

불확실성과 소프트웨어 프로젝트 성과에 대한 국가간 비교연구

김기윤*, 나관식**, 최광돈***

*광운대학교 경영학과, **서원대학교 경영학부, ***한세대학교 IT학부

A Cross-National Comparison of Uncertainty Profile and Software Project Performance

Kim, Ki-Yoon, Na, Kwan-Sik, Choi, Kwang-Don

Kwangwoon University, Seowon University, Hansei University

E-mail : min1203@daisy.kwangwoon.ac.kr, ksna@seowon.ac.kr, kdchoi@hansei.ac.kr

요약

다양한 소프트웨어개발 위험관리 분야의 연구들 중, 미국의 소프트웨어 회사들을 대상으로 한 Nidumolu의 논문에서는 표준화와 요구사항 불확실성이 잔여성과위험이라는 매개변수를 통해서, 소프트웨어 개발프로세스와 제품 성과에 영향을 미친다는 결과를 발표하였다. 본 연구에서는 먼저, 개발도상국인 우리나라에서도 이러한 모형이 적용될 수 있는지를 검증하였는데, 결론적으로 잔여성과 위험이라는 매개변수는 모형에서 의미가 없는 것으로 나타났다. 이는 우리나라 소프트웨어 업체의 경우 성숙도가 상대적으로 낮아서, 프로젝트 관리자와 같은 개인의 능력에 성과가 영향을 받는 때문인 것으로 보인다. 따라서 본 연구에서는 이러한 차이의 원인에 대해서 논의하고, 나아가서 우리나라의 기업 환경에 적합한 새로운 모형을 제시하고자 한다.

1. 서론

소프트웨어 개발은 많은 전문가들이 하나의 프로젝트를 위해서 일하는 전형적인 협업체제이기 때문에, 매우 복잡하고 예측하기 힘든 과업이다. 이러한 개발 프로세스의 예측 불가능성은 재해가 될 수 있다. 정보시스템 관련 논문과 서적에서는, 소프트웨어개발에 있어서 어려움을 느끼게 하는 대부분의 원인이, 부적합한 정보에 기인하는 것으로 밝히고 있으며(Zmud, 1980), 이러한 소프트웨어 개발의 불확실성은 높은 소프트웨어 개발실패율과, 50%를 넘는 개발비용 초과라는 결과를 초래하고 있다(Gibbs, 1994). 사실 소프트웨어의 납기지연은 이제 산업의 관행처럼 일반화되고 있는

실정이다(Genuchten, 1991; Glass, 1997). 불확실성으로 인하여 발생하는 이러한 현상들은 원활하지 못한 의사소통이나 잘못된 정보, 갈등 등의 복합적인 요소에 기인하기도 한다. Keil and Robey (2001), Smith et al. (2001), Zhang et al. (2002) 등은 소프트웨어 프로젝트의 납기지연이나 예산초과 등에 대한 추가적인 지식과 정보를 제공해 주고 있다.

특히 Nidumolu는 표준화와 요구분석이, 잔여성과위험(residual performance risk)이라는 매개변수를 통해서, 프로세스와 제품 성과에 영향을 미친다는 모형을 제안하였다. 잔여성과위험은 소프트웨어 개발 프로젝트의 후기 단계에 존재하는 성과위험을 반영하는 변수이다. 이 모형에 의하면, 요구사항불확실성은 잔여성과위험을 증가 시키고,

소프트웨어 개발 표준화는 잔여성과위험을 감소시키며, 나아가서 잔여성과위험은 프로세스와 제품성과에 부의 영향을 미치는 것으로 가설을 설정하였다. 또한 요구사항 불확실성은 제품과 프로세스성과에 직접적으로 부의 영향을 미치는 것으로 제안 하였다. 64개의 미국 소프트웨어 개발회사의 프로젝트로부터 얻은 데이터로 분석한 결과에 의하면, 요구사항불확실성과 제품성과 간의 부의 관계에 관한 가설(H3)을 제외하고는, 이 Nidumolu의 모형과 가설들을 모두 지지하는 것으로 나타났다. Nidumolu의 결론은 미국 소프트웨어 기업들에 있어서, 잔여성과위험은 표준화와 요구사항 불확실성이 프로세스와 제품성과에 미치는 영향을 조절한다는 것이다.

이러한 관점에서 본 연구의 기본 방향은, CMM의 분류상 미국보다 대체로 낮은 단계에 머물고 있는 우리나라 기업들에 있어서, Nidumolu의 모형이 받아들여질 수 있는지를 검증하고자 하는 것이다. 이 연구를 통해서, 위험관리 전략이 소프트웨어 개발프로세스와 제품성과에, 국가에 따라서 어떻게 다른 영향을 미치는지에 대한 이해를 높일 수 있으며, 나아가서 관련이론 발전에 기여할 수 있을 것으로 기대한다.

