소프트웨어 개발 프로젝트를 위한 RMS 기반의 시스템 안전성 개선방안 연구

On the Improvement of Software Development Project for System Safety based on RMS

김 종 걸* · 박 지 성** · 김 형 만**

Jong-Gurl Kim* · Ji-Sung Park** · Hyung-Man Kim**

Abstract

IT기술의 발달과 결부된 소프트웨어분야의 지속적인 기술 개발은 IT기술의 이용효율성을 증대시켜서 컴퓨터의 이용범위 확대와 이용률을 제고 시키고 있다. 컴퓨터의 이용률이 높아지면서 다양한 분야에서의 소프트웨어 개발에 대한 필요성이 제기되었고 소비자들은 원하는 소프트웨어 선택의 폭을 넓히고 있다. IT분야는 소프트웨어가 IT자체의 제품선택에 영향을 미치는 중요한 관건이 되는 시기로 접어들게 되었다.

본 연구에서는 IEC61508을 기반으로 하여 시스템 안전을 위한 소프트웨어 개발 프로젝트 개선에 관한 통합적인 접근 방법을 제안한다.

Keywords: 소프트웨어, 프로젝트, 소프트웨어 안전성, 소프트웨어 품질관리 IEC 61508, ISO/IEC 15504

* 성균관대학교 시스템경영공학과

** 성균관대학교 산업공학과

1. 서 론

최근 급속한 IT기술의 융복합화로 사용자가 요구하는 기능이 다양해짐에 따라 소프트웨어의 품질측정이 갈수록 중요해 지고 있다. 소프트웨어 개발에 있어서는 기능성, 신뢰성, 안전성 등이 중요한 요소들로 꼽히고 있다[6]. 이러한 요소들은 소프트웨어 개발 프로젝트에서 중요하게 다뤄져야 하며 지속적인 연구, 개발 또한 같이 이루어져야 한다. 소프트웨어의 품질 확보를 위한 국제표준이 강화됨에 따라 국내 소프트웨어 개발 업체들도 국제기준에 부합하는 조치가 요구되고 있다. 소프트웨어개발 부분에 있어 국제적인 안전성 및 신뢰성 기준을 만족하는 기준을 충족하는 개발 프로젝트를 거친다면 소프트웨어의 품질을 국제적인 수준으로 향상시킬 수 있을 것이다. 본 연구에서는 프로젝트 관리 수명주기, 소프트웨어 개발 프로젝트 수명주기, 안전 수명주기의 효율적인 통합방안을 이론적으로 제시하고자 한다.

2. 이론적 고찰

2.1 프로젝트관리

일반적으로 관리는 어떤 목표를 달성하기 위하여 재료나 사람 같은 필요한 자원을 조화롭게 배분하고 운영하는 의사결정, 의사전달, 통솔 등의 노력이라 할 수 있다. 프로젝트 관리 지식 체계(PMBOK : Project Management Body Of Knowledge)는 미국 프로젝트 관리 협회(PMI : Project Management Institute)에서 효율적인 프로젝트 관리를 위해 발간한 관리 지침서이다[10]. 프로젝트 관리 지식 체계 가이드(2004)에서는 “프로젝트관리는 프로젝트 요구사항을 충족시키기 위하여 관련지식, 기량, 도구 및 기법 등을 프로젝트 활동에 적용하는 것”이라고 정의하고 있다. 또한 이와 같은 프로젝트 관리는 프로젝트 착수, 기획, 실행, 감시 및 통제, 종료단계로 진행되는 프로젝트관리 프로세스의 통합과 적용을 통해 이루어지며, 여기에는 다음과 같은 활동이 포함된다고 하였다.[5]

프로젝트 요구사항의 식별
°분명하고 달성 가능한 프로젝트 목표의 설정
°프로젝트의 품질, 범위, 원가, 일정에 대한 요구조건 간에 적절한 균형 유지
°다양한 프로젝트 이해관계자의 서로 다른 관심사항과 기대치에 부응하는 사양이나 계획 및 접근방식의 채택

