajou	4	4	1	4-2	-
Yonsei	6	6	2	4-1	4-2
Ewha	6	6	2	4-1	4-2
CNU	6	6	2	4-1	4-2
KoreaTech	6	6	4	3-1	4-2
KPU	6	6	2	4-1	4-2

5. 종합설계 교과 운영현황

12개 프로그램 중 9개 프로그램이 종합설계 교과를 두 학기 연속된 2개 교과로 운영하고 있으며, 2개 프로그램만 1과목으로 운영하고 있다. 2과목으로 운영하고 있는 9개 프로그램 중 8개 프로그램이 4학년 1학기부터 교과를 편성하고 있다. 이들 9개 프로그램 중 2개를 제외하고는 각 교과는 3학점 순수설계 교과로 운영하고 있다. 따라서 종합설계 교과 2개를 모두 이수하면 6학점의 설계학점을 이수하게 된다.

컴퓨터공학 분야 종합설계의 경우 오픈 마켓에 배포가 가능한 작품은 배포하도록 하여 배포, 유지보수, 운영 단계를 경험해볼 수 있도록 할 수 있다. 또한 특히 교육과 연계하여 새 아이디어를 특허로 출원하는 교육도 가능하다. 본 대학은 이를 위해 공학설계와특허라는 교과를 운영하고 있다.

6. 요소설계 교과 운영현황

가. 요소설계 교과 구성

공학인증 초창기에는 설계학점 최소 이수 요건이 18학점이었기 때문에 기초입문설계와 종합설계 교과를 제외하더라도 많은 교과를 부분설계 교과로 운영할 수밖에 없었으며, 이들 교과는 전공선택으로 분류되더라도 공학인증 졸업요건을 충족해야하기 때문에 해당 교과가 필수 교과로 인식되는 경우가 많았다. 이에 교과 설패강 문제와 연계되어 거의 대부분의 전공교과를 부분설계 교과로 운영하는 프로그램도 존재하였다. 이 경우 다음과 같은 문제들이 발생할 수 있다.

- 부분설계 교과이지만 해당 교과에서 공학인증에 맞는 설계교육이 이루어지지 않을 수 있음
- 부분설계 교과에서는 대부분 학기 프로젝트를 수행한다. 이 때문에 학생들의 학습 부담이 많이 증가할 수 있음
- 부분설계 교과에서 해당 설계 비중만큼 충실하게 설계교육을 할 경우에는 해당 교과에서 원래 다루어야 하는 교육내용에 영향을 줄 수 있음
- 전체적인 설계 이수체계 측면에서 각 요소설계 교과의 설계교육 내용을 결정하는 것이 아니라 각 교과의 특성과 담당교수에 의해 결정되는 경우가 많음. 이 경우에는 전체 설계 교육과정에서 설계 구성요소와 현실적 제한조건에 대한 교육이 체계적이고 균형 있게 이루어지기 힘들 수 있음

많은 경우 학기 프로젝트 수행이 가능한 교과를 요소설계 교과로 지정하고 설계 구성요소나 현실적 제한조건에 대한 고려 없이 각 교수가 알아서 설계교육을 진행하도록 운영하는 경우가 많다. 이 경우 교과의 성격을 고려하여 팀 프로젝트를 수행하게 되며, 이 프로젝트는 특정 설계 구성요소에 집중된 설계교육을 하는 것이 아니라 전체 설계 과정을 간단하게 경험하는 형태로 진행된다. 이와 같은 경험은 해당 교과의 목표를 달성하는데 도움이 될 수 있지만 설계교육 측면에서는 잘 못된 접근방법일 수 있다.

이를 극복하는 한 가지 방법은 각 학년별로 해당 학년에 맞는 순수설계 교과를 개설하여 운영하는 것이다. 본 대학에서도 2학년에 컴퓨터시스템기초설계라는 교과를 2014년에 신설하

여 운영하고 있으며, 다른 대학에서도 다음과 같이 유사한 교과를 운영하고 있다.