따라서 본 논문의 첫 번째 연구목적은, Nidumolu의 모형을 검증하는 것이다. 이를 위해서 각 변수와 가설들을 정의하고, 자료수집 방법과 분석 기법에 대해서 기술하며, 통계적 분석 결과를 제시하고자 한다. 두 번째 연구목적은, 한국의 기업 환경에 적합한 모형을 제시하고, 본 연구의 시준점에 대해서 논의 하고자 한다.

2. 표본과 자료수집

본 연구에서 사용된 데이터는, 3개의 우리나라 대표적인 소프트웨어 개발회사에서 최근에 완료된 123개의 프로젝트를 대상으로 하였고, 각 회사 내의 소프트웨어 품질담당 부서의 협조 하에, 대상 프로젝트 전체에 대한 양질의 데이터를 얻을 수 있었다. 조사는 1999년과 2000년 사이에 이루어 졌으며 우편을 통한 설문지 배부와 전화면접을 병행하였다. 대상 프로젝트의 요약 정보는 <표 1>에 제시되어 있다.

<표 1> 프로젝트 특성

프로젝트 특성	평균	표준편차	최소값	최대값
프로젝트 금액 (천만원)	6,686,700	14,654,745	5	54,000,000
투입 인력 (man months)	36	46	2	200
수행 기간 (개월)	21	19	2	120

본 연구의 첫 번째 목적이, Nidumolu(1996b)의 모형을 한국의 기업 환경에서 재 검증하는 것이므로, 그의 연구와 동일한 측정 도구를 이용하였다.

3. 측정도구

본 연구에서 사용된 5가지의 이론적인 변수들에 대해서는 Nidumolu(1996b)의 모형에 정의 되어 있다. 이 중, 두 가지의 종속변수는 프로세스와 제품성과(process and product performance)이며, 프로세스성과는 소프트웨어 개발프로세스에 관한 것으로서 (1) 소프트웨어 개발 도중에 얻어지는 학습효과, (2)프로젝트 관리 정도, (3)정보시스템 부서와 사용자간의 의사소통 품질 등으로 구성되어 있다. 제품 성과는 최종 완성품의 성과에 관한 것으로 (1) 소프트웨어의 기술적인 성과, (2)사용자의 요구사항에 대한 부합정도, (3) 소프트웨어가 새로운 제품이나 사용자의 요구사항 변화에 대응할 수 있는 유연성 등의 항목들로 구성되어 있다.

독립 변수들은 요구사항 불확실성과 표준화 이다. 요구사항불확실성은 사용자의 요구사항을 식별하기 위해서 필요한 정보와, 개발자가 실제로 가지고 있는 정보사이의 차이에 대한 것이다. 이 변수는 다음의 3가지 차원으로 구성된다. (1)소프트웨어 개발 생명주기 동안에 발생하는 요구사항의 불안정성과 변화, (2) 사용자들 사이의 이질적인 요구사항, (3)사용자가 원하는 사항을 제품 요구사항으로 변환시키는 과정을, 객관적인 절차로 간소화할 수 있는 정도. 본 연구에서는 결과물 통제표준화(output control standardization)와 행위통제 표준화(behavior control standardization)를 모두 고려하고자 한다. 결과물 통제는 관리자가 사전에 설정해 놓은 각

단계별 이정표와 개발팀의 결과물을 기준으로 이루어지며, 행위 통제는 소프트웨어 개발을 위해서 관리자가 사전에 설정해 놓은 도구(tools)와 기법(techniques) 들을 기준으로 이루어진다. 매개변수인 잔여성과 위험은 프로젝트 계획수립과 요구사항 분석이 완료된 후, 프로젝트의 후기 단계에서 평가된다. 즉, 소프트웨어 개발의 후기 단계에서 성과를 예측하는데 있어서의 어려움을 의미한다.

본 논문에서는 미국의 기업들을 대상으로 연구된 것과 동일한 측정 도구와 변수들을 이용하고자 한다. 한국의 기업들을 대상으로 조사한 측정 모형은 미국에서의 연구결과와 매우 유사하다. 예를 들어서 한국기업을 대상으로 한 본 연구의 신뢰도(reliability)는 .65에서 .84이고, 미국기업을 대상으로 한 연구 결과는 .65에서 .86이다. 이 값은, 두 연구 모두 내부 일관성 유지에 대해서는 일반적으로 받아들여 질만 한 수준이다. 또한 요인분석도 두 연구가 비슷한 결과를 보이고 있다. 합성 변수(composite variable) 생성을 위해서도 미국 기업을 대상으로 한 Nidumolu의 연구와 동일한 절차를 따르고자 한다. 이 합성변수는 각 변수마다 적합한 하위 개념들에 의해 합성된다. 합성 절차는, 각 하위개념에 주성분분석에 의한 요인 적재 값을 가중치로 부여하여 산출하는 것이다.

4. 가설

본 연구에서는, Nidumolu(1996b)의 미국 소프트웨어 업체들을 대상으로 한 연구에서와 동일한 가설들을 한국 기업들을 대상으로 검증하고자 하며, 구체적으로는 다음과 같다.

