

한국의 소프트웨어 개발 프로젝트 위험 관리 현황

The Present Condition of Risk Management for Software Development Project in Korea.

류나정¹⁾, 고석하²⁾

요약

잘못되거나 불확실한 결과가 예상되는 프로젝트를 계속 지속하는 행위는 어떠한 분야의 프로젝트에서도 발견 할 수 있는 보편적인 현상이다. 그러나 소프트웨어 개발 분야는 그러한 어떠한 분야보다도 이 문제에 더 민감하며 그 파급효과 또한 크다. 그 이유는 소프트웨어 그 자체가 형태를 가지고 있지 않아 프로젝트 진행 도중에는 그 완성도를 예측하기가 무척 어렵기 때문이다. 또한 소프트웨어는 프로젝트 범위가 자주 변동되기 때문에 통제와 관리에 어려움이 많다.

이런 이유로 소프트웨어 개발 생명주기의 각 단계에서 발생하는 위험 요소들을 찾아 그 위험 요소들이 발생했을 때 프로젝트에 미치는 영향정도 파악하여 그 해당 위험 요소에 대응하는 방안들을 모색하는 것이 피해를 줄이기 위해 가장 최선의 방법이다.

본 논문에서는 위에서 거론된 소프트웨어 위험 관리에 대한 관련 문헌을 조사, 검토하고 설문을 통해 조사한 실무자들의 경험을 바탕으로 위험에 대한 인식과 대응 반응을 파악하고 연구하고자 한다.

소프트웨어 프로젝트 계획 단계에서 위험 요소가 추후 발생할 것을 예측하여 실제로 위험 요소가 발현했을 때, 적극적으로 대처 할 수 있는 방법을 찾아 해당 위험이 주는 영향을 최소화 할 수 있는 방법을 찾고자 한다.

1) 충북대학교 경영정보 석사과정

2) 충북대학교 경영정보학과 교수 shkoh@chungbuk.ac.kr

I. 서론

소프트웨어 개발 프로젝트가 대형화되면 필수로 치밀한 계획과 통제의 필요성이 증가한다는 점에서 소프트웨어 프로젝트 관리 시스템의 중요성이 있다. 소프트웨어 개발 관리 프로젝트의 특성이 다양하고 눈에 보이지 않는 소프트웨어를 대상으로 하기 때문에 관리에 많은 어려움이 있다¹⁾.

정보 시스템 구축 프로젝트의 수행에 있어서 위험 요소는 지금까지 수행된 모든 프로젝트에 내재되어 있었다. 그럼에도 불구하고, 위험 요소가 지닌 불확실성(발생 가능성, 영향도, 대안의 적합성 등)으로 인하여, 아직까지도 구체적인 틀로써 위험 요소들을 추적하고 관리하는 것은 매우 어려운 과제로 남아 있다. 더욱이 정보시스템의 개발이 단순한 소프트웨어 개발에서, 이른바 시스템 통합이라 일컬어지는 사용자 중심의 통합 솔루션 제공을 위한 개념으로 확대 발전함으로써, 프로젝트 수행의 규모와 복잡성이 매우 커졌다. 다시 말해 대규모 개발 조직이 장기간에 걸쳐 다양한 구성 요소들을 개발, 통합하게 되면서 각각의 위험 요소들이 기하급수적으로 증가하게 되었고, 이에 따라 프로젝트 수행 시 효과적인 위험관리 프로젝트 성공의 주요 핵심 요소로 자리하게 되었다.

오늘날 프로젝트 관리자들은 과거보다 프로젝트의 많은 위험 요소를 확인해야 하고, 통제해야 하며 프로젝트의 성공보다는 완료를 목표로 힘겨운 싸움을 하고 있다. 특히 SI(System Integration) 업체들이 그 동안 일반기업(그룹의 System Management 부문)의 프로젝트만을 수행해 오다가, 대규모 국가 기간 망 사업을 수행하면서 겪어야 하는 시행착오는 프로젝트의 완료 여부를 떠나서, 국가 경쟁력의 저하 및 막대한 국가 예산을 낭비하고 있다는 비난과 함께, 해당 업체의 프로젝트 수행 능력을 의심 받아야 하고 관련자들 또한, 심한 좌절감을 겪고 있는 것이 현실이다²⁾.

오늘날 조직 내에서 프로젝트를 관리하는 관리자들은 프로젝트에 대한 부정적인 정보가 입수되었을 때 이를 계속 수행할 것인가 아니면 중단하거나 변경할 것인가 하는 의사결정을 내려야 하는 상황에 종종 직면하고 있다. 이러한 상황에서 프로젝트 관리자의 잘못된 결정은 앞에서 언급한 바와 같이 돌이킬 수 없는 손실을 기업에게 가져다 줄 수 있다.

