The Transactions of the Korea Information Processing Society
TKIPS Vol.13, No.6, pp.278~283

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.6.278

A Study on the Analysis of Internal and External Factors of Software Threat Elements

Lee Eun Ser†

ABSTRACT

When implementing software, there can be side effects that pose a threat to human life. Therefore, it is necessary to measure the impact of software on safety and create alternatives to mitigate and prevent threats. To conduct a software safety assessment to measure the impact of threat factors, the following components are necessary. This paper aims to classify the threat factors of software into internal and external factors and quantitatively demonstrate the impact of these threat factors.

Keywords : Software Test, Reliability, Software Safety

소프트웨어 위협 요소의 내부적·외부적 요인 분석에 관한 연구

이 은 서†

요 약

소프트웨어를 구현할 때 인간의 생명에 위협이 될 수 있는 부작용이 발생할 수 있습니다. 그러므로 소프트웨어가 안전에 미치는 영향을 측정하고 위협을 완화하고 예방하기 위한 대안을 만드는 것이 필요합니다. 위협 요소에 의한 영향도 측정을 위하여 소프트웨어 안전 진단을 수행하기 위한 구성요소가 필요하다. 본 논문은 소프트웨어의 위협 요인을 내부 요인과 외부 요인으로 분류하고 이러한 위협 요인의 영향을 정량적으로 시연하는 것을 목표로 합니다.

키워드 : 소프트웨어 테스트, 신뢰성, 소프트웨어 안전성

1. 서 론

소프트웨어의 사용 분야는 나날이 증가하고 있다. 소프트웨어는 시스템을 구축하여 기능을 수행하기 위한 구현의 결과물로서 그 내용은 기능 수행과 제어를 목적으로 구현이 된다. 이와 같은 목적을 완전히 수행하기 위하여 소프트웨어 구현 시에 완성도와 신뢰성의 필요성이 요구된다. 소프트웨어의 신뢰성과 완성도 향상은 내포하고 있는 결함을 제거하는 것도 포함된다[1,2].

그러나 결함을 제거는 요구사항 대비 기능의 구현을 차이 없이 만들어서 기능 사용의 만족도를 높이는 것이다. 구현의 완성도가 높은 것은 요구사항의 만족도를 높이지만 구현된 소프트웨어 사용에서 발생하는 문제는 구현 시에 예측이 불가능할 수 있다. 이와 같은 이유는 개발자가 구현되는 소프트웨어가 사용되는 영역의 지식이 부족하거나 개발자의 숙련도가 떨어져서 발생하게 된다[3-5].

소프트웨어가 구현되고 사용하는 과정에서 발생하는 문제를 미리 예측하여 사용상의 안전을 제공하는 것이 요구되고 있다.

본 논문에서는 소프트웨어 안전을 측정하여 위험을 확인하기 위하여 정량적인 평가를 내부적인 요인과 외부적인 요인으로 구분하여 제시하고 한다.

2. 기반 연구

2.1 시스템 안전 시스템

시스템의 안전한 사용을 위하여 안전 시스템들이 존재한다. 이와 같은 시스템들은 시스템 안전에 문제를 인지하여 소프트웨어의 문제가 시스템의 전반적인 안전을 위협할 수 있으므로 위험 요소별로 원인을 파악하고 있다. 위험 요소의 원인은 소프트웨어 기능이 예상한 것과 다르게 수행되는 경우가 있다[7-9]. 또한 위험 요소가 생명을 위협하는 경우 소프트웨

※ 이 논문은 국립안동대학교 기본연구지원사업에 의하여 연구되었음.
† 종신회원 : 국립안동대학교 컴퓨터공학과 교수
Manuscript Received : April 30, 2024
Accepted : May 14, 2024
* Corresponding Author : Lee Eun Ser(eslee@anu.ac.kr)

어 기능이 대응을 제대로 하지 않는 경우이다. 그리고 위험 상황이 발생 된 경우 소프트웨어가 해결하지 못하고 정상적인 상태로 복원할 수 없는 경우이다. 마지막으로 사고 피해를 최소화하기 위한 소프트웨어 기능이 작동되지 않는 경우이다. 위험 원인과 밀접한 소프트웨어 품질 요소로는 기능성, 정확성, 가용성, 신뢰성, 견고성, 안전성, 보안성 등이 있다[10,11].