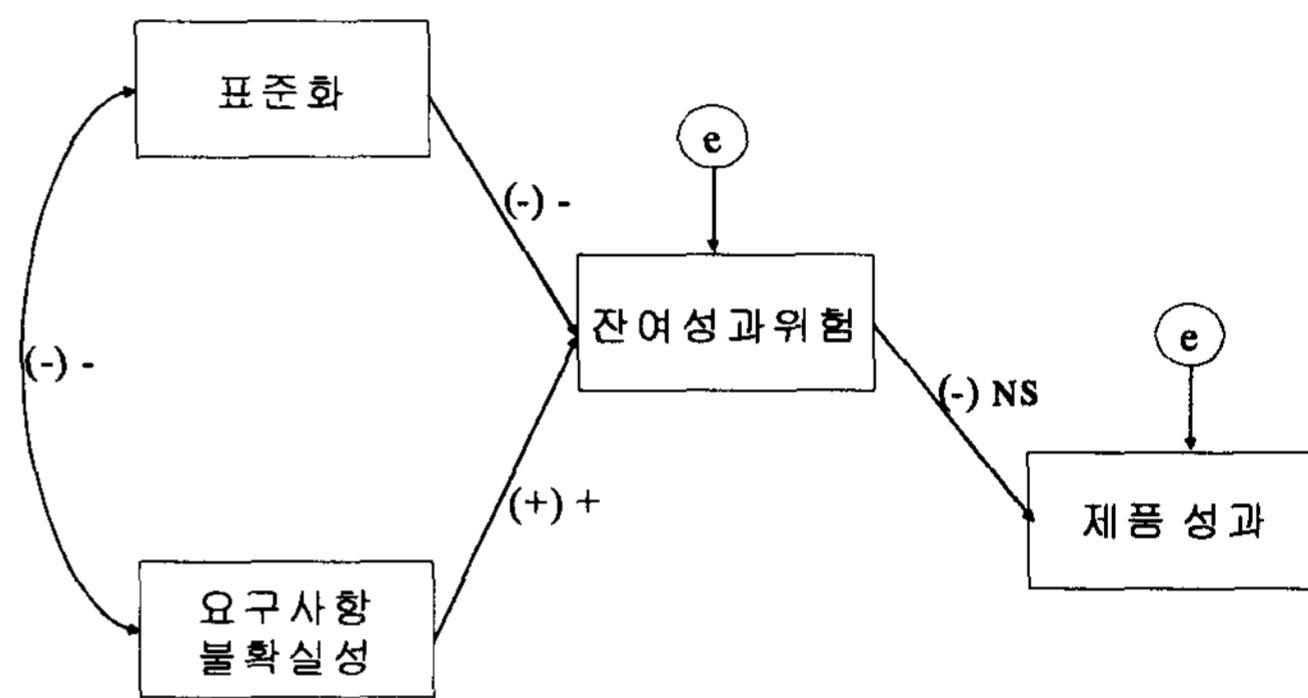
- H1: 요구사항 불확실성이 증가함에 따라 잔여성과 위험이 증가한다.
- H2: 요구사항 불확실성이 증가함에 따라 소프트웨어 개발프로세스 성과가 감소한다.
- H3: 요구사항 불확실성이 증가함에 따라 소프트웨어 제품 성과가 감소한다.
- H4: 잔여성과 위험이 증가함에 따라 소프트웨어 개발프로세스 성과가 감소한다.

H5: 잔여성과 위험이 증가함에 따라 소프트웨어 제품 성과가 감소한다.

H6: 표준화가 증가함에 따라 잔여성과 위험이 감소한다.

