

위험관리 프로세스와 평가의 새로운 접근

Risk Management and Assessment Methodology in System Design

조희근* 박영원**
Cho, Hee-Keun, Park, Young-Won

요약

위험관리는 최근 다양한 분야의 산업에 적용 가능한 매우 중요한 토픽이다. 본 연구에서는 기본적인 위험관리의 일반적인 방법론과 새로운 위험관리 평가의 방법론을 제시한다. 위험관리의 실제 적용에 있어서 재정, 보험산업 등의 분야에서는 이미 많은 현실적 방법론들이 개발되었다. 다양한 산업현장에서 발전된 많은 적용사례들이 있으나 여전히 위험관리는 어려운 부분이며, 두 가지 주요한 위험평가 방법론이 시스템 엔지니어링 분야에서 개선되고 발전되어왔다. 그중 하나는 본래의 위험평가의 기본개념을 바탕으로 하여 발전한 것이며 다른 하나는 소프트웨어 엔지니어링분야의 위험평가에서 비롯된 것이다. 규모가 크고 복잡한 시스템에서 효과적인 위험관리는 프로젝트의 최종 목적을 달성하는데 있어서 불확실성에 적절히 대처하여 총 개발비용을 줄이고 목표일정과 성능을 달성하는 효과를 가져온다.

ABSTRACT

Recently, risk management is a very important issue in many industrial applications. This paper describes a general structure for risk management and suggests a new risk assessment methodology. In risk management applications of financial or insurance industries, there are many methodologies developed for practical use. However, areas for improvement exist to facilitate the application of the methods. Two major risk assessment methodologies have been developed and widely applied in system engineering. One is in its original development application from aerospace and defense industry, and the other was developed in applied software engineering. In a large and complicated system development application, an effective risk management can reduce total development cost as well as uncertainty in achieving project goals of schedule and performance.

주요기술용어 : 위험관리(Risk Management), 아키텍처(Architecture), 위험도평가(Risk Assessment)

1. 서론

위험 관리는 복잡하고 거대한 시스템의 일부분으로

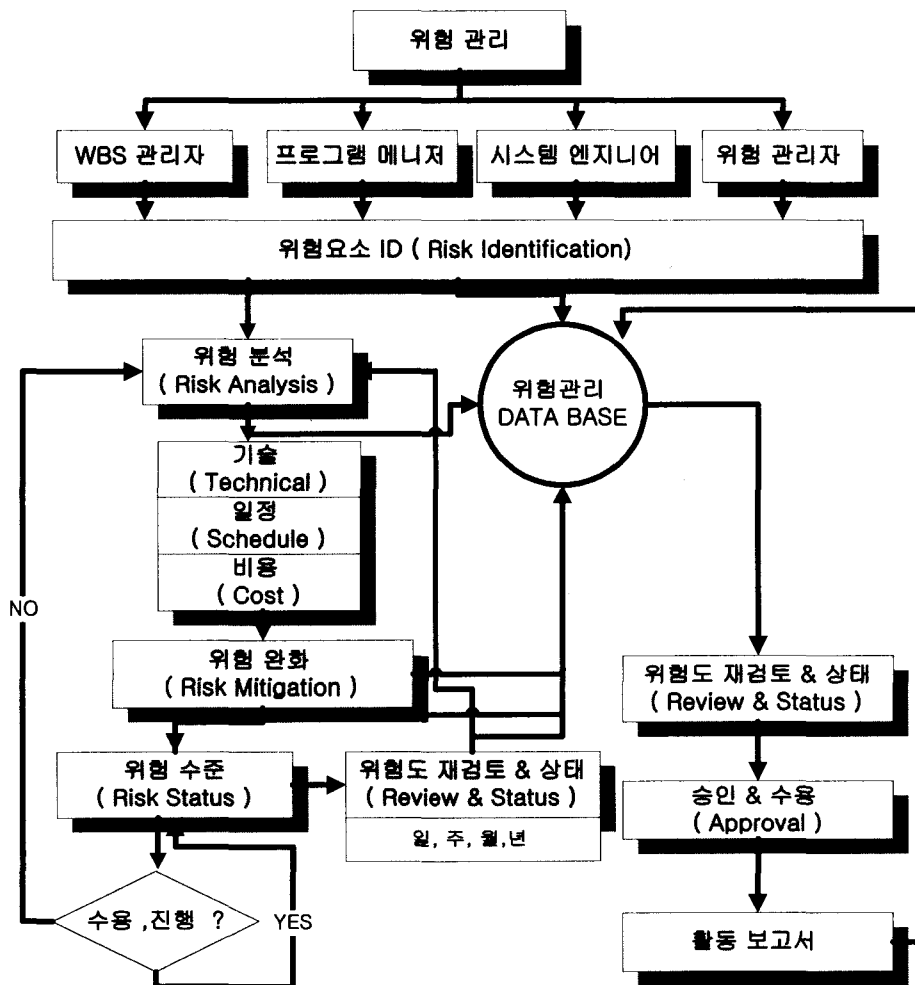
* 아주대학교 시스템공학과 박사과정

** 아주대학교 시스템공학과 교수

서 반드시 수행되어야 하는 중요한 하부과정이다. 시스템이 복잡하고 규모가 클수록 내부적인 불확실성이 증가하기 때문에 많은 위험요소를 포함하고 있다. 따라서 복잡하고 다양한 시스템 환경 하에서 시스템 엔지니어는 요구사항을 분석하고 최소의 위험성으로

고객의 필요성을 만족시키는 가장 적절한 방법론과 절차를 결정해야한다. 시장에서의 제품성능이나, 주어진 일정 혹은 예산 내에서 제품개발에 대한 불확실성에 기인하는 위험성은 그 분석과 평가방법에 많은 문제점을 가지고 있다. 수년간 기본적인 위험도 관리 절차와 방법론 그리고 위험도를 분석하고 평가하는 척도가 설정되었고 꾸준히 개발 진보 되어왔다. 시스템이 복잡 다양해짐으로써 많은 변수들이 생겨났고 요구사항이 점점 다양해짐에 따라 기존의 시스템 위험

분석 및 평가 방법에 이의를 제기하기 시작했고 이러한 요구에 따라 새로운 개념의 위험도 평가 방법이 제시되었다. 이러한 방법론들은 서로 상충되기도 하고 개개인에 따라서 서로 다른 견해를 가지고 시스템 위험도를 평가하고 있다. 본 논문에서는 위험도를 평가함에 있어서 기존의 위험도 평가방법의 문제점을 인식하고 보다 효율적이고 정확한 위험도 평가 방법을 제시한다. 또한 이러한 기존방법과 새로운 방법에 의한 위험도 평가의 차이점을 알아본다.



