

# 소프트웨어 개발 프로젝트의 위험요인 도출에 대한 델파이 연구

정 경 수\* · 강 명 희\*\* · 김 용\*\*\*

## <목 차>

I. 서론	3.3 조작적 정의
II. 이론적 배경	IV. 연구 결과
2.1 프로젝트 관리	4.1 표본의 기술적 특성
2.2 소프트웨어 개발 프로젝트 관리 위험 중심적 접근법	4.2 연구 결과
2.3 소프트웨어 개발 프로젝트 위험 요인에 관한 선행연구	V. 결론
III. 연구 설계 및 조사 방법	5.1 연구의 요약
3.1 델파이 방법	5.2 연구의 한계점 및 향후 연구 과제
3.2 자료 수집과 분석 방법	참고문헌
	Abstract

## I. 서론

오늘날 기업들은 전 세계를 무대로 하는 글로벌 시장에서 효과적으로 경쟁하고 품질, 생산성 및 고객 만족도를 높이기 위해 사업 관행을 바꾸고 경영혁신을 시도하고 있다. 기업경영에 있어 모든 레벨의 업무수행이 사업목적과 목표를 달성하는 능력을 기반으로 재평가되고 있는 중이며, 이러한 과정에서 정보시스템의 역할은 그 어느 때 보다도 중요하게 부각되고 있다.

하지만, 시스템 개발 방법과 도구의 향상에도 불구하고 빈번한 요구 사항 변경, 인력 관리와 관리 감독의 변동, 개발 인력의 빈번한 교체와 미흡한 통제 및 관리 등으로 인하여 기업들이 추진하고 있는 프로젝트들 중 다수가 일정이 지연되거나 비용을 초과하게 되고 완성 이전에 프로젝트 자체가 포기되기도 한다.

\* 경북대학교 경영학부, kschung@knu.ac.kr

\*\* 롯데정보통신(롯데호텔), angelliza@empal.com

\*\*\* 경북대학교 대학원, idealplace@naver.com

특히, 소프트웨어의 개발에 있어서는 산출물이 무형이고, 소프트웨어 개발 과정에 대한 이해가 쉽지 않다. 또한 대형 소프트웨어 개발의 경우 한번 사용되고 끝나는 프로젝트인 경향이 있다는 등의 이유로 예산의 범위를 초과하고 계획이 지연되는 경우가 많다(이경환, 1992). 소프트웨어 개발 프로젝트를 성공시키기 위한 요인으로는 효과적인 프로젝트 관리의 실행, 개발 표준안과 정책의 수립, 효과가 입증된 개발 패턴 사용 등을 들 수 있다(조영재 & 김석수, 2000).

오늘날 대부분의 시스템이 대형화, 복잡화 되어가고 있는 추세이며, 시스템이 복잡해질수록 위험 관리의 필요성은 증가하고 있다(Higuera & Haimes, 1996). 프로젝트가 실패하는 원인 중 하나는 프로젝트 진행 중에 나타날 수 있는 위험에 대한 고려를 소홀히 했기 때문이며, 성공적인 프로젝트의 관리를 위해서는 프로젝트와 관련된 위험을 확인하고 분석하여 프로젝트가 완수될 때까지 위험을 지속적으로 관리해야 한다.

프로젝트 위험 관리를 위해서 위험 요인의 파악이 가장 먼저 선행되어야 할 요소이지만, 프로젝트의 위험요인과 관련된 국내 연구는 부족한 실정이며, 기존의 위험 요인에 관한 연구는 시기적, 지리적, 문화적 차이 등으로 인하여 현재 국내의 상황과는 차이가 있는 것으로 보인다.

본 연구에서는 국내의 소프트웨어 개발 프로젝트 전문가를 대상으로 프로젝트 위험 요인을 찾아내고 이들을 체계적으로 정리하여 조직 문화와 국가 문화의 차이 등 국내 환경을 반영한 프로젝트 위험 요인을 찾고자 하였다. 또한 델파이 기법을 사용하여 20개의 프로젝트 위험 요인에 대한 우선순위를 도출한 후에 이를 해외 선행 연구와 비교하여 제시하였다.

## II. 이론적 배경

### 2.1 프로젝트 관리

프로젝트란 제품이나 서비스를 창출하기 위해 수행하는 한시적인 활동으로, 예산 충족, 기간 내 완수, 사용자 요구사항 만족의 세 가지 상호 관련된 목표를 가지며 (Mantel et al., 2001), 프로젝트 관리는 프로젝트를 체계적이고 경제적인 방법으로 완성하고 시간, 비용, 기술 또는 서비스 등의 정해진 목적을 충족시키기 위해 자원, 인력, 비용을 통제하고 관리하는 활동을 말한다 (Spinner, 1997).

소프트웨어 개발 프로젝트란 계획된 시간과 예산 내에 사용자의 요구사항에 맞는 소프트웨어를 개발하는데 필요한 기술 및 관리 활동을 말한다. 이는 계획, 조직화, 인력 관리, 감시, 통제 그리고 프로젝트를 이끄는 통솔력 등을 포함하며, 훌륭한 소프트웨어 프로젝트 관리자가 되기 위해서는 훌륭한 소프트웨어 개발자로는 충분치 않다. 기술적인 능력 이외에 프로젝트의 초기 단계부터 리더십과 계획, 고객과의 관계, 기술적 리더십 등의 특정한 관리적 능력이 요구된다.

적절한 프로젝트 관리가 이루어지지 않으면 계획된 비용을 초과하게 되고 완성된 이후에도 유지 보수가 많이 필요한 소프트웨어를 생산할 가능성이 커지게 된다. 그러므로 소프트웨어를 개

발할 때는 첫째, 정해진 일정에 맞게 개발하고, 둘째, 계획된 예산 내에서 개발해야 하며, 셋째, 사용자 요구사항을 만족시키는 제품 개발의 세 가지 요건을 반드시 고려해야 한다(Bennatan, 2000).

## 2.2 소프트웨어 개발 프로젝트 관리의 위험 중심적 접근법

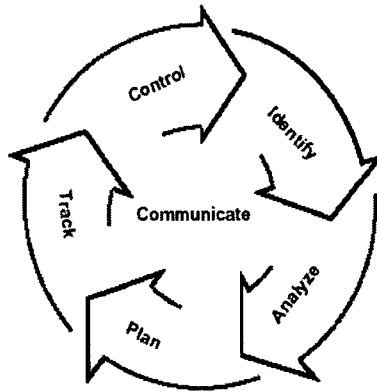
### 2.2.1 소프트웨어 개발 프로젝트의 위험 관리

위험 관리와 프로젝트 관리는 완전히 같은 것을 의미하지는 않으며 정보 획득이나 의사 결정에 있어 서로 밀접하게 얽혀있다. 프로젝트 관리의 목적은 납기와 예산, 품질과 관련된 목표를 달성하는 것이다. 프로젝트가 성공적일 때, 그것은 문제가 없었기 때문이 아니라 문제가 성공적으로 극복되었기 때문이다. 위험 관리가 성공을 보장하지는 않지만, 문제 상황을 피하기 위해 충분한 리드 타임을 가지고 잠재적인 문제를 확인하고 대응하는 최우선 목표를 가짐으로써 프로젝트 관리가 그 목표를 이룰 수 있도록 한다.