잘못되거나 불확실한 결과가 예상되는 프로젝트를 계속 지속하는 행위는 어떠한 분야의 프로젝트에서도 발견 할 수 있는 보편적인 현상이다. 그러나 소프트웨어 개발 분야는 그러한 어떠한 분야보다도 이 문제에 더 민감하며 그 파급효과 또한 크다. 그 이유는 소프트웨어 그 자체가 형태를 가지고 있지 않아 프로젝트 진행 도중에는 그 완성도를 예측하기가 무척 어렵기 때문이다³⁾. 또한 소프트웨어는 프로젝트

1) 문용은, 1996

2) 김경내, 2000

범위가 자주 변동되기 때문에 통제와 관리에 어려움이 많다¹⁾.

일반적으로 소프트웨어 개발 생명 주기(Software Development Life Cycle)란 소프트웨어를 어떻게 개발할 것인가에 대한 추상적 표현으로 순차적 또는 병렬적인 단계의 연속으로 구성된다. 생명주기는 프로세스 모델(Process Model)이라고도 하며, 소프트웨어 공학(Software Engineering Paradigm)이라고도 한다. 각 단계는 제품이 완벽하게 작동될 수 있도록 관련된 여러 활동이 있으며, 이러한 활동으로 다음 단계에 활용될 수 있는 산출물을 만들어 낸다. 생명주기는 주어진 예산과 자원으로 개발 방법 및 개발 환경, 개발 관리에 대한 포괄적인 접근 방법을 설정하여 높은 품질의 소프트웨어를 만들어 내기 위한 것이다.

소프트웨어의 생명 주기를 구성하는 단계는 모델에 따라 약간씩 다르겠지만, 일반적으로 정의 단계, 개발 단계, 유지 보수 단계로 구성되며 각 단계마다 보호 활동들이 수행된다²⁾.

따라서 소프트웨어 개발 프로젝트에 있어 수행의 지속 행위를 설명할 수 있는 체계적이고 설득력 있는 이론을 제시할 필요가 있으며 이러한 연구의 결과는 프로젝트 수행의 지속의지에 영향을 미치는 요소들을 제시함으로써 소프트웨어 프로젝트를 수행하는 관리자에게 도움을 줄 수 있을 것이다³⁾.

본 연구의 목적은 소프트웨어 프로젝트의 위험 요소들이 생명주기 각 단계에서 발생한 정도와 그에 따른 대응은 어떠한지를 분석하고자 한다. 더 나아가 각 위험 요소들의 발생했을 때 가장 적절한 대응책이 무엇인지를 찾아 더 나은 프로젝트 관리를 할 수 있는 방법을 찾는 데 그 목적이 있다.

1. 연구의 배경 및 목적

본 논문의 연구 목적은 서론에서 밝힌 바와 같이 소프트웨어 개발 생명주기의 각 단계에서 발생하는 위험 요소들을 찾아 그 위험 요소들이 발생했을 때 프로젝트에 미치는 영향정도 파악하고, 대응하는 방안들을 모색하고자 한다.

또한 이러한 위험 요소를 사전에 예측하였을 경우와 예측하였지만 대응하지 못했을 경우 그리고, 예측하여 대책을 마련했을 경우에 대해서도 조사를 한다.

본 연구에서는 다음 2가지 단계를 통해 분석하고자 한다.

첫째, 실제로 해당 요소가 발현하였는지 여부 : 만약 여러 단계에서 발생하였다면 가장

3) Abdel-Hamid and Mandnick, 1991

1) Abdel-Hamid and Mandnick, 1991; Zmud, 1980

2) 김수동, 1999

3) 김종은, 2000

심각하게 발현한 단계를 조사하고, 둘째, 해당 위험 요소가 최종적으로 프로젝트에 얼마나 큰 영향을 미쳤는지를 조사한다.

II. 소프트웨어 위험 관리

CMU/SEI의 A Software Engineering Body of Knowledge Version 1.0에서는 소프트웨어 관리의 정의를 다음과 같이 내리고 있다.

소프트웨어 관리란 프로젝트 관리, 위험 관리, 소프트웨어 품질 그리고 형상관리에 관련된 활동들을 포함한 소프트웨어 제품과 프로젝트 관리 기술, 방법, 개념 등을 관리하는 일련의 행위라고 한다.

소프트웨어 관리와 관련된 지식 영역은 소프트웨어 프로젝트 관리, 소프트웨어 위험 관리, 소프트웨어 품질 관리, 소프트웨어 형상 관리, 소프트웨어 프로세스 관리 그리고 마지막으로 소프트웨어 획득으로 나누고 있다.

본 논문에서는 소프트웨어 프로젝트 지식 영역 6개 영역 중 첫번째 위험 관리 영역을 다룬다.

