KAIS '97 추계학술대회

소프트웨어 안전성 평가를 위한 메트릭스들

김성기 · 허상녕 · 천평욱 · 박만곤
부경대학교 전자계산학과

요 약

오늘날 우리는 소프트웨어 시스템의 안전성에 대해 관심이 집중되고 있다. 본 논문에서는 시스템의 안전성에 대해 설명하고, 시스템 특히 소프트웨어 안전성을 유지하기위한 요구사항을 점검하며, 소프트웨어 안전성과 관련된 메트릭스들을 보인다. 시스템 설계부터 구현까지 메트릭스를 활용한 개발 점검은 시스템의 안전성을 높이는 중요한 역할을 할 것이라 제안한다.

1. 서 론

오늘날 인간은 직접 느끼지는 못하지만, 삶을 영위하는데 있어 아주 다양한 소프트웨어 시스템의 도움을 받고 있다. 예를 들자면, 도시가스시스템; 교통 제어 시스템, 통신 시스템, 의료시스템 등 수많은 시스템이 있다. 이들 시스템이 마비 된다면 어떤 상황이 발생할지는 명약관화하다.

이들 시스템의 안전성은 매우 중요하며 안전성에 관한 연구가 진행되고 있다. 일반적으로 시스템의 안전성은 하드웨어 제작 기술의 발달로 인해 소프트웨어의 안전성과 결부된다. 일반적인 소프트웨어 즉 기업회계 소프트웨어, 게임, 각종 에뮬레이터들은 100%의 안전성을 만족시키지 못하더라도 어느 정도의 신뢰도를 만족시킬 수 있다면 시스템에 문제가 발생하지 않으며, 실제로 납품되어 사용하고 있다. 그러나 임계 시스템들은 조그만 실수가 수많은 인명과 재산을 앗아가는 시스템이므로 철저한 관리가 필요하다.

임계시스템, 특히 소프트웨어시스템을 개발하고 유지하는데 있어 시스템의 안전성을 만족시키는 일관성 있는 표준이 필요하다고 할 수 있다.

2 절에서는 소프트웨어 안정성을 소개하며, 3 절에서는 소프트웨어개발 생명주기별에 따른 안전성 요구사항을 제안 할 것이다. 4 절에서는 소프트웨어시스템의 안전성을 점검하도록 도와주는 메트릭스를 소개하며, 마지막으로 5 절에서는 결론을 내리고자 한다.

2. 소프트웨어 시스템의 안전성

소프트웨어의 안전성은 포괄적인 안전성 프로그램의 일부분이고, 시스템 안전성을 강화하기 위해 위험들을 제거하고, 위험 요소들을 제어하는 시스템 안전성 공학의 한 부분으로 정의한다. 이때 안전성의 의미는 인간에게 발생하지 않았으면 하는 사건들로서, 원자력 발전소 방사능 유출, 핵미사일 제어 불능, 우주선 생명 유지 장치 작동, 오류 등을 예로 들 수 있다.

소프트웨어안전성 프로세서는 다음을 보증해야 한다.

- 시스템/서브시스템 안전성 분석은 소프트

웨어가 임계 안전성이다.

- 시스템/서브시스템 안전성 분석은 소프트웨어요구명세서로의 키 입력이 명백하다.
- 소프트웨어 요구명세서는 소프트웨어안전성 요구를 포함한다.
- 소프트웨어 설계와 구현이 소프트웨어안전성 요구와 적절히 결합된다.
- 소프트웨어안전성 요구의 적절한 구현을 보증하기 위해 적절한 증명자 확인이 되어야 한다.
- 테스트 계획과 절차가 소프트웨어안전성 증명의 계획을 만족시킨다.
- 소프트웨어안전성 증명의 결과가 만족스럽다.

2.1 시스템 안전성 분석

시스템 안전성 분석은 시스템 요구사항 단계에서 소프트웨어의 역할이 정의 되거나 특정 설계 논리 또는 특정 설계 처리와 연관된 위험이 감지될 때 수행한다. 소프트웨어가 위험의 잠재적인 원인이 되거나, 위험한 제어를 가한 경우 임계 안전성 영역으로 분류한다. 임계 안전성 소프트웨어는 잠재적인 위험 함수 또는 위험장치를 직접 제어 수행하는 소프트웨어, 임계장치 부품을 감시하는 소프트웨어, 임계 조건 또는 임계 상태 가능 시스템을 감시하는 소프트웨어로 정의한다.