소프트웨어가 시스템 오동작 및 위험에 미칠 수 있는 요인으로는 이벤트/액션, 로직/알고리즘 오동작, 통신 및 입출력, 타이밍, 사용자 오류/사용자 인터페이스 오류 등이 있다[12,13].

2.2 소프트웨어 안전 진단

요구사항을 기능화하여 소프트웨어를 구현한 후에 요구사항이 잘 구현되었는지 단위 테스트를 활용하여 기능을 시험한다. 소프트웨어 품질 안전성 진단은 구현물을 확인하는 것과 동작이 요구사항 대비 잘 작동하는지를 확인하는 것이다. 소스코드 진단은 메모리 반환 오류, 잘못된 연산 구문의 오류를 확인하여 소프트웨어의 결함을 찾는 것이다. 동작의 정확성은 기능과 요구사항을 비교하여 일치하는가를 판단한다. 그리고 사용되는 데이터가 잘못된 경우와 빠진 경우, 입력 값의 오류 등을 진단하게 된다[14].

소프트웨어 안전 진단은 개발과 구동 환경을 고려해야 한다. 사용되는 서버와 데이터베이스, 미들웨어와 이외에 하드웨어, 응용 소프트웨어, 운영체제, 도구를 진단해야 한다. 소프트웨의 진단은 개발 환경과 구동환경이 완전한 상태에서 구현된 소프트웨어의 안전성을 진단해야 한다. 환경이 완전하지 않은 상태에서 구현된 소프트웨어의 안전성을 진단하면 오류가 발생되어도 구현의 문제인지 환경의 문제인지 판별이 어렵게 된다[16]. 따라서 개발 전에 시스템 아키텍처의 신뢰성과 안전성을 먼저 확인해야 한다.

본 논문에서는 소프트웨어의 안전을 확인하기 위하여 정량적인 기준을 제시하고 이를 확인하고자 한다.

3. 연구 방법

소프트웨어는 많은 분야에서 활용이 되고 있으며 작업과 신뢰도 측면에서 중요한 부분을 차지하여 필수요소가 되고 있다. 소프트웨어를 사용하면서 예상하지 못한 결과를 초래하게 되면 단순한 문제에서부터 사람의 생명을 위협하는 요소가 발생될 수 있다. 3장에서는 소프트웨어를 사용 시에 있어서 발생할 수 있는 위험 요소를 측정하여 위험 정도를 평가하고자 한다. 이와 같은 것은 소프트웨어 안전을 위하여 필요한 요소가 되며 이를 활용하기 위하여 다음과 같은 안전을 위협하는 단계(STP : Safety Threaten Phase, 이하 STP로 정의)를 제시한다.

소프트웨어 위험 요소를 측정하기 위하여 STP와 위험 요소를 내부, 외부 요소로 구분하여 분석을 한다. STP는 상, 중 하로 구분하였고 위험 요소의 내부, 외부 요인을 제시하였다.

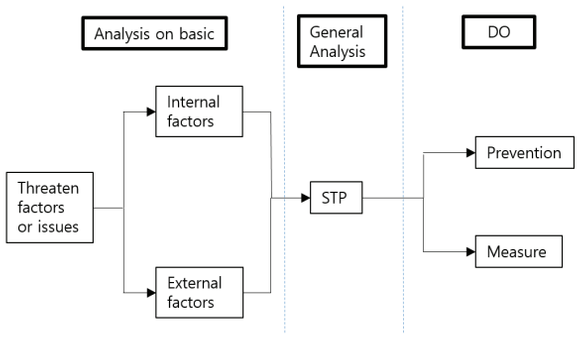

Fig. 1. Measurement Process of SW Risk

전체 구조도는 다음 Fig. 1과 같다.