Higuera & Haimes(1996)에 의하면 위험을 관리해야 하는 필요성은 시스템의 복잡성에 따라 증가하며, 개인적 지식, 판단, 그리고 경험을 보충하기 위해서 체계적인 방법과 도구의 필요성이 계속적으로 증가하고 있다고 한다. 소프트웨어 개발 프로젝트에서의 위험은 예산 혹은 기한 초과, 사용자 요구사항 불만족 등의 만족스럽지 못한 결과가 나올 가능성과 그러한 결과 발생 시 입게 되는 손실의 두 가지 차원으로 정의되며, 프로젝트 위험 요인은 프로젝트의 성공에 위협이 되는 것, 즉 프로젝트 위험을 발생시키는 원인이 되는 것으로 정의할 수 있다(Hall, 1998).

Bennatan(2000)에 따르면 위험 관리란 정보시스템 개발 프로젝트 수행 시 발생할 수 있는 문제점의 발생확률과 영향정도를 예측하여 보다 나은 해결책을 제시하기 위해 수행되는 활동을 말하며, 프로젝트 관리에 있어서 위험을 가장 중심적인 것으로 간주하고, 이를 통제함으로써 프로젝트의 성과를 높일 수 있다는 것이 위험중심적 접근법이라 할 수 있다(Hall, 1998). 미국 SEI(Software Engineering Institute)는 소프트웨어 개발 위험을 관리하기 위한 체계적인 방법이 필요하며, 이러한 체계적인 방법이 소프트웨어 제품의 품질과 비용, 일정을 통제할 수 있다는 가정 하에 <그림 1>과 같은 소프트웨어 개발 위험 관리 모형을 제시하였다.

위험관리 모형은 위험 관리가 지속적인 과정이라는 것을 강조하기 위해 원으로 표현되었고, 화살표는 위험 관리 활동들간의 논리적이고 일시적인 정보 흐름을 나타낸다. 그리고 의사소통(Communicate)은 모든 정보가 흘러가는 것을 이어주는 동시에, 위험 관리에 있어 가장 큰 장애물이기도 하기 때문에 중간에 위치한다. 시스템이 복잡해질수록, 프로젝트가 커질수록, 위험 관리의 필요성은 증대되며, SEI에서 제시한 성공적인 위험 관리란 다음과 같다(Bennatan, 2000).



<그림 1> SEI의 위험관리체계(Bennatan, 2000)

“성공적인 위험 관리란 프로젝트 자원의 효과적인 사용과 위험의 감소를 위해서 상대적인 중요도에 따라 위험의 지속적인 분석과 통제를 통하여 문제의 발생을 미연에 방지하고 제품의 품질과 생산 일정에 긍정적인 영향을 미치는 것을 말한다.”

## 2.2.2 위험중심적 접근법의 이론적 배경

위험 관리가 현재 소프트웨어 프로젝트에 사용되고 있는 데는 여러 가지 이유가 있다. 급격히 변화하는 환경과 복잡한 시스템의 설계를 위한 요구사항의 증가 등으로 인하여 프로젝트의 불확실성이 증가하고 있다. 또한 이윤이 축소되고 경쟁이 심한 사업 환경, 국제 경제와 불확실한 시장 상황, 그리고 급격한 기술적 진보에 의한 경쟁 압력 등 모두가 그 이유라고 할 수 있다.

위험 관리 기법은 1980년대 소프트웨어 위험 관리의 아버지로 일컬어지는 Boehm이 반복적이고 위험에 초점을 둔 소프트웨어 수명 주기 모델인 나선형 모형을 제시함으로써 소개되었다. McFarlan(1981)의 연구에서는 프로젝트에서 위험을 따로 분리하여 평가하는 것이 프로젝트 관리자가 더 나은 의사결정을 할 수 있게 하고 더욱 성공적인 결과를 확신할 수 있게 한다고 주장했다. Boehm(1988)은 위험평가단계를 포함하는 나선형 모형을 제시함으로써 위험 관리 기법을 소개했고 위험 요인의 확인과 적절한 대응책을 마련하는 것을 포함하는 다섯 단계로 이루어진 위험 관리 계획을 제시하였다. 또한 그는 복잡한 응용시스템을 개발할 때 개발 초기에 위험을 확인하고 다루는 것이 장기적인 비용을 절감하고 소프트웨어 재앙을 방지하는데 도움을 준다고 주장하였으며, 위험의 확인, 위험 분석, 위험 평가, 위험 관리 계획, 위험 해결, 위험 감시의 여섯 단계로 이루어지는 소프트웨어 위험 관리 단계를 제시하였다(Boehm, 1991).

Fairley(1994)는 일반적인 프로젝트 수행에 있어서 위험 관리가 명확한 프로젝트 관리 활동으로 적용되지 않는 이유는 실제적이고 단계적인 접근법을 제공하는 지침이 없기 때문이라고 하였으며, 여러 가지 유형의 소프트웨어 프로젝트에 적용될 수 있는 위험 관리를 위한 일곱 단계의 프로세스를 제안했다. Baskerville(1996)은 시스템 개발자와 사용자 사이의 커뮤니케이션을 강조

하는 프로토타이핑의 가치를 향상시키기 위해 위험 경감 모형과 위험 관리 프로세스를 사용한 접근법을 제시하였다. 또한 그는 실험연구를 통해 이러한 위험 분석 접근법이 프로토타이핑 환경에서도 일관성 있는 과정을 제공하고 우선 순위 결정에도 유효함을 검증하였다.

Barki et al.(2001)의 연구에서는 소프트웨어 프로젝트 위험 관리와 조직이론의 상황이론을 결합하여 소프트웨어 프로젝트 위험 관리의 통합적 상황 모델을 개발하였다. 그들은 이러한 모형을 프로젝트 관리자와 최종사용자들을 대상으로 조사한 결과 제안된 상황 모형이 지지되었으며 프로젝트 성과를 향상시키기 위해 프로젝트 위험에 따라 프로젝트 관리가 변화될 필요가 있다는 것을 주장하였다.

## 2.3 소프트웨어 개발 프로젝트 위험 요인에 관한 선행연구

소프트웨어 개발 프로젝트에서 잠재적 위험은 예측할 수 없기 때문에 프로젝트 기간 전체에 걸쳐 분석되어야 하는데 여러 가지 이유로 인하여 프로젝트와 관련된 위험들을 발견하는 것이 점차 어려워지고 있다. 그 이유 중 하나는 산업 환경의 변화로 인하여 상대적으로 적은 경험을 가진 사람들에게 책임과 권한이 이양되고 이들에 의해서 보다 복잡한 시스템의 개발이 수행되고 있기 때문이다(Grey, 1995).

해당 프로젝트에서 효과적인 위험 관리를 수행하기 위해서는 우선 위험을 확인하여 관리 대상이 되는 요인들을 체계적으로 정의하고 해당 프로젝트에 영향을 미칠 것으로 예상되는 요인들을 추출하여야 한다. 소프트웨어 공학 분야에서 이루어진 프로젝트 위험 요인에 대한 연구는 다음과 같다.

