연구보고서 2016-019

소프트웨어 안전 분야 재직자 역량 제고를 위한 교육 커리큘럼 개발에 관한 연구

A Study on the Development of Educational Curriculum to Improve Employee Competency in Software Safety Field

진회승/박태형/민상윤/지은경/유준범/박정희

2017.04.

SPRi 소프트웨어정책연구소

이 보고서는 2016년도 미래창조과학부 정보통신·방송연구개발사업의 연구결과의 보고서 내용은 연구자의 견해이며, 미래창조과학부의 공식입장과 다를 수 있습니다.

목 차

표 목 차

그 림 목 차

요 약 문

1. 제 목

소프트웨어 안전 분야 재직자 역량 제고를 위한 교육 커리큘럼 개발에 관한 연구

2. 연구 목적 및 필요성

오늘날 ICT 융합 시스템에서 소프트웨어가 차지하는 비중은 지속적으로 증가하고 있다. 이러한 상황에서 소프트웨어의 결함으로 인한 사고는 특히 사회, 경제적 피해가 막대할 가능성이 있는 만큼 소프트웨어의 기능적 안전성은 필수적인 요건으로 점차 부각되고 있다.

하지만 국내에서는 실무에서 소프트웨어 안전과 일반적 품질에 대한 구분을 못하여 혼재되어 있는 상황이며, 소프트웨어 안전관련 서비스 제공업체도 특정 기법에 초점을 맞춘 경우가 많아 체계적이고 포괄적인 소프트웨어 안전성을 보증하기에는 어려움이 많은 현실이다.

본 연구에서는 이러한 현실을 감안하여 안전 필수 소프트웨어 개발에 필요한 역할 및 기술을 조사하고 실수요자의 기대사항을 조사함으로써, 소프트웨어 안전 관련 교육 커리큘럼(안)을 개발함으로써 재직자 중심의 소프트웨어 안전 역량 강화를 위한 정책 기초자료로 활용하고자 한다.

3. 연구의 구성

본 연구는 총 7장으로 구성되어 있다.

제 1장에서는 서론으로 소프트웨어 안전 전문가 가격제도 연구의 배경, 필요성, 연구의 목적 및 본 연구를 수행한 연구 수행 방법을 제시한다.

제 2장에서는 이론적 배경으로 소프트웨어 안전 정의, 소프트웨어 공학, 소프트웨어 안전 기술 역량 도출을 위해 철도 분야와, 미연방에너지국 안전 소프트웨어 품질 보증 표준을 설명하고 안전 시스템 소프트웨어 개발을 위한 기술 집합을 도출한다.

제 3장에서는 민간 전문기관, 대학, 공공기관, 민간 인증기관의 소프트웨어 안전 해외 교육 현황에 대해 분석한다.

제 4장에서는 소프트웨어 안전 국내 교육 현황에서 대해 조사하고 분석한다.

제 5장에서는 실제 국내 실무자의 니즈 파악을 위하여, 국내 안전 필수 소프트웨어 제공자 및 사용자를 위한 필요교육에 대한 수요조사를 실시하고 분석한다. 먼저 소프트웨어 안전 분야의 재직자를 대상으로 소프트웨어 안전교육에 대한 현황 및 수요에 대한 설문조사를 진행한 후, 안전 인증 경험이 있는 선도 업체들을 대상으로 심층인터뷰를 실시, 설문조사의 내용과 비교, 분석하여 설문조사의 결과에 신뢰성을 높일 수 있도록 하였다.

제 6장에서는 소프트웨어 안전 교육의 문제점을 개선할 수 있는 방안을 제시하고, 소프트웨어 안전 기술 집합과 국내외 교육 현황을 기반으로 하여 예상 교육 과목을 도출한다.

제 7장은 결론으로서, 연구의 요약, 향후 연구 방향, 연구의 한계점 등에 대해 논의한다.

4. 연구 내용 및 결과

본 연구는 소프트웨어 안전 기술과 역량 및 이에 관련된 교육과정 수립에 앞서 필요한 소프트웨어 안전 관련 배경지식을 설명하고, 이를 배경으로 산학연 전문위원단과 논의하여 안전 시스템 소프트웨어 개발을 위한 주요 기술 집합을 도출하였다.

더불어 해외 교육 현황 파악을 위하여, 해외 선진국에서 진행되고 있는 소프트웨어 안전 관련 교육 현황들에 대한 조사 결과, 다음과 같은 특징들을 살펴볼 수 있었다.

- 기초적인 이론 중심의 교육들은 대학 중심으로 정규과정 혹은 비정규과정 형태로 진행되고 있다. 대학별로 전공 교수의 관심사에 따라 특화된 기술영역에 중점을 두는 추세이다.
- 실제 산업에서는 소프트웨어 안전 교육이 시스템 수준에서 HW, SW까지 포괄적으로 이루어지고 있었다. 그리고 다양한 도메인의 공통적 특성에 맞춘 IEC61508 중심의 교육들이 존재하는 것을 확인할 수 있었다.
- 대부분의 소프트웨어 안전 교육들이 소프트웨어 안전 관련 전문회사 및 기관들에 의해 제공되고 있으며, 도메인별 표준에 관련된 교육들은 인증기관이나 관련 표준기관에 의해 제공됨을 알 수 있다.

해외 현황 조사와 동시에, 국내 교육 실정을 파악하기 위하여 국내에서 현재 진행되

고 있는 소프트웨어 안전 관련 교육 현황들에 대한 조사 결과, 다음과 같은 특징들을 살펴볼 수 있었다.

- 대학 중심의 교육은 연구 중심으로 특정 대학들에서 대학원 과정의 해당 전공자들에게 한정적으로 제공되고 있으나, 해외와 같이 포괄적이고 깊게 제공되고 있지는 않다.
- 도메인별 교육은 자동차 산업 분야가 가장 활발하게 이루어지고 있으며, 대부분의 도메인 교육이 해당 도메인의 전문 컨설팅 회사를 중심으로 이루어지고 있다. 자동차 외의 산업분야에서는 필요에 따라 간헐적인 교육이 일어나고 있으나, 공식적인 정규과정들이 정기적으로 개설되고 운영되고 있지는 않다.
- 표준 중심의 교육들은 관련 컨설팅 업체와 국내에서 사업 중인 해외 인증서비스 제공업체에서 수행되고 있다.

소프트웨어 안전교육에 대한 설문조사는 소프트웨어 안전에 대한 이해 및 현재 교육 현황에 주안점을 두고 진행하였으며, 조사 모집단은 소프트웨어 안전성 전문가 포럼 회원 기관 및 산업별 안전 규제 기관을 참조하여 안전 소프트웨어 사용 비중이 높은 다양한 산업군에서 표본을 추출하고 산학연 자문위원단이 이를 검토하여 조사 대상 기관을 선정하였다.

소프트웨어 안전교육에 대한 설문조사의 결과 필요성이 높은 교육 과정은 소프트웨어 안전 인식 제고, 소프트웨어 안전성 테스팅 기법, 소프트웨어 안전성 관리이었으며, 시급성이 높은 교육 과정은 형상관리 및 결함관리, 소프트웨어 안전성 테스팅 기법, 시스템 안전 설계였다. 소프트웨어 안전 구현에 더 중요한 과정은 낮은 순위에 위치하여 소프트웨어 안전이 중요한 기업도 아직 소프트웨어 안전에 대해 정확한 이해가 부족한 것으로 추정된다.

국내외 교육 현황 조사 결과와 국내 실무자 수요 조사를 통하여 결론적으로 다음과 같은 결과를 도출하였다. 기본적인 소프트웨어 안전 지식이 부족하고, 관련 교육에 대한 환경이 해외에 비해 매우 부실한 실정이다. 이와 같은 현실을 개선하기 위하여 우선 정부 차원에서 범 도메인 차원의 소프트웨어 안전 관련 기초 개념 교육을 마련하여 이를 통해 소프트웨어 안전 인식을 개선하고, 각 도메인별로 진행되는 국제 표준에 대한 산발적 교육보다는 대학 및 대학원에서 진행되는 이론 교육과 연결된 체계적 교육 환경 마련이 필요하다.

또한 시급성에 따라 산업 도메인별로 국제 수준에 준하는 소프트웨어 안전과 관련된 법규 및 자격인증을 강화하고 일정부분 소프트웨어 안전 관련 교육을 필수화하는 등

시스템 및 소프트웨어 안전 관련 교육의 수요를 높이는 것 또한 필요하다고 판단된다. 그리고 이러한 인식 개선 및 필요성 증가에 대비하여 다양한 소프트웨어 안전 교육을 개설 및 확대하고, 이에 대한 홍보를 강화하여 교육이 수월하게 진행될 수 있도록 정부와 각 기관에서의 협조가 필요할 것이다.

5. 정책적 활용 내용

본 연구결과는 소프트웨어 안전 확보에 필요한 기술을 조사하고 실수요자의 기대사항을 조사하여 소프트웨어 안전 분야 재직자 역량 제고를 위한 교육 커리큘럼(안)과 현재 교육의 개선방안을 제시함으로써 소프트웨어 안전 교육 정책 마련에 기초 자료로 활용될 수 있다.

또한 소프트웨어 안전 확보를 위한 전문가 양성 정책의 일환으로 소프트웨어 안전 관련 자격제도 정책 수립 시 필요한 소프트웨어 안전 기술 및 역할 정의에 활용 가능하다.

6. 기대효과

본 연구는 교육의 3대 요소인 교육 콘텐츠, 학습자, 교수자 중 교육 콘텐츠에 대한 것으로 소프트웨어 안전 관련 학습자 역량 강화 및 교수자 양성 방안 마련의 기초가 되며, 궁극적으로 국내 소프트웨어 안전 역량을 강화하여 소프트웨어 안전 관련 산업 및 소프트웨어 안전이 중요한 산업의 국제 경쟁력 확보에 기여할 것이다.

SUMMARY

The role of software is dramatically increasing in ICT system, especially for the systems with mechanical and electronical hardware. Under this circumstances, the fault of software may cause incident and disaster as we all know. This risk may consequently result in fatal losses in economy, social security. Thus, the functional safety of software in ICT system is mandatory requirements. We have seen many fatal incident reports due to software from all over the world in recent years.

However, the difference between traditional software quality and software functional safety is not recognized and understood correctly in domestic industry. Even though comprehensive technical approach for software functional safety is required over entire software development lifecycle, most of domestic software organizations focus on merely validation and testing activities, which is very limited activity for functional safety assurance.

In this study, we survey and analyze practitioner's expectation regarding software functional safety engineering. And we expect to use the result for further strategy for developing and leveraging domestic engineer's capability for software functional safety.

In this report, we firstly explain the background knowledge for software functional safety, prior to actual planning of training program for the functional safety. Also, with the help from experts supervising committee consisting of experts from academic, research, and industry organization, we elicit appropriate corresponding skill sets. We also use related international standards and models, which is related to functional safety and software engineering.

Besides, we also survey overseas training programs as reference information. As result, we found the following major findings.

- The fundamental knowledge training programs are conducted as a regular or even-based course in various universities.
- In actual industry, the software functional safety training programs are conducted including system and hardware level based on IEC 61508 standard.
- Most of domain specific training programs are conducted domain specific software safety expert companies like consulting firms and auditing bodies.

In contrast, the training status in domestic ecosystem is much more immature or weak. The followings are the major findings of domestic training status.

- The training courses in universities are provided in limited form to certain graduate studies. Also, the training curriculums are not comprehensive as U.S. and Europe.
- Regarding domains, automotive industry is the most active domain where software functional safety training programs are actively conducted. The other domains have only event-based training activities which are in much more passive attitude.
- Standard-based training is provided by corresponding experts groups(i.e. conducting firms and international auditor bodies)

We also conducted practitioner's expectation survey for various domains including nuclear energy, train, aviation, automotive, medical, and etc. This survey has been conducted in two ways; on-line survey and face-to-face in-depth interview to compliment the weakness of each approach. Based on the survey, while using the training curriculums developed by our experts board members, we have the training courses need results sorted in two aspects; urgency and necessity.

Based on domestic and overseas training survey and domestic practitioner's needs survey, we come out with following conclusions:

Currently correct understanding of software functional safety is still needed and the environment for related education is insufficient compared to overseas.

To overcome these weaknesses, it may be important for government to provide basic functional safety training programs, which can be used as cross-domain fundamental functional safety training. It is also necessary to provide fundamental functional safety concept training programs. And these can become seamless training bridge between university training courses and domain-specific industry-driven training. These government-driven training programs must be conducted by qualified software safety engineering experts, not by certification-oriented personnel.

It shall be also good plans if government develop the qualification system of software safety expert as well as functional safety related regulations and laws to assure social safety. For these activities, cross-government cooperation and collaboration shall be needed for effective policy deployment.

CONTENTS

제1장 서 론

제1절 연구 배경 및 필요성

최근 후쿠시마 원전사고, 영종대교 106종 추돌 등 자연재난과 인재가 합쳐진 형태의 대형 재난 안전사고를 계기로 2015년 “안전혁신 마스터플랜”이 작성되었으며, 이에 따라 재난·안전체계를 원점에서 재검토하여 개선하는 중·장기 혁신계획을 세우고, 범정부 실행체계를 수립하여 계획을 실행하고 있다.[1] 재난을 미리 예방·대비하기 위해 관리 감독하고, 재난 대응 역량을 강화하며, 생활 속 안전 문화 확산이 목표이다. 그러나 재난의 예방과 재난 시 사고 대응 방안은 소프트웨어보다는 하드웨어에 국한되어 있다.

2016년 다보스포럼의 주제로 선정된 제4차 산업혁명에 의해 경제와 사회의 디지털화에 따라 전통적 산업과 사회적 규범이 파괴적 혁신에 직면하고 있다. 제4차 산업혁명의 키워드인 디지털화에 가장 큰 영향을 주는 것 중에 하나가 소프트웨어이다. 2014년 “SW중심사회실현전략”에 따르면 소프트웨어 중심사회는 소프트웨어가 개인·기업·정부에 광범위하게 사용되어 삶의 질을 향상시키고, 기업과 정부의 경쟁력을 지속적으로 제고되는 사회로 정의한다. 개인은 소프트웨어를 통해 문제를 해결하고, 기업은 소프트웨어로 신사업을 창출할 수 있는 창조적 파괴가 가능하며, 정부는 소프트웨어 기반으로 국가 시스템을 효율적이고 능동적으로 운영하게 된다고 한다. [2]

소프트웨어정책연구소에서는 소프트웨어 중심사회에서 안전을 보장하기 위해서는 소프트웨어 안전이 필수적이라 판단하고, 2015년 “소프트웨어 안전 체계 확보와 중점 추진과제”에서 체계적 사고예방 시스템 구축, 전문적 사고대응 체계 확립, 현장 중심의 소프트웨어 안전 고급인력 양성, 시장기반 확충을 통한 소프트웨어 안전산업 활성화, 범부처 정책조정 기능 강화라는 5개의 추진전략을 선정했다.[3]

소프트웨어가 차지하는 비중은 지속적으로 증가하고 있는 상황에서 전력 관리 시스템 오류로 인한 광범위 지역 정전, 자동차 급발진 사고, 항공기 추락사고 등의 사례로 볼 때, 안전 필수 시스템에서의 제어 소프트웨어의 기능 안전성은 필수적인 요건으로 점차 부각되고 있다. 따라서 안전 필수 시스템에서 운영되는 소프트웨어 시스템에서

1) 관계부처 합동, 안전혁신 마스터플랜, 2015.3
2) 관계부처 합동, 소프트웨어중심사회 실현 전략, 2014.7
3) SPRi, SW안전 체계 확보와 중점 추진과제, 2015

발생할 수 있는 장애 및 위험 요인들을 식별하여 이를 회피, 처리할 수 있는 안전 체계를 올바르게 갖추고 실행하는 것이 필수적이다. 이러한 소프트웨어 기능안전에 대한 전문지식은 일반적인 소프트웨어 개발보다 훨씬 고수준의 안전공학 기술을 요구한다.

소프트웨어정책연구소에서 2015년 실시한 “국내 소프트웨어 안전 산업동향 조사”에 의하면 소프트웨어 안전 선진국에서는 민간 주도로 소프트웨어 안전 시험·인증 등의 서비스 분야가 활성화 되고 있다. 민간 주도로 국제 표준을 선점하고 있으며, 정부는 표준을 준수하도록 제도화하여, 안전 소프트웨어 산업에 주도권을 강화하고 있다.[4] 반면 국내에서는 소프트웨어 안전에 대한 개념조차 성립되어 있지 않아, 품질·보안과의 차이도 인식하지 못하고 있다. 안전 필수 시스템 개발 관련자들은 안전 표준 적용 시, 소프트웨어 공학의 기술을 많이 사용하는 표준의 용어를 이해하기조차 어려운 상황이다. 시스템 발주자들은 소프트웨어 안전에 대한 개념 및 인식이 낮아 소프트웨어 안전에 대한 중요성에 간과하는 분위기가 있다.

