J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

STPA를 적용한 의료기기 시스템의 안전성 프로세스 가이드라인☆

A Safety Process Guideline of Medical Device System Based on STPA

최 보 윤[1] 이 병 걸[2*]
Bo-yoon Choi Byong-gul Lee

요 약

의료기기의 오작동이나 고장은 인명 피해나 큰 상해가 발생할 수 있으므로 안전성 확보가 필수적인 분야이다. 이를 위해 국제 표준을 제정하고 준수하도록 하고 있지만 이들 표준은 상호의존적이고 특히, 안전성 분석 활동의 경우 시간, 노력, 비용이 많이 소요되어 표준 활동을 통합하고 커스터마이징할 필요가 있다. 따라서 본 논문에서는 의료기기 소프트웨어 개발 단계별 수행 활동과 안전성 프로세스의 활동을 통합하였다. 통합된 프로세스를 통해 하나의 프로세스로 체계적인 관리가 가능하며, STPA 기법을 기반으로 하는 프로세스 단계별로 가이드가 제공되어 안전성 활동을 효과적으로 수행할 수 있다.

☞ 주제어 : 의료기기 소프트웨어, 안전성 프로세스, STPA

ABSTRACT

Malfunctions and failures linked to medical devices may result in significant damage for human being. Thus, in order to ensure that safety of medical device is achieved, it should be established and applied the international standard. It is required to integrate and customize activities at standards, owing to reference relationship between standards, especially, activities based safety analysis is too expensive. This paper proposes a integration process that integrate activities of development lifecycle and safety process. Additionally, we derived a guidance based on STPA for integration process. As a result, we can be performed systematically from early stage of the development and increased effectiveness of integration process by the guidance.

which is to identify hazard for preventing the accidents and minimizing the potential harm

☞ keyword : Medical device Software, Safety process, STPA

1. 서 론

의료기기들은 치료 과정에서 생명이나 건강에 영향을 미칠 수 있어 안전성 확보가 필수적인 분야로 의료기기의 안전성을 강화하여 의료 사고를 예방하고자 하는 시도가 다각도에서 진행 중이다. 이와 같은 노력의 일환으로 의료기기 소프트웨어 안전성을 강화하기 위한 국제 표준을 제정하고 이를 준수하고 입증하는 의료기기 시스템을 인허가하고 있다. 이들 표준은 안전성 확보를 위한 각각의 활동을 제시하면서도 서로의 내용을 참조하고 있기 때문에 이들간의 상호연계성을 효과적으로 관리할 수 있도록 통합 프로세스와 가이드라인이 요구된다. 특히, 안전성 분석 활동의 경우 위험을 발생시키는 상황에 대한 정보가 필요하고 상호의존적인 관계의 표준들에 대한 기반 지식이 필요하다. 하지만 의료기기 시스템의 안전성 분석은 많은 시간과 비용, 노력이 소요되는 활동이며 안전성 분석 경험도 부족하다. 또한, 기존 표준에서는 각 프로세스에 대한 요구사항만을 제시하고 있다는 문제점이 있다. 따라서 개발 프로세스와 안전성 분석 프로세스를 통합하여 지원하고자 하는 연구가 수행중이지만 안전성 분석 경험이 부족한 조직에서는 이를 커스터마이징해서 사용하는데 어려움이 있다.

1 Department of Information Security, Seoul Women's University, Seoul, 01797, Korea.
2 Department of Data Science, Seoul Women's University, Seoul, 01797, Korea.
* Corresponding author (byongl@swu.ac.kr)
[Received 8 November 2021, Reviewed 16 November 2021, Accepted 22 November 2021]
☆ 이 논문은 2019학년도 서울여자대학교 연구년 수혜 지원을 받았음.
☆ 이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임. (No. 2021R1I1A1A01041641)

본 논문에서는 의료기기 소프트웨어의 안전성 평가를 지원하는 표준에서 명시하고 있는 분석 절차 및 활동을 통합하였다. 통합된 프로세스는 안전성 분석 단계별 활동을 개발 초기 단계부터 이후 단계들에서도 적용할 수 있도록 STPA 기법을 통해 수행하도록 한다.

본 논문의 구성은 다음과 같다. 2장에서 의료기기 안전성과 관련된 표준들과 연구 동향을 설명하고 논문에서 활용하는 해저드 분석 기법 중 하나인 STPA 기법을 소개한다. 3장에서 의료기기 안전성 통합 프로세스를 제시하고 STPA 기법을 활용하여 프로세스의 각 활동을 수행하는 방법을 설명한다. 4장에서는 제안하는 방법을 의료기기 시스템에 적용하고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

의료기기 소프트웨어에 대한 안전성을 확보하기 위한 활동과 위험관리 활동을 통합한 프로세스를 제안하는 연구들의 동향을 살펴보고, 의료기기 소프트웨어 안전성과 관련된 표준 및 기법을 소개한다.

2.1 의료기기 시스템의 안전성 프로세스에 대한 연구 동향

Jessyka 등의 연구[4]에서는 안전성 분석 활동과 요구공학 활동의 통합 모델에 대한 연구를 비교 분석하였다. 통합 모델에 대한 57개의 연구를 비교하여 안전성 분석 활동 중 요구 공학 활동을 통해 수행되어야 할 활동들과 안전성 분석 활동을 지원할 수 있는 요구공학의 산출물 및 도구 등을 제시한다. 김영민 등의 연구[5]에서는 시스템의 설계 프로세스와 안전성 활동 프로세스를 통합하는 모델을 정의하였다. 설계 단계에서 안전성 확보의 중요성을 설명하고 SysML을 기반으로 안전성을 반영한 설계 프로세스를 제안하였다. 기존의 설계 단계에서의 안전 보장 활동은 상세설계단계를 중점으로 수행하였으나 이 연구에서는 시스템의 설계 단계 중 첫 번째 단계인 개념 설계 단계부터 수행할 수 있도록 제안하고 있다. 이 두 연구에서는 특정 프로세스를 중심으로 안전성 활동을 식별하고 통합 프로세스를 제시하고 있다.