McFarlan(1981)의 사례연구에서는 위험 요인을 세 가지로 제시하였다. 첫째, 프로젝트 수행을 위한 인력 수준, 경과된 시간, 참여 부서의 수 등 프로젝트 크기가 커질수록 위험이 증가하게 된다. 둘째, 프로젝트팀과 IS 조직이 하드웨어와 운영 시스템, 데이터베이스, 프로젝트 응용 언어에 대해 잘 알지 못하는 것과 같이 기술적인 문제가 증가 할수록 위험이 커진다. 셋째, 개념화 단계에서 산출물을 완벽하게 정의할 수 없는 업무를 다룰 때 프로젝트 위험은 확대된다.

Boehm & Ross(1988)는 프로젝트 위험을 모든 프로젝트에 공통되는 일반적인 위험과 해당 프로젝트 고유의 특성에 의해 발생하는 위험으로 분류하였다. 일반적인 프로젝트 위험으로 인력 부족, 비현실적인 일정과 예산, 부적절한 요구사항, 외부 구성요소와 업무 지식의 부족, 기술적 부족 등을 제시하였다.

Barki et al.(1993)은 프로젝트 개발 위험을 프로젝트와 관련된 불확실성과 프로젝트 실패에 따른 잠재적인 손실의 크기로 정의하였으며 문헌연구를 통해 기술, 어플리케이션의 크기, 전문성, 어플리케이션의 복잡성, 조직환경 등 다섯 가지 위험 영역을 도출하였다. 또한 선행연구들을 바탕으로 35개의 위험 변수를 도출하고 이러한 변수들을 구성하는 질문을 개발하였다.

Moynihan(1997)는 프로젝트 위험 관리 문헌들의 현실성을 검증하기 위해서 아일랜드의 시스템 개발자 14명을 대상으로 개인 구성 개념 추출법을 사용하여 조사를 실시하였다. 조사 결과 21개의 위험 변수들이 도출되었고 이것을 Barki et al.(1993)과 Carr et al.(1993)의 위험요소와 비교

하여 고객 관련 위험이 기존의 연구에서 소홀히 다루어지고 있음을 지적하였다.

Keil et al.(1998)은 소프트웨어 개발 프로젝트의 높은 실패율은 관리자들의 프로젝트 위험의 관리 소홀에 기인하는 것으로 간주하였다. 그리고 핀란드, 홍콩, 미국의 소프트웨어 개발 프로젝트 관리자들을 대상으로 델파이 분석을 실시하여 주요 위험 요인을 도출하였으며 이를 상황 적합적 관점에서 범주화하였다. Schmidt et al.(2001)은 신뢰할 수 있는 프로젝트 개발 위험 요인 리스트를 개발하기 위해서 홍콩, 핀란드, 미국의 소프트웨어 개발 프로젝트 관리자들을 대상으로 델파이 분석을 실시하여 위험 요인들을 밝혀내었다.

서창교 & 정은희(2003)는 프로젝트 성과에 영향을 미치는 요인을 위험관리 요인과 프로젝트 위험요인으로 구분하여 실증연구를 행하였다. 그 결과 위험관리 요인 중에서 사용자 참여, 개발 팀이 성과에 유의한 영향을 미쳤으며 프로젝트 위험 요인 중에서는 기술적인 위험, 요구사항 위험, 조직 환경 위험 모두가 성과에 부정적으로 영향을 미치는 것으로 나타났다. 이상의 연구결과를 요약하면 <표 1>과 같다.

<표 1> 프로젝트 위험 요인에 관한 선행연구

연구자	연구 방법	연구 내용
McFarlan (1981)	사례조사	3가지 위험 영역 도출 (프로젝트 크기, 기술 경험, 프로젝트 구조)
Boehm (1991)	설문조사	10개의 위험 요인 도출 (예: 잘못된 기능과 특성의 개발, 계속되는 요구사항 변화 등)
Barki et al. (1993)	문헌조사	5가지 위험 영역 도출 (복잡성, 크기, 기술, 전문성, 조직환경)
Moynihan (1997)	개인 구성개념 추출법	21개의 위험요인 도출
Keil et al. (1998)	델파이 분석	11가지 위험 요인을 도출하고 상황 적합적 관점에서 범주화
Schmidt et al. (2001)	델파이 분석	홍콩, 핀란드, 미국의 관리자들 대상으로 위험 요인 도출
서창교 & 정은희 (2003)	설문조사	3가지 위험요인 도출 (기술위험, 요구사항위험, 조직환경위험)

### Ⅲ. 연구 설계 및 조사 방법

#### 3.1 델파이 방법

델파이(Delphi) 기법은 전문 토론을 거치지 않고 구성원들로부터 전문적인 견해를 얻어내는 방법으로, 예측하려는 문제에 관하여 전문가들의 견해를 유도하고 종합하여 정리하는 일련의 절

차라고 할 수 있다. 이 방법은 미국의 랜드연구소(Rand Corporation)에서 개발하였으며 대면토의에서 나타나는 한계점들을 제거하고 긴급한 국방문제에 관하여 전문가들의 합의를 도출하기 위해서 1950년대 최초로 사용된 이래로 여러 분야에서 미래를 예측하는 문제뿐만 아니라 연구방법으로 사용되었다.

델파이 방법(Delphi method)은 ‘추정하려는 문제에 관한 정확한 정보가 없을 때는 두 사람의 의견이 한 사람의 의견보다 정확하다’라는 계량적 객관의 원리와 ‘다수의 판단이 소수의 판단보다 정확하다’는 민주적 의사 결정의 원리에 논리적 근거를 두고 있다(이종성, 2001). 델파이 방법은 이슈의 상대적 중요도에 대한 그룹의 합의를 형성하기 위해서 사회사업과 교육, 경영 과학 그리고 정보시스템과 같은 다양한 분야에 사용되어 왔다 (Schmidt, 1997). 델파이 방법이 정보시스템 연구 분야에서 사용된 예들을 구체적으로 살펴보면 다음과 같다. Dickson et al.(1984)과 Niederman et al.(1991)은 1980년대와 1990년대 정보시스템 관리에 관한 주요 문제점들을 밝혀내기 위해서 델파이 기법을 사용했으며 최근 King et al.(2002)은 지식관리시스템 구축 시 시스템 설계자들과 관리자들과 마주치는 문제점들을 확인하고 이에 대한 우선순위를 부여하기 위해 델파이 기법을 사용하였다.

프로젝트의 성공적인 수행을 위해서는 잠재된 위험을 잘 관리해야 하며 이것을 위한 첫 번째 단계가 관련된 위험 요인들을 탐색하는 것이다. 이것을 위해서 본 논문에서는 위험 요인의 발견을 통해 신뢰성 있는 리스트를 개발하고 어떠한 위험 요소들이 중요한가를 밝혀내고자 하였다. 올바른 연구의 수행을 위해서는 소프트웨어 개발 프로젝트에 관한 신뢰할 수 있는 정보가 있어야 하기 때문에 현업에서 소프트웨어 개발 프로젝트를 다년간 관리한 경험이 있는 전문가 집단을 연구대상으로 하였다. 그리고 이들로부터 신뢰성 있고 타당성 있는 자료를 수집하기 위해서 다양한 의견을 받아들일 수 있도록 개방형 질의를 하고 피드백 기반의 의견 조합을 추구하였다. 또한 소프트웨어 개발에 실제로 포함되는 요인의 도출, 통제된 피드백의 반복을 통해 전문가들의 견해를 도출할 수 특징을 지니는 순위 형식 델파이 방법(Schmidt, 1997)을 본 연구의 연구 방법으로 선택하였다.