따라서 안전 관련 시스템 발주자, 관리자 및 실무자의 소프트웨어 안전 인식 제고뿐만 아니라, 소프트웨어 안전 관련 전문성 및 국제 경쟁력을 확보하기 위해 체계적이고 전문적인 역량 개발의 필요가 있음을 정부 차원에서 인식한 상태이다. 이를 위하여 재직자 중심의 소프트웨어 안전 역량 강화를 위한 소프트웨어 안전 교육 정책 마련을 위한 기초 자료가 필요하다.

제2절 연구 목적

소프트웨어 안전을 보장하고 소프트웨어 안전 산업을 발전시키기 위해서는 체계적인 소프트웨어 안전 교육이 필요하다. 대학에서는 물론 재직자들의 지속적인 재교육도 필요하다. 해외와 국내를 비교할 때 안전 교육의 목표는 다르지 않으나, 안전에 대한 인식 및 소프트웨어 안전산업의 성숙도가 다르므로, 교육의 범위 및 수준은 같지 않을 것이다.

본 연구는 국내 소프트웨어 개발 기관의 체계적인 안전공학 역량 확보를 위하여 현재 국내 및 해외의 안전공학 기술 현황을 파악하고 향후 국내 개발 기관들이 갖추어야 할 필요 기술들을 분석하여 이를 기반으로 소프트웨어 안전 교육 전략 수립을 목적으로 한다.

---

4) SPRi, 국내 소프트웨어 안전 산업동향 조사, 2015.11

이 보고서에서는 소프트웨어 안전에 대한 개념을 정의하고, 그에 따라 소프트웨어 안전을 지키기 위한 역량 목록을 개발하고, 현장 중심의 고수준의 안전공학 기술을 보유하는 소프트웨어 안전 고급인력 양성의 전략을 실행하기 위한 재직자 중심의 소프트웨어 안전 고급인력을 위한 교육커리큘럼(안)을 제안한다. 이를 위하여 안전 필수 소프트웨어 관련 재직자를 대상으로 실 교육 수요자 입장에서의 기술 습득 현황과 필요 교육 내용에 대한 조사를 실시한다. 이러한 조사를 통해 각 산업 도메인별 소프트웨어 기능안전 분야의 필요 기술을 정립하고 교육과정(안)을 개발하여 향후 교육 시행을 위한 효과적인 추진 전략을 수립하여 체계화하는 정책의 기초자료로 사용하고자 한다.

제3절 연구 내용

본 연구는 앞에서 언급한 소프트웨어 안전 역량 강화를 위한 교육 정책 수립을 위하여 다음과 같이 구성으로 연구를 진행하고자 한다.

서론에서는 소프트웨어 안전 기술 교육 프로그램 개발 연구의 배경과 필요성, 목적, 내용, 방법에 대해 기술한다.

이론적 배경에서는 소프트웨어 안전의 개념, 소프트웨어 공학에서 사용되는 기술, 소프트웨어 안전에 필요한 기술들에 대해 논의한다.

해외 주요국의 교육현황에서는 미국 및 유럽에서 현재 활발하게 수행되고 있는 소프트웨어 안전 관련 교육 활동들에 대한 조사를 실시하였다. 소프트웨어 안전은 안전 필수 시스템에서 중요 시 되므로, 소프트웨어 안전 표준 기관이 실시하는 교육이 도메인별로 기본적인 교육이 시행되고 있다. 사설 교육 기관의 경우는 소프트웨어 안전 산업과 관련된 업체가 시행하고 있다. 이러한 선진국의 소프트웨어 안전 관련 교육들은 산업 도메인별로 주도되는 교육, 대학 주도의 교육, 공공기관/인증기관 주도의 교육 차원에서 분류하여 조사하였다.

국내 교육현황에서는 국내에서 수행되고 있는 소프트웨어 안전 관련 교육 활동들에 대한 조사를 실시하였다. 국내 소프트웨어 안전 관련 교육들은 선진국에 비하여 체계화되어있지는 않으며, 관심사와 분야별로 국소적으로 진행되고 있는 상태이다.

설문조사 및 분석에서는 안전 필수 소프트웨어 제공자 및 사용자를 위한 필요 교육에 대한 수요를 조사하고, 결과를 분석한다. 먼저 소프트웨어 안전 분야의 재직자를 대상으로 소프트웨어 안전 교육에 대한 현황 및 수요에 대한 설문조사를 진행한 후,

안전 인증 경험이 있는 선도 업체들을 대상으로 심층 인터뷰를 실시하였다.

조사 모집단은 주요 산업군 중 안전 소프트웨어 사용 비중이 높은 원자력, 철도/지하철, 항공/국방, 의료, 승강기, 미래형자동차 등의 분야를 우선으로 구축하였다. 또한 한국표준산업분류, 특수 분류 산업 중 소프트웨어를 사용/제작하는 업체, 주요 정보통신기반시설 중 안전 관련 시스템을 사용하는 기관 등을 참조하여 다양한 산업군에 대해 조사할 수 있도록 하였다.

국내외 교육과정들에 대한 조사 결과와 실제 재직자들을 대상으로 한 수요조사를 기반으로 시사점을 도출하고, 현재 필요한 소프트웨어 안전 관련 교육 커리큘럼과 교육 개선 방안에 대하여 제안한다.

제4절 연구 방법

본 연구는 체계적인 소프트웨어 안전 기술 교육 프로그램에 대한 최초의 연구로 국내외의 자료조사가 선행되었다. 자료조사를 바탕으로 한 문헌연구에서는 소프트웨어 안전뿐만 아니라, 소프트웨어 공학에 대한 연구도 검토하여 소프트웨어 안전 교육에 필요한 기술 요소를 산출한다. 소프트웨어 안전 표준, 소프트웨어 공학기술과 소프트웨어 안전 관련 전문위원들의 자문에 기반을 두어 소프트웨어 안전 필요 기술 집합을 정의한다.

문헌 조사를 통하여 국내외 소프트웨어 안전 교육현황을 조사하고, 국내 소프트웨어 안전 교육의 문제점을 도출한다.

설문조사와 전문가 심층인터뷰를 통하여, 소프트웨어 안전 관련 재직자의 역량과 수요를 조사하고, 문헌연구에서 나타타지 않은 소프트웨어 안전 산업 및 소프트웨어 안전 교육에 대한 문제점을 도출한다.

문헌연구와 설문조사, 전문가 심층인터뷰를 바탕으로 소프트웨어 안전 교육 커리큘럼을 마련하고 전문가 자문을 통하여 커리큘럼의 실용성을 검증한다.

제2장 이론적 배경

본 장에서는 본 사업에서 조사하고자 하는 소프트웨어 안전 기술과 역량 및 이에 관련된 교육과정 수립에 앞서, 필요한 소프트웨어 안전 관련 배경지식을 설명하고자 한다.

소프트웨어 품질에 비해 소프트웨어 안전은 이론적 연구도 부족하며, 소프트웨어 공학처럼 대학 과정으로도 찾아보기 어렵다. 지금까지는 안전이 중요한 시스템에서 일부 다루고 있었으나, 소프트웨어가 점점 많은 산업 분야에 쓰이면서 소프트웨어 안전 교육이 전 산업분야에 필요하게 되었다. 소프트웨어 안전에 대한 교육을 위해서는 소프트웨어 안전이 무엇인지 정확히 정의하고, 필요한 기술요소는 무엇인지 정리가 필요하다. 소프트웨어 안전에 대한 정의를 하고 소프트웨어 공학 기술, 산업 도메인별 소프트웨어 안전 기술, 정부 주도의 소프트웨어 안전 관련 역량 기준을 검토한다. 다양한 도메인의 모델들과 연구모델과 표준들 중에서 가장 체계화되고 안정적으로 적용되는 사례인 SWEBOK, 철도 소프트웨어 안전 기술, 미연방에너지국 안전 소프트웨어 품질 보증 기능 영역 자격 기준에 대해 정리해 봄으로써 소프트웨어 안전에 필요한 기술요소가 무엇인지 기술하도록 하겠다.

첫 번째로 소프트웨어 안전의 정의를 품질, 보안과 비교하여 설명한다.

두 번째로 소프트웨어 공학의 주요 기술들은 소프트웨어 안전 기술 집합의 가장 기초적 기술로, 소프트웨어 안전 교육 내용 구성을 위해 소프트웨어 공학의 주요 기술들에 대하여 살펴본다. 그 사례로 소프트웨어 공학의 주요 기술 집합 중 가장 체계적으로 정립되고 사용되고 있는 모델이 SWEBOK을 설명하겠다.

세 번째로 안전 필수 시스템을 개발에 참여하는 참여자의 역할과 기술에 대해 설명하도록 하겠다. 현재 국제적으로 다양한 도메인에서 소프트웨어 안전 관련 필요 기술 집합에 대한 정의와 적용들이 이루어지고 있다. 이들 중, 가장 체계적으로 개발 프로세스와 참여자의 역량 기술을 묘사하고 있는 분야인 철도 소프트웨어 분야의 사례를 설명하고자 한다. 다음 사례로 국가 차원에서 구체화되어 정의되고 적용되고 있는 소프트웨어 안전 관련 대표적인 역량모델 중 하나인 미연방에너지국 안전 소프트웨어 품질 보증 기능 영역 자격 기준을 살펴보도록 한다.

마지막으로 안전 필수 소프트웨어 개발에 필요한 역할 및 기술에 대해 소프트웨어 공학 기술과 안전 표준을 비교 정리하여 소프트웨어 안전 교육을 위한 기술 집합을 만들고 산학연 전문가 자문을 받아 검증하겠다.

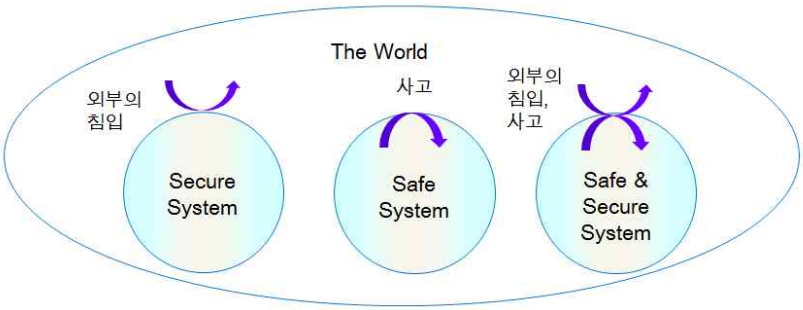
제1절 소프트웨어 안전의 정의

소프트웨어 안전에 대해 국내에서는 정확히 정의하고 있지 않으며, 품질, 보안, 안전과의 차이도 크게 인지하지 못하고 있다. 소프트웨어 품질은 ISO 25010(2011) 모델에 의하면, 기능성, 신뢰성, 효율성, 사용성, 보안, 호환성, 유지보수성, 이식성을 가지고 있어야 한다. ISO 25010는 ISO 9126에 비해 호환성과 보안을 추가했다는 것이 다르며, 보안에 대한 중요성이 강조된 것이라고도 볼 수 있다. 이중 신뢰성은 성숙성(Maturity), 고장방지능력(Fault Tolerance)과 복구성(Recoverability) 등의 특성으로 안전과 개념을 혼동하기도 한다. 성숙성은 소프트웨어 오류로 인한 고장을 피해가는 능력으로 소스코드의 Mean Time to Failure(MTTF)로 측정한다. 고장방지능력은 소프트웨어가 오류가 있어도 어느 정도 작동이 가능한 능력이다. exception handling 이나 중복을 통해서 방지가능하다. 복구성은 성능, 데이터, 시간에 대해 검토할 필요가 있다. 성능이나 데이터 보장은 중복으로 구현이 가능하다.[5] 이러한 신뢰성은 안전을 위해 필요한 특성이며, 신뢰성을 보장하기 위한 소프트웨어적 기법은 안전성을 보장하기 위한 기법으로 사용이 가능하다. 그러나 안전성 보장을 위해는 추가적인 분석과 구현이 필요하다.

소프트웨어 품질에 대한 인식은 국내에 비교적 많이 알려져, 품질 향상을 위한 SPICE(ISO15504), CMM(Capability maturity Model), CMMI(CMM 통합모델) 등의 품질 평가 모델을 사용하고 있다.

소프트웨어 보안은 소프트웨어 시스템 밖에서 침입을 방지하여 소프트웨어가 올바르게 동작하도록 하는 것이며, 안전은 소프트웨어 내에서 문제를 발생하지 않고 소프트웨어가 올바르게 동작하도록 것이다. [6]

[그림 2-1] 안전과 보안의 차이



자료 : C. Warren Axelrod , Engineering Safe & Secure Software systems

5) F. Losavio et, Quality Characteristics for SW Architecture, 2003.3

6) C. Warren Axelrod , Engineering Safe & Secure Software systems, pp61, Artech House, 2013

좋은 품질의 소프트웨어는 기술적 요구사항에 따라 오류 없이 작동하는 시스템을 구현한다. 품질 표준인 ISO 9126에 따르면 품질 특성 중에 안전과 특히 관련 있는 특성은 신뢰성(Reliability)이며, 기능성에 보안성(Security)을 보장하도록 되어 있다.

품질, 보안, 안전은 시스템이 문제없이 작동한다는 사실은 동일하나, 목적, 문제의 원인과 처리 방법에서 다르게 볼 수 있다. 보안의 목적은 외부의 침입에서 보호이며, 안전의 목적의 재난 방지이다. 기본적으로 시스템이 문제없이 작동하기 위해서는 오류 없는 좋은 품질의 소프트웨어를 제작하여야 하며, 보안과 안전을 지키기 위한 기술, 기법이 들어간다. 품질, 보안, 안전은 서로 다른 목적을 가지고 있으나, 서로의 목적 달성을 위해서는 각자의 기능 충족이 필요하다. 즉, 품질의 요소에 보안과 안전이 있으며, 안전 달성을 위해서는 보안이 된 좋은 품질의 소프트웨어가 구현되어야 한다.

국가안전관리기본계획에 의하면 안전이란 자연적 혹인 인적・인위적 위험요인이 없거나, 이러한 위험 요인에 대한 충분한 대비가 되어 있는 상태이다. IEEE 표준 1228-1994에 따르면 소프트웨어의 안전이란 소프트웨어 위험요소 제거를 통해 소프트웨어 오류로 인한 시스템의 사고를 예방하는 것이다. 소프트웨어 안전이란 수용 가능한 사고 위험 수준으로 소프트웨어가 시스템과 운영 환경에서 안전하게 실행되도록 하는 일련의 프로세스이다. 소프트웨어 안전은 수용할 수 있는 사고 위험을 가진 소프트웨어의 기능을 구현함으로써 보장된다. 안전을 보장하기 위해서는 하드웨어와 사람, 다른 소프트웨어와의 인터페이스 시 사고를 일으킬 수 있는 위험원은 파악되어야 하고, 제거되어야 한다. 소프트웨어의 안전은 시스템의 안전의 일부분이다. [7]

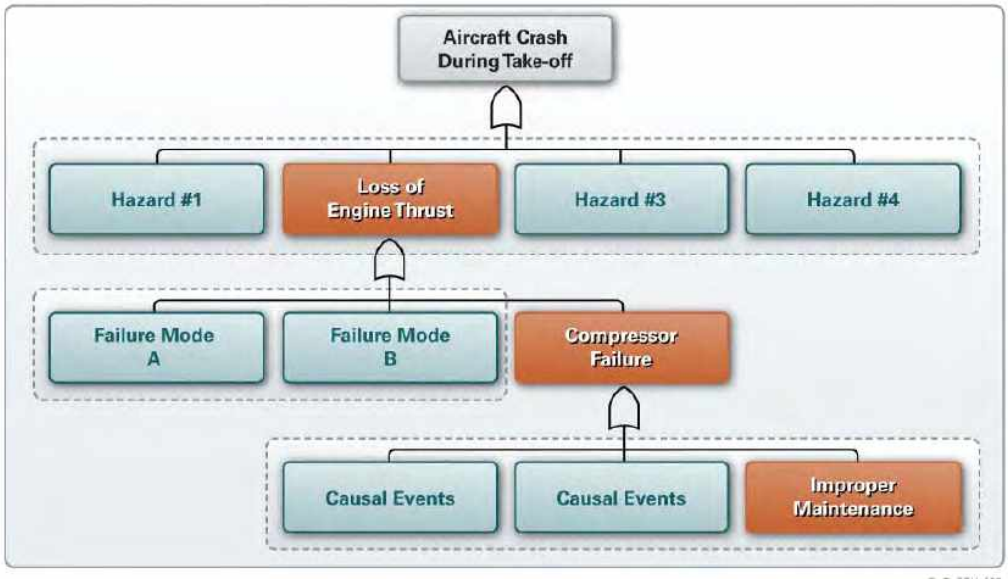
시스템 안전이란 기능, 시간, 코스트 등의 제약조건하에서 달성되는 안전의 가장 적합한 정도를 말한다. 시스템의 모든 수명단계를 통해서 해저드(Hazard, 위해도)가 최소화되도록 시스템 안전관리 및 시스템 안전공학을 정확하게 적용하는데 의해서 달성된다.[8] 안전과 밀접히 연관된 소프트웨어들은 자동차, 항공, 철도, 원자력 등 하드웨어가 점점 소프트웨어화되어 가는 과정의 시스템에 많아, 시스템의 안전을 고려하기 위해서는 하드웨어와 소프트웨어, 둘 간의 인터페이스를 모두 고려해야 한다.