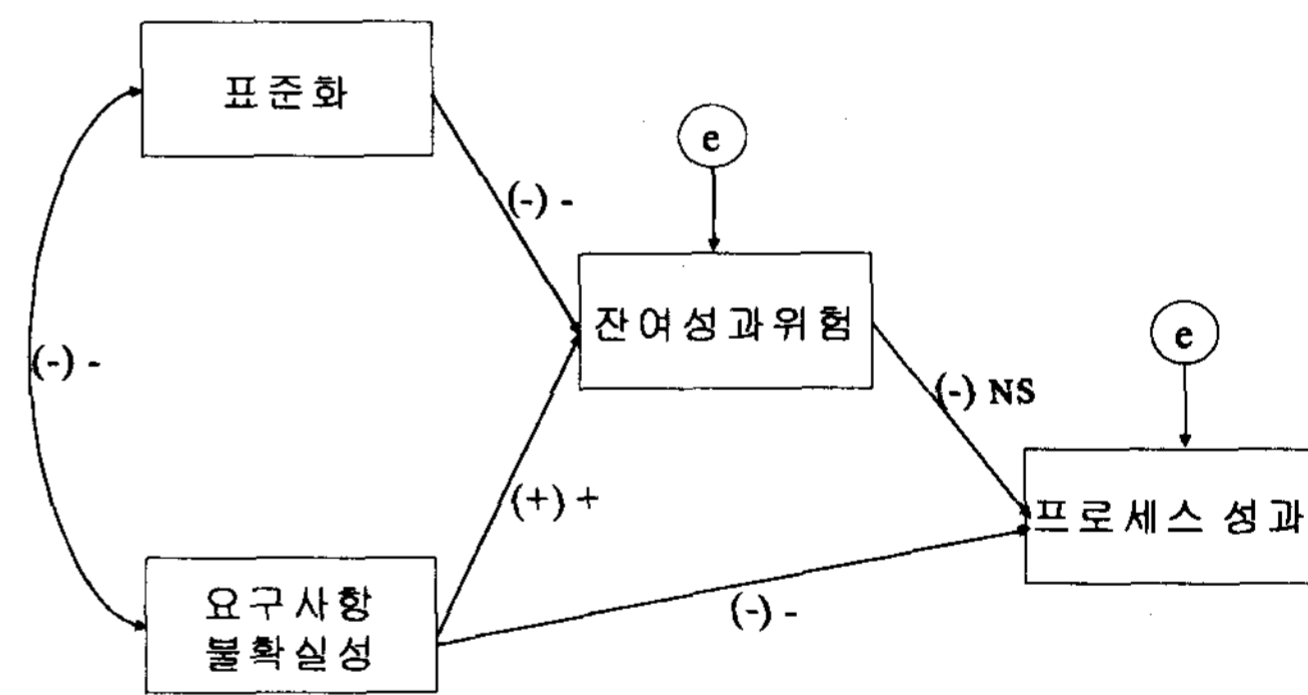
5. 검증결과

위의 가설들을 검증하기 위해서 AMOS 4.0을 이용한 구조방정식 모형(Structural Equation Model)을 이용하고자 하며, 분석 결과는 다음 <그림 2>, <그림 3>과 같다.



() 미국결과(Nidumolu 1996b) X²의 유의수준 = (0.27) 0.001
NFI = (0.92) 0.989

<그림 2> 제품성과에 대한 모형 비교



() 미국결과(Nidumolu 1996b) X²의 유의수준 = (0.96) 0.000
NFI = (0.990) 0.990

<그림 3> 프로세스성과에 대한 모형 비교

한국과 미국의 기업들에 대한 연구결과를 비교하기 위해서 미국기업의 결과(Nidumolu, 1996b)를 괄호 안에 제시하였다. 이 그림에서 + 표시는 해당 경로의 계수가 양의 유의한 값(p<0.05)을 보이고 있다는 것을 의미한다. 반대로 - 표시는 해당 경로의 계수가 음의 유의한 값(p<0.05)을 보이고 있다는 것을 의미하며, NS는 해당 경로가 유의하지 않다(non significant)는 것을

의미한다. 미국 기업을 대상으로 한 Nidumolu의 연구에서는, H3을 제외한 모든 결과가 가설에서 설정한 것과 같았지만, 우리나라 기업을 대상으로 한 연구에서는 잔여성과 위험과 두 가지의 성과변수 간에는 유의한 관계가 없는 것으로 나타났다.

즉, 미국기업을 대상으로 한 연구에서와 마찬가지로 H1, H2, H3, H6는 결과가 일치하였다. 먼저 요구사항 불확실성(H1)은 한국과 미국의 기업 모두 정의 방향으로 잔여성과 위험과 유의한 관계가 있는 것으로 나타났다. 또한 한국과 미국 모두 소프트웨어 개발프로세스 성과(H2)에는 유의한 영향을, 제품성과(H3)에는 유의하지 않은 영향을 미치는 것으로 나타났다. 표준화(H6)는 한국과 미국 모두 잔여성과 위험에 부의 유의한 관계가 있는 것으로 나타났다. 하지만 미국 기업을 대상으로 한 연구에서는, 잔여성과 위험이 소프트웨어 프로세스(H4)와 제품성과(H5)에 부의 관계가 있는 것으로 발표 되었는데, 우리나라 기업을 대상으로 한 연구에서는 유의하지 않은 것으로 나타났다.

미국 기업을 대상으로 한 Nidumolu 모형의 적합도(overall model fit)는 매우 높았다. NFI(Normed Fit Index)는 최소 권장 값인 .9를 훨씬 넘었고, 두 모형에 대한 카이스퀘어(chi-square) 값의 유의성도 임계치인 0.05보다 큰 것으로 나타났다. LM(Wald and Lagranger Multiplier) 검정 또한 모형의 적합도를 개선 시킬 수 있는 여지가 거의 없는 것으로 나타났다. 우리나라 기업을 대상으로 한 연구에서도 이와 같은 모형 적합도에 관한 검정을 시행한 결과, Nidumolu의 연구결과와는 달리 개선할 여지가 많은 모형인 것으로 나타났다. 따라서 미국 기업을 대상으로 도출된 연구모형은 우리나라 기업에 대해서는 설명력이 크게 떨어지므로 모형의 수정이 요구된다.