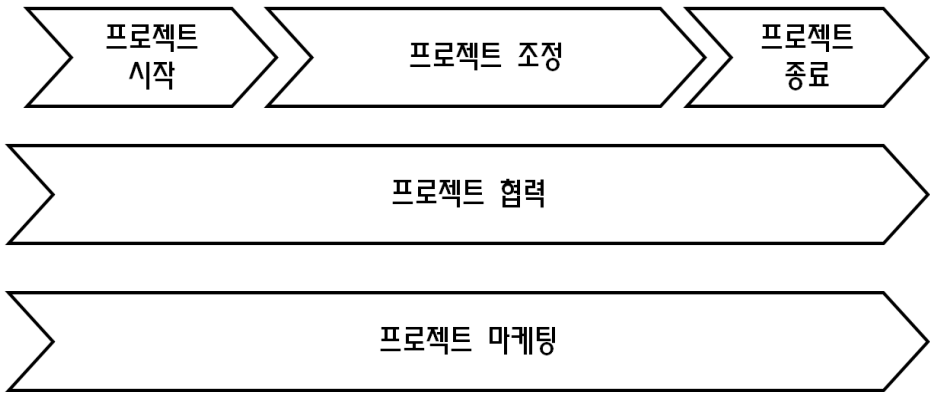
2.1.1 프로젝트관리 수명주기

프로젝트관리를 효과적으로 수행하기 위하여 프로젝트를 연속되는 여러 단계로 세분화할 수 있다. 이와 같이 세분된 단계를 총칭하여 프로젝트 수명주기라고 한다. 프로젝트 수명주기는 일반적으로 프로젝트 시작, 프로젝트 조정, 프로젝트 종료 등 크게

3가지의 하부과정으로 나누어진다. 동시에 프로젝트 협력과 프로젝트 마케팅은 계속적으로 수행된다[5]. 프로젝트 수명주기는 각각의 단계에서 프로젝트 환경의 변화나 전략의 변경, 프로젝트 타당성 재검토의 필요성 등으로 인해 단계별 평가가 필수적 이며 이것이 프로젝트관리 수명주기의 주된 목적이라 할 수 있다[9].

프로젝트 시작: 프로젝트 시작과정의 목표는 프로젝트의 목표를 정하는 것과 전체적인 계획을 정하는 것이다. 프로젝트 계획은 프로젝트 조절과 협력을 위한 기초를 제공한다.

프로젝트조정: 프로젝트 진행과정은 매우 역동적이기 때문에 프로젝트를 조정하는 것은 필수적이라 할 수 있다. 프로젝트 조정과정에서는 계획된 과정과 실제 진행과정을 비교한다.



[그림 1] 프로젝트 관리 수명주기

프로젝트 협력: 프로젝트 협력과정은 지속적으로 프로젝트의 수행과 질을 확인한다. 이 과정은 프로젝트의 품질을 관리하는 부분이라고 볼 수 있다.

프로젝트 마케팅: 프로젝트 마케팅은 프로젝트를 수행하는 전체 팀에 의해 수행되어야 한다. 프로젝트 팀원들의 사기를 장려하고, 프로젝트와 일체감을 갖도록 프로젝트 마케팅을 수행하여야 한다.

프로젝트 종료: 프로젝트 종료과정은 남은 업무의 계획, 프로젝트 평가, 프로젝트 팀의 해산 등을 위해 실시한다. 프로젝트를 진행하면서 얻은 경험들은 문서화 하는 작업도 종료과정에 포함된다.

