http://dx.doi.org/10.12985/ksaa.2016.24.2.081
ISSN 1225-9705(print) ISSN 2466-1791(online)

Technical Review

항공용 소프트웨어 안전성 및 개발시 주의사항에 대한 고찰

이백준*, 최종연*, 남기욱*

A Study on Safety of Airborne Software and Considerations during Development

Baeckjun Yi*, Jong Yeoun Choi*, Gi Wook Nam*

ABSTRACT

It is recognized that safety is a key point of technical competency. Its adoption is widely spread in development of products and it is essentially necessary in aerospace industry because airborne system and equipment are used complex high-technology and implemented systematic performance using software. This study reviews system safety assessment, development assurance level, airborne software, RTCA DO-178 process, and considerations & pitfalls in software development.

Key Words : Airborne Software(항공용 소프트웨어), System Safety Assessment(시스템안전성평가), Considerations during Development(개발 고려사항)

1. 서 론

시스템의 안전성은 항공우주, 교통, 원자력, 의료 분야를 포함한 여러 분야에서 중요한 사항이다. 무엇보다 항공분야는 극저온, 고온, 다습 등 다양한 환경에 노출되고, 비행 중 고장 또는 작동 불능 시 대처가 용이하지 않으며, 사고 발생 시 인명/물질적 피해가 매우 크기 때문에 안전성이 최우선으로 고려되어야 한다. 또한, 최근 제품은 하드웨어와 소프트웨어가 통합적으로 개발되지만 사실상 소프트웨어가 전체 시스템의 안전성을 결정하는 핵심 요소이다[1].

따라서 안전한 소프트웨어를 개발하기 위해서는 먼저 시스템 안전성평가를 수행하여 적정한 개발수준을 결정하는 절차와 항공용 소프트웨어를 개발하기 위해 현재 국제표준으로 인정되는 RTCA DO-178에 대해 고찰하고, 항공용 소프트웨어 개발자가 지켜야 할 주의사항 및 흔한 실수 등에 대해 살펴보고자 한다.

Received : 09. May. 2016. Revised : 28. May. 2016. Accepted : 15. Jun. 2016
* 한국항공우주연구원 SBAS사업단
연락저자, E-mail : ybj@kari.re.kr
대전광역시 유성구 과학로 169-84

2. 본 론

2.1 시스템안전성평가

항공기는 수십만 개 이상의 정밀 부품을 이용하는 초정밀 첨단기술을 적용할 뿐만 아니라 3차원 공간의 고고도에서 고속으로 비행하기 때문에 항공기 및 관련제품 개발 시 안전성 확보가 필수적으로 요구되며, 항공기 및 관련 부품은 첨단기술과 체계적인 개발 절차를 적용하므로 설계부터 운용까지 안전성평가가 필수적이다.

안전성평가에는 항공기 개발 활동을 위하여 요구조건을 설정하고 이를 검증하는 과정, 항공기의 기능을 평가하는 과정, 시스템 및 개별 부품에 대한 위험성을 사전에 분석하고 이에 대하여 적절한 조치를 취하는 과정, 대상 품목에 대한 조치의 적절성에 대한 판단 등이 포함되며, 이는 개발 및 설계 과정과 병행하여 유기적으로 진행하여야 한다[2].

이러한 시스템 안전성평가 프로세스에 대해서는 미국 자동차기술자협회(Society of Automotive Engineers, SAE) Aerospace Recommended Practices (ARP) 4754A 및 4761 문서에 자세하게 기술되어 있는데, 예상되는 모든 고장 조건을 사전에 규정

하고, 이러한 고장 조건을 유발할 수 있는 고장의 조합 중 중요한 모든 사항을 고려하여 이에 대하여 적절한 조치가 취해졌다는 것을 보장할 수 있도록 계획 및 관리되어야 한다. 또한, 통합 시스템의 안전성평가 프로세스에서는 개별 부품 및 서브시스템의 통합으로 인해 발생할 수 있는 추가적인 복잡성 및 상호의존성까지 고려하여야 한다. 통합 시스템을 포함한 모든 안전성 평가 대상품목에 대해서는 개발 초기부터 시스템의 적절한 안전성 목표가 수립되어야 하고, 이 목표의 만족여부를 종합적으로 판단하기 위해서 안전성 평가 프로세스는 필수적인 과정이다[2, 3].