<그림 2>와 <그림 3>에 제시된 바와 같이, 우리나라 기업을 대상으로 한 연구에서 잔여성과 위험과 두 가지 소프트웨어 개발성과(소프트웨어 개발 프로세스성과와 제품성과) 간에는 유의한 관계가 없는 것으로 나타났다. 이에 대한 가능한 설명은, 한국 소프트웨어 업체들에 있어서 잔여성과 위험의 영향이 적다는 것이다. 잔여성과 위험의 개념은 소프트웨어 위험 평가가 성공적인

위험관리를 위해서 사전에 수행해야 할 필수적인 요소라는 사실에 기초하고 있다(Haimes, 1998). 일반적으로 소프트웨어 프로젝트에 대한 종합적인 사후 위험측정은 일련의 위험 요인들을 식별해야만 한다. 비록 많은 연구들이 이러한 위험 요소들을 식별하고자 노력하였지만, 모든 연구자들이 동의 할만한 연구결과는 아직 없었다(Barki, Rivard and Talbot, 1993; Moymhan, 1997; Jiang and Klein, 2000; Schmidt et al., 2001). 이러한 합리적인 위험 요소 리스트를 만드는데 있어서 가장 큰 문제점은 프로젝트 관리자와 사용자들이 위험 요인에 대해서 각기 다른 인식을 가지고 있다는 것이다(Keil, Tiwanna and Bush, 2002)

이러한 전통적인 위험관리 접근법들의 한계점 때문에 연구자들은 프로젝트의 불확실성 측면에 대해서, 보다 발전된 기법들을 개발하게 되었다(Meyer, Loch and Pitch, 2002). 이들 연구자들은, 혁신적인 프로젝트들이 전통적인 위험관리 기법으로 통제할 수 있는 예측 가능 불확실성이외에, 예측 불가능한 불확실성과 카오스(chaos)를 가지고 있다고 주장 하였다. 이러한 연구자들의 관점에 입각해서, 본 연구에서는 잔여성과 위험을 다음과 같이 두가지 부분으로 분해 하고자 한다.

잔여성과 위험 = 잔여 통제가능 위험 + 통제 불가능 위험

잔여 통제가능 위험은 프로젝트의 후기 단계에서 계속적으로 존재하지만, 다양한 방법으로 통제 가능한 위험이다. 통제 불가능 위험은 예측할 수 없거나 통제 불가능한 위험이다. 이러한 분해 공식을 기초로 해서 한국 기업들의 잔여성과 위험이 미국 기업들 보다 더 클 것이라고 가정할 수 있다. 먼저 많은 연구 결과(문대원 외, 1998; 나관식, 1999)들에서 나타난 바와 같이, 한국의 소프트웨어 개발 업체들은 전반적으로 미국 기업들에 비해 기술 수준이 현저히 낮은 것으로 알려져 있다. 이들 연구결과에 의하면 조사시점 기준으로 우리나라 대부분의 업체들이 CMM(Capability Maturity Model)의 1 또는 2단계에 속해 있는 것으로 나타났다. CMM은 카네기 멜런 대학의 SEI(Software Engineering Institute)에서 개발된 종합적인 소프트웨어 프로세스 개선 모형으로서,

세계적으로 널리 이용되고 있다. 이 모형은 소프트웨어 개발 조직을, 공학과 관리 측면의 기술수준을 기준으로하여, 5개의 등급으로 분류한다. 특정 조직이 체계적인 소프트웨어 공학 도구를 가지고 있지 않으면 대개 1또는 2단계에 속하는데, 이들 조직의 소프트웨어 개발 성과는 관리자의 경험등과 같은, 표준화 되지 않은 요인에 의존하게 된다(Paulk, 2001; Paulk et. Al., 1995).

만약, 소프트웨어 개발의 후기단계에서 위험을 통제할 만한 대책이 있다면, 잔여성과 위험은 프로젝트 성과에 직접적으로 영향을 미치지 않을 것이다. 예를 들어서 프로젝트 관리자의 경험이, 프로젝트 관리 도구의 미흡한 점을 보상할 수 있다(Henderson and Lee, 1992; Moymhan, 1997). 나아가서 프로젝트의 전체적인 조정도 프로젝트 개발 성과를 개선 시킬수 있다(Andres and Zmud, 2002; Nidumolu, 1996a). Nidumolu(1006b)는 미국 기업들을 대상으로 한 그의 연구에서, 후기 단계에서의 이러한 조정이나 관리자의 경험 등과 같은 대책들을 고려하지 않았다. 따라서 우리나라 기업을 대상으로 한 본 연구에서 Nidumolu(1996b)가 제시한 모형의 적합도가 낮은 것은, 이러한 후기 단계에서 잔여성과 위험을 감소 시키고자 하는 노력들에 기인 한 것이라고 해석하는 것이 합리적일 것이다.

이들 관련 연구 결과들을 토대로 대안적인 모형을 제시 하고자 한다. 즉, 표준화와 요구사항의 불확실성이, 이들 두 가지 소프트웨어 프로젝트 성과에 직접적으로 영향을 미친다는, 수정 모형을 본 연구에서 제안 하고자 한다.