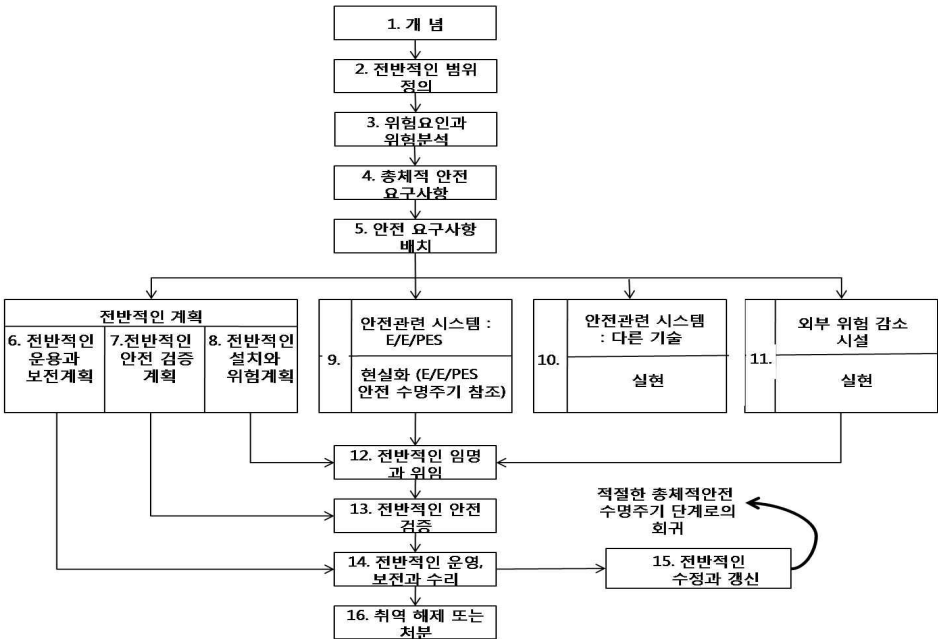
2.2 IEC 61508

IEC 61508은 안전 시스템에 대한 요구사항 명세·설계·개발·설치·운영·유지보수 표준이다. IEC 61508은 안전 기능을 수행하기 위해 사용되는 전기적/전자적/프로그램 할 수 있는 전자부품의 안전 관련 시스템(E/E/PE)의 모든 안전 수명 활동의 일반적 접근에 맞추어 구성되어 있다. 이 통합적 접근은 합리적이고 일치된 기술적 정책이 모든

전자를 기초로 한 안전 관련시스템을 발전시키기 위해 만들어졌다. 중요한 목적은 적용 부문의 표준을 발달시키는데 용이하게 하는 것이다[8].

IEC 61508은 전기적/전자적/프로그램 가능한 전자부품의 가능한 전자의 안전성 관련 시스템의 기능적 안전이라는 일반 표제 아래에 다음과 같이 구성된다[2].

Part 1 : 일반적 요구사항
Part 2 : 전기적/전자적/프로그램 가능한 전자 안전 관련 시스템에 대한 요구사항
Part 3 : 소프트웨어 요구사항
Part 4 : 정의와 약어
Part 5 : 안전도 수준의 결정 방법의 예
Part 6 : IEC 61508-2와 IEC 61508-3의 적용 가이드라인
Part 7 : 기술과 측정의 개관

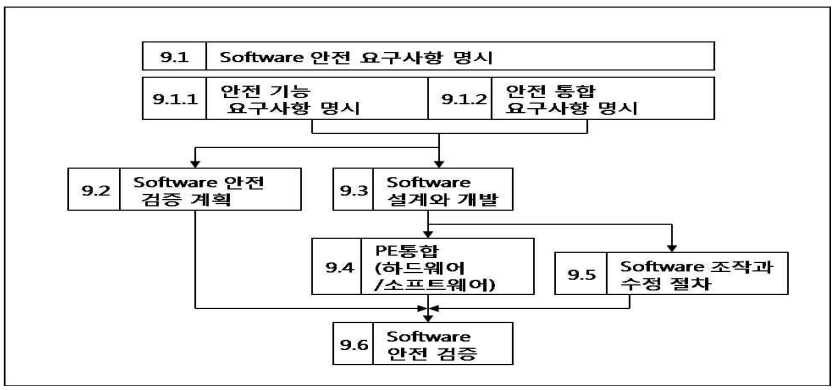


[그림 2] IEC 61508의 전반적인 안전수명주기

이 국제 표준은 E/E/PE가 안전기능을 수행하기 위해 사용될 때, 모든 단계, 즉 전반적으로 E/E/PE와 소프트웨어 안전수명 단계(예를 들어 초기개념, 디자인, 수행, 운용, 보전, 폐기에 이르는)를 고려한다. 이러한 각각의 단계에 기존의 분석기법 방법 들을 비슷한 부분을 통합시키면 빠르게 발전하고 있는 다양한 기술에 적용할 수 있으며, 구성은 강건하고 포괄적으로 되며, 미래의 기술 개발 안전에 많은 도움을 줄 수 있다. 이것은 안전과 경제적 이익 모두를 가져올 수 있다[6].