소프트웨어 프로젝트 관리의 일반적인 개념과 프로젝트 수행에 따라 각 생명주기 단계별로 존재하는 위험 관리에 대한 개념을 소개한 후, 프로젝트의 위험 관리와 위험, 프로젝트의 지속 의지에 미치는 영향에 대해 관련된 선행 연구들을 살펴보자.

1. 프로젝트의 위험 관리

(1) 프로젝트 위험관리의 개념

1) 위험 관리의 개념

위험관리란 정보시스템 개발 프로젝트 수행 시 발생할 수 있는 문제점의 발생 확률과 영향 정도를 예측하여 보다 나은 해결책을 제시하기 위해 수행되는 활동을 말한다¹⁾.

2) 프로젝트 위험관리 개념

1) Anton D. Buttingeg B.Sc., Risk Management in a Software Development Life Cycle, 참조일 : 2003.07.08

프로젝트 위험관리는 어떤 위험 사건들이 프로젝트에 영향을 미칠 가능성이 있겠는가를 결정하는 위험인식(Risk Identification) 발생 가능한 결과들의 범위 평가와 그 결과들이 실현될 가능성을 평가하는 위험의 계량화(Risk Qualification), 긍정적인 결과들을 가져오게 할 기회들을 부각하고 부정적인 결과들을 가져오게 할 위험 요소들을 작게 완화하는 조치들을 마련하는 대책개발(Risk Response Development), 프로젝트가 수행되는 기간 동안 변화되는 위험수준에 대해 대응하는 위험 통제(Risk Response Control)로 구분하여 관리한다.

비록 각 과정들은 개별적 요소로 분리되어 표현되어 있지만, 실제 수행 과정에서는 서로 중첩되어 상호작용을 한다.

각 과정들은 각 프로젝트 단계에서 최소한 한번은 발생하게 되며 각 과정에는 프로젝트의 성격에 따라서 한 명 또는 그 이상의 인력 투입이 필요하거나 또는 여러 그룹의 참여가 요구되기도 한다.

이들 과정은 잘 정의된 상호작용과 함께 명확히 분리된 요소들로 표현되었지만, 실제로는 여기에 세부적으로 나타나지 않은 방법으로 서로 중복되거나 상호작용을 한다.

위험 관리의 목표는 이런 위험 요소를 식별하고, 위험 요소가 프로젝트에 미치는 영향을 최소화하기 위한 행동을 취하는 것이다. 소프트웨어에서 위험 관리(상대적으로 새로운 분야이다¹⁾).

3) 프로젝트 위험 요소

프로젝트 위험 요소는 발생 여부가 불명확한 조건이나 사건을 말하는데, 발생할 경우에는 시스템의 출력에 좋지 않은 결과를 초래할 수 있는 발생 가능한 조건이나 사건을 말한다. 위험 요소의 두 가지 핵심 요소는 위험 조건의 발생 가능성과 발생할 경우 입을 수 있는 프로젝트의 피해이다

효과적인 위험 관리활동을 수행하기 위해서는 관리의 대상이 되는 요소들을 체계적으로 정의하고, 해당 프로젝트에서 영향을 미칠 것으로 예상되는 요소들을 추출하여야 한다. 소프트웨어 공학 분야의 연구는 이질적인 프로젝트 그룹에 대한 설문 조사를 통해 다양한 위험 요소들을 제시하고 있다²⁾.

4) 대책 개발

일단 잠재적인 기회 또는 위협에 대하여 대응하기로 결정하고 나면, 대책 개발(수립)과정을 통하여 구체적으로 무엇을 할 것인가를 결정하게 된다.

1) B. Boehm, 1989

2) 정은희, 2000

- 가능성 감소

예방 될 수 없는 위험들은 우선순위 계획에 의해 발생 가능성을 감소시킴.

- 위험 회피

프로젝트는 평가 기간의 지연과 복잡한 기능의 증가에 의한 스케줄 지연에 의해 발생할 위험을 예방함.

- 위험 이전

다른 시스템들로 위험을 전가하거나 보험 등에 가입함.

- 상황적 계획

위험이 실제 발생했을 때의 영향을 감소시키기 위한 계획을 세움

2. 위험관리 절차

위험관리를 위한 기본전략 및 성공요소를 요약하면 다음과 같다.