### 3.2 자료 수집과 분석 방법

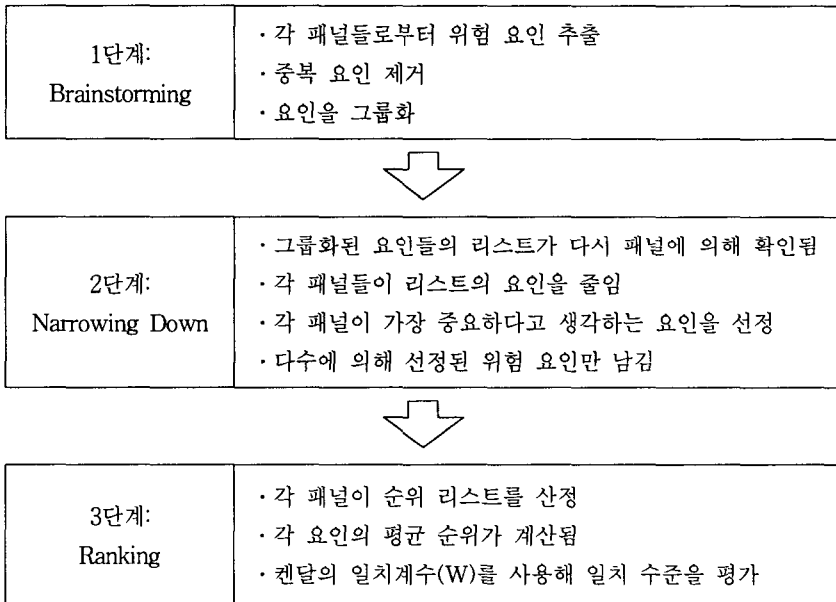
자료 수집과 분석을 위한 델파이 분석 과정은 Schmidt et al.(2001)의 연구에서 행해진 방법에 기초하여 <그림 2>와 같이 세 단계로 나누어 행해졌다.

첫 번째는 브레인 스토밍 단계로 패널들로부터 가능한 많은 항목을 추출하기 위해 시행되어졌다. 전문가들로부터 한국적인 상황에 적합한 자료를 추출하고 고착화 현상을 배제하기 위해 백지상태로 전문가들에게 각자 담당하고 있는 문제영역에 대해 응답하도록 하였다. 각 패널들에게 최소한 여섯 가지 이상의 요인을 제출하도록 요구하였으며, 이 때 패널들이 같은 항목을 다른 용어를 사용해서 말할 수 있기 때문에 패널들에게 각 요인들에 대해 짧은 설명을 하도록 하였다. 다음으로 각 패널들이 제출한 항목들을 분류하고 정리하여 전반적으로 간결한 항목으로 만들어 두 번째 단계의 설문에 사용할 수 있게 하였다.

두 번째 단계는 리스트의 항목을 줄이는 단계이다. 첫 번째 단계에서 만들어진 통합 리스트가 각 패널들에게 다시 보내지고 각 패널들은 리스트에 있는 항목 중 가장 중요하다고 생각하는 요인을 최소 10개에서 20개까지 응답하도록 하였다.

20개의 항목을 가지는 리스트를 만드는 것을 목표로 하며 높은 빈도를 가지는 항목으로 이루어지는 리스트를 만들었다. 본 연구에서는 두 개의 20위 요인이 등장하여 모두 21개의 항목을 포함하는 리스트가 만들어졌다.

세 번째 단계는 선택된 요인들의 순위를 매기는 것으로, 각 패널들은 두 번째 단계에서 만들어진 리스트의 항목에 대해 중요도 순으로 순위를 매기게 된다. 각 패널들에게 프로젝트의 성공적인 수행을 위한 상대적인 중요도에 따라 위험 요인들에 대해 순위를 매기도록 하였으며, 순위가 매겨진 요인 리스트를 받아 평균 순위에 따라 위험 요인 중요도에 따른 리스트를 도출하였다. 또한, 본 연구에서는 켄달의 일치계수 (Kendall's Coefficient of Concordance: W)를 사용하여 패널 사이의 의견의 일치도가 어느 정도인지 분석하였다.



<그림 2> 분석 절차

양훈모(1991)의 연구에 따르면 델파이 기법에서 집단 규모는 그 특성이 동질적인 경우 10명에서 15명이면 족하며, 집단 특성이 독립변수일 때는 수백 명에 이르기까지 다양하게 조사할 수도 있다고 하였다. 본 연구에서는 집단의 특성이 소프트웨어 개발 프로젝트 관리 경험자라는 동일한 특성을 가지고 있으므로 30명 정도를 선택하여 연구를 수행하였다. 그러나 1차 조사에서 21명이 응답하였으며 2차와 3차에서는 18명만이 참여하였다. 각 단계별 조사 기간과 응답자 수는 <표 2>에 나타나 있는 바와 같다.



<표 2> 조사기간 및 각 단계별 응답자 수

	조사기간	응답자수
제 1차 조사	2002년 8월 5일 ~ 2002년 9월 20일	21명
제 2차 조사	2002년 10월 4일 ~ 2002년 10월 10일	18명
제 3차 조사	2002년 10월 14일 ~ 2002년 10월 24일	18명

본 연구는 분석을 위하여 SPSS 10.0 Window용 프로그램을 사용하였으며 사용한 분석 방법은 평균 순위와 의견의 일치도를 알아보기 위한 켄달의 일치계수이다.

### 3.3 조작적 정의

본 연구는 국내 소프트웨어 개발 프로젝트 위험 요인을 도출하는 것으로 프로젝트 위험과 위험 요인에 관한 정의가 필요하다. 본 연구에서 프로젝트 위험은 Barki et al.(1993)의 정의에 따라 예산 혹은 기한 초과, 사용자 요구사항 불만족 등의 만족스럽지 못한 결과가 나올 가능성과 그러한 결과 발생 시 입게 되는 손실의 두 가지 차원으로 정의하며, 프로젝트 위험 요인은 프로젝트의 성공에 위협이 되는 것, 즉 프로젝트 위험을 발생시키는 원인이 되는 것으로 정의하였다.

## IV. 연구 결과

### 4.1 표본의 기술적 특성

본 연구에서는 프로젝트 경력 5년 이상이며 프로젝트 관리 경험이 있는 사람을 대상으로 설문을 실시하였으며, 최종 응답자 18명의 프로젝트 경력 기간은 <표 3>과 같으며 응답자의 평균 프로젝트 경력 기간은 9년 6개월로 나타났다.

<표 3> 프로젝트 경력 기간

프로젝트 경력	빈도(명)	비율(%)
5~10년	11	61.1
10~15년	5	27.8
15년 이상	2	11.1
합 계	18	100.0
평균 프로젝트 경력 기간		9.6년

전체 표본 중에서 소프트웨어 개발 기업의 사업유형은 시스템 통합 사업, 소프트웨어 수탁 개발 사업, 패키지 소프트웨어 개발 공급사업, 소프트웨어 관련 서비스 사업, 기타 등의 순으로 나타났고, 개발 프로젝트 유형은 기업업무용 소프트웨어가 가장 많았으며 기타 응용 소프트웨어, 행정 업무용, 교육용 소프트웨어 순으로 나타났다.

