항공안전을 강화하기 위한 소프트웨어 안전성 법제도 방안

The Legal System Method of Software Safety to Strengthen Aviation Safety

지정은*, 이상지*, 신용태**

Jung-Eun Jee*, Sang-Ji Lee* and Yong-Tae Shin**

요 약

지식 · 정보 · 기술력 중심의 지식기반 경제 원천인 소프트웨어로 인한 결함은 항공기의 운용에 핵심 역할을 수행하는 엔진에 영향을 준다. 따라서 소프트웨어의 안전성분석을 통해 항공안전을 강화하여 결함으로부터의 위험을 최소화해야 한다. 본 논문에서는 항공기 결함과 소프트웨어 안전성 법·제도를 살펴보고 항공안전을 강화하기 위한 법·제도 개선 및 제정 방안을 제안한다. 안전성분석과 관련된 용어 정립, 안전성이 포함된 품질인증 기준, 안전성분석서를 첨부해야 하는 품질인증 신청, 평가 및 인증기관 세부지침 개정 등의 항목으로 기존 법·제도를 개선해야 한다. 또한, 소프트웨어 평가 및 인증 의무화, 소프트웨어 생명주기에 다른 지속적 평가, 표준화된 개발방법론 도입 의무화, 고급인력 양성 제도 강화 등의 항목으로 신규 법·제도를 제정해야한다. 소프트웨어 안전성과 관련된 기존 법·제도를 개선하고 신규 법·제도를 제정하여 소프트웨어의 품질 향상과 강화된 항공안전을 기대할 수 있다.

Abstract

The defect caused by the software industry that is the source of knowledge-centric, information-centric and technology-centric affects an engine which operate a major role for operation of aircraft. Therefore, we should minimize the danger from the defect by strengthening the stability of aviation through the stability analysis of software. In this paper, we examine the laws and systems about the aircraft defects and software safety and propose the enhancement and the enactment of the law or measures to strengthen aviation safety. We should the existing law or system as items, such as the revision related to the safety analysis, standards of quality assurance including safety, application of quality assurance that you must attach the safety analysis report, assessment of detailed instructions of certification authorities. In addition, we should enact the new law and system as items such as the mandatory software evaluation and certification, continuous assessment based on the software life cycle, mandatory introduction of a standardized development methodology, strengthening of advanced workforce system. We can expect the improvement of software quality and an enhanced aviation safety by improving existing laws or systems and enacting new laws or systems.

Key words : aircraft defect, software safety, aviation safety, the legal system method of software safety

* 숭실대학교 컴퓨터학과 (Department of Computing, Soong-sil University)
** 숭실대학교 컴퓨터학부 (School of Computer Science and Engineering, Soong-sil University)
· 제1저자 (First Author) : 지정은
· 투고일자 : 2011년 9월 5일
· 심사(수정)일자 : 2011년 9월 6일 (수정일자 : 2011년 10월 13일)
· 게재일자 : 2011년 10월 30일

Ⅰ. 서 론

항공산업은 지구상에서 어느 목적지라도 하루 이내에 도착할 수 있도록 도와주는 등 현재의 정보화, 세계화 시대에 이동 및 전달 수단으로써 편리함을 제공해주고 있다. 항공, 기계, 전기, 전자, 화학, 재료 등 여러 공학의 결합으로 완성된 항공산업을 흔히, 종합공학 및 과학의 산물이라고 한다. 이러한 항공산업의 발전은 다른 산업에 큰 파급효과를 준다. 항공산업의 핵심인 항공기는 수만에서 수십만 개의 부품들로 구성되어 있으며 항공기는 엔진에 의해 작동된다. 최근 항공기의 운용에 중심 역할을 수행하는 엔진 결함으로 인한 항공사고가 빈번하게 발생하고 있다. 따라서 항공사고의 원인을 해결하고 피해를 줄이기 위해 소프트웨어 산업의 안전성에 대한 중요성이 제고되고 있다[1].

소프트웨어 산업은 급변하는 산업 환경에서 기업 및 국가의 경쟁력이 노동과 자원의 중심에서 지식, 정보, 기술력 중심의 지식기반 경제로 급속히 진화되면서 전체 산업의 발전을 변화시키는 원천으로 자리매김하고 있다. 소프트웨어가 차지하는 부분이 많아질수록 그에 따른 피해도 증가하고 있다. 정보기기의 고장, 시스템 및 네트워크의 마비 등과 같은 장애를 발생시켜 막대한 손실을 준다. 따라서 결함이 없는 소프트웨어를 개발하여 안전하게 사용할 수 있도록 해야 한다.

소프트웨어 안전성은 소프트웨어가 사용되는 특정한 상황에서 다양한 환경에 대한 위해로부터 위험 수준을 유지하는 소프트웨어 제품의 능력을 의미한다[2]. 소프트웨어 안전성분석을 통해 소프트웨어 개발시 발생할 수 있는 결함을 제거하여 신뢰성을 확보하고, 소프트웨어 사용시 사용자가 감내할 수 있는 위험 수준을 제공해야 한다. 이를 위해서는 일관성 있는 안전성분석과 이를 뒷받침 할 수 있는 법·제도 마련이 필요하다.