1) 기본 전략

- 위험은 완전히 감소되지 않더라도, 추적 및 통제 대상이 되어야 함
- 효과적인 위험관리는 위험상황의 분석·감소 및 제거활동이 포함됨
- 위험사항을 예견할 수 있는 풍부한 프로젝트 경험을 활용함
- 위험관리는 프로젝트 관리 계획 수립에서 종료단계까지 지속적으로 실시함

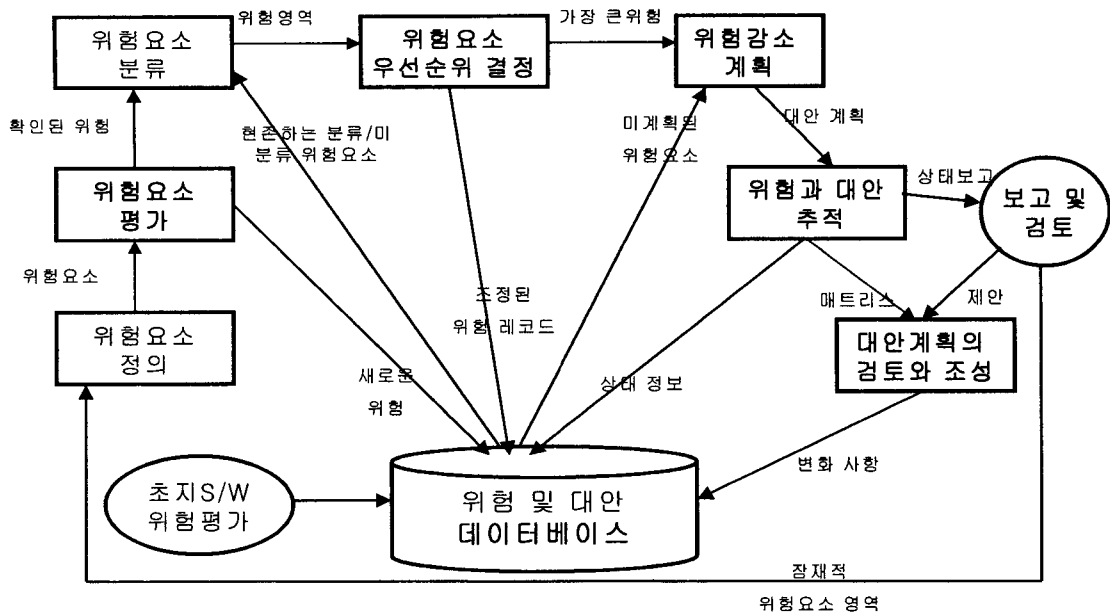
2) 성공요소

- 위험발생확률과 영향도를 미리 예견하고 사전에 방지함
- 전체 프로젝트 상황에 대해 주기적인 위험감시 실시
- 위험의 식별, 분석, 평가, 관리의 지속적 수행

위험관리를 위한 기본전략 및 성공요소를 기반으로 프로젝트 수행에 있어 좀 더 체계적인 위험관리를 절차적으로 정의하면 다음과 같은 모형이 가장 전형적인 접근 방법이다.

<그림-1> 위험관리 실행 체계¹⁾

1) WIL, 1997 & 김경내, 2000



III. 소프트웨어 위험 요소와 대응

1. 위험 요소 분류

(1) 위험 요소 분류

위험 요소는 프로젝트에 부정적이거나 해로운 영향을 끼치는 발생 가능한 사건과 조건이다. 위험 요소는 관리 측면의 조정이나 조치가 필요한 사건이나 조건과는 혼동하지 말아야 한다. 프로젝트 관리는 언제나 발생 할 수 있는 상황을 다루어야 하고, 이것을 위한 계획을 작성해야 한다. 그러나 그러한 발생 가능한 사건의 정확한 특성은 발생하기 전까지는 정확히 알 수 없다는 것이다. 그렇다고 해서 이런 상황들이 항상 위험한 것은 아니다.

위험관리를 실시하기 위해서는 관리의 대상이 되는 요소들을 체계적으로 정의하고, 해당 프로젝트에서 영향을 미칠 것으로 예상되는 요소들을 추출해 내야 한다.

다음 <표-1> 는 1989년 Boehm이 정리한 상위 10개 항목의 소프트웨어 위험 요소와 그에 따른 위험 관리 반응을 정리하였다.

<표-1> 소프트웨어 위험 목록1)

1) Boehm, 1998

위험 요소	위험 관리 반응
직원 부족	최고의 재능을 갖는 직원 양성 작업 분담 팀 구축 의욕 구축
비현실적인 스케줄과 예산	다각적인 비용과 스케줄에 대한 상세한 추정 비용 설계 소프트웨어 재사용 충분적 개발 요구사항 수정하기
잘못된 소프트웨어 기능	조직적인 분석 임무 분석 운영상의 개념 정형화 사용자 지원 초기 사용자 매뉴얼 프로토타이핑
잘못된 사용자 인터페이스	프로토타이핑 시나리오 작업 분석
두드러진 특징 (gold plating)	요구사항 수정 비용-이익 분석 프로토타이핑 비용 설계
요구사항 변경의 지속적인 흐름	높은 변경 임계치 정보 은닉 충분적 개발 (변화가 지연됨)
외부에서 공급된 요소들의 부족	벤치마킹 참조 검사 정밀 검사 호환성 분석
외부에서 수행된 작업의 부족	참조 검사 시상금 계약 경쟁적인 설계 또는 프로토타이핑 선시상 회계 감사 팀구축
실시간 성능 문제	시뮬레이션 모델링 도구(Instruction) 벤치마킹