응답자의 일반적 특성을 보면 <표 4>에서 보는 바와 같이 직위에 있어서는 이사 1명, 부장 4명, 차장 4명, 과장 4명, 팀장 3명, 기타 2명이었고 연령은 20대가 5.6%, 30대가 72.2%, 40대 이상이 22.2%로 나타났다.

<표 4> 표본의 특성

구 분		빈도(명)	비율(%)
직 위	이사	1	5.6
	부장	4	22.2
	차장	4	22.2
	과장	4	22.2
	팀장	3	16.6
	기타	2	11.2
연 령	20대	1	5.6
	30대	13	72.2
	40대 이상	4	22.2

## 4.2 연구 결과

### 4.2.1 위험 요인 리스트

1단계에서 확인된 위험 요인의 리스트는 <표 5>와 같다. 분류 기준은 Schmidt et al.(2001)의 위험 요인 분류를 참고하여 14개의 그룹으로 구성하였으며, 총 45개의 위험 요인으로 정리하였다.

이를 기존의 위험요인 도출 연구인 Boehm(1991)과 Keil et al.(1998), Schmidt et al.(2001)의 위험 요인과 비교했을 때, 프로젝트 비용이나 기간, 상위관리자의 관심과 지원, 사용자의 협조, 프로젝트 관리 기술, 팀원의 기술 등과 같은 일반적인 항목은 거의 유사하게 포함되었지만, 새로운 위험 요인으로 다음과 같은 것들이 포함되었다.

첫 번째는 기업 환경과 관련된 내용으로 중소기업의 비도덕적인 프로젝트 수행과 무조건 저렴한 가격에 개발하려 하는 IT산업의 부적절한 투자, 그리고 좁은 국내 시장에서 다수 업체의 무리한 경쟁으로 인한 비현실적인 제안서의 초래, 무리한 사업 영역의 설정 등이다. 두 번째는 요구사항

과 관련된 것으로 비효율적인 요구사항관리 관행, 프로젝트 팀원의 구축할 시스템 및 대상 업무에 대한 이해 부족, 그리고 고객의 무리한 요구 등이다. 이 밖에도 개발 단계별 의사결정 지연으로 인한 일정 지연이나 업무 절차 변경의 어려움, 정확한 예측 부족으로 인한 잘못된 과제수행계획 등이 새롭게 등장한 위험 요인들이다.

<표 5> 첫 번째 단계 위험 요인 리스트

그룹	항목
기업 환경	수주 후 지원(예: 인력, 자금 등)소홀 등의 비도덕적인 프로젝트 수행
	새로운 비즈니스 프로세스와 기존 프로세스와의 부조화
	IT 부분의 부적절한 투자(예: 가격 위주의 개발)
	다수 업체의 무리한 경쟁으로 인한 비현실적인 제안서 초래
	기업 능력을 무시한 무리한 사업 영역의 설정
지원	문제 발생 시 적절한 지원 체제 및 관련 조직의 지원 부족
	사용자의 관심과 준비 및 지원 부족
	테스트 환경 지원 부족
	상위 관리자의 지속적인 검토와 지원 부족
관계 관리	사용자로부터의 협력 부족(예: 사용자가 테스트 환경 지원을 거절함)
	사용자의 적절한 참여 부족
	고객과 프로젝트팀간의 커뮤니케이션 부족
프로젝트 관리	프로젝트 관리자의 업무 능력 부족
	변화 관리가 적절히 이루어지지 않음
	효과적인 프로젝트 관리 기술의 부족(예: 형상관리와 효율성 분석 등)
	위험 관리가 제대로 이루어지지 않음
	팀원, 외주 업체 및 외부 컨설턴트의 명확하지 않은 책임과 역할의 정의 및 분배
	효과적인 프로젝트 수행 방법론의 부재
범위	개발 단계별 필요한 의사결정 사항의 지연
	범위와 목표의 변경
	범위관리 실패
요구 사항	명확하지 않은 업무 범위
	비효율적인 요구사항관리 관행
	사용자 요구사항 변동 및 불명확한 요구사항으로 인한 확정된 요구사항 부족
	프로젝트 팀원간의 구축할 시스템 및 대상 업무에 대한 이해 부족
자금	고객의 무리한 요구
일정 관리	무리한 프로젝트 비용의 산정
	무리한 프로젝트 기간의 산정

### 4.2.2 위험 요인의 순위

3차에 걸쳐 조사된 위험 요인의 순위가 <표 6>에 나타나 있다. 국내 소프트웨어 개발 프로젝트에서 가장 중요한 위험 요인으로는 무리한 프로젝트 비용의 산정, 범위와 목표의 변경, 사용자의 요구 사항 변동 및 불명확한 요구사항으로 확정된 요구사항의 부족, 명확치 않은 업무 범위, 무리한 프로젝트 기간의 산정 순으로 나타났다.

<표 6> 위험 요인 최종 순위

순위	위험 요인 항목
1	무리한 프로젝트 비용의 산정
2	범위와 목표의 변경
3	사용자의 요구사항 변동 및 불명확한 요구사항으로 확정된 요구사항 부족
4	명확치 않은 업무 범위
5	무리한 프로젝트 기간의 산정
6	비효율적인 요구사항관리 관행
7	고객의 무리한 요구
8	다수 업체의 무리한 경쟁으로 비현실적인 제안서 초래
9	정확한 예측 부족으로 인한 잘못된 과제수행계획
10	프로젝트 관리자의 업무 능력 부족
11	고객과 프로젝트팀간의 커뮤니케이션 부족
12	사용자로부터의 협조 부족
12	의사결정 사항의 지연
14	능력 있는 개발자 확보 실패 및 부적당한 인력 배치
15	위험관리가 제대로 이루어지지 않음
16	적절한 지원 체제의 부재 및 관련 조직의 지원 부족
17	효과적인 개발 프로세스/방법론 및 절차와 기준의 부재
18	IT산업의 투자가 적절히 이루어지지 않음
19	수주기업의 비도덕적인 프로젝트 수행
20	가능한 전문 인력의 부재

특히, 본 연구의 결과로 나타난 주요프로젝트 위험 요인 20가지 항목은 거의 대부분이 관리적인 측면과 관련된 것으로 나타났다. 이는 기존의 Boehm(1991), Keil et al.(1998)과 Schmidt et al.(2001)등의 연구에서 기술적인 면과 관련된 위험 요인 항목이 거의 보이지 않았던 것과 유사

하며 프로젝트 관리에 있어 기술적인 측면보다 사용자나 팀원 등과 관련된 관리적인 측면에 더욱 치중해야 한다는 것을 보여준다고 할 수 있다.