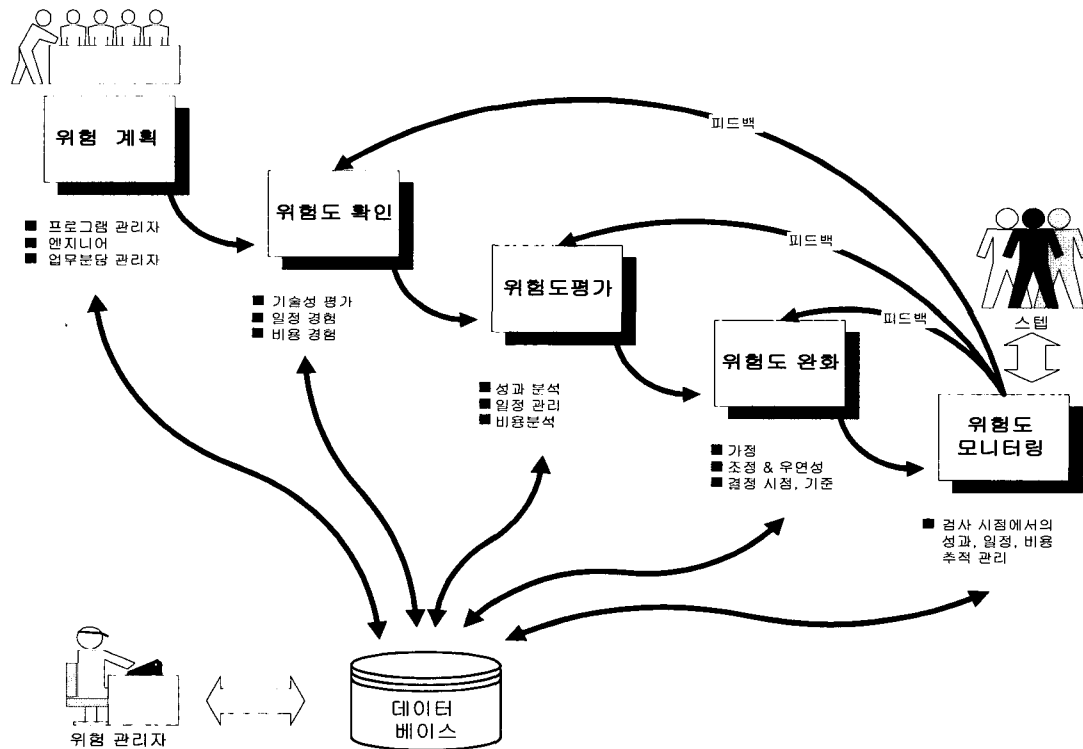
(그림 1) 위험관리 아키텍처

2. 일반적인 위험관리 방법론

위험관리는 일어날 확률이 있는 문제점들이 발생하기 전에 관리하여 그 손실을 막기 위한 전략이다. 시스템의 위험 관리를 어떻게 수행 할 것인가 알기 위해서는 위험의 정의 및 위험성을 조정할 수 있는 절차에 대해서 알아야 한다. 위험도는 아직 발생하지는 않았지만 발생가능성이 있거나 미래에 일어나지 않을 수도 있는 특성을 가지는 반면 문제점은 현재 일어나고 있거나 벌써 시스템의 성능에 나쁜 영향을 미치는 인자들이다. 위험도의 특징으로는 불확실성과 손실이다. 이러한 불확실성을 특징으로 하기 때문에 Monte Carlo 분석에 의한 시뮬레이션 방법도 많이 행해지고 있다. 넓은 의미에서 위험관리란 시스템의 성능 및 비용에 나쁜 영향을 미칠 가능성 있는 인자들과 이러한 인자들의 발생

가능성에 관한 지속적인 분석 및 평가의 절차이며 위험성을 조정하기 위한 대체 전략의 결정, 수행 가능한 전략방법의 선택이라 할 수 있다. (Kirkpatrick,1993)

위험관리의 아키텍처를 정의하면 그림 1과 같으며 최상위 단계에서 WBS(Work Breakdown Structure) 관리자, 프로그램 매니저, 시스템엔지니어, 위험관리자 등이 모여서 위험요소를 도출한다. 위험관리 DATABASE는 상부수준 혹은 하부수준 등의 구별이 있는 것이 아니고 위험관리의 모든 프로세서에서 동등하게 접근 가능한 수준의 프로세스이다. 위의 위험관리 아키텍처에서 위험을 분석하는 서브 시스템에서는 비용, 일정, 성능의 관점에서 위험요소를 분석하고 평가한다. 위험관리 아키텍처의 핵심부를 이루는 것은 위험도 평가의 프로세스이며 이 위험도 평가의 결과에 따라 시스템의 개발이 중단될 수도 있다. 위험도 평가는



(그림 2) 위험관리(Risk Management) 기능흐름

시스템 전 수명주기동안 계속 반복되는 프로세스이며 각각의 반복과정을 통하여 위험 분석한 결과가 시스템 개발에 반영되도록 하여야한다. 이 반복적 피드백 과정을 통하여 시스템의 위험도를 줄일 수 있다. 위험관리에서의 각 단계에서의 기능적 흐름은 그림 2와 같다.