시스템 안전 보장하는 활동 중에 중요한 것은 위험에 대한 분석이다. 다음은 사고 사례로써 해저드는 시스템 단위의 위험으로, 해저드에 어떤 행위가 가해질 때 사고로 이어지게 되는 사례를 분석한 그림이다. 시스템의 안전을 보장하고 향상시키기 위해서 해저드 분석은 필수요소이며, 해저드에 대한 분석은 안전 전문가에 의해 일어나야 한다.

---

7) Clifton Ericson II, “Software Safety Primer” p4, CreateSpace Inc, 2013
8) 최상복, 시스템 안전, 산업안전대사전, 2004,5

[그림 2-2] 사고 위험 사례



자료 : 미국 Department of Defence, Software System Safety Engineering Handbook p40

소프트웨어의 안전은 시스템 안전과 마찬가지로 소프트웨어 개발 전 공정에서 고려되어야 한다. 요구분석 단계에서 가장 중요한 것은 위험을 분석하고, 제거할 수 있는 계획을 세우는 것이다. 디자인 단계에서는 분석된 위험을 소프트웨어가 수용할 수 있는 단계까지 디자인하고, 소프트웨어 안전을 지키기 위한 기술을 사용하여 개발하고, 테스트하고, 검증하는 단계가 필요하다.

소프트웨어 안전을 지키기 위한 기술은 기본적으로 소프트웨어 공학 기술을 많이 차용하고 있다. 소프트웨어 위험 분석단계에서는 기존의 안전공학의 분석 기술을 사용하나, 하드웨어와 소프트웨어의 차이점을 고려해야 한다. 디자인단계 부터는 위험 단계에 따라 소프트웨어 공학기술 중 어느 정도 엄격한 수준의 기술을 쓸 것인지가 결정된다.

제2절 소프트웨어 안전에 적용되는 소프트웨어 공학 기술 영역

1. Software Engineering Body of Knowledge (SWEBOK) 개요

소프트웨어 공학의 주요 기술들은 소프트웨어 안전 기술 집합의 가장 기초적 기술이다. IEEE Computer Society의 프로젝트 결과인 Software Engineering Body of Knowledge (SWEBOK)[9]은 세계적으로 소프트웨어 공학에 대해 일관성 있는 정보를 전달하고, 소프트웨어 공학의 범위를 명확히 정하고 전산학, 수학, 프로젝트 관리와 같은 다른 활동과의 차이를 명백히 설명한다.

SWEBOK는 소프트웨어 공학의 내용을 설명하기 위해서, 소프트웨어 공학의 지식체계에 대한 쉬운 하향식 접근방법을 제공하고, 인증이나 자격증의 교과 과정을 위한 기반을 제공하기 위한 목적으로 작성되었다. 15개의 소프트웨어 공학 지식 영역(knowledge area)이 체계적으로 분류, 정리되어 있는데, SWEBOK에 정의된 소프트웨어 지식 영역 및 지식 영역 하위의 세부주제들이 소프트웨어 안전 확보를 위한 기술 집합 구성에 유용한 참조가 될 수 있다. 15가지의 지식 영역은 아래와 같다.

(1) Software Requirements (소프트웨어 요구사항)
(2) Software Design (소프트웨어 설계)
(3) Software Construction (소프트웨어 구현)
(4) Software Testing (소프트웨어 테스팅)
(5) Software Maintenance (소프트웨어 유지보수)
(6) Software Configuration Management (소프트웨어 형상 관리)
(7) Software Engineering Management (소프트웨어 공학 관리)
(8) Software Engineering Process (소프트웨어 공학 프로세스)
(9) Software Engineering Models and Methods (소프트웨어 공학 모델과 방법론)
(10) Software Quality (소프트웨어 품질)
(11) Software Engineering Professional Practice (소프트웨어 공학 전문가 기량)
(12) Software Engineering Economics (소프트웨어 공학 경제학)
(13) Computing Foundations (컴퓨팅의 기반)

---

2) P. Bourque and R.E. Fairley, eds.,,IEEEComputerSociety,2014;www.swebok.org.

(14) Mathematical Foundations (수학적 기반)

(15) Engineering Foundations (공학적 기반)

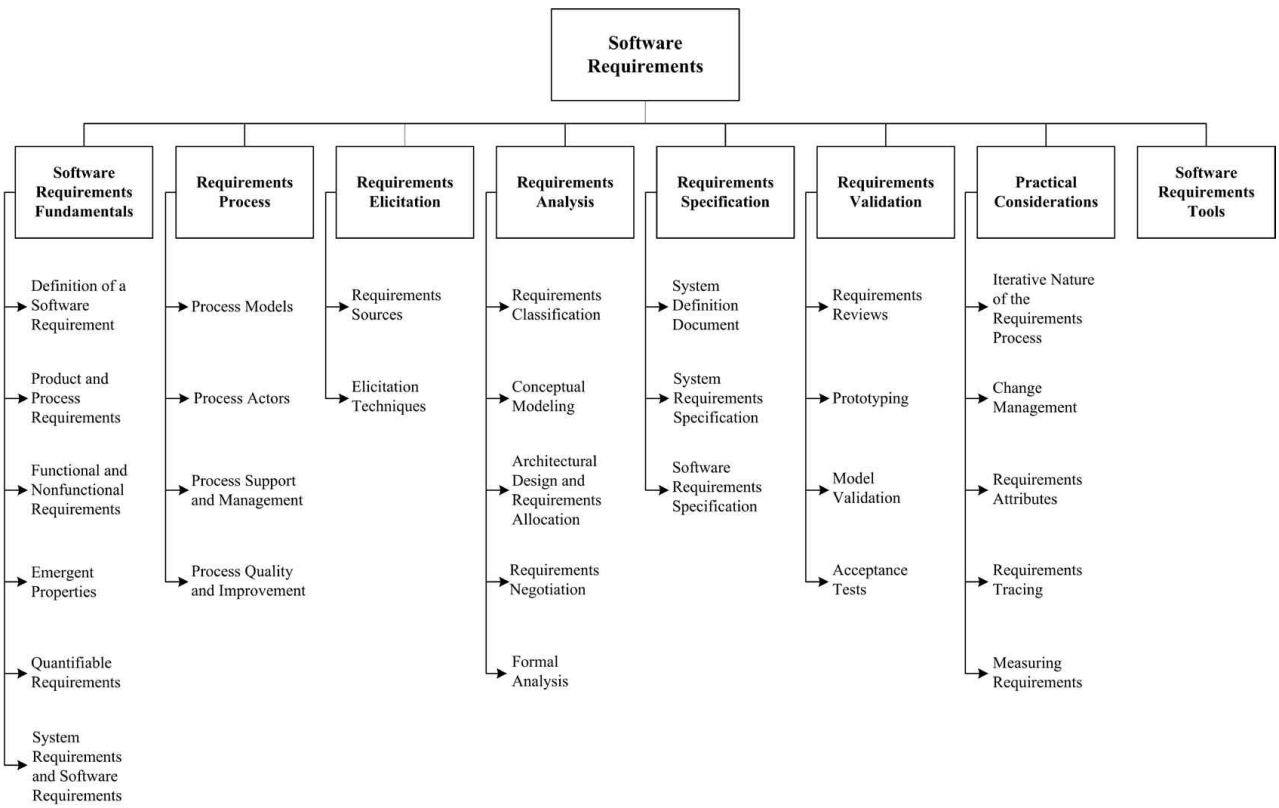
2. SWEBOK 지식 영역

SWEBOK의 15가지 지식영역 중 소프트웨어 안전과 밀접히 관련되어 있는 11가지 지식영역에 대해 정리했다. 소프트웨어 공학 경제학, 컴퓨팅의 기반, 수학적 기반, 공학적 기반은 제외하였다. 참고로 소프트웨어 공학 경제학은 소프트웨어 비용을 결정하고, 소프트웨어 공학에 대한 미시 경제학의 기본 개념을 적용하고, 소프트웨어 공학 의사 결정에 경제적 분석을 활용하는 영역이다. 컴퓨팅 기반은 프로그램 언어, 알고리즘, 시스템, 컴퓨터 구조, OS, 네트워크 등 컴퓨터공학의 전반적인 영역이다.

1) 소프트웨어 요구사항

소프트웨어 요구사항 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-3] 소프트웨어 요구사항 분야 필요 주제



소프트웨어 요구사항 분야는 요구사항 프로세스, 도출, 분석, 명세, 검증, 변경관리에 대한 내용을 다루고 있다.

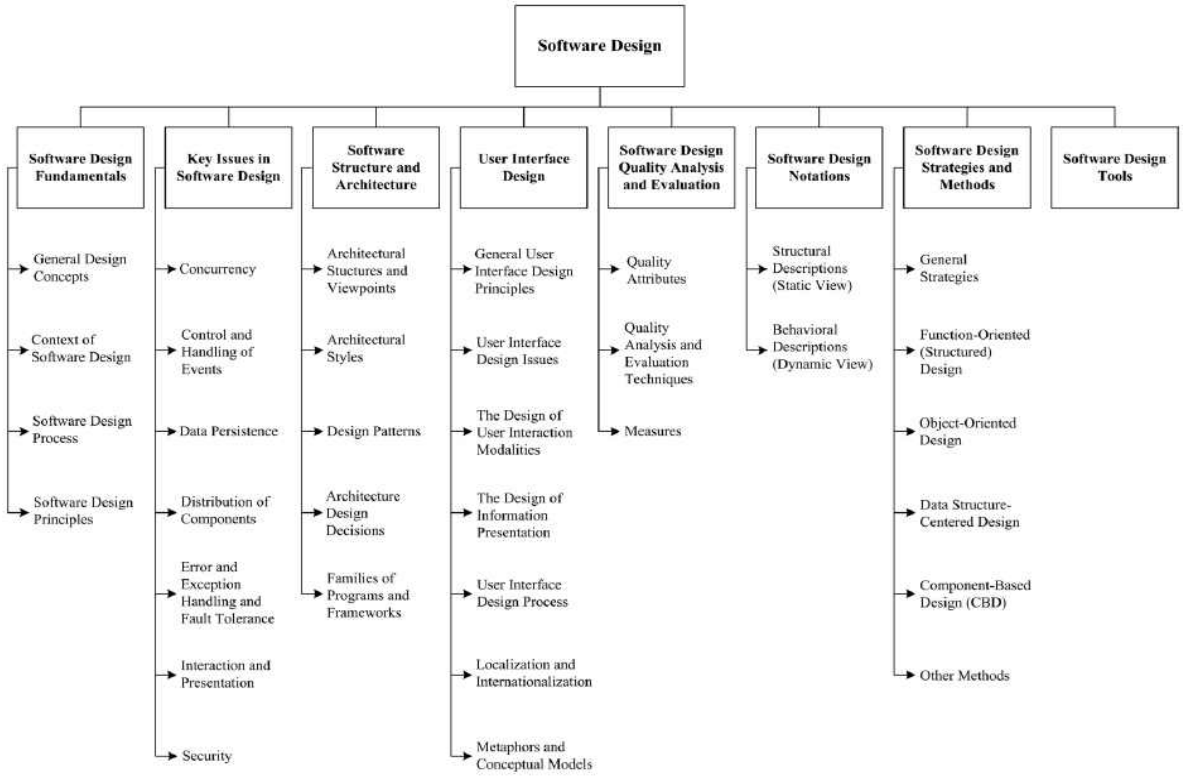
- Software Requirements Fundamentals - 소프트웨어 요구사항의 정의, 생산 및 실행, 기능/비기능 요구사항, 시스템/소프트웨어 요구사항 분류
- Requirements Process - 프로세스 모델, 프로세스 담당자, 프로세스 지원 및 관리, 프로세스 품질 관리 등
- Requirements Elicitation - 요구사항 도출 기술
- Requirements Analysis - 요구사항 분류, 개념 모델링, 아키텍처 디자인 및 요구사항 할당, 요구사항 협상, 정규 분석 등
- Requirements Specification - 시스템 정의서, 시스템/소프트웨어 요구사항 명세
- Requirements Validation - 요구사항 리뷰, 프로토타이핑, 모델 확인, 인수 테스트
- Practical Considerations - 변경 관리, 요구사항 속성, 요구사항 추적, 요구사항 측정

2) 소프트웨어 설계

소프트웨어 설계 분야는 소프트웨어 설계 원칙, 설계 방법, 소프트웨어 아키텍처, 사용자 인터페이스 디자인, 소프트웨어 디자인 품질, 도구 등을 포함한다.

소프트웨어 설계 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-4] 소프트웨어 설계 분야 필요 주제



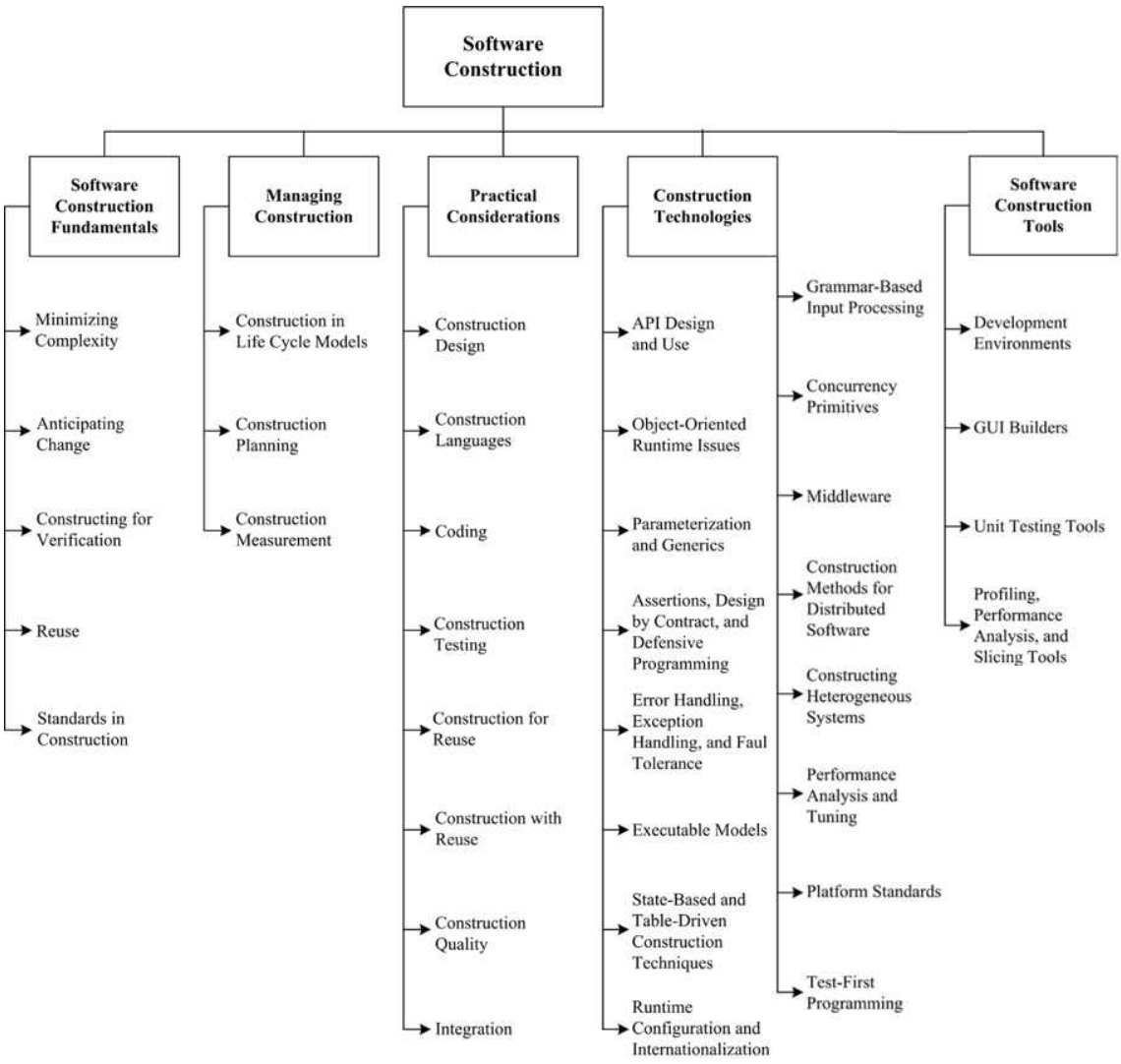
소프트웨어 설계 분야는 설계를 위한 기초 지식과 소프트웨어 구조, UI, 품질 속성, 설계 표현 방식, 설계 전략, 도구 등을 다루고 있다.