각 중요 항목을 그룹별로 나누어서 보았을 때, 기업환경과 관련된 위험 요인그룹과 프로젝트 관리 그룹, 요구사항 그룹이 각각 세 가지 위험 요인을 포함하였고 인력관리, 기술, 외부의존과 관련된 위험 항목은 중요도 순위 20위안에 하나도 포함되지 않았다. 따라서 인력관리나 기술, 외부의존보다는 기업환경, 프로젝트 관리, 요구사항과 관련된 위험 요인들이 더 중요하게 부각되었다. 위험 요인 1위부터 20위까지의 평균과 표준편차, 분산, 범위, 최소값, 최대값, 최빈값 등은 <표 7>에 나타나 있다. 특히 가장 중요한 위험 요인으로 선정된 무리한 비용의 산정은 응답자 18명 가운데 5명이 1위로 응답해 가장 많은 응답자가 1위로 선정 하였으며 응답자들 사이의 의견 일치가 어느 정도는 이루어진 것으로 보인다.

<표 7> 평균, 분산 및 기타 자료

	평균	표준편차	분산	범위	최소값	최대값	최빈값
1위	5.56	4.31	18.61	11	1	12	1
2위	6.28	4.35	18.92	19	1	20	5
3위	7.06	2.92	8.53	11	2	13	4, 5
4위	7.11	5.37	28.81	16	1	17	1, 4
5위	7.28	4.81	23.15	15	2	17	2
6위	7.61	3.55	12.60	14	3	17	8, 9
7위	8.56	4.38	19.20	15	3	18	3
8위	9.00	7.10	50.35	18	1	19	1
9위	9.33	6.02	36.24	18	2	20	3, 4
10위	10.56	5.61	31.44	18	3	21	12
11위	10.78	3.86	14.89	17	1	18	13
12위	12.22	3.99	15.95	15	6	21	10
12위	12.22	5.09	25.95	17	2	19	16
14위	16.00	3.46	12.00	12	9	21	16, 17, 18
15위	13.61	5.45	29.66	18	3	21	15
16위	13.83	5.42	29.32	19	1	20	14
17위	14.28	6.07	36.80	19	1	20	18, 19, 20
18위	14.39	7.47	55.78	20	1	21	20, 21
19위	14.44	6.85	46.97	19	2	21	21
20위	16.00	3.46	12.00	12	9	21	15

항목별 의견 일치도를 보면, 3위 사용자의 요구사항 변동 및 불명확한 요구사항으로 확정된 요구사항 부족과 6위 비효율적인 요구사항관리 관행, 14위 능력 있는 개발자 확보 실패 및 부적당한 인력 배치, 20위 가능한 전문 인력의 부재 등은 분산 13이하로 응답자들 사이의 의견일치가

높은 항목으로 판단된다. 반면 8위 다수 업체의 무리한 경쟁으로 비현실적인 제안서 초래, 9위 정확한 예측 부족으로 인한 잘못된 과제수행계획, 10위 프로젝트 관리자의 업무 능력 부족, 17위 효과적인 개발 프로세스/방법론 및 절차와 기준의 부재, 18위 IT산업의 투자가 적절히 이루어지지 않음, 19위 수주기업의 비도덕적인 프로젝트 수행 등 6가지 항목은 분산 30이상으로 응답자들 사이의 일치도가 낮은 항목으로 판단된다.

### 4.2.3 국외 연구와의 비교

본 연구의 목적은 국내의 소프트웨어개발 프로젝트 위험 요인의 도출이므로 외국의 경우와 비교했을 때, 어떠한 점들이 차이가 나는지를 살펴보았다. 본 연구에서는 델파이 기법을 적용하여 41명의 미국, 홍콩, 핀란드의 프로젝트 관리자를 대상으로 위험 요인을 도출한 Schmidt et al.(2001) 연구와 비교를 하였는데 이는 <표 8>에 잘 나타나 있다.

먼저 가장 큰 차이점은 가장 중요한 위험 요인으로 선정된 무리한 비용의 산정이다. Schmidt et al.(2001)의 연구에서는 비용과 관련된 항목은 첫 번째 단계에서의 위험 요인 전체 리스트에만 포함되었을 뿐 중요도 순위는 20위권 밖으로 밀려날 정도로 그 중요성은 그리 높지 않았다. 하지만 국내의 경우는 비용과 관련된 측면이 가장 중요한 위험 요인으로 선정되었고 비용이 모든 위험의 근본 문제라고 생각하는 경우도 있었다. 이는 좁은 국내 시장과 다수 업체의 경쟁으로 무리한 수준의 프로젝트 비용이 산정되기 때문이며 특히 프로젝트 수주에 있어서 하청 받은 기업이 다시 하청을 주고 또 다시 하청을 주는 등의 부적합한 제하청 관행 또한 주요 이유라고 판단된다.

따라서 비용 결정에 있어 무조건 저렴하게만 구축하려 하기보다는 좀 더 현실적으로 긴 안목을 갖고 비용에 대한 결정을 내려야 할 것이며 부적절한 하청 관행 또한 없애도록 해야 할 것이다. 이와 유사한 항목으로 무리한 프로젝트 기간의 산정을 들 수 있다. 이 항목 또한 Schmidt et al.(2001)의 연구에서는 중요도 순위 20위에 들지 못했지만 국내에서는 5위에 랭크되었다. 이 또한 비용과 유사한 이유로 설명될 수 있으며 장기적인 시각으로 프로젝트기간과 비용을 설정해야 할 것이다.

두 번째로는, 중요도 순위 4위의 명확치 않은 업무 범위이다. 이는 기존의 Boehm(1991), Keil et al.(1998)과 Schmidt et al.(2001)의 연구에서는 나타나지 않는 항목이지만 국내에서는 매우 중요한 위험 요인 중 하나로 밝혀졌다. 프로젝트는 제품이나 서비스를 창출하기 위한 한시적인 활동으로 팀원들은 기존의 업무와 프로젝트로 인한 업무 두 가지를 병행하게 된다. 따라서 업무 범위에 있어서의 모호함과 더불어 이중의 부담을 질 수도 있으며 이것이 중요한 위험 요인이 되는 것으로 보인다. 특히 국외보다 국내의 경우는 적은 인력과 낙후된 개발 환경으로 이러한 업무 범위의 불명확함과 업무 부담이 더욱 심각한 것으로 그 이유를 생각해 볼 수 있다.

세 번째는, 다수 업체의 무리한 경쟁으로 인한 비현실적인 제안서 초래를 들 수 있다. 프로젝트를 수주하기 위해서는 사업배경과 목적, 범위, 추진방향, 제안업체의 일반현황과 프로젝트 일반 계획 등에 관한 제안서를 제출하게 되는데, 국내의 경우는 좁은 시장에서 다수의 업체가 경쟁함으로써 실행가능성이 희박한 비현실적인 제안서를 제출하게 되며 이것이 국내에서 독특하게 위험 요인으로 도출되는 항목 중 하나이며 상당히 중요한 위험 요인임을 알 수 있다.