국내에는 소프트웨어산업진흥법, 소프트웨어산업진흥법시행령, 소프트웨어산업진흥법시행규칙, 정보화 촉진기본법 등 소프트웨어 안전성분석과 관련된 법이 있다. 또한, 소프트웨어 제품 품질 향상을 위한 GS(Good Software) 및 KS 인증제도가 있다. 그러나 소프트웨어 안전성 관련 법·제도에는 소프트웨어의 결함으로 야기될 피해를 줄이기 위한 세부 항목이 제시되어 있지 않다. 따라서 본 논문에서는 기존 법·제도를 기반으로 소프트웨어 안전성분석 관련 기존 법·제도 개선 및 신규 법·제도 제정 방안을 제안한다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 항공기 결함과 소프트웨어 안전성 법·제도에 대해 살펴본다. 3장에서는 소프트웨어 안전성 관련 기존 법·제도 개선을, 4장에서는 소프트웨어 안전성 관련 신규 법·제도 제정 방안을 제안한다. 마지막 5장에서는 결론을 맺는다.

Ⅱ. 항공기 결함과 소프트웨어 안전성 법제도

2-1 항공기 결함

항공기 부품 중 하나인 엔진은 열에너지를 기계에너지로 변환시켜 외부에서 공급하는 기계장치로 항공기에서 핵심적 역할을 수행한다. 엔진에는 한 가지 목적을 위해 공동작용을 하는 일련의 프로그램들에 대해 전반적인 운영을 조정하거나 응용프로그램 내에서 중심적인 기능을 하고 있는 프로그램이 내장되어 있다. 결함은 실행되어야하는 기능을 수행하지 못하게 하는 조건이 되므로 프로그램의 기능을 제대로 수행할 수 없게 하는 원인을 제공한다.

국내뿐 아니라 미국, 호주 등 많은 국가에서 항공결함으로 인한 사고가 빈번하게 발생하고 있다[3]. 가장 많은 사고 원인은 엔진 이상으로 인해 발생하였으며, 국가별 사고 원인과 피해 내용은 표 1에서 보여준다.

항공 사고의 원인으로 엔진 결함은 엔진이 정지, 미작동, 진동, 연료 누수 등 이 발생하는 것이며 기체 결함은 기체 유압기, 연료계기판, 연료 분배장치, 여압 장치 등 항공기 기체의 기계적인 결함이다. 또한, 정비 결함은 안전 유지를 위하여 결함의 수정 및 결함의 사전 제거를 이행하지 않아 생긴다. 이런 원인으로 항공기는 비상 착륙, 회항, 시간 지연 등으로 대처하나 피해는 심각하다.

표 1. 국가별 항공사고 원인과 피해 사례
Table 1. The cases of causes and damages about aviation accidents in the respective countries

국가	날짜	사고 원인	피해 내용
국내	2009.6.	엔진에서 연료 누수	비상 착륙
	2009.12.	엔진 정지	비상 착륙
	2010.1.	엔진전기계통 및 보조동력장치 고장	비상 착륙
	2010.1.	상공에서 엔진 이상	회항
	2010.5.	이륙직전 기체결함	주활주로에 정지 및 활주로 폐쇄
	2010.9.	기체 유압기 이상	비상 착륙
	2010.10.	부품 이상	시간 지연
	2010.12.	연료계기판 이상	시간 지연
	2011.1.	연료누수	시간 지연
	2011.3.	기체 하단부 이상 징후	회항
	2011.6.	연료 분배장치 결함	사망 사고 발생
	2011.6.	내부 기압 조절하는 여압 장치 결함	비상 착륙
미국	2011.4.	정비 결함	항공기 파손
	2011.5.	엔진 이상	시간 지연
	2011.5.	기체 결함	시간 지연
호주	2010.11.	엔진 고장	회항
	2011.5.	엔진 진동	회항
필리핀	2011.5.	기체 결함	출발 지연
싱가포르	2011.7.	기체 떨림, 엔진 정지	회항
뉴질랜드	2011.7.	연료 누수	운항 취소
캐나다	2011.7.	기체 결함	운항 취소

국내 및 국외에서의 항공기 사고의 대부분은 엔진 결함으로 인한 것이다. 엔젠제어장치는 엔진의 시동, 정치, 정상상태 제어, 가감속 제어, 모니터링 및 자기진단, 엔진 운용조건 및 작동상태 관리 등 엔진 작동을 직접적으로 제어한다. 전자식 엔진제어장치(EEC; Electronic Engine Control)가 보편적으로 엔진에 장착되었으나 최근 전자동디지털식 엔진제어장치(FADEC; Full Authority Digital Electronics Control)도 사용되고 있다. 엔진제어장치는 엔진의 작동을 직접적으로 제어하기 때문에 엔진제어장치에 사용되는 소프트웨어에 대한 잠재적인 오류는 항공기 전체의 안전에 중대한 영향을 주어 심각한 상황을 유발할 수 있다[4].

