

중소규모 조직을 위한 소프트웨어 위험관리 프로세스 프레임워크 개발

박영미, 도성룡*, 한혁수**
*상명대학교 일반대학원 컴퓨터학과
**상명대학교 컴퓨터과학부
e-mail : fullymp@smu.ac.kr

Developing a Framework of Software Risk Management Process for Small Organizations

Young Mi Park, Sung Ryong Do*, Hyuk Soo Han**
*Dept. of Computer Science, Sang Myung University
**School of Computer Science, Sang Myung University

요 약

소프트웨어 프로젝트는 본질적으로 모호한 요구사항, 정확하지 않은 산정과 기술의 어려움 등 많은 불확실성을 포함하고 있다. 이러한 불확실성은 프로젝트가 내부에 잠재적인 문제인 위험(Risk)을 지니고 있다는 것을 의미한다. 위험은 발생되면 프로젝트 성공에 부정적 영향을 미칠 수 있다. 따라서 이러한 위험을 사전에 예방하고 최소화하기 위한 프로세스(Process)의 정립이 필요하다. 상당수의 중소기업 조직이 이러한 위험관리의 중요성을 인지하고 있지만, 위험관리 프로세스를 정의하고 이행하는 것은 전문적인 지식을 요구한다. 따라서 특정 조직과 프로젝트에 적합한 프로세스가 안정적 단계에 이르기 위해서는 많은 시행착오의 과정이 필요하다. 이러한 어려움 때문에 중소기업 조직에서는 쉽게 프로세스 구축을 시도하지 못하고 있다. 관련 표준과 참조모델에서도 위험관리 프로세스 영역을 포함하여 중요성을 강조하고 있지만, 상위수준의 활동목표와 수행되어야 하는 프랙티스(Practice)만을 제시하고 있고 세부적인 절차나 방법들에 대해서는 언급하지 않고 있다. 이에 본 논문에서는 PMBOK 과 CMMI, 그리고 기존 전문가들의 연구를 바탕으로 소프트웨어 위험관리 프로세스 프레임워크(Framework)를 개발하였다. 중소기업 조직에서는 이 프레임워크를 활용하여 규모와 프로젝트의 특성을 반영한 위험관리 프로세스를 정의하고 적용할 수 있을 것이다.

1. 서론

소프트웨어 개발은 자주 요구사항이 변경되고, 개발자에 의존도가 높기 때문에 기간 및 비용을 정확히 산정하거나, 참여 인력의 규모를 파악하기 어렵다. 소프트웨어가 가지고 있는 비가시적인 특성 때문에 프로젝트 개발 과정에 대한 정확한 이해나 진행상황을 파악하기도 힘들다. 이러한 요소들은 납기 지연, 비용 초과 등을 유발하여, 소프트웨어 프로젝트를 실패로 가져갈 수 있다. 그러므로 소프트웨어 프로젝트에서는 프로젝트가 시작하는 시점에서 프로젝트에 발생할 수 있는 위험들에 대해 미리 짚어보고, 이에 대한 대응을 준비해 놓는 위험관리 프로세스가 필요하다.

하지만 조직의 적합한 위험관리 프로세스를 정의하는 것은 많은 경험과 전문적인 지식을 요구한다. 정의된 프로세스가 여러 번의 개선과정을 거쳐 안정적 단계에 이르기 위해서는 많은 시행착오의 과정이 있어야 한다. 뿐만 아니라 대부분의 개발자들은 위험관리 활동이 추가적인 작업이나 비용만을 발생시킨다는 오해를 가지고 있기 때문에 중소기업 조직에서는 위험관리 프로세스 구축을 시도하지 못하고 있다[1].

관련 표준인 PMBOK(Project Management Body of

Knowledge)이나 CMMI(Capability Maturity Model Integration)에서는 위험관리 프로세스 영역을 제시하고 있지만, 해야 할 일들(What)만을 정의하고, 어떻게 구현해야 하는 가에 대한 방법(How)들은 각 조직에게 맡기고 있다.