	프로토타이핑
	세부 조정
컴퓨터 과학	기술적 분석
능력의 극복	프로토타이핑
	비용/이익 분석
	참조 문헌

IV. 연구 설계

1. 연구 모형 및 설계

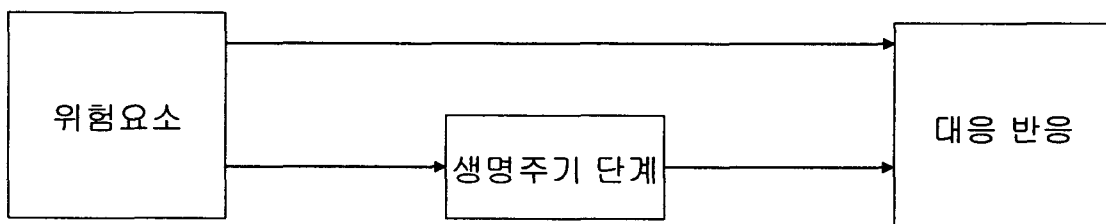
(1) 연구 모형의 설계

본 연구에서는 선행 연구들의 위험관리 모형과 최근 소프트웨어 프로젝트 개발에 관련된 위험에서 나타난 위험 요소들을 반영하여 소프트웨어 프로젝트 개발 생명주기 단계에서 발생하는 위험을 고려하였다.

우선 위험 요소들은 본 연구에서 분류한 주요 위험 요소들 중에 PMBOK(Project Mnagement Body Of Knowledge)의 프로젝트 관리상의 위험 요소들과 “현대 SYSTEM 개발 방법론” 상의 위험 관리 항목을 참조하여 정은희(2002)의 위험요소 발생율과 영향 정도와 Pankaj(2003)에서 연구한 위험 요소들 간의 우선 순위를 참고하여 설문 문항을 작성하고 프로젝트 관리 경험이 많은 프로젝트 매니저와 지도 교수님과의 검토/협의 과정을 거쳐 작성하였다.

그리고 기존 연구에서 입증되어진 소프트웨어 프로젝트 개발 생명주기 단계에서 발생할 위험 요소들이 소프트웨어 프로젝트 지속 의지에 직접 및 간접적으로 영향을 미칠 것이라는 것을 제안하면서 다음 <그림-2> 같은 연구 모형을 설정하였다.

<그림-2> 연구 모형



소프트웨어의 개발은 기술의 변화가 크기 때문에 다른 분야에 비해서 예측 또는 평가가 어려우며, 개발자나 프로젝트 관리자와 같은 인간적인 요소에 대한 의존성이 높기 때문에 프로젝트의 관리와 개발 경험이 매우 중요한 역할을 한다.

본 연구에서는 프로젝트 위험관리에서 소프트웨어 개발 프로젝트의 생명주기 단계를 계획-> 요구사항-> 설계-> 코딩/통합/테스팅-> 설치/초기운영 5 단계로 나누어 모든 단계에서 발생한 위험 요소들이 프로젝트에 미치는 영향을 고려하여 프로젝트 지속 반응과 관련성을 분석한다.

(2) 설문 자료 수집 대상과 방법 및 기관

본 연구의 설문지 개발을 위해서 실무에 종사하고 있는 전문가들의 자문을 구하였으며, 또한 관련된 문헌 조사를 실시하였다. 설문지 내용은 최근에 참여한 프로젝트를 대상으로 일관성 있게 응답해야 하는 항목들로 구성된 설문지는 1차 예비 검토를 걸쳐서, 설문 내용이 본 연구에 적절하도록 수정 보완하였다. 표본 추출 대상은 우리나라 소프트웨어 개발 업체를 대표하는 기업들을 중심으로 추출하였다. 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 배포수는 200부로, 회수된 49부를 분석하였다. 소프트웨어 프로젝트 담당자들로부터 회수된 설문지는 전체의 50%이고, 응답이 부실한 2부를 제외한 총 47개의 설문지를 분석 대상으로 했다.

설문지 배포 및 수집은 전화를 통해 협조를 구한 뒤 인터넷을 이용하여 e-mail로 해당 기업이나 기관의 소프트웨어 개발 경험이 있는 프로젝트 관리자나 프로젝트의 세부 분야별 리더나 또는 경험이 풍부한 사원들에게 보내 e-mail에서 응답하여 회수를 할 수 있도록 하였으며, 해당 응답자가 아닐 경우 타인에게 전달 가능 등의 안내를 첨부하여 응답의 이용률을 높일 수 있도록 하였다. 그 외에 해당 업체를 직접 방문하여 설문을 회수하였다.