따라서 항공기 결함을 예방하기 위해서는 엔진제어장치 소프트웨어를 확인 및 검증할 수 있도록 안전성을 강화해야 한다.

2-2 소프트웨어 안전성

소프트웨어 안전성은 소프트웨어에 부정적으로 영향을 주고 전체 시스템이 실패하는 원인이 되는 잠재적인 위험의 식별과 평가에 초점을 맞추고 있는 소프트웨어 품질보증 활동이다. 또한, 소프트웨어에서 추구하는 안전성은 일련의 생산 과정에서 발생할 수 있는 결함을 최소화하고 최대의 신뢰성을 유지함으로써, 소프트웨어 사용 시점에 사용자가 수용할 수 있는 일정 수준의 위험을 감내할 수 있도록 소프트웨어 능력을 확보하는 것이다[5].

일반적으로 소프트웨어는 요구사항 분석, 설계, 구현, 테스팅, 유지보수 등의 생명주기(life-cycle)를 갖는다. 기존의 많은 사고들이 소프트웨어 요구사항 단계에서의 결함으로 인해 발생되었다.

대다수 컴퓨터들이 동일한 소프트웨어를 사용하고 컴퓨터들이 네트워크로 연결되어 있어 보안상의 소프트웨어 결함은 다음 표 2와 같은 침해요인이 발생된다.

표 2. 소프트웨어 안전성 침해요인
Table 2. The reason of infringement of software safety

침해요인	내용
개발 환경으로 인한 문제점	• 이·기종 환경 및 다양한 플랫폼에서의 개발 • 개발 환경의 빠른 변화
시스템의 복잡도 증가	• 시스템의 복잡도 증가로 분석하기 어려워 안전성, 신뢰성 보장이 어려움 • 악의적 코드 또는 결점을 찾아내기 어려움
개발자로 인한 문제점	• 여러 명의 개발자가 공동으로 개발 • 다양한 언어와 많은 코드량으로 인한 Code Inspection의 어려움
내장된 확장성	• 수정·변경·갱신 또는 확장 과정에서 악의적 코드 삽입 가능

소프트웨어가 여러 분야의 시스템에 중요한 역할을 수행하고 있다. 따라서 소프트웨어 안전성을 침해할 수 있는 요인을 제거하기 위해서는 소프트웨어 안전성과 관련된 법·제도에 관심이 높아지고 있다.

2-3 소프트웨어 안전성 법제도

국내 소프트웨어 안전성분석과 관련된 법·제도에서 소프트웨어 안전성 관련 법 조항은 다음 그림 1과 같다[6].

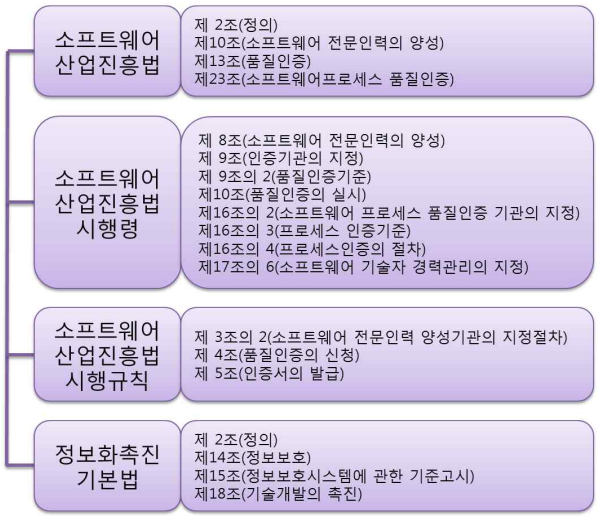


그림 1. 소프트웨어 안전성 관련 법
Fig. 1. The safety-related law of software

소프트웨어 안전성분석과 관련된 제도로는 우수 국산 소프트웨어를 발굴하고 제품의 품질을 향상시키기 위해 일정한 수준의 품질을 갖춘 소프트웨어 제품에게 국가가 부여하는 GS 인증이 있다. 또한 KS에 적합한 제품을 인증기관을 통해 심사를 받아 인정을 받는 제품인증제도인 KS인증 등의 제도가 있다[7].

국제공통평가기준(CC : Common Criteria)은 세계 여러 국가에서 개발 및 생산되고 있는 정보보호제품에 대한 평가기준을 국제적으로 표준화한 것이다[8]. 각 나라 및 지역별로 서로 다른 평가기준을 운용함으로써 발생하는 중복평가 문제를 해소하고 기술적으로 보다 진보된 범세계 공통의 보안 평가기준을 지향하기 위하여 단일 평가기준 제정의 필요로 공통평가기준을 만들게 되었다.

국가마다 상이한 평가기준을 연동시키고 평가결과를 상호인증하기 위해 제정된 CC는 세계적으로 유통되는 정보보호제품에 대한 단일 평가기준을 마련하여 관련 시장에서 해당 제품들에 대한 신뢰성을 보장해준다.