항공 산업계에서 통용되고 있는 안전성평가는 기능위험평가(Functional Hazard Assessment, FHA), 예비 시스템안전성평가(Preliminary System Safety Assessment, PSSA), 시스템안전성평가(System Safety Assessment, SSA)로 구성된다.

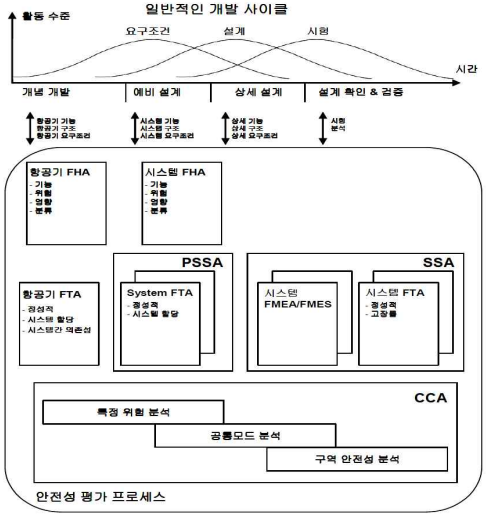

Fig. 1 Process of System Safety Assessment[3]

2.2 안전성평가와 개발보증수준의 관계

하드웨어의 경우 정량적인 확률로 신뢰성을 규정할 수 있으나, 소프트웨어의 경우에는 요구조건을 완전하고 올바르게 설정하기 어려움, 논리적으로 복잡함, 전체 조건의 시험이 불가함 등으로 완벽한 소프트웨어 개발은 상당히 어렵고, 물리적인 시스템 요소보다는 논리적인 시스템 요소로 구성되어 있어 소프트웨어의 모든 고장은 설계 또는 프로세스의 오류로 인한 것이라 고장 메커니즘이 하드웨어와는 달라서 명확하게 소프트웨어의 안전성/신뢰성을 정량적으로 표현할 수 없다.

그래서 적절한 프로세스를 따라 진행 단계마다 지속적이고 반복적인 검토, 점검 등 개발 활동을 통한 오류 제거를 기본 철학으로 하는 RTCA DO-178과 같은 보증 표준을 만들어 이 표준을 준수함으로써 소프트웨어의 신뢰성을 절차적으로 보장하도록 요구하는 것이다. DO-178에서는 고장영향이 클수록 더 엄격한 절차(더 많은 목표 수)를 배정하여 개발 과정에서의 소프트웨어의 오류를 줄이기 위한 노력을 시스템적으로 구현한다.

다시 말해 하드웨어의 경우, FHA에서 치명 고장상태인 경우 수송급 항공기에서는 비행시간당 10^{-9}의 고장확률을 보장하여야 하지만, 소프트웨어의 경우에는 이를 정량적으로 구현할 수 없기 때문에 DO-178에서는 소프트웨어 기능이 치명 고장상태를 유발할 수 있는 시스템에서는 개발보증수준을 레벨 A로 설정하여 소프트웨어적인 오류가 발생하지 않도록 설계 단계뿐만 아니라 검증단계에서도 검토(review), 분석(analysis), 시험(test) 등의 검증 절차를 아주 엄격하게 수행하도록 프로세스를 규정하는 것이다[4].

2.3 항공용 소프트웨어 인증기준

현재까지 국제 사회에서 개발 통용되고 있는 소프트웨어 표준은 크게 군사용 기술기준, 민간단체 기술기준, ISO 기술기준, 항공전문용 기술기준으로 나눌 수 있는데, 항공용 소프트웨어 표준은 현재 RTCA DO-178 Rev.C가 국제 표준으로 인정되고 있다.