2. 설문 결과

(1) 주요 연구 결과

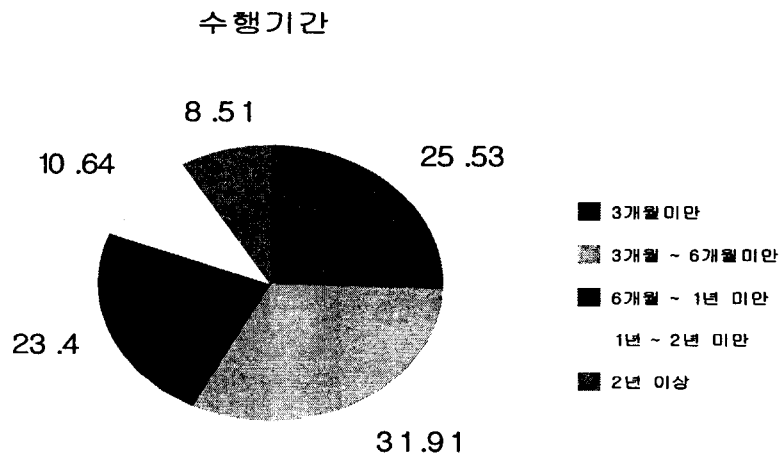
수집된 자료에 대해 일반적인 각종 현황자료 및 통계자료를 도표와 빈도 등의 비교를 통해 해당 프로젝트의 수행 기간과 인력 규모와 대상 영역, 기업/기관의 형태와 조직 규모, 소속 기업의 매출액, 응답자들의 직급과 실무경험, 프로젝트 내 수행 임무에 따른 차이를 살펴보고, 교차 분석을 통해 각 그룹간의 차이에 대한 유의적인 수준을 확인하고 데이터에 내재된 구조나 패턴을 찾아 연구 결과를 분석하였다.

본 연구에서는 위험 요소들이 소프트웨어 프로젝트 생명주기 각각의 단계에서 소프트웨어 프로젝트 수행의 성공에 영향을 미치게 되는지를 알아보는 것을 목적으로 한다.

이 연구를 통하여, 프로젝트 관리자들에게 프로젝트 성공에 부정적인 영향을 미치는 요소에 대한 이해를 통해 잘못된 프로젝트를 지속적으로 수행하는 실수를 방지함으로써 기업의 불필요한 자원 낭비를 막는데 도움을 줄 수 있을 것이다.

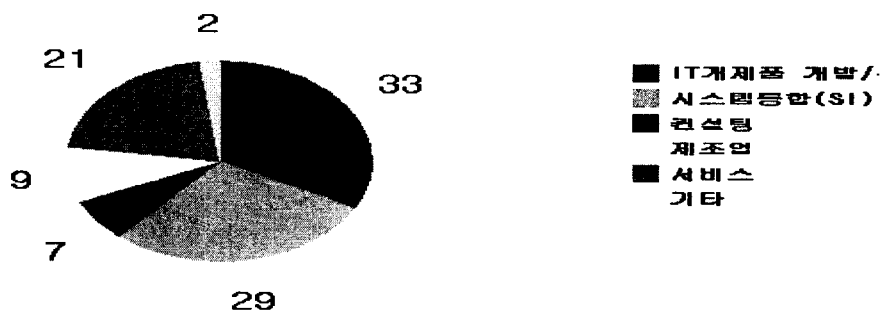
(2) 프로젝트의 수행 기간 및 기업/기관 유형

<그림-3> 해당 프로젝트의 수행 기간



<그림-4> 기업/기관 유형

기업/기관 유형



설문의 각 카테고리 응답분석 결과 <표-2>와 같이 나왔으며, 프로젝트의 성공에 부정적 위협 미치는 위험 요소가 프로젝트의 성공에 미치는 위협의 크기와 발생 단계에 대해 평균을 구하여 순위별로 정리하였다.