[그림 3]소프트웨어 개발을 위한 안전성 수명주기는 IEC 61508-1의 조항과 일치하는 안전성 계획안에 의해 전문화 된다. 품질과 안전성에 관한 보증 절차를 통합한 모

델로 소프트웨어 안전수명주기 각각의 단계는 각각 단계의 전문화된 범위, 투입물, 산출물과 함께 기본적인 활동들로 나누어진다. 소프트웨어 안전수명주기의 어떤 단계도, 초기의 수명주기 상황에 관해 변화가 이뤄 졌을 때, 초기의 안전수명주기 상황과 그 다음의 상황은 계속 반복되어지는 특성을 갖고 있다[2].



[그림 3] 소프트웨어의 안전수명주기

이 국제 표준은 E/E/PE가 안전기능을 수행하기 위해 사용될 때, 모든 단계, 즉 전반적으로 E/E/PE와 소프트웨어 안전수명 단계(예를 들어 초기개념, 디자인, 수행, 운용, 보전, 폐기에 이르는)를 고려한다[6].

2.3 소프트웨어 개발 프로젝트 수명주기

정보시스템 소프트웨어를 구축하는 것은 여러 사람들이 서로 협력하여 공동으로 개발하는 것으로 연관성이 매우 복잡한 일이다. 따라서 소프트웨어 프로젝트를 관리하는 데는 소프트웨어 특성과 소프트웨어 프로젝트 관리의 특수성을 고려해야 한다. 소프트웨어 프로젝트 관리에 중요한 요소는 일정, 위험, 품질, 자원, 범위 등으로 정리할 수 있다[10].

IEC 61508에서는 소프트웨어의 안전수명주기를 통해 소프트웨어 개발 프로젝트에서 안전성을 확보할 것을 요구하고 있다[2]. 소프트웨어의 안전성을 평가하기 위해서는 소프트웨어가 개발되는 라이프사이클 동안에 각각의 개발단계에서 소프트웨어의 특성에 따라 가장 적절한 방법으로 결함이 분석되고, 처리되어야 한다.

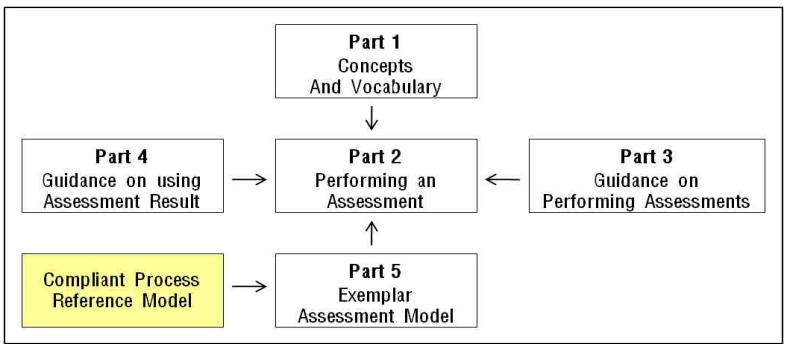
2.3.1 ISO/IEC 15504

ISO/IEC 15504는 소프트웨어 프로세스 전체를 심사하여 소프트웨어 개발 프로세스를 개선하고 개발자의 능력을 향상시킴으로써 개발과정에서의 위험을 통제하기 위한 목적으로 국제표준화 기구(ISO : International Organization for Standardization)에서 추진하는 소프트웨어 품질 표준화 심사 평가 모델이다. 이 모델은 정보 통신 분야의 소프트웨어 프로세서를 평가하고 개선함으로써 품질 및 생산성을 높이기 위한 목적의

표준이다. ISO/IEC 15504는 소프트웨어 프로세스와 능력레벨에 대한 참조 모델을 제시함과 함께 참조 모델을 토대로 실제로 소프트웨어 프로세스를 평가하기 위한 지침과 심사원의 자격 등에 관한 사항을 명시하고 있다[3].