소프트웨어의 기능이 내포하고 있는 위협 요소나 이슈를 내부, 외부적인 요소로 영향도를 분석하여 STP의 3단계 중에서 해당되는 단계를 추출하여 전체적인 소프트웨어 위험도에 따른 위협 정도를 파악한다. 이후에는 위협 정도에 따라서 예방을 통하여 위협 요소가 발생되는 것을 막을 수 있고 위협 요소가 발생된 경우에는 처리를 하기 위하여 전체 구조도를 제시하였다. 구조도에서는 소프트웨어 기능과 연관된 내포된 내부, 외부의 위험요소를 분석하기 위한 기초적인 특성 분석 단계와 STP를 기반으로 전체적으로 위협이 되는 사항이 발생되는 것을 측정하기 위한 전반적인 분석단계, 마지막으로 분석된 결과물을 통하여 예방과 처리를 할 수 있는 실행 단계의 세 부분으로 구성되어 있다.

STP는 오류나 결함과는 다른 형태를 가지고 있다. 개발단계 관점에서의 위험은 일정, 품질, 비용에서 예상하지 못한 문제로 프로젝트가 실패되는 경우를 의미한다. 따라서 소프트웨어에 잠재되어 있은 많은 요소들 중에서 긍정적인 요소 이외에 부정적인 요소로 인하여 시스템 장애 뿐만이 아니라 사람의 생명까지 위협하게 된다. 위험 단계가 발생하면 이미 소프트웨어 안전을 위협하는 단계로 들어서게 된다. 따라서 위험 요소로 전이되기 전에 소프트웨어 안전을 위협하는 요소를 찾아서 전이되지 않도록 예방하는 것이 중요한 요소가 된다.

소프트웨어를 사용하는 분야는 너무 방대하여 공통적인 소프트웨어 안전 요소를 정의하여 안전을 위협하는 단계를 파악하여 예방하기는 어렵다. 따라서 이를 기반으로 소프트웨어 안전관리 방안과 분야별로 위협 요소를 제시하여 관리할 필요가 있다.

소프트웨어 안전을 위협하는 분야를 선정하여 위험 정도를 분석하는 방법과 이를 기반으로 영향도를 측정하여 안전을 위협하는 요소를 예방하는데 활용할 수 있다.

위험도 측정은 내부적인 요인과 외부적인 요인으로 구분하여 분석될 수 있다.

1) 내부적인 요소

위험도 측정을 위한 내부적인 요소는 소프트웨어 안전과

연관하여 개발 과정에서 발생하는 위협 요소를 추출하였다. 위험도 측정을 위한 위협요인의 내부적인 요소로는 오동작, 잘못된 프로그래밍, 소프트웨어 개발 프로세스 미비, 시험 프로세스 미비로 제시하였다.

① 오동작

소프트웨어에서 정의된 기능과 다른 동작을 수행하는 경우이다. 이와 같은 내용은 정상적인 과정을 거치지 않고 기대하지 않은 기능이 수행되거나 일관성 있고 지속적인 기능 수행을 하지 않은 경우이다. 오동작의 정도에 따라서 상, 중, 하로 구분하였다. “상”은 오동작에 의하여 지속적인 동작이 불가능한 경우이다. “중”은 오동작에 의하여 기능에 문제가 생기지만 일정시간의 수정시간을 통하여 정상 작동이 가능한 경우이다. “하“는 단순한 기능 문제로 즉시 해결이 가능한 경우이다. 오동작은 M(Misoperation)이라고 지칭한다.

② 잘못된 프로그래밍

개발자가 설계의 문서를 잘못 이해해서 프로그래밍하는 경우이다. 또한 프로그래머의 미숙한 스킬에 의하여 소스코드가 잘못 구현된 경우이다. “상”은 잘못된 프로그래밍에 의하여 시스템의 동작이 불가능한 경우이다. “중”은 일정시간의 수정시간을 통하여 정상 작동이 가능한 경우이다. “하”는 단순한 기능 문제로 즉시 해결이 가능한 경우이다. 잘못된 프로그래밍은 BP(Bad Programming)라고 지칭한다.

③ 소프트웨어 개발 프로세스

소프트웨어를 개발하는 과정에서 적용하는 생명주기, 방법론의 부재로 인하여 위험 요소가 발생하는 경우이다. 개발 과정에서 발생하는 산출물이 정의되지 않고 일정 지연과 품질의 하락이 발생되는 경우이다. “상”은 주요 프로세스가 2개미만으로 구축된 경우이다. “중”은 생명주기, 개발 방법론 등의 주요한 프로세스가 2개 -4개 구축이 되어 있는 경우이다. “하”는 소프트웨어 개발시에 모든 표준화된 프로세스가 구축되어 있는 경우이다. 소프트웨어 개발 프로세스는 SDP(Software Development Process)라고 지칭한다.