미국과 한국 기업을 대상으로 한, 두 연구결과에서 명확하게 검정 된 결과에 따라, 원래의 모형에서 잔여성과위험 변수를 제거 하였다. 또한 두 가지 성과변수에 의해 만들어진 두 모형을 하나로 통합하였다. 이 수정된 모형은 프로세스 성과가 결과적으로 제품성과에 영향을 미친다는, 소프트웨어 생명주기 모형(ISO/IEC, 1995)의 흐름에 바탕을 두고 있다. 요구사항 불확실성은 소프트웨어 제품성과에 영향을 미칠 수 있기 때문에, 본 연구에서 그 영향을 측정 하고자 한다. 하지만 요구사항 불확실성은 소프트웨어 프로세스 성과에 정과 부, 양측 모두에 영향을 미칠 수 있기 때문에,

본 연구에서는 고려하지 않는다. 예를 들어서 높은 요구사항 불확실성은 일정 지연이나 예산 초과지출 등의 형태로 프로세스 성과에 부의 영향을 미치는 동시에, 이를 통해서 사용자와 개발자 간의 의사소통을 촉진하고 학습의 기회를 증대 시킴으로써, 프로세스 성과에 정의 영향을 미칠 수가 있다. 이러한 논의 결과를 기초로 하여 다음과 같은 가설들이 도출 되었다.

H7: 표준화가 증가 될수록 소프트웨어 프로세스 성과가 증가될 것이다.

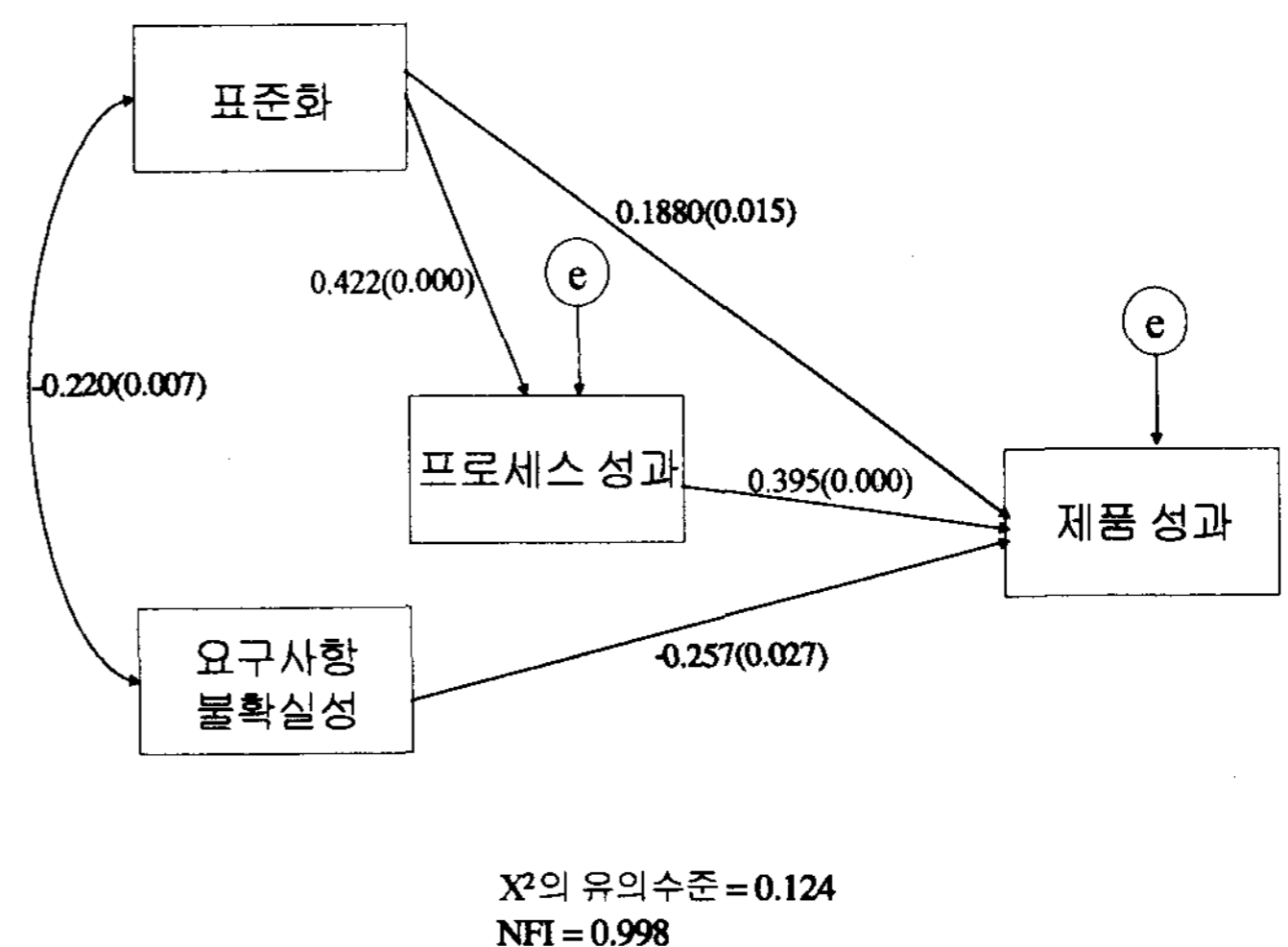
H8: 표준화가 증가될수록 소프트웨어 제품 성과가 증가될 것이다.