기존 연구에서는 프로젝트의 성공에 영향을 미치는 위험요인을 파악하기 위해 실무경험자들로부터의 설문 결과를 분석하거나[2], 위험관리가 반영된 프로젝트 관리 모형을 제시하고 이 모형이 프로젝트 성과에 미치는 영향을 검증하는[3] 등 위험관리의 중요성만을 강조해왔다. 중소기업 조직들이 프로세스의 구축을 위해 참조할 수 있는 위험관리 프로세스 구축 지침에 관한 연구는 부족한 실정이다.

이에 본 연구에서는 PMBOK 과 CMMI 의 위험관리 그리고 전문가들의 기존 연구를 기반으로 중소기업 조직에서 적용 가능한 체계적이고 구조적인 소프트웨어 위험관리 프로세스 프레임워크를 개발하였다. 위험관리 프로세스를 정립하고자 하는 중소기업 조직은 이 프레임워크를 활용하여 효과적으로 조직의 규모와 프로젝트의 특성에 적합한 위험관리 프로세스를 정의할 수 있을 것이다.

2. 관련연구

위험관리란, 프로젝트 목적에 대해 긍정적인 사건의 확률(가능성) 및 결과는 극대화하고 부정적인 사건의 확률 및 결과는 최소화 하는 것이다. 여기서 위험은 프로젝트를 진행하면서 일정, 비용, 범위, 품질과 같은 프로젝트 요소들에 치명적인 영향을 주어 프로젝트를 실패로 이끌 수 있는 이벤트 또는 상태를 말한다[4].

위험관리 활동들은 표준이나 모델, 기존 연구마다 조금씩 다르게 표현하고 있다.

미국 PMI(Project Management Institute)에서 개발한 PMBOK 은 9 개의 지식영역(Knowledge Area)과 44 개의 프로세스(Process)로 구성되어 있다. 9 개의 지식영역 안에 포함된 위험관리(Risk Management) 영역은 위험 관리 계획(Risk Management Planning), 위험 식별(Risk Identification), 정성적 위험 분석(Qualitative Risk Analysis), 정량적 위험 분석(Quantitative Risk Analysis), 위험 대응 계획(Risk Response Planning), 위험 모니터링 및 통제(Risk Monitoring and Control) 등 6 개 단계로 구성되어 있다[4].

미국 카네기 멜론 대학에서 개발한 소프트웨어 프로세스 개선모델인 CMMI 는 성숙도가 높은 소프트웨어 개발 조직이 갖추어야 할 22 개의 프로세스 영역을 1 부터 5 까지 5 단계의 성숙도 수준으로 구성하고 있다. 성숙도 단계 3 에 속하는 위험관리 프로세스 영역은 3 개의 목표와 7 개의 프랙티스로 구성되어 있다. 그 중 위험관리 준비는 위험요소 및 분류정의, 매개변수 정의, 전략 수립으로 구성되며, 위험 식별과 분석에는 위험식별, 위험평가, 위험분류, 위험우선순위 결정이 있으며, 위험 완화는 위험완화 계획 개발과 위험 완화 계획 이행으로 구성되어 있다[5].

Boehm 은 위험관리 활동을 크게 위험평가(Risk Assessment)와 위험통제(Risk Control)로 구분하였다. 위험평가는 위험식별, 위험분석, 위험 우선순위로 구성하고, 위험통제는 위험관리 계획, 위험해결, 위험 모니터링으로 구성하였다[6].

각각의 위험관리 활동들은 단계별 활동 명칭이 조금씩 차이가 있거나 하나의 단계를 여러 단계로 표현했을 뿐 그 속에 담겨있는 내용은 비슷하다.

각 모델들은 성공적인 위험관리를 위해 필요한 프로세스들(What)만을 정의하고, 그 프로세스들을 어떻게 구현해야 하는 가에 대한 방법(How)들은 각 조직에게 맡기고 있다. 즉 중소기업도 조직들이 프로세스의 구축을 위해 참조할 수 있는 위험관리 프로세스 구축 지침에 관한 연구는 부족한 실정이다.

