제26회 한국정보처리학회 추계학술발표대회 논문집 제13권 제2호 (2006. 11)

의료용 소프트웨어생명주기 프로세스 분석

○
최민용, 강영규, 허찬회, 이정림, 박기정, 박해대, 이인수, 김혁주
식품의약품안전청 의료기기평가부 전자의료기기팀
e-mail: mychoi@kfda.go.kr

A Analysis of Medical Device Software Life Cycle Processes

Min-Yong Choi, Young-Kyu Kang, Chan-Hoi Hur, Jeong-Rim Lee,
Ki-Jung Park, Hae-Dae Park, In-Soo Lee, Hyeog-Ju Kim
Dept of Medical Device and Radiation Health,
Korea Food and Drug Administration

요 약

2006년 5월 국제전기기술위원회(International Electrotechnical Commission, IEC) TC62(의료용전기기기기술위원회) SC62A(의료용전기기기 공통특성에 관한 세부분과위원회)에서 의료기기 소프트웨어의 생명주기 프로세스에 관한 국제규격인 IEC 62304, Ed.1을 제정하였다. 전체 내용은 크게 일반적인 요구사항(General requirements)과 소프트웨어 개발과 유지보수에 대한 프로세스(Software development & maintenance process)로 구성되어 있다. 그리고 소프트웨어 개발과 유지보수 프로세스가 진행되는 동안에 기본적으로 확보되어야 하는 소프트웨어 위험관리와 형상관리에 대한 프로세스(Software risk management & configuration management process)를 규정하고, 또한 문제 발생 시 이를 해결하기 위한 소프트웨어 문제 해결 프로세스(Software problem resolution process)를 규정하고 있다. 이는 기존의 정보통신 분야에서 사용되던 소프트웨어 생명주기 프로세스인 ISO/IEC 12207 규격과 외형상 많은 차이를 나타내고 있다. 이에 본 논문에서는 의료기기 소프트웨어의 생명주기 프로세스에 관한 국제규격인 IEC 62304를 분석하여 실제 의료용 소프트웨어 개발 및 유지보수 작업들이 어떠한 방식으로 이루어지는지를 분석하고, 또한 기존의 소프트웨어 생명주기 프로세스인 ISO/IEC 12207 규격과의 차이점을 비교·분석하여 본 규격에 대한 근본적인 활용 방안을 모색하고자 한다.

1. 서론

최근 전자의료기기의 개발 동향을 살펴보면 정보통신 기반기술의 확산으로 인하여 다양한 기술과 융합된 형태로 개발된 제품들이 증가하고 있다. 이러한 제품들은 대부분 의도된 기능을 사용하는데 있어 소프트웨어를 이용하고 있기 때문에 궁극적으로 사용상의 편의성을 제공하고 있다.

의료용 소프트웨어란 일반적으로 의료기기의 목적으로 개발되어 의료기기내에 내장되거나 하드웨어에 종속적이지 않고 단독으로 사용되는 소프트웨어라 할 수 있다. 이러한 의료용 소프트웨어는 정보통신 분야 및 기타 산업 분야에서 사용되는 소프트웨어와는 달리 사용대상이 대부분 의사, 간호사, 환자 또는 의료기기를 사용하는 개인 등으로 국한되어 결과적으로 인간이라고 볼 수 있다. 따라서 의료용 소프트웨어를 사용하기 위해서는 사용상에 있어 인간에게 안전한가를 먼저 고려하여야 하며, 이것을 전제로 사용에 따르는 효능 및 효과가 입증되어야 한다.

2006년 5월 국제전기기술위원회(International Electrotechinical Commission, IEC) TC62(의료용전기기기기술위원회) 내의 SC62A(의료용전기기기의 공통특성에 관한 세부분과위원회)에서 의료용 소프트웨어에 대한 생명주기 프로세스 IEC 62304, Ed.1, (Medical Device Software - Software Life Cycle Processes)를 제정하였다. 이는 기본적으로 의료용 소프트웨어의 개발과 유지보수를 위한 생명주기 프로세스를 규정하고 있으며, 이와 함께 품질관리시스템(Quality Management System)과 위험관리시스템(Risk Management System)을 적용하는 방법을 규

정하고 있다. IEC 62304 규격의 기원은 기존의 정보통신 분야에서 널리 활용되었던 소프트웨어 생명주기 프로세스에 대한 국제규격인 ISO/IEC 12207:1995, (Information Technology - Software Life Cycle Processes(Amendment 1: 2002, Amendment 2: 2004))이라 할 수 있다. ISO/IEC 12207 규격은 IEC 62304 규격이 제정되기 이전에 의료기기 분야에서 사용되기도 하였으며, 비단 의료기기 분야뿐만이 아니라 산업 전반에서 사용되는 소프트웨어에 적용 가능 하였다.