- 경북대: 소프트웨어설계(2-2, 3학점, 2설계)
- 경희대: 설계프로젝트C(3-1, 3학점, 3설계), 설계프로젝트 D(3-2, 3학점, 3설계)
- 동국대: 주니어디자인프로젝트(3-1, 3학점, 3설계)
- 서강대: 고급소프트웨어실습1(3-2, 3학점, 3설계)
- 서울과기대: 객체지향분석설계(3-2, 3학점, 2설계)
- 아주대: 컴퓨터프로그램설계(1-2, 3학점, 2설계), 소프트웨어분석설계(3-2, 3학점, 2설계)
- 연세대: 창의소프트웨어설계(4-1, 3학점, 3설계), X-design (4-1, 3학점, 3설계)
- 한국기술교육대: 컴퓨터시스템기초설계(2-1, 3학점, 3설계)

나. 소프트웨어공학 교과 활용 현황

컴퓨터공학의 경우에는 소프트웨어공학을 대부분 요소설계 교과로 활용할 수 있으며, 실제로 반드시 활용되어야 한다. 또 공인원에서 제시하는 2014년 자체평가보고서 작성양식에 의하면 이 교과는 순수설계 교과로 충분히 사용할 수 있다. 그 이유는 이 교과에서 강의하는 모든 이론 내용이 설계 절차에 대한 교육이기 때문이다. 하지만 이와 같은 이유로 소프트웨어공학을 순수 설계교과로 설정해 운영하는 대학은 거의 없다.

Table 3 Operation Status of Software Engineering Course

Univ.	Credit	Design Credit	Year	Semester	Mandatory
KNU	3	1	3	1	×
KHU	3	0	3	2	×
Dongguk	3	1	3	2	△
Mokpo	3	1	3	2	○
Sogang	3	2	4	1	△
SeoulTech	3	0	3	1	×
SSU	3	0	3	1	×
Ajou	3	1	4	2	×
Yonsei	3	1	3	2	×
Ewha	3	1	3	2	○
CNU	3	1	3	2	×
KoreaTech	3	1	3	2	×
KPU	3	0	3	1	○

△: 선택적 필수

조사된 13개 프로그램 중 9개 프로그램이 소프트웨어공학을 요소설계 교과로 운영하고 있으며, 13개 프로그램 중 7개 프로그램이 3학년 2학기에 개설하고 있다. 소프트웨어공학을 요소설계로 운영하고 있지 않는 원인은 다양할 수 있지만 종종 인증 평가가 부담되어 설계 교육을 하고 있음에도 불구하고 설계 교과로 표시하지 않는 경우도 있다. 특히, 인증기준에서 최소이수 설계학점이 줄어든 이후에는 이와 같은 경향이 늘고 있다.

7. 컴퓨터공학 설계 이수체계

지금까지 살펴본 바에 따라 컴퓨터공학 분야에서 설계 이수체계는 다음과 같이 구성하는 것이 바람직하다.

- 1학년에 기초입문설계 교과를 순수설계 교과로 운영함
- 2학년의 경우 2학기에 문제정의, 분석에 초점을 둔 순수설계 교과를 운영함
- 3학년에는 설계 과정에 대한 전문지식을 교육하기 위해 소프트웨어공학 교과를 설계 교과로 운영함
- 4학년에는 2학기 연계된 종합설계 교과를 운영함
- 이론 중심의 교과나 저학년 교과의 경우에 경우에는 해당 교과에서 다루어야 할 내용이 많기 때문에 설계 교육까지 병행하기 힘들
- 시스템프로그래밍, 네트워크프로그래밍, 데이터베이스프로그래밍처럼 실습 중심의 교과는 요소설계 교과로 활용이 가능함. 다만, 종합설계 이전에 이수할 수 있도록 편성되어야 함
- 설계 구성요소 중 시험 및 디버깅, 유지보수 등에 대해 교육하기 위해 고급 소프트웨어공학 교과의 운영도 검토해볼 수 있음

IV. 컴퓨터공학에서 요소설계 교과 운영방법: KoreaTech 사례 중심으로

소속 학부에서 2006년 공학교육인증제 운영 프로그램을 시작한 이래 여러 개의 요소설계 교과를 운영한 경험이 있다. 2010년 컴퓨터공학부로 구조조정된 이후 가장 최근까지 요소설계 교과로 운영한 교과는 2학년 2학기에 편성되어 있는 자바프로그래밍과 4학년 1학기에 편성된 객체지향개발론및실습이다.