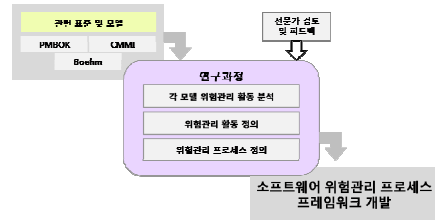
이에 본 연구에서는 중소기업도 조직에서 적용 가능한 체계적이고 구조적인 소프트웨어 위험관리 프로세스 프레임워크를 개발하였다.

3. 중소기업 조직을 위한 소프트웨어 위험관리 프로세스 프레임워크 개발

3.1 소프트웨어 위험관리 프로세스 프레임워크 개발 과정

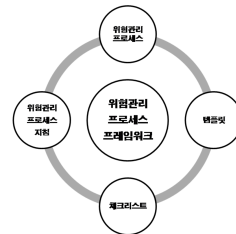
본 연구에서는 (그림 1)과 같이 소프트웨어 위험관리 프로세스 프레임워크를 개발하기 위하여 선행작업으로 PMBOK 과 CMMI 그리고 Boehm 이 제시하는 위험관리 활동을 분석하였다. 분석된 위험관리 활동들을 기반으로, 중소기업도 조직에 적합한 위험관리 프로세스를 정의하였고, 최종적으로 전문가의 검토를

받아 위험관리 프로세스 프레임워크를 개발하였다.



(그림 1) 소프트웨어 위험관리 프로세스 프레임워크 개발 과정

위험관리 프로세스 프레임워크는 (그림 2)와 같이 위험관리 프로세스, 프로세스 지침, 프로세스 단계별 템플릿 그리고 프로세스의 각 활동을 검토할 수 있는 체크리스트로 구성된다.

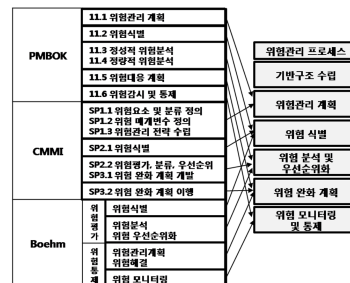


(그림 2) 소프트웨어 위험관리 프로세스 프레임워크 구성

본 논문의 3 장에서는 위험관리 프로세스와 지침만을 제시하고, 4 장에서 템플릿과 체크리스트를 포함한 간단한 예를 제시하였다.

3.2 위험관리 프로세스와 지침

본 연구의 위험관리 프로세스는 각 모델들이 공통적으로 표현하는 활동들을 통합하여 (그림 3)과 같이 6 단계로 정의하였다. 6 단계는 위험관리 활동의 기반구조를 수립하는 단계를 포함하여 위험관리 계획, 위험식별, 위험분석 및 우선순위화, 위험 완화 계획, 위험 모니터링 및 통제 활동으로 구성된다.



(그림 3) 위험관리 프로세스

각 프로세스 단계별 지침은 다음과 같다.

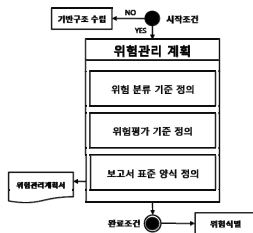
3.2.1 기반구조 수립

기반구조 수립은 위험관리 활동의 수행을 위하여 반드시 확립되어야 하는 단계로, 정책 확립, 자원 할

당, 담당자 임명, 팀 구성원에게 필요한 교육을 수행한다.