④ 시험 프로세스

요구사항에서 정의된 기능을 검사하기 위한 시험 프로세스가 없은 경우이다. 또한 요구사항과 관련된 문서가 없어서 기능 시험을 하지 못하는 경우이다. “상”은 시험프로세스 내용이 일관성이 없고 정의되어 있지 않는 경우이다. “중”은 시험 프로세스 내용이 정성적, 정량적으로 정의되어 있는 경우이다. “하”는 시험 프로세스의 모든 부분이 정량적으로 정의되어 있는 경우이다. 시험 프로세스는 TP(Test Process)라고 지칭한다.

내부적인 위협 요소를 정량적인 수치로 표현하기 위하여 요소별로 위협도의 점수를 평균 값(25점)으로 제시하였다. 가중치의 값은 정도를 기반으로 “상” 위협 요소는 차이를 많이 주었고 나머지 “중”, “하”는 범위의 평균으로 제시하였다. 가중치 값에 대해서는 적용하는 영역과 시스템, 개발되는 기능에 따라서 가중치를 변경해야 한다. 요구사항의 내부적인 위협 요소를 측정하기 위하여 요구사항의 복잡도를 위협 요소 연산에서 고려를 한다. 이를 I라고 표기한다. I는 0.1-1까지 표현되며 1에 가까울수록 복잡도가 높은 것을 의미한다.

Table 1. Table of Internal Threaten Factors

Items(I)	Wi		
	High	Middle	Low
M (25 Point)	1	0.5	0.25
BP (25 Point)	1	0.5	0.25
SDP (25 Point)	1	0.5	0.25
TP (25 Point)	1	0.5	0.25

2) 외부적인 요소

위험도 측정을 위한 외부적인 요소는 소프트웨어 안전과 연관하여 개발 과정 이외의 외부 환경에서 발생하고 영향을 주는 위협 요소를 추출하였다. 위험도 측정을 위한 위협요인의 외부적인 요소로는 사이버 공격, 사용상의 법적 규제, 데이터 무결성, 정보 안전성, 소프트웨의 악용으로 제시하였다.

① 사이버 공격

외부 침입에 의해서 소프트웨어의 작동을 방해한다던지 다른 기능을 수행하게 하는 경우이다. 또한 민감한 정보를 빼내고 잠재적으로 오류를 발생시키기 위하여 랜섬웨어와 같은 소프트웨어를 설치하는 경우이다. “상”은 사이버 공격에 의해서 시스템이 복구 불가능한 경우이다 “중”은 일정 시간이 소요되지만 복구가 가능한 경우이다. “하”는 사이버 공격이 시스템에서 즉시 복구가 되거나 영향이 없는 경우이다. 사이버 공격은 CA(Cyber Attack)라고 지칭한다.

② 데이터 무결성

소프트웨어의 기능 수행은 데이터를 기반으로 작동된다. 따라서 원본이 훼손되지 않은 데이터의 제공은 소프트웨어 안정적인 실행을 위하여 필수요소가 된다. 데이터는 외부로부터 제공받기 때문에 외부적인 요소가 된다. “상”은 외부 데이터를 가져올 때 데이터가 무결하지 않는 경우 시스템과 기능이 수행되지 않는 경우이다. “중”은 손상된 데이터를 구별하여 찾을 수 있으며 일정 시간이 소요되더라도 수정을 하여 시스템과 기능을 정상적으로 수행할 수 있는 경우이다. “하”는 데이터 무결성이 있더라도 즉시 복구가 되거나 영향이 없는 경우이다. 데이터 무결성은 DI(Data Integrity)라고 지칭한다.

③ 정보 안전성

소프트웨어는 기본적인 기능과 데이터, 안전적인 정보를 기반으로 수행되게 된다. 따라서 정보를 접근할 수 있는 권한을 분류하고 제공하는 것이 필수요소가 된다. 정보 안전성은 정보와 시스템 등에 접근할 수 있는 권한과 책임에서 발생되는 외부 위협 요소이다. “상”은 시험프로세스 내용이 일관성이 없고 정의되어 있지 않는 경우이다. “중”은 시험 프로세스

Table 2. Table of External Threaten Factors

Items(E)	We		
	High	Middle	Low
CA (25 Point)	1	0.5	0.25
DI (25 Point)	1	0.5	0.25
IS (25 Point)	1	0.5	0.25
SE (25 Point)	1	0.5	0.25

내용이 정성적, 정량적으로 정의되어 있는 경우이다. “하”는 시험 프로세스의 모든 부분이 정량적으로 정의되어 있는 경우이다. 정보 안전성은 IS(Information Safety)라고 지칭한다.