이에 본 논문에서는 의료기기 소프트웨어에 대한 생명주기 프로세스를 규정하고 있는 IEC 62304, Ed.1를 분석하여 실제 소프트웨어를 개발하고 유지보수 하는데 수행되어야 하는 일련의 작업들을 알아보고 의료기기의 특성상 기본적으로 요구되는 품질관리와 위험관리에 대한 요구사항에 대해서도 알아보고자 한다.

2. 일반 요구사항

의료용 소프트웨어에 대한 일반 요구사항은 안전성을 확보하는데 개념을 두고 있다. 그러나 소프트웨어의 안전성을 완벽하게 보증해 줄 수 있는 방법은 존재하지 않으므로 본 규격에서는 의료용 소프트웨어의 안전성을 향상시키기 위한 방법을 3가지 주요사항으로 제시하였다.

- 품질관리(Quality Management)
- 위험관리(Risk Management)
- 소프트웨어공학(Software Engineering)

2.1. 품질관리시스템(Quality Management System)

의료용 소프트웨어의 제조자는 자신의 회사에서 개발하는 의료용 소프트웨어를 고객의 요구사항과 법적 요구사항을 준수하여 공급할 수 있는 능력을 증명하여야 한다.

품질관리시스템(Quality Management System)은 일반적으로 국제표준화기구(International Standard Organization)에서 제정한 ISO 13485:2003(Medicals Devices - Quality Management Systems - Requirements for Regulatory Purposes) 규격의 적용을 통하여 이루어지며, 만일 국가에서 법적으로 요구하는 품질관리시스템이 있는 경우 그 요구사항의 만족을 통하여 형성될 수 있다. 또한 일반적인 소프트웨어에 대한 품질관리시스템의 요구사항은 ISO/IEC 90003:2004, (Software Engineering - Guidelines for the application of ISO 9001:2000 to computer software)에서 찾아볼 수 있다.

2.2. 위험관리(Risk Management)

위험관리시스템(Risk Management System)은 이미 국제표준화기구(International Standard Organization)에서 제정한 ISO 14971:2000(Medical Device - Application of Risk Management to Medical Devices) 규격이 의료기기 분야에서 폭넓게 활용되고 있다. 이를 반영하여 IEC 62304에서도 ISO 14971의 내용을 참조하여 위험관리를 하도록 규정하고 있다. 위험관리는 일반적으로 의료용 소프트웨어와 관련된 위해요인의 식별(Hazard Identification)을 시작으로 이루어진다.

2.3. 소프트웨어 안전성 분류(Software Safety Classification)

의료용 소프트웨어의 안전성 분류는 소프트웨어의 사용으로 인해 발생할 수 있는 상해(injury)의 심각도(severity)에 따라 다음과 같이 3가지 등급으로 구분할 수 있다.

<표 1> 의료용 소프트웨어 안전성 분류체계

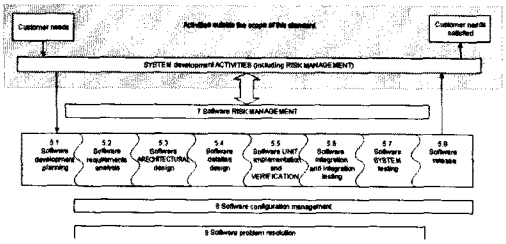
분류	심각도
Class A	아무런 상해 또는 손상 없음
Class B	심각한 상해 없음
Class C	죽음 또는 심각한 상해

의료용 소프트웨어가 Class C나 B에 해당한다면 하드웨어 위험통제(Hardware Risk Control)나 고장의 발생 및 심각도를 경감시키는 방법을 통하여 의료용 소프트웨어의 안전성 분류 결과를 ISO 14971에서 정의하고 있는 수용 가능한 레벨(acceptable level)로 낮추어야 한다.

3. 의료용 소프트웨어 생명주기 프로세스

3.1. 소프트웨어 개발 프로세스

의료용 소프트웨어 개발 프로세스는 아래의 (그림 1)와 같이 8가지의 단계를 거쳐 이루어진다.