Pecoraro 등의 연구[6]에서는 의료기기 소프트웨어 생명주기에 위험관리 프로세스를 추가하였다. 의료기기 소프트웨어에 대한 미국과 유럽의 규정, 표준, 지침 등을 비교 분석하여 의료기기 소프트웨어 생명주기에 위험관리를 통합한 프레임워크를 제안하였다. 이 프레임워크는 소프트웨어 개발 각 단계마다 위험 분석을 수행하여 예상하지 못한 시스템 사고와 소프트웨어 위험원을 파악할 수 있도록 한다. 김동엽 등의 연구[7]에서는 의료기기 소프트웨어 개발생명주기를 기반으로 개발 단계별로 위험관리에 필요한 요소를 제시하였다. 개발생명주기와 위험관리 프로세스 요구사항의 연관성 분석 결과를 토대로 의료기기 소프트웨어 개발생명주기와 위험관리 프로세스가 통합한 프레임워크를 제안하였다. 이 연구들에서는 전 개발 단계에 대한 안전성 통합 활동을 제시하고 있지만 통합 활동의 수행 방법에 대한 가이드를 다루고 있지는 않다. 연구 동향을 비교한 결과는 표 1과 같다.

(표 1) 안전성 프로세스에 대한 연구 비교
(Table 1) Comparison of safety process

구분	목적	통합 프로세스	적용 단계	적용 형태
Jessyka [4]	안전 분석 활동과 요구 공학 활동의 통합 모델 제시	- 요구 공학 프로세스 - 안전 분석 프로세스	요구사항 단계	가이드라인
김영민 [5]	설계 프로세스와 안전 활동 프로세스 통합 모델 제시	- SysML 언어 기반 - 시스템공학 국제 설계 표준 (EIA-632) - 미국 국방부 시스템 안전표준 (MIL-STD-882)	시스템 설계 단계	통합 프레임워크
Pecoraro [6]	의료기기 소프트웨어 생명주기와 위험관리 프로세스의 통합 모델 제시	- 의료기기 소프트웨어 생명주기 프로세스 (IEC 62304) - 의료기기 소프트웨어 위험 관리 프로세스 (ISO 14971) - 의료기기 소프트웨어에 대한 ISO 14971 적용 지침 (IEC 80002-1:2009)	소프트웨어 개발 전 단계	통합 프레임워크
김동엽 [7]	의료기기 소프트웨어 생명주기와 위험관리 프로세스의 통합 모델 제시	- 의료기기 안전규격 표준 (IEC 60601-1) - 위험관리 프로세스 (ISO 14971)	소프트웨어 개발 전 단계	통합 프레임워크

(표 2) 의료기기 소프트웨어 안전성 관련 표준 비교
(Table 2) Comparison of medical device software standards

	목적	안전성 확보 방법	한계	비고
IEC 60601-1 [1]	의료기기 안전성 및 필수 요구사항 정의	의료기기 개발에 필요한 요구사항 활동을 정의함	적용 가이드라인을 제시하지 않음	위험 관리는 ISO 14971를 참조하며, 소프트웨어에 대한 부분은 IEC 62304를 참조
IEC 62304 [3]	의료기기 소프트웨어 개발 프로세스 정의	위험 등급에 따라 개발 프로세스의 상세 활동을 결정할 수 있도록 함	안전성 분석 방법과 안전 요구사항을 도출하는 방법을 제시하지 않음	위험 관리는 ISO 14971를 참조하며, IEC 60601-1의 요구사항을 참조
ISO 14971 [2]	의료기기 위험관리 프로세스 정의	의료기기의 위험 요소를 식별하고 이를 통제할 수 있도록 관리하는 활동을 정의함	위험관리의 기준과 가이드라인을 제시하지 않음	-

2.2 의료기기 안전성 표준에 대한 연구 동향

의료기기의 소프트웨어는 소프트웨어공학 프로세스, 위험 관리 등의 다양한 관점에서 의료기기 안전성에 대한 기준을 제시하고 이에 따라 개발할 수 있도록 표준이 제정되어있으며, 서로 참조하도록 구성되어 있다. 각 표준들을 비교한 결과는 표 2와 같다.

IEC 60601-1 (의료용 전기 기기 - 1부:기본 안전 및 필수 성능을 위한 일반 요구사항: Medical electrical equipment – Part 1:General requirements for basic safety and essential performance) 표준에서는 의료기기의 하드웨어와 소프트웨어에 대한 기본 안전성 및 필수 성능에 대한 일반적인 요구사항을 정의하고 있다. IEC 60601 시리즈중에서 IEC 60601-1은 공통적인 요구사항에 대해 기술하고 있으며, 소프트웨어가 적용된 의료기기에 대한 요구사항은 IEC 62304의 요구사항과 거의 유사하다.

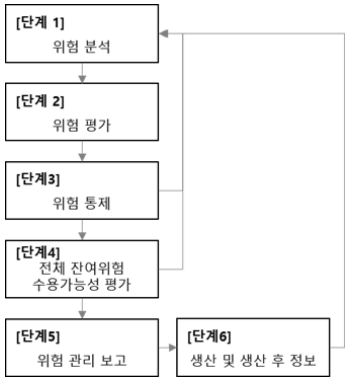
IEC 62304 (의료기기 소프트웨어 – 소프트웨어 생명주기 프로세스: Medical device software – Software life cycle process) 표준에서는 SW 요구사항 식별 및 개발 계획 수립, SW 요구사항 분석, SW 아키텍처 설계, SW 상세 설계, SW 구현 및 단위 테스트, SW 통합 및 통합 테스트, SW 시스템 테스트, SW 배포 등 8단계의 의료기기 소프트웨어 개발 생명주기와 각 단계에서 의료기기 SW를 안전하게 개발하기 위한 활동, 방법 등을 정의하고 있다. 의료기기 소프트웨어는 표 3에서와 같이 위험 등급을 구분하며, 이 등급에 따라 준수해야 할 세부 활동이 결정된다. 의료기기 소프트웨어의 안전성을 확보하기 위해 각 단계에서 어떠한 활동을 해야 하는지를 정의하고 있지만 이를 수행하기 위한 구체적인 방법을 제시하고 있지 않다.