[그림 5]소프트웨어 개발 수명주기(ISO/IEC 15504)는 계획의 확립, 실행과 테스트 3가지로 나누어진 소프트웨어 개발 수명주기를 나타낸다. 정보 통신 분야의 소프트웨어 프로세서를 평가하고 개선함으로써 품질 및 생산성을 높이기 위한 목적의 표준이다. 이 수명주기는 소프트웨어 프로세스와 능력레벨에 대한 참조 모델을 제시함과 함께 참조 모델을 토대로 실제로 소프트웨어 프로세스를 평가하기 위한 지침과 심사원의 자격 등에 관한 사항을 명시하고 있다[3].

계획의 확립단계는 요구사항 개발과 소프트웨어 디자인 과정을 포함한다[5].
계획 확립단계가 끝나면 모든 소프트웨어 요구사항과 디자인 문서는 실행단계로 넘어가게 된다. 실행단계는 소프트웨어 구조와 소프트웨어 통합과정으로 구성된다. 이 단계에서는 소프트웨어 개발에 관련된 모든 요구 조건들이 실행되고 통합되어 테스트될 때 실행단계가 완료된다. 테스트 단계는 계획 확립단계에서 발전되어진 부분이다. 테스트활동은 프로젝트 결과물이 처음 계획했던 것들을 만족하는 지 분석 및 평가하는 작업이다[1].



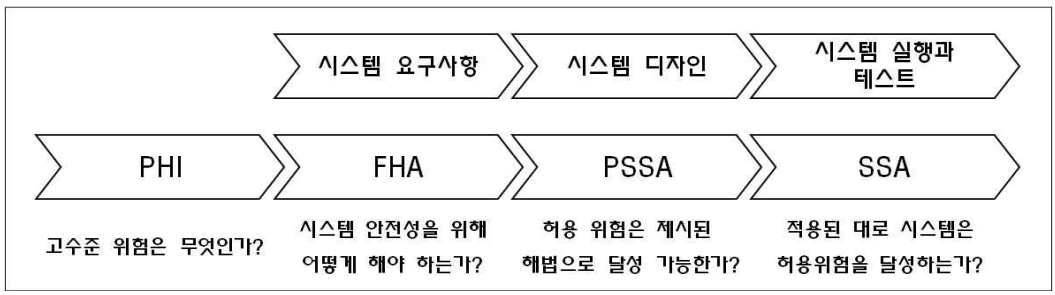
[그림 4] ISO/IEC 15504 의 기본 구조

[그림 5] 소프트웨어 개발 수명주기 (ISO/IEC 15504)

2.4 안전성 수명주기

안전성 수명주기에는 예비위험 확인(PHI : Preliminary Hazard Identification), 기능적 위험 평가(FHA : Functional Hazard Asessment), 예비시스템 안전성 평가(PSSA : Preliminary System Safety Assesment), 시스템 안전성 평가(SSA : System Safety Assesment) 등으로 구성되어 있다. 예비위험 확인(PHI)에서는 고수준의 위험들, 발생 가능한 위험들에 대해 조사 및 검토를 통해 예방하는 작업을 수행한다. 기능적 위험 평가(FHA)에서 수행되는 분석은 안전에 대한 계획을 세우고, 확인된 위험들에 대해 대비책을 세우는 역할을 한다. 예비시스템 안전성 평가(PSSA)는 시스템 디자인 부분에서 수행된다. 제시된 대비책으로 위험요소들을 제거할 수 있는지, 기능적 요구사항들을 만족하고 있는지 검증하는 작업을 수행한다. 만약 확인된 위험들이 기능적 요구사항들을 만족하지 못할 경우 만족할 수 있도록 시스템 디자인을 변경하고, 예비시스템 안전성 평가(PSSA)를 반복해야한다.