시스템 안전성 분석은 소프트웨어요구사항 명세서 개발을 지원하기 위해 소프트웨어 안전성 요구사항을 감정하는 첫 단계로서, 요구사항은 소프트웨어 요구사항 문서에 포함되어야 하며, 시스템 안전성 분석은 프로젝트 생명주기에 연속된다.

2.2 소프트웨어 안전성

소프트웨어 안전성 활동들은 시스템과 소프트웨어개발주기의 모든 단계에 위치한다.

2.2.1 소프트웨어 안전성 계획

소프트웨어 안전성 계획은 소프트웨어경영계획 또는 안전성 경영계획으로 문서화 되어야 하며, 시스템 안전성 분석과 소프트웨어 안전성 분석, 소프트웨어개발 노력 사이의 상호관계의 정립을 목표로 한다.

2.2.2 소프트웨어요구 명세서 개발

소프트웨어 요구명세서를 개발하는 동안 소프트웨어에 할당된 시스템 수준의 요구로서 분석되고, 문서화 된다. 소프트웨어 안전성 요구의 개발과 잠재력 위험성에 대한 소프트웨어 요구 분석의 소프트웨어 안전성 작업은 소프트웨어 개발생명주기의 면에서 실행될 것이다. 소프트웨어 요구 명세화에 대한 안전성 요구의 성공적인 개발은 안전한 소프트웨어를 개발하는데 비용이 저렴하면서도 필수 불가결하다.

2.2.3 소프트웨어 설계

구조적 설계 프로세스는 소프트웨어안전성 요구를 구현하는 안전성 설계 특징과 기법의 확인을 포함한다. 안전성 명세 코딩 표준화는 안전성-임계 코드의 주석에 대한 요구와 소프트웨어 안전성과 관계된 프로그램 언어 특징에 대해 개발되어야 한다. 소프트웨어 안전성 요구명세서를 소프트웨어설계에 대한 소프트웨어 안전성 요구의 할당 후 구성요소 수준과, 안전성-임계 컴퓨터 소프트웨어 구성요소를 검증한다. 이때 안전성 연계 정보는 모든 사용자 메뉴얼에 포함되어야 하며, 주의, 경고, 안전성 연계 절차와 위험을 다루기 위한 절차를 포함한다. 소프트웨어설계 단계동안 소프트웨어안전성 요구를 검증하는 테스트 절차가 개발되며, 이 테스트 절차는 컴퓨터 환경 목록과 시스템과 레벨 테스팅 수용을 지원한다.

2.2.4 소프트웨어 구현

소프트웨어 구현은 상세화 설계에서 선택한 프로그래밍 언어로 코딩 한다. 통합과 소

프트웨어 안전성 요구를 검증하는 수락 테스트 절차는 이 단계동안 완성할 것이다.

2.2.5 소프트웨어 통합과 수락 테스팅

소프트웨어 안전성 요구의 정확성을 검증하기 위해 테스트를 수행한다. 수락 테스트는 시스템 하드웨어와 작동자와 협력하여 안전성 임계 컴퓨터 소프트웨어 구성요소의 정확한 작동을 검증한다.
수락 테스팅은 제한조건과 시스템 결합면에서 정확히 작동하는지를 검증한다.

2.2.6 소프트웨어 작동과 유지

명세화 되고, 개발되고, 분석하기 위해 정의되어진 소프트웨어 안전성 과정과 테스트 안전성 임계 컴퓨터 소프트웨어 구성요소는 변화가 일어날 때 사용된다.

3. 소프트웨어 안전성 요구사항

소프트웨어 안전성 요구사항 분석은 시스템 수준의 소프트웨어요구, 인터페이스 제어 문서, 실행 소프트웨어 요구사항 명세서 개발에 있어 안전성 임계 소프트웨어 요구사항, 고수준 안전성 요구사항에 관한 재구성의 정확성과 완벽성의 확신, 설계과정과 테스팅 과정에 대한 안전성과 관련된 권고를 검증해야 한다.

