

요구사항 불확실성, 통제 표준화, 상호작용이 소프트웨어 품질에 미치는 영향*

김기윤**, 나관식***, 양동구****

The Effects of Requirement Uncertainty, Control Standardization, Interactions on Software Quality

Ki-Yoon Kim, Kwan-Sik Na, Dong-Gu Yang

Risk management or risk-based approach of software project management was developed to explain the effects of requirement uncertainty, control standardization, interactions on software quality. Based on a prior theory, five hypotheses were derived and empirically tested using a survey design. Data from 117 members in 3 SI companies support for the path model, and three of five hypotheses.

The results showed that decreases in requirement uncertainty and increases in control standardization were directly associated with increases in the interactions between user and project teams, which, in turn, led to increases software quality. The findings suggested that the direct effect on software quality is primarily due to the interactions between user and project teams, rather than requirement uncertainty and control standardization.

* 이 논문은 2002년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

** 광운대학교 경영학과 교수, E-mail: min1203@daisy.kwangwoon.ac.kr

*** 서원대학교 경영정보학과 부교수, E-mail: ksna@seowon.ac.kr

**** (주)우린정보 IR 팀장, E-mail: dgyang@woorin.co.kr

I. 서론

전통적으로 소프트웨어 개발 프로젝트는 기능적 행위로만 생각했었고, 프로젝트 팀에서 숙련된 기술자들이 그들의 개발도구 및 방법을 통제했었다. 그러므로, 소프트웨어 개발조직이 미성숙할수록 소프트웨어 개발 프로젝트의 비용, 일정, 품질 등에서 소프트웨어 개발위험이 높아졌다. 소프트웨어 개발 프로젝트에 투입된 조직은 사용자의 요구사항에 따라 소프트웨어와 문서를 생산하는 임시조직이다. 연구초점은 주어진 예산 내에서 적시에 완성해야하는 소프트웨어 개발프로젝트의 진행과정에서 존재하는 위험들이다. 이러한 위험에는 불확실성과 손실이라는 두 가지 특성이 있다. 여기서 불확실성은 조직 내에서 완전한 정보의 부재를 의미하며, 손실은 원치 않는 결과를 의미한다.

Carnegie Mellon University의 SEI(Software Engineering Institute)에서는 소프트웨어 개발위험을 제품공학, 개발환경, 프로그램 제약으로 분류했다. 제품공학의 요소는 요구사항, 설계, 코드 및 단위시험, 통합 및 시험, 공학 전문가들이고, 개발환경의 요소는 개발과정, 개발시스템, 관리과정, 관리방법, 작업환경이고, 프로그램 제약의 요소는 자원, 계약, 프로그램 인터페이스(program interface)이다. 이러한 개별 요소들을 또 다시 여러 가지 속성으로 세부분류를 했다. 이와 같은 소프트웨어 개발프로젝트의 위험요소들 중에서 사용자 요구사항의 불확실성과 프로젝트의 절차나 기술에 대한 표준화가 프로젝트의 성과인 소프트웨어 품질에 미치는 영향은 매우 크다.

시스템 개발을 위한 사회 기술적 모형의 4가지 구성요소는 과업, 기술, 행위자, 구조이다. 여기서 과업은 도달해야 할 목표이고, 기술은 문제해결 도구이고, 행위자는 소프트웨어 개발에 의해서 이득을 얻게 되는 개인 및 집단이고, 구조는 의사소통, 권한체계, 작업흐름체계 등이다. 소프트웨어 개발 프로젝트에서 과업 측면에서 사

용자 요구사항을 충족시키는 소프트웨어 품질이 중요하고, 기술 측면에서는 통제 표준화가 중요하고, 행위자 및 구조 측면에서는 사용자와 프로젝트팀간의 상호작용이 중요하다. 본 연구의 목적은 소프트웨어 개발프로젝트의 위험관리 관점에서 요구사항 불확실성과 통제 표준화가 상호작용과 소프트웨어 품질에 미치는 영향을 분석하는 것이다. 또한, 사용자와 소프트웨어 개발프로젝트 팀간의 상호작용이 소프트웨어 품질에 미치는 영향을 분석하는 것이다.

II. 이론적 배경과 연구가설

2.1 이론적 배경

소프트웨어 개발위험 혹은 소프트웨어 위험(software development risk or software risk)에 관한 연구는 의사결정 및 위험관리(혹은 위험기반 접근방법(risk-based approach)) 관점에서 연구되었다. 의사결정 관점에서의 연구[Boehm and Ross, 1989; Charette, 1989; Boehm, 1991]는 가능한 손실을 줄이고 이익을 늘리는 게임으로 일어날 수 있는 결과에 대한 확률분포를 고려해야 하므로, 확률분포를 추정해야 하는 모호한 측면이 있다. 이와 같은 확률추정은 단지 결과를 추정하는 단순한 접근방법에 지나지 않는다. 이 경우에 위험은 일반적으로 부(-)의 효과에 대한 확률과 강도(severity)의 크기로 정의되고 있다. 또 다른 방법은 위험을 기술하기 위해서 퍼지 집합 이론을 이용해서 위험을 측정하는 접근방법이 있다[Kangari and Boyer, 1989].

소프트웨어 위험에 대한 위험관리연구 중 고전적인 연구[Lyytinen, Mathiassen and Ropponen, 1998]는 다차원적이고 정성적인 사회 기술적 모형으로서 다음과 같이 네 가지 접근방법으로 구분할 수 있다. 첫째, Alter와 Ginzberg[1978]의 실행 접근방법은 위험을 "프로젝트 완료에 대한 불확실성"으로 정의했고, 다차원 구성개념으로

8개의 위험항목(설계자의 경험부족, 부적절한 사용자, 다중 이용자 혹은 설계자, 사용자 혹은 설계자 혹은 유지보수자의 부재, 지원의 결여 혹은 손실, 목적 혹은 이용 유형을 사전에 명시하지 못한 것, 예기치 못한 충격, 기술적 혹은 비용 효과적 문제)을 명목적으로도 측정했다. 둘째, Davis [1982]의 상황적합 접근방법은 위험을 “정확하고 완전한 사용자 요구사항을 충족시키는 것에 대한 어려움”으로 정의했고, 요구사항 명세화에 대한 3개의 위험항목(요구사항의 안전성, 사용자의 요구사항 명세화 능력, 시스템 분석자의 요구사항 도출 및 평가 능력)을 서열적으로도 측정했다. 셋째, McFarlan[1982]의 포트폴리오 접근방법은 위험을 “수익획득의 실패, 계획수준을 초과한 실행 비용 및 시간, 추정치 이하의 기술적 성과, 시스템과 하드웨어 및 소프트웨어의 불일치”로 정의했고, 프로젝트 관리에 대해서 3개의 위험항목(프로젝트 크기, 기술경험, 프로젝트 구조)을 서열적으로도 측정했다. 넷째, Boehm[1991]의 소프트웨어 위험 접근방법은 위험을 “불만족스런 결과의 발생확률과 손실을 곱한 것”으로 정의했고, 다차원 구성개념으로 10개의 위험항목(인력부족, 비현실적인 일정 및 예산, 잘못된 기능 개발, 잘못된 사용자 인터페이스(user interface) 개발, 비용분석 오류, 연속적인 요구사항 변화, 외부 공급부품의 부족, 외부 수행 과업의 부족, 실시간 성과 부재, 무리한 컴퓨터 기법 이용)을 명목적으로도 측정했다.