3.2.2 위험관리 계획

위험관리 계획은 프로젝트에서 수행할 위험관리 활동을 어떻게 접근하고 계획 할 것인지를 정의하는 단계로, (그림 4)과 같이 진행된다. 먼저 위험의 분류 기준을 정의한다. 위험 분류 기준은 발생될 수 있는 위험들을 특징 별로 분류한다. 일반적으로 위험 분류는 프로젝트 위험, 기술 위험, 비즈니스 위험으로 분류한다[7]. 하지만 분류 기준은 각 조직의 프로젝트 특성에 맞게 정의할 수 있다. 위험 평가 기준은 위험 요소들의 우선순위를 정의하기 위한 것으로 프로젝트의 특성에 맞게 위험요소의 발생 가능성(Risk Likelihood)과 위험 영향도(Risk Impact)의 범위를 정의한다. 위험 발생 가능성은 위험이 발생할 확률(Probability)을 말하며, 위험 영향도는 위험이 발생할 경우 프로젝트에 미치는 손실(Loss)을 말한다. 또한 각 단계별 문서들의 표준 양식을 정의한다.



(그림 4) 위험관리 계획 프로세스

3.2.3 위험 식별

위험 식별은 프로젝트에 영향을 미칠 수 있는 위험요소들을 수집하는 것이다. 위험요소 수집방법에는 브레인스토밍(Brainstorming)방식, 델파이(Delphi)방식, 인터뷰 방식 등이 있다. 수집된 위험요소들은 위험식별 보고서에 작성하여 위험관리 담당자에게 보고한다.

3.2.4 위험 분석 및 우선순위화

위험 분석 및 우선순위화는 식별된 위험요소들을 가지고 위험 발생 가능성과 영향도를 평가하여 우선순위를 정의하는 것이다. 프로젝트를 진행하면서 가장 발생하기 쉽고 또한 심각한 영향을 미칠 수 있는 위험이 높은 우선순위를 갖게 된다. 우선순위가 결정된 위험요소들을 위험 분석 및 우선순위 평가결과서에 기록한다.

3.2.5 위험 완화 계획

위험 완화 계획은 위험의 발생을 최소화 하기 위한 계획과 위험 발생시 조치 활동을 결정하고 이행하는 것이다. 각 위험들의 완화 계획은 명확하고 구체적인 수단으로 작성하는 것이 바람직하다. 그리고 위험 완화 활동 수행자를 명확히 구분하여야 한다. 완화 활동 수행 후 중복으로 수행하지 않기 위해 각 활동의 완료시점에 대해서도 명확히 해야 한다.

3.2.6 위험 모니터링 및 통제

위험 모니터링 및 통제는 식별된 위험의 추적, 잠재되어 있는 위험의 모니터링, 프로젝트 수행 중에 새롭게 발생하는 위험의 식별 등 지속적으로 위험요소들을 모니터링 하고 통제하는 것이다. 이 단계에서는 위험 모니터링 및 통제 결과서를 작성하여, 다른 프로젝트에서 유사 위험이 발생하는 것을 예방할 수 있도록 한다[8].

위험관리 계획부터 모니터링 및 통제까지 위험관리 활동을 수행하면서, 전체적인 단계별 활동들이 위험 발생 예방에 어느 정도 효과가 있었는지 평가하고 문서화 하는 것도 중요하다.

4. 소프트웨어 위험관리 프로세스 프레임워크 적용 및 검증

소프트웨어 위험관리 프로세스 프레임워크의 효과를 검증하기 위하여, 배낭 여행 웹사이트 개발 프로젝트에 적용하였다. 배낭여행 웹사이트 개발 프로젝트는 프로젝트 관리자 1명, 웹 기획자 1명, 웹 디자이너 1명, 웹 프로그래머 2명으로 구성되고, 9주에 걸쳐 웹사이트를 개발하는 프로젝트이다. 개발환경은 Windows Server 2003, php, My-sql, Apache Server 와 Tomcat 을 사용하였다.

먼저 프레임워크를 기반으로 위험관리 프로세스를 정의하였으며, 그 정의된 프로세스에 따라 위험관리 활동들을 수행하였다.

4.1 기반구조 수립

프로세스에 따라 프로젝트 관리 규정에 정책을 확립하고 자원으로는 문서도구를 선정하였다. 소규모의 프로젝트로서 팀 구성원이 5명인 것을 감안하여, 프로젝트 관리자를 위험관리 담당자 및 조치 담당자로 임명하였다. 프로젝트 참여인원 전원에게 위험관리 프로세스 교육을 실시하였고, 담당자인 프로젝트 관리자에게는 산출물 작성 방법 등을 교육하였다.