(표 3) 의료기기 소프트웨어 안전 등급 분류
(Table 3) Medical device software safety classes

안전 등급	심각도
Class A	부상이나 신체적 피해가 발생할 가능성이 없음
Class B	심각하지 않은 부상(경상)이 발생할 가능성이 있음
Class C	사망 또는 심각한 부상(중상)이 발생할 가능성이 있음

ISO 14971 (의료기기 – 의료기기에 대한 위험관리 적용: Medical devices – Application of risk management to medical devices) 표준에서는 의료기기 소프트웨어와 관련된 위험원을 식별하고 위험을 평가 및 관리하는 절차를 정의하고 있다. 그림 1에서와 같이 위험관리 프로세스를 위험 분석 단계부터 총 6개의 단계로 구분하고 각 단계별 활동에 대해 정의하고 있다. 단계1인 위험 분석 단계에서는 의료기기의 안전과 관련된 특성들과 위험원을 식별하고 각 위험원에 대한 위험을 예측하며 단계2 위험 평가 단계에서는 식별된 위험 상황에 대해 위험 감소가 필요한지를 결정한다. 단계3 위험 통제 단계에서는 허용 가능한 수준까지 위험 감소 활동을 수행하고 단계4 잔여 위험 평가 단계에서 허용할 수 없는 기준의 잔여 위험이 있는지를 평가하고 잔여 위험에 대한 처리 및 공개 방법을 결정해야 한다. 단계5는 위험 관리 보고 단계이며 위험관리 프로세스를 검토하고 결과를 위험관리 보고서로 기록 및 보관해야 한다. 끝으로 단계6 생산 및 생산 후 정보 단계에서는 의료기기에 대한 정보를 수집하기 위한 시스템을 수립하고 유지해야 한다. 이 표준에서 정의하고 있는 위험

관리 프로세스와 각 단계별 활동은 개발 생명주기의 각 단계별로 수행되어야 하지만 어떻게 적용해야 하는지에 대해서는 명시하고 있지 않다.

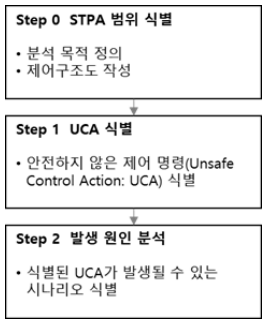


(그림 1) ISO 14971 위험관리 절차
(Figure 1) ISO 14971 risk management process

2.3 위험원 분석 기법 동향

위험원 분석 기법은 시스템에서 사고를 발생할 수 있는 요인들을 파악하는 방법으로 개발이 진행되는 동안 위험원을 식별하고 이를 완화하거나 예방하는 활동이 지속적으로 수행해야 한다. 대표적인 기법은 PHA[8], FTA[8], FMEA[8], HAZOP[8], STPA[9, 10] 등이 있고 이들 기법을 비교하면 표 4와 같다. PHA 기법은 시스템에 대한 정보가 상세하지 않은 개발 초기 단계에서 가장 우선적으로 수행되는 기법이고, FTA 기법은 분석 대상인 사고 또는 위험의 원인을 트리 형태로 연결하면서 분석을 수행한다. FMEA 기법은 시스템에서 발생 가능한 잠재적 고장을 식별하고 시스템에 미치는 영향과 원인을 분석하며, HAZOP 기법은 가이드워드를 활용하여 설계 또는 운용상에서 의도와 다르게 발생하는 상황과 위험 원인을 분석한다. STPA 기법은 사고의 발생 원인을 시스템 이론 관점에서 컴포넌트들이나 시스템들 간의 실패한 상호작용으로 분석한다. 본 연구에서는 기존 위험원 분석 기법들과 달리 위험을 복합적 시각에서 접근하는 STPA 기법을 적용한다.

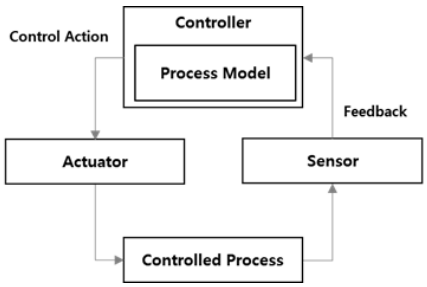
STPA 기법은 준비 단계인 0 단계를 포함한 3단계로 구분할 수 있으며(그림 2), 0단계에서는 STPA 기법을 수행하기 위한 대상 시스템을 파악하고 제어구조도(그림 3)를 작성한다. 제어구조도는 시스템의 제어기(Controller)를 중심으로 제어되고 있는 프로세스의 상태와 상태를 변화시킬 수 있는 제어명령을 전달하고 이에 대한 피드백을 전달받는다. 이를 기반으로 1단계에서는 사고를 발생시킬 수 있는 요인(Unsafe Control Action: UCA)을 식별하고 2단계에서는 사고 발생 원인을 분석한다.



