

국내 소프트웨어 개발 프로젝트 위험 요인 도출에 관한 연구

강명희 (경북대학교 경영학과)

정경수 (경북대학교 경영학과 교수)

I. 서론

1.1 연구 배경 및 목적

글로벌 시장에서 더욱 효과적으로 경쟁하도록 압력이 높아짐에 따라, 기술의 변화가 드러나고 있고 기업들은 품질, 생산성 및 고객 만족도를 높이기 위해서 그들의 사업수행 관행을 재정의 하려고 애를 쓰고 있다. 주요 경영진에 이르기까지 모든 레벨의 업무수행이 사업목적과 목표를 달성하는 능력을 기반으로 재평가되고 있는 중이며, 그 결과 기업은 프로젝트 지향적으로 변하고 있다. 하지만, 빈번한 요구 사항 변경, 인력 관리와 관리 감독의 변동, 개발 인력의 빈번한 교체와 미흡한 통제 및 관리 등으로 시스템 개발 방법과 도구의 향상에도 불구하고 기업들이 추진하고 있는 프로젝트들 중 다수가 일정이나 비용을 초과하거나 완성 이전에 프로젝트 자체가 포기되기도 한다.

특히, 소프트웨어의 개발에 있어서는 산출물이 무형이고, 소프트웨어 개발 과정에 대한 확실한 이해를 할 수 없으며, 대형 소프트웨어 시스템의 경우 한번 사용되고 끝나는 프로젝트인 경향이 있다는 등의 이유로 프로젝트가 늦어지고, 예산의 범위를 초과하고 계획이 지연된다 (이경환, 1992). 소프트웨어 개발 프로젝트를 성공시키기 위한 요인으로는 효과적인 프로젝트 관리 실행, 개발 표준안과 정책 수립, 효과가 입증된 개발 패턴 사용 등을 들 수 있다 (조영재 & 김석수, 2000).

시스템이 대형화, 복잡화 되어가고 있고, Higuera & Haimes(1996)에 따르면 시스템이 복잡해질수록 위험 관리의 필요성이 증가한다고 한다. 프로젝트가 실패하는 원인 중 하나는 프로젝트 진행 중에 나타날 수 있는 위험에 대한 고려를 소홀히 했기 때문이며, 성공적인 프로젝트를 위해서는 반드시 프로젝트와 관련된 위험을 확인 및 분석함으로써 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 요인 확인에 관한 연구는 오래되었을 뿐만 아니라, 지리적·문화적 차이로 현재 국내에서 그대로 활용하기에는 무리가 있다 (Schmidt et al, 2001).

따라서, 본 연구의 목적은 기존 외국 문헌의 소프트웨어 위험 요인과 비교하여 국내의 소프트웨어 개발 프로젝트 전문가를 대상으로 프로젝트 위험 요인과 이들의 우선 순위를 도출하여 조직 문화, 기업 문화 등 국내 환경을 반영한 프로젝트 위험 요인을 찾고자 하는 것이다.

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 시간, 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 기술이 요구된다.

적절한 프로젝트 관리가 이루어지지 않으면 계획된 비용을 초과하고 완성된 이후에도 유지 보수가 많이 필요한 소프트웨어를 생산할 가능성이 커지게 된다. 그러므로 소프트웨어를 개발할 때는 첫째, 정해진 일정에 맞게 개발, 둘째, 계획된 예산 내에서 개발 완료, 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 개념

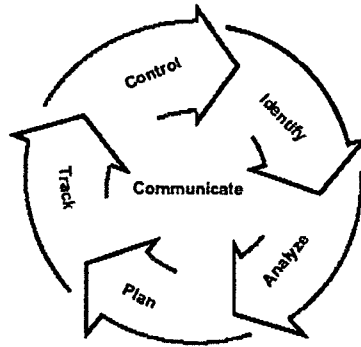
위험 관리는 프로젝트 관리와 완전히 같은 것을 의미하지는 않고, 그것을 대체하거나 완전히 별개의 것도 아니다. 전통적인 프로젝트 관리의 명확한 확장이라기보다는 정보 획득이나 의사 결정에 있어 밀접하게 서로 얽혀있다. 프로젝트 관리의 목적은 목표를 이루는 프로젝트이다. 프로젝트가 성공적일 때, 그것은 문제가 없었기 때문이 아니라 문제가 성공적으로 극복되었기 때문이다. 위험 관리가 성공을 보장하지는 않지만, 문제 상황을 피하기 위해 충분한 리드 타임을 가지고 잠재적인 문제를 확인하고 대응하는 최우선 목표를 가짐으로써 프로젝트 관리가 그 목표를 이룰 수 있도록 한다.