3.1 소프트웨어 안전성 구조적인 설계 분석

소프트웨어 구조적인 설계 분석을 수행하고 문서화 시켜야 한다. 구조적인 설계와 시스템 위험 분석은 구조적인 안전성 설계 분석의 입력 자료가 된다. 소프트웨어 구조적인 설계 분석은 소프트웨어 요구사항 명세서, 테스트 계획, 그리고 구조적인 설계에 대해서 다음을 검증을 해야 한다.

- 소프트웨어 안전성 요구사항 분석에 의해 감정된 소프트웨어 안전성 요구사항을 수행하는 안전성 임계 컴퓨터 소프트웨어 구성요소의 소프트웨어 부품들을 검증
- 구조적인 설계의 정확성과 완벽성, 상세한 설계를 위한 안전성과 관련된 권고
- 소프트웨어 안전성 요구사항의 테스트 범위를 확실히 하는 개발과 절차 테스트를 위한 권고를 한다.

3.2 소프트웨어 안전성 상세화 설계 분석

상세 설계, 소프트웨어 안전성 요구 사항 분석, 소프트웨어 구조적인 설계 분석과 시스템 위험 분석이 안전성 상세화 설계 분석의 입력 자료가 된다. 안전성 상세화 설계 분석은 유니트 수준 소프트웨어 부품들로서 다음을 검증해야 한다.

- 안전성 임계 컴퓨터 소프트웨어 구성요소들의 검증을 정제한 상세한 설계
- 상세한 설계의 완벽성과 정확성의 확신, 코드 실행을 위한 안전성과 관련된 권고
- 소프트웨어 안전성 요구사항의 테스트 범위의 확신, 사용자 안내와 다른 특정 문서에 포함되는 안전성과 관련된 정보

3.3 코드 안전성 분석

상세 설계, 소프트웨어 안전성 요구사항 분석, 소프트웨어 구조적인 설계 분석, 소프트웨어 안전성 상세화 설계 분석과 시스템 위험 분석이 코드 안전성 분석의 입력 자료가 된다. 코드 안전성 분석은 다음에 대해 검증해야 한다.

- 코드의 정확성과 완벽성, 잠재한 불안전한 상태의 검증, 안전성 임계 컴퓨터 소프트웨어 구성요소 들에 대한 적절한 주석
- 소프트웨어 안전성 요구사항의 테스트 범위를 확실히 하는 코드
- 사용자 가이드와 다른 특정 문서에 포함

되는 안전성과 관련된 정보의 갱신

3.4 소프트웨어 테스트 안전성 분석

테스트 결과는 모든 안전성 요구 사항이 기입되어졌다는 것을 입증하기 위해 분석된다. 또한 이런 분석은 모든 검증된 위험이 평가되어졌거나 받아들일 수 있는 수준의 위험으로 제어되어졌다는 것은 입증한다. 이러한 테스트 안전성 분석의 결과는 수행중인 시스템 안전성 분석의 활동을 위해 제공된 것이다.

3.5 소프트웨어 변화 분석

소프트웨어 변화 분석은 제안된 변화가 위험한 상태로 변하는가를 위험제어에 영향을 미치는가, 위험한 상태의 가능성을 증가시키는가, 안전성 임계 소프트웨어에 악영향을 미치는가, 소프트웨어 구성요소의 임계성을 변화 시키는가에 대해 평가한다. 분석은 어떤 영향을 미친 문서가 안전성 연계가 발생한 어떤 변화를 정확하게 반영하기 위해 갱신되어진다는 것을 검증해야 한다.

3.6 품질 보증 전망

- 소프트웨어 안전성 계획이 실행되고, 개선되고, 구현되는 것
- 소프트웨어 안전성 활동에 기인한 기술적 추천은 변화 제어 권한에 의해 재검토되어지고, 적절한 곳에 구현되어진다.
- 재검토와 회계, 감사는 소프트웨어 안전성 관계, 요구, 안내를 목표로 둔다.
- 소프트웨어 안전성 과정, 생산품 표준, 절차는 추종하고, 충족되어진다.

4. 소프트웨어 안전성 메트릭스

이 절에서는 미국방성(DoD)에서 제안한 소프트웨어 안전성 메트릭스들을 보인다.

4.1 시스템 안전성 구성원

시스템 안전성 관리자는 시스템 안전성 구성원에 대해 최소한의 자격을 만족하는 것을 필요로 한다. 일반적으로 시스템 안전성 구성원은 시스템 안전 작업에 필요한 감독 책임과 기술력 승인 권한을 가진 사람에게 제한된다. <표 1>에서 키 시스템 안전성 구성원에 대한 최소한의 자격을 보인다.