시스템 안전성 평가(SSA)는 모든 안전성 수명주기들이 제대로 수행되었는지 반복적으로 확인하는 작업을 수행한다[7].



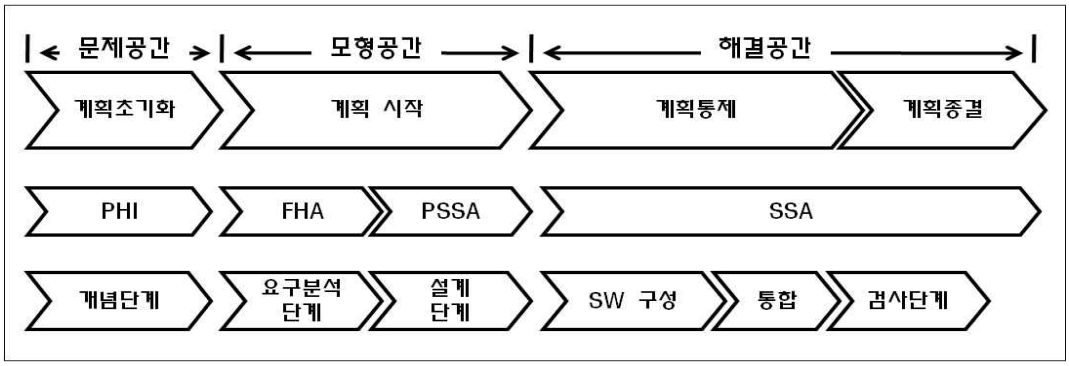
[그림 6] 안전성 수명주기

시스템의 안전성평가는 모든 안전 요구사항들을 수행하였는지 확인하고, 남아있는 위험이 허용될 수 있는 위험들인지 확인하는 역할을 한다.

3. 통합된 소프트웨어 개발 프로젝트 수명주기

프로젝트 관리 수명주기, 소프트웨어 개발 프로젝트 수명주기, 안전성 수명주기를 통합하여 프로젝트 구축과정에 있어 프로젝트 관리 수명주기에 상호관계가 맞도록 고려하여 다음과 같이 통합된 프로젝트 관리 수명주기를 도출해 낼 수 있다[1].

통합된 소프트웨어 개발 프로젝트 수명주기는 문제 공간, 모형 공간, 해결공간으로 크게 3분야로 나누어 볼 수 있다. 프로젝트의 계획단계는 '문제 공간'에 위치하고, 프로젝트 시작단계는 '모형 공간', 그리고 프로젝트 테스트 단계 및 종료부분은 '해결 공간'에 위치하게 된다[1].



[그림 7] 통합된 소프트웨어 개발 프로젝트 수명주기

3.1 문제 공간

문제공간은 특정한 문제, 예비 연구, 고객 요청, 아이디어 등에 대한 시스템을 개발하기 위한 기초 단계이다. 시스템의 목적과 목표는 분명히 정의되어야 하고, 개별적인 시스템의 거시적 개념뿐만 아니라 시스템의 범위의 정의에 대해서도 분명히 정의 되어야 한다. 프로젝트의 목표가 결정되면 예비위험확인(PHI)을 수행하여 위험요인에 대한 파악도 동시에 이루어 져야 한다. 이러한 내용을 바탕으로 개략적인 계획 설정이 가능하게 되며, 이 계획을 바탕으로 프로젝트의 가능성을 판단 후 실현가능성이 있다면 다음 단계인 계획시작(모형 공간)으로 넘어가게 된다.

3.2 모형 공간

프로젝트의 목표가 설정되면 본격적인 프로젝트를 진행하기 위해 계획을 세운다. 프로젝트 업무의 범위를 정하고 수명주기를 정의한다. 수명주기 부분에서는 프로젝트의 일정 및 프로젝트의 속성들에 대한 추정치를 정하고 자원분배를 정의한다. 모형공간에서의 안전성 수명주기는 기능적 위험 평가(FHA)와 예비 시스템 안전성 평가(PSSA)를 통해 이루어진다.