3. 전통적 방법에 의한 위험도 평가(Traditional Risk Assessment)

위험 평가 기능에서는 위험도가 높은 시스템의 객체를 입력데이터로 하여 그 위험의 심각성을 결정한다. 이 수준의 분석에서 시스템 엔지니어는 하드웨어, 소프트웨어, 복잡성, 기술성 등의 성숙도 등 많은 인자들을 고려하는 정량적인 모델에 적용한다. 위의 모든 인자들로부터 두 가지 중요한 수치적 변수 인자를 도출 할 수 있다.

- P_f : 실패 확률
- C_f : 실패 결과의 심각성

위의 두 변수들은 위험지수(Risk Factor)를 결정하는데 사용된다. 위험지수를 결정하는데 있어서 많은 정보가 필요하다. 전통적으로 시스템 위험도를 평가하는데 사용되는 위험지수(Risk Factor)를 계산하는 방법은 다음과 같다.

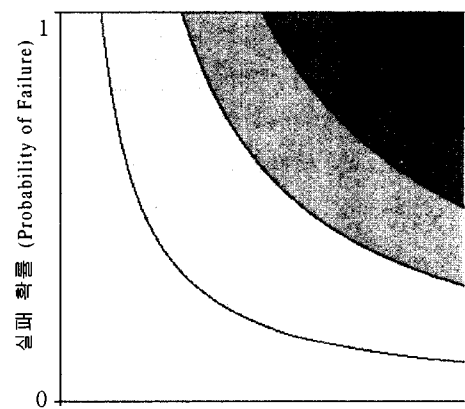
- 전통적 위험지수(Risk Factor)

$$RF=(P_f)(C_f)$$

[표 1] P_f 와 C_f 특성

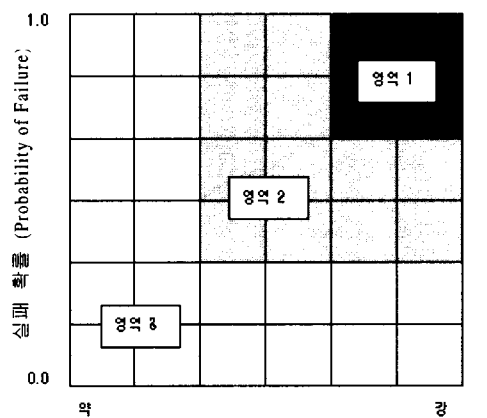
P_f	C_f
실패 가능성 (Probability of Failure)	실패 결과성 (Consequence of Failure)
원인 (Due To)	결과 (Result In)
H/W 성숙도 (기술성) S/W 성숙도 (복잡성)	성능 Impact 일정 Impact 비용 Impact

표 1은 위험지수 (Risk Factor) 계산에 있어서 P_f 와 C_f 지수간의 특성을 비교한 것이다. P_f 는 원인 의존적인 지수 즉, 하드웨어, 소프트웨어, 시스템 복잡성 등의 원인에 의해서 측정되어지는 지수이다. C_f 는 결과 과정 지향적인 지수이며 이는 시스템의 성능, 비용, 일정에 영향을 미쳐서 시스템에 나쁜 영향을 주는 Function 들이다. 위험지수(RF)는 일반적으로 0~1 사이의 값을 가지며 고, 중고, 중, 중하, 하 등으로 분류된다. 정량적인 위험분석은 매우 복잡한 토픽이며 많



약 실패 결과성 (Consequence of Failure) 강

(그림 3) 등위험도 곡선



약 실패 결과성 (Consequence of Failure) 강

(그림 4) 위험도 매트릭스

은 문제점들이 제기되고 있다. 그림 3는 위험지수 (Risk Factor)를 등고선의 형태로 이차원으로 나타낸 것이고 그림 4는 실제 위험도 평가에 사용하기 위하여 이를 매트릭스의 형태로 나타낸 것이다. 여기서 명암은 위험도의 차이를 나타낸다. 위험도가 높을수록 등-위험도 곡선에서 어둡게 나타난다.

실패 가능성을 나타내는 지수 P_i 를 계산하는데 고려 되는 속성들로는 개발, 요구, 기술성, 설계, 통합, 검증, 제조, 폐기, 운영 등 시스템 전 수명주기에 걸친 시스템 결과물 뿐만이 아니라 보조형 하부시스템도 포함된다. 실패 결과 지수 C_i 는 실패에 기인하는 시스템의 성능저하와 관계된 기술적 인자 (Technical Factor), 비용상승에 의한 비용 인자(Cost Factor), 일정 지연에 의한 일정 인자(Schedule Factor)등으로부터 계산 할 수 있다.

시스템의 특성에 따라서 각각 인자들의 중요성이 서로 다를 수 있기 때문에 서로 다른 가중치를 곱하여 결과 지수를 구한다. 이와 같은 전통적인 방법에 의한 위험도 평가 방법에서는 실패에 따른 위험도는 아주 높지만 일어날 확률이 적은 위험 항목들에 대해서는 위험 요소로 인정하지 않고 있다. 또한 실패에 따른 심각성은 적지만 일어날 확률이 높은 위험 항목에 대해서도 위험 요소로 인정하지 않고 있다.

예를 들어 사람의 인명이나, 시스템 전반에 걸친 영향력이 있는 중요한 항목들에 대해서는 비록 일어날 확률은 적지만 위험 항목으로 분류하여 적절한 위험 관리가 행해져야 한다. 시스템에 따라서는 실패의 결과성은 비록 적지만 자주 발생하면 위험도가 누적 되는 경우가 발생하는 경우가 있다. 이러한 시스템인 경우 전통적인 방법에 의한 위험 지수(Risk Factor) 계산은 실제의 시스템 위험도와는 거리가 멀어지게 된다.