Higuera & Haimes(1996)에 따르면 위험을 관리해야 하는 필요는 시스템의 복잡성에 따라 증가하며, 개인적 지식, 판단, 그리고 경험을 보충하기 위한 더욱 체계적인 방법과 도구의 필요성이 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 정의된다 (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 활동으로 정의되며, 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 가정하에 소프트웨어 개발 위험 관리를 구성하는 활동의 모형을 제시하였다 <그림 1>.



<그림 1> SEI의 위험관리체계

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 커지고 있으며, SEI의 성공적인 위험 관리란 다음과 같다.

“성공적인 위험 관리란 위험이 상대적인 중요도에 따라 지속적으로 확인되고 분석되는 것이고, 프로그램 자원을 효과적으로 사용하기 위해 위험이 줄어들고, 추적되며, 통제되는 것이다. 문제가 발생하기 전에 방지되고, 직원들은 의식적으로 제품의 품질과 일정에 영향을 미칠 수 있는 것에 초점을 맞춘다” (Bennatan, 2000).

2.2.2 위험중심적 접근법의 이론적 배경

위험 관리가 현재 소프트웨어 프로젝트에 사용되고 있는 데는 여러 가지 이유가 있다. 프로젝트에 있어 불확실성을 관리하기 위한 능력은 부족한 자원과 기술적인 진보를 다루기 위해 설계된 요구사항과 급격히 변화하는 환경에서 복잡한 시스템을 위해 증가된 요구사항이다. 현재의 이윤이 축소된 사업 환경, 국제 경제와 불확실한 시장 상황, 그리고 급격한 기술적 진보에 의한 경쟁 압력 등 모두가 그 이유라고 할 수 있다. 이러한 필요에 따라, 위험 관리 방법이 소프트웨어 관리자와 엔지니어에 의해 사용되고 맞추어지고 있다.

위험 관리 기법은 1980년대에 소프트웨어 위험 관리의 아버지로 일컬어지는 Barry Boehm이 반복적이고 위험 지향적인 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 주장했으며, 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 제시했다.

Boehm(1991)은 복잡한 어플리케이션 영역에서 순차적이고 문서 지향적인 폭포수 프로세스 모델이나 코드 지향적인 진화적 개발 프로세스 모델이 높은 위험 요인을 발생시키고 프

로젝트를 실패로 이끈다고 하며, 개발 초기에 위험을 확인하고 다루는 것이 장기적인 비용을 절감시키고 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.

Fairley(1994)는 위험 관리가 명확한 프로젝트 관리 활동으로 적용되지 않는 이유는 실제적이고 단계적인 접근법을 제공하는 지침이 없기 때문이라고 하며, 모든 유형의 소프트웨어 프로젝트에 적용될 수 있는 위험 관리를 위한 일곱 단계의 프로세스를 제안했다.

Baskerville(1996)은 시스템 개발자와 사용자사이의 커뮤니케이션을 강조하는 프로토타이핑 개발방법의 가치를 향상시키기 위해 위험 경감 모형과 위험 관리 프로세스를 사용한 접근법을 제시하였다. 또한 실행연구(Action Research)를 통해 이러한 위험 분석 접근법이 프로토타이핑 환경에서도 일관성 있는 과정과 우선 순위 결정에 유효함을 검증하였다.

Barki et al.(2001)의 연구에서는 소프트웨어 프로젝트 위험 관리와 조직 이론 문헌의 상황 이론으로부터 소프트웨어 프로젝트 위험 관리의 통합적 상황 모델을 개발하여 75개 소프트웨어 프로젝트의 프로젝트 관리자와 주 사용자를 대상으로 조사한 결과 제안된 상황 모형이 지지되었고 프로젝트 성과를 향상시키기 위해 프로젝트 위험 관리가 프로젝트 위험에 따라 변화될 필요가 있다는 것을 주장했다.