H9: 요구사항 불확실성이 커질수록 소프트웨어 제품성과가 커질 것이다.

H10: 소프트웨어 프로세스 성과가 증가할수록 제품 성과도 증가할 것이다.

7. 수정모형의 분석

수정 모형에 대한 χ^2 검증이 유의하지 않은 것으로 보아(p-value=0.124), 예측된 공분산 매트릭스가 표본의 공분산 매트릭스와 다르지 않다는 것을 알 수 있다, 나아가서 NFI(Normed Fit Index) 값도 0.998로 일반적인 최소권장 기준 .90을 넘었고, 추가적인 모형 적합도 값들도, 본 모형이 데이터를 잘 표현하고 있다는 것을 보여주고 있다. 이들 결과는 이 수정 모형이 원래의 모형보다 더 우수하다는 것을 의미한다.



<그림 4> 수정모형의 검정 결과

<그림 4>에서 제시된 계수 값들은, 본 수정 모형에, 사용된 변수들의 관계를 잘 나타내 주고

있으며, 각 계수의 유의도는 괄호 안에 제시되어 있다. 이 검정 결과는 H7, H8, H9, H10을 모두 강하게 지지하고 있다. 즉, 표준화가 증가 될수록 소프트웨어 개발 프로세스와 제품의 성과가 높아진다. 요구사항 불확실성의 증가는 제품 성과를 감소시키며, 프로세스 성과가 증가함에 따라 제품성과도 증가하는 것으로 나타났다.

이들 결과는, 미국과 같은 선진국과는 달리, 우리나라와 같은 신흥 정보통신국가에서는 본 연구에서 제시된 수정 모형이 보다 더 적합하다는 것을 보여주고 있다. 요구사항 불확실성과 표준화는 전통적인 소프트웨어 위험관리 도구로 사용되고 있지만, 예측 불가능한 불확실성이나 혼돈(chaotic) 상황에서는 일반적으로 적합하지 않은 것으로 알려져 있다. 따라서 기술 혁신적이고 환경이 가변적인 소프트웨어 개발 프로젝트의 관리자는, 이러한 전통적인 위험관리 도구 이외에, 보다 예측능력이 뛰어난 접근법들도 적극적으로 활용하여야 할 것이다.

8. 결론

본 연구는, 정보기술 수준이 위험관리기법과 위험, 프로젝트 성과 등의 관계에 어떠한 영향을 미치는지에 관한, 기본적인 의문점들의 해답을 얻기 위한 것이다. 이를 위해서 우리나라에서 수집된 자료를 구조방정식 모형을 사용하여, 미국의 연구결과와 비교분석 하였다. 동일한 분석 기법을 이용한 본 연구에서, 미국의 기업들을 대상으로 제시된 모형은 한국의 데이터에서는 더 이상 적합하지 않다는 것을 보여 주었다. 나아가서 잔여성과위험의 평균값과 프로젝트 성과에 미치는 영향은 양국이 서로 다른 것으로 나타났다. 본 논문에서는 이러한 연구 결과를 토대로, 이 같은 결과가 정보기술 능력의 차이에서 기인한 것이라는 것을 검정 하였다.

미국의 기업들과 비교해 볼 때, 우리나라 기업들의 소프트웨어 개발프로젝트에 대한 위험관리 수준은 아직 미흡한 것으로 알려져 있다. 따라서 우리나라 기업 환경에서는 중요한 변수로 인식되어야 할 프로젝트 관리자의 경험과 같은, 기타의 위험통제 수단들을 고려하지 않은 것이, 이러한 양 국가의 연구결과 차이를 가져온 것으로 볼 수 있다.

본 연구에서 제시된 수정모형은 이같은 이유에 의해, 잔여성과위험 변수를 제거한 것이다. 이 수정 모형은 한국의 데이터로 검증 되었다. 그 결과에 의하면 소프트웨어 개발 표준화는 소프트웨어 개발 프로세스와 제품 성과 모두를 향상 시키는 것으로 나타났다. 또한, 요구사항 불확실성은 소프트웨어 제품성과를 개선시키며, 최종적으로 소프트웨어 개발 프로세스 성과는 제품성과에 정의 영향을 미치는 것으로 나타났다.