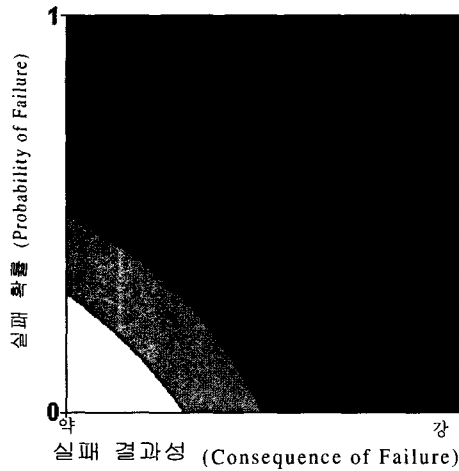
예를 들어 항공기가 벼락에 맞는 경우를 생각하게 되면 비록 일어날 확률은 적지만 만약 벼락에 맞을 경우 결과의 심각성이 너무 크기 때문에 반드시 위험요소로 분류되어야 할 것이다.

4. 소프트웨어 위험도 평가(Software Risk Assessment)

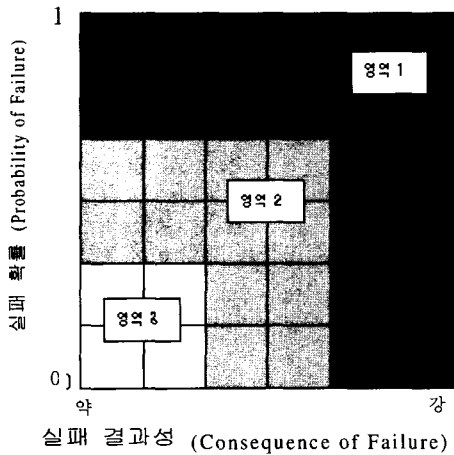
위험도 관리가 다양한 분야에서 적용되고, 널리 받아들여지고 있지만 이 과정에서 도출된 위험을 분석하고 평가하는 방법에는 많은 문제점이 있다. 위험도를 분석하고 평가하는 방법이나 기준은 개인적인 견해에 달려있다고 보는 주장도 있으나 전통적인 위험도 평가의 접근방법은 잘못될 가능성과 결과의 심각성에 초점이 있다. 이 방법에서 위험지수 $RF=(P_i)(C_i)$ 로 나타내며 C_i 를 0~1 사이의 구간으로 변환하여 나타내면 그림 4와 같다.

이러한 접근 방법은 기대값이 같으면 같은 수준의 위험을 가지고 있다는 것이다. 즉 300만원의 손실이 있을 확률이 3%인 것과 30만원의 손실이 있을 확률이 30%인 것이 서로 같다. 실제 위험요소의 위험도를 평가하여 표에 기입함으로써 전체 시스템의 위험수준을 파악한 뒤 후속 조치사항들을 결정해야 한다. 그림 5의 등-위험도 곡선의 관점을 그림 6의 매트릭스에 적용하면 우측 상단은 높은 위험도를 나타내고 중간 부분은 중간 위험도를, 하단은 낮은 위험도를 나타낸다. 여기서는 시스템의 크기에 따라 위험요소가 될 수도 있고 안될 수도 있다. 예를 들면 같은 100만원의 손실은 1천 만원 시스템에서는 위험요소가 될 수 있지만 10억의 시스템에서는 위험요소가 되지 않을 수 있다.

이러한 전통적인 위험도 평가 방법에서는 치명적 결과를 가져올 수 있는 위험요소라 할지라도 발생 가능성이 충분히 적은 경우와 실패의 심각성이 가벼운 것이면서 발생빈도가 높은 위험요소는 문제가 되지 않는다. 이러한 전통적 방법에 반해 Charette는 1990년 위험관리를 소프트웨어에 적용하였다. 그는 위험평가의 이론적인 기초를 제공하였으며 시스템 개발에 있어서 소프트웨어는 위험 관리 기준을 만족하는데 많은 문제점이 있다는 것을 밝혔다. 그는 등위험도 곡선



(그림 5) Charette의 등위험도 곡선



(그림 6) Charette의 위험도 매트릭스

을 위험지수에 기초하여 만들어 냈다. 실패 가능성은 성공의 결핍이라는 개념에서 새로운 위험지수(Risk Factor)를 만들어 냈다.

$$RF = 1 - (1-P_f)(1-C_f) \\ = P_f + C_f + (P_f)(C_f)$$

위의 위험지수를 가지고 등-위험도 곡선을 그려보면 그림 5와 같으며 그림 6은 이를 매트릭스의 형태로 나

타낸 것이다. Charette 는 위험지수(Risk Factor)를 실패는 성공의 결핍이라는 관점에서 도출하였기 때문에 많은 문제점을 가지고 있다. 실패의 심각성이 크면 그 일어날 확률에는 무관하게 똑같이 위험도가 높은 요소로 평가되며 또한 실패할 확률이 큰 요소에 대해서는 실패결과에 심각성에 상관없이 똑 같이 위험도가 높은 요소로 평가된다는 점이다. 예를 들면 10만원의 손실이 일어날 확률이 90%인 것과 1000만원의 손실이 일어날 확률이 90%인 것은 Charette의 위험도 매트릭스에서는 똑같은 위험도를 가지게 된다. 이상에서와 같이 위험도를 평가하는 기준으로 위험지수를 구하게 되는데 시스템의 특성과 주변 환경에 따라서 보다 더 적절하고 정확하게 위험지수(Risk Factor)를 구해야 할 필요성을 느끼게 된다. 만약 시스템이 거대하고 복잡하다면 위험도 매트릭스에서 위험영역을 어느 부분으로 정하느냐에 따라서 위험관리 대상이 되는 위험 항목들의 수가 틀려지게 된다. 뿐만 아니라 이러한 항목들을 관리하고 조정하기 위한 비용, 시간, 인력, 일정 등의 손실을 초래하게 된다.