<표 1> 키 시스템 안전성 구성원에 대한 자격

프로그램 복잡도	학력	경험	증명서
매우 복잡	공학, 물리학 등의 석사	시스템 안전성 또는 연관된 분야의 4년 유경험자	최상: 안전성 공인전문가 또는 전문 공학가
보통	시스템 안전성 훈련을 받은 학사	시스템 안전성 또는 연관된 분야의 2년 유경험자	상순: 안전성 공인전문가 또는 전문 공학가
낮음	시스템 안전성 훈련을 받은 학사	시스템 안전성 또는 연관된 분야의 2년 유경험자	

4.2 심각한 위험 평가 메트릭스

모든 위험들은 각 위험별로 발생빈도와 실제 위험이 발생하였을 때 미칠 수 있는 영향에 대해 평가를 해야 한다. <표 2>에서 심각한 위험 평가 메트릭스를 보인다.

<표 2> 심각한 위험 평가 메트릭스

발생빈도 / 범주	(1) 대이변	(2) 임계	(3) 주변	(4) 무시할 수 있음
(A)자주발생 ($X>10^{-1}$)*	1A	2A	3A	4A
(B)발생 ($10^{-1}>X>10^{-2}$)*	1B	2B	3B	4B
(C) 늦음 ($10^{-2}>X>10^{-3}$)	1C	2C	3C	4C
(D)매우 늦음 ($10^{-3}>X>10^{-6}$)*	1D	2D	3D	4D
(E)거의 없음 ($10^{-6}>X$)*	1E	2E	3E	4E

• 양적인 임계 예

심각한 위험 인덱스	제안된 임계
1A, 1B, 1C, 2A, 2B, 3A	받아들일 수 없음
1D, 2C, 2D, 3B, 3C	바람직하지 않음
1E, 2E, 3D, 3E, 4A, 4B	관리자가 재검토를 받아들 일 수 있음
4C, 4D, 4E	재검토 없이 수용

4.3 잔여 위험에 대한 결정 권한 메트릭스

시스템 안전성 관리자료 시스템에 존재하는 위험이 무엇인지 알아야 하며, 존재하는 위험에 대한 평가를 하여야 한다.
<표 3>에서 잔여 위험에 대한 결정 권한 메트릭스를 보인다.

<표 3> 잔여위험에 대한 결정 권한 메트릭스

범주 / 발생빈도	CATASTROPHIC	CRITICAL	MARGINAL	NEGLIGIBLE	MARGINAL	NEGLIGIBLE
자주 발생	HIGH	HIGH	SERIOUS	SERIOUS	SERIOUS	SERIOUS
발생	HIGH	HIGH	SERIOUS	LOW	SERIOUS	LOW
낮은	HIGH	SERIOUS	LOW	LOW	LOW	LOW
매우 낮은	SERIOUS	LOW	LOW	LOW	LOW	LOW
거의 없음	SERIOUS	LOW	LOW	LOW	LOW	LOW

심각한 위험 레벨	결정 권한
HIGH	구성요소 구입 행정부
SERIOUS	프로그램 직원
LOW	프로그램 관리자

4.4 소프트웨어 위험 임계성 메트릭스

소프트웨어 위험 임계성 메트릭스는 심각한 위험 평가 메트릭스와 유사하다. 행에 대해서는 불운한 진단을 사용하고, 열에 대해서는 소프트웨어 제어 진단을 사용한다. "1"은 위험을 받아들일 수 없으며, "2"에서 "4"는 바라지 않거나 경영 활동으로부터 수락을 필요로 한다.
형색인 번호는 설계가 받아들일 수 없다는 것을 의미하지 않으며, 더욱 많은 자원들의 분석과 소프트웨어와 시스템간의 상호작용 테스트에 적용될 필요가 있다. <표 4>에서 소프트웨어 위험 임계성 메트릭스를 보인다.