Ⅲ. 소프트웨어 안전성 기존 법제도 개선

3-1 소프트웨어 안전성 분석 관련 용어 정립

소프트웨어산업진흥법 조항에 소프트웨어 안전성 관련 내용이 언급되고 있지 않으므로 소프트웨어 안전성과 관련된 용어를 정립해야한다. 소프트웨어산업진흥법 제2조(정의)에 정립해야하는 용어는 다음 표 3과 같다.

표 3. 소프트웨어 안전성 분석 관련 용어
Table 3. The analysis-related terminologies of software safety

용어	내용
소프트웨어 안전성	9. "소프트웨어 안전성"이라 함은 소프트웨어가 사용되는 특정한 상황에서 사람, 비즈니스, 소프트웨어, 자산 혹은 환경에 대한 위해로부터 감수할 만한 위험 수준을 유지하는 소프트웨어 제품의 능력이다.
소프트웨어 안전성 분석	10. "소프트웨어 안전성분석"이라 함은 소프트웨어 안전성을 검증하여 소프트웨어의 오류 및 결함이 발생하는 것을 사전에 방지하고자 하는 분석활동을 통칭한다.
소프트웨어 생명주기	11. "소프트웨어 생명주기"라 함은 소프트웨어 프로젝트가 시작될 때부터, 폐기 될 때까지 전 과정을 말한다. 소프트웨어 생명주기는 명세, 설계, 구현, 테스트, 유지보수의 단계로 구분된다.
소프트웨어 오류	12. "소프트웨어 오류"라 함은 소프트웨어 명세서에서 정의되어 있는 기능을 동작하지 않거나 정의되어 있지 않은 기능을 동작하는 것을 말한다.
소프트웨어 유지보수	13. "소프트웨어 유지보수"라 함은 소프트웨어가 개발되고 사용 중에 있는 소프트웨어의 성능향상, 오류수정, 기능향상을 위하여 계속적으로 관리하는 것을 말한다.

소프트웨어 산업은 여러 분야에서 사용되어지고 이로 인한 소프트웨어 결함에 의해 발생되는 피해를 막기 위해 소프트웨어 안전성에 대한 개념 정립이 필요하다.

3-2 소프트웨어 품질인증 기준 변경

소프트웨어산업진흥법 시행령 제9의2(품질인증기준)①에는 소프트웨어의 안전성에 대한 특성이 품질인증기준으로 제시되고 있지 않다. 따라서 품질인증기준에 안전성에 대한 특성을 추가해야 한다. 변경된 품질인증기준은 다음 표 4와 같다.

표 4. 소프트웨어 안전성이 포함된 품질인증기준
Table 4. The quality certification standard included software safety

변경 전	변경 후
① ① 법 제13조 제3항에 따른 소프트웨어 품질인증의 기준은 다음 각 호와 같다. 1. 소프트웨어의 기능을 정확하게 실행 할 것 2. 소프트웨어의 신뢰성·효율성·사용과 유지·보수의 편의성 및 이식의 용이성이 지식경제부장관이 정한 수준 이상일 것	① ① 법 제13조 제3항에 따른 소프트웨어 품질인증의 기준 은 다음 각 호와 같다. 1. 소프트웨어의 기능을 정확하게 실행할 것 2. 소프트웨어의 신뢰성·효율성·사용과 유지·보수의 편의성 및 이식의 용이성 및 안전성이 지식경제부장관이 정한 수준 이상일 것

소프트웨어의 품질을 높이기 위해서는 소프트웨어 생명주기에 걸친 전체 과정에서 품질 특성을 관리해야 한다. 소프트웨어 안전성을 품질특성에서 관리하도록 하여 모든 소프트웨어 품질 인증 시 안전성에 대한 인증을 받을 수 있도록 한다.

3-3 품질인증 신청 제출서류 추가

소프트웨어산업진흥법 시행규칙 제4조(품질인증의 신청)에 소프트웨어품질인증을 신청할 때에 제품설명서, 사용자취급설명서뿐만 아니라 소프트웨어 안전성 분석 기법을 통해 제품의 안전성을 인증할 수 있는 안전성분석서도 제출하도록 추가해야 한다. 안전성분석서가 포함된 품질인증 신청은 다음 표 5와 같다.

표 5. 안전성분석서가 포함된 품질인증 신청
Table 5. The quality certification application included an analysis document of safety

변경 전	변경 후
영 제10조제1항에 따라 품질인증을 신청하고자 하는 자(이하 “신청인”이라 한다)는 별지 제6호서식의 소프트웨어품질인증신청서에 다음 각 호의 서류 및 인증받고자 하는 소프트웨어를 첨부하여 인증기관의 장에게 제출하여야 한다. 1. 제품설명서 2. 사용자취급설명서	영 제10조제1항에 따라 품질인증을 신청하고자 하는 자(이하 “신청인”이라 한다)는 별지 제6호서식의 소프트웨어품질인증신청서에 다음 각 호의 서류 및 인증받고자 하는 소프트웨어를 첨부하여 인증기관의 장에게 제출하여야 한다. 1. 제품설명서 2. 사용자취급설명서 3. 안전성분석서

3-4 평가 · 인증기관의 세부 지침 개정

국내 법·제도를 기반으로 평가 및 인증기관에서 소프트웨어 안전성을 강화할 수 있는 세부 지침을 개정하여 시행해야 한다. 세부 지침 개정 사항은 다음 표 6과 같다.