- Software Design Fundamentals – 디자인 일반 개념, 소프트웨어 디자인의 배경, 소프트웨어 디자인 프로세스, 소프트웨어 디자인 원칙
- Key Issues in Software Design – 병행 수행, 이벤트 컨트롤, 데이터의 일관성, 컴포넌트 배포, 오류 및 예외 처리, 보안 등
- Software Structure and Architecture – 아키텍처 상의 구조 및 관점, 스타일, 디자인 패턴, 아키텍처 디자인의 결정, 프로그램 군 및 프레임워크
- User Interface Design – GUI 일반 원칙, 유저 인터페이스 쟁점, 유저 인터페이스의 디자인 양식, 정보 표현 방식, 언어 문제, 은유 및 개념 모델
- Software Design Quality Analysis and Evacuation – 품질 속성, 품질 분석 및 평가 기술, 측정
- Software Design Notations – 구조적 표현(Static View) 및 행위적 표현(Dynamic View)
- Software Design Strategies and Methods – 일반 전략, 함수지향 디자인, 객체지향 디자인, 데이터 지향 디자인, 컴포넌트 지향 디자인 등
- Software Design Tools – 설계 도구

3) 소프트웨어 구현

소프트웨어 구현 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-5] 소프트웨어 구현 분야 필요 주제



[소프트웨어 구현 분야 필요 주제]

소프트웨어 구현 분야는 구현을 위한 기술과 도구에 대해 다루고 있다.

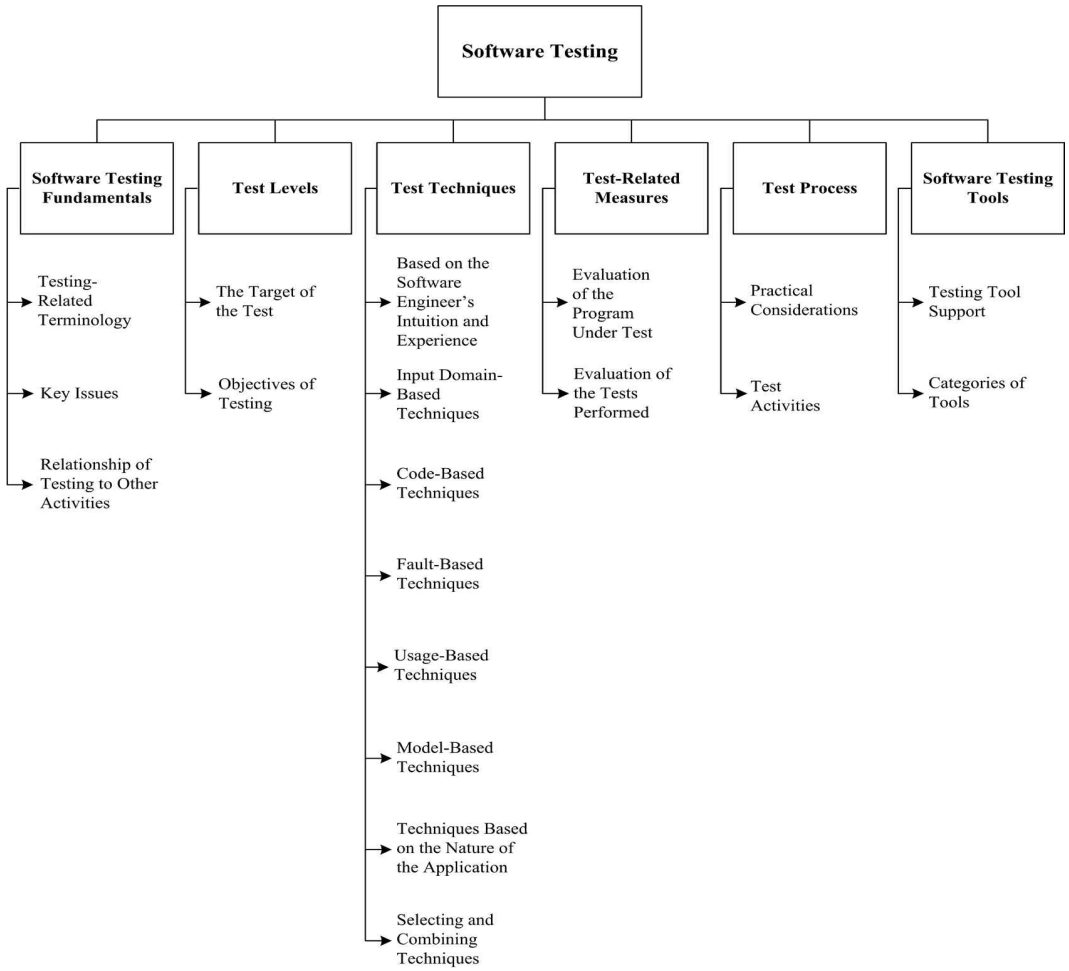
- Software Construction Fundamentals - 복잡도 최소화, 변경 예측, 재사용 등
- Managing Construction - Life-Cycle 모델, 조립 계획 및 측정
- Practical Considerations - 조립 디자인, 언어, 코딩, 테스팅, 재사용 관련, 품질, 통합 등
- Construction Technologies - API 디자인 및 사용, 객체지향 런타임 이슈, 파라미터 및 일반화, Assertion, 오류 및 예외 처리, 실행 가능 모델, Table-Driven 조립 기술, 문법 기준 입력 프로세스, 미들웨어 등

◆ Software Construction Tools - 개발 환경, GUI 빌더, 단위 테스트 도구, 프로파일링 및 성능 분석 도구

4) 소프트웨어 테스팅

소프트웨어 테스팅 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-6] 소프트웨어 테스팅 분야 필요 주제



[소프트웨어 테스팅 분야 필요 주제]

소프트웨어 테스트 분야는 테스팅 목표, 기술, 측정 방법, 도구 등을 포함한다.

◆ Software Testing Fundamentals - 테스팅 관련 용어, 핵심 이슈, 테스팅과 타 활동과의 관계

◆ Test Levels - 테스트 대상, 테스팅 목표

◆ Test Techniques - 소프트웨어 엔지니어의 직관 및 경험, 도메인, 코드, 오류, 사용예, 모델 등에서의 기술, 기술의 선택과 조합

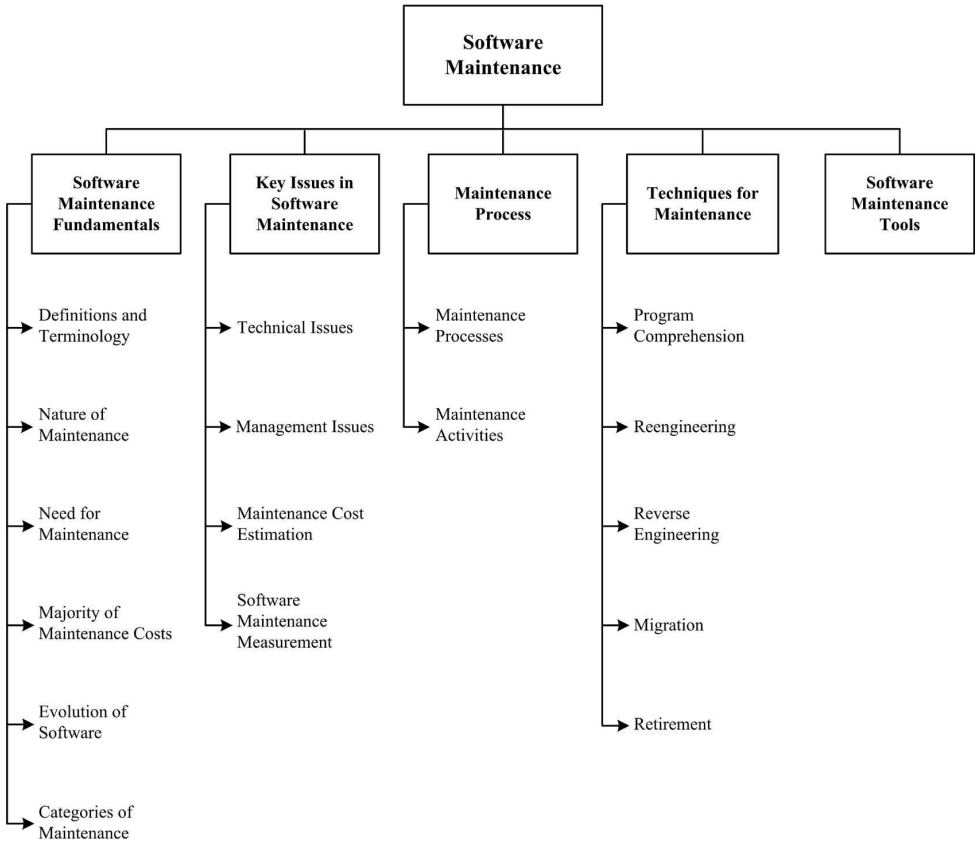
◆ Test-Related Measures - 테스트 상에서의 프로그램 측정, 테스트 성능 측정

- Test Process - 실사용에서의 고려사항, 테스트 활동
- Software Testing Tools - 테스트 지원 도구, 도구 카테고리

5) 소프트웨어 유지보수

소프트웨어 유지보수 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-7] 소프트웨어 유지보수 분야 필요 주제



[소프트웨어 유지보수 분야 필요 주제]

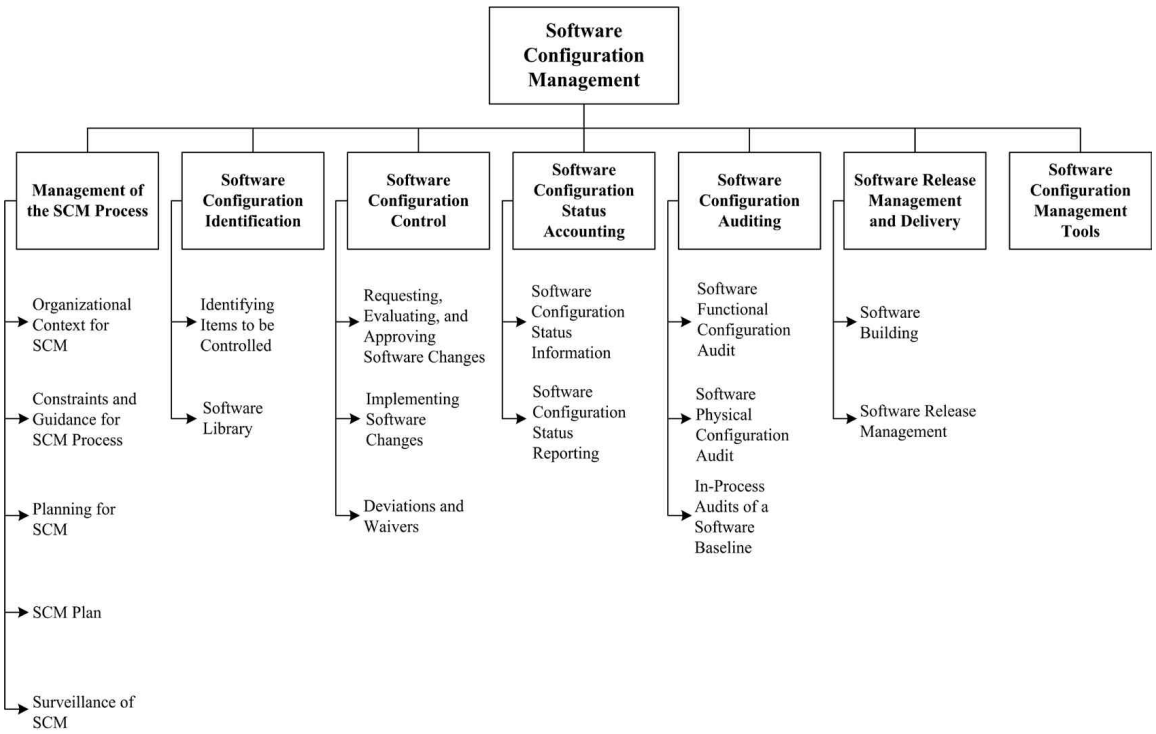
유지보수 분야는 유지보수를 위한 기술, 프로세스, 도구 등을 포함한다.

- Software Maintenance Fundamentals - 정의 및 용어, 유지 및 보수의 필요성, 비용, 측정, 유지 및 보수의 카테고리
- Key Issues in Software Maintenance - 기술적 이슈, 관리 이슈, 유지보수 비용 예측
- Maintenance Process - 유지 프로세스, 유지 활동
- Techniques for Maintenance - 프로그램 이해, 재공학, 역공학, 이관 및 종료
- Software Maintenance Tools

6) 소프트웨어 형상관리

소프트웨어 형상관리 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-8] 소프트웨어 형상관리 분야 필요 주제



[소프트웨어 형상관리 분야 필요 주제]

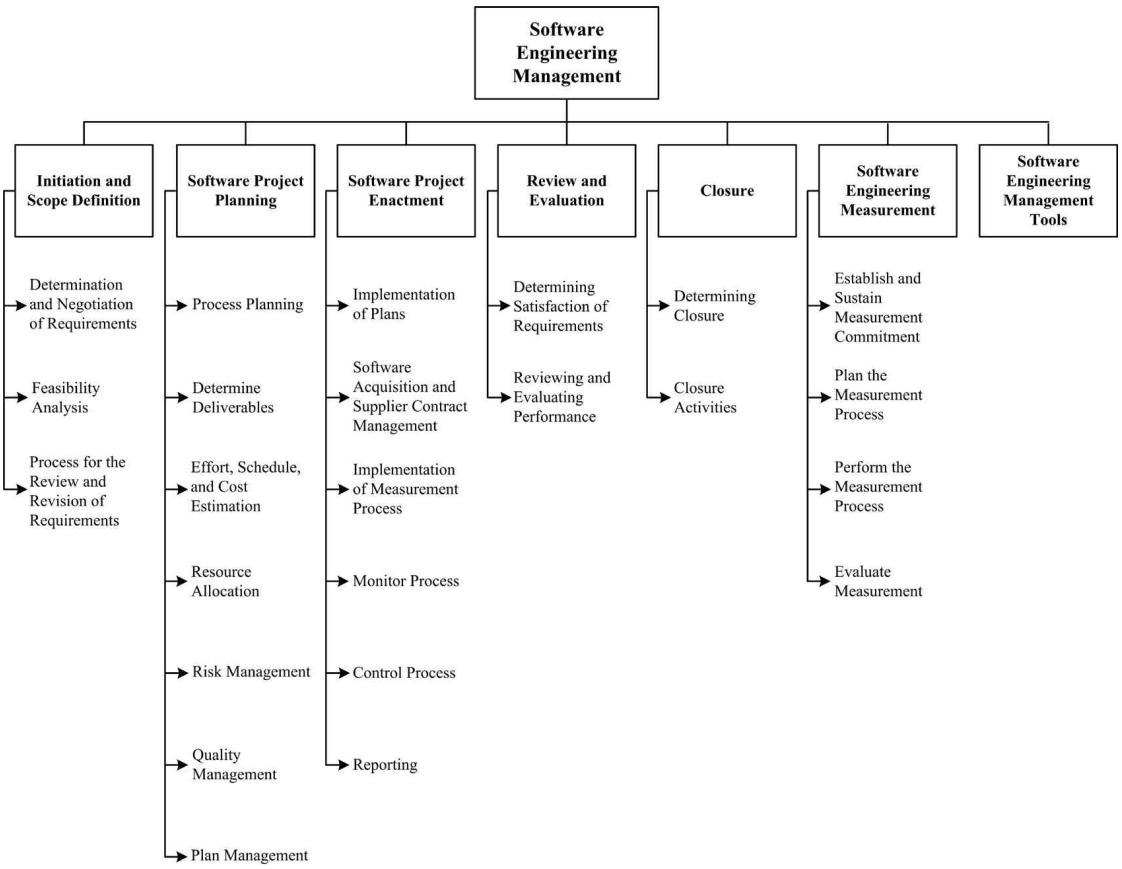
형상관리에서는 형상관리 프로세스와 관리 방법, 소프트웨어 빌드, 도구 등을 다루고 있다.