④ 소프트웨어 악용

의도적으로 소프트웨어의 정의된 기능을 목적 이외에 사용하여 위협요소를 발생하는 경우이다. 기능의 악용과 함께 정의된 환경 이외에서 의도적으로 사용하는 경우도 해당된다. “상” 의도적으로 소프트웨어를 목적이외에 사용하여 시스템을 망가뜨리는 경우이다. 이 경우에는 복구가 불가능하게 된다. “중”은 소프트웨어 악용을 감지하여 일정 시간을 소요하여 시스템을 복구할 수 있는 경우이다. “하”는 소프트웨어 악용이 경미하거나 즉시 수정이 가능하여 시스템에 영향이 없는 경우이다. 소프트웨어 악용은 SE(Software Exploit)라고 지칭한다.

외부적인 위협 요소를 정량적인 수치로 표현하기 위하여 요소별로 위협도의 점수를 평균 값(25점)으로 제시하였다. 가중치의 값은 정도를 기반으로 ”상” 위협 요소는 차이를 많이 주었고 나머지 “중”, “하”는 범위의 평균으로 제시하였다. 가중치 값에 대해서는 적용하는 영역과 시스템, 개발되는 기능에 따라서 가중치를 변경해야 한다. 요구사항의 외부적인 위협 요소를 측정하기 위하여 요구사항의 복잡도를 위협 요소 연산에서 고려를 한다. 이를 E라고 표기한다. E는 0.1-1까지 표현되며 1에 가까울수록 복잡도가 높은 것을 의미한다.

내부적 · 외부적인 위협 요소를 분석하여 정량적으로 산출한 후에는 전반적인 소프트웨어 위협 단계를 산출해야 한다.

STP는 소프트웨어의 기능개발 완료된 이후에 서비스 형태로 제공되는 과정에서 발생할 수 있는 단계를 기준으로 위험요소를 발견하고 위험도를 측정하는 기준을 제시한다.

안전을 위협하는 단계는 각 단계를 측정하기 위하여 정량적 또는 정성적 기준을 제시한다.

안전을 위협하는 단계는 3단계로 제시하였으면 다음과 같다.

위험 (상) : 사람의 생명을 위협하거나 영향을 주는 단계

위험 (중) : 전체 시스템이 멈추는 단계

위험 (하) : 시스템의 일부가 멈추거나 장애를 일으키는 단계

내부적인 위협요소 수치(I X Wi)와 외부적인 위협요소 수치(E X We)의 결과 값은 전반적인 STP의 수치를 산출하기 위한 기초 자료가 된다. STP를 산출하기 위한 식은 다음 Equation (1)과 같다. (STP는 1을 초과할 수 없음)

Table 3. Weight of STP

Items(O)	Ws		
	High	Middle	Low
Threaten Value of Interna and External	1	0.5	0.25

$$STP = ((\sum_{i=1}^{4} I X W_i/4 + \sum_{e=1}^{4} E X W_e/4)/2) X W_s \quad (1)$$

4. 적용 및 사례

3장에서 제시한 이론을 적용하기 위하여 비닐하우스 자동관리 시스템을 선정하였다. 개발 시스템에서 기능을 추출하여 내부적인 · 외부적인 위협 요소를 추출하였다. 또한 요구사항을 기능적인 요구사항과 비기능적인 요구사항으로 분류하였다.

STP를 적용하기 위하여 기능적 요구사항과 비기능적 요구사항에서 일부 요구사항을 선정하였다. 기능적 요구사항으로는 온습도 기능 제어 판단(R-0001), 미세먼지 수치 판단(R-0002), 온습도 모니터링(R-0003)을 선정하였다. 비기능적 요소에서는 인터넷 연결(NR-0001), 하드웨어 구동 및 연결(NR-0002)을 선정하였다. 선정한 기능적 요구사항의 설명은 다음과 같다.