이러한 위험관리 접근방법 중에서 McFarlan [1981], Bohem[1989], Barki[1993] 등은 특히 프로젝트 성과와 관련된 위험인 성과위험에 연구의 초점을 두었다. 소프트웨어 위험에 대한 위험관리연구 중 최근연구인 Nidumolu[1996]의 위험기반 접근방법에 의하면, 구조적 상황적합(structural contingency) 관점에서 요구사항의 불확실성과 표준화가 프로젝트 성과에 영향을 준다고 했다. 이러한 소프트웨어 개발위험을 관리하기 위해서는 지속적인 위험관리(CRM; Continuous

Risk Management)가 필요하다. 위험관리 패러다임은 프로젝트 생명주기 동안 연속적인 행위들로 파악되는 일련의 기능들로서 위험에 대한 식별 → 분석 → 계획 → 추적 → 통제 → 의사소통에 대한 지속적인 관리이다. 여기서 식별은 프로젝트에 대한 불확실성을 기술하고 측정할 수 있는 명확한 위험으로 전환시키는 과정이다. 분석은 위험의 속성을 평가하고 분류해서, 위험한 정도에 대해서 순위를 결정하는 과정이다. 계획은 위험에 대해서 무엇을 해야만 하는가를 결정하는 기능이다. 추적은 위험추적에 책임이 있는 사람이 위험자료를 수집해서 집계하는 과정이다. 통제는 위험추적에 대한 보고서를 근거로 위험을 완화시키는 의사결정을 효과적으로 실행하는 과정이다. 의사소통은 사용자와 소프트웨어 개발 프로젝트 팀 간의 상호작용으로서 프로젝트에 참여하는 구성원들 간에 위험자료를 이해하고, 프로젝트의 제약된 자원 내에서 위험을 완화시키는 적절한 대체안들을 이해하는 과정이다. 이와 같은 지속적인 위험관리(CRM) 관점에서 식별 → 분석 → 계획은 사용자 요구사항 불확실성을 감소시키는 단계이고, 추적 → 통제는 통제 표준화를 증가시키는 단계이고, 궁극적으로 의사소통은 사용자와 소프트웨어 개발 프로젝트 팀 간의 상호작용을 증가시켜서 소프트웨어 품질을 높이게 할 것이다.

2.2 연구가설의 설정

2.2.1 요구사항 불확실성이 상호작용과 소프트웨어 품질에 미치는 영향

소프트웨어 개발 프로젝트에 대한 사회 기술적 모형에서 과업은 프로젝트팀이 달성해야하는 목표로서 궁극적으로 사용자 요구사항을 충족시키는 것이다. 일반적으로 과업과 관련된 성질에는 과업 불확실성과 과업 복잡성이 있다. 과업 불확실성은 과업관련정보의 가용성, 타당성, 신뢰성으

로 정의되고, 과업 복잡성은 과업을 수행하는데 필요한 관련 정보의 양으로 나타난다[Lyytinen, Mathiassen, and Ropponen, 1998]. 소프트웨어 개발 프로젝트에서 과업 불확실성은 요구사항 불확실성으로 나타나고, 이러한 불확실성을 낮추기 위해서 사용자-프로젝트팀 간의 상호작용이 실행된다. Walsh와 Schneider[2000]의 행위자 영향 모형(actor influence model)에 의하면, 프로젝트 팀인 프로젝트 리더, 시스템 분석자, 프로그래머가 최종사용자와 서로 상호작용을 하면서 영향을 주고받아서, 위험과 보상을 균형 시키는 의사결정을 유도한다고 했다. Jones와 Harrison[1996]에 의하면, 이러한 프로젝트팀의 성과에 영향을 주는 요인들은 사용자 요구사항에 대한 이해와 시스템 설계에 대한 팀 구성원들의 몰입이라고 했다.

Lawrence와 Low[1993]는 정보시스템 전문가가 아닌 현업의 사용자 집단의 대표자가 개발 전 과정에 주도적으로 참여하여 정보시스템을 구현하는 경우를 사용자 주도개발(user-led development)이라고 했다. 그러나, 일반적으로 현업 사용자들은 구현 이후에 시스템 사용에서는 주체적인 역할을 하지만, 개발과정에서는 개발 일부단계에서만 소극적으로 참여하게 된다[Cotterman and Kumar, 1989]. 우리나라의 소프트웨어 개발 프로젝트 경우에는 대부분 사용자 주도 개발보다는 사용자-프로젝트 팀 간의 상호작용에 의해서 이루어지고 있다. Nidumolu[1996a]는 사용자 요구사항의 불확실성이 감소할수록 수직적 조정 및 수평적 조정이 증가해서 공정통제와 제품유연성이 증가한다고 했다. 이를 반대의 의미로 표현하면, 요구사항 불확실성이 높을수록 사용자-프로젝트 팀 간의 상호작용은 낮아져서, 그 결과로 프로젝트 성과는 낮아질 것이라는 추론이 가능하다.

가설 1: 요구사항 불확실성이 높을수록 사용자-

프로젝트 팀 간의 상호작용은 낮아진다.

정보시스템의 성과는 많은 학자들에 의해서 다차원적인 개념으로 연구되어 왔다. DeLone과 McLean[1992]은 정보시스템 성공의 차원을 시스템의 품질, 정보의 질, 사용도, 사용자 만족도, 개인에게의 영향, 조직에게의 영향 등으로 구분하였다. 소프트웨어 프로젝트의 성과인 소프트웨어 품질의 국제표준으로서 제품 품질에는 ISO/IEC 9126, ISO/IEC 14598 등이 있고, 프로세스 품질에는 ISO/IEC 12207, SEI CMM, ISO/IEC 15504(SPICE) 등이 있다. ISO/IEC 9126의 품질 특성은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성(portability)이다. ISO/IEC 9126은 소프트웨어 품질을 계량적으로 측정하기 위한 품질 특성을 정의한 반면에, ISO/IEC 14598은 ISO/IEC 9126의 사용을 위한 절차와 상황을 규정하고 있다. ISO/IEC 12207의 생명주기 프로세스는 5가지 기본 프로세스, 8가지 지원 프로세스, 4가지 조직 프로세스로 구분한다. CMM은 조직의 성숙도에 따라서 5단계 별로 핵심 프로세스 영역을 구분한다. SPICE에서 프로세스 차원은 고객-공급자, 조직, 관리, 공학, 지원 등 5개영역으로 구분되고, 능력 차원은 6개 수준으로 구분된다. 예로서, 요구사항관리는 ISO/IEC 12207의 지원프로세스의 문서화와 관련되어 있고, CMM의 성숙도 수준 2단계에, 그리고 SPICE의 고객-공급자 영역에 해당된다.

Nidumolu[1996b]는 생명주기 관점에서 소프트웨어 개발 프로젝트의 성과를 과정성과 제품성으로 구분했다. 과정성은 프로젝트 중에서 획득되는 학습, 과정통제, 상호작용의 품질로서 분류했고, 제품성은 소프트웨어 운영유연성, 소프트웨어 효율성, 소프트웨어 반응성 등으로 분류했다. 소프트웨어 개발 프로젝트의 제품성 중에서 소프트웨어 운영유연성이란 업무변화를 수용할 수 있는 소프트웨어의 전반적인 유연성의 정도이고, 소프트웨어 운영효율성이란 소

프웨어 운영과 관련된 신뢰성, 운영비용, 사용 용이성 등 기술적 효율성의 정도이고, 소프트웨어 반응성이란 사용자에게 제공되는 산출물의 변경능력, 범위, 적응성 등의 정도이다. 이러한 소프트웨어 품질에 영향을 많이 주는 위험원천으로는 사용자 요구사항, 통제 표준화, 사용자-프로젝트팀 간의 상호작용 등이 있다. Chittister와 Haimes[1996]는 시스템 통합관점에서 소프트웨어 위험의 원천을 HHM(Hierarchical Holographic Modeling)을 이용해서 소프트웨어 개발, 임시적 리더쉽, 환경, 구입, 품질, 기술로 구분했다. 이 중에서 소프트웨어 개발위험의 원천에는 사용자 요구사항, 명세서, 구조, 과정, 제품, 통합을 위한 지원시스템 등의 요소들이 있다. 소프트웨어 개발위험의 원천 중에서 가장 중요시되는 요구사항 불확실성은 사용자 요구사항과 관련된 정보의 결핍을 의미하며, 이러한 위험이 높을수록 프로젝트 성과인 소프트웨어 품질은 물론, 개발비용과 개발완료시간에 부정적인 영향을 줄 것이다. Nidumolu[1996b]는 요구사항 불확실성이 잔여성과위험을 증가시켜서 프로젝트의 성과를 감소시킨다고 했다. 이와 같이 소프트웨어 위험 중에서 특히 요구사항 불확실성이 높을수록 프로젝트 성과인 소프트웨어 품질은 낮아질 것을 예상하여 다음과 같은 가설을 설정하였다.

가설 2: 요구사항 불확실성이 높을수록 소프트웨어 품질은 낮아진다.