(그림 2) STPA 프로세스
(Figure 2) STPA process

(표 4) 위험원 분석 기법 비교
(Table 4) Comparison of hazard analysis methods

기법	방법	한계	비고
PHA	시스템 개발 정보가 적은 초기 단계에서 시스템의 위험요인, 위험 상황 등을 분석함	시스템에 대한 많은 사전 정보가 요구됨	준비단계
FTA	고장의 원인 분석을 목적으로 고장을 유발하는 원인을 논리조합을 이용해 하향식으로 분석함	시스템에 대한 많은 사전 정보가 요구되며 하위 시스템에 적용하기 어려움	전단계
FMEA	의도하지 않은 시스템의 동작이 발생했을 때 고장이 시스템에 미치는 영향과 원인을 상향식으로 분석함	시스템 고장을 식별하기 위해 사전 정보가 요구되며 소프트웨어 구조가 복잡해질수록 적용이 어려움	전단계 (유형별 차이 있음)
HAZOP	시스템 상태 파라미터와 가이드워드를 조합하여 고장을 도출하고 그 영향과 원인을 분석함	시스템에 대한 많은 사전 정보(설계정보)가 요구됨	설계, 운용단계
STPA	사고의 발생 원인을 시스템 이론 관점에서 컴포넌트들이나 시스템들 간의 실패한 상호작용으로 분석하는 기법	시스템과 시스템 또는 구성요소들 간 제어정보가 요구됨	전단계



(그림 3) 제어구조도
(Figure 3) Control structure

2.4 안전성 테스트 케이스 생성 방법[11]

안전성 테스트 케이스는 안전 요구사항을 테스트하기 위해 생성하기 때문에 STPA 기법을 활용하여 컴포넌트 간의 상호작용이 실패하는 상황을 테스트할 수 있다. 안전성 테스트 케이스는 위험이 발생 가능한 상황과 특정 조건에 대한 정보가 포함되어야 하므로, STPA 분석 정보를 기반으로 테스트 시나리오와 테스트 케이스를 생성한다(그림 4).

테스트 시나리오는 위험 상황이 어떻게 발생하는지를 설명하기 위해 컴포넌트 간의 상호작용이 실패하는 흐름과 사고 발생 원인에 대한 정보를 통해 식별한다. 테스트 케이스는 테스트 시나리오를 기반으로 컴포넌트 간의 상호작용이 실패하게 된 조건을 테스트 케이스로 도출한다. 테스트 케이스의 입력값은 식별된 대상 컴포넌트의 출력값이 되고, 전제 조건은 프로세스 모델 변수와 UCA 정보를 활용하며, 예상 결과값은 제어 명령을 통해 작성한다(표 5).

(표 5) 테스트 케이스 생성 정보
(Table 5) Test case generation parameters

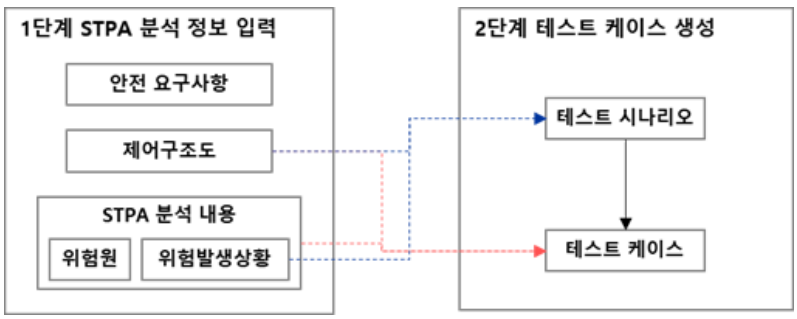
	STPA 활용 정보
전제조건	프로세스 모델 변수, UCA
입력 값	컴포넌트 출력값
예상 결과	제어 명령

3. 제안한 방법

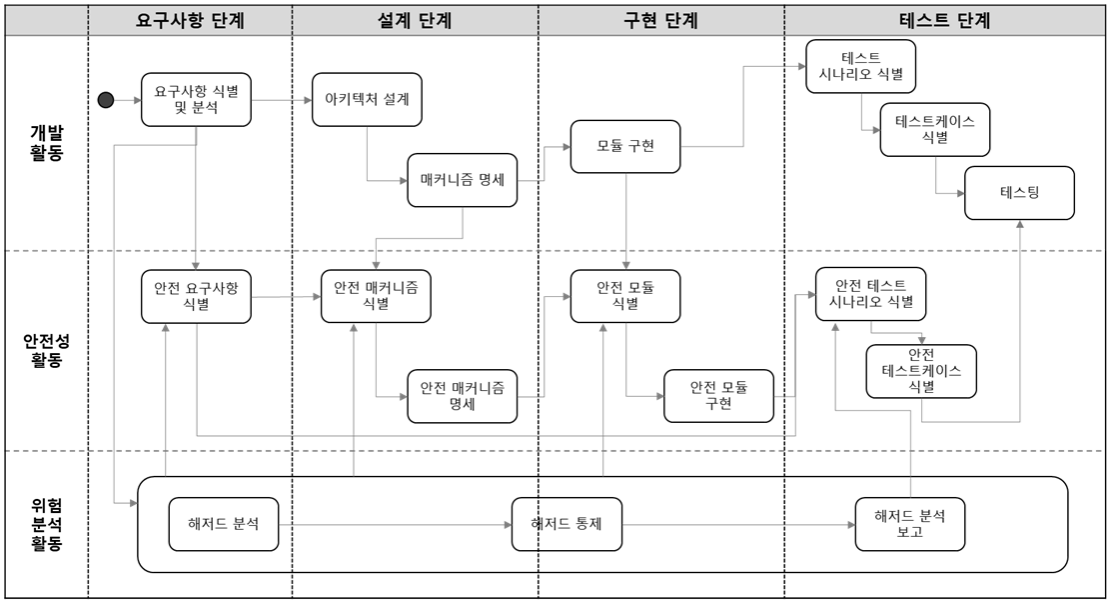
의료기기 소프트웨어의 안전성과 관련된 표준들을 적용할 수 있도록 가이드를 제공하는 연구가 수행되고 있다. 의료기기 소프트웨어를 개발하는 중소기업의 개발 프로세스에 따라 ISO/IEC 62304를 적용할 수 있는 방안에 대한 연구[12]나, 의료기기 소프트웨어 개발 프로세스와 위험관리 프로세스를 통합하는 연구[6, 13]가 수행되고 있지만 이들 연구에서는 각 단계별로 수행되어야 할 활동과 산출물에 대한 구체적인 적용 가이드라인이 없다.