온습도 기능 제어(R-0001)는 비닐하우스의 각 기능 제어 여부를 판단한다. 자동제어 시 적절 온습도와 비닐하우스의 온습도를 비교하여 각 기능 작동여부를 판단한다. 수동제어 시 어플리케이션에서 선택한 기능의 작동여부에 따라 기능을 제어한다.

미세먼지 수치 판단(R-0002)은 미세먼지 수치를 판단한다. 미세먼지 수치를 측정하여 장치 제어 여부를 판단한다.

온습도 모니터링(R-0003)은 비닐하우스의 온습도 상태를 모니터링한다. 관리자는 온습도를 측정하여 제어하기 위해서 어플리케이션을 이용하여 비닐하우스의 온습도를 모니터링한다.

선정한 기능적 요구사항의 설명은 다음과 같다.

인터넷 연결(NR-0001)은 라즈베리파이가 서버와 통신할 수 있도록 조치한다. 라즈베리파이와 어플리케이션 간의 데이터를 원활히 송수신 할 수 있도록 인터넷 연결 환경을 구축한다.

하드웨어 구동 및 연결(NR-0002)은 라즈베리파이의 장치들 간의 연결을 원활하게 한다. 각 센서들을 통해 수치를 측정하고 각 기능이 동작할 수 있도록 하드웨어 간의 연결을 확인한다.

추출된 요구사항은 전체 요구사항 중에서 일부를 선정하였으며 요구사항 정의와 타당성을 수행하여 클래스 다이어 그램을 작성하였다. 클래스 다이어그램은 다음 Fig. 2와 같다.

STP를 산정하기 위하여 내부적인 위협요소와 외부적인 위협요소를 요구사항별로 산정하였다.

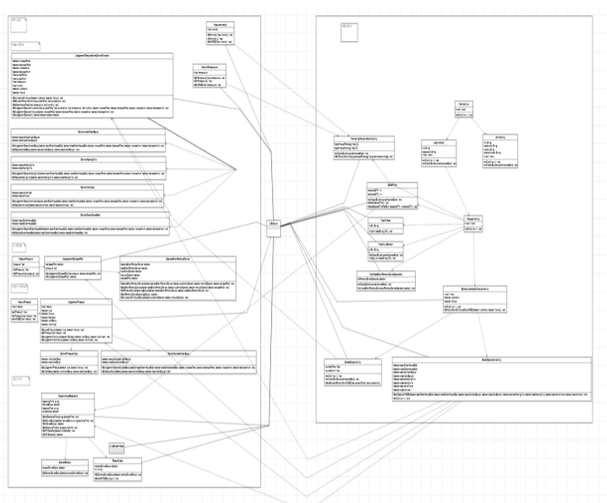

Fig. 2. Class Diagram

Table 4. Computation of Internal Threaten Factors

	Internal Threaten factors					Result
	I	M	BP	SDP	TP	
R-0001	0.7	1	0.5	0.25	0.5	0.39375
R-0002	0.2	0.25	0.25	0.25	0.25	0.1421875
R-0003	0.7	0.5	0.5	0.25	0.5	0.39375
NR-0001	0.2	0.5	0.25	1	1	0.1375
NR-0002	0.5	0.5	0.25	0.5	0.5	0.21875

내부적인 위협요소의 산정 내용은 Table 4와 같다. 산정을 위한 세부 사항은 다음과 같다.

R-0001의 내부적인 위협요소는 다음과 같다. 0.7 x 1 + 0.7 x 0.5 + 0.7 x 0.25 + 0.7 x 0.5 = 1.575 따라서 내부적인 위협요소는 1.575/4 = 0.39375로 산정되었다.

R-0002의 내부적인 위협요소는 다음과 같다. 0.2 x 0.25 + 0.2 x 0.25 + 0.2 x 0.25 + 0.2 x 0.25 = 0.2 따라서 내부적인 위협요소는 0.56875/4 = 0.1421875로 산정되었다.

R-0003의 내부적인 위협요소는 다음과 같다. 0.7 x 0.5 + 0.7 x 0.5 + 0.7 x 0.25 + 0.7 x 0.5 = 1.575 따라서 내부적인 위협요소는 1.925/4 = 0.39375로 산정되었다.