<표 8> 프로젝트 위험 요인 순위의 비교

순위	본 연구의 위험 요인	Schmidt et al.의 위험 요인
1	무리한 프로젝트 비용의 산정	최고 경영층의 지원 부족
2	범위와 목표의 변경	사용자 관여 및 관심의 획득 실패
3	확정된 요구사항의 부족	요구사항에 대한 이해 부족
4	명확치 않은 업무 범위	적절한 사용자 참여의 부족
5	무리한 프로젝트 기간의 산정	프로젝트 팀원의 지식/기술의 부족
6	비효율적인 요구사항관리 관행	고정된 요구사항의 부족
7	고객의 무리한 요구	범위/목표의 변화
8	비현실적인 제안서	새로운 기술의 도입
9	잘못된 과제수행계획	최종 사용자 기대의 관리 실패
10	프로젝트 관리자의 업무 능력 부족	부적합한 인력 배치
11	고객과 팀간의 커뮤니케이션 부족	사용자 부서 사이의 갈등
12	사용자 협조 부족	사용자로부터의 협동 부족
13	의사결정 사항의 지연	소유자 또는 상위 관리자의 변화
14	부적당한 인력 배치	직원 배치에 있어서 지속성 부족
15	위험관리가 이루어지지 않음	효과적 개발 프로세스/방법론 부족
16	지원 부족	변화가 적절히 관리되지 않음
17	효과적 개발 프로세스/방법론 부족	효과적인 프로젝트 관리 기술 부족
18	IT산업의 투자가 적절치 않음	효과적인 프로젝트 관리 방법론 부족
19	수주기업의 비도덕적 프로젝트 수행	명확치 않은 범위/목표
20	가능한 전문 인력의 부재	역할과 책임의 부적합한 정의

네 번째는, 정확한 예측 부족으로 인한 잘못된 과제수행계획이다. 제안서를 통해 프로젝트를 수주하게 되면 프로젝트 관리자가 결정되며 프로젝트 관리자가 일정이나 인원, 조직, 시스템, 하드웨어, 소프트웨어 등에 관해 자세하게 과제수행계획서를 제출하게 된다. 하지만 실제로는 과제수행에 예견되는 기간, 비용, 인원 등 자원 할당의 정확한 예측 부족으로 대부분의 과제수행계획서가 실제와 차이가 나타나게 되며, 정확한 예측 능력의 부족이 잘못된 계획으로 이어져 전체 프로젝트의 실패로 나타나게 된다.

마지막으로 중요도 10위로 선정된 프로젝트 관리자의 업무 능력 부족을 들 수 있다. 국외의 경우는 프로젝트 관리자의 업무 능력이 프로젝트 관리 기술 항목의 하부 요인으로 포함될 정도로 크게 부가되지 않는 반면에, 국내의 경우는 프로젝트 관리자의 능력을 아주 중요하게 여기고 있었다. 특히 이 항목에는 기술적인 면뿐만 아니라 사람을 관리하는 능력이나 리더십, 관련 업무 경험과 지식 등 폭넓은 면에서의 능력을 요구했으며 프로젝트 관리자에 대한 기대와 책임이 막중하다고 여기고 있었고, 프로젝트 관리자의 업무 능력이 부족할 경우 전체 프로젝트를 잘못된 방향으로 흐르게 할 수 있을 정도로 매우 중요한 위험 요인으로 여기고 있다는 것을 알 수 있었다.

<표 9> 켈달의 일치계수 W의 해석

W	해석	순위에의 확신
0.1	매우 약간 일치함	확신 불가능
0.3	약간 일치함	약간 확신 가능
0.5	어느 정도 일치함	어느 정도 확신 가능
0.7	강하게 일치함	확신 가능
0.9	매우 강하게 일치함	매우 확신 가능

자료원 : Schmidt, R. C., (1997), p. 767.

#### 4.2.4 의견 일치도에 관한 분석

본 연구에서는 델파이 기법을 사용하여 정해진 전체 순위가 과연 얼마만큼 패널들 사이의 일치가 이루어졌는가를 알아보는 방법으로 켈달의 일치계수 W를 사용하였다. 이는 순위에 대한 일치도를 알아보기 위해서 Schmidt(1997)의 연구에서 제시된 해석을 기준으로 삼았으며 앞의 <표 9>에 일치계수 W에 대한 해석이 나타나있다. 본 연구 결과에 있어서 켈달의 일치계수 W는 0.31로 나타났으며 결과 순위에 있어서 패널들의 의견은 약간 일치하는 수준으로 나타났다.

## V. 결론

### 5.1 연구의 요약

치열한 경쟁에서 살아남기 위해 기업들은 정보기술의 전략적인 활용을 위해 노력하고 있으며 소프트웨어개발 프로젝트를 성공시키기 위해서 많은 프로젝트 방법론들이 제시되고 있다. 프로젝트가 실패하는 원인 중 하나는 프로젝트 진행 중에 나타날 수 있는 위험에 대한 고려를 소홀히 했기 때문이며, 시스템이 복잡화, 대형화 될수록 위험 관리의 필요성은 더욱 증가하므로 성공적인 프로젝트를 위해서는 반드시 프로젝트와 관련된 위험을 확인 및 분석함으로써 프로젝트가 완료될 때까지 위험을 지속적으로 관리해야 한다. 특히 소프트웨어 개발 프로젝트는 실패 위험이 높으며 개발 과정 또한 정형화되어 있지 않아서 프로젝트 관리와 관련한 위험 요인의 도출에 관한 연구의 필요성이 절실하다.

본 연구에서는 국내 소프트웨어 개발 프로젝트 전문가를 대상으로 순위 형식 델파이 기법을 사용하여 14개 그룹, 45개 항목으로 구성된 위험 요인 리스트와 그 중요도에 따른 위험 요인 순위를 도출하였다.

국내의 소프트웨어 개발 프로젝트 관리자들이 중요한 위험 요인으로 생각하고 있는 것은 무리한 비용의 산정, 범위와 목표의 변경, 사용자의 요구사항 변동 및 불명확한 요구사항으로 확정된 요구사항의 부족, 명확치 않은 업무 범위, 무리한 프로젝트 기간의 산정 순으로 나타났다.



국의 연구와의 가장 큰 차이점은 최고 경영층의 지원 부족과 무리한 프로젝트 비용의 산정이었으며 응답자들간의 의견 일치도를 알아보기 위한 켄달의 일치계수  $W$ 는 0.31로 나타나 결과 순위에 있어서 응답자들간의 의견이 일치도는 조금 약한 것으로 보인다.

연구 결과에 기초하여 본 연구의 시사점을 살펴보면 다음과 같다. 첫째, 21명의 패널에 의해 확인된 45개의 위험 요인 리스트는 소프트웨어 프로젝트 위험 평가 지침을 개발하는데 사용되는 종합적인 체크리스트로 제공될 수 있다. 본 연구에서 제시한 리스트는 국외에서의 연구 결과가 아니라 국내의 프로젝트 관리자들 통해 얻어진 결과이므로 더욱 근거 있고 활용 가능하다는 장점을 지닌다.

둘째, 프로젝트에 있어서 중요한 위험 요인은 기술적인 측면과 관련되기보다는 대부분이 관리적인 측면과 관련된다는 것을 알 수 있었다. 이러한 점은 국외의 기존 연구와 거의 유사한 결과를 보였으며, 따라서, 프로젝트 관리를 함에 있어서 새로운 기술이나 방법에 관련된 것보다는 팀원, 사용자 참여, 지원 등과 관련된 관리적인 측면을 더욱 중요하게 여겨야 할 것이다.