본 프로그램에서는 출발부터 기존 수업시간을 고려하여 학기가 총 w주로 구성되어 있고 주마다 h시간의 교육을 하고, 총 학점이 n이며, 설계학점 d(<n)일 때, w\*h\*d/n 시간 이상의 설

계교육이 이루어지면 설계 비중에 맞게 교육한 것으로 내구화하여 운영하고 있다. 설계교육이 이루어지는 시간은 정규 수업 시간에 실제 설계교육을 한 시간과 학생들이 설계과제를 수행하기 위해 투자해야 하는 평균 시간을 합하여 계산한다.

이 장에서는 두 개의 교과 중 자바프로그래밍에서 설계교육을 어떻게 진행하였고, 어떤 개선 과정이 있었는지 설명하고자 한다.

### 1. 자바프로그래밍 교과 개요

자바프로그래밍 교과는 객체지향 프로그래밍에 초점을 두고 있는 기초 프로그래밍 교과로서 2013년 2학기까지는 다음과 같이 요소설계 교과로 운영하였다.

- 편성학기: 2학년 2학기
- 학점: 3-2-2-1 (3학점, 이론 2시간, 실습 2시간, 설계 1학점)
- 교과목 학습성과: 설계능력, 문제해결, 실무능력, 의사소통

### 2. 2011년까지 운영방식

2011년까지는 문제해결 능력, 알고리즘 설계 능력에 초점을 두고, 승강기 시뮬레이션 프로그램을 구현하는 학기 개인 프로젝트를 실시하였다. 이 출발은 전체적인 설계교육 측면에 대한 고려보다는 해당 교과의 교육목표에 초점을 두는 접근이었다. 또한 기초 교과이기 때문에 팀워크에 대한 고려 없이 각 학생들의 개별 역량을 검증하고자 개인프로젝트로 진행하였다.

전체적인 설계 이수체계 측면에서 설계교육을 진행하지 않고 개별 교과의 담당교수가 해당 교과의 교육목표를 고려하여 교육내용을 결정하는 문제는 설계교육만의 문제는 아니다. 전체적인 교과과정, 해당 교과의 개설학년 및 학기에 함께 개설되는 다른 교과들을 고려하여 교육내용과 학습량이 결정되어야 하지만 많은 경우 담당교수의 판단에 의해 결정되는 경우가 보통이다. 특히, 요소설계에서 설계 교육 내용은 해당 교과의 교육내용보다 전체적인 설계 이수체계 측면에서 결정되어야 한다. 그렇지 않을 경우 종합설계 이전에 적절하고 균형 있는 설계교육이 이루어지기 힘들다.

승강기 시뮬레이션 프로그램은 10층 건물에 하나의 버튼에 연동된 2개의 승강기가 있다고 가정하고, 랜덤으로 이벤트를 발생하도록 하여 이벤트에 맞게 승강기를 시뮬레이션하는 프로그램이다. 따라서 설계 구성요소 중 분석 일부와 구현에 초점을 두는 형태가 되었다. 하지만 현실적 제한조건은 승강기 탑승인원 제한 등 공학인증기준에서 말하는 현실적 제한조건이 아닌 문제 정의에 포함되는 제한조건만을 사용하였다. 현재

공학교육인증 평가에서 교과목 포트폴리오를 보면 아직도 이와 같은 문제 정의에 포함되는 제한을 해당 교과에서 고려한 현실적 제한조건이라고 제시하는 경우가 많다.

이와 같이 전체 설계이수 체계측면에서 또한 설계교육에 대한 정확한 이해 없이 설계교육을 하는 것은 다음과 같은 문제점이 있다.

- 첫째, 종합설계 이전에 균형 있는 설계 교육이라는 측면에서 요소설계 교과로 충분한 역할을 하지 못한다. 특히, 다른 교과와 조율을 통해 교과에서 중점을 둘 설계 구성요소나 현실적 제한조건을 결정하지 않고 있다.
- 둘째, 개인 프로젝트를 진행함에 따라 팀워크 능력과 의사소통 능력을 배양하지 못한다.
- 셋째, 이처럼 특정 주제 하나만을 이용하여 진행하면 주제의 다양성이 없어 간접 경험을 통한 교육효과가 크지 않다.

### 3. 2012년 이후 운영방식

2011년까지 운영방식의 문제 때문에 2학년 2학기에 어떤 설계 교육을 하는 것이 바람직할 것인지 고민해 보았다. 대다수 요소설계 교과처럼 해당 교과 관련 학기 프로젝트를 실시할 수 있지만 이것은 설계교육 측면보다 해당 교과에서 달성해야 하는 교과의 교육목표에 초점을 두는 방식이라 적절하지 않다.