*소프트웨어 제어 범주

I. 소프트웨어는 위험 발생 배제의 중재 가능성 없이 잠재적으로 위험한 하드웨어 시스템, 부 시스템 또는 구성요소에 대한 자체적인 제어를 실험한다.

II. 소프트웨어는 독립적인 안전성 시스템에 의해 중재를 위해 시간으로 하여금 위험을 완화 시키는 잠재적으로 위험한 하드웨어 시스템, 서브시스템, 또는 구성요소에 대한 제어를 실험한다.

III. 소프트웨어 요소 논쟁은 인간의 작용으로 하여금 제어기능을 완성하기를 필요로 하는 잠재적으로 위험한 하드웨어 시스템, 서브시스템, 구성요소에 대한 명령을 내린다.

IV. 소프트웨어는 안전성 임계 하드웨어시스템, 서브시스템 또는 구성요소를 제어하지 못하고, 안전성 임계 정보를 제공하지 못한다.

<표 4> 소프트웨어 위험 임계성 메트릭스

범주 / 제어범주	CATASTROPHIC	CRITICAL	MARGINAL	NEGLIGIBLE
I	1	1	3	5
II	1	2	4	5
III	2	3	5	5
IV	3	4	5	5

심각한 위험 레벨　　제한된 임계

1　매우 위험-주용한 분서고가 테스팅의 근원
2　중간 위험-요구가 설계분석과 필요한 상세한 테스팅
3-4　보통 위험-고수위 분석과 경영 활동 승인에 받아들일 수 있는 테스팅
5　저 위험-수용 가능

4.5 응용 메트릭스

시스템 안전성 프로그램의 중요한 도전 중 하나는 작업과 위험 분석의 유용성의 선택과 타이밍이다. 이 선택에 있어 이정표와 이론적 근거를 제공해야 한다. 이정표는 공학자, 운영자, 제공자들이 발견한 문제점에 대한 질문과 해답을 도출하는 토의 과정에서 정립되었다.

만일 적당한 작업이 선택되었다면, 작업은 특별히 주문되며 MA에 의해 상세화 되어야 한다. 각각의 작업에 대해 요구된 타이밍과 수준은 개인적인 경험과 프로그램 요구에 매우 의존하게 되므로 엄격하고 빠른 법칙을 적용할 수 없다.

<표 1> <표 2>에서 일반적인 작업 선택 룰을 보였다. 시스템 안전성 프로그램 개발을 위해 메트릭스의 사용은 적절한 이정표가 된다. <표 5> ~ <표 7>에서 이들 메트릭스들을 보인다.

<표 5> 시스템 응용 개발을 위한 메트릭스

TASK	TITLE	TASK TYPE	PROGRAM PHASE 0	I	II	III
101	시스템 안전성 프로그램	MGT	G	G	G	G
102	시스템 안전성 프로그램 계획	MGT	G	G	G	G
103	관련 계약자, 하도급 계약자 그리고 AE 회사에 관한 통합 관리	MGT	S	S	S	S
104	시스템 안전성 프로그램 재검/감사	MGT	S	S	S	S
105	SSG/SSWG 지원	MGT	G	G	G	G
106	위험 수락과 위험 해결	MGT	S	G	G	G
107	시스템 안전성 진행 요약	MGT	S	G	G	G
108	건수(발전)안전성 프로그램 요구사항	MGT	S	S	S	S
201	위험 예비 리스트	ENG	G	S	S	S
202	위험 예비 분석	ENG	G	G	G	GC
203	안전성 요구사항/기준 분석	ENG	G	S	S	S
204	서브시스템 위험 분석	ENG	N/A	G	G	GC
205	시스템 위험 분석	ENG	N/A	G	G	GC
206	운영과 지원 위험 분석	ENG	S	G	G	GC
207	건강 위험 평가	ENG	G	G	G	GC
301	안전성 평가	ENG	G	G	G	G
302	안전성 검사와 평가	ENG	G	G	G	G
303	기술 변화에 제안에 관한 안전성 기준, 역에서 변화 주의, 소프트웨어 문제점 보고서, 그리고 탈선과 기권에 대한 요청	ENG	N/A	G	G	G
401	안전성 검증	ENG	S	G	G	S
402	안전성 추종 평가	ENG	S	G	G	S
403	폭발적인 위험 분류와 특징적인 자료	MGT	S	S	S	S
404	폭발적인 군수품 데각 자료	MGT	S	S	S	S

NOTES:

TASK TYPE
ENG - 시스템 안전성 공학
MGT - 시스템 안전성 관리

PROGRAM PHASE
O - 개념 진단
I - 프로그램 정의와 위험 축소
II - 공학과 제조 개발
III - 생산, 수비/전개와 운영 지원

APPLICABILITY CODES
S - 선택적 적용 가능
G - 일반 적용가능
GC - 설계 변화시 적용 가능
N/A - 적용 불가능

<표 6> 기능 획득 응용 메트릭스

TASK	TITLE	TASK TYPE	PROGRAM PHASE I	II	III	IV
101	시스템 안전성 프로그램	MGT	G	G	G	G
102	시스템 안전성 프로그램 계획	MGT	S	G	G	S
103	관련 계약자, 하도급 계약자 그리고 AE 회사에 대한 통합 관리	MGT	S	S	S	S
104	시스템 안전성 프로그램 검토/감시	MGT	G	G	G	G
105	SSG/SSWG 지원	MGT	G	G	G	G
106	위험 추적과 위험 해결	MGT	G	G	G	G
107	시스템 안전성 진행 요약	MGT	S	S	S	S
108	전술(탈전)안전성 프로그램 요구사항	MGT	S	S	S	S
201	위험 예비 리스트	ENG	G	N/A	N/A	S
202	위험 예비 분석	ENG	G	S	N/A	S
203	안전성 요구사항/기준 분석	ENG	G	S	S	GC
204	서브시스템 위험 분석	ENG	N/A	S	G	GC
205	시스템 위험 분석	ENG	N/A	G	G	GC
206	운영과 지원 위험 분석	ENG	S	G	G	GC
207	갱한 위험 평가	ENG	G	S	N/A	N/A
301	안전성 평가	ENG	N/A	S	G	S
302	안전성 검사와 평가	ENG	G	G	G	G
303	기술 변화에 제안에 관한 안전성 개관, 명세서 변화 주의, 소프트웨어 문제점 보고서, 그리고 탈선과 기권에 대한 요청	ENG	S	S	S	S
401	안전성 검증	ENG	N/A	S	S	S
402	안전성 추종 평가	MGT	N/A	S	S	S
403	폭발적인 위험 분류와 특징적인 자료	ENG	N/A	S	S	S
404	폭발적인 군수품 매각 자료	MGT	N/A	S	S	S

NOTES:

TASK TYPE

ENG - 시스템 안전성 공학

MGT - 시스템 안전성 관리

PROGRAM PHASE

I - 프로그래밍과 요구

II - 개념 설계

III - 마지막 설계

IV- 구축

APPLICABILITY CODES

S -선택적 적용 가능

G - 일반 적용가능

GC - 설계 변화시 적용 가능

N/A - 적용 불가능

<표 7> 비용과 위험 정도에 기반을 둔 전형적인 프로그램을 위한 작업 선택 메트릭스

Small Dollar or Low Risk Program	Medium Dollar or Average Risk Program	Large Dollar or High Risk Program
TASK 101 - System Safety Program TASK 102 - SSPP TASK 201 - Preliminary Hazard List TASK 202 - PHA TASK 205 - SHA TASK 207 - HHA TASK 301 - Safety Assessment	TASK 101 - System Safety Program TASK 102 - SSPP TASK 104 - Reviews/ Audits TASK 105 - SSG/SSWG TASK 106 - Hazard Tracking TASK 201 - Preliminary Hazard List TASK 202 - PHA TASK 204 - SSHA TASK 205 - SHA TASK 206 - O&SHA TASK 207 - HHA TASK 301 - Safety Assessment	TASK 101 - System Safety Program TASK 102 - SSPP TASK 103 - Integration/Mgmt of Contractors TASK 104 - Reviews/Audits TASK 105 - SSG/SSWG TASK 106 - Hazard Tracking TASK 107 - Safety Progress Reports TASK 201 - Preliminary Hazard List TASK 202 - PHA TASK 204 - SSHA TASK 205 - SHA TASK 206 - O&SHA TASK 207 - HHA TASK 301 - Safety Assessment TASK 302 - Test and Eval Safety TASK 303 - Safety ECPs TASK 401 - Safety Verification TASK 403 - Explosive Hazard Class

주의:

1. 각 선택된 작업은 용도나 목적에 맞게 만들어져야 하고 MA 상세서가 계약서에 첨부되어야 한다.
2. 이러한 작업들을 프로그램의 다른 면에 적용될 수 있다.