본 논문에서 제시하는 프로세스는 개발 생명주기와 안전성 프로세스를 통합한 프로세스로 의료기기 개발 생명주기에 따라 시스템이 개발될 때 안전성 프로세스를 어떻게 수행되는지를 명시한다. 각 단계마다 반복되는 안전성 평가 프로세스 활동은 ISO 14971에서 정의한 위험관리 프로세스를 적용한다.

통합 프로세스는 그림 5와 같이 프로세스 흐름도를 보여주고 각 활동에 대한 세부 지침은 활동 정의서로 정의한다.



(그림 4) 테스트 케이스 생성 프로세스
(Figure 4) Test case generation process



(그림 5) 통합 안전성 프로세스 절차
(Figure 5) Integration of safety process flow

3.1 개발 및 안전성 활동

그림 5에서 개발 활동과 안전성 활동은 IEC 62304를 기반으로 하는 개발 및 안전성 프로세스의 활동들이다. 요구사항 단계의 활동부터 살펴보면 개발 활동으로는 대상 의료기기 SW의 요구사항을 식별하고 분석해야 한다. 안전성 활동에서는 안전 요구사항을 식별하는 활동을 수행한다. 안전 요구사항을 도출하기 위해서는 의료기기 SW의 안전성과 관련된 정보들이 먼저 식별되어야 하며, 시스템의 위험 원인을 분석하는 활동인 해저드 분석 활동이 선행되어야 한다. 설계 단계의 개발 활동은 의료기기 SW의 요구사항을 정확히 구현하기 위해 SW 아키텍처를 설계하고 상세화하는 활동을 수행한다. SW 단위들을 식별하고 이들 간의 관계를 명시한다. 안전성 활동에서는 이에 대한 해저드를 분석하고 통제할 수 있는 방법을 도출해야 한다. 안전 메커니즘을 식별하기 위해서는 안전 요구사항 식별과 해저드 분석의 결과를 활용하고 안전 요구사항으로 인해 발생하는 새로운 해저드 등을 확인하기 위해 해저드 분석 활동을 수행해야 한다. 구현 단계에서 개발 활동은 SW 단위별로 코드를 작성하는 활동을 수행하며, 안전성 활동에서는 코드의 안전성을 높이기 위해 코딩 표준을 명세하고 준수하도록 한다. 테스트 단계의 개발 활동은 의료기기 소프트웨어의 단위별 코드부터 각 단위들을 통합하며 테스트를 수행한다. SW 단위별로 구현하고 SW 단위들을 테스트하여 기능적 적합성을 검증한다. 안전성 활동에서는 안전성 요구사항이 정확하게 구현되었는지 확인해야 한다. 표 6은 활동 정의서 중에서 요구사항 단계의 활동 정의서의 일부이다.

3.2 위험 분석 활동

그림 5에서 위험 분석 활동은 ISO 14971를 기반으로 하는 위험 관리 프로세스의 활동들이다. 위험 분석 활동에서 해저드 분석 활동은 3단계로 구성되고 첫 번째 단계에서는 SW 특성을 식별하고 두 번째 단계에서는 해저드를 식별하며 세 번째 단계에서 해저드가 발생할 수 있는 위해 상황을 식별한다. 이 활동은 STPA 기법에서 제안하는 3단계 프로세스를 통해서 수행할 수 있다. 첫 번째 SW 특성을 식별하는 활동은 STPA 0단계의 제어구조도를 정의하며 분석 대상을 파악하는 활동으로 수행할 수 있다. 두 번째 활동은 STPA 1단계 활동에서 안전하지 않은 제어 명령(UCA)을 식별하는 활동이다. 마지막 활동은 STPA 2단계에서 식별된 UCA가 발생하는 원인을 파악하는 활동으로 수행할 수 있다. 해저드 통제 활동도 3단계로 수행

(표 6) 요구사항 단계 활동 정의서
(Table 6) Activity definition of requirement phase

목적	SW 요구사항을 정의하고 정의된 고객의 기술적인 요구사항 등에 대하여 검증된 명세를 개발한다.			
시작 조건		종료 조건		
1. SW 프로젝트 계획이 수립됨		1. SW 요구사항 명세서가 작성되어 검토 및 승인됨		
		2. 작업산출물을 형상관리함		
작업	**세부 지침**	**작업 산출물**	**담당**	**지원**
1. SW 요구사항 식별	• <요구사항 분석가>는 SW 요구사항을 문서로 작성함 ✓ 요구사항 명세서에 포함될 항목의 예시: SW 시스템 입력 및 출력, 인터페이스, 의료기기 설치 및 수용 요구사항, 운영 방법 및 유지보수 관련 요구사항, 보안 요구사항 등 ✓ [SW 요구사항 명세서] 양식 참조 • <요구사항 분석가>는 의료기기 SW가 충족해야 할 요구사항을 식별함	• SW 요구사항 명세서	• 요구사항 분석가	• 프로젝트 관리자
2. 위험 분석 및 통제	• <안전성 분석가>는 SW의 위험 요소를 분석하고 위험을 통제할 수 있는 방법을 도출함 • SW 요구사항에 대한 안전 요구사항들이 포함되어야 함 • SW 요구사항에서 위험에 대한 요구사항을 추적할 수 있도록 식별함	• 위험분석 결과보고서 • 안전 요구사항이 추가된 SW 요구사항 명세서	• 안전성 분석가	• 요구사항 분석가 • 프로젝트 관리자
3. 산출물 형상관리 수행	• <요구사항 분석가>와 <안전성 분석가>는 요구사항 식별 및 분석 단계에서 생성된 산출물들을 형상관리함 ✓ [형상관리 가이드라인] 참조		• 요구사항 분석가 • 안전성 분석가	

되며, 첫 번째 활동에서는 위험 수용 기준에 따라 해저드를 관리하기 위한 조치를 식별하고 두 번째 활동에서는 식별된 해저드 통제 조치를 구현하며 세 번째 활동에서는 잔여 위험에 대한 평가를 수행한다. 해저드 분석 보고 단계에서는 해저드 분석 프로세스를 검토하고 검토 결과를 문서화한다. 위험 분석 프로세스의 각 활동의 세부 지침은 활동 정의서로 작성되어 있으며 표 7은 해저드 분석 활동에 대한 활동 정의서의 일부이다.