2학년 2학기는 설계 이수체계 측면에서 기초입문설계 교과를 이수한 상태이고, 전공교육 측면에서는 기초적 프로그래밍 기술만 배운 상태이다. 따라서 전공심화 지식을 아직 학습하지 않은 상태에서 어떤 설계 경험을 할 수 있도록 하는 것이 효과적인지 검토할 필요가 있다. 설계 구성요소를 단순화하여 문제정의, 분석, 설계, 구현, 테스트로 구분하였을 때, 2학년에게 어떤 설계 구성요소에 대한 교육을 하는 것이 바람직할 것인지에 대한 답으로부터 설계교육에 대한 답을 얻을 수 있다.

저학년의 경우에는 프로그래밍 언어만 배운 상태이며, 자료구조의 개설학기에 따라 자료구조조차도 배우지 않은 상태일 수 있다. 또한 자바나 C++와 같은 객체지향 언어를 보통 1, 2학년 때 배우지만 객체지향 설계까지 충분히 학습하기는 어렵다. 따라서 문제정의부터 테스트까지 어떤 단계도 2학년 때 충실히 해보는 것이 쉽지 않다. 그러나 자세한 이론적 내용을 모르더라도 설계 교육측면에서 어떤 경험을 해보는 것이 효과적인지 생각해 보면 문제정의, 분석 단계는 2학년 때 충분한 이론적 교육 없이도 가능하다는 것을 알 수 있다. 또한 객체지향 언어를 배운 상태라면 해당 프로그램을 구현하기 위해 개발해야 하는 클래스를 도출하는 것은 가능하며, 추가적인 교육을 한다면 이런 클래스들간의 관계나 동적 행위를 시퀀스 다이어그램



으로 도식화하는 것도 가능하다.

이와 같은 고민의 결과로 2012년부터는 다음과 같은 변화를 주었다,

- 팀단위의 자율 프로젝트를 수행함
- 보고서 양식(팀보고서, 개인보고서)을 개발하여 배포함
- 객체지향 설계를 보다 충실하게 할 수 있도록 자바프로그래밍 교육도 객체지향 관련 요소를 후반부에 교육하는 방식 대신에 처음부터 객체지향에 초점을 두는 방식(early object approach)으로 변경(Lister et al., 2006, Johnson et al., 2008)함
- 문제정의, 분석에 초점을 두는 설계 과제를 진행하지만 자바 교과의 교육목표도 고려하여 설계 결과 중 일부만을 자바 스윙 라이브러리를 이용하여 구현하도록 하였으며, 이를 위해 학기 초부터 실습시간에 스윙 라이브러리 실습을 진행함

#### 가. 설계프로젝트 팀구성

처음에는 학생들이 자율적으로 팀을 구성하도록 하였다. 하지만 팀을 자율적으로 구성하도록 할 경우 팀 구성을 못하는 학생들도 있으며 학습능력이 우수한 학생들끼리 또는 그 반대끼리 팀을 구성하는 경우도 많아 교육효과 측면과 평가측면에서 모두 부정적인 결과를 초래하였다. 따라서 후에는 남학생과 여학생을 나누고 고학년과 저학년을 나누어 랜덤하게 팀을 구성하였다.

설계 프로젝트 진행 과정에서 팀별로 총 3차례 발표를 하도록 하였는데 수업시간 내에 모든 팀의 발표를 완료하는 것이 효과적이므로 전체 팀 수는 10팀 이하로 제한하였다. 따라서 수강인원 수에 따라 각 팀별 인원수가 결정되는 형태이다.

#### 나. 설계프로젝트 주제선정

팀 구성 이후 주제를 자율적으로 정하도록 하였다. 처음에는 학생들이 자신들의 구현 능력의 한계를 고려하여 다른 수업의 학기 프로젝트처럼 작은 규모의 주제를 정하였다. 하지만 작은 규모의 주제는 문제 정의, 요구사항 분석이 너무 간단할 수 있어 적절하지 않기 때문에 두 번째 학기부터는 구현이 목적이 아니라 하는 것을 충분히 설명해주고 규모가 비교적 큰 주제를 정하고 요구사항 분석은 상용 수준의 유사 프로그램에서 제공하는 모든 기능을 포함하도록 하였다. 이렇게 주제를 선정한 결과 일부 학생들은 나중에 해당 주제를 종합설계 주제로 활용한 경우도 있었다.