셋째, 기존의 국외 연구에서는 나타나지 않았지만 국내의 프로젝트 위험 요인으로 나타난 대표적인 항목은 무리한 프로젝트 비용의 산정, 명확치 않은 업무 범위, 무리한 프로젝트 기간의 산정, 다수 업체의 무리한 경쟁으로 비현실적인 제안서 초래, 정확한 예측 부족으로 인한 잘못된 과제수행계획, 프로젝트 관리자의 업무 능력 부족 등이다. 이 중, 비용과 기간의 산정, 비현실적인 제안서 초래 등은 좁은 국내 시장과 다수 업체의 무리한 경쟁 그리고 하청과 재하청이 반복되는 부적합한 하청 관행이 그 근본 원인이며, 좀 더 멀리 볼 수 있는 장기적이고 현실적인 시각과 더욱 많은 지원, 적절한 하청 관행의 확립 등이 필요하다고 할 수 있다.

## 5.2 연구의 한계점 및 향후 연구 과제

본 연구의 수행에 있어 몇 가지 한계점을 지적하면 다음과 같다. 첫째, 델파이 분석을 사용한 다른 연구에서와 같이 연구 결과가 제한된 수의 응답자에 기초했다는 것이다. 문항이 주관식으로 시작하며 연구 기간이 길고 여러 차례 응답해야 하는 어려움 등으로 많은 수의 응답자를 확보하는 것이 어려웠다.

둘째, 응답자 수의 부족으로 무작위적으로 패널들이 선택되지 못했고 프로젝트의 유형이나 크기, 산업의 유형, 프로젝트 관리자의 경력 등에 따라 통제하지 못했다.

셋째, 패널들의 의견 일치도를 높이기 위해 한번 더 조사를 실시하고자 하였으나 응답자들의 협조 부족으로 3단계에서 조사를 끝내게 되었다.

따라서, 향후 연구에서는 충분한 수의 패널들의 참여와 적극적인 협조가 필요할 것이고 산업별, 프로젝트 성격별 혹은 프로젝트 수행 경험 기간별로 위험 요인에 어떠한 차이가 있는지 연구할 필요가 있다. 그리고 이러한 프로젝트 위험 요인에 대한 연구 결과를 바탕으로 각각의 위험 요인 항목에 대한 대응책은 어떠한 것이 있는지를 알아보는 연구나 상이한 이해관계자들이 위험 요인에 대해 어떠한 의견 차이를 가지고 있는지에 대한 연구도 필요할 것이다.

## 참고문헌

- 서창교, 정은희, “프로젝트 위험과 위험관리가 소프트웨어 개발 프로젝트 성과에 미치는 영향”, 경영정보학연구, 제13권 제2호, 2003, pp. 199-217.
- 양훈모, “집단 의사결정 기법들의 효율성 비교연구 : 상호작용집단, 브레인스토밍, 명목집단 및 델파이 기법을 중심”, 성균관대학교, 박사학위논문, 1991.
- 이경환, 소프트웨어 공학, 청문각, 1992.
- 이종성, 델파이 방법, 교육과학사, 2001.
- 조영재, 김석수, 소프트웨어 프로젝트 개발 방법론, 홍릉과학출판사, 2000.
- Barki, H. Rivard, S. & Talbot, J., “Toward an Assessment of Software Development Risk,” *Journal of Management Information Systems*, Vol. 10, No.2, 1993, pp. 203-225.
- Barki, H. Rivard, S. & Talbot, J., “An Integrative Contingency Model of Software Project Risk Management,” *Journal of Management Information Systems*, Vol. 17, No. 4, 2001, pp. 37-69.
- Baskerville, R. L., “Controlling Prototype Development through Risk Analysis,” *MIS Quarterly*, Vol. 20, No. 4, 1996, pp. 481-504.
- Bennatan, E. M., *On Time Within Budget : Software Project Management Practices and Techniques*, New York : John Wiley & Sons, 2000.
- Boehm, B. W., “Software Risk Management : Principles and Practices,” *IEEE Software*, Vol. 8, No. 1, 1991, pp. 32-41.
- Carr, M. J., Konda, S. L., Monarch, I., Ulrich, F. C. & Walker, C. F., Taxonomy-Based Risk Identification, Software Engineering Institute, Technical Report CMU/SEI-93-TR-6, ESC-TR-93-183, June 1993.
- Dickson, G., Leitheiser, R., Wetherbe, J. and Nechis, M., “Key information systems issues for the 1980s,” *MIS Quarterly*, Vol. 8, No. 3, 1984, pp. 35-159.
- Fairley, R., “Risk Management for Software Projects,” *IEEE Software*, Vol. 11, No. 3, 1994, pp. 57-67.
- Grey, S., *Practical Risk Assessment for Project Management*, New York : John Wiley & Sons, 1995.
- Hall, E. M., *Managing Risk : Methods for Software Systems Development*, Addison-Wesley, 1998.
- Higuera, R. P. & Haimes, Y. Y., Software Risk Management, Software Engineering Institute, Technical Report, CMU/SEI-96-TR-012, ESC-TR-96-012, June 1996.
- Keil, M., Cule, P. E., Lyytinen, K. & Schmidt, R. C., “A Framework for Identifying Software Project Risks,” *Communications of the ACM*, Vol. 41, No. 11, 1998, pp. 76-83.

- King, W. R., Marks, P. V. and McCoy, S., "The most import issues in Knowledge Management," *Communications of the ACM*, Vol. 45, No. 9, 2002, pp. 93-97.
- McFarlan, F. W., "Portfolio Approach to Information Systems," *Harvard Business Review*, Vol. 59, No. 5, 1981, pp. 142-150.
- Mantel, S. J., *Project Management in Practice*, New York : John Wiley & Sons, 2001.
- Moynihan, T., "How Experienced Project Managers Assess Risk," *IEEE Software*, Vol. 14, No. 3, 1997, pp. 35-41.
- Niederman, F., Brancheau, J. and Wetherbe, J., "Information systems management issues for the 1990s," *MIS Quarterly*, Vol. 15, No. 4, 1991, pp. 475-500.
- Schmidt, R. C., "Managing Delphi Surveys Using NonparametricStatistical Techniques," *Decision Sciences*, Vol. 28, No. 3, 1997, pp. 763-774.
- Schmidt, R. C., Lyytinen, K., Keil, M. & Cule, P., "Identifying Software Project Risks : an International Delphi Study," *Journal of Management Information Systems*, Vol. 17, No. 4, 2001, pp. 5-36.
- Spinner, M. P., *Project Management : Principles and Practices*, UpperSaddle River N. J. : Prentice Hall, 1997.
- Willcocks, L. and Griffiths, C., "Predicting Risk of Failure in Large Scale Information Technology Projects," *Technological Forecasting and Social Change*, Vol. 47, 1994, pp. 205-228.

<Abstract>

## A Delphi Study on Software Project Risks

Kyung-Soo Chung, Myeong-Hui Kang, Yong Kim

Most of the software development projects bear risks that need analysis and management. Risk management plays a critical role for the success of software project management. In this study, we have used delphi method to delineate critical risk factors.

The study pulls out 20 project risk factors from 21 project managers. It is certainly clear that certain features are more risky than others. Our study shows that unrealistic cost estimation and changes in scope and objective are more risky than other features.

**Keyword** : Project Risk, Project Management, Delphi Method