2.2.2 통제표준화가 상호작용과 소프트웨어 품질에 미치는 영향

공식적 통제에 대한 개념은 조직의 대리이론(agency theory)에서 성과평가 전략을 위해서 제시되었고[Eisenhardt, 1985], 행위통제(behavioral control)와 산출통제(outcome control)로 구분된다[Eisenhardt, 1989; Klepper, 1990]. 실증적 연구에 의하면, 행위통제는 적절한 행위들이 알려

져 있거나, 혹은 통제하는 사람이 통제 받는 사람들을 관찰할 수 있을 때 실행된다고 했다. 산출 통제는 산출이 측정될 수 있을 때, 즉 통제하는 사람이 목표 도달여부를 평가할 수 있을 때 실행된다고 했다[Eisenhardt, 1985; Kirsch, 1997]. Kirsch[1997]는 정보시스템 개발(ISD: Information System Development) 프로젝트에 대한 통제를 크게 공식적인 통제와 비공식적인 통제로 구분한 다음, 또 다시 공식적인 통제는 행위통제와 산출통제로, 비공식적인 통제는 집단수준의 파벌통제(clan control)와 개인수준의 자기통제(self-control or self management)로 구분하였다. 사례연구에 의해서 시스템 개발 중에 공식적인 통제가 적절히 실행되어야 하고, 특정의 통제는 책무의 특성, 역할 기대, 프로젝트 관련 지식과 기술에 따라서 실행된다고 했다.

Phan, Vogel and Nunamaker[1995]의 사례 연구에 의하면, 사용자 요구사항을 분류해서 상대적 중요도 순으로 순위화해서 프로젝트 요구사항으로 변경시키고, 자원통제전략을 실행시킨 후에, 소프트웨어 품질보증을 위해서 인적자원을 관리해야 한다고 했다. McFarlan[1982]은 프로젝트 관리와 관련된 프로젝트 3가지 위험(규모, 기술경험, 프로젝트 구조)을 줄이기 위한 공식적인 통제들을 제시했다. 특히 Davis[1982]는 요구사항 불확실성(사용자 요구사항의 안전성, 명세성, 분석성 등)을 줄이기 위한 공식적인 통제들(계획대비 실적보고서, 변경통제원칙, 이정표 제시 회의, 계획 이탈, 등)을 제시했다. 또한, Nidumolu[1996b]는 요구사항 불확실성과 통제 표준화 간의 음의 상관관계가 통계적으로 유의하다고 했다. 이와 같이 요구사항 불확실성이 높을수록 통제표준화가 낮아지고, 반대로 통제표준화가 높을수록 요구사항 불확실성은 낮아질 것이다. 소프트웨어 개발 절차나 기술을 위한 표준화된 도구와 기술을 이용하는 정도를 나타내는 통제표준화의 행위 및 산출 통제를 높일수록,

프로젝트 팀에 대한 사용자의 몰입을 증가시키는 상호작용이 높아질 것이다. Nidumolu[1996b]는 통제 표준화가 높을수록 잔여성과위험이 낮아져서 프로젝트의 성과는 높아진다고 했는데, 이와 같이 프로젝트 성과가 높아지도록 위험을 감소시키는 의사소통 과정이 바로 사용자-프로젝트 팀 간의 상호작용이다. 그러므로, 통제 표준화가 높을수록 사용자-프로젝트 팀 간의 상호작용은 높아질 것으로 예측된다.

가설 3: 통제 표준화가 높을수록 사용자-프로젝트 팀 간의 상호작용은 높아진다.

소프트웨어 개발 절차나 기술을 위한 표준화된 도구와 기술을 이용하는 정도를 나타내는 통제 표준화에는 행위통제와 산출통제가 있다. Phan, Vogel and Nunamaker[1995]의 사례연구에 의하면, 소프트웨어 개발 자동화 도구인 IDSS(Integrated Development Support System)에 의해서 개발생명주기 단계를 통제하는 산출통제는 물론, 설계, 코딩, 품질보증, 등에 관한 행위통제를 함으로써, 개발 프로젝트의 생산성을 22% 이상 증가시킨다고 했다. Nidumolu[1996a, 1996b]는 통제 표준화가 높을수록 잔여성과위험이 낮아져서 프로젝트의 과정 및 제품 성과는 높아진다고 했다. 그러므로, 공식적인 통제인 행위통제와 산출통제에 대한 표준화가 높을수록 프로젝트 성과인 소프트웨어 품질은 높아질 것이다.

가설 4: 통제표준화가 높을수록 소프트웨어 품질이 높아진다.

2.2.3 상호작용이 소프트웨어 품질에 미치는 영향

사용자가 정보시스템 개발과정에 참여해서 개발 프로젝트 팀과 상호작용 함으로써, 개발되는 정보시스템에 대한 이해도가 향상되고 사용자 요구사항도 정확하게 반영되므로, 궁극적으로

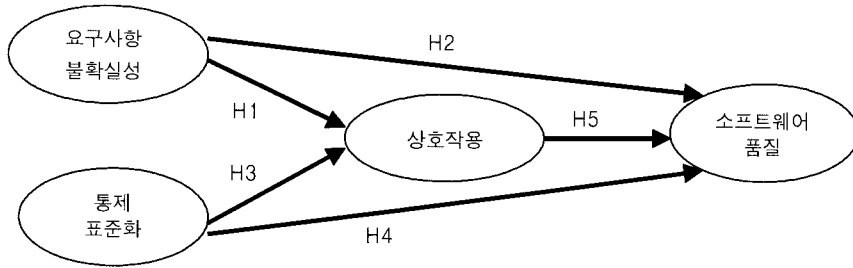
사용자 정보 만족도를 극대화한다고 주장했다 [McKeen, Guimaraes, 1994; Doll and Torkzadeh, 1989]. 또한, 사용자 참여로 인한 상호작용으로 정보 시스템의 기능성이 향상되고, 정보시스템 사용도 혹은 활용도가 높아진다고 주장했다 [Anderson, 1985; Baroudi, Olson, and Ives, 1983]. 이와 같이 사용자와 프로젝트 팀 간의 상호작용은 소프트웨어 개발 프로젝트의 성과인 사용자 만족도와 시스템 활용도는 물론 소프트웨어 품질에도 영향을 줄 것이라고 예상하여 다음과 같은 가설을 설정하였다.

가설 5: 사용자-프로젝트팀 간의 상호작용이 높을수록 소프트웨어 품질이 높아진다.

Ⅲ. 연구 설계

3.1 연구 모형

구조방정식모형(structural equation modeling)으로서 LISREL(LInear Structural RELations)은 잠재변인들의 복잡한 인과적 관계성을 검증하고, 각 잠재변인을 측정하는 관찰변인이 얼마나 잠재변인을 적절히 측정하는지를 파악할 수 있는 복합적인 통계기법이다. 본 연구에서는 이론적으로 안정된 개념적 틀 속에서 간명한 구조방정식 모형을 설정하고, 이를 타당하게 분석하기 위해서, 우선 소프트웨어 개발 프로젝트의 성과인 소프트웨어 품질에 영향을 주는 개발위험으로서, 요구사항 불확실성, 통제표준화, 그리고 사용자-프로젝트 팀 간의 상호작용과 같은 변수들을 도출하였다. 그리고, 기존 연구에서 입증되어진 요구사항 불확실성과 통제 표준화가 사용자-프로젝트 팀 간의 상호작용에는 직접적으로, 소프트웨어 품질에는 직접 및 간접적으로 영향을 미칠 것이라는 것을 제안하면서 다음과 같은 연구모형을 설정하였다.



<그림 1> 연구 모형

3.2 표본 설계

본 연구의 설문지 개발을 위하여 실무에 종사하고 있는 전문가들의 자문을 구하였으며, 또한 관련된 문헌조사를 실시하였다. 설문지 내용은 최근에 수행한 특정한 프로젝트를 대상으로 일관성 있게 응답해야 하는 항목들로 구성된 설문지는 1차 예비 검토를 걸쳐서, 설문 내용이 본 연구에 적절하도록 수정 보완하였다. 표본추출대상기업은 우리나라 소프트웨어 개발 업체를 대표하는 LG-CNS, 삼성 SDS, DACOM-ST 3개 회사이다. 소프트웨어 개발 프로젝트 및 응답자들에 관한 표본의 각 항목들이 비교적 고르게 분포되어 있어서, 본 연구에서 사용된 표본은 모집단을 비교적 잘 대표한다고 볼 수 있다. 연구자료 수집을 위한 프로젝트 관리자(PM)를 포함한 프로젝트 내의 세부분야별 리더를 대상으로 했다. 설문지 배포 수는 3개 회사별로 40부씩 120부이며, 소프트웨어 품질 담당부서장들로부터 회수된 설문지는 120부 모두 회수되었으나, 응답이 부실한 3개 설문지를 제외한 총 117개의 설문지를 분석대상으로 했다.