[참고문헌]

- [1] 문 대원 외 (1997), 국가 정보화 촉진을 위한 품질정책 연구, '97한국전산원 연구보고서, 한국전산원
- [2] 나 관식 (1999), 소프트웨어 산업의 경쟁력 강화를 위한 품질정책 연구, 정보통신 학술연구과제, 정보통신부
- [3] Andres, H., Zmud R. (2002), A Contingency Approach to Software Project Coordination, *Journal of Management Information Systems*, 18, 3, 41-70.
- [4] Barki, H., Rivard, S., Talbot J. (1993), Toward an assessment of software development risk, *Journal of Management Information Systems*, 10, 2, 203-225.
- [5] Boehm, B. (1991), Software Risk Management: Principles and Practices, *IEEE Software*, 8, 1, 32-41.
- [6] Chittister, C., Haines, Y. (1994), Assessment and management of software technical risk, *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 2, 187-202.
- [7] Chittister, C., Haines, Y. (1996), System integration via software risk management, *IEEE Transactions on Systems, Man, and Cybernetics*, 26, 5, 521-532.
- [8] Fairley, R. (1994), Risk management for software projects, *IEEE Software*, 11, 5, 57-67.
- [9] Genuchten, M. (1991), Why is software late? An empirical study of the reason for delay in software development, *IEEE Transactions on Software Engineering*, 17, 6, 582-90.
- [10] Gibbs, W. (1994), Software's chronic crisis, *Scientific American*, 271, 3, 86-95.

- [11] Glass, R. (1997), Software runaways-some surprising findings, *The DATA BASE for Advances in Information Systems*, 28, 3, 16-19.
- [12] Haimes, Yacov Y. (1998), *Risk Modeling, Assessment and Management*, John Wiley & Sons Inc.
- [13] Henderson, J., Lee S. (1992), Managing I/S design teams: a control theories perspective, *Management Science*, 38, 6, 757-77.
- [14] Hofstede, G. (1980), *Culture's Consequences*, Sage Publications, Newbury Park.
- [15] Huchzermeier, A., Loch C. (2001), Project management under risk: using the real options approach to evaluate flexibility in R&D, *Management Science*, 47, 1, 85-101.
- [16] ISO/IEC (1995), *ISO/IEC 12207 Information Technology - Software - Part 1 : Software Life Cycle Processes*, First edition
- [17] Keil, M., et al. (2000), A cross-cultural study on escalation of commitment behavior in software projects, *MIS Quarterly*, 24, 2, 299-325.
- [18] Keil, M., Robey, D. (2001), Blowing the whistle on troubled software projects, *Communications of the ACM*, 44, 4, 87-93.
- [19] Keil, M., Tiwana, A., Bush A. (2002), Reconciling user and project manager perceptions of IT project risk: a Delphi study, *Information Systems Journal*, 12, 2, 103.
- [20] Kitchenham B., Linkman S. (1997), Estimates, uncertainty and risk, *IEEE Software Magazine*, May/June, 69-74.
- [21] Lyytinen, K., Mathiassen L., Ropponen J. (1998), Attention shaping and software risk - a categorical analysis of four classical risk management approaches, *Information Systems Research*, 9, 3, 233-255.
- [22] Meyer A., Loch C., Pich M. (2002), Managing project uncertainty: from variation to chaos, *Sloan Management Review*, 60-67.
- [23] Moymhan, T. (1997), How experienced project managers assess risk, *IEEE Software*, 14, 3, 35-41.
- [24] Montealegre, R., Keil, M. (2000), De-escalating information technology projects: lessons from the Denver International Airport, *MIS Quarterly*, 24, 3, 417-447.
- [25] Nidumolu, S. (1995), The effect of coordination and uncertainty on software project performance: residual risk as an international variable, *Information System Research*, 6, 3, 191-219.
- [26] Nidumolu, S. (1996a), A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects, *Journal of Management Information Systems*, 13, 2, 77-113.
- [27] Nidumolu, S. (1996), Standardization, requirements uncertainty and software project performance, *Information & Management*, 31, 135-150.
- [28] Paulk, M., et al. (1995), *The Capability Maturity Model: Guidelines for Improving Software Process*, Addison-Wesley Publishing.
- [29] Paulk, M. (2001), Extreme programming from a CMM perspective, *IEEE Software*, 18, 6, 1-8.
- [30] Ropponen, J. (1999), *Software Risk Management-Foundations, Principles and Empirical Findings*, Jyvaskyla University Printing House.
- [31] Schmidt, R., et al. (2001), Identifying software project risks: an international Delphi study, *Journal of Management Information Systems*, 17, 4, 5-36.
- [32] Smith, H., et al. (2001), Keeping mum as the project goes under: towards an explanatory model, *Journal of Management Information Systems*, 18, 2, 189-228.
- [33] Trigeorgis, L. (1996), *Real Options, Managerial Flexibility and Strategy in Resource Allocation*, The MIT Press, Cambridge, Massachusetts.
- [34] Zhang, P., et al. (2002), Predicting information technology project escalation: a neural network approach, *European Journal of Operations Research* (in press).
- [35] Zmud, R. (1980), Management of large software development efforts, *MIS Quarterly*, 4, 2, 45-55.