2.3 소프트웨어 개발 프로젝트 위험 요인에 관한 선행연구

소프트웨어 개발 프로젝트에 있어서 프로젝트 위험은 소프트웨어 개발 프로젝트에 예상치 못한 결과를 가져다 줄 수 있다. 하지만 단순히 잠재적인 위험을 확인한다는 것은 더욱 어려워지고 있으며, 그 이유 중 하나는 책임과 권력의 이전을 일으키는 기술과 산업의 변화가 상대적으로 더 적은 경험을 가진 사람들에게 의해 더욱 더 복잡한 시스템이 수행되기 때문이다. 위험은 본질적으로 예측할 수 없기 때문에 창의성을 필요로 하며 위험의 확인은 프로젝트 기간 전체에 걸쳐 수행되어야 한다 (Grey, 1995).

소프트웨어 위험은 소프트웨어 프로젝트, 프로세스, 또는 제품에 영향을 미치는 불만족스러운 결과의 가능성과 손실을 의미한다. 효과적인 위험 관리를 위해서는 우선 위험이 무엇인지 확인되어야 하며, 관리의 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)은 사례 조사를 통해 위험 요인을 프로젝트 크기, 기술 경험, 프로젝트 구조로 파악하였으며, Boehm(1991)은 설문 조사를 통해 잘못된 기능과 특성의 개발, 계속되는 요구사항의 변화 등 10개의 위험 항목을 추출하였다. Barki et al.(1993)은 문헌 조사를 통해 어플리케이션의 복잡성과 크기, 기술, 전문성, 조직환경요인의 5개 위험영역을 도출하였으며, Moynihan(1997)은 개인구성개념추출법을 통해 21개의 위험 요인을 도출, 이를 Barki et al.(1993), Carr et al.(1993)의 위험 요소와 비교하였다. 그리고 Keil et al.(1998)은 델파이 분석을 통해 요구사항정의를 포함한 11가지 위험변수를 도출해 이를 상황 적합적 관점에서 범주화하였고, Schmidt et al.(2001)은 홍콩, 핀란드, 미국 세 나라에서 델파이 분석을 통해 위험 요인 항목을 도출하고 이를 비교하는 연구를 하였다.

Ⅲ. 연구 설계 및 조사 방법

3.1 델파이 방법

델파이(Delphi) 기법은 전문 토론을 거치지 않고 성원들로부터 전문적인 견해를 얻어내는 방법으로 (박운성, 1996), 예측하려는 문제에 관하여 전문가들의 견해를 유도하고 종합하여 집단적 판단으로 정리하는 일련의 절차라고 정의할 수 있다. 이 방법은 미국의 랜드연구소(Rand Corporation)에서 개발하였으며 대면토의에서 나타나는 제한점을 제거하고 긴급한 국방문제에 관하여 전문가들의 합의를 도출하는데 1950년대에 최초로 사용된 후 여러 분야에서 미래를 예측하는 문제뿐만 아니라 연구방법으로 발전되었다.

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).

경영학과 관련된 부분에서도 수정된 접근법이 이슈의 상대적 중요도에 대한 그룹 합의를 형성하는데 사용되어 왔으며, 이러한 델파이 방법의 순위 형식은 사회 사업과 교육, 경영 과학 그리고 정보시스템과 같은 다양한 분야에 사용되어 왔다 (Schmidt, 1997).

본 논문의 목적은 위험 요인에 관한 신뢰성 있는 리스트를 개발하고 어떠한 위험 요소가 가장 중요한지를 알아보는 것이다. 그러한 정보를 위한 분명한 근거는 실제로 소프트웨어 개발 프로젝트를 관리하는 데 있어 몇 년 동안의 경험이 있는 전문가일 것이다. 그러나 단일의 전문가는 요인들의 종합적인 리스트를 산출하는데 필요한 모든 개발 환경의 경험을 가지지 못할 것이다. 신뢰성 있고 타당성 있는 자료 수집 과정을 확보하기 위해 다양한 의견을 받아들일 수 있도록 개방형 질의를 해야 하고, 피드백 기반의 의견 조합을 추구해야 하며, 소프트웨어 개발에 실제로 포함되는 요인을 도출해야만 한다. 따라서 통제된 피드백의 반복을 통해 전문가들의 견해를 도출하는 순위 형식 델파이 방법(Schmidt, 1997)을 본 논문의 연구 방법으로 선택하였다. 몇 년간의 경험을 가지는 프로젝트 관리자들로 패널을 구성하였고, 위험 요인을 도출하고 순위를 매기기 위해 체계적인 절차를 사용하였다.