NR-0001의 내부적인 위협요소는 다음과 같다. 0.2 x 0.5 + 0.2 x 0.25 + 0.2 x 1 + 0.2 x 1 = 0.55 따라서 내부적인 위협요소는 0.55/4 = 0.1375로 산정되었다.

NR-0002의 내부적인 위협요소는 다음과 같다. 0.5 x 0.5 + 0.5 x 0.25 + 0.5 x 0.5 + 0.5 x 0.5 = 0.875 따라서 내부적인 위협요소는 0.875/4 = 0.21875로 산정되었다.

외부적인 위협요소의 산정 내용은 Table 5와 같다. 산정을 위한 세부 사항은 다음과 같다.

R-0001의 외부적인 위협요소는 다음과 같다. 0.2 x 0.5 + 0.2 x 0.5 + 0.2 x 0.5 + 0.2 x 0.5 = 0.4 따라서 외부적인 위협요소는 0.4/4 = 0.1로 산정되었다.

Table 5. Computation of External Threaten Factors

	External Threaten factors					Result
	E	CA	DI	IS	SE	
R-0001	0.2	0.5	0.5	0.5	0.5	0.1
R-0002	0.5	0.25	0.25	0.5	0.25	0.15625
R-0003	0.2	0.25	0.5	0.5	0.25	0.075
NR-0001	0.7	1	0.5	0.5	0.5	0.4375
NR-0002	0.7	0.25	0.5	0.25	0.5	0.2625

R-0002의 외부적인 위협요소는 다음과 같다. 0.5 x 0.25 + 0.5 x 0.25 + 0.5 x 0.5 + 0.5 x 0.25 = 0.625 따라서 외부적인 위협요소는 0.625/4 = 0.15625로 산정되었다.

R-0003의 외부적인 위협요소는 다음과 같다. 0.2 x 0.25 + 0.2 x 0.5 + 0.2 x 0.5 + 0.2 x 0.25 = 0.3 따라서 외부적인 위협요소는 0.3/4 = 0.075로 산정되었다.

NR-0001의 외부적인 위협요소는 다음과 같다. 0.7 x 1 + 0.7 x 0.5 + 0.7 x 0.5 + 0.7 x 0.5 = 1.75 따라서 외부적인 위협요소는 1.75/4 = 0.4375로 산정되었다.

NR-0002의 외부적인 위협요소는 다음과 같다. 0.7 x 0.25 + 0.7 x 0.5 + 0.7 x 0.25 + 0.7 x 0.5 = 1.05 따라서 외부적인 위협요소는 1.05/4 = 0.2625로 산정되었다.

요구사항별 STP의 가중치는 다음과 같다.

STP를 산정하기 위하여 내부적인 위협요소와 외부적인 위협요소를 요구사항별로 산정하였다. 산정 내용은 Table 6과 같다. 이를 기준으로 STP를 산정하면 다음과 같다.

R-0001의 STP는 다음과 같다. (0.39375 (내부적인 위협요소) + 0.1 (외부적인 위협요소)) x 0,25 (STP 가중치) = 0.1234375로 산정되었다.

R-0002의 STP는 다음과 같다. (0.1421875 (내부적인 위협요소) + 0.15625 (외부적인 위협요소)) x 0,25 (STP 가중치) = 0.074609375로 산정되었다.

R-0003의 STP는 다음과 같다. (0.39375 (내부적인 위협요소) + 0.075 (외부적인 위협요소)) x 0,25 (STP 가중치) = 0.1171875로 산정되었다.

NR-0001의 STP는 다음과 같다. (0.1375 (내부적인 위협요소) + 0.4375 (외부적인 위협요소)) x 0,5 (STP 가중치) = 0.2875로 산정되었다.

NR-0002의 STP는 다음과 같다. (0.21875 (내부적인 위협요소) + 0.2625 (외부적인 위협요소)) x 0,5 (STP 가중치) = 0.240625로 산정되었다.