학생이 자율적으로 주제를 선정하지만 주제 선정 결과를 사전에 듣고 동일 또는 유사 주제를 선정한 경우에는 주제를 변경하도록 하여 모든 팀이 다른 주제로 과제를 진행하게 하였다. 이렇게 함으로써 간접경험 효과를 극대화할 수 있다.

#### 다. 설계프로젝트 주차별 진행방법

설계프로젝트의 주차별 진행방법은 다음과 같다.

- 1주 (교과목 소개): 설계 과제 진행 방식을 소개함
- 2주 (팀배정): 팀 배정을 한 후에 5주까지 해야 할 일을 자세히 설명함
- 3주 (주제 확정): 각 팀별 팀장으로부터 주제 선정 결과를 간단히 듣고 필요한 조율을 함
- 2주~4주 (요구사항 분석): 주제 확정 후 팀별로 시스템 요구사항 분석, 사용 시나리오 분석, 사용자 인터페이스 분석을 진행함
- 5주 (1차 발표): 주제선정보고서를 제출받고 이를 바탕으로 1차 발표를 진행함
- 6주: 1차 발표에 대한 피드백을 진행하고, 10주까지 해야 할 일을 자세히 설명함
- 6~9주: 1차 발표에 대한 피드백 결과를 반영하여 데이터 분석을 한 후에, 클래스를 도출하고 정적, 동적 분석을 진행함
- 10주 (2차 발표): 설계보고서를 제출받고 이를 바탕으로 2차 발표가 진행됨
- 11주: 2차 발표에 대한 피드백을 진행하고, 15주까지 해야 할 일을 자세히 설명함
- 11~14주: 2차 발표에 대한 피드백 결과를 반영하여 시스템의 일부를 자바 스윙 라이브러리를 이용하여 구현함
- 15주 (최종 발표): 최종보고서를 제출받고 이를 바탕으로 최종 발표 및 시연이 진행됨
- 16주: 성적과 더불어 최종 피드백을 진행함

주차별 활동 중 몇 가지 보충설명이 필요하다. 첫 주에 팀 배정을 하지 못하는 이유는 첫 주에 수강정정 기간이 있기 때문인데 수강인원이 정확하게 확정된 이후에 배정하여야 변동 없이 진행이 가능하다. 학기가 끝날 때까지 팀별로 총 3차례 발표를 하는데, 각 발표마다 다른 학생이 반드시 발표하도록 하였다. 10주까지는 구현에 대한 고려 없이 상용 수준의 프로그램이 갖추어야 할 모든 기능을 고려하도록 하였다. 또한 각 팀보고서를 제출할 때마다 개인보고서를 제출하도록 하여 개인평가에 반영하였다.

#### 4. 개선효과

이와 같은 설계 교육을 진행하였더니 체계적인 절차에 따라 소프트웨어를 개발하는 과정을 이해하게 되었고, 구현이 소프트웨어 개발에 전부가 아닌 것을 충분히 인식하는 좋은 계기가 되었다. 또한 보고서와 발표자료를 작성하면서 의사소통 능력을 배양할 수 있었으며, 다른 학생들의 발표와 교수의 피드백을 통해 다양한 프로젝트에 대한 간접경험도 얻을 수 있었고,

팀워크 능력도 배양이 되었다.

저학년에서 이와 같은 교육은 고학년 심화 교육에 매우 긍정적인 효과가 있음이 이미 발표된 바가 있다(Liew, 2005). 더욱이 제시된 사례처럼 부분설계 교과에서 이와 같은 교육을 하는 것이 아니라 저학년 순수설계 교과를 개설하여 이와 같은 교육을 하면서 능동학습 형태로 운영하면 교육효과를 더욱 높일 수 있다(Ludi et al., 2005).

## V. 결 론

이 논문에서는 컴퓨터공학에서 어떻게 효과적으로 설계 교육을 진행할 수 있는지 살펴보았다. 컴퓨터공학은 공학인증 기준에서 제시하는 설계 교육을 적용하기 어렵다고 생각하는 경우가 많지만 살펴본 바와 같이 오히려 가장 체계적이며, 효과적으로 할 수 있는 학문분야 중 하나이다. 이 논문에서 살펴본 컴퓨터공학 관련 설계 교육의 특징을 요약하면 다음과 같다.