RTCA DO-178C는 항공기 탑재 소프트웨어가 인증을 획득하기 위해 준수해야 할 지침으로, 소프트웨어 기술이 급격히 발전하고 있고, 그 규모 및 복잡성이 증가하고 있으며, 더욱 더 많은 비행안전과 관련된 분야에서 사용되고 있는 현실에서 그 중요성이 부각되고 있다.

DO-178C에서는 소프트웨어 안전수준을 결정하기 위해 시스템 엔지니어링 업무의 일환으로 시스템 안전성평가가 요구되고, 그에 따라 소프트웨어가 달성하여야 하는 목표도 달라지는데 부록 A에서 그 목표 71가지와 산출물 22건을 제시하고 있다. 이러한 목표는 전체 개발 활동과 통합 프로세스로 구성되는데, 부록 A에 다음과 같이 제시되어 있다[5].

- 소프트웨어 계획 프로세스
- 소프트웨어 개발 프로세스
- 소프트웨어 요구조건 프로세스 출력의 검증

- 소프트웨어 설계 프로세스 출력의 검증
- 소프트웨어 코딩 및 통합 프로세스 출력의 검증
- 통합 프로세스 출력의 시험
- 검증 프로세스 결과의 검증
- 소프트웨어 형상관리 프로세스
- 소프트웨어 품질 보증 프로세스
- 인증 지원 프로세스

2.4 항공용 소프트웨어 개발 단계별 주의사항 및 흔한 실수

소프트웨어 개발 단계별로 개발자가 주의할 점은 다음과 같은 것들이 있다[6].

1) 계획서
- 개발자가 수립된 계획을 준수하지 않음
- 계획서가 서명되지 않거나 형상관리 되지 않음
- 계획이 너무 일반적임
- 계획서에서 DO-178C의 모든 목표를 언급하지 않음
- 계획서의 변경 통제가 이뤄지지 않음
- 전환기준이 명확하지 않음
- 계획서의 작성이 늦어짐
- 품질보증 권한이 명시되지 않음
- 계획서 사이의 일관성/연계성이 없음

2) 표준서
- 너무 막연하여 준수하기 어려움
- 너무 제한적이거나 장황함
- 불완전하거나 비현실적임

3) 요구조건
- 요구조건이 잘 정의되지 않음
- 요구조건의 변경이 너무 늦은 경우가 많음 (적기에 소프트웨어에 구현하기 어려움)
- 추적성 매트릭스가 빈약함(추적성이 손실되거나 파생요구조건이 누락됨)
- 시스템 개발자와 소프트웨어 개발자간의 의사소통이 없음
- 파생 요구조건이 안전성 평가 인원에게 전달되지 않음