표 6. 세부 지침 개정 사항
Table 6. The details of guideline revisions

지침	개정 내용
소프트웨어 평가 기준 등급화	• 오류발생률과 위험도에 따라 등급을 나누어 소프트웨어의 안전성 분석과 소프트웨어 평가
등급우선구매제 도입	• 모든 소프트웨어가 평가·인증을 받는 제도 • 더 높은 등급을 받은 소프트웨어 우선 구매
평가기관의 평가	• 평가기관을 평가할 수 있는 상위 기관에 대한 기준 설정 • 조직과 인력 및 환경조건에 따른 평가기관의 계층적 분류

평가 및 인증기관의 개정 내용은 통일성 있는 평가 기준으로 등급화하고, 평가기준에 따라 등급별로 우선적으로 구매될 수 있다. 또한, 평가기관의 공정하고 정확한 평가를 통해 평가기관을 계층화해야 한다.

3-5 법 · 제도 개선 기대효과

소프트웨어 안전성분석 법·제도를 개선함으로써 기대되는 효과는 다음 그림 2와 같다.

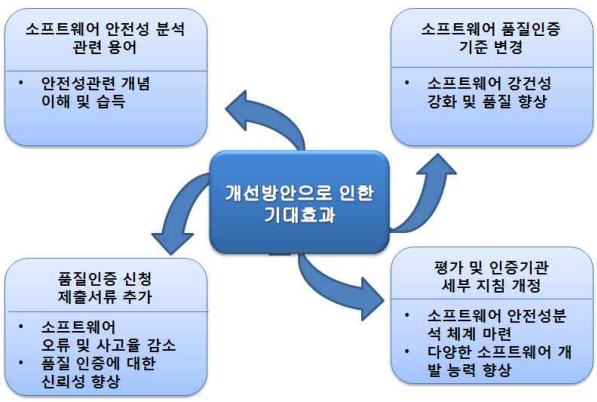


그림 2. 법 · 제도 개선방안으로 인한 기대효과
Fig. 2. The expected effect through an improvement method of a legal system

소프트웨어 안전성 분석 관련 용어의 정립은 안전성에 대한 정확한 개념을 이해할 수 있고 습득할 수 있다. 소프트웨어 품질인증 기준에 안전성을 추가하여 소프트웨어의 안전성을 평가하면 강건성을 높일 수 있고 품질을 향상시킬 수 있다.

품질인증 신청 제출서류에 안전성분석서를 추가하면 소프트웨어 및 시스템의 오류 및 사고율을 감소시킬 수 있게 해준다. 또한, 품질인증을 신청할 때 안전성분석서를 포함시켜 인증의 신뢰성을 향상시킬 수 있다.

평가 및 인증기관에 세부 지침을 개정하면 소프트웨어 안전성분석 체계를 갖출 수 있고 다양한 소프트웨어의 개발 능력을 높일 수 있다.

Ⅳ. 소프트웨어 안전성 신규 법제도 제정

4-1 신규 법제도 제정체계

소프트웨어 안전성 강화를 위한 신규 법·제도는 국가에서 마련한 법·제도를 기준으로 평가 및 인증기관에서 이를 위한 지침을 마련하여 시행되어야 한다. 또한, 현재 미흡한 법·제도를 제정하기 위해 그림 3과 같이 소프트웨어 안전성 관련 법·제도 제정체계가 필요하다.

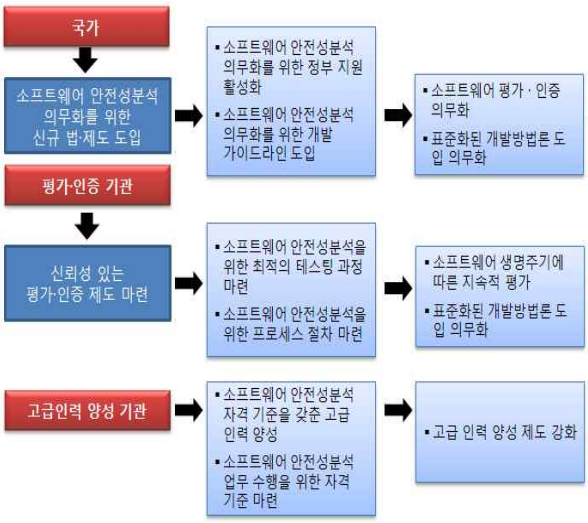


그림 3. 신규 법제도 제정체계
Fig. 3. The establishment system of a new legal system

신규 법·제도를 제정하기 위해서는 국가, 평가·인증 기관, 고급인력 양성 기관 등의 협력을 통해 이루어질 수 있다. 먼저, 국가는 소프트웨어 안전성분석 의무화를 위한 신규 법·제도를 제정하여 제도적으로 근거를 마련해줘야 한다. 소프트웨어 안전성분석 의무화를 위해서는 정부 지원이 활성화되어야 하며 정부 차원의 개발 가이드라인이 도입되어야 한다.