<표-2> 위험 요소 우선순위

(단위 : %)

순위	변수명	평균	표준편차	응답수
1	작업 일정의 지연	3.59	0.99	47
2	사용자의 요구사항의 변경과 증가	3.45	1.16	47
3	인력의 부족	3.32	1.16	46
4	프로젝트 팀 지식/경험의 부족 : 개발 대상 업무에 대한	3.13	1.07	46
5	의사소통의 부족/실패: 프로젝트 팀과 사용자/고객간의	3.13	1.15	47
6	오류: 사용자 요구사항	3.11	0.99	46
7	프로젝트 팀 지식/경험의 부족 : 개발 방법론에 대한	3.11	1.06	46
8	오류: 코딩, 통합 및 테스트	2.96	1.12	47
9	프로젝트 팀 지식/경험의 부족 : 기술에 대한 (예: 새로운 H/W, S/W의 사용으로 인한)	2.96	1.05	46
10	개발 범위의 불명확함	2.94	1.13	47
11	오류: 설계	2.85	1.11	47
12	의사소통의 부족/실패: 프로젝트 팀 내의	2.77	1.13	47
13	사용자의 참여 부족	2.74	1.11	46
14	비용 초과 위험 크기	2.61	1.13	46
15	새로운 시스템의 필요성에 대한 이해관계자들간의 공감대 부족	2.60	0.96	45
16	개발 책임자의 권한/의지 부족	2.59	1.02	46
17	프로젝트 요원의 교체	2.41	1.24	46

위험 요소들의 위협의 크기는 5점 척도를 통해 구하였다. 위험 요소들의 위협의 크기 평균은 2.5 이상의 값으로 대부분 상위 쪽에 분포되어 있는 것으로 나타났다. 설문지 분석 결과 <표-2>에서 나타난 위협의 크기가 가장 큰 위험 요소는 작업 일정의 지연으로 일정을 얼마나 정확하고 효율적으로 운영하는가가 프로젝트의 성공에 열쇠라고 할 수 있다.

두 번째 위험 요소인 사용자의 요구사항의 변경과 증가는 요구분석의 오류 혹은 잘못 된 분석자료, 그리고 이에 따른 요구사항의 변경과 증가를 수반하게 되어 중

요한 위협으로 나타나고 있음을 확인하였다. 즉, 정확한 요구사항 분석의 중요성을 지적하고 있는 것이라고 할 수 있다.

세 번째와 네 번째로 인력의 부족의 위협 요소와 개발 대상 업무에 대한 프로젝트 팀 지식/경험의 부족의 위협 요소도 상위 순위를 차지했다.

작업 일정 지연의 위협요소와 요구사항의 변경과 증가 그리고 인력의 부족과 개발 대상 업무에 대한 프로젝트 팀 지식/경험의 부족의 위협 요소는 프로젝트의 성공에 얼마나 큰 영향력을 미치는가를 심각하게 제기되고 있는 본 응답에서 확인 할 수 있었다. 그러나 본 설문 대상이 프로젝트 관리자 중심의 설문이다 보니 사용자 입장에서의 위협 요소를 다룬다기 보다는 소프트웨어 개발 프로젝트에 한해서 개발자나 관리자 입장에서 해석을 해 나가고 있다는 것을 다시 한번 밝힌다.

다섯 번째 프로젝트 팀과 사용자/고객간의 의사소통의 부족/실패의 위협 요소 역시 프로젝트 개발의 성공을 위해서 반드시 해결해야 할 문제이다. 고객과 부서간의 이해관계로 인한 대립은 중요한 의사결정을 적절한 시기에 해주지 못하는 경우가 발생하게 되며 이는 프로젝트의 지연을 초래하게 되고 방향 설정에 어려움을 주게 된다.

여섯 번째 사용자의 요구사항 위협 요소로 상위 위협 요소에서 원인이 되었던 해당 위협 요소는 프로젝트 개발 할 당시 기본적으로 해결을 하고 시작을 하기 때문에 큰 위협으로 나타나지는 않은 것으로 보여진다.

그 다음 위협 요소는 코딩, 통합 및 테스트 오류 업무 영역에 대한 프로젝트 팀의 지식이 부족하여 발생할 수 있는 문제이다.

기술에 대한 프로젝트 팀 지식/경험의 부족의 위협 요소는 새로운 소프트웨어나 하드웨어 또는 경험해 보지 못한 소프트웨어나 하드웨어 사용시 시스템 운영자가 사용에 대한 적절하게 교육을 받지 못했을 경우 위협에 처하게 된다. 더욱이 기존 하드웨어와 새로운 하드웨어 간에 통합에 문제가 발생하게 되면 사용자가 만족하지 못하는 프로젝트 산출물을 만들어 사용자에게 불신과 실망을 안겨주게 되고 수정하는데 필요한 시간과 태스크 완료시간이 늦어질 수 있게 되어 결국 가장 큰 위협 요소인 작업 일정 지연에 도달하게 된다.

그러나 해당 위협 요소는 새로운 환경, 새로운 기술을 적용하게 되면 시행 착오와 개발 기간에 사전에 고려하는 사항이 있어야 할 것이다. 그리고 일반적인 경향이 검증되지 않은 환경에서는 개발을 피하려는 것이 일반적인 프로젝트 관리자들의 성향으로 나타나 큰 위해를 주는 위협 요소로 응답되지는 않았다.