4) 설계
- 설계가 복잡하거나 혼동되게 작성됨(코딩하는 사람이 이해하기 어려움)
- 시스템 개발자와 소프트웨어 개발자간에

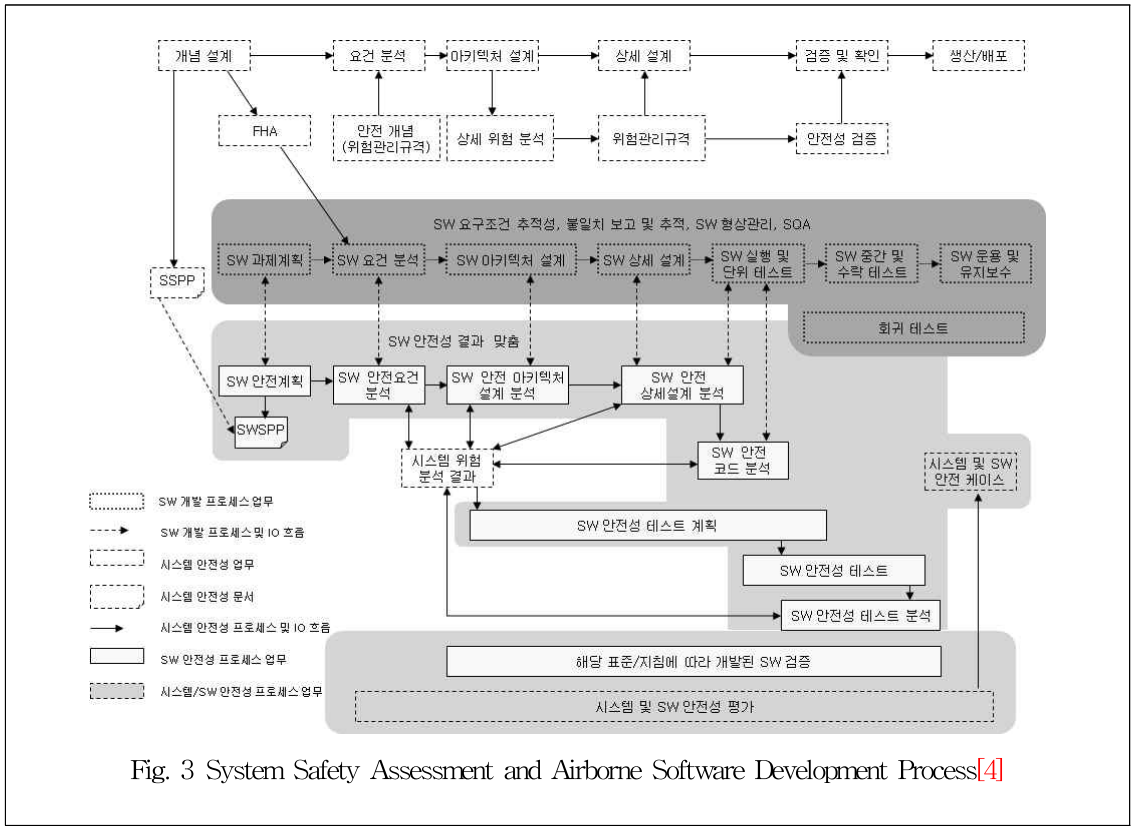

Fig. 3 System Safety Assessment and Airborne Software Development Process[4]

의사소통이 잘 이뤄지지 않음
- 시스템이 분할되어야 하는 경우 진정으로 분할되지 않음
- 다수의 개발자가 시스템 사양에서 바로 저수준 요구조건을 바로 도출하려 함(종종 설계 오류 유발)
- 데이터가 형상관리의 통제를 받지 않음
- SQA가 적극적으로 관여하지 않음
- 긴밀하게 연결된 모듈(한 모듈에 변경이 발생하면 파급효과가 큼)
- 설계 표준을 준수하지 않음
- 추적성이 불완전하거나 손실됨
- 설계가 검증가능하지 않음
- 파생 요구조건이 안전성 평가팀에게 전달되지 않음

5) 코딩
- 저수준 요구조건 및 아키텍처가 완전하지 않음
- 코딩 표준을 따르지 않음
- 요구조건의 변경 없이 코드가 변경됨
- 요구조건이 변경되었으나 코드에 반영되지 않음
- 코드가 너무 치밀하여 유지보수가 어려움

6) 검증
- 빈약한 추적성
- 모든 요구조건이 검증되지 않음
- 올바르지 않거나 부적정한 커버리지 분석 방법
- 계획서에 제시된 것처럼 개발단계에서 검토 및 검사가 이뤄지지 않음
- 소프트웨어 요구조건 시험 대신 시스템 요구조건 시험을 수행
- 독립성 부족
- 검증 계획서를 따르지 않음
- SQA의 관여가 없음
- 시험 케이스가 완전하지 않음
- 시험 결과가 문서화되지 않음
- 문제 보고서가 문서화되지 않음
- 시험 대신 분석이 수행됨
- 시간 제약으로 변두리 부분의 시험이 수행되지 않음
- 강건성 시험(robust test)이 누락되거나 부적절함