3.3 변수의 조작적 정의와 측정

본 연구의 주요변수는 SEI(Software Engineering Institute)의 위험관리(CRM; Continuous Risk Management) 이론을 근거로 선택되었고, 측정 항목은 소프트웨어 위험관리의 이론적 틀을 처음으로 제시한 Nidumolu[1996a, 1996b]모형을

근거로, 한국 SI업계의 현황을 고려해서 선택 측정 되어졌다.

요구사항 불확실성(requirement uncertainty)의 개념에는 불안전성, 다양성, 분석가능성 등이 있다. 여기서 불안전성이란 프로젝트의 초기 및 후기 진행과정 중에 사용자 요구사항이 변화되는 정도이고, 다양성이란 사용자들 간에 요구사항에 대해서 서로 다르게 인식하는 정도이다. 또한, 분석성이란 사용자의 요구사항을 명세서로 변환시키는 정도이다. 본 연구에서 요구사항 불확실성은 요구사항 불안전성으로 정의했다. 요구사항 불확실성은 3개 항(요구사항들의 후기단계에 변경, 초기단계 요구사항들이 후기단계와 동일, 요구사항들에 대한 미래 변경 가능성)으로 구성되었고, 5점 척도를 사용하여 측정하였으며, 점수가 높을수록 소프트웨어 개발 위험이 높은 것을 의미한다.

통제 표준화(control standardization)의 개념에는 소프트웨어 개발 절차나 기술을 위한 표준화된 도구와 기술을 이용하는 정도를 나타내는 행위통제 표준화와 산출통제 표준화가 있다. 여기서 행위통제 표준화란 소프트웨어 개발 프로젝트를 수행해야하는 직원들의 행위를 통제하기 위해서 표준화된 도구와 기술의 이용 정도이다. 또한, 산출통제 표준화란 관리자들이 설정한 산출 목표를 달성하기 위해서 개발 단계별로 이정표를 작성해서 계획대로 실행 및 통제하는 정도이다. 본 연구에서 통제 표준화는 산출통제 표준화로 정의했다. 통제 표준화는 4개 항목(프로젝트의 개발단계 구분, 프로젝트 개발 단계별 이정

<표 1> 응답자 및 소프트웨어 개발 프로젝트의 특성

항 목	세 부 항 목	빈도수(%)	항 목	세 부 항 목	빈도수(%)	
응답자 직 책	프로젝트관리자	21(17.9%)	프로젝트 대상 업종	건설/교통	12(10.4%)	
	프로젝트세부분야 리더	46(39.3%)		공공행정	28(24.3%)	
	팀 원	49(41.8%)		국방/치안	5(4.3%)	
	무응답	1(1%)		금융/보험	17(14.8%)	
	합 계	117(100%)		제 조	10(8.7%)	
응답자의 근무 경력	36개월 미만	19(16.2%)		서비스업	28(24.3%)	
	37개월~72개월	36(30.7%)		기 타	17(13%)	
	73개월~120개월	33(28.2%)		합 계	117(100%)	
	121개월 이상	27(23%)		프로젝트 규 모	1억 이하	10(8.5%)
	무응답	2(1.9%)			1억~10억	41(35%)
	합 계	117(100%)	10억~50억		22(18.8%)	
프로젝트의 성격	일괄통합시스템	84(71.7%)	50억 이상		14(11.9%)	
	S/W 수탁개발	13(11.1%)	무응답		30(25.8%)	
	DB제작서비스	1(0.8%)	합 계	117(100%)		
	컨설팅	1(0.8%)	프로젝트 수행 기간	6개월 이하	14(11.9%)	
	자료입력서비스	2(1.7%)		6개월~12개월	34(29%)	
	정보제공서비스	2(1.7%)		12개월~24개월	33(28.2%)	
	유지보수	1(0.8%)		24개월 이상	21(17.9%)	
	시스템관리 및 운영	6(5.1%)		무응답	15(13%)	
	네트워크 서비스	1(0.8%)	합 계	117(100%)		
	멀티미디어 콘텐츠 제작서비스	1(0.8%)	프로젝트 투입 인력	10명 이하	39(40.2%)	
	기 타	6(5.1%)		10명~20명	17(17.5%)	
	합 계	117(100%)		20명~40명	14(14.4%)	
				40명이상	27(27.8%)	
		합 계		117(100%)		

표 작성, 프로젝트 개발 단계별 완료 관련문서 작성)으로, 5점 척도를 사용하였으며, 점수가 높을수록 통제표준화가 구체적인 절차까지 준수된 것을 의미한다.

본 연구에서 사용자-프로젝트 팀 간의 상호작용이란 사용자들과 프로젝트 팀원들 간의 의사소통을 포함한 전반적인 상호작용의 정도를 의미한다. 사용자-프로젝트 팀 간의 상호작용은 4개 항목(사용자 훈련의 완성도, 자료 처리부서와 사용자사이의 의사소통, 프로젝트 참여에 대한 사용자들의 반응, 사용자들의 전반적인 상호작용)

으로 구성되었고, 5점 척도를 사용하였으며, 점수가 높을수록 상호작용의 정도가 높은 것을 의미한다.

우리나라의 공공프로젝트 경우에는 소프트웨어 개발 완료시점에서 최종감리를 받을 때, 제품성과 관점에서 업무실행에 대한 부분적인 실험에 의해서 감리보고서가 작성되므로, 개발자 입장에서도 개관적인 소프트웨어의 품질을 측정할 수 있다고 가정할 수 있다. 그러므로, 본 연구에서 소프트웨어 품질은 소프트웨어 개발 프로젝트의 제품성과로서 소프트웨어 유연성으로 정의

했다. 구체적으로 소프트웨어 개발 프로젝트의 성과는 업무변화를 수용하기 위한 수정비용 및 유지보수비용, 업무변화를 수용하는데 소요되는 시간, 그리고 소프트웨어의 전반적인 장기 유연성으로 정의하는 것이 합리적이다. 소프트웨어 품질은 4개 항목(업무변화를 수용하기 위한 소프트웨어 수정비용, 업무변화를 수용하는데 소요되는 시간, 소프트웨어 유지보수 비용, 소프트웨어의 전반적인 장기 유연성)으로 구성되었고, 5점 척도를 사용하였으며, 점수가 높을수록 제품성과가 높은 것을 의미한다.

IV. 실증 분석

4.1 단일차원성, 신뢰성, 집중타당성 및 상관관계 분석

4.1.1 단일차원성 및 신뢰성 분석

구조모형의 분석을 통해서 가설검증을 하기

<표 2> 탐색적 요인분석 및 신뢰성분석 결과

연구변수의 측정항목	요인적재값	아이겐값	분산비(%)	신뢰성 계수
소프트웨어 품질				
업무변화를 수용하기 위한 소프트웨어 수정비용	.724	4.143	27.619	.6374
업무변화를 수용하는데 소요되는 시간	.760			
소프트웨어 유지보수 비용	.779			
소프트웨어의 전반적인 장기 유연성	.734			
상호작용				
사용자 훈련의 완성도	.718	2.654	17.697	.8717
자료 처리부서와 사용자 사이의 의사소통	.622			
프로젝트 참여에 대한 사용자들의 반응	.845			
사용자들의 전반적인 상호작용	.818			
요구사항 불확실성				
요구사항들의 후기 단계에 변경	.683	1.699	11.326	.7929
초기단계 요구사항들이 후기단계와 동일	.797			
요구사항들에 대한 미래 변경 가능성	.760			
통제 표준화				
프로젝트의 개발 단계 구분	.873	1.344	8.962	.7809
프로젝트 개발 단계별 이정표 작성	.815			
프로젝트 개발 단계별 완료 관련문서 작성	.845			
계획에 따라 승인절차 실행	.824			

에 앞서 먼저 각 이론변수들의 단일차원성을 확보하기 위해 측정변수들에 대한 탐색적 요인분석을 실시하여, 부적합한 측정항목을 제거한 것이 <표 2>이다. 최종요인 분석결과 요구사항 불확실성, 통제 표준화, 상호작용, 소프트웨어 품질을 측정하는 항목들은 모두 단일차원성을 지니고 있는 것으로 나타나고 있으며, 동일한 요인에 0.6 이상의 높은 적재값을 보이고 있다. 신뢰성 분석 결과, 신뢰성 계수인 Cronbach alpha 값은 0.64~0.87 로서, 측정변수들은 전반적으로 높은 내적 일관성을 가지고 있는 것으로 평가될 수 있다.