- 컴퓨터공학에서 설계 구성요소는 소프트웨어공학에서 정의되는 개발 단계를 이용하여 재정의하여 사용하면 된다. 특히, 최근에는 오픈 마켓의 활성화로 다른 학문분야와 달리 배포, 운영, 유지보수까지 교육과정 내에서 경험해 볼 수 있다.
- 컴퓨터공학에서 현실적 제한조건은 몇 가지 제한조건을 제외하고는 충분히 활용할 수 있다.
- 컴퓨터공학에는 소프트웨어공학이라는 설계와 관련된 세부 학문이 있다. 이 교과는 심지어 순수설계 교과로 운영할 수 있는 교과이다.
- 컴퓨터공학에만 적용되는 것은 아니지만 기존 전공교과를 요소설계 교과로 운영하기 보다는 각 학년별로 학년에 맞는 순수설계 교과를 운영하는 것이 보다 효과적이다.
- 요소설계 교과에서 어떤 설계 교육을 할 것인지는 해당 교과의 특성보다는 전체 설계교육 이수체계 측면에서 결정되어야 한다.
- 각 학년별 순수설계 교과를 운영할 때 2학년의 경우에는 문제정의, 분석, 설계에 초점을 두는 설계를 진행하면 효과적일 수 있다.

향후에는 여러 대학에서 시도하고 있는 학년별 요소설계 교과의 운영현황을 보다 심도 있게 분석하여, 각 학년에 맞는 효과적인 요소설계 교과의 교육내용과 방법을 제시하고자 한다.

이 논문은 2014년도 한국기술교육대학교 교수교육연구진흥비 지원에 의하여 연구되었음

## 참고문헌

1. Dym, C. L. et al.(2005). *Engineering Design Thinking, Teaching, and Learning*, Journal of Engineering Education, 94(1) : 103-120.
2. 한국공학교육인증원(2012a), 공학교육인증기준 2005 (2012. 7. 개정) (KEC2005).
3. 한국공학교육인증원(2012b), 컴퓨터·정보기술인증기준2010 (KCC2010).
4. 한국공학교육인증원(2013a), 2014년 평가용 KEC2005 인증평가 판정가이드.
5. 한국공학교육인증원(2013b), 2014년 평가용 교육기관 자체평가보고서(KEC2005) 작성 양식.
6. 한국공학교육인증원(2014a), 공학교육인증기준2015 (KEC2015).
7. 한국공학교육인증원(2014b), 2015년 평가용 KEC2015 인증평가 판정가이드.
8. 한국공학교육인증원(2014c), 2015년 평가용 교육기관 자체평가보고서(KEC2015) 작성 양식.
9. 한국공학교육인증원(2014d), 컴퓨터·정보기술인증기준2015 (KCC2015).
10. Winston, R.(1970), Managing the Development of Large Software Systems. Proc. of IEEE WESCON, 26 : 1-9.
11. Martin, R. C.(2003), Agile Software Development: Principles, Patterns, and Practices. Prentice Hall.
12. Lister, R. et al.(2006), Research Perspectives on the Objects-early Debate, Proc. of ITiCSE, 146-165.
13. Johnson, R.A. et al.(2008), Object-first vs. Structures-first approaches to OO Programming Education: An Empirical Study, Proc. of the Allied Academies Int'l Conf., 15(2) : 244-248.
14. Liew, C. W.(2005), Teaching Software Development Skills Early in the Curriculum Through Software Engineering, Proc. of the ITiCSE'05, 133-137.
15. Ludi, S. et al.(2005), An Introductory Software Engineering Course that Facilitates Active Learning, Proc. of the SIGCSE'05, 302-396.



**김상진 (Kim, Sangjin)**

1995년: 한양대학교 전자계산학과 공학사  
 1997년: 동 대학원 전자계산학과 공학석사  
 2002년: 동 대학원 전자계산학과 공학박사  
 2003년-현재: 한국기술교육대학교 컴퓨터공학부 교수  
 관심분야: 공학인증, 교수학습법

Phone: 041-560-1490  
 Fax: 041-560-1462  
 E-mail: sangjin@koreatech.ac.kr