4. 사례 연구

4.1 분석 대상

본 연구에서 제시한 의료기기 시스템의 통합 안전성 프로세스에 STPA를 적용한 결과를 확인하기 위해 방사선 치료를 위한 선량을 계획하는 방사선치료계획시스템(Radiation Treatment Planning System: RTPS)을 대상으로 분석을 수행한다. 방사선치료계획시스템은 CT 데이터를 통해 환자 치료 부위에 적절한 방사선량이 암 부위에 조사될 수 있도록 계획하고 검증하는 등의 방사선 치료를 위해 중요한 기반 작업을 수행한다. 이때 사용되는 환자의 의료 영상은 X-ray, CT, MRI 등의 촬영 영상과 환자 개인정보 등을 의료영상 국제표준인 DICOM(Digital Imaging and Communications in Medicine)으로 변환하고 의학영상정보시스템(Picture Archiving and Communication System: PACS)을 통해서 관리한다. 방사선치료계획시스템의 기능은 아래 표 8과 같이 요약될 수 있고, 이를 구조로 표현하면 그림 6과 같다.

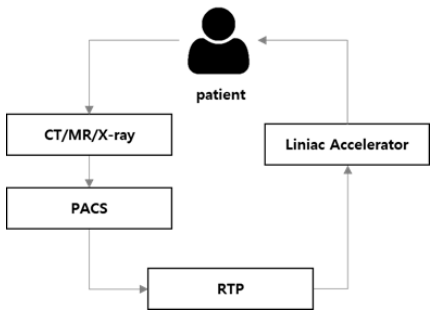
방사선치료계획시스템의 위험 요소는 방사선이 정확하게 투여되지 않았을 때 발생하므로 위험 요소에 대한 해저드 분석 활동은 치료 계획의 알고리즘에는 문제가 없다는 가정을 하고 환자의 의료 영상에 대한 관리 기능에 대해 수행한다. 따라서 통합 안전성 프로세스는 선량이 설정된 양보다 많이 조사되어 피폭의 위험이 발생하는 경우를 위험 상황으로 정의하고 적용한다.

(표 7) 해저드 분석 활동 정의서
(Table 7) Activity definition of hazard analysis

목적	SW의 해저드를 식별하고 발생 가능한 위험을 파악한다.			
시작 조건		종료 조건		
1. SW의 해저드 분석 계획이 수립됨		1. 해저드 식별		
		2. 위해 상황 예측		
작업	세부 지침	작업 산출물	담당	지원
1. SW 특성 식별	• <안전관리 담당자>는 의료기기 SW를 안전성과 관련된 특성 및 용도를 식별함 • 의료기기의 안전성에 영향을 미칠 수 있는 정성적 및 정량적 특성을 파악함 • SW가 요구사항대로 사용되거나 제품 설계와 구성을 따르는 일반적인 경우를 식별함 • SW의 발생 가능한 오용 사례를 식별함		• 안전관리 담당자	• 요구사항 분석가 • 프로젝트 관리자
2. 해저드 식별	• <안전관리 담당자>는 SW의 해저드를 분석함 • 정상 및 고장 상태에서 의료기기와 관련된 알려지고 예측 가능한 위험을 식별함		• 안전관리 담당자	• 프로젝트 관리자
3. 위해 상황 식별	• <안전관리 담당자>는 위해 상황을 발생시킬 수 있는 사건 또는 사건의 조합을 식별함 • 이전에 식별된 위해 상황을 활용하거나 아직 식별되지 않은 위해 상황을 식별하기 위한 특정 상황을 찾아야 함 • 위험도는 발생 확률과 결과를 활용하여 예측함 • 위험은 정성적이거나 정량적으로 예측할 수 있음	• 해저드 분석 보고서	• 안전관리 담당자	• 요구사항 분석가 • 개발자 • 프로젝트 관리자

(표 8) 방사선치료계획시스템의 기능
(Table 8) Function of RTPS

No	기능
1	PACS로 전환된 의료 영상을 관리함
2	방사선 치료장치에서 조사해야 할 방사선량을 계산함
3	방사선 치료에 필요한 정보를 생성함
4	방사선 치료 계획을 평가함
5	방사선 치료장치로 방사선량 등의 필요한 정보를 전달함



(그림 6) 방사선치료계획시스템의 구조도
(Figure 6) System architecture of RTPS

4.2 통합 안전성 프로세스 적용

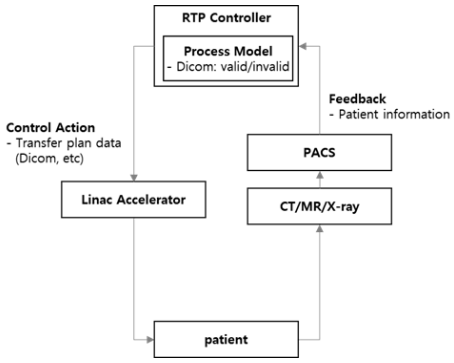
본 사례에서는 3장에서 제안한 통합 프로세스 중에서 안전성 프로세스 활동을 기반으로 요구사항과 테스트 단계의 활동을 중심으로 적용한다.

4.2.1 요구사항 단계

요구사항 단계에서는 요구사항과 안전 요구사항이 식별되어야 하며, 안전 요구사항은 통합 안전 프로세스에서 위험 분석 활동 중 해저드 분석을 수행한 결과를 통해 식별할 수 있다. 해저드 분석 활동의 3단계는 SW 특성 식별, 해저드 식별, 위험 상황 식별이다.