4.2 위험관리 계획

위험 분류 기준은 프로젝트 관리부부와 프로젝트 개발부분으로 구분하였다. 일반적으로 위험분류에 포함되는 비즈니스 관련 부분은 이 웹사이트가 상업용이 아니기 때문에 제외시켰다.

관리부분에는 계약사항, 예산, 비용, 일정 등에 영향을 미칠 수 있는 위험들이 식별되었고, 개발부분에는 요구사항 분석, 설계, 코딩, 테스트 등 개발 주기 동안 발생할 수 있는 위험들이 포함되었다. 위험 평가 기준은 팀 구성원이 회의를 통해 이전에 수행된 유사 프로젝트들의 위험 발생 빈도와 그 영향도를 참고하여 다음과 같이 정의하였다.

- 위험 발생 가능성
 - 상 - 60% 이상
 - 중 - 15% 이상 ~ 60% 미만
 - 하 - 15% 미만
- 위험 영향도
 - 심각 - 70% 이상
 - 중요 - 15% 이상 ~ 70% 미만
 - 경미 - 5% 이상 ~ 15% 미만
 - 무가치 - 5% 미만

4.3 위험 식별

본 프로젝트에 대한 위험은 구성원들의 브레인스토밍을 통하여 식별하였다. 관리부분에는 개발자 이탈,

관리자의 평가, 하드웨어 고장의 위험이 식별되었고, 개발 부분에는 새로운 기능 추가 등의 위험이 식별되었다.

4.4 위험 분석 및 우선순위와 완화계획

식별된 위험요소를 가지고 위험 관리 계획에서 정의한 평가 방법을 통해, 우선순위를 결정하였다. 결정된 위험요소에 대한 완화 계획을 다음과 같이 수립하고 계획에 따라 수행하였다.

NO	위험 분류	위험 요소	위험기준	발생 가능성	영향도	우선 순위
1	M	개발자 이탈	경력 10년 이상 경력 3년 이상 ~ 10년 미만 경력 3년 미만	하 하 하	중요 중요 경미	4 4 5
		설명	프로젝트 수행 중 개발자 이직			
		완화 계획	- 관리자와 격일로 자유로운 의사소통 수행 (개발자의 업무, 건강 등) - 편안한 작업환경 제공 - 이탈 발생시 대체 인력에 대한 보상 (대체인력 나머지 개발자 및 웹 디자이너)			
2	M	프로젝트 관리자 평가	10일 이상 3일 이상 ~ 10일 미만 3일 미만	하 하 하	중요 중요 무가지	4 4 5
		설명	프로젝트 전체 관리를 맡고 있는 관리자가 평가			
		완화 계획	- 대체 인력 교육(대체인력- 웹 기획자)			
3	E	기능 추가	1000만원 이상 100만원 이상 ~ 1000만원 미만 100만원 미만	중 상 상	심각 심각 중요	2 1 2
		설명	새로운 기능 추가			
		완화 계획	- 요구사항 분석단계에서 고객과 충분한 의사소통 수행 - 기능 추가에 대한 회의 소집(결승원 제의의)			
4	M	하드웨어 고장	작업중단 발생	하	심각	3
		설명	하드웨어 고장으로 업무 중단			

위험 완화 계획을 수행하면서 각 활동들의 수행 여부를 다음과 같은 체크리스트를 사용하여 검토하였다.

NO	질문	YES	NO	N/A	비고
1. 완화 계획 수립					
1.1	관리자와 개발자 사이의 의사소통이 격일로 이루어지고 있는가?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1.2	작업환경에 대한 개발자들의 불만사항은 없는가?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1.3	개발자 이탈 발생시 대체인력에 대한 교육이 이루어졌는가?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1.4	프로젝트 관리자 평가 발생시 대체 인력 교육이 제대로 이루어졌는가?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1.5	개발 환경 점검이 제대로 이루어졌는가?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
2. 완화 활동 수행					
2.1	...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

4.5 위험 모니터링 및 통제

프로젝트 관리자는 주간회의를 통해 식별된 위험요소들에 대해 모니터링 및 통제 결과를 보고 받았다.