5. 위험도 평가(Risk Assessment)의 새로운 접근 방법

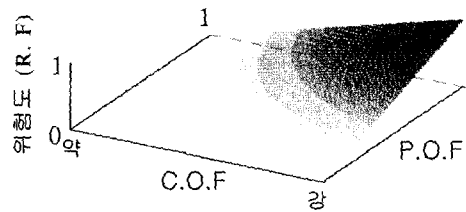
전통적인 방법과 Charette 의 3차원 등-위험도 곡선은 그림 7, 그림 8과 같다. 두 그림을 서로 비교해 보면 양쪽 모서리 부분에서 차이가 남을 알 수 있다. 뿐만 아니라 전반적인 위험 공간이 서로 다름을 알 수 있다. 가운데 지점에서는 Charette의 위험도가 전통적인 방법에 비해서 3배가 되며 Charette의 방법에서는 고전적 방법보다 훨씬 많은 위험영역을 가지고 있다. 고전적 방법이 위험도를 훨씬 적게 평가했는가? 차이점을 이해하기 위해서 컴퓨터를 개발하는 경우를 예로 들어보자. 새로운 접근(Charette)에서 보면 컴퓨터가

벼락에 맞거나 휴즈가 끊어지는 경우는 모두 아주 높은 위험요소로 간주된다. 벼락에 맞는 경우는 가능성은 적지만 그 결과성은 매우 심각하기 때문에 높은 위험 요소이고, 휴즈가 끊어지는 경우는 결과의 심각성은 적지만 가능성은 매우 크기 때문에 위험도가 높은 요소가 된다. 반면 하드디스크가 고장나는 경우는 발생 가능성도 비교적 적고 복원 가능한 요소이기 때문에 저 위험도 요소가 된다. 따라서 컴퓨터를 개발하는 엔지니어는 하드디스크 고장보다는 컴퓨터가 벼락을 맞거나 휴즈가 나가는 문제점을 해결하기 위해 많은 시간과 노력을 낭비 할 것이다. 이러한 개념을 가지고 있는 엔지니어는 발생 가능성이 희박한 극단적인 일이나, 발생 가능성은 높지만 아주 사소한 일에 집착하여 아무 일도 못할 것이다. 똑같은 사항이라 할지라도 시스템의 특성에 따라서 위험도가 틀려지게 된다. 자가용 엔진과 여객기 엔진이 운행도중 멈추는 경우를 생각해 보면 비록 거의 동일한 사건이지만 여객기의 엔진이 멈추는 경우가 훨씬 더 위험도가 높을 것이다. 따라서 우리는 시스템에 따른 위험도의 재평가를 하지 않으면 안 된다. 이러한 위험도의 견해는 다음과 같은 위험도 평가의 관점이 서로 다른데서 그 원인을 찾을 수 있다. 전통적인(Traditional Method)방법에서는 RF(Risk Factor)를 단지 두 실패지수에 의해서 결정된다고 보는 입장이다. 따라서 위험은 실패의 결과라는 결론을 얻을 수 있다. 반면 Charette 방법에서는 실패는 성공의 결핍이라고 보는 입장이다. 이러한 입장에서 위험도를 평가하면 전통적인 방법에 비해서 더 많은 영역에 걸쳐서 위험도가 높게 나타난다. 기대값이 같으면 위험도를 같게 평가한다는 의미이다. 이러한 두 방법은 위험도 평가에 있어서 서로 다른 견해를 가지고 있지만 어느 방법이 옳고 그른가를 단정 지을 수는 없다. 시스템의 특성에 따라서 두 가지 방법 중 어느 한 방법이 그 시스템의 위험도를 잘 나타낼 수 있기 때문이다. 그러면 시스템 위험도 평가에 있어서

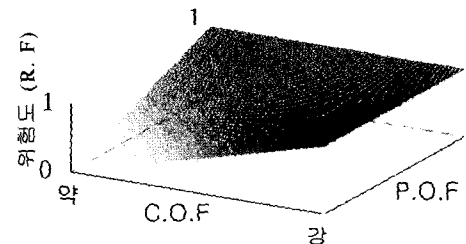
새로운 접근 방법은 무엇인가? 이러한 문제점에 대한 접근방법은 새로운 위험도 평가 방법을 통하여 할 수 있다. 기존의 문제점을 보완할 수 있는 새로운 위험도 평가 방법론을 제시한다.

(표 2) 전통적 방법과 Charett 방법

전통적인 방법	Charett 방법
<ul style="list-style-type: none"> RF(RiskFactor)는 단지 두 실패 지수에 의해서 나타난다. 전통적인 개념에서는 위험을 단지 실패의 결과라고 본다 	<ul style="list-style-type: none"> 실패확률은 성공의 결핍이다. 소프트웨어 엔지니어링에서 유래되었다. 위험영역이 다소 많아짐.
$RF = (P_f)(C_f)$	$RF = 1 - (1 - P_f)(1 - C_f)$ $= P_f + C_f + (P_f)(C_f)$



(그림 7) 3차원 전통적 방법의 위험도 곡선



(그림 8) 3차원 Charette의 등위험도 곡선

C.O.F : 실패 결과의 심각성(Consequence Of Failure)

P.O.F : 실패 확률(Probability Of Failure)

고전적 방법과 Charette의 방법 모두 시스템의 특성에 따라서 위험도를 평가함에 있어서 능동적으로 대처하지 못하고 있다. 그림 6과 그림 7에서 위험도 차이가 큰 양쪽 모서리 부분을 시스템에 특성에 따라 능동적으로 보정해줄 수 있는 새로운 위험지수(Risk Factor)가 필요하다.