3.2 자료 수집과 분석 방법

자료 수집과 분석은 Schmidt et al.(2001)의 연구에서 행해진 방법에 기초했으며 델파이 분석 과정은 크게 세 단계로 나누어진다.

첫 번째 단계는 브레인 스토밍과 같은 단계로 패널들로부터 가능한 많은 항목을 추출하기 위해 시행되어 지는 단계이다. 완전히 구성되지 않는 자유로운 형태, 즉 백지상태(blank type)로 전문가들에게 각자 담당하고 있는 문제영역에 대해 응답하도록 한다. 각 패널들은 최소한 여섯 가지 이상의 요인을 제출하도록 요구되어지며, 각 패널들이 같은 항목을 다른 용어를 사용해서 말할 수 있기 때문에 패널들에게 각 요인들에 대해 짧은 설명을 하도록 요구된다. 각 패널들로부터 받은 항목들을 비슷한 항목은 분류정리하고 중재자의 입장에서 중요하지 않은 항목은 제외시켜 전반적으로 간결한 항목을 만들어 두 번째 단계의 설문서 사

용할 수 있게 한다.

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 요구된다. 20개의 항목을 가지는 리스트를 만드는 것을 목표로 하며 높은 빈도를 가지는 항목으로 이루어지는 리스트를 만든다. 본 연구에서는 20위 요인의 동 순위로 인하여 21개의 항목을 포함하는 리스트가 만들어졌다.

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

양훈모(1991)의 연구에 따르면 델파이 기법에서 집단 규모는 그 특성이 동질적인 경우 10명에서 15명이면 족하며, 집단 특성이 독립변수일 때는 수백명에 이르기까지 다양하게 조사할 수도 있다고 하며 본 연구의 경우는 집단의 특성이 소프트웨어 개발 프로젝트 관리 경험자라는 동일한 특성을 가지고 있으므로 10명에서 15명이면 족하다고 할 수 있다. 응답자는 프로젝트 경험이 5년 이상 되고 프로젝트 관리 경험이 있는 사람으로 선정했으며 각 단계별 조사 기간과 응답자 수는 <표 1>에 나타나 있다.

<표 1> 조사기간 및 각 단계별 응답자 수

	델파이 조사기간	응답자수
제 1차 조사	2002년 8월 5일 ~ 2002년 9월 20일	21명
제 2차 조사	2002년 10월 4일 ~ 2002년 10월 9일	18명
제 3차 조사	2002년 10월 14일 ~ 2002년 10월 23일	18명

본 연구의 분석 기법은 SPSS 10.0 Window용 프로그램을 사용하였으며 사용한 분석 방법은 평균 순위와 의견 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

프로젝트 경력 5년 이상이며 프로젝트 관리 경험이 있는 사람을 대상으로 설문을 실시하였으며, 최종 응답자 18명의 프로젝트 경력 기간은 <표 2>와 같으며 응답자의 평균 프로젝트 경력 기간은 9년 6개월로 나타났다.

<표 2> 응답자의 프로젝트 경력 기간

프로젝트 경력 기간	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		9년 6개월

4.2 연구 결과

4.2.1 위험 요인 리스트

1단계에서 확인된 위험 요인의 리스트는 <표 3>과 같다. 분류 기준은 Schmidt et al.(2001)의 위험 요인 분류에 기초해 14개의 그룹으로 구성하였으며, 총 45개의 위험 요인으로 정리하였다.