- Management of the SCM Process - 형상관리 콘텐츠의 구성, 형상관리 프로세스의 제약사항 및 가이드, 형상관리 계획, 형상관리 감시
- Software Configuration Identifications - 아이템 구분, 소프트웨어 라이브러리
- Software Configuration Control - 소프트웨어 변경에 대한 요청, 평가, 승인, 구현, 편차
- Software Configuration Status Accounting - 소프트웨어 설정 상태의 정보 및 보고
- Software Configuration Auditing - 소프트웨어 기본 설정 감사, 소프트웨어 물리적 설정에 대한 감사, 소프트웨어 베이스라인에 대한 감사
- Software Release Management and Delivery - 소프트웨어 빌드 및 릴리즈 관리
- Software Configuration Management Tools - 형상 관리 도구

7) 소프트웨어 공학 관리

소프트웨어 공학 관리 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-9] 소프트웨어 공학 관리 분야 필요 주제



[소프트웨어 공학 관리 분야 필요 주제]

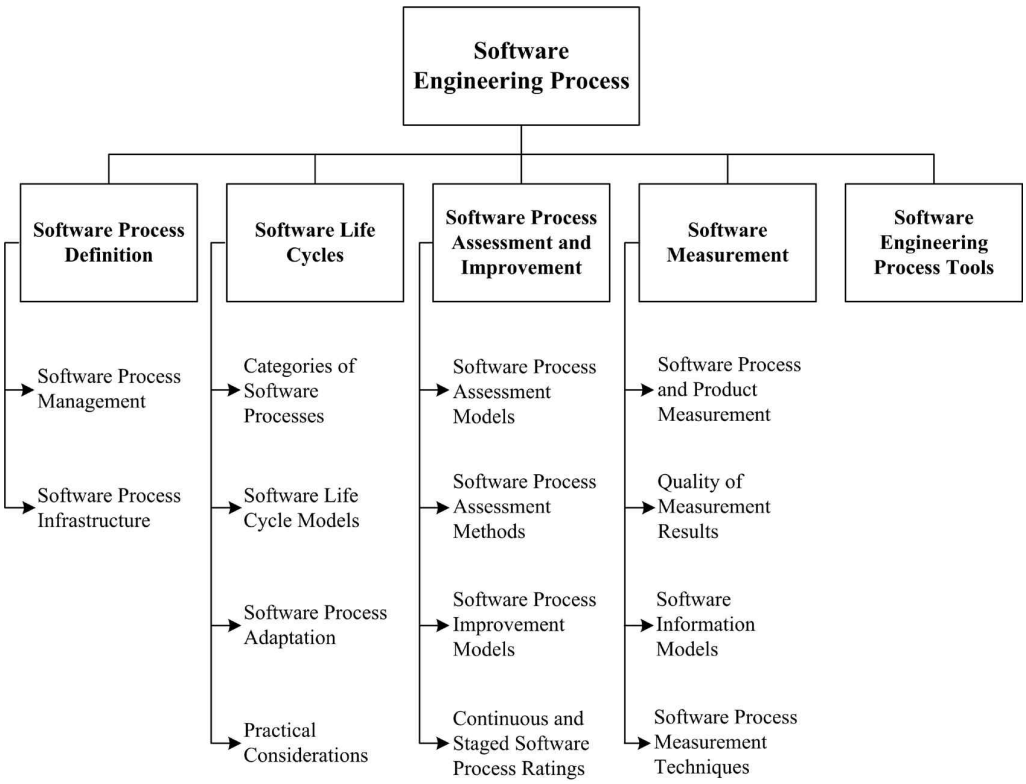
소프트웨어 공학 관리에서는 프로세스 전반에 대한 관리 방법과 측정방법을 설명한다.

- Initiation and Scope Definition - 요구사항에 대한 결정 및 협상, 실행 가능성 분석, 요구사항의 리뷰 및 수정사항에 대한 프로세스
- Software Project Planning - 프로세스 계획, 공수, 일정, 비용에 대한 예측, 리소스 할당, 위험 관리, 품질 관리, 계획 관리
- Software Project Enactment - 계획의 실행, 소프트웨어 인수 및 제공자 계약 관리, 측정 프로세스의 실행, 모니터 프로세스, 컨트롤 프로세스, 보고
- Review and Evaluation - 요구사항 만족도, 성능에 대한 리뷰 및 측정
- Closure - 종료 결정 및 해당 활동
- Software Engineering Measurement - 측정 프로세스의 계획 및 실행
- Software Engineering Management Tools - 관리 도구

8) 소프트웨어 공학 프로세스

소프트웨어 공학 프로세스 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-10] 소프트웨어 공학 프로세스 분야 필요 주제



[소프트웨어 공학 프로세스 분야 필요 주제]

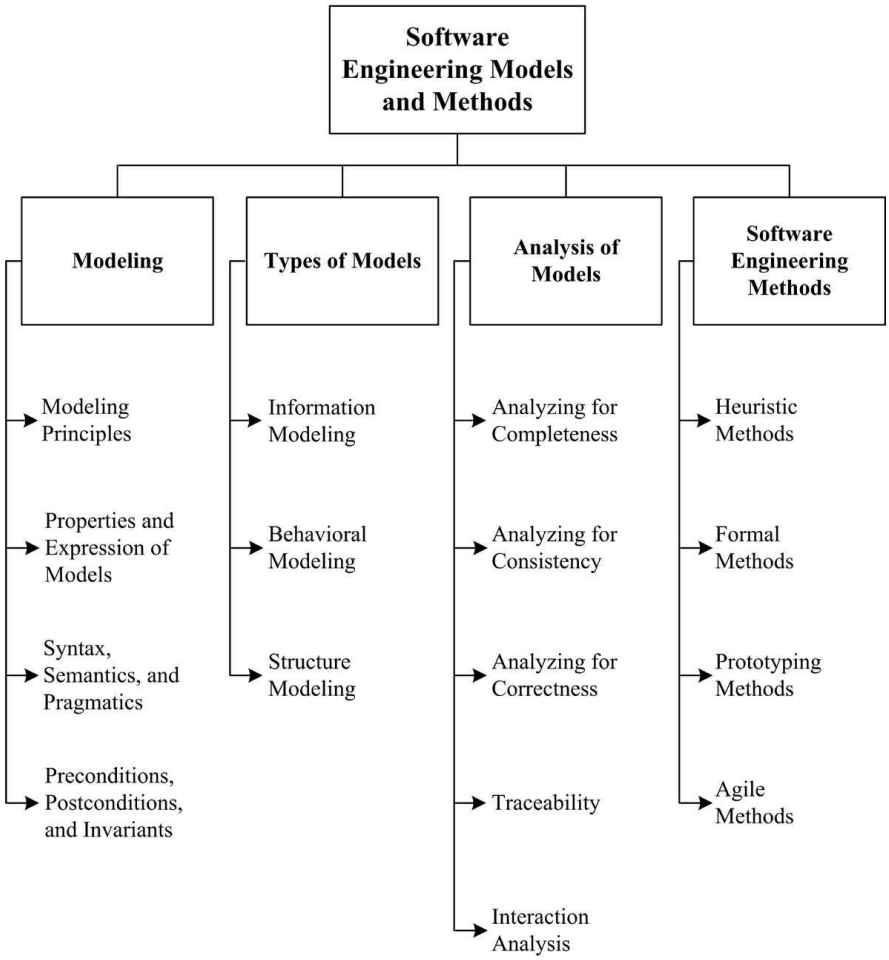
소프트웨어 공학 프로세스는 소프트웨어 생명주기, 프로세스 평가, 소프트웨어 평가, 프로세스 도구를 포함한다.

- Software Process Definition - 소프트웨어 프로세스 관리 및 기반 환경
- Software Life Cycles - 소프트웨어 프로세스의 카테고리, 모델, 적응, 실무에서의 고려사항
- Software Process Assessment and Improvement - 소프트웨어 프로세스 평가 모델, 평가 방법, 소프트웨어 프로세스 향상 모델
- Software Measurement - 소프트웨어 프로세스 및 제품 측정, 측정 결과의 품질, 소프트웨어 정보 모델, 소프트웨어 프로세스 측정 기술
- Software Engineering Process Tools

9) 소프트웨어 공학 모델과 방법론

소프트웨어 공학 모델과 방법론 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-11] 소프트웨어 공학 모델과 방법론 분야 필요 주제



[소프트웨어 공학 모델과 방법론 분야 필요 주제]

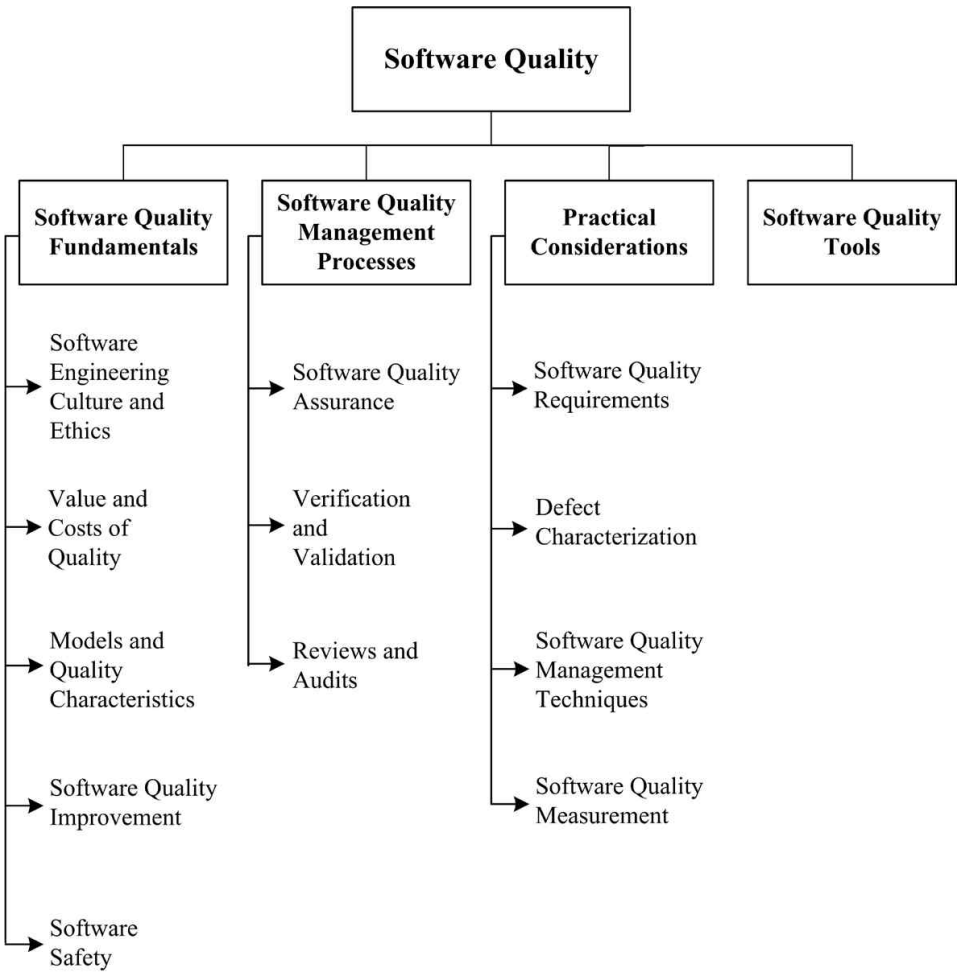
소프트웨어 공학 모델과 방법론은 모델링 방법, 모델링 유형, 모델 분석 등을 설명한다.

- Modeling - 모델링 원칙, 모델의 특징 및 표현, 문법, 의미 및 인용, 전제조건 및 사후조건, 불변 요소 등
- Types of Models - 정보 모델링, 행동 모델링, 구조 모델링
- Analysis of Models - 완전성 분석, 일관성 분석, 정확성 분석, 추적, 분석 상호작용
- Software Engineering Methods - 경험, 정규, 프로토타입, 애자일 방법 등

10) 소프트웨어 품질

소프트웨어 품질 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-12] 소프트웨어 품질 분야 필요 주제



[소프트웨어 품질 분야 필요 주제]

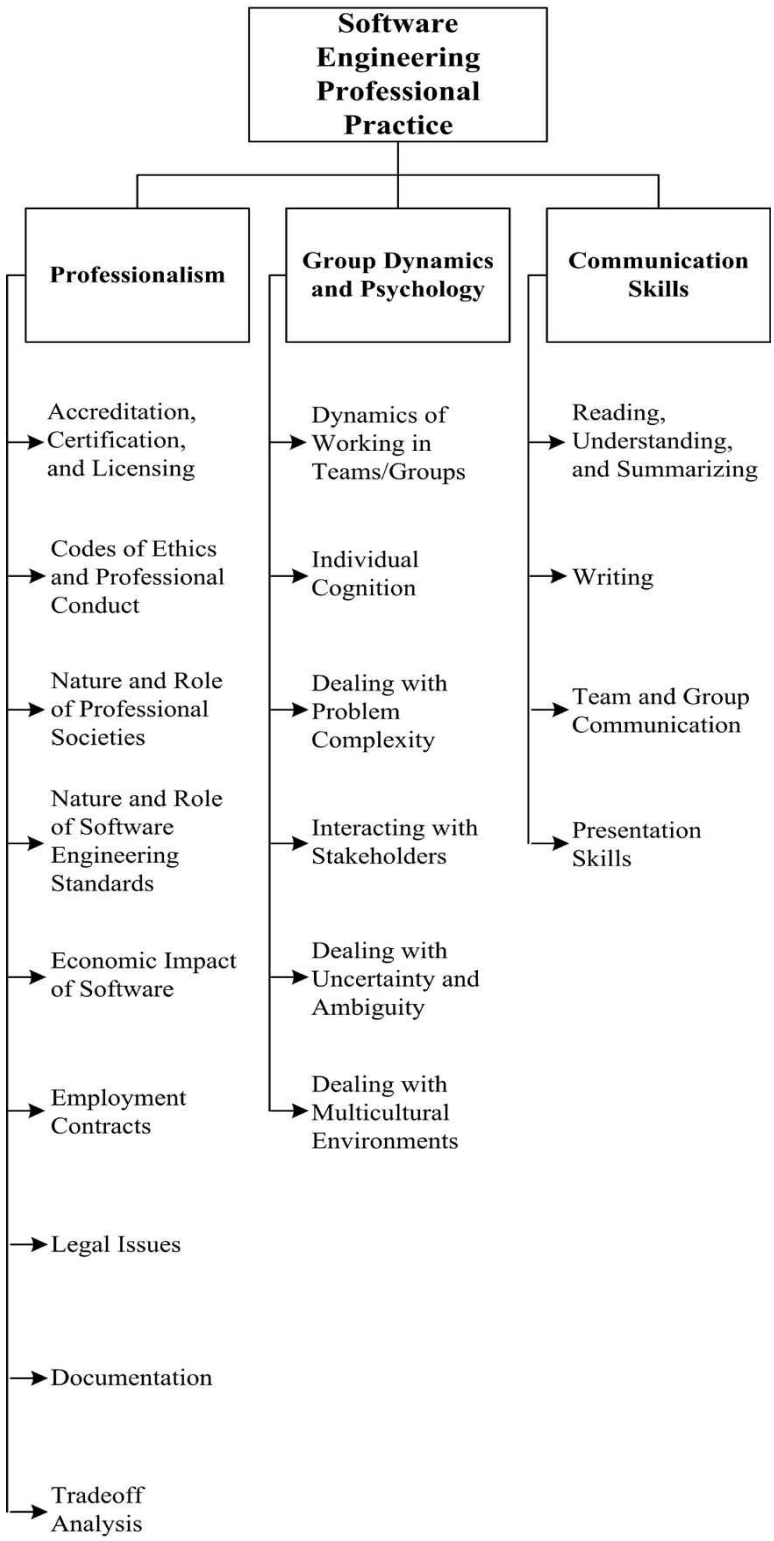
소프트웨어 품질 분야에서는 품질관리 프로세스와 품질관리 고려사항을 포함한다.

- Software Quality Fundamentals - 소프트웨어공학 문화 및 윤리, 품질의 가치 및 비용, 모델 및 품질 특징, 소프트웨어 품질 향상, 소프트웨어 안전
- Software Quality Management Process - 소프트웨어 품질 보증, 검증 및 확인, 리뷰 및 감사
- Practical Considerations - 소프트웨어 품질 요구사항, 결함의 특징, 소프트웨어 품질 관리 기술, 소프트웨어 품질 측정

11) 소프트웨어 공학 전문가 기량

소프트웨어 공학 전문가 기량 지식 영역에 대한 주제들을 세분화하면 다음과 같다.

[그림2-13] 소프트웨어 공학 전문가 기량 분야 필요 주제



[소프트웨어 공학 전문가 기량 분야 필요 주제]

소프트웨어 공학 전문가 기량 분야는 저작권, 고용 계약 등 법적인 문제와 팀과 같이 일하기 위한 소통 및 협동 방법에 대해 다룬다.

- Professionalism - 인가, 증명, 저작권, 윤리 및 전문적 행동에 대한 코드, 전문 사회에서의 역할, 소프트웨어의 경제적 영향, 고용 계약, 법적 이슈, 문서화, 트레이드오프 분석
- Group Dynamics and Psychology - 팀/그룹에서의 일 역학관계, 개인적 인식, 문제점 처리, 이해관계자와의 소통, 불확실성 및 모호성 처리, 다문화 환경에 대한 처리
- Communication Skills - 읽기, 이해 및 요약, 쓰기, 팀/그룹 커뮤니케이션, 발표 기술

제3절 소프트웨어 안전 분야 재직자 역량과 기술 표준 사례

1. 산업 도메인 분야 표준 사례

소프트웨어 안전 분야 재직자 역할과 기술에 대해 분석하기 위해, 사례로 체계적으로 안전 필수 시스템(Safety Critical System) 개발 프로세스와 참여자의 역량 기술을 묘사하고 있는 철도 소프트웨어 분야에서 정의한 소프트웨어 안전 기술에 대하여 설명하고자 한다.