따라서 이러한 요구에 따른 수정 위험지수(Correction Risk Factor)를 구해보면

$$RF = P_f + C_f + (P_f)(C_f) - \alpha \times C_f(1 - P_f) - \beta \times P_f(1 - C_f) = \alpha P_f + \beta C_f - (P_f)(C_f)(1 + \alpha + \beta)$$

α : 위험도의 결과성 의존 수정계수(0~1)

β : 위험도의 발생가능성 의존 수정계수(0~1)

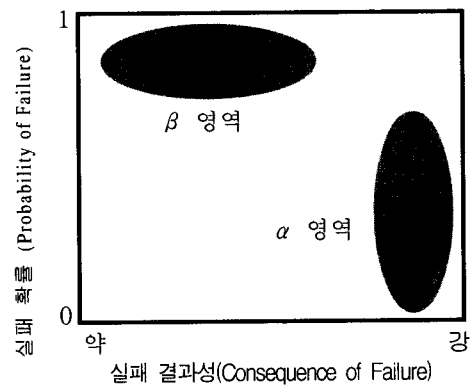
위의 식에서 α 의 값이 1에 접근할수록 위험도가 실패 결과성에 의존적이고 β 의 값이 1에 접근할수록 위험도가 발생가능성에 의존적이다. 즉 α, β 가 모두 1인 경우는 Charette의 위험도평가 방법이다.

[표 3] 위험도 평가의 문제점

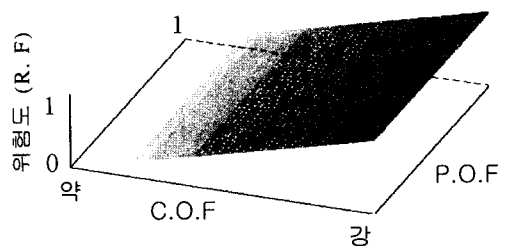
전통적인 방법	Charette 방법
1. 위험을 단순한 실패 결과로써 추정.	1. 위험도는 실패 기대값에 의존한다.
2. 실패 확률은 적지만 위험이 큰 요소에 대한 과소평가	2. 실패기대값이 같으면 위험도가 같다.
3. 위험의 누적성에 대한 무시.	3. 실패 확률이 아주 적은 요소에 대한 위험의 과대 평가.
4. 전체 시스템에 대한 위험도 평가 기준으로 부적합.	4. 수용 가능한 위험 인자에 대한 위험 과대평가.
5. 상대적으로 낮게 평가된 위험영역.	5. 위험영역의 과대로 인한 위험관리 차원의 비용, 시간, 인력의 낭비

전통적인 위험도 평가 방법과 Charette의 위험도 평가 방법에서 차이를 보이고 있는 오른쪽하단 α 영역과 왼쪽 상단 β 영역을 시스템에 따라서 유연하게 보정해 줌으로써 보다 위험지수가 실제 시스템의 위험도에 더 접근할 수 있다. 또한 위험도 영역을 결정함에 있어서 α, β 영역을 독립적으로 조정할 수 있으므로 기존의 방법보다 더 효율적이고 정확한 위험도 평가가 가능하다. 그림 10과 그림 11은 각각 $\alpha=1, \beta=0$ 인 경우와 $\alpha=0, \beta=1$ 인 경우의 위험도 분포를 나타낸다.

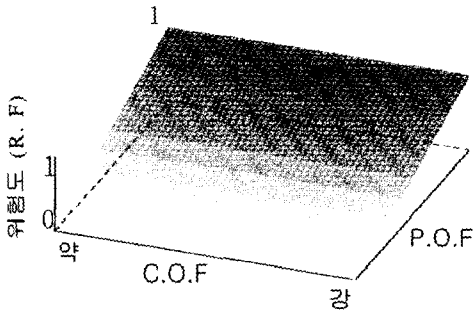
α 값이 큰 경우의 위험인자의 유형은 실패결과에 영향력이 큰 경우이며 인명이나 대형재난을 일으킬 수 있는 위험 요소 등이 포함 될 수 있다. β 값이 큰 경우는 시스템 내부에 위험도가 누적되는 경우나 발생빈도가 시스템에 악영향을 주는 경우 등이다.



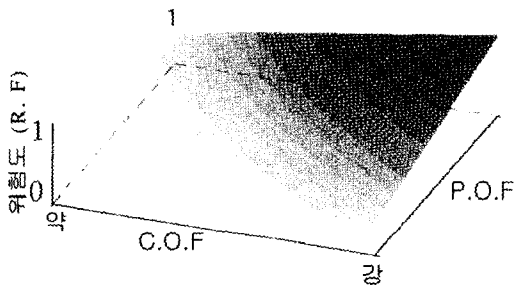
(그림 9) 위험도 변수영역



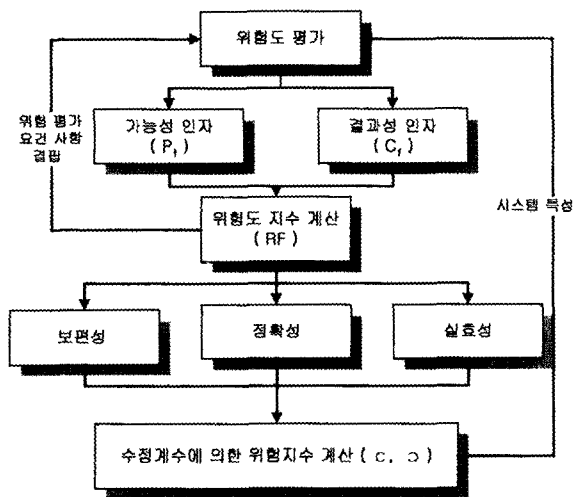
(그림 10) 위험도 분포 ($\alpha=1, \beta=0$)



(그림 11) 위험도 분포 ($\alpha=0, \beta=1$)



(그림 12) 위험도 분포 ($\alpha=0.6, \beta=0.6$)