4.1.2 집중타당성 및 상관관계 분석

탐색적 요인분석에 의해서 찾아낸 연구변수들에 대한 집중타당성을 검증하기 위해서, LISREL 측정모형을 이용해서 확인적 요인분석을 실시한 것이 <표 3>이다. 네 가지 연구변수들의 측정항목들에 대한 표준 요인적재 값에 대한 t 값은 모

<표 3> 확인적 요인분석 결과

연구변수와 측정항목	요인적재 값	표준오차	t 값	개념 신뢰성	분산추출 값
소프트웨어 품질					
업무변화를 수용하기 위한 소프트웨어 수정비용	.80	0.09	8.97	0.783	0.488
업무변화를 수용하는데 소요되는 시간	.87	0.09	9.64		
소프트웨어 유지보수 비용	.51	0.09	5.43		
소프트웨어의 전반적인 장기 유연성	.54	0.09	5.77		
상호작용					
사용자 훈련의 완성도	.79	0.09	8.69	0.797	0.502
자료 처리 부서와 사용자 사이의 의사소통	.84	0.09	9.19		
프로젝트 참여에 대한 사용자들의 반응	.53	0.10	5.54		
사용자들의 전반적인 상호작용	.64	0.09	6.90		
요구사항 불확실성					
요구사항들의 후기 단계에 변경	.71	0.11	6.36	0.657	0.462
초기단계 요구사항들이 후기단계와 동일	.74	0.11	6.60		
요구사항들에 대한 미래 변경 가능성	.40	0.11	3.79		
통제 표준화					
프로젝트의 개발 단계 구분	.82	0.08	10.31	0.874	0.635
프로젝트 개발 단계별 이정표 작성	.84	0.08	10.50		
프로젝트 개발 단계별 완료 관련문서 작성	.73	0.08	8.74		
계획에 따라 승인절차 실행	.79	0.08	9.63		

두 +3.29 이상으로 나타나고 있어($p < 0.001$), 각 개념에 대한 측정항목의 요인적재 값이 통계적으로 유의함을 나타내 주고 있다. 따라서 각 개념의 척도들은 집중타당성을 가지고 있다고 평가 할 수 있다. 또한, 구성개념에 대한 대표성을 갖는지를 평가하기 위하여 구성개념 신뢰성과 분산추출 값을 계산한 결과를 제시했다. 일반적인 추천기준값이 구성개념 신뢰성은 0.7이고, 분산추출 값은 0.5인데, 요인적재 값이 적은 측정항목이 포함된 연구변수인 요구사항불확실성 경우에는 이 기준에 다소 못 미친다.

확인적 요인분석을 실행한 전체 측정모형에 대한 적합도 지수들은 다음과 같이 적합도에 문제가 없는 것으로 나타났다. Chi-square(χ^2) = 138.17, 자유도(df) = 87, 확률 값(p-value) = 0.00040, 원소간 평균차이(RMR; Root Mean Square Residual) = 0.10, 적합도 지수(GFI; Goodness on Fit Index) = 0.87, 적합도 지수를 자유도(df)에 대해 변환시킨 수정 적합도 지수(AGFI; Adjusted

GFI) = 0.82, 표준적합지수(NFI; Normed Fit Index) = 0.81, 비표준적합지수(NNFI; Non-normed Fit Index) = 0.90, 비교적합지수(CFI; Comparative Fit Index) = 0.92 이다. 이 모형은 Null 모형에서 상관관계계수가 $p < 0.05$ 수준에서 유의하지 않은 요구사항불확실성과 통제 표준화, 통제 표준화와 소프트웨어 품질 간의 두 경로를 제외시킨 모형이다.

확인요인분석결과 단일차원성이 입증된 각 연구변수들에 대해서 서로 어떤 방향으로, 어느 정도의 관계를 갖는지를 알아보기 위해서 상관관계분석을 실시하였다. 연구변수들 간의 상관관계 분석 결과, 상호작용과 요구사항 불확실성, 통제 표준화, 소프트웨어 품질과 각각 $p < 0.01$ 수준에서 통계적으로 유의하고, 소프트웨어 품질과 요구사항 불확실성이 $p < 0.05$ 수준에서 통계적으로 유의한 것으로 나타났다(Pearson 상관계수의 절대 값이 0.183이상일 경우 $p < 0.05$ 수준에서 유의하고, 0.26이상일 경우 $p < 0.01$ 수준에서 유

<표 4> 상관계수 분석 결과

연구 변수	1	2	3	4
1. 요구사항 불확실성	1.000			
2. 통제 표준화	-0.036	1.000		
3. 상호작용	-0.318**	0.238**	1.000	
4. 소프트웨어 품질	-0.213*	0.151	0.359**	1.000

주) ** p < 0.01, * p < 0.05

의함). 요구사항 불확실성과 통제 표준화만이 약한 상관관계를 나타내고 있고, 특히 요구사항 불확실성과 나머지 변수들과는 부(-)의 상관관계를 나타내고 있다.

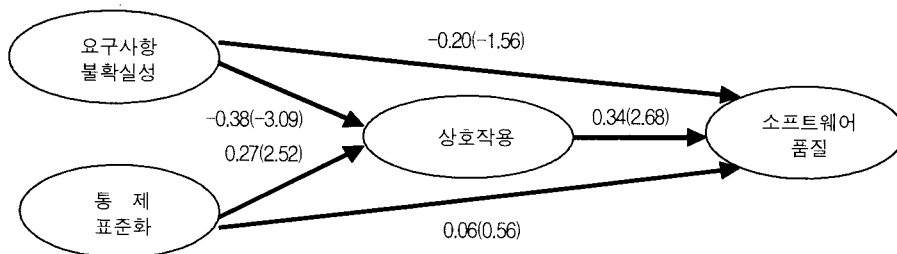
4.2 가설 검증

4.2.1 연구모형의 검증

본 연구에서는 연구모형에서 제시된 구성개념들 간의 관계를 검증하는데 중점을 두고 있으므로, 공변량 구조분석을 통해 구성개념들 간의 관계를 검증하고 평가하는데 유용한 LISREL을 이용하였다. Null 모형에 대한 적합도는 Chi-square (χ^2) = 128.37, df = 84, p-value = 0.0013, RMR = 0.076, GFI = 0.87, AGFI = 0.82, NFI = 0.82, NNFI = 0.91, CFI = 0.93이다. Null 모형과 매우 유사한 최종 연구모형은 상관관계가 약한 요구사항 불확실성과 통제 표준화 간의 관계를 제외하고는 네 가지 연구변수들 간에 경로가 모두 설정되어

있는 비교적 단순한 모형이다. 그러므로, 더 이상 명확하지 않은 경로를 제거해서 추정될 미지수를 줄여서 자유도를 높임으로써, 간명한 모형을 찾아가는 절차 혹은 도출된 여러 모형들 간의 비교분석을 더 이상 추가적으로 할 필요는 없다. 본 연구의 최종 연구모형에 대한 적합도 검증결과를 살펴보면, <그림 2>와 같은 구조모형이 도출되었다. Chi-square (χ^2) = 128.39, df = 85, p-value = 0.0017, RMR = 0.076, GFI = 0.87, AGFI = 0.82, NFI = 0.82, NNFI = 0.91, CFI = 0.93이다. Q-plot에 의하면, 잔차에 대한 분포가 좌표의 45도를 따라 분포하고 있으므로, 다변량 정상성을 가정하고 있다고 할 수 있다.