5개 요구사항 (3개의 기능적 요구사항과 2개의 비기능적 요구사항)의 STP 결과 값을 산출하였다. 5개 모두 안전을 위협하는 단계가 30% 미만으로 치명적인 위협요인 요구사항으로 분석되지 않았다. 5개 요구사항 중에서 비기능적 요구사항이 기능적 요구사항보다 높은 안전 위협 요소로 판단되었다. 따라서 이를 기반으로 대안을 세워서 프로젝트를 수행할 때 안전을 위협는 요소를 예방하여 전체 시스템이 사람의 생명을 위협하거나 시스템이 멈추는 상황을 방지할 수 있다.

Table 6. Computation of STP

	Value of Internal Threaten	Value of External Threaten	Weight of STP(Ws)	STP
R-0001	0.39375	0.1	0.25	0.1234375
R-0002	0.1421875	0.15625	0.25	0.074609375
R-0003	0.39375	0.075	0.25	0.1171875
NR-0001	0.1375	0.4375	0.5	0.2875
NR-0002	0.21875	0.2625	0.5	0.240625

5. 저작권

출판된 모든 원고는 한국정보처리학회의 자산이 되며, 서면허가 없이 다른 곳에 출판되어서는 안 된다. 출판이 결정되면 저자는 저작권양도 서식을 기재하여 팩스, 우편 또는 이메일로 학회 사무국에 보내야 한다.

6. 결 론

소프트웨어가 사용되는 상황에서 발생할 수 있는 위협 요소를 측정하여 위험의 영향도를 분석할 필요가 있다. 본 연구에서는 이를 위하여 선행적으로 위험을 구성하는 요소를 내부와 외부적인 요인으로 제시하였다. 안전을 위협하는 요소를 내부적인 요소와 외부적인 요소로 정량적인 분석을 하였다. 내부적인 것은 기능과 연관된 개발자 관점이다. 또한 외부적인 요소는 외부환경에 영향을 받는 사항을 고려하였다. 안전 위협 요소를 산정하기 위하여 시스템과 사람의 생명에 위협적인지에 대한 사항을 고려하였다. 이를 정량적인 기준에 의하여 산정하였다.

향후 연구내용으로는 안전을 위협하는 요소는 산업분야와 개발 프로젝트의 특성을 고려하여 내부적인 요소와 외부적인 요소, 안전을 위협하는 영향도에 대하여 카테고리별로 제시하여 공용적인 부분과 가변적인 부분을 구분하여서 안전 위협 요소의 기준을 제시하고자 한다. 또한 내부적 · 외부적인 위협 요소에 대하여 산업분야별로 적용을 하여 검증을 통한 요소의 보정작업을 수행하고자 한다.

References

[1] IEC 61508:1998 Functional safety of electrical · lectronic · programmable electronic safety-related systems.
[2] ISO 26262:2011 Road vehicles – Functional safety.
[3] NASA-GB-8719.13: NASA Software Safety Guidebook, 2004.
[4] NASA-GB-8719.13C Software Safety Standard, 2013.
[5] NASA System Safety Handbook - volume1, 2010.
[6] DoD JOINT Software Systems Safety Engineering Handbook, 2010.
[7] 한국정보통신기술협회(TTA), 소프트웨어 안전 진단 가이드, 2021.
[8] 정보통신산업진흥원(NIPA), SW안전 국제표준화 동향과 시사점, 2019.
[9] NUREG-0800, Standard Review Plan: BTP HICB-14, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, USNRC, 1997.
[10] Beizer, B., Software Testing Techniques (2nd edition), Van Nostrand Reinhold: Boston, 1990.
[11] Gilb, Tom and Graham, Dorothy, "Software Inspection," Addison Wesley: Reading, MA, 1993.
[12] IEEE 14764:2006 Software Life Cycle Processes – Maintenance.
[13] IEEE 1016:1998 IEEE Recommended Practice for Software Design Descriptions.
[14] EIA 12207:1995 Industry Implementation of International Standard.
[15] ANS/ISA S84.01 Application of safety instrumented systems for the process industries, ANS/ISA, 1996.

이 은 서

https://orcid.org/0000-0002-7637-3036
e-mail : eslee@anu.ac.kr
2001년 ~ 현 재 ISO/IEC 15504 국제 선임 심사원
2004년 중앙대학교 컴퓨터공학과(박사)
2008년 ~ 현 재 국립안동대학교 컴퓨터공학과 교수
관심분야 : CBD, Formal method, Quality model, SPI(Defect Analysis)