SW 특성 식별 작업에서는 방사선치료계획시스템의 제어구조도를 정의한다. 이 시스템은 환자 정보를 통해 방사선 치료에 대한 계획을 수립하므로 방사선치료계획시스템을 Controller로, 환자를 Controllered Process로 구성한다. 위험 분석의 범위를 환자의 의료 영상에 대한 관리 기능으로 제한하고 있기 때문에 방사선치료계획시스템이 방사선치료계획데이터를 제어명령으로 전달하기 위해 필수적인 정보를 Process Model로 정의하고 제어명령이 이행되어 방사선 치료가 이뤄진 결과를 Feedback으로 전달

한다. 그림 6의 시스템 구조도를 기반으로 제어구조도를 정의하면 그림 7와 같다.



(그림 7) 방사선치료계획시스템의 제어구조도
(Figure 7) Control structure of RTPS

해저드 식별 작업에서는 STPA에서 제공하는 4개의 가이드워드에 따라 UCA를 식별하고 식별된 결과는 표 9와 같다.

(표 9) UCA 식별
(Table 9) Identified UCA

	전송
Not Provide	a. 방사선 치료 계획 데이터가 전송되지 않음
Provide	b. 부정확한 방사선 치료계획이 전송됨
Too late or early	c. 방사선 치료 계획이 빠르게/늦게 전송됨
Soon or long	해당없음

식별된 UCA 중에서 a, c의 상황은 위험 상황이 아닌 것으로 판단하여 삭제하고, 최종적으로 식별된 UCA는 'b. 부정확한 방사선 치료 계획이 전송된다'이다. 세 번째 작업인 위해 상황 식별 작업에서는 UCA의 발생 원인 파악이며, 발생 원인은 4개의 기준에 따라 파악하게 되고 기준은 다음 표 10과 같다.

(표 10) 발생 원인 식별 기준
(Table 10) Criteria of identified causal factor

No	발생원인
①	외부입력 제어나 정보가 잘못되거나 없어짐
②	제어 알고리즘이 올바르지 않음
③	정보를 제공하는 시스템 또는 센서가 잘못된 경우
④	수행 대상이 실패하는 경우

기준별로 식별된 UCA의 발생 원인은 표 11과 같으며 각 발생 원인에 해당하는 UCA의 번호를 함께 표기하였다. 방사선치료계획시스템의 제어 SW에서 제어 알고리즘이 잘못되었을 경우 유효하지 않은 Dicom이 제공될 수 있으며, Dicom 유효성 검증 모델의 오류로 인해 Dicom 검증이 잘못 수행될 수 있다. 또한 PACS 시스템과 방사선치료계획시스템 사이의 통신상의 문제로 유효하지 않은 Dicom이 제공될 수 있다.

(표 11) UCA의 발생 원인
(Table 11) List of causal factors of UCA

	기준	발생원인
RTP Controller	②	제어 SW의 오류로 유효하지 않은 Dicom이 제공됨
	③	Dicom 유효성 검증 모델의 오류로 검증이 잘못됨
PACS	③	PACS 시스템과 방사선치료계획시스템 사이의 통신상의 문제로 유효하지 않은 Dicom이 제공됨

따라서 안전요구사항을 다음과 같이 식별한다.
- 제어 SW는 데이터의 유효성을 입증해야 한다.
- 의료영상(Dicom)의 안전한 데이터 전송을 입증해야 한다.
- 방사선치료계획 시스템의 제어기는 장애가 발생하더라도 시스템 운영에 영향을 주지 않도록 한다.

4.2.2 테스트 단계

테스트 단계에서는 안전성 테스트는 안전 요구사항의 구현을 확인하기 위해 수행되어야 하므로 안전 요구사항을 도출하는 과정에서 사용된 정보를 활용하여 테스트 케이스를 작성한다. 테스트 시나리오는 위험 상황이 어떻게 발생하는지를 설명하기 위해서 상호작용이 실패하는 흐름을 대상 시스템의 제어구조도와 함께 통합 프로세스의 위험분석 활동 중 해저드 분석 활동 3단계의 사고 발생 원인에 대한 정보를 통해 식별한다. 요구사항 단계에서 식별된 원인 중에서 "PACS 시스템과 방사선치료계획시스템 사이의 통신상의 문제로 유효하지 않은 Dicom이 제공됨"에 대한 테스트 시나리오를 표 12와 같이 생성할 수 있다. 이 테스트 시나리오에서는 PACS와 RTP Controller 사이의 네트워크 환경에서 Dicom 통신 내용을 확인하여 Dicom의 유효성을 확인한다.

(표 12) 테스트 시나리오
(Table 12) Test scenario

컴포넌트	PACS → RTP Controller
테스트 시나리오	PACS와 RTP Controller 사이의 Dicom 통신을 확인함

테스트 케이스는 테스트 시나리오를 기반으로 위험 상황을 활성화하는 특정 조건으로 생성된다(표 13). 따라서 컴포넌트들간의 상호작용이 실패하게 된 조건을 테스트 케이스로 도출하고 테스트 시나리오에서 식별한 대상 컴포넌트의 출력값이 테스트 케이스의 입력값이 된다. 입력값이 시스템에 입력되어도 시스템의 상태에 따라 위험 상황이 발생하지 않을 수 있으므로 프로세스 모델 변수와 UCA 정보를 활용하여 전제 조건을 작성할 수 있다. Dicom 통신을 하기 위해 메시지 교환 규격에 따라 데이터를 전송하도록 하며, 이를 DIMSE(DICOM Message Service Element)라고 하고 이를 테스트 케이스의 입력값으로 사용한다. DIMSE 파라미터를 전송하기 전에 PACS와 RTP Controller 간 전송/표시/통신 방법 등의 통신 정의(Association Negotiation)를 먼저 수행하도록 한다.