<표 3> 국내 소프트웨어 개발 프로젝트의 위험 요인 리스트

1. 기업환경
1.1 수주기업의 비도덕적인 프로젝트 수행 : 프로젝트 수주시 까지만 열성적으로 임하고, 이후에 인력 등 각종 자원을 제대로 투입하지 않음
1.2 새로운 시스템에 필요한 비즈니스 프로세스가 현재의 기업문화와 맞지 않음
1.3 IT산업의 투자가 적절히 이루어지지 않음 : 무조건 저렴하게만 개발하려 함
1.4 다수 업체의 무리한 경쟁으로 비현실적인 제안서 초래
1.5 무리한 사업영역의 설정 : 기업의 능력과 관계없이 어떠한 사업도 수행이 가능하다는 기업경영방식
2. 자원
2.1 적절한 지원 체제의 부재 및 관련 조직의 지원 부족 : 전문적인 문제나 예상치 못한 문제 발생 시 이를 해결 할 수 있는 적절한 지원 체제가 없어 솔루션, 품질 보증 등 관련 조직의 지원이 제대로 이루어지지 않음
2.2 사용자의 관심과 준비 및 지원 부족
2.3 테스트 환경 지원 부족 : 테스트에 필요한 데이터나 시스템 환경을 구성하지 못하는 경우, 단위 업무에 대한 테스트만 진행이 되거나 개발자가 테스트 함
2.4 상위 관리층의 지속적인 검토와 지원 부족

3. 관계 관리
<p>3.1 사용자로부터의 협조 부족 : 사용자가 요구 사항 제시를 거부하거나 테스트 환경 지원을 거절</p> <p>3.2 사용자의 적절한 참여 부족</p> <p>3.3 고객과 프로젝트팀간의 커뮤니케이션 부족 : 적절한 방법을 통해 정기적인 의사 소통으로 문제를 조기 식별/조치하여야 하나 이러한 활동이 거의 되지 않거나 형식적임. 시스템에 대한 이해 차이 발생</p>
4. 프로젝트 관리
<p>4.1 프로젝트 관리자의 업무 능력 부족 : 프로젝트 관리자의 관리 및 관련 기술에 대한 경험과 지식 부족</p> <p>4.2 변화가 적절히 관리되지 않음</p> <p>4.3 효과적인 프로젝트 관리 기술의 부족 : 형상관리와 효율성 분석 등을 포함하는 관리 기술의 부족</p> <p>4.4 위험관리가 제대로 이루어지지 않음 : 문제가 발생하고 나서야 수습</p> <p>4.5 팀원 및 외부업체와 외부컨설턴트 등의 책임과 역할의 정의 및 분배가 제대로 되지 않음</p> <p>4.6 효과적인 프로젝트 수행 방법론의 부재</p> <p>4.7 의사결정 사항의 지연 : 개발 단계별 필요한 의사결정 사항의 지연으로 빠른 진행이 어려움</p>
5. 범위
<p>5.1 범위와 목표의 변경</p> <p>5.2 범위관리 실패</p> <p>5.3 명확치 않은 업무 범위</p>
6. 요구사항
<p>6.1 비효율적인 요구사항관리 관행</p> <p>6.2 사용자 요구사항 변동 및 불명확한 요구사항으로 확정된 요구사항 부족</p> <p>6.3 프로젝트 팀원의 구축할 시스템 및 대상 업무에 대한 이해 부족</p> <p>6.4 고객의 무리한 요구</p>
7. 자금
<p>7.1 무리한 프로젝트 비용의 산정</p>
8. 일정관리
<p>8.1 무리한 프로젝트 기간의 산정</p>