Chi-square 값에만 의존할 때, 표본크기가 커지면 실제모형이 적합하여도 적합하지 않은 것으로 잘못된 결정을 내리게 되므로, 일반적으로 Chi-square 값이 자유도의 두 배를 넘지 않으면, 확률 값(p)이 작아도 적합한 모형으로 평가한다. 그러므로, 본 연구모형은 공변량 구조분석의 일반적인 평가기준인 Chi-square 값에 대한 확률 값



주) 숫자는 표준경로계수, () 안의 값들은 t 값을 나타낸다.

<그림 2> 연구모형의 가설검증 결과

<표 6> LISREL 분석 결과

가설	경로	경로명칭	경로계수	표준오차	t값	검증결과
1	요구사항 불확실성(ξ_1) → 상호작용(η_2)	γ_{21}	-0.38	0.12	-3.09	채택
2	요구사항 불확실성(ξ_1) → 소프트웨어 품질(η_1)	γ_{11}	-0.20	0.13	-1.56	기각
3	통제표준화(ξ_2) → 상호작용(η_1)	γ_{22}	0.27	0.11	2.52	채택
4	통제표준화(ξ_2) → 소프트웨어 품질(η_1)	γ_{12}	0.06	0.11	0.57	기각
5	상호작용(η_2) → 소프트웨어 품질(η_1)	β_{12}	0.34	0.13	2.68	채택

주) 유의수준은 95% 수준에서 결정(t 값 > ±1.96)

(p)의 기준을 충족시키지는 않으나, Chi-square 값이 자유도의 두 배 보다 적으므로, 확률 값이 작아도 적합한 모형으로 평가할 수 있다. 일반적으로 RMR은 0.05 보다 작고, GFI, AGFI, NFI, NNFI, CFI는 0.9 보다 크면(최적모형에서는 1), 매우 양호한 모형으로 평가된다. 본 연구모형에서 RMR이 0.076 이고, GFI, AGFI, NFI, NNFI, CFI가 모두 0.8 이상으로 나타나고 있으므로, 전체모형의 적합도는 받아들여질 수 있는 것으로 판단된다.

4.2.2 가설에 대한 검증

변수들 간의 인과관계를 중심으로 설정된 가설을 검증한 결과, 95%의 신뢰수준에서 가설 2와 가설 4는 기각되었고, 가설 1, 가설 3, 그리고 가설 5는 채택되었다. 채택 가설 1에 의하면, 요구사항 불확실성은 사용자-프로젝트팀 간의 상호작용에 유의한 영향을 미치고 있다. 즉 요구사항 불확실성이 높을수록 사용자-프로젝트팀 간의 상호작용은 낮아진다. 채택 가설 1과 기각 가설 2의 검증결과, 요구사항 불확실성은 소프트웨어 품질에 직접적으로 영향을 미치지 않고, 반드시 사용자-프로젝트 팀 간의 상호작용에 영향을 미친 후에 소프트웨어 품질에 영향을 미치는 것으로 나타나고 있다. 또한, 가설 3에 의하면, 통제 표준화는 사용자-프로젝트팀 간의 상호작용에 유의한 영향을 미치고 있다. 즉 통제 표준화가 높을수록 사용자-프로젝트 팀 간의 상호

작용은 높아진다. 채택 가설 3과 기각 가설 4의 검증결과, 통제 표준화는 소프트웨어 품질에 직접적으로 영향을 미치지 않고, 반드시 사용자-프로젝트 팀 간의 상호작용에 영향을 미친 후에 소프트웨어 품질에 영향을 미치는 것으로 나타나고 있다. 마지막으로 가설 5에 의하면, 상호작용이 소프트웨어 품질에 유의한 영향을 미치고 있다. 즉 사용자-프로젝트팀 간의 상호작용이 높을수록 소프트웨어 품질이 높아진다.

본 연구결과에 따르면, 요구사항 불확실성과 통제 표준화는 소프트웨어 품질에 직접적인 영향을 미치지 않고, 사용자-프로젝트 팀 간의 상호작용에 직접적인 영향을 미치며, 이를 통해서 소프트웨어 품질에 간접적인 영향을 미친다. 그러므로, 요구사항 불확실성이 낮을수록, 그리고 통제 표준화가 높을수록, 사용자-프로젝트 팀 간의 상호작용은 높아져서 궁극적으로는 소프트웨어 품질이 높아진다고 해석할 수 있다.

V. 결 론

5.1 연구의 요약

본 연구에서는 요구사항 불확실성과 소프트웨어 품질, 그리고 통제 표준화와 소프트웨어 품질 간의 관계에 있어서 사용자-프로젝트 팀 간의 상호작용이 매개적 역할을 하는지에 대하여 살펴보고자 하였다. 이를 위해 관련된 기존연구의 검

토를 통해 이론적 배경을 탐색하고, 연구문제 해결을 위한 가설을 설정하였으며, 이를 실증 분석하였다. 본 연구에서 제시된 연구가설에 대한 실증분석 결과와 시사점들은 다음과 같다.

첫째, 요구사항 불확실성이 높을수록 사용자-프로젝트 팀 간의 상호작용은 낮아지는 것으로 나타났다. 그러나, 요구사항 불확실성이 높을수록 소프트웨어 품질은 낮아진다는 것은 유의적으로 나타나지 않았다. 대부분 사용자 요구사항이 애매하거나 변경되어서 요구사항 불확실성이 높아질수록, 사용자와 프로젝트 팀 간의 상호작용은 오히려 낮아져서 궁극적으로는 소프트웨어의 개발위험을 증가시키게 된다. 그러나, 이와 같은 소프트웨어의 개발위험이 증가되어도 프로젝트의 성과인 소프트웨어 품질을 직접적으로 감소시키지 않는다는 것이다.

둘째, 통제 표준화가 높을수록 사용자-프로젝트 팀 간의 상호작용은 높아지는 것으로 나타났다. 그러나, 통제표준화가 높을수록 소프트웨어 품질이 높아진다는 것은 유의적으로 나타나지 않았다. 소프트웨어에 대한 통제 표준화는 소프트웨어 개발자들에게 체계적이고 일관성 있게 소프트웨어를 개발할 수 있도록, 사용자와 프로젝트 팀 간의 상호작용을 증가시키게 된다. 그러나, 이와 같은 통제 표준화가 증가되어도 프로젝트의 성과인 소프트웨어 품질을 직접적으로 증가시키지 않는다는 것이다.

셋째, 사용자-프로젝트 팀 간의 상호작용이 높을수록 소프트웨어 품질이 높아지는 것으로 나타났다. 사용자 요구사항이 명확하고 일관성이 있을수록, 표준화된 도구와 기술이 이용될수록 사용자와 프로젝트 팀 간의 상호작용에 직접적으로 영향을 주고, 이러한 상호작용이 높아질수록 프로젝트의 성과인 소프트웨어 품질을 높인다는 것이다.

결론적으로 요구사항 불확실성이 낮을수록, 그리고 통제 표준화가 높을수록, 사용자-프로젝트 팀 간의 상호작용은 높아져서 궁극적으로는

소프트웨어 품질이 높아지는데, 반드시 사용자-프로젝트 팀 간의 상호작용이 매개적 역할을 한다는 것이다. 그러므로, 소프트웨어 품질향상을 위해서는 사용자-프로젝트 팀 간의 상호작용을 제고시키려는 노력이 필수적임을 시사하는 결과라고 할 수 있다.