평가·인증 기관은 제정된 법·제도를 기반으로 신뢰성 있는 평가·인증 제도를 마련해야 한다. 신뢰성 있는 평가·인증 제도는 여러 번의 테스팅 과정을 포함하여 안전성을 확인해야 하고 안전성을 인증할 수 있도록 프로세스 절차를 도입하여 지속적인 평가를 수행해야 한다.

고급인력 양성 기관에서는 우수한 교육과정으로 최고의 자격 기준을 갖춘 고급인력을 양성하여야 하며 소프트웨어 안전성분석 업무를 위한 자격기준을 마련해야 한다.

4-2 신규 법제도 제정항목

신규 법·제도 제정항목에는 소프트웨어 평가 및 인증 의무화, 소프트웨어 생명주기에 다른 지속적 평가, 표준화된 개발방법론 도입 의무화, 고급인력 양성 제도 강화 등이 있다.

1) 소프트웨어 평가·인증 의무화

소프트웨어 평가·인증 의무화는 국가에서 인정하는 평가·인증을 보급화하기 위한 방안으로 통일성 있는 평가기준을 마련하고, 평가기준에 따라 인증을 차등으로 등급화시켜 등급별로 우선구매제를 도입하기 위한 것이다.

표 7과 같이 국가에서 수립해야 하는 평가기준을 마련하여 소프트웨어를 평가할 때 적용해야 한다.

표 7. 소프트웨어 평가기준
Table 7. The appraisal standard of software

Level	평가 특성	오류발생률(%)	위험도
VG	매우 안전	0이상~0.01미만	0
G	안전	0.01~0.1	1
N	보통	0.1~1	2
B	위험	1~10	3
VB	매우 위험	10~100	4

[오류발생률=오류량/전체코드라인]

소프트웨어 평가기준은 오류발생률과 위험도에 따라 5레벨로 나누어진다. 오류발생률은 오류량을 전체코드라인으로 나눈 값이다. 위험도는 매우 안전한 상태를 나타내는 0부터 매우 위험한 상태를 나타내는 4까지로 구분된다.

소프트웨어 평가기준의 5레벨을 구별하는 기준은 다음과 같다.

레벨VG(Very Good)는 매우 안전한 소프트웨어의 상태를 의미하며 오류발생률은 0.01%미만이다. 레벨G(Good)는 안전한 소프트웨어의 상태로 오류발생률은 0.01%~0.1%이다. 레벨N(Normal)은 보통의 소프트웨어의 상태로 오류발생률은 0.1%~1%이다. 레벨B(Bad)는 위험한 소프트웨어의 상태로 오류발생률은 1%~10%이다. 레벨VB(Very Bad)는 안전한 소프트웨어의 상태로 오류발생률은 10%~100%이다. 위험도가 레벨VG의 0부터 레벨VB의 4까지로 위험도와 오류발생률에 따라 평가 특성이 설정된다.

소프트웨어 평가기관은 국가에서 수립한 평가기준으로 소프트웨어의 안전성을 분석하여 소프트웨어를 평가해야 한다. 평가기관에서 평가한 소프트웨어의 레벨 및 분석 자료는 인증기관으로 전달되며 인증기관에서 인증 등급화를 위한 기초 자료로 활용된다.

우선구매제는 모든 소프트웨어가 평가·인증을 받도록 장려하기 위한 제도이다. 우선구매제를 등급화하는 이유는 소프트웨어 개발 회사로 하여금 더욱 안전성 있는 소프트웨어를 만들도록 하기 위함이다.

소프트웨어는 소프트웨어 평가기관으로부터 평가레벨에 대한 기초자료를 인증기관에 전달하면 인증기관에서는 평가 자료를 토대로 소프트웨어의 인증등급을 결정한다. 높은 등급을 받은 소프트웨어는 동일한 기능을 가지고 있는 낮은 등급의 소프트웨어보다 우선적으로 구매될 수 있다.

평가기관이 공정하고 정확하게 평가를 진행할 수 있도록 하는 평가기관의 계층화가 필요하다. 이를 위해 평가기관 선별의 투명성과 평가기관을 평가할 수 있는 상위 기관에 대한 기준이 마련되어야 한다. 평가기관의 등급화는 국가기관에 의해 결정된다. 이 기준은 평가기관으로 신청한 기관의 조직과 인력 및 환경조건에 따라 계층적으로 적용할 수 있다.

2) 소프트웨어 생명주기에 따른 지속적 평가

소프트웨어 생명주기 테스팅 과정에서 최적의 임계시점을 찾아서 소프트웨어 안전성분석을 평가해야 한다. 최적의 임계시점은 소프트웨어 분야에 따라 다르게 적용될 수 있으며, 최적의 임계시점은 비용과 시간의 상충관계로 결정되어질 수 있다. 테스트 횟수에 따른 최적의 임계시점에 대한 선택은 그림 4에서 보여준다.