9. 개발 프로세스
9.1 효과적인 개발 프로세스/방법론 및 절차와 기준의 부재 : 단계별 연관성 취약, 근시안적 설계, 분석/설계 미비, 잘못된 architecture 선정 등
9.2 업무 절차 변경의 어려움
10. 인력관리
10.1 프로젝트 팀원에게 요구되는 지식/기술/경험의 부족
10.2 프로젝트 관리자의 인적관리 능력 부족
10.3 프로젝트 수행자들의 팀웍 부족 : 수행팀 내부의 협력 부족
10.4 기술적인 부분에만 치중하는 개발자들의 마인드
11. 인력배치
11.1 가능한 전문 인력 부재 : 필요할 때 적당한 기술을 가진 인력을 쓸 수 없음
11.2 능력 있는 개발자 확보 실패 및 부적당한 인력 배치 : 인원이 충분치 않거나 관련 기술이 부족한 인력이 프로젝트에 할당됨
11.3 외주 인력의 무분별한 사용 : 비용 절감 위해 검증되지 않은 외주 인력 사용
12. 기술
12.1 새로운 기술의 도입 위험 : 초기 도입되어지는 개발기술의 적용에 있어 예기치 못한 상황 발생으로 프로젝트 지연
12.2 잘못된 개발 툴/솔루션 선정
12.3 신기술에 대한 맹신
13. 외부 의존
13.1 공동개발의 경우 나쁜 팀웍
13.2 공동개발시 프로젝트 인원의 기술 파악 어려움 : 특히, 외주업체와의 공동개발시 인원의 기술을 개인 프로파일에만 의지하기에는 위험함
13.3 부실한 협력업체의 활용 : 비용을 줄이기 위해 재무구조가 취약하거나 직원 구성이 열악한 협력업체와 계약함으로써 인원의 잦은 교체, 향후 지속적인 지원 불가능
14. 계획
14.1 개발이후의 운영/교육/유지보수/재개발에 대한 계획(마스터플랜)의 부재
14.2 정확한 예측 부족으로 인한 잘못된 과제수행계획 : 과제 수행에 예견되는 기간, 비용, 인원 등 자원 할당의 정확한 예측 부족으로 대부분의 과제수행 계획서가 실제와 거리가 나타남

이를 기존의 위험 요인 도출 관련 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, 새롭게 위험 요인으로 추출된 것들이 있었다.

첫 번째는 기업 환경과 관련된 내용으로 수주기업의 비도덕적인 프로젝트 수행과 무조건 저렴하게만 개발하려 하는 IT산업의 부적절한 투자, 그리고 좁은 국내 시장에 따른 다수 업

체의 무리한 경쟁으로 비현실적인 제안서의 초래, 무리한 사업 영역의 설정 등이었고, 두 번째는 요구사항과 관련된 것으로 비효율적인 요구사항관리 관행, 프로젝트 팀원의 구축할 시스템 및 대상 업무에 대한 이해 부족, 그리고 고객의 무리한 요구 등이며, 이 밖에도 개발 단계별 의사결정 사항의 지연으로 인한 일정 지연, 업무 절차 변경의 어려움, 정확한 예측 부족으로 인한 잘못된 과제수행계획 등이 새롭게 등장한 위험 요인들이다.

4.2.2 위험 요인의 순위

3차에 걸쳐 조사된 위험 요인의 순위가 <표 4>에 나타나 있다.

<표 4> 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

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.

특히, 본 연구의 결과로 나타난 주요 프로젝트 위험 요인 20가지 항목 중 기술적인 측면과 관련된 것은 하나도 없었으며, 20가지 항목 모두 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 유사하며 프로젝트 관리에 있어 기술적인 측면보다 사용자나 팀원 등과 관련된 관리적인 면에 더욱 치중해야 한다는 것을 보여준다고 할 수 있다.

각 중요 항목을 그룹별로 나누어서 보았을 때, 기업환경과 관련된 위험 요인 그룹과 프로젝트 관리 그룹, 요구사항 그룹이 각각 세 가지 위험 요인을 가졌고 이에 반해 인력관리, 기술, 외부의존과 관련된 위험 항목은 중요도 순위 20위안에 하나도 포함되지 못했다. 따라서 인력관리나 기술, 외부의존보다는 기업환경, 프로젝트 관리, 요구사항과 관련된 위험 요인들이 더 중요한 것으로 생각할 수 있겠다.

4.2.3 국외 연구와의 비교

본 연구의 목적은 국내의 소프트웨어 개발 프로젝트 위험 요인의 도출이므로 외국의 경우와 비교했을 때, 어떠한 점들이 차이가 나는지를 살펴보았다. 본 연구에서는 델파이 기법을 적용하여 미국, 홍콩, 핀란드의 소프트웨어 개발 프로젝트 위험 요인을 도출한 Schmidt et al.(2001)의 연구 결과인 20개의 위험 요인과 비교해 보았다.

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 정해질 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 산정 되기 때문이며 특히 프로젝트 수주에 있어서 하청 받은 기업이 다시 하청을 주고 또 다시 하청을 주는 등의 부적합한 재하청 관행 또한 주요 이유라고 생각된다. 따라서 비용 결정에 있어 무조건 저렴하게만 구축하려 하기보다는 좀 더 현실적으로 더 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 프로젝트 기간과 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 그에 비해 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.