(그림 1) 의료용 소프트웨어 개발 프로세스

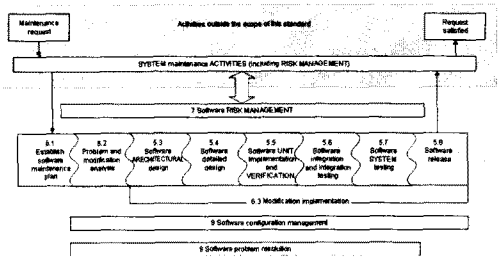
이는 ISO 12207에 규정되어 있는 소프트웨어 개발 프로세스와 유사하지만 기본적인 개념의 차이를 가지고 있다. 그 차이점을 살펴보면 IEC 62304 규격에는 시스템(System)의 개념이 빠져있고, 의료기기를 위해 규정된 중복되는 활동들을 생략하며, 위험관리 프로세스(Risk Management Process)와 소프트웨어 릴리즈 프로세스(Software Release Process)가 추가되었다. 그리고 프로세스 구현(Process Implementation) 활동과 계획(Planning) 활동들이 하나의 소프트웨어 개발 계획(Software development planning) 활동으로 통합되었으며, 안전성(Safety) 요구에 따라 요구사항이 식별된다는 것 등이 있다.

<표 2> IEC 62304와 ISO/IEC 12207의 소프트웨어 개발프로세스 비교

IEC 62304	ISO 12207
Software development planning	Process implementation
Software requirements analysis	System requirements analysis
Software ARCHITECTURAL design	System architectural design
Software detailed design	Software requirements analysis
SOFTWARE UNIT implementation and verification	Software architectural design
Software integration and integration testing	Software detailed design
SOFTWARE SYSTEM testing	Software coding and testing
Software release	Software integration
	Software qualification testing
	System integration
	System qualification testing
	Software installation
	Software acceptance support

3.2. 소프트웨어 유지보수 프로세스

의료용 소프트웨어의 유지보수 프로세스는 크게 소프트웨어의 유지보수계획을 수립(Establish software maintenance plan)하는 단계와 문제와 수정을 분석(Problem and modification analysis)하는 단계 그리고 수정을 구현(Modification implementation)하는 단계로 구분해 볼 수 있다. 그러나 세부적으로 살펴보면 수정을 구현하는 단계는 의료용 소프트웨어 개발 프로세스에서 소프트웨어 요구사항 분석 후의 나머지 활동들에 해당된다. 따라서 개발 프로세스나 유지보수 프로세스나 계획을 수립하거나 분석을 하는 대상만 다를 뿐이지 본질적으로 동일한 구조의 프로세스라 할 수 있다. 이러한 의료용 소프트웨어 유지보수 프로세스는 아래의 (그림 2)에 나타나 있다.



(그림 2) 의료용 소프트웨어 유지보수 프로세스

하지만 유지보수 프로세스와 개발 프로세스는 두 가지 측면에서 다른 점을 가지고 있다. 첫 번째는 유지보수 프로세스는 제조자가 급박한 문제에 대한 신속한 변경을 구현할 때 소프트웨어 개발 프로세스에 보다 작은 규모의 프로세스를 사용하는 것이 허용되는 것이고, 두 번째는 출시된 제품과 관련된 소프트웨어 문제 보고(Software Problems Reports)에 있어 제조자는 문제를 규명하는 것뿐만 아니라 법과 관련된 사항들을 만족하여야 한다는 것이다.

3.3. 소프트웨어 위험관리 프로세스

의료용 소프트웨어의 위험관리는 개발 프로세스와 유지보수 프로세스의 전 영역에 걸쳐 지속적으로 요구되는 사항이다. 위험관리의 주요 내용을 살펴보면 아래와 같다.

- 소프트웨어가 위험한 상황을 초래하는 경우를 분석
- 위험통제 측정
- 위험통제 측정 검증
- 소프트웨어 변경에 대한 위험관리

위험통제는 특정 수준 이하로 위험을 감소시키거나 유지하는 작업을 말한다. 그리고 위험통제 측정은 하드웨어, 소프트웨어, 작업환경, 또는 사용자 운영방법을 대상으로 수행될 수 있다.