7) 소프트웨어품질보증(Software Quality Assurance)
- SQA 담당인원의 소프트웨어공학 배경지식이 충분하지 않음
- 인원이 불충분함
- 제한적인 관여
- 권한의 부족
- 독립성 부족
- 경영층의 지원 부족

8) 형상관리
- 소프트웨어 수명주기 환경이 정의되지 않음
- 고유하게 형상을 식별하는데 실패함
- 부품번호 부여 방법이 불명확함
- 품질보증 기록이 형상관리되지 않음
- 변경 프로세스를 따르지 않고 변경이 발생함
- 소프트웨어 빌딩 절차가 반복성이 없음
- 탑재시 사용되는 CRC(Cyclic Redundancy Check)가 평가되지 않음

그리고 DO-178C를 적용하는데 흔히 있는 대표적인 실수 13가지를 소개한다[7].
- DO-178C를 소프트웨어 개발 생태계로 고려하지 않음
- SAE ARP 4754A에 따르지 않은 불충분한 소프트웨어인증계획서(Plan for Software Aspects of Certification, PSAC)
- 열등한 품질보증
- 부적절하거나 자동화되지 않은 추적성
- 요구조건에서 부적절한 수준의 내용
- 기능시험에서 부족한 경로 커버리지
- 툴 검증: 불충분한 커버리지 및 부족한 PSAC
- 정형 표시법 없이 검증을 줄이기 위한 정형기법 적용
- 파라미터 취급 실패
- 검증 어려움으로 인한 모델링 회피
- 제한 없는 C++의 사용
- 재사용가능 소프트웨어 구성품 고려않음
- 레벨 A 검증 미고려(Modified Condition /Decision Coverage, 객체코드 상관관계 등)

3. 결 론

항공기는 수십만 개 이상의 정밀 부품을 이용하는 초정밀 첨단기술을 적용하는 분야로, 하드웨어와 소프트웨어가 통합적으로 개발되지만 사실상 소프트웨어가 전체 시스템의 안전성을 결정하는 핵심 요소이다. 안전한 소프트웨어를 개발하기 위해 항공기 시스템안전성평가와 연계하여 개발보증수준을 규정하고 있다.

이에 항공기 시스템안전성평가의 개요 및 항공 탑재용 소프트웨어의 사실상의 표준(de facto standard)인 RTCA DO-178에 대해서 간략히 살

펴보았다. 또한, 시스템안전성평가와 내장형 소프트웨어 개발보증수준 관계를 알아보고 개발자가 고려할 사항과 흔한 실수를 각 단계별로 살펴보았다.

우리나라에서도 정보통신기술 기술을 기반으로 항공전자 시스템을 개발하였거나 개발을 준비중인 사례가 늘어가고 있어, 본고에서 다룬 사항들을 적절하게 설계 프로세스에 반영하면 안전하고 더 나은 항공전자 시스템 및 항공 탑재용 소프트웨어를 개발할 수 있을 것으로 기대한다.

후기

본 연구는 국토교통부 항공안전기술개발사업의 연구비지원(16ATRP-A087579-03)에 의해 수행되었습니다.

Reference

1) Baeckjun Yi, Seung-kyem Kim, “Software Consideration in Airborne System Design and Certification”, Aerospace Engineering and Technology, Vol.3 No.2, 2004, pp.177-182
2) SAE, ARP 4754A “Certification Considerations for Highly Integrated or Complex Aircraft Systems”, 2010
3) SAE, ARP 4761 “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment”, 1996
4) Baeckjun Yi, “A Study on Aircraft System Safety Assessment and Airborne Software Development Assurance Level”, 2014 SASE Fall Conference, 2014
5) RTCA, DO-178C “Software Considerations in Airborne Systems and Equipment Certification”, 2011
6) Baeckjun Yi, Youngkwon Jin, “Airborne Software Approval and Common DO-178B Pitfalls”, 2011 SASE Fall Conference, 2011
7) Vance Hilderman, “DO-178B to 178C: Avoiding the Unlucky 13 Mistakes”, Atego HighRely, 2011, pp.26-40