열 번째 위협 요소인 프로젝트의 불명확한 개발 범위는 프로젝트 범위가 제대로 정의되어 있지 않은 상태에서 프로젝트 팀의 초점이 결여되고 프로젝트 경계 밖의 영역을 조사하여 프로젝트를 진행 시켜나가는 노력의 낭비를 초래할 수 있다.

V. 결 론

1. 연구 결과 및 시사점

최근에는 인터넷을 활용한 전자상거래의 증가로 소프트웨어 개발 프로젝트 규모와 횟수가 급속도로 증가하고 있다. 그러나 그 증가에도 불구하고 잘못된 프로젝트를 조기에 중단하거나 변경하지 못함으로 인해 프로젝트의 실패율은 매우 높다. 따라서 프로젝트 수행의 지속행위에 대한 연구가 절실히 필요하다.

이에 본 연구에서는 소프트웨어 개발 프로젝트에 있어서 수행의 지속 행위를 설명할 수 있는 체계적이고 설득력 있는 이론을 제시한 선행 연구를 참고하여 조사하였다. 그리고 기존의 연구 결과들과 비교를 통하여 각각의 소프트웨어 프로젝트 생명주기 각각의 단계에서 존재하는 위험 요소에 따라 달라지는 대응 반응의 영향을 연구해 볼 수 있었다.

프로젝트를 위한 위험 요소는 발생 여부가 확실하지 않은 조건이긴 하지만, 프로젝트에 영향을 미칠 수 있다. 본 연구에서는 소프트웨어 프로젝트를 위한 위험 요소의 개념과 위험 관리의 일반 개념에 대해 설명하였다. 또한 각 영역별로 구분된 17가지 위험 요소들이 프로젝트 지속에 어떻게 영향을 미치는지에 대해서 연구를 하였다.

설문 결과를 분석한 결과 발생빈도가 아주 높은 위험 요소가 있었으며, 그 위험 요소들이 생명주기 각 단계별로 발생하는 발생빈도는 차이가 있었다. 프로젝트 실무 경험자들의 가장 큰 공통된 의견은 프로젝트 진행 중의 작업 일정의 지연 문제였다. 이를 위한 해결책으로는 개발자 입장에서는 충분한 계획과 사전 경험을 토대로 개발을 진행시켜 나가야하며 이해관계자들인 고객의 입장에서는 끊임없는 참여와 관심을 갖고 상호간의 의사소통을 통해서 해결해 나가야 한다. 그러므로 소프트웨어 개발 프로젝트의 실패를 줄일 수 있는 원천이 될 것이다.

2. 연구 결과의 한계

발생할 수 있는 위험에 대한 대응 방법이 프로젝트의 지속 여부에 영향을 미치는 정도가 단계별로 다를 수 있다. 본 연구에서는 위험 분류 체계에 있어서 영역별 위험의 세분화된 분류를 하는데 있어서 미흡한 부분이 존재한다. 수집된 데이터의 양이 적어서 일반화를 시키기엔 너무 부족하여 차후 연구에서는 미흡한 데이터를 보강하여 좀더 실증적인 분석에 의한 분류를 실시할 수 있었으면 좋겠다. 한 단계 나아가 실증적인 분석에 의해 분류된 위험 요소들이 위험의 반응에는 어떤 영향을 미치는가를 연구하는 것도 좋겠다.

참고문헌

- 1) 김기윤, "정보기술에 대한 위험분석방법", 기업경영연구, 광운대학교 기업경영연구소, vol. 3, 1994.11, pp.1-18.
- 2) 장활식, 김종기, 오창규, 임호섭 "정보시스템 프로젝트의 비정상적 종료에 영향을 미치는 요인에 관한 연구, 1999 , pp.231-239.
- 3) 정은희, "프로젝트의 위험과 위험 관리가 소프트웨어 개발 프로젝트 성과에 미치는 영향, 2002.
- 4) Alter, S., "Implementation Risk Analysis", TIMS Studies in the Management Sciences, North Holland, 13, (1979), 103-119.
- 5) Alter, S.,and M. Ginzberg., "Managing Uncertainty in MIS Implementation", Sloan Management Review, 20, 1 (1978), 23-31.
- 6) Anton, D. Buttingeg B.Sc., Risk Management in a Software Development Life Cycle, 참조일: 2003.07.08.
- 7) Barki Henri, An Integrative Contingency Model of Software Project Risk Management, JMIS, 2001.
- 8) Barki, H. and J. Hartwick, "Measuring User Participation, User Involvement, and User Attitude," MIS Quarterly, 18, 1 (1994), 59-82.
- 9) Boehm, B. W., " Software risk management: Principles and practices. IEEE Software, 8(1), 1991, pp.32-41.
- 10) Boehm, B.W., Software Risk Management, IEEE Computer Society Press, Los Alamitos, CA, 1989.