(표 13) 테스트 케이스
(Table 13) Test case

전제조건	Association Negotiation
테스트 데이터	C-FIND 요청(DIMSE 파라미터)
예상 결과	C-FIND 응답

5. 결 론

본 논문에서는 의료기기 소프트웨어의 안전성을 확보하기 위해 의료기기 개발 생명주기 표준인 IEC 62304와 의료기기 소프트웨어 위험 관리 표준인 ISO 14971을 기반으로 하는 통합 안전성 프로세스를 제시하였다. 이 프로세스는 의료기기 SW 안전성 프로세스를 전체 개발 생명주기에 통합하여 개발 초기 단계에서부터 SW의 안전성 품질을 확보할 수 있도록 한다. 뿐만 아니라 하나의 프로세스로 체계적인 관리를 할 수 있으며, 단계별로 안전성 프로세스가 반복적으로 수행되어야 하는 시점을 파악할 수 있다.

또한 STPA 기법을 요구사항 단계부터 테스트 단계까지 개발 생명주기 전 단계에서 활용할 수 있는 방법을 사례연구로 보임으로써 프로세스 활동을 어떻게 수행할 수 있는지에 대한 방법을 제시하였다. STPA 기법은 프로세스 단계별로 가이드가 제공되어 안전성 분석 활동의 경험이 적은 경우에도 상대적으로 분석 활동이 용이하므로, 제시한 통합 안전성 프로세스에서 효과적으로 접근하고 관리할 수 있다.

향후에는 본 연구에서 통합 안전성 프로세스의 수행 단계 및 활동에 대해서 중점을 두고 있지만 이에 대한 검증 및 확인 프로세스가 연계되어 있지 않으므로 해당 프로세스를 추가하는 연구가 필요하다. 또한 다른 의료 분야의 시스템들에도 적용한 다양한 사례 연구가 수행되어야 하며, STPA 이외의 다른 기법들을 적용한 비교 연구가 수행되어야 한다.

참고문헌(Reference)

[1] IEC, Medical electrical equipment – Part1: General requirements for basic safety and essential performance, IEC 60601-1, 2020. https://webstore.iec.ch/publication/67497

[2] ISO, Medical devices – Application of risk management to medical devices, ISO 14971, 2019. https://www.iso.org/standard/72704.html

[3] IEC, Medical device software - Software life cycle processes, IEC 62304, 2006. https://www.iso.org/standard/38421.html

[4] J. Vilela, J. Castro, L.E.G. Martins, and T. Gorschek, "Integration between requirements engineering and safety analysis: A systematic literature review," Journal of Systems and Software, Vol. 125, pp. 68-92, 2017. https://doi.org/10.1016/j.jss.2016.11.031

[5] Y.M. Kim, and J.C. Lee, "On the Use of SysML Models in the Construction of the Design Process for Safety-Critical Systems," Journal of Korea Safety Management & Science, Vol. 15, No. 3, pp. 7-17, 2013. https://doi.org/10.12812/ksms.2013.15.3.7

[6] F. Pecoraro and D. Luzi, "The integration of the risk management process with the lifecycle of medical device software," Methods Inf Med, Vol. 53, No. 2, pp. 92-98, 2014. https://doi.org/10.3414/me13-01-0014

[7] D.Y. Kim, Y.S. Park and J.W. Lee, "Development Life Cycle-Based Association Analysis of Requirements for Risk Management of Medical Device Software," KIPS Transactions on Software and Data Engineering, Vol. 6, No. 12, pp. 543-548, 2017. https://doi.org/10.3745/KTSDE.2017.6.12.543

[8] C.A. Ericson Ⅱ, "Hazard analysis techniques for system safety," John Wiley & Sons, 2005. https://onlinelibrary.wiley.com/doi/book/10.1002/0471739421

[9] W. Young and N. Leveson. "Systems thinking for safety and security," Proceedings of the 29th Annual Computer Security Applications Conference, pp. 1-8, 2013. http://hdl.handle.net/1721.1/96965

[10] N. Leveson, "Engineering a safer world: Systems thinking applied to safety," MIT Press Cambridge, 2011. https://mitpress.mit.edu/books/engineering-safer-world

[11] B.Y. Choi, S.K. Han, S.Y. Lee, S.Y. Chung and B.G. Lee, "A Tool for Safety Requirements Test Case Generation Based on STPA," Conference of Computing Science and Engineering, pp. 420-422, 2018. https://www.dbpia.co.kr/journal/voisDetail?voisId=VOIS00380440

[12] Kasisopha, Natsuda, and Panita Meananeatra. "Applying ISO/IEC 29110 to ISO/IEC 62304 for medical device software SME," Proceedings of the 2nd International Conference on Computing and Big Data, pp.121-125, 2019. https://doi.org/10.1145/3366650.3366670

[13] D.Y. Kim. Y.S. Park. and J.W. Lee. "Development Life Cycle-Based Association Analysis of Requirements for Risk Management of Medical Device Software," KIPS Transactions on Software and Data Engineering 6(12), pp543-548, 2017. https://doi.org/10.3745/KTSDE.2017.6.12.543

◐ 저 자 소 개 ◑

최 보 윤(Bo-yoon Choi)
2007년 서울여자대학교 컴퓨터학과(공학사)
2009년 서울여자대학교 대학원 컴퓨터학과(이학석사)
2017년 서울여자대학교 대학원 컴퓨터학과(이학박사)
2017년~2018년 상명대학교 산학협력단 산학협력교수
2019년~현재 서울여자대학교 미래산업융합대학 정보보호학과 초빙강의교수
관심분야 : 소프트웨어 안전성, 소프트웨어 프로세스, 소프트웨어 테스팅 etc.
E-mail : choiby@swu.ac.kr

이 병 걸(Byong-gul Lee)
1988년 University of Bridgeport 물리학과(이학사)
1996년 Auburn University 대학원 전산학과(공학석사)
1998년 Auburn University 대학원 전산학과(공학박사)
1998년~현재 서울여자대학교 미래산업융합대학 데이터사이언스학과 교수
관심분야 : 소프트웨어 보안, 소프트웨어 안전성, 소프트웨어 아키텍처, 소프트웨어 프로세스 etc.
E-mail : byongl@swu.ac.kr