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 과제 수행에 예견되는 기간, 비용, 인원 등 자원 할당의 정확한 예측 부족으로 대부분의 과제수행계획서가 실제와 거리가 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 부각되지 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.

4.2.4 의견 일치도에 관한 결과

본 연구에서는 델파이 기법을 사용하여 정해진 전체 순위가 과연 얼마만큼 패널들 사이의 일치도가 이루어졌는가를 알아보는 방법으로 켄달의 일치계수 W를 사용하였다. 이는 순위에 대한 일치도를 알아보는 개념으로 Schmidt(1997)의 연구에서 제시된 해석을 기준으로 삼았으며 <표 5>에 일치계수 W에 대한 해석이 나타나있다.

<표 5> 켄달의 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

본 연구 결과에 있어서 켄달의 일치계수 W는 0.31로 나타났으며 결과 순위에 있어서 패널들의 의견이 약간 일치하며, 이러한 일치 수준에의 확신이 약간은 가능한 것으로 볼 수 있다.

V. 결론

5.1 연구의 요약

더욱 효과적으로 경쟁하기 위해 기업들은 프로젝트 지향적으로 변하고 있으며 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 완수될 때까지 위험을 지속적으로 관리해야 한다.

특히 소프트웨어 개발 프로젝트는 그 산출물의 특성상 실패 위험이 높으며 개발 과정 또한 정형적이지 않으므로 프로젝트에 있어서 어떠한 위험 요인이 관리되어야 하는지에 따른 위험 요인 자체에 관한 연구가 필요하다.

본 연구에서는 위험 중심적 접근법으로 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 <표 4>에서 보듯이 무리한 비용의 산정, 범위와 목표의 변경, 사용자의 요구사항 변동 및 불명확한 요구사항으로 확정된 요구사항의 부족, 명확치 않은 업무 범위, 무리한 프로젝트 기간의 산정순으로 나타났다.

국의 연구와의 가장 큰 차이점은 최고 경영층의 지원 부족과 무리한 프로젝트 비용의 산정이었으며 응답자들간의 의견 일치도를 알아보기 위한 켄달의 일치계수 W는 0.31로 나타나 결과 순위에 있어서 응답자들간의 의견이 약간 일치하며, 이러한 일치 수준에의 확신이 약간은 가능한 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. <표 3>의 리스트는 국외에서의 연구 결과가 아니라 국내의 프로젝트 관리자들 통해 얻어진 결과이므로 더욱 근거있고 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 연구과제

본 연구의 수행에 있어 몇 가지 한계점을 지적하면 다음과 같다.

첫째, 텔파이 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, 연구자의 경험이 부족한 관계로 1단계 응답 결과를 토대로 리스트를 만드는 과정에서 통합 및 분류 작업에 있어 부족함이 있었다. 하지만, 이러한 과정을 돕기 위해 각 항목에 그에 대한 설명을 적도록 요구하였고 생소한 부분이나 명확치 않은 부분은 응답자들에게 개별적으로 질문하여 이러한 단점을 보완하였다.

넷째, 연구자들의 의견 일치도를 높이기 위해 한번 더 조사를 실시하고 싶었으나, 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 요인을 바라보는데 어떠한 차이가 있는지도 연구할 필요가 있을 것으로 보인다. 그리고, 이러한 프로젝트 위험 요인에 대한 연구 결과를 바탕으로 이러한 각각의 위험 요인 항목에 대한 대응책은 어떠한 것이 있는지를 알아보는 연구나 상이한 이해관계자들이 위험 요인에 대해 어떠한 의견차이를 가지고 있는지에 대한 연구도 필요할 것이다.

참고 문헌

- 박운성, 현대경영학원론, 박영사, 1996.
- 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.
- 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 : Wiley, 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.
- 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 : Wiley, 2001.
- Moynihan, T., "How Experienced Project Managers Assess Risk," *IEEE Software*, Vol. 14, No. 3, 1997, pp. 35-41.
- Schmidt, R. C., "Managing Delphi Surveys Using Nonparametric Statistical 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, Upper Saddle River N. J. : Prentice Hall, 1997.