1) 철도 분야 소프트웨어 안전 기술 개요

철도 분야에서는 IEC 62279[10] 표준 내 Annex B에 주요 소프트웨어 담당자 역할 및 책임(Key software roles and responsibilities)이 명시되어 있다. 요구사항 관리자(Requirements Manager), 설계자(Designer), 구현자(Implementer), 테스터(Tester), 검증자(Verifier), 통합자(Integrator), 확인자(Validator), 평가자(Assessor), 프로젝트 관리자(Project Manager), 형상 관리자(Configuration Manager), 품질보증 관리자(Quality Assurance Manager), 검토자(Reviewer)와 같은 12가지 역할에 대해서 각 역할이 수행해야 할 책임사항과 각 역할 수행에 필요한 주요 역량에 대한 내용을 포함하고 있다. 철도 분야 소프트웨어 안전 기술 집합 구성 시 IEC 62279 표준 Annex B를 참조할 수 있다.

2) 철도 분야 직무별 역할 명세

철도 분야에서 정의하는 안전 필수 시스템 개발 참여자의 역할 및 각 역할에 따라 수행해야 할 책임과 역할 수행을 위해 갖추어야할 역량에 대해 설명한다.

(1) 요구사항 관리자(Requirements Manager) 역할 명세

요구사항 관리자는 요구사항 명세에 책임이 있으며, 요구사항 추적성을 수립하고 유지해야 한다. 주요 역량으로는 요구공학 역량과 도메인의 안전 특성에 대한 경험, 적용 가능한 규제에 대한 지식이 있어야 한다.

---

10) IEC 62279 Edition 2.0, Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems, 2015.

<표 2-1> 요구사항 관리자 책임 및 역량

역할: 요구사항 관리자	
책임 (Responsibilities)	1) 소프트웨어 요구사항을 명세에 책임이 있다. 2) 소프트웨어 요구사항 명세를 소유해야 한다. 3) 시스템 수준 요구사항으로부터 및 시스템 수준 요구사항으로의 추적성을 수립하고 유지해야 한다. 4) 명세와 소프트웨어 요구사항이 상태, 버전, 권한 부여 상태를 포함하여 변경 및 형상관리 하에 있음을 보증해야 한다. 5) (사용자 요구사항과 응용 프로그램의 최종 환경을 참조하여) 소프트웨어 요구사항 명세 내 일관성과 완전성을 보증해야 한다. 6) 소프트웨어 요구사항 문서를 개발하고 유지해야 한다.
주요 역량 (Key competences)	1) 요구 공학 역량이 있어야 한다. 2) 응용 프로그램의 도메인에 경험이 있어야 한다. 3) 응용 프로그램 도메인의 안전 특성에(safety attributes) 대한 경험이 있어야 한다. 4) 시스템의 전반적 역할과 응용 프로그램의 환경을 이해해야 한다. 5) 분석적 기술과 결과를 이해해야 한다. 6) 적용 가능한 규제를 이해해야 한다. 7) IEC 62279의 요구사항을 이해해야 한다.

(2) 설계자(Designer) 역할 명세

설계자는 소프트웨어 요구사항을 설계로 변경 시, 설계 방법과 지원도구를 정의하고, 적절한 설계원리와 표준을 적용해야 하는 책임이 있다. 주요 역량은 안전 설계 원리, 설계 분석 및 설계 테스트 방법론이다. 물론 문제 도메인과 표준 조항을 이해해야 한다.

<표 2-2> 설계자 책임 및 역량

역할: 설계자	
책임	1) 명세된 소프트웨어 요구사항을 허용 가능한 해법(acceptable solutions)으로 변환해야 한다. 2) 아키텍처와 흐름을 따른 해법을 가져야 한다. 3) 설계 방법과 지원 도구를 정의하고 선택해야 한다. 4) 적절한 설계 원리와 표준을 적용하여야 한다. 5) 적합한 곳에 컴포넌트 명세를 개발하여야 한다. 6) 명세된 소프트웨어 요구사항으로의 및 소프트웨어 요구사항으로부터의 추적성을 유지해야 한다. 7) 설계 문서를 개발하고 유지해야 한다. 8) 설계 문서가 변경 및 형상관리 하에 있음을 보증해야 한다.
	1) 적용 분야에 적합한 공학에 역량이 있어야 한다.

주요 역량	2) 안전 설계 원리에 역량이 있어야 한다. 3) 설계 분석 및 설계 테스트 방법론에 역량이 있어야 한다. 4) 주어진 환경에서 설계 제약사항 내에서 일할 수 있어야 한다. 5) 문제 도메인을 이해할 수 있는 역량이 있어야 한다. 6) 하드웨어 플랫폼, 운영 체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항들을 이해해야 한다. 7) IEC 62279의 관련 조항/부조항들을 이해해야 한다.

(3) 개발자(Implementer) 역할 명세

개발자는 안전 설계 원리를 적용하여 설계 해법을 데이터/소스 코드/다른 설계 표현으로 변환하고 소프트웨어를 대상 기계에 통합하는 책임을 가지고 있다. 주요 역량은 개발 언어와 지원 도구 역량과, 하드웨어 관련 제약사항과 표준 조항을 알고 있어야 한다.

<표 2-3> 개발자 책임 및 역량

	역할: 개발자
책임	1) 설계 해법을 데이터/소스 코드/다른 설계 표현으로 변환해야 한다. 2) 소스코드를 실행 가능한 코드/다른 설계 표현으로 변환해야 한다. 3) 안전 설계 원리를 적용해야 한다. 4) 지정된 데이터 준비/코딩 표준을 적용해야 한다. 5) 중간 산출물을 검증하기 위한 분석을 수행해야 한다. 6) 소프트웨어를 대상 기계에 통합해야 한다. 7) 적용된 방법, 데이터 타입, 리스팅을 포함하는 구현 문서를 개발하고 유지해야 한다. 8) 설계로의 또한 설계로부터의 추적성을 유지해야 한다. 9) 변경 및 형상관리 하에 있는 생성 또는 변경된 데이터/코드를 유지해야 한다.
주요 역량	1) 적용 분야에 적합한 공학에 역량이 있어야 한다. 2) 개발 언어와 지원 도구에 역량이 있어야 한다. 3) 지정된 코딩 표준과 프로그래밍 스타일을 적용할 수 있어야 한다. 4) 하드웨어 플랫폼, 운영 체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항들을 이해해야 한다. 5) IEC 62279의 관련 조항/부조항들을 이해해야 한다.

(4) 테스터(Tester) 역할 명세

테스터는 테스트 명세를 개발하여 계획된 테스트를 구현하고, 테스트 결과 보고서를 변경 관리 기관에 통보해야 하는 책임을 가지고 있다. 주요 역량은 테스팅이 수행되는

도메인 역량과, 다양한 테스트 및 검증 접근법/방법론 역량이다.

<표 2-4> 테스터 책임 및 역량

	역할: 테스터
책임	1) 테스트 활동들이 계획됨을 보증해야 한다. 2) 테스트 명세(목적 및 테스트 케이스)를 개발해야 한다. 3) 지정된 소프트웨어 요구사항에 대한 테스트 목적의 추적성 및 지정된 테스트 목적에 대한 테스트 케이스의 추적성을 보증해야 한다. 4) 계획된 테스트가 구현되고 지정된 테스트가 수행됨을 보증해야 한다. 5) 예상 결과에서 벗어난 것을 식별해야 하고, 그것을 테스트 보고서에 기록해야 한다. 6) 평가 및 결정을 위해 예상 결과에서 벗어난 것에 대해 관련 변경 관리 기관에 통보해야 한다. 7) 산출물을 보고서에 수록해야 한다. 8) 소프트웨어 테스트 장치(equipment)를 선택해야 한다.
주요 역량	1) 소프트웨어 요구사항, 데이터, 코드 등 테스팅이 수행되는 도메인에 역량이 있어야 한다. 2) 다양한 테스트 및 검증 접근법/방법론에 역량이 있어야 하고, 주어진 상황에서 가장 적합한 방법을 식별할 수 있어야 한다. 3) 주어진 명세로부터 테스트 케이스를 도출할 수 있어야 한다. 4) 분석적 사고 능력과 좋은 관찰 기술을 가져야 한다. 5) IEC 62279의 관련 조항/부조항들을 이해해야 한다.

(5) 검증자(Verifier) 역할 명세

검증자는 소프트웨어 검증 계획을 개발하고, 지정된 검증 목표와 검토, 통합 및 테스트에서 문서화된 증거의 적합성을 확인해야 하는 책임이 있다. 주요 역량은 관련 도메인, 표준과 다양한 검증 접근법/방법론이다.

<표 2-5> 검증자 책임 및 역량

	역할: 검증자
책임	1) 무엇이 검증을 필요로 하는지, 어떤 타입의 프로세스 (예, 검토, 분석 등) 및 테스트가 증거로서 요구되는지를 언급하는 (품질 이슈를 포함할 수 있는) 소프트웨어 검증 계획을 개발해야 한다. 2) 지정된 검증 목표와 검토, 통합 및 테스트에서 문서화된 증거의 적합성(완전성, 일관성, 정확성, 관련성 및 추적성)을 확인해야 한다. 3) 예외를 식별하고, 예외를 위험 (영향) 용어로 평가하며, 기록하고, 평가 및 결정을 위해 이것들을 관련 변경 관리 기관에 통보해야 한다. 4) 검증 프로세스(검토, 통합 및 테스팅)를 관리하고 필요에 따라 활동의 독

	립성을 보장해야 한다. 5) 검증 활동에 대한 기록을 개발하고 유지해야 한다. 6) 검증 활동의 결과를 알리는 확인 보고서를 작성해야 한다.
주요 역량	1) 소프트웨어 요구사항, 데이터, 코드 등 검증이 수행되는 도메인에 역량이 있어야 한다., 2) 다양한 검증 접근법/방법론에 역량이 있어야 하고, 주어진 상황에서 가장 적합한 방법 또는 방법들의 조합을 식별할 수 있어야 한다. 3) 주어진 명세로부터 검증의 타입을 도출할 수 있어야 한다. 4) 분석적 사고 능력과 좋은 관찰 기술을 가져야 한다. 5) IEC 62279의 관련 조항/부조항들을 이해해야 한다.

(6) 통합자(Integrator) 역할 명세

통합자는 필요한 입력 컴포넌트, 통합 활동의 순서 및 결과적으로 통합된 컴포넌트를 포함하는 설계자의 컴포넌트 명세 및 소프트웨어 컴포넌트에 대한 소프트웨어 및 소프트웨어/하드웨어 통합 테스트 명세를 개발하고 관리하는 책임을 가지고 있다. 주요 역량은 관련 도메인, 표준과 다양한 통합 접근법/방법론 역량이다.

<표 2-6> 통합자 책임 및 역량

	역할: 통합자
책임	1) 소프트웨어 베이스라인을 사용하여 통합 프로세스를 관리해야 한다 2) 필요한 입력 컴포넌트가 무엇인지, 통합 활동의 순서 및 결과적으로 통합된 컴포넌트가 무엇인지를 말해주는, 설계자의 컴포넌트 명세 및 아키텍처에 기반하여, 소프트웨어 컴포넌트에 대한 소프트웨어 및 소프트웨어/하드웨어 통합 테스트 명세를 개발하여야 한다. 3) 통합 활동에 대한 기록을 개발하고 유지해야 한다. 4) 통합 예외를 식별하고, 기록하며, 평가 및 결정을 위해 이것들을 관련 변경 관리 기관에 통보해야 한다. 5) 통합 결과물을 말해주는 컴포넌트 및 전체 시스템 통합 보고서를 작성해야 한다.
주요 역량	1) 관련 프로그래밍 언어, 소프트웨어 인터페이스, 운영 체제, 데이터, 플랫폼, 코드 등 컴포넌트 통합이 수행되는 도메인에 역량이 있어야 한다. 2) 다양한 통합 접근법/방법론에 역량이 있어야 하고, 주어진 상황에서 가장 적합한 방법 또는 방법들의 조합을 식별할 수 있어야 한다. 3) 다양한 중간 수준에서 요구되는 설계 및 기능을 이해하는 능력이 있어야 한다. 4) 통합된 기능들의 집합으로부터 통합 테스트의 타입을 도출할 수 있어야 한다. 5) 분석적 사고 능력과 시스템 수준 관점에 도움이 되는 좋은 관찰 기술이 있어야 한다. 6) IEC 62279의 관련 조항/부조항들을 이해해야 한다.

(7) 확인자(Validator) 역할 명세

확인자는 요구사항 충족을 보증하기 위한 확인 계획을 수립하고, 소프트웨어 확인을 위한 필수 작업과 활동을 지정하며 이 계획을 평가자와 동의해야 하는 책임이 있다. 확인 계획에는 소프트웨어 요구사항과 추적성, 표준에 부합하는 소프트웨어, 테스팅 확인 등이 포함되어 있어야 한다. 주요 역량으로는 관련 도메인, 안전 특성 경험, 표준과 다양한 확인 접근법/방법론이다.

<표 2-7> 확인자 책임 및 역량

	역할: 확인자
책임	1) 응용 프로그램의 의도된 환경 내에서 소프트웨어에 대한 시스템 이해를 개발해야 한다. 2) 확인 계획을 수립하고, 소프트웨어 확인을 위한 필수 작업과 활동을 지정하며 이 계획을 평가자와 동의해야 한다. 3) 의도된 환경/사용에 대한 소프트웨어 요구사항을 검토해야 한다. 4) 모든 소프트웨어 요구사항이 충족되었는지 보증하기 위해서 소프트웨어 요구사항에 대해 소프트웨어를 검토해야 한다. 5) 소프트웨어 프로세스와 개발된 소프트웨어가 할당된 SIL을 포함하여 표준의 요구사항에 부합하는지 평가해야 한다. 6) 검증 및 테스팅의 정확성, 일관성, 충분성을 검토해야 한다. 7) 테스트 케이스와 실행된 테스트의 정확성, 일관성, 충분성을 확인해야 한다. 8) 모든 확인 계획 활동들이 수행되었는지 보증해야 한다. 9) 모든 편차를 위험 (영향) 관점에서 검토 및 분류, 기록해야 하며, 변경 관리 및 의사 결정에 대한 책임이 있는 기관에 제출하여야 한다. 10) 의도된 사용에 대해 소프트웨어의 적합성에 대한 추천을 제공할 수 있어야 하며, 적절할 시 응용 제약사항을 표시하여야 한다. 11) 확인 계획과의 편차를 수록해야 한다. 12) 개발의 다양한 단계에서 적절히 (일반 개발 프로세스의 구체화로서) 전체 프로젝트에 대한 감사(audit), 검사(inspection) 또는 검토(review)를 수행해야 한다. 13) 적절할 시 이전 응용 프로그램에 관한 확인 보고서를 검토하고 분석해야 한다. 14) 개발된 솔루션들이 소프트웨어 요구사항에 추적가능한지 검토해야 한다. 15) 관련된 위험요소 로그와 잔존하는 부적합 사항들이 검토되었는지, 모든 위험요소들이 제거 또는 위험 관리/전송 수단을 통해 적절한 방식으로 마감되었는지 보증해야 한다. 16) 확인 보고서를 작성해야 한다. 17) 소프트웨어 릴리즈(release)에 대한 동의/비동의 의견을 제공해야 한다.
주요 역량	1) 확인이 수행되는 도메인에 역량이 있어야 한다. 2) 응용 프로그램 도메인의 안전 특성에(safety attributes) 대한 경험이 있어야 한다. 3) 다양한 확인 접근법/방법론에 역량이 있어야 하고, 주어진 상황에서 가장 적합한 방법 또는 방법들의 조합을 식별할 수 있어야 한다. 4) 의도된 활용을 명심하면서, 주어진 명세로부터 요구되는 확인 증거의 타입을 도출할 수 있어야 한다. 5) 증거의 다른 소스 및 타입들을 결합할 수 있어야 하며, 응용 프로그램의 목

적 또는 제약사항 및 한계점에 대해 적합한지에 대한 전체적인 관점을 종합할 수 있어야 한다.
6) 분석적 사고 능력과 좋은 관찰 기술을 가져야 한다.
7) 응용 프로그램 환경에 대한 이해를 포함하여 전반적 소프트웨어 이해 및 관점을 가져야 한다.
8) IEC 62279의 요구사항을 이해해야 한다.