5. 결론

위험을 제거하기 위해 프로그램에 초기에 기울인 노력이 시스템 개발비용의 감소를 가져올 수 있다. 물론 사용자에게는 프로그램의 궁극적인 미래를 위험에 빠뜨리고, 그것들로 하여금 결과적으로 일어날 사고에 대한 책임을 초래하질도 모르는 심각하고, 악명 높은 위험을 피할 의욕이 있다.

설계에 파고들 위험들은 아마도 공학 변화 제안 프로세스를 통해 목표가 지정되어야 한다. 그러나 공학 변화 제안 프로세스는 사업용으로는 고가의 기법이다. 프로그램 메니저가 반대하는 가장 어려운 공학 변화 제안은 안전성 표시다.

만일 안전성 문제가 개발의 후반부까지 발생되거나 수정되지 않은 채로 남아있다면, 곤경이 당신의 예산과 스케줄에 파괴를 가할 것이며 예기치 못한 상황을 가져올 것이다.

본 논문은 안전한 시스템을 획득하기 위한 세가지 단계를 제안한다.

첫째, 불필요한 위험의 초기 발생을 방지해야 한다. 개발자와의 의사소통을 통해 이러한 행동을 하면 안전성이 여러분 개인에게 중요하게 될 것이다. 위험이 시스템 초기에 존재하고, 많은 수와 극심한 잠재적 불상사를 최소화하기 위해, 설계하는 동안 개발자로 하여금 설계 공학자에게 가장 최적화된 메트릭스를 기저로 한 시스템 위험에 주의하도록 지시하라. 이 첫번째 단계는 중요한 비용 증가와 문제회피기술에 대해 가장 효과적임이 역사적으로 증명되었다.

두 번째로 시스템 안전성 활동이 명세성 프로그램 요건을 충족시킬 수 있어야 한다. 그러기 위해서는 최적화된 메트릭스는 필수적이다. 만일 위의 첫 단계를 생략한다면, 설계시 발생하게 될 많은 수와 다양한 위험들을 처리하게 되어 더욱더 큰 시스템 안전성 노력이 필요하게 될 것이다.

마지막으로, 여러분은 남은 위험을 다룰 필요가 있다. 그들의 본질과 충격을 이해하고, 그들이 적절히 배치되는 것을 가장 적절한 메트릭스를 통해 확신함으로써 당신들은 이러한 것을 할 수 있다. 수락된 위험 대신에 적절한 수위의 권한에 발생할 위험이 수락을 확신하는데 신경을 기울여라.

일반적으로 위험이 더 클수록, 인정을 위해 필요한 승인 수준이 더 높아진다. 높은 수준의 위험이 결정을 짓는 사람들에 정당화되어야 하지, 안전성 집단에 정당화 되어서는 안된다는 것을 주의하라

참고문헌

[1] Alan Underwood., "A Framework for Certifying Critical Software Systems.", Twelfth Annual CSR Workshop, Bruger pp. 12-15 September 1995.

[2] Navcy G. Leveson, Peter R, Havry., "Analyzing Software Safety", IEEE Transactions on Reliability, Vol. R-35, No. 1, pp. 114-118, April 1986.

[3] John C. Munson, Taghi M. Kheshgobt, "Applications of a Relative Complexity Metric for Software Project Management", J. Syst. Software, 12(3): pp. 283-291, 1990.

[4] NMI 2410.10, NASA Software Management "Assurance and Engineering Policy.", April 20, 1993.

[5] DoD-STD-2167A-Military Standard, "Defence System Software Development", February 29, 1988.

[6] Norman Fenton, Austin Melton., "Deriving Structurally Based Software Measures", J Syst. Software, 12(3), pp. 177-187, 1990.

[7] J. Voas, L.Morell, K. Miller., " Predicting

Where Faults Can Hide From Testing," IEEE Software, 8(2), Marh 1991.

[8] Nancy G. Leveson., "Software Safety: Why, What, and How", ACM Computing Surveys, Vol. 18, No. 2, pp. 125-163, June 1986.

[9] Mark D, Hansen, Ronalel L. Watts., "Software System Safety and Reliability", Proceedings Annual Reliability and Maintainability Symposium, pp. 214-217, 1998.