3.3 해결 공간

프로젝트 계획이 끝나고 프로젝트가 진행되며 시스템을 실제적으로 만드는 공간이다. 프로젝트의 통제 및 종료가 진행되는 곳으로 안전성 평가 및 검사활동은 검사 관리계획안에 명시되어 있어야 한다. 검사 관리계획은 프로젝트 관리 계획에서 작성한 것을 바탕으로 진행한다. 시스템의 안전성을 위해 안전성 평가 및 검사는 필수적이며, 필요에 따라 반복적으로 검사를 하기도 한다. 해결공간에서 쓰는 안전성 수명기로는 시스템 안전성 평가(SSA)가 있다. 시스템 안전성 평가는 프로젝트가 종료된 이후에도 계속 진행한다.

4. 결 론

소프트웨어 분야의 급성장으로 인해 고객들의 요구는 급변하고 다양화됨에 따라 소프트웨어 개발 업체들의 경쟁이 심화되고 있다. 그리하여 소프트웨어 개발 업체들은 제품의 품질, 제품출시 시간, 고객 만족 등 개발 프로젝트를 진행하며 고려해야 할 부분들도 증가되었다. 프로젝트 관리자는 전체적인 프로젝트 수명주기를 정의하고 적절하게 각각의 단계에 알맞은 시스템을 배치하여야 한다. 이번 연구에서는 전체적인 프로젝트 모형을 정의하고 그 안에 소프트웨어 개발 프로젝트 수명주기, 안전성 수명주기를 배치하여 보았다. 이 통합된 프로젝트로 비용절감 및 개발 시간 단축, 일의 효율성 및 효용성향상 등 경제적 이익을 가져올 수 있다고 생각한다.

하지만 '해결 공간'부분에 프로그램 코딩부터 검사까지 너무 많은 부분이 들어가 있기 때문에 '해결 공간' 부분을 좀 더 세분화 시킬 필요가 있다. '문제 공간 '및 '모형 공간'을 부분적으로 통합 후 '해결 공간'에 있는 안전성 평가 부분의 이동을 고려해 보아야 한다. 또한 검사 부분을 독립적으로 구성하는 방안도 고려해 보아야 할 것이다.

이 연구는 이론적으로 구상하였고, 실제 프로젝트 나 기업 시스템에 적용해 보지 않았기 때문에 실제 시험을 통해 수정 및 보완할 부분들을 고쳐나가야 할 부분이 있을 것이다. 다음 연구에서는 실제 프로젝트나 기업사례를 통해 실제 적용에 관한 연구를 해야 할 것이다.

5. 참 고 문 헌

[1] Hans Tschüurtz, Gabriele Schedl "An Integrated Project Management Life Cycle Supporting System Safety", Making System safer 2009

[2] IEC(1998), IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems

[3] IEC(2004), ISO / IEC 15504, Information technology-process assessment.

[4] 강창욱 외11, 프로젝트 관리학 pp27-86, 북파일, 2009

[5] 김광현, "소프트웨어 품질과 프로젝트 성과와의 관계에 관한 탐색 연구", 대한경영정보학회지 Vol. 25 pp275-296, 2008. 06

[6] 김종걸, 박지성 "RMS 기반으로 한 소프트웨어 품질의 안전성 평가 개선방안 연구", 2010 산업공학회 춘계학술대회 논문집 2010. 6

[7] 김종걸, 김창수, "리스크 분석 기법에 대한 조사연구" 안전경영과학회 2004년 추계학술대회논문집 2004. 10

[8] 김종걸, 김인희 "소프트웨어 분야의 리스크경영시스템 도입방안에 관한 연구" 안전경영과학회 2009년 추계학술대회논문집 2009. 10

[9] 김용경 ,김필중 "공공 소프트웨어 프로젝트의 관리 형태에 관한 탐색적 연구", 한국경영정보학회 Vol. 13, No. 4, 2006. 12

[10] 전순천, 홍사능 "소프트웨어 프로젝트 관리 영역간의 상호영향을 고려한 성숙도 모델", 2008 한국경영정보학회 춘계학술대회, pp850-858 2008. 4