(8) 평가자(Assessor) 역할 명세

평가자는 소프트웨어 프로세스와 개발된 소프트웨어가 할당된 SIL을 포함하여 표준의 요구사항에 부합하는지와 프로젝트 참여자와 조직의 능력을 평가해야 하는 책임을 가지고 있다. 평가 보고서에서 소프트웨어 요구사항과의 편차를 위험(영향) 관점에서 식별하고 평가한 사항이 포함되어야 한다. 주요 역량은 관련 도메인, 표준과 안전 원리 적용 경험이 필요하며, 공인된 안전 기관으로부터의 승인/라이선스가 요구된다.

<표 2-8> 평가자 책임 및 역량

역할: 평가자	
책임	1) 응용 프로그램의 의도된 환경 내에서 소프트웨어에 대한 시스템 이해를 한다. 2) 평가 계획을 수립하고 이것을 안전 기관 및 고객 조직(평가자의 계약 기관)에 통보해야 한다. 3) 소프트웨어 프로세스와 개발된 소프트웨어가 할당된 SIL을 포함하여 표준의 요구사항에 부합하는지 평가해야 한다. 4) 소프트웨어 개발을 위한 프로젝트 참여자와 조직의 능력을 평가해야 한다. 5) 확인 및 검증 활동과 지원 근거를 평가해야 한다. 6) 소프트웨어 개발을 위해 채택된 품질 관리 시스템을 평가해야 한다. 7) 형상 및 변경 관리 시스템과 그 사용 및 적용 근거를 평가해야 한다. 8) 평가 보고서에서 소프트웨어 요구사항과의 편차를 위험(영향) 관점에서 식별하고 평가해야 한다. 9) 평가 계획이 구현되었음을 보증해야 한다. 10) 개발의 다양한 단계에서 적절히 전체 개발 프로세스에 대한 안전 감사(audit) 및 검사(inspection)를 수행해야 한다. 11) 제약 조건, 적용 조건 및 위험 관리에 대한 관찰을 자세히 언급하며, 개발된 소프트웨어가 의도된 사용에 대해 적합한지에 대한 전문적인 관점을 제공해야 한다. 12) 평가 보고서를 작성하고 평가 프로세스에서의 기록들을 유지해야 한다.
주요 역량	1) 평가가 수행되는 도메인/기술에 역량이 있어야 한다. 2) 공인된 안전 기관으로부터의 승인/라이선스를 가져야 한다. 3) 안전 원리 및 적용 도메인 내 원리의 적용에 있어서 충분한 수준의 경험을 지속적으로 얻기 위해 노력해야 한다. 4) 주어진 상황에 적합한 방법 또는 방법의 조합이 적용되었는지 확인하는 능력이 있어야 한다. 5) IEC 62279의 요구사항을 만족시킴에 있어서 관계가 있는 안전, 인적 자원, 기술 및 품질 관리 프로세스를 이해할 수 있어야 한다. 6) 평가 접근법/방법론에 능력이 있어야 한다.

7) 분석적 사고 능력과 좋은 관찰 기술을 가져야 한다.
8) 증거의 다른 소스 및 타입들을 결합할 수 있어야 하며, 응용 프로그램의 목적 또는 제약사항 및 한계점에 대해 적합한지에 대한 전체적인 관점을 종합할 수 있어야 한다.
9) 응용 프로그램 환경에 대한 이해를 포함하여 소프트웨어 이해 및 관점을 가져야 한다.
10) (품질 관리, 형상 관리, 확인 및 검증 프로세스와 같은) 모든 개발 프로세스의 충분성을 판단할 수 있어야 한다.
11) IEC 62279의 요구사항을 이해해야 한다.

(9) 프로젝트 관리자(Project Manager) 역할 명세

프로젝트 자원 관리와 소프트웨어의 제공 및 배포에 대해 책임이 있으며, 이해관계자들로부터의 안전 요구사항들도 충족되고 전달됨을 보장해야 한다. 주요 역량은 표준 및 안전과 개발 프로세스 요구사항을 이해해야 한다.

<표 2-9> 프로젝트 관리자 책임 및 역량

역할: 프로젝트 관리자	
책임	1) 프로젝트에 대해 품질 관리 시스템과 5.1에 따른 역할의 독립성이 자리 잡고 있는지 보증해야 하며 프로젝트 진행이 계획 대비 점검되고 있는지 보증해야 한다. 2) 역할들의 필수적인 독립성을 숙지하면서, 안전 작업을 포함해 필수적인 업무를 수행하기 위해 프로젝트에 충분한 수의 능력 있는 자원을 할당해야 한다. 3) IEC 62279에 정의된 것과 같이 프로젝트에 적합한 확인자가 선임되었음을 보증해야 한다. 4) 소프트웨어의 제공 및 배포에 대해 책임이 있으며, 이해관계자들로부터의 안전 요구사항들도 충족되고 전달됨을 보장해야 한다. 5) 안전 작업의 적절한 구현 및 이행을 위해서 충분한 시간을 허용해야 한다. 6) 개발 과정에서 부분적 및 완전한 안전 결과물을 확인해야 한다. 7) 안전 관련 의사 결정에서 충분한 기록 및 추적성이 유지되는지 보증해야 한다.
주요 역량	1) IEC 62279의 품질, 역량, 조직 및 관리 요구사항을 이해해야 한다. 2) 안전 프로세스의 요구사항을 이해해야 한다. 3) 각기 다른 선택사항들을 비교 검토할 수 있어야 하며, 어떤 결정이나 선택된 사항이 안전 성능에 미치는 영향을 이해해야 한다. 4) 소프트웨어 개발 프로세스의 요구사항을 이해해야 한다. 5) 다른 관련 표준들의 요구사항을 이해해야 한다.

(10) 형상 관리자(Configuration Manager) 역할 명세

형상 관리자는 형상 관리 시스템 내에서 모든 소프트웨어 컴포넌트가 명확히 식별되고 독립적으로 버전 관리되도록 하는 책임이 있으며, 주요 역량은 형상 관리 능력과 표준 이해이다.

<표 2-10> 형상 관리자 책임 및 역량

역할: 형상 관리자	
책임	1) 소프트웨어 형상 관리 계획에 대한 책임을 져야 한다. 2) 형상 관리 시스템을 소유해야 한다. 3) 모든 소프트웨어 컴포넌트가 형상 관리 시스템 내에서 명확히 식별되고 독립적으로 버전 관리 되도록 제정해야 한다. 4) 소프트웨어 컴포넌트의 호환되지 않는 버전을 포함한 릴리즈 노트를 작성해야 한다.
주요 역량	1) 소프트웨어 형상 관리 능력이 있어야 한다. 2) IEC 62279의 요구사항을 이해해야 한다.

(11) 품질 보증 관리자(Quality Assurance Manager) 역할 명세

품질 보증 관리자는 프로젝트 관리자와 협력하여 품질 목표를 정의하고, 소프트웨어 품질 보증 계획에 따라, 품질 보증 프로세스를 관리하는 책임이 있다. 주요 역량은 소프트웨어 품질 관리와 보증이다. 이를 위해 소프트웨어 공학, 소프트웨어 안전 표준, 소프트웨어 프로세스 개선 방법, 형상관리 등의 지식이 있어야 한다.

<표 2-11> 품질 보증 관리자 책임 및 역량

역할: 품질 보증 관리자	
책임	1) 소프트웨어 품질 보증 계획에 대한 책임을 져야 한다. 2) 주어진 프로젝트에 품질 관리 시스템을 적용해야 한다. 3) 확인 및 검증 활동들이 효과적으로 수행될 수 있도록 감사 추적이 설정될 수 있도록 보증해야 한다. 4) 프로젝트 관리자와 협력하여 품질 목표를 정의해야 한다. 5) 소프트웨어 품질 보증 계획에 정의된 것과 같이 소프트웨어 품질 보증 프로세스를 관리해야 한다.

	6) 소프트웨어 품질 보증 계획에 계획된 대로 품질 보증 활동의 결과를 알리는 소프트웨어 품질 보증 보고서를 작성해야 한다. 7) NOTE 소프트웨어 품질 보증 보고서는 IEC 62425에 정의된 품질 관리 보고서에 포함된다.
주요 역량	1) 소프트웨어 품질 관리 능력이 있어야 한다. 2) 주어진 프로젝트에서 품질 관리를 이해해야 한다 3) 주어진 프로젝트에서 품질 관리를 조정(tailoring)할 수 있어야 한다. 4) ISO 9001, ISO/IEC 90003, ISO/IEC 25010 및 IEC 62279에서 요구되는 것과 같은 소프트웨어 품질 보증 활동에 역량이 있어야 한다. 5) 회사 품질 시스템에 역량이 있어야 한다. 6) 소프트웨어 프로세스 개선 방법에 역량이 있어야 한다. 7) 소프트웨어 형상 관리, 문서 관리 및 추적성, 데이터 기록 및 분석에 대한 전반적 이해를 가져야 한다. 8) 감사의 성능을 관할해야 한다. 9) 소프트웨어 공학에 대한 전반적 이해가 있어야 한다. 10) 프로젝트 관리에 대한 전반적 이해가 있어야 한다. 11) 분석적 사고 능력과 좋은 관찰 기술을 가져야 한다. 12) 좋은 의사소통 능력을 가져야 한다. 13) IEC 62279의 요구사항을 이해해야 한다.

(12) 검토자(Reviewer) 역할 명세

검토자는 검증자의 감독 아래 출력 문서 검토의 책임이 있다. 주요역량은 관련 도메인 역량이다.

<표 2-12> 검토자 책임 및 역량

	역할: 검토자
책임	1) 검증자의 감독 아래 정의된 기준에 따라 출력 문서를 검토해야 한다. 2) 7.2.4.22와 연관된 것들 같은 검토 결과의 기록을 제공해야 한다. 3) 문서 작성자와 같은 사람이면 안 된다.
주요 역량	1) 소프트웨어 요구사항, 설계, 테스팅 등 리뷰가 수행되는 도메인에 역량이 있어야 한다. 2) 검토를 위해 선정된 목표를 달성할 능력이 있어야 한다.

3) 철도 분야 소프트웨어 안전 관련 역량

철도 분야 IEC 62279에서 각 직무별로 소프트웨어 관련 일반적인 역량을 제외하고, 안전에 특화된 역량을 정리하면 다음과 같다. IEC 62279 관련 지식은 공통이므로 아래 표에서는 제외하였다.

<표 2-13> 소프트웨어안전 직무별 역량

직 무	책임	역 량
요구사항 관리자	요구사항 명세, 추적	도메인의 안전 특성에 대한 경험 적용 가능한 규제 이해
설계자	설계 문서 개발, 유지	안전 설계 원리 문제 도메인 이해 하드웨어 플랫폼, 운영 체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항들을 이해
개발자	개발, 개발 문서 작성	안전 설계 원리 적용 가능한 공학 역량 구현 언어와 지원 도구 역량 하드웨어 플랫폼, 운영 체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항들을 이해
테스터	테스트 명세, 테스트	소프트웨어 요구사항, 데이터, 코드 등 테스팅이 수행되는 도메인 역량
검증자	검증, 예외 식별	소프트웨어 요구사항, 데이터, 코드 등 검증이 수행되는 도메인 역량
통합자	SW, HW 통합	프로그래밍 언어, 소프트웨어 인터페이스, 운영 체제, 데이터, 플랫폼, 코드 등 컴포넌트 통합이 수행되는 도메인 역량
확인자	요구사항 충족 보증	확인이 수행되는 도메인 역량 도메인의 안전 특성에 대한 경험
평가자	SIL 기반 평가	평가가 수행되는 도메인/기술에 역량 공인된 안전 기관으로부터의 승인/라이선스 안전 원리 지식과 적용 경험
프로젝트 관리자	자원, 결과물 관리	안전 프로세스 요구사항 이해 선택 사항이 안전 성능에 미치는 영향 이해
형상 관리자	형상, 버전 관리	안전 표준의 요구사항 이해
품질보증 관리자	품질 보증	안전 표준의 요구사항 이해
검토자	문서 검토	검토가 수행되는 도메인 역량

2. 국가 차원 표준 사례

안전 필수 시스템을 개발에 참여하는 참여자의 역할과 기술에 대해 설명하기 위해 두 번째 사례로 국가 차원에서 구체화되어 정의되고 적용되고 있는 소프트웨어 안전 관련 대표적인 역량모델 중 하나인 미연방에너지국 안전 소프트웨어 품질 보증 기능 영역 자격 기준을 살펴보도록 한다.

1) 미연방에너지국 안전 소프트웨어 품질 보증 기능 영역 자격 기준 개요

미연방에너지국 (U.S. Department of Energy, 이하 DOE)은 DOE-STD-1172-2011[11] "안전 소프트웨어 품질 보증 기능 영역 자격 기준(Safety Software Quality Assurance Functional Area Qualification Standard)" 표준 문서에서 DOE 국방 원자력 시설 기술 요원의 자격 기준을 명시하고 있다. 안전 소프트웨어 품질 보증(Safety Software Quality Assurance, SSQA) 기능 영역 자격 기준(Functional Area Qualification Standard, FAQS)은 안전 시스템 소프트웨어, 안전 & 위험 분석 소프트웨어 및 설계 소프트웨어, 안전 관리와 행정 제어 소프트웨어를 포함해 안전 소프트웨어의 실행에 대한 도움 또는 방향 지시, 안내, 감독 또는 평가를 제공하는 업무를 담당한 DOE 요원 선발에 적용될 공통 기능 부문 역량 요건들을 정하고 있다.

요원 선발에 적용되는 역량들 중, 요원이 구비해야 하는 것으로 기대하는 수준의 지식 또는 기술을 "필수 수행 활동들(mandatory performance activities)"로 지정하고 있다. 기준 내용에서 "해야 한다(must)"라는 단어는 필수 요건을 암시하는 것이며, "할 것을 권한다(should)"란 단어는 필수적으로 요구된 것은 아니나 할 것을 권하는 실무를 암시하며, 또한 "-할 수 있을지 모른다(may)"란 단어는 선택 여부에 따라 할지 말지 결정할 수 있는 활동을 암시한다.

여기에 포함된 역량들은 지식을 친숙한 수준(familiarity level), 실무 수준(working level) 또는 전문가 수준(expert level)으로 규정하고 있거나 개인이 업무와 관련된 한 과제나 활동을 수행할 수 있는 능력이 있다는 것을 입증할 것을 요구한다. 이 수준들은 다음과 같이 정의한다.

친숙한 수준(Familiarity level) : 더 높은 수준의 지식을 가진 개인들과 그 주제나 프로세스에 대해 논의하기에 충분할 정도의 기초 지식이나 노출 경험으로 정의된다.

---

11) Department of Energy, DOE STD-1172-2011, *Safety Software Quality Assurance Functional Area Qualification Standard*, 2011.

실무 수준(Working level) : 기기 가동/활동들을 모니터링하고 평가하고, 수용 가능한 수행 기준을 적용할 수 있고, 적절한 전문가(예, 기술적, 법적, 안전 전문가)의 조언을 찾고 구할 필요성을 인식하거나 또는 DOE의 활동들의 안전을 보장하기 위해 요구되는 적절한 참조 자료들을 살펴보는데 요구되는 지식으로 정의된다.

전문가 수준(Expert level) : 절차 안내가 없는 경우 조언을 제공할 만큼 충분히 그 주제나 프로세스에 대한 포괄적이며 심층적인 지식으로 정의된다.

능력을 입증(Demonstrate the ability) : 정책, 절차, 가이드라인과 수용되는 업계 또는 DOE의 관행들(practices)에 따라 한 과제나 활동을 실제 수행하는 것으로 정의된다.

미연방에너지국 안전 소프트웨어 품질 보증 기능 영역 자격 기준에 명시된 역량들을 소프트웨어 안전 확보를 위한 기술 집합 구성에 참조할 수 있으며, 아래에서 세부 내용을 기술한다.

2) 미연방에너지국 안전 소프트웨어 품질 보증 요원들의 의무와 책임

다음은 안전 소프트웨어 품질 보증(SSQA) 기능 영역에 배정된 요원들에게 기대되는 전형적인 의무와 책임들이다.

- DOE O 414.1C[12]와 고객이 요청한 다른 요구사항들과 일관된 DOE SSQA 프로그램의 개발 및 실행 책임을 진 DOE의 선임 관리자(senior manager)의 업무를 돕는 것.
- DOE 관리에 이용되는 인적 자원이자 SSQA 활동들을 위한 기술 연락책 역할을 하는 것.
- 안전 소프트웨어를 적절한 절차에 따라 확인하고 평가하고 제어한다는 것을 보증하기 위해 안전 및 품질 보증 문서를 검토하는 것.
- DOE O 414.1C. 안전 소프트웨어가 DOE O 414.1C의 내용들과 일관되게 개발되고, 조달되고, 검증되고, 확인되고, 사용되고 유지 관리되어 왔다는 것을 검증하는 것.
- SSQA 프로그램들을 개발, 조달, 검증, 확인, 사용 또는 유지 관리를 담당한 DOE 요원들과 하도급업체 요원들을 위한 훈련과 자격 심사(qualification) 프로그램들을 검토하고 평가하는 것.
- DOE와 하도급 업체의 소프트웨어 품질 보증(SQA) 프로그램들, 계획들과 프로세스

12) DOE, 2005, 품질 보증, Quality Assurance, DOE / NNSA 제품 및 서비스의 품질이 고객의 기대를 충족 하거나 초과하는 것을 보장

들이 위험요소(hazards) 및 위험(risks) 평가에 기초해 개발되었다는 점과 이러한 요소들은 DOE 지시들 및 적용되는 요구사항들을 준수한다는 것을 검증하는 것.