5.2 연구의 한계 및 미래 연구방향

본 연구는 다음과 같은 한계점을 가지고 있다. 첫째, 본 연구에서는 소프트웨어 품질을 제품성과 중에서 운영유연성을 프로젝트 개발자가 사업완료 시점에서 최종감리를 받으면서 소프트웨어를 시험한 경험을 근거로 측정했기 때문에, 객관적 평가가 결여된 측면이 있다. 소프트웨어를 운영해 본 후에 발생하는 업무변화를 수용하는데 소요되는 비용과 시간을 사용자 입장에서 측정되는 것이 바람직할 것이다. 둘째, 설문지의 응답자들에게 프로젝트 완료 후 시점에서 과거 경험을 근거로 대답을 요구했기 때문에, 이로 인한 측정오차가 발생될 수도 있다. 셋째, 본 연구에서는 표본 확보의 어려움으로 인하여 3개 SI 업체에서 추출한 표본크기 117개는 층분치 못해서 전체모형에 대한 적합도지수가 아주 양호하게 나오지 못했다. LISREL 같은 구조방정식 모형에서 이용되고 있는 추정방법인 최우도 추정(maximum likelihood estimation: MLE)을 적절하게 이용하는 것이 보장되려면, 명확한 준거는 없지만, 최소한 100개~200개까지의 표본크기를 가져야 적합도지수를 해석하는데 오류를 범하지 않는다[Boomsma, 1987]. 또한, 표집오차를 줄이기 위해 표본크기는 추정해야 할 미지수 개수의 5배 혹은 10배 이상이 되어야 한다[Bentler and Chou, 1987]. 그러나, 표본 수는 최소한의 표본규모 100개를 초과하고 있고, 필요한 표본규모에 영향을 미치는 모형의 복잡성 측면에서도, 본 논문의 모형은 상대적으로 단순하기 때문에, 다소 표본규모를 증가시킨다고 해도 분석결과는 거의

동일하게 나오리라 유추된다.

끝으로 미래연구를 위해서 다음과 같은 추가적인 제언을 한다. 첫째, 소프트웨어 품질에 대한 측정을 일정기간 소프트웨어를 운영해 본 후에, 제품성과로서 소프트웨어 운영유연성 이외에 소프트웨어 효율성 및 소프트웨어 반응성, 혹은 과정성과로서 학습, 혹은 사용자 만족도 등의 개념으로도 측정해 볼 수도 있다. 둘째, 요구사항 불확실성은 요구사항 불안전성 이외에, 요구사

항 다양성, 요구사항 분석가능성 등의 개념으로 측정될 수도 있고, 통제 표준화는 산출통제 표준화 이외에, 행위통제 표준화의 개념으로 측정될 수도 있다. 셋째, 소프트웨어 개발 프로젝트의 성과에 영향을 주는 정보시스템 방법론으로서, 특히 공공부문의 정보시스템에서는 감리 시 기준으로 관리기법 1(Method 1)을 활용하므로, 이러한 관점에서 본 연구모형을 확장하여 적용해야 할 것이다.

〈참 고 문 헌〉

- [1] Anderson, E.E., "Managerial Considerations in Participative Design of MIS/DSS," *Information and Management*, Vol. 9, 1985, pp. 201-207.
- [2] Barki, H., Rivard, S. and Talbot, J., "Toward an Assessment of Software Development Risk," *J. of Management Information System*, Vol. 10, No. 2, Fall 1993, pp. 203-225.
- [3] Baroudi, J.J., Olson, M.H., and Ives, B., "An Empirical Study of the Impact of User Involvement on System Usage and User Satisfaction," *Communications of the ACM*, Vol. 29, No. 3, 1983, pp. 232-238.
- [4] Bentler, P.M. and Chou, C.P., "Practical Issues in Structural Modeling," *Sociological Methods and Research*, Vol. 16, 1987, pp. 78-117.
- [5] Boehm, B.W., *Software Risk Management*, Washington, DC: IEEE Company Society Press, 1989.
- [6] Boehm, B.W., "Software Risk Management: Principles and Practices," *IEEE Software*, Jan. 1992, pp. 32-41.
- [7] Boehm, B.W. and Ross, R., "Theory-W Software Project Management: Principles and Examples," *IEEE Trans. on Software Engineering*, Vol. 15, No. 7, 1989, pp. 902-916.
- [8] Boomsma, A., The Robustness of Maximum Likelihood Estimation in Structural Equation Models, in P. Cuttance and R. Ecob (eds.), *Structural Modeling by Example: Applications in Educational, Sociological, and Behavioral Research*, New York: Cambridge University, 1989.
- [9] Carr, Marvin J., Konda, Suresh L., Monarch, Ira, Ulrich, F. Carol. and Walker, Clay F., *Taxonomy-Based Risk Identification (CMU/SEI-93-TR-6, ESC-TR-93-183)*, Pittsburg, Pa: Software Engineering Institute, Carnegie Mellon University, 1993.
- [10] Cotterman, W.W. and Kumar, K., "User Cube: A Taxonomy of End Users," *Communication of the ACM*, Vol. 32, No. 11, 1989, pp. 1313-1320.
- [11] Charette, R.N., *Software Engineering Risk Analysis and Management*, McGraw-Hill New York, 1989.
- [12] Chittister, Clyde G. and Haimes, Yacov Y., "System Integration via Software Risk Management," *IEEE Transactions on System, Man,*

- and *Cybernetics - Part A: Systems and Humans*, Vol. 26, No. 5, Sep. 1996, pp. 521-532.
- [13] Davis, G.B., "Strategies for Information Requirements Determination," *IBM Systems Journal*, Vol. 21, No. 1982, pp. 4-30.
- [14] Doll, W., and Torkzadeh, J.G., "A Discrepancy Model of End-User Involvement," *Management Science*, Vol. 35, No. 10, 1989, pp. 1151-1171.
- [15] Dorofee, Audrey J., Walker, Julie A., Alberts, Christopher J., Higuera, Ronald P., Murphy, Richard L., and Williams, Ray C., *Continuous Risk Management*, Pittsburg, Pa: Software Engineering Institute, Carnegie Mellon University, 1996.
- [16] Eisenhardt, K.M., "Control: Organizational and Economic Approaches," *Management Science*, Vol. 31, No. 2, 1985, pp. 134-149.
- [17] _____, "Agency Theory: An Assessment and Review," *Academy Management Review*, Vol. 14, No. 1, 1989, pp. 54-74.
- [18] Gallaher, Brian P., Alberts, Christopher J., and Barbour, Richard E., *Software Acquisition Risk Management Key Process Area(KPA - A Guidebook Version 1.0)(CMU/SEI-97-HB-002)*, Pittsburg, Pa: Software Engineering Institute, Carnegie Mellon University, 1997.
- [19] Ginzberg, M., "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions," *Management Science*, Vol. 27, No. 4, 1981, pp. 459-478.
- [20] Haimes, Yakov Y., *Risk Modeling, Assessment, And Management*, John Wiley and Son, Inc., 1998.
- [21] Higuera, Ronald P. and Haimes, Yakov Y., *Software Risk Management (CMU/SEI-96-TR-012, ESC-TR-96-012)*, Pittsburg, Pa: Software Engineering Institute, Carnegie Mellon University, 1996.
- [22] Jones, David, "Project Zeus Risk Management Plan at NASA," <http://satc.gsfc.nasa.gov/crm/publications/rmpzeus.html>.
- [23] Jones, Mary C. and Harrison, Allison W., "IS Project Team Performance: An Empirical Assessment," *Information & Management*, Vol. 31, 1996, pp. 57-65.
- [24] Kangari, R. and Boyer, L.T., "Risk Management by Expert System," *Project Management Journal*, Vol. 20, No. 1, 1989, pp. 40-48.
- [25] Kirsch, Laurie J., "Portfolios of Control Modes and IS Project Management," *Information Systems Research*, Vol. 8, No. 3, Sep. 1997, pp. 215-238.
- [26] Klepper, R., "An Agency Theory Perspective on Information Centers," in *Proc. Twenty-Third Annual Hawaii International Conf. on System Sciences*, Ralph H. Sprague, Jr., (Ed.) Kailua-Kona, HI, Jan. 2-5, 1990, pp. 251-259.
- [27] Lawrence, M. and Low, G., "Exploring Individual User Satisfaction within User-Led Development," *MIS Quarterly*, Vol. 17, No. 2, 1993, pp. 195-208.
- [28] Lyytinen, Kalle, Mathiassen, Lars, and Ropponen, Janne, "Attention Shaping and Software Risk - A Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research*, Vol. 9, No. 3, Sep. 1998, pp. 233-255.
- [29] McFarlan, F.W., "Portfolio Approach to Information Systems," *Harvard Business Review*, Vol. 59, No. 4, Jan. ~ Aug. 1981, pp. 142-150.
- [30] McFarlan, F.W., "Portfolio Approach to Information Systems," *J. Systems Management*, Jan. 1982, pp. 12-19.
- [31] McKeen, J., Guimaraes, D.T. and Wetherbe, J.C., "The Relationship between User Par-