(그림13) 위험도 평가 Process

그림 13은 위험도를 평가하는 프로세스를 나타낸 것이다. 위험지수를 계산하는데 있어서 보편성과 정확성,

실효성 등의 요건을 만족시키기 위하여 새로운 위험지수의 도출이 필요하게 되었다. 새로운 위험지수를 계산함에 있어서 수정계수 α 와 β 를 정의함으로써 시스템의 특성을 고려하였다. 시스템 위험도를 평가함에 있어서 α 와 β 값의 결정변수 인자가 매우 중요한 역할을 한다. 시스템의 위험을 유도하는 인자들을 도출하여 α, β 를 결정한다. 여러 시스템 인자들을 동시에 고려함으로써 시스템 전체의 특성을 잘 나타내게 할 수 있다. 시스템의 일부분에서의 어떤 부분적인 특성이 아니라 전반적이고 통합적인 관점에서 평가하지는 것이다.

6. 수정계수 값 계산

새로운 방법에 의한 위험도 평가에서는 α 와 β 값을 결정하는 것이 주요 관점이 된다. 따라서 이 두 변수가 어떤 특성을 가지고 있는지 확인한 다음, 분명하고 정량적인 방법에 의하여 계산하여야 한다. α, β 의 특성을 요약하면 다음과 같다.

[표 4] α, β 특성

α 특성	
1.	α 가 1에 접근할수록 위험도가 실패 결과성에 의존적임
2.	위험도 Matrix의 α 문제 영역을 보정할 수 있다.
3.	위험도 Matrix의 α 영역에서 $\alpha=1$ 인 경우는 Charette 위험도와 같고 $\alpha=0$ 인 경우는 전통적인 방법의 위험도와 같다.
4.	실패결과에의 영향력이 큰 경우 의미를 가진다.
β 특성	
1.	β 가 1에 접근할수록 위험도가 발생가능성 (확률)에 의존적임
2.	위험도 Matrix의 β 문제 영역을 보정할 수 있다.
3.	위험도 Matrix의 β 영역에서 $\beta=1$ 인 경우는 Charette 위험도와 같고 $\beta=0$ 인 경우는 전통적 방법의 위험도와 같다.
4.	시스템 내부에 위험도가 누적되는 경우에 유효.

α 값은 결과성 의존인자이므로 아래의 각각의 항목에 대해서 위험도 의존지수를 결정하여 계산 할 수 있다. α 는 다음의 3가지 항목에 의해서 결정된다. 기술성(Technical), 일정(Schedule), 비용(Cost)의 각각의 상대적 위험 비율 지수를 R_i 라고 정의하면 R_i 는 0~1 사이의 값을 갖으며 전체 시스템 위험도에서 각 항목의 기여도이다.(예: 기술성 0.5, 일정 0.3, 비용 0.2)

$$\alpha = \frac{\sum_i W_i R_i}{3}$$

$\sum W_i = 1$: 가중치

R_i : 상대적인 위험 비율 지수

β 값은 발생가능성 의존인자이므로 아래의 다음의 항목에 대해서 상대적 위험 비율 지수를 결정하여 계산한다. β 값의 결정인자 항목으로써는 요건사항(Requirement), 설계(Design), 통합(Integration), 제조(Manufacturing), 검증(Verification), 작동(Operation), 프로그램 보조가 있다.

$$\beta = \frac{\sum_i W_i R_i}{7}$$

$\sum W_i = 1$: 가중치

R_i : 상대적인 위험 비율 지수

7. 비교 분석

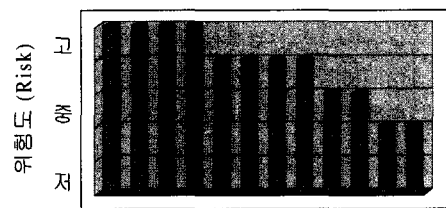
그림 14는 US Air Force Risk Management Guide에 있는 한 예를 나타낸 것이다. 이 예제 위험도 매트릭스를 Charette 방법과 전통적 방법에 의하여 위험도를 평가한 결과를 그림 15, 그림 16에 나타내었다. 그림 17는 수정 계수 $\alpha=1$ 인 경우의 위험도 분포를 나타낸

다. Charette에 의한 방법에서는 전체적으로 위험도가 높은 것을 알 수 있다.

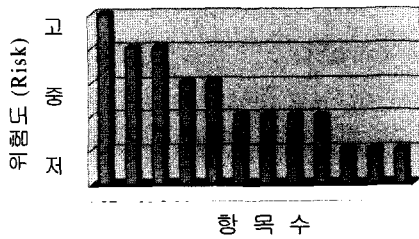
그림 14의 위험도 매트릭스에서 시험프로그램, 산업기기 같은 항목들이 전통적인 방법에서는 수용 가능한 위험요소지만 Charette의 방법에서는 높은 위험요소로 평가되고 있다. 위험도를 결정할 때 이러한 결정을 보완하기 위하여 적당한 수정 계수를 사용하여 위험도 매트릭스 상에서의 위험영역을 결정할 필요가 있다. α 와 β 의 값을 조정함으로써 시스템의 특성에 맞는 위험도 영역을 보다 효과적이고 정확하게 결정할 수 있다. 뿐만 아니라 시스템 전 수명주기에서 초기 개념설계 단계에서 부터 시스템 수준의 엔지니어는 전통적인 방법을 사용하고 소프트웨어 그룹은 Charette의 방법을 사용하기 때문에 평가 기준의 차이에 의해 두 그룹의 상호간에 새로운 위험요소를 발생시킬 수 있다.