3.4. 소프트웨어 형상관리 프로세스

의료용 소프트웨어 형상관리 프로세스 또한 위험관리 프로세스와 마찬가지로 개발과 유지보수 프로세스의 거의 전 영역에 걸쳐 지속적으로 요구되고 있다. 이러한 형상관리 프로세스는 아래와 같이 크게 3가지로 구분해 볼 수 있다.

- 형상 식별
- 변경 통제
- 형상 상태 축적

의료용 소프트웨어 제조자는 형상 아이템과 그 아이템의 버전을 유일하게 식별하기 위한 체계를 설립하여야 하며, 이 체계를 이용하여 다른 소프트웨어 제품이나 출처가 알려지지 않은 소프트웨어(Software of Unknown Provenance, SOUP) 그리고 문서들도 관리하여야 한다.

형상 아이템의 변경을 변경 요청에 의해서만 이루어져야 한다. 그리고 제조자는, 소프트웨어 시스템과 소프트웨어 아이템에 대한 소프트웨어 안전성 분류의 변경을 포함하여, 변경의 결과로 인하여 반복적으로 요구되는 일련의 활동들을 식별하고 수행하여야 한다. 그리고 시스템 형상을 포함하여 통제되는 형상 아이템의 변경 기록을 축적하여 문제 발생 시 그 상태를 회복 가능하도록 하여야 한다.

3.5. 소프트웨어 문제해결 프로세스

의료용 소프트웨어 문제해결 프로세스는 이전의 모든 프로세스에서 발생하는 문제들을 분석하고 해결하기 위한 프로세스이다. 이 프로세스는 소프트웨어공학 분야에서는 "결함추적(Defect Tracking)"으로 불리기도 하며, ISO/IEC 12207과 IEC 60601-1-4, Amendment 1. 에서는 "문제 해결(Problem Resolution)"으로 불린다.

문제 보고(Problem Report)는 안전하지 못하거나 사용의도에 벗어난 또는 정해진 기준에 반대되는 소프트웨어 제품의 실제 또는 잠재적인 행위를 기록하는 것이다. 문제해결 프로세스를 위해서 먼저 이 문제 보고를 준비하여야 한다. 문제 보고는 다음의 3가지 요소를 이용하여 식별하여야 한다.

- 유형(예: 시정, 예방, 적합 등)
- 범위(예: 변경 크기, 관련 디바이스 번호, 변경 시간 등)
- 중상(예: 성능, 안전성, 보안 등에 영향)

문제에 대한 해결이 이루어지면 그 결과와 문제 보고에 대한 검증이 이루어진다. 그리고 문제해결로 인한 추가적인 문제가 발생하는가에 대해서도 검증하여야 한다.

4. 결론

소프트웨어는 부가가치가 높은 기술이지만 그 실체가 특정 모습으로 형상화되지 않기 때문에 그의 존재를 간과하는 경우가 종종 발생한다. 의료기기 분야에서는 현재 소프트웨어가 대부분의 전자의료기기에 내장된 형태로 존재하기 때문에 의료기기를 구성하는 한 부분으로 인식되고 의료기기의 형상이 소프트웨어의 모습을 대변해 주고 있지만 실제 안전한 의료기기를 사용하기 위한 노력의 핵심에는 소프트웨어의 안전성이 확보되어야 한다는 것을 인지하여야 할 것이다.

본질적으로 의료용 소프트웨어에 대한 생명주기 프로세스를 규정하고 있는 IEC 62304 규격은 의료분야에서 중요하게 생각하는 안전성을 기반으로 두고 있기 때문에 ISO/IEC 12207의 생명주기 프로세스와 기본적은 개념에서 차이를 나타내고 있으나 소프트웨어를 개발하고 유지보수 하는 기술적인 활동은 유사한 모습을 나타낸다. 결과적으로 의료용 소프트웨어의 생명주기 프로세스를 효율적으로 사용하기 위해서는 의료기기의 특성을 반영하여 위험관리와 품질관리를 수행하고 그것을 바탕으로 소프트웨어공학의 기술을 적용하여야 할 것이다.

참고문헌

[1] IEC 62304, Ed.1, Medical Device Software - Software Life Cycle Processes

[2] ISO/IEC 12207:1995, Information Technology - Software Life Cycle Processes

[3] ISO 13485:2003 Medicals Devices - Quality Management Systems - Requirements for Regulatory Purposes

[4] ISO 14971:2000 Medical Device - Application of Risk Management to Medical Devices

[5] ISO/IEC 90003:2004, Software Engineering - Guidelines for the application of ISO 9001:2000 to computer software