- icipation and User Satisfaction: An Investigation of Four Contingency Factors," *MIS Quarterly*, Vol. 31, No. 4, 1988, pp. 897-923.
- [32] Nidumolu, Sarma, "The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable," *Information Systems Research*, 1995, Vol. 6, No. 3. pp. 191-219.
- [33] _____, "A Comparison of the Structural Contingency and Risk-Based Perspectives on Coordination in Software-Development Projects," *Journal of Management Information Systems*, Vol. 13, No. 2, Fall 1996a, pp. 77-113.
- [34] _____, "Standardization, Requirements uncertainty and software project performance," *Information & Management*, 31, 1996b, pp. 135-150.
- [35] Phan, Dien D., Vogel, Douglas R. and Nunamaker, Jr. Jay F., "Empirical Studies in Software Development Projects: Field Survey and OS/400 Study," *Information & Management*, Vol. 28, 1995, pp. 271-280.
- [36] Pinkerton, Andrew, "Software Risk Management," <http://www.eas.asu.edu/~riskmgmt/intro.html>.
- [37] Rosenberg, Linda H. and Hyatt, Lawrence E., "Software Metric Program for Risk Assessment," http://satc.gsfc.gov/support/IAC_OCT96/iaf.html.
- [38] Walsh, Ken and Schneider, Helmut, "Software Development Risk and Agency Theory," <http://isid.bus.lsu.edu/cvoc/projects/software/risk/Index4.html>.

〈부록: 설문지〉

본 연구는 국내 소프트웨어 산업에서, 요구불확실성과 소프트웨어 개발표준이 프로젝트 성과에 미치는 영향을 실증적으로 설명하고자 하는데 그 목적이 있습니다. 연구자료 수집을 위한 설문지는 과제관리자(PM)와 프로젝트 내의 세부 분야별 리더를 대상으로 하고 있으며, 경우에 따라서는 경험이 풍부한 사원이 응답할 수도 있습니다. 설문 내용은 최근에 수행한 특정한 프로젝트를 대상으로 일관성 있게 응답하여야 하는 항목으로 구성되어 있습니다.

I. 먼저 응답자와 소속회사, 프로젝트 특성에 관한 질문입니다.

1. 응답자의 직책 ()
 (1) 프로젝트 관리자 (2) 프로젝트 세부분야 리더 (3) 팀원
2. 응답자의 정보시스템 개발분야 총 근무경력 (년 개월)
3. 응답자의 학력 ()
 (1) 고졸 (2) 초대졸 (3) 대졸 (4) 대학원졸 이상
4. 응답자의 직급 ()
 (1) 평사원 (2) 대리급 (3) 과장급 (4) 차장급
 (5) 부장급 (6) 이사급 (7) 최고경영자급
5. 응답 회사의 구분 ()
 (1) SI 업체 (2) 소프트웨어 개발업체
6. 귀사는 어떤 품질시스템 인증을 취득하였습니까? ()
 (1) ISO (2) TickIT (3) 기타() (4) 미 취득
7. 응답대상 프로젝트의 규모는? (천원)
8. 응답대상 프로젝트의 수행기간은? (년 개월)
9. 응답대상 프로젝트의 투입인력은? (명, MM)
10. 응답대상 프로젝트의 발주기관은? ()
 (1) 공공 (2) 민간

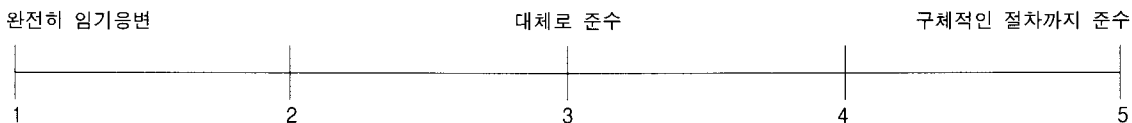
11. 프로젝트 성격 ()

- (1) 일괄시스템통합서비스(SI)
- (2) S/W 수탁개발
- (3) DB제작서비스
- (4) 컨설팅
- (5) 자료입력서비스
- (6) 정보제공서비스
- (7) 유지보수
- (8) 시스템관리 및 운영(SM)
- (9) 네트워크서비스
- (10) 멀티미디어 콘텐츠 제작서비스
- (11) 기타()

12. 프로젝트 대상업종 ()

- (1) 건설/교통
- (2) 공공행정
- (3) 국방/치안
- (4) 교육
- (5) 금융/보험
- (6) 레저
- (7) 무역/유통
- (8) 관광
- (9) 물류
- (10) 제조
- (11) 서비스업
- (12) 기타()

II. 소프트웨어 개발 프로젝트에서 정보처리 부서에서 정한 절차나 기술을 계획대로 단계 별로 얼마나 잘 준수되었는지에 관한 질문입니다. 최근에 수행했던 프로젝트의 개발시점에서, 아래 항목을 참조하여 동의 정도를 답해 주시기 바랍니다.



- 1. 프로젝트 수행 중 각 개발단계의 구분 ()
- 2. 프로젝트 수행 기간에 대해, 개발 단계별로 이정표(milestone) 작성 ()
- 3. 프로젝트 개발 각 단계가 완료될 때마다 관련문서들 작성 ()
- 4. 계획에 따라서 승인 절차를 실행 ()

III. 소프트웨어 개발 프로젝트에서 요구사항의 변경이 많을수록 사용자 요구사항을 충족시켜야 하는 과업의 불확실성 정도가 많아집니다. 최근에 수행했던 프로젝트의 개발시점에서, 아래 항목을 참조하여 동의 정도를 답해 주시기 바랍니다.



- 1. 요구사항들이 후기단계에서 부분적으로 조금만 변경되었다. ()
- 2. 프로젝트 초기에 도출된 요구사항들이 후기단계와 완전히 동일했다. ()
- 3. 요구사항들이 초기단계에서는 앞으로 거의 변경되지 않을 것이라고 예상되었다. ()

IV. 소프트웨어 개발 프로젝트에서 사용자들과 프로젝트 팀 간의 의사소통을 포함한 전반적인 상호작용에 대한 질문입니다. 최근에 수행했던 프로젝트의 개발시점에서, 아래 항목을 참조하여 동의 정도를 답해 주시기 바랍니다.



1. 사용자에게 제공된 **훈련의 완성도** ()
2. 자료처리 부서와 사용자 사이의 **의사소통** ()
3. 프로젝트 참여에 대한 **사용자들의 반응** ()
4. 사용자들의 전반적인 **상호작용** ()

V. 소프트웨어 개발 프로젝트의 제품성과로서 유연성 관점에서의 소프트웨어 품질에 대한 질문입니다. 최근에 수행했던 프로젝트의 개발시점에서, 아래 항목을 참조하여 동의 정도를 답해 주시기 바랍니다.



1. 업무변화를 수용하기 위한 **소프트웨어 수정 비용** ()
2. 업무변화를 수용하는데 **소요되는 시간** ()
3. 소프트웨어 **유지보수 비용** ()
4. 소프트웨어의 전반적인 **장기 유연성** ()

◆ 저자소개 ◆



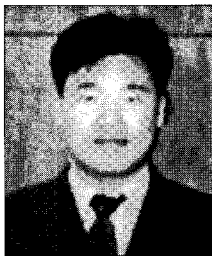
김기윤 (Kim, KI-Yoon)

현재 광운대학교 경영학과 재직 중이며, 고려대학교 공과대학 졸업 후, 동 대학원에서 경영학 석박사 학위를 취득하였고, 주요 관심분야는 보안관리, 위험 분석, IT 프로젝트 및 소프트웨어 위험 등이다.



나관식 (Na, Kwan-Sik)

현재 서원대학교 경영정보학과 재직 중이며, 광운대학교 경영학과 졸업 후, 동 대학원에서 경영학석박사 학위를 취득하였고, 주요 관심분야는 보안관리, 위험분석, IT 프로젝트 및 소프트웨어 위험 등이다.



양동구 (Yang, Dong-Gu)

현재 (주)우린정보에 팀장으로 재직 중이며, 광운대학교 경영학과 졸업 후, 동 대학원에서 경영학석사 학위를 취득하였고, 현재 박사과정 중이며, 주요 관심 분야는 보안관리, 위험분석, IT 프로젝트 및 소프트웨어 위험 등이다.

◆ 이 논문은 2002년 4월 3일 접수하여 1차 수정을 거쳐 2002년 10월 7일 게재 확정되었습니다.