웹사이트 개발 3 주차 주간회의에서 프로젝트 관리자의 평가가 10 일 이상인 위험이 발생되었고 프로젝트 관리자는 프로젝트 전체 관리 및 위험관리 담당과 조치 담당을 동시에 담당하고 있어 프로젝트 수행에 큰 차질이 빚어질 수 있음을 감지하였다. 하지만 위험 완화 계획으로 대체 인력인 웹 기획자에게 프로젝트 관리자 맡고 있는 관리자 역할과 위험관리 담당자

역할을 교육 해왔기 때문에 영향을 최소화 할 수 있었다.

또한 개발 7 주차에 프로젝트 착수 전 완화계획에 따라 개발 환경 점검을 했음에도 불구하고 웹 디자이너 작업 PC 가 장비노후로 인해 디자인 작업이 불가능하였다. 하지만 위험완화 계획으로 장비 대여 방안을 강구해 두었기 때문에 조속히 장비를 대여할 수 있었고 위험 발생을 예방할 수 있었다.

5. 결론 및 향후 연구 방향

소프트웨어 개발 프로젝트에서 발생할 수 있는 위험들에 대해 미리 짚어보고, 이에 대한 대응을 준비해 놓는 위험관리 프로세스는 성공을 위해 필요하다.

하지만 위험관리 관련 표준 및 참조 모델에서는 상위수준의 활동목표와 수행되어야 하는 프랙티스만을 제시하고 있고, 세부적인 절차나 방법들에 대해서는 언급하지 않고 있다. 따라서 중소기업 조직들은 프로젝트를 수행하면서 위험 관리를 수행하기가 쉽지 않았다. 이에 본 연구에서는 위에서 제시한 한계점을 극복하기 위해, PMBOK 과 CMMI 등 기존 전문가들의 연구를 바탕으로 중소기업 조직들을 위한 소프트웨어 위험관리 프로세스 프레임워크를 개발하였다.

이 프레임워크를 통해 중소기업 조직에서는 관련 표준 및 모델에 기반한 위험관리 프로세스를 구축하고 위험관리 능력을 향상시킴으로써 프로젝트 성공을 높일 수 있을 것이다.

위험관리 프로세스 프레임워크의 효과를 확인하기 위해 소규모의 배낭 여행 웹사이트 개발 프로젝트에 적용하였고, 그 결과 위험관리 활동에 대한 효용성을 확인할 수 있었다.

향후 연구에서는 위험관리 프로세스 프레임워크의 효용성을 객관적인 지표를 통해 측정하는 것이 요구된다. 또한 다양한 프로젝트를 대상으로 위험관리 프로세스 프레임워크를 적용하고 그 결과를 바탕으로 지침을 개선 할 필요가 있다.

참고문헌

- [1] Y. H. Kwak, J. Stoddard, "Project risk management: Lessons learned from software development environment", 2004
- [2] 김경내, "소프트웨어개발 프로젝트의 위험관리", 석사학위논문, 건국대학교, 2000
- [3] 서창교, 정은희, "프로젝트 위험과 위험관리가 소프트웨어 개발 프로젝트 성과에 미치는 영향", 경영정보학연구, 2003
- [4] Project Management Institute, Inc., "A Guide to the project management body of knowledge (PMBOK)", 3rd edition, 2004
- [5] Mary Beth Chrissis, Mike Konrad, Sand Shrum, "CMMI Second Edition: Guidelines for Process Integration and Product Improvement", Addison-Wesley, 2006
- [6] Barry W. Boehm, "Software Risk Management: Principles and Practices", IEEE Software, January 1991
- [7] Pressman, Roger S., Ph.D., "Software Engineering: A Practitioner 's Approach 5th", McGraw-Hill College, 2001
- [8] 한혁수, "소프트웨어 공학의 소개", 홍릉과학출판사, 2008