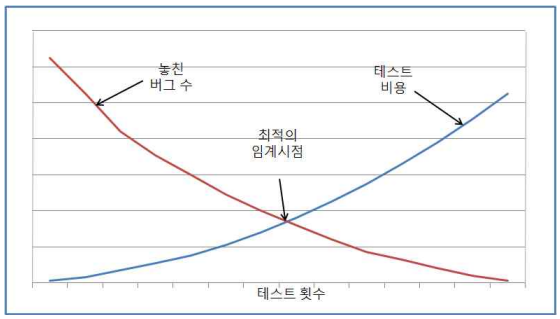


그림 4. 테스트 횟수에 따른 최적의 임계시점
Fig. 4. The optimal critical stage by the number of tests

테스트 횟수에 따른 최적의 임계시점은 전체 소프트웨어 개발규모와 규모에 따른 오류율로 결정된다. 전체 소프트웨어 개발규모는 소프트웨어 생명주기와 투입되는 인력 및 비용으로 책정된다. 오류율은 소프트웨어 안전성 분석을 통해 예측가능하다.

소프트웨어 생명주기 동안에 발생하는 오류는 많다. 취약성이란 소프트웨어의 결함으로 소프트웨어가 출시된 후 발견될 수 있으며 보안을 고려하지 않은 소프트웨어에서 자주 발생된다. 소프트웨어의 안전성을 평가함으로써 취약성에 대해 대비할 수 있지만 한 번의 평가로 모든 오류를 찾아낼 수는 없다. 그

래서 주기적인 평가를 통해 소프트웨어의 안전성을 강화시켜야 한다.

3) 표준화된 개발방법론 도입 의무화

개발 시작단계에서부터 보안을 고려하여 개발되는 안전한 프로그래밍을 위한 보안 코딩이 필요하다. 소프트웨어의 보안을 고려하지 않고 개발된 소프트웨어는 출시된 이후에도 많은 위험에 노출된다. 따라서 개발단계부터 보안을 고려하는 표준화된 개발 방법론을 도입하여 발생 가능한 소프트웨어의 취약점을 사전에 차단해야 한다.

소프트웨어를 개발하기 위해 개발자들은 여러 가지 상황에 놓일 수 있다. 여러 가지 상황에 대한 예를 들면, 소프트웨어 개발기간이 단축되거나, 투입되는 인력이 감소되거나, 한정된 소프트웨어 개발비용으로 많은 요구사항을 충족시켜야 하는 경우 등이 있다. 급변하는 소프트웨어 개발환경에 대비하기 위한 소프트웨어를 각 상황별 개발방법론을 위한 가이드라인을 규정해야 한다.

4) 고급인력 양성 제도 강화

소프트웨어 개발자는 소프트웨어의 안전성 관련 모든 사항을 관리하고 분석할 수 있는 능력을 갖추어야 한다. 따라서 소프트웨어 개발 프로젝트를 진행할 수 있는 고급 자격증을 신설하여 소프트웨어 안전성 관리능력을 검증할 수 있어야 한다. 또한 자격증 보유자의 도덕적 해이로 인한 프로젝트 관리 부실이 발견되는 경우, 자격증 발급기관에서 해당자격증의 효력을 상실시킬 수 있도록 해야 한다.

평가기관의 인력에 대한 자격요건의 기준을 수립하는 것이 필요하다. 평가 전문가는 평가 업무를 수행하는 사람으로, 소프트웨어를 개발하는 전체 과정을 총괄해야 한다. 평가 전문가는 프로젝트 관리 지식과 소프트웨어 전문 지식을 보유해야 한다. 또한, 평가 전문가를 선별한 기관은 주기적으로 전문가 테스트를 통해 평가 전문가 자격을 검증할 수 있어야 한다.

4-3 신규 법제도 제정전략

소프트웨어 안전성분석 강화를 위한 제정전략은 그림 5와 같다.

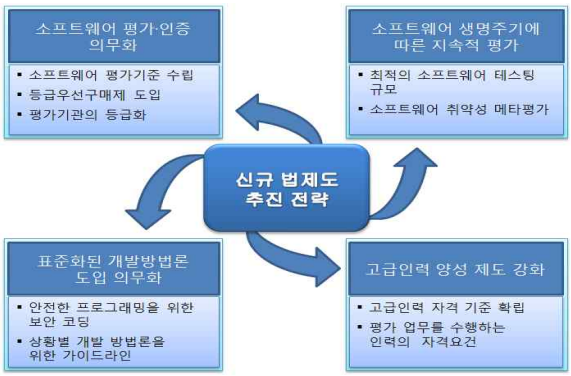


그림 5. 소프트웨어 안전성분석 강화 제정전략
Fig. 5. The establishment strategy to strengthen the safety analysis of software

소프트웨어 평가·인증 의무화는 모든 소프트웨어가 안전성을 평가인증을 받아야 하는 제도로 국가에서 인정하는 평가·인증을 보급화하기 위한 것이다. 소프트웨어 생명주기에 따른 지속적 평가는 생명주기마다 안전성을 인증할 수 있는 평가 및 인증평가 후의 재평가를 통해 소프트웨어가 소멸되기까지 전체 과정에서 평가가 이루어져야 한다. 표준화된 개발방법론 도입 의무화는 안전한 소프트웨어 개발을 위한 지침을 마련하여 의무적으로 사용하도록 하기 위한 것이다. 고급인력 양성 제도 강화는 개발자로 인한 오류를 감소시켜 주며 고급 소프트웨어 개발역량을 강화시킨다. 따라서 이들은 소프트웨어 안전성분석을 강화하기 위한 전략이다.