1 위험요구 사항추진	시험 프로그램			
		지원장비	안내 시스템	
		훈련장비	어댑터	화재 조정장치
	국부활동	시험장비	안전장치	
		탄두		산업기기
0	약 실패 결과성(Consequence of Failure) 강			

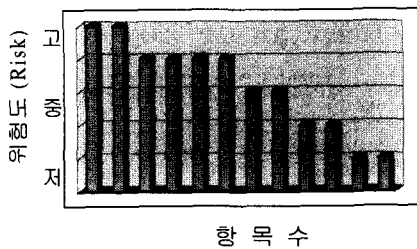
(그림 14) Sample 위험도 매트릭스



(그림 15) Charette 방법에 의한 위험도 분포



(그림 16) 전통적 방법에 의한 위험도분포



(그림 17) $\alpha=1, \beta=0$ 일 때 위험도분포

8. 위험도 평가시 고려사항

확산효과. 시스템 위험도 평가시 또 다른 고려사항으로는 한 위험인자가 다른 위험요소를 발생시키는 범위의 확산성과 시간이 지날수록 위험의 심각성이 심화되는 내용의 확산성이 있다. 예를 들면 조그만 부품하나가 시간이 지나면서 시스템 전체에 영향을 미칠 정도로 위험 요소가 될 수 있는 경우이다. 물론 과장된 말이지만 실제 시스템 설계에서도 이런 일은 얼마든지 일어날 수 있다.

소멸효과. 초기 개발단계에서의 많은 위험인자들은 서로 상호작용을 하면서 연관 관계를 가진다. 이 관계 속에서 어떤 위험인자들은 시간이 지나면서 다른 위험요소의 소멸과 동시에 없어지는 경우가 있다. 시간 의존적인 위험성이 대표적 경우이며 예를 들면 기술수준이 발전함으로써 위험요소가 안 되는 경우가 있다.

위험도 범위. 위험도를 결정함에 있어서 발생가능

성과 결과성의 전 범위를 고려했지만 실제 상황에서 고려해야 되는 위험인자들은 합리적이고 타당한 범위 내의 위험요소들만이 고려의 상황이 된다.

상호작용과 선택성. 위험관리 면에서 보면 많은 위험인자들이 위험도출, 평가, 분석, 해결의 과정을 거치면서 시스템 프로세서 기간동안 위험관리 사이클을 순환한다. 이런 반복과정을 거치면서 내, 외부적인 다른 요소들과 서로 상호작용을 하면서 소멸한다. 한 위험요소의 해결책이 다른 위험요소를 발생하는 경우가 많다. 자동차 안전벨트는 치명적인 위험을 피 할 수 있지만 운전자가 부상당할 위험은 언제나 남아있다. 이 과정에서 어느 쪽을 선택할 것인가 하는 결정을 내려야 할 때 위험도 평가기준은 어디에 둘 것인가 하는 것은 중요하다.

데이터 베이스 구축. 시스템개발에 관계된 시스템 엔지니어, 프로젝트관리자, 소프트웨어 엔지니어, 각 업무분담구조의 관리자등이 도출한 위험 목록, 회의 내용, 위험완화 방안, 위험관리 계획, 일정, 후속계획 등 모든 위험관리와 관계된 데이터들이 위험관리자에 의해 위험관리 데이터 베이스에 저장된다. 위험관리자는 이 데이터를 이용하여 위험완화 계획을 조정하고 용이하게 한다. 한 프로세서기간동안 위험관리 데이터 베이스는 계속 첨가, 수정 반복되어 다음 일어날 위험의 참고자료가 된다.

9. 결 론

위험 관리를 함으로써 확실한 이득이 있는 것은 널리 알려진 사실이다. 시스템의 특성에 맞게 위험도를 분석하고 평가하는 것은 프로그램 관리자나 개발자의 의무이며 프로그램 관리자나 시스템엔지니어들은 위험분석과 평가를 함에 있어서 보다 효과적이고 정확한 방법에 의하여 결정을 내려야 한다. 전통적인 위험도 평가 방법은 Charette의 위험도 평가 방법과 위험지

수(Risk Factor)를 보는 관점이 서로 다르며 시스템에 따라서 위험지수를 계산함에 있어서 비효과적이며 경직성을 가지고 있었다. 위험도를 평가함에 있어서 위험지수를 시스템에 따라서 다르게 정의함으로써 시스템의 특성에 맞게 위험도를 정확하게 평가할 수 있을 뿐만 아니라 위험항목들을 분류하고 관리하는데 효율적이다. 또한 위험요소들을 관리하고 위험완화를 위한 해결 방안들을 실행함으로써 야기되는 비용, 인력, 일정 등의 손실을 줄일 수 있다. 새로운 위험지수(Risk Factor)를 사용함으로써 위험관리 대상 시스템의 종류에 따라 유연성 있게 위험도를 평가할 수 있었고 위험도 평가의 잠재적인 항목들의 정의는 시스템의 불확실성의 영역을 줄여 개발 목표를 명확하게 한다.

참 고 문 헌

1. Stuart Booth, "Risk Assessment and Mitigation Planning Early in the Development Life Cycle" NCOSE Annual International Symposium(1995).
2. B.B. Roberts, R.J. Winterlin "Integrated Risk Assessment : A Case Study " Proceedings of the Sixth Annual International Symposium.
3. L.T. Brekka, G.J. Vlay-TIFF "Life Cycle Risk Management" Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering.
4. Richard H. "Software Requirements Engineering" IEEE Computer Society Press Los Alamitos, California.
5. William C. Ford, Jr., Ph. D. " Risk Management on the Advanced Tomahawk Weapon Control System : A Practical Application of Risk Management in Today's Defense Environment".
6. Benjamin S. Blanchard & Wolter J. Fabrycky "Systems Engineering and Analysis" Second Edition. pp. 78-99.
7. Ian Sommerville & Pete Sawyer "Requirements Engineering" Willey. pp. 299-330.
8. Eberhardt Rechtin " Systems Architecting" Prentice Hall. pp. 125-141.
9. 시스템엔지니어링, 도서출판 문원, 민성기. pp. 75-98.