Ⅴ. 결 론

본 논문에서는 항공사고의 원인으로 항공기 결함에 대해 살펴보고 항공기 엔진을 동작시키는 소프트웨어와 관련된 안전성 법·제도에 대해 살펴보았다. 그리고 소프트웨어의 결함을 해결하기 위해 기존 법·제도를 기반으로 소프트웨어 안전성분석 관련 법·제도 개선과 신규 법·제도 제정 방안을 제안하였다.

제안하는 안전성분석 관련 법·제도 개선 방안으로 소프트웨어산업진흥법에 안전성관련 용어 정립, 품

질인증특성에 소프트웨어 안전성 추가, 품질인증 신청 제출서류에 소프트웨어 안전성분석서 추가, 평가 및 인증기관 세부 지침에 대한 개정 내용을 제시하였다. 신규 법·제도 제정 방안으로는 소프트웨어 평가 및 인증 의무화, 소프트웨어 생명주기에 다른 지속적 평가, 표준화된 개발방법론 도입 의무화, 고급인력 양성 제도 강화 등의 항목으로 제정전략을 제시하였다.

따라서 소프트웨어 안전성분석과 관련된 용어 정립은 안전성관련 개념의 이해 및 습득의 효과, 안전성이 포함된 품질인증 기준은 소프트웨어의 강건성 강화 및 품질 향상의 효과, 안전성분석서를 첨부한 품질인증 신청은 소프트웨어 오류 및 사고율을 감소시키고 품질 인증에 대한 신뢰성 향상의 효과, 평가 및 인증기관 세부 지침 개정은 소프트웨어분석 체계를 갖추고 소프트웨어 개발 능력을 향상하는 효과를 볼 수 있다.

기존 법·제도 개선과 신규 법·제도 제정으로 소프트웨어 산업의 안전성을 확보하여 항공산업뿐만 아닌 소프트웨어를 원천으로 사용하고 있는 타 산업에 안전하게 관리 및 유지할 수 있는 기반을 마련하여 신뢰성을 확보해야 한다.

참 고 문 헌

[1] Darr, S. Ricks, W. Lemos, K.A, “Safer systems: A NextGen aviation safety strategic goal,” *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th,* pp. 2.A.1-1 - 2.A.1-8, Oct 2008.

[2] *국가사이버안전센터*, “소프트웨어 안전성 분석 기술,” 2008.

[3] *국토해양부*, “항공안전정보,” 2011. 7.

[4] 이강이, 한상호, 진영권, 이상준, 김귀순, “항공기 엔진제어시스템 인증기술 개발,” *한국항공우주학회지, 제33권 제1호*, pp. 104-109, 2005. 1.

[5] Briones, J. Fernandez, de Miguel, M. Silva, J.P. Alonso, “Integration of Safety Analysis and Software Development Methods,” *The 1st Institution of Engineering and Technology International Conference on, System Safety,* pp. 275-284, June 2006.

[6] *지식경제부*, “소프트웨어법령,” 2011.

[7] *지식경제부 기술표준원*, “기술표준백서,” 2008.

[8] Dusart, P. Sauveron, D, “Which Trust Can Be Expected of the Common Criteria Certification at End-User Level?,” *Future Generation Communication and Networking (FGCN 2007)*, pp. 423-428, Dec 2007.

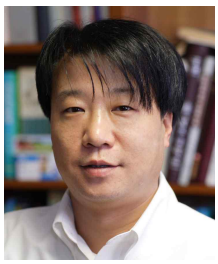
지 정 은 (池汀垠)

2001년 2월 : 수원대학교 전자계산학과(이학사)
2009년 2월 : 숭실대학교 정보과학대학원 정보통신학과 (공학석사)
2009년 3월~현재 : 숭실대학교 컴퓨터학과 박사과정
관심분야 : 네트워크 보안, 정보보호, 차세대인터넷기술

이 상 지 (李尙知)

2011년 2월 : 안양대학교 컴퓨터공학과(공학사)
2011년 3월~현재 : 숭실대학교 컴퓨터학과 석사과정
관심분야 : 네트워크 보안, 인터넷 보안

신 용 태 (愼鏞台)



1985년 2월 : 한양대학교 산업공학과 (공학사)
1990년 12월 : University of Iowa, Computer Science(이학석사)
1994년 5월 : University of Iowa, Computer Science(이학박사)
1995년 3월~현재 : 숭실대학교 컴퓨터학부 교수
2009년 6월~현재 : 한국인터넷진흥원(KISA) 이사
2009년 8월~현재 : 정보통신산업진흥원(NIPA) 이사
관심분야 : 멀티캐스트, 센서네트워크, 컨텐츠보안, 모바일 인터넷, 차세대인터넷기술, 정보보호