- SSQA 프로그램들과 프로세스들의 충분성, 구현 효과성 및 적용되는 DOE 지시들과 요구사항들의 준수를 검증하며, DOE와 하도급 업체들이 이들 프로그램들과 프로세스들을 실행하는 것을 모니터링하고 평가하는 계획들을 개발하고 실행하는 것.
- SSQA 프로그램들, 계획들과 절차들 및 다른 감독 활동들의 검토, 감시 및 평가를 리드하거나 수행하며, 또한 결과들을 문서로 기록하고 보고서를 준비하고 결과로 나온 실행 조치들을 모니터링 하는 것.
- DOE의 SSQA 프로그램들의 효과적인 실행을 보장하기 위해 DOE 본부와 현장 부서들, 규제 기관들과 이해 관계자들과 의견을 조율하는 것.
- 모든 안전 소프트웨어가 DOE O 414.1C와 적용되는 다른 요구사항들을 준수하고 이에 맞춰 등급이 정해졌다는 것을 검증하는 것.
- 학습한 교육 내용을 정리하고 이를 필요로 할 업체들과 개인들에게 배포하고 또한 수정 조치가 효과적으로 실행된다는 것을 보장하기 위해 안전 소프트웨어와 관련된 보고 가능한 일들이 발생하는 것을 보장하기 위해 하도급 업체를 모니터링 하는 것

3) 미연방에너지국 안전 소프트웨어 품질 보증 요원들의 필수 기술 역량

(1) 소프트웨어 관리와 품질 보증

가) 안전 소프트웨어 품질 보증 요원들은 DOE의 품질 보증 정책, 프로그램들과 다음 문서들에 수록된 프로세스들에 대해 실무 수준의 지식을 보유하고 있다는 것을 입증해야 한다. :

- DOE O 414.1C, 품질 보증

- 10 CFR 830[13], 서브 파트 A, “품질 보증”

뒷받침하는 지식 및/또는 기술들(Skills)

13) DOE, 2011, NUCLEAR SAFETY MANAGEMENT, 원자력 안전 관리
CFR, Code of Federal Regulations, 미국연방규정집

a. DOE O 414.1C와 10 CFR 830 서브 파트 A의 개발 목적과 상호 관계에 대해 논의한다.

b. 한 품질 보증 프로그램(QAP)의 개발, 검토, 승인 및 실행을 위해 필요한 DOE와 하도급 업체의 요구사항과 이들의 책임들에 대해 논의한다.

c. DOE 0 414.1C에 명시된 안전 소프트웨어 목록 요구사항들이 어떻게 부합될 수 있는지에 대해 논의하고 한 예를 제시한다.

d. DOE의 품질 보증 요구사항들을 실행하는 과정에서 단계적 접근법을 적용한 목적, 이 접근법의 이점들과 제약들에 대해 논의한다.

e. 10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4, 안전 소프트웨어 가이드, DOE O 414.1C, 품질 보증에 기술된 것 같은 한 소프트웨어 품질 보증 프로그램의 실행 방법에 대해 논의한다.

f. 미국 기준 또는 국제적으로 합의된 기준들이 품질 보증 요구사항들에 준하는 수준을 제공할 수 있는 실무 프로세스를 개발하고 실행하는데 어떻게 사용될 수 있는지에 대해 논의하고 한 예를 제시한다.

나) 안전 소프트웨어 품질 보증 요원들은 성공적인 소프트웨어 품질 보증 프로그램의 구성 요소들에 대한 실무 수준의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 안전 소프트웨어 품질과 관련된 경우, 다음과 같은 타입의 소프트웨어 관리 계획(들)의 개발 목적, 범위 및 수록 내용에 대해 논의한다:

- 소프트웨어 프로젝트 관리 계획/소프트웨어 위험 관리 계획
- 소프트웨어 개발 계획
- 소프트웨어 안전 계획
- 소프트웨어 품질 보증 계획
- 소프트웨어 테스트, 확인 및 검증 계획
- 소프트웨어 형상 관리 계획
- 소프트웨어 획득 및 납품업체 관리 계획
- 소프트웨어 문제 보고 및 수정 조치 계획
- 소프트웨어 통합 계획
- 소프트웨어 유지 관리 계획

- 소프트웨어 설치, 운영, 훈련, 퇴거(retirement) 계획

b. 납품업체 평가와 공급원 시찰 프로세스들을 포함해 안전 소프트웨어 획득 방법을 규정하고 기술한다.

c. 아래 참조 문서들을 이용해 안전 소프트웨어 개발, 이용, 등급 지정 및 유지 관리를 위해 수용 가능한 안전 소프트웨어 품질 보증 프로그램의 다양한 구성 요소들에 대해 기술한다.

- 10 CFR 830, “원자력 안전 관리”
- DOE O 414.1C, 품질 보증
- 10 CFR 830 하부 파트 A, 품질 보증 요구사항들과 함께 이용하기 위한 DOE G 414. 1-4, 안전 소프트웨어 가이드, DOE O 414.1C.
- DOE O 200.1A, 정보 기술 관리
- 파트 I, 파트 II의 하부 파트 2.7과 파트 IV의 하부 파트 4.1을 포함한 ASME NQA-1-2000, 원자력 시설 적용을 위한 품질 보증 요구사항들
- IEEE 730.1, IEEE의 소프트웨어 품질 보증 계획 기준

(2) 안전 소프트웨어 및 시스템 관계

가) 안전 소프트웨어 품질 보증 요원들은 안전 소프트웨어 유형과 등급 책정 수준들에 대한 실무 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 안전 시스템 소프트웨어, 안전 및 위험 분석 소프트웨어 및 설계 소프트웨어, 안전 관리 및 행정 제어 소프트웨어를 포함해 안전 소프트웨어의 일반 특징들, 적용 및 안전 중요성에 대해 설명한다.

b. 제시된 안전 소프트웨어 예들을 감안해, 등급 책정 수준들, (10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4, 안전 소프트웨어 및 DOE O. 414 C, 품질 안전 보증 같이) 안전 소프트웨어와 관련해 적용 가능한 SQA 작업 활동들을 규정한다.

c. 안전 소프트웨어를 규정하기 위한 프로세스를 기술한다.

d. (DOE G 414.1-4에 정의된 것 같이) 다음에 나온 안전 소프트웨어 유형의 기능에 대해 설명하고 각 소프트웨어 유형에 대한 예시를 제공 한다:

- 고객 맞춤 개발된 소프트웨어 유형(Custom developed)
- 구성 가능한(configurable) 소프트웨어 유형
- 취득한(acquired) 소프트웨어 유형
- 유틸리티 계산(utility calculations) 소프트웨어 유형
- 상용 설계 및 분석(commercial design and analysis)

e. 실시간 소프트웨어와 비-실시간 소프트웨어를 구분한다.

나) 안전 소프트웨어 품질 보증 요원들은 안전 시스템 소프트웨어와 시스템-수준 설계 사이의 기능 인터페이스들에 대한 친숙한 수준(familiarity-level)의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 안전 시스템-수준 요구사항들이 정하지는 방법과 이 요구사항들이 하드웨어, 소프트웨어 및 인적 구성 요소들에 배정되는 방법을 식별한다.

b. IEEE 830, IEEE 소프트웨어 요구사항 명세를 위한 권고 실무 (IEEE Recommended Practice for Software Requirements Specifications) 및 IEEE 7-4.3.2, 원자력 발전소의 안전 시스템에 있는 디지털 컴퓨터들의 표준 기준 (IEEE 7-4.3.2, Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations)에 기술된 것 같이 안전 시스템 소프트웨어 컴포넌트와 시스템 수준 설계 사이의 기능 인터페이스들을 정의하는 전형적인 요구사항들을 식별한다.

c. 컴포넌트들과 시스템-수준 설계 간 기능 인터페이스들이 정해진 방법과 안전 시스템 소프트웨어가 서브시스템, 컴포넌트, 인터페이스의 안전 기능들을 제어하는 방법을 한 예를 들어 설명한다.

다) 안전 소프트웨어 품질 보증 요원들은 소프트웨어로 모델된 안전 및 엔지니어링 시나리오들에 대한 친숙한 수준(familiarity-level)의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 수학적 모델과 관련 전산 모델 같이 안전성 분석이나 설계 코드의 개발 프로

세스에서 전형적으로 따르고 있는 단계 순서에 대해 한 예를 들어 설명한다.

b. 안전 소프트웨어를 자극하기 위해 적합한 안전 분석 시나리오의 한 예시를 내놓고 이에 대해 논한다.

c. 안전 관련 설계에 대한 결정들을 돕기 위해 설계 소프트웨어를 적절하게 사용하는 한 예시를 내놓고 이에 대해 논한다.

라) 전 소프트웨어 품질 보증 요원들은 원자력 위험요소들과 안전 관리 및 행정 제어 소프트웨어가 다루는 제어 및 보호 기능 간 관계에 대해 친숙한 수준(familiarity-level) 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 안전 관리의 소프트웨어 기능 요구사항들과 행정 제어 소프트웨어들이 정의되는 방식, 문서화 되는 방식 및 원자력 위험요소 제어와 보호, 위험 관리 및 설계 제약들과 관련해 제어되는 방식을 식별한다.

b. 안전 관리 및 행정 제어 소프트웨어의 범위를 감안해, 방사선 위험 요소 제어와 소프트웨어에 의해 다루어지는 보호가 소프트웨어 기능 요구사항으로 변환되는 방법 및 이 소프트웨어가 원자력 위험 요소 위험 관리 기능들을 지원하는 방식에 대해 설명한다.

마) 안전 소프트웨어 품질 보증 요원들은 DOE의 안전 소프트웨어 중앙 레지스트리(Central Registry)의 개발 목적, 특징들과 내용들에 대해 친숙한 수준(familiarity-level) 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 안전 소프트웨어 중앙 레지스트리의 개발 목적에 대해 논의하고 그 중앙 레지스트리에 들어 있는 현재 코드들(current codes)을 식별한다.

b. 다음 문서들이 안전 소프트웨어 중앙 레지스트리에 있는 코드와 관련 있을 경우, 이 문서들의 의도와 사용에 대해 논의한다. :

- 코드 격차 분석 보고서(code gap analysis report)
- 코드 안내 보고서(code guidance report)

c. 참고 문서들로 10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4, 안전 소프트웨어 가이드, DOE O 414.1C, 품질 보증을 이용해 중앙 레지스트리 코드들에 적용되는 SQA 요구사항들과 DOE 사용자들이 중앙 레지스트리 코드들을 위해 수행할 필요가 있는 SQA 활동들에 대해 기술한다.

(3) 소프트웨어 엔지니어링, 개발 및 유지 관리

가) 안전 소프트웨어 품질 보증 요원들은 소프트웨어의 생명 주기 프로세스들에 대한 실무-수준의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 폭포형(waterfall) 모델, 나선형(spiral) 모델, 증분형(incremental) 모델 및 진화형(evolutionary) 모델 같이 다양한 생명 주기 모델들의 차이들에 대해 논의하고, 이러한 모델들이 소프트웨어 개발 프로세스와 관련된 방식에 대해 논의한다.

b. 10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4, 안전 소프트웨어 가이드, DOE O 414.1C, 품질 보증 문서 및 IEEE 1074, IEEE 소프트웨어 프로젝트 생명 주기 프로세스 개발 기준 문서에 기술된 것 같이, 한 전형적인 소프트웨어 생명 주기 모델의 각 단계, 관련된 SQA 실무 활동들 및 산출물에 대해 기술한다. 각 단계에서 품질 보증, 형상 관리, 확인 및 검증이 하는 역할들에 대해 설명한다.

나) 안전 소프트웨어 품질 보증 요원들은 소프트웨어 요구사항 식별 및 관리에 대한 실무 수준 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술

a. 소프트웨어 요구사항 명세서(SRS)의 개발 및 소프트웨어의 생명 주기가 끝날 때까지 사용되는 방식에 대해 설명한다.

b. SRS의 속성들을 정의하고 그 속성들이 10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4, 안전 소프트웨어 가이드, DOE O 414.1C, 품질 보증 문서 및 IEEE 1074, IEEE 소프트웨어 프로젝트 생명 주기 과정 개발 기준 문서에 기술된 것 같이 안전 소프트웨어와 관련된 경우 그것들에 대해 논의한다.

c. SRS의 개발 목적, 범위 및 내용과 요구사항 추적성 매트릭스(requirements traceability matrix)에 대해 논의하고 SRS의 모든 요소들을 다루고 있다는 것을 보증할 수 있는 방법들에 대해 기술한다.

d. DOE G 414.1-4에 기술된 것 같이 다음에 나온 소프트웨어 요구사항들에 대해 기술하고 이들의 구체적인 예들을 제시한다. :

- 기능 요구사항
- 성능 요구사항
- 접근 제어(access control) 요구사항
- 안전 요구사항들과 인터페이스 요구사항
- 설치 시 고려할 사항 요구사항
- 설계 제약 요구사항

다) 안전 소프트웨어 품질 보증 요원들은 안전 소프트웨어 설계 개념들에 대해 친숙한 수준(familiarity-level)의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음에 나온 소프트웨어 설계 개념들이 IEEE 1016, 소프트웨어 설계 명세를 위한 IEEE의 권고 실무(IEEE Recommended Practice for Software Design Descriptions)에 기술된 것 같이 안전 소프트웨어와 관련 있을 때 이 개념들에 대해 논의한다:

- 모듈러 설계
- 외부 인터페이스, 안전 컴포넌트와 비안전 컴포넌트 간 인터페이스
- 인터페이스 무결성(integrity), 데이터 무결성
- 흐름 제어(flow control)
- 예외와 오류 대처
- 단순성(simplicity)
- 디커플링(decoupling)
- 격리(Isolation)
- 소프트웨어 고장 모드 분석(software failure mode analyses)

라) 안전 소프트웨어 품질 보증 요원들은 안전 소프트웨어 설계와 구현 실무에 대해 친숙한 수준(familiarity-level) 지식을 보유하고 있음을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음 개념들이 안전 소프트웨어 코딩과 관련 있는 경우 이 개념들에 대해 논의한다. :

- 개발 환경
- 타겟 환경 및 재사용 가능한 컴포넌트들
- 데이터 구조
- 논리 구조
- 내장된 의견들(embedded comments)
- 동료 평가(peer reviews)

b. 다음 문서들에 대해 논의하고 각 문서가 안전 소프트웨어 코딩을 뒷받침하는 방식을 기술한다. :

- 설계 명세들
- 프로그램 명세들
- 코딩 기준들
- 시스템 설계 문서
- 프로그래머 매뉴얼
- 사용자 매뉴얼

마) 안전 소프트웨어 품질 보증 요원들은 소프트웨어가 모든 의도된 안전 기능들을 충실히 완수한다는 것을 보증하는 안전 소프트웨어 확인 및 검증 프로세스에 대한 실무 수준의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음 프로세스들과 문서들이 IEEE 1012, 소프트웨어 확인 및 검증을 위한 IEEE 기준에 기술된 것 같이 안전 소프트웨어 확인 및 검증과 관련 있을 때 그것들에 대해 기술한다.

- 요구사항에 대한 확인(validation)
- 설계에 대한 확인 및 검증

- 소스 코드에 대한 확인 및 검증
- 단위/컴포넌트 테스팅
- 통합 테스팅
- 시스템 테스팅
- 확인 및 검증 테스트 케이스들
- 확인 및 검증 보고서들
- 도구에 대한 확인 및 검증
- 독립된 확인 및 검증
- 수용 테스팅(acceptance testing)
- 설치와 체크아웃 테스팅

b. 확인 및 검증 프로그램을 검토하기 위해 사용하는 방법들에 대해 기술한다.

c. 안전성 분석 또는 설계 코드의 한 예시를 들어 다음 것들을 설명한다. :

- 검증이 어떻게 전산 모델이 내재된 수학 모델/솔루션을 표현하는지를 판단하는 프로세스가 되는지에 대해 설명한다
- 확인이 어떻게 한 모델이 그것의 의도된 용도라는 관점으로부터 모형화되는 물리적 현상/특징(예측 능력)을 정확하게 표현하는지를 판단하는 프로세스가 되는지에 대해 설명한다.

d. 응용 안전 소프트웨어에 대한 확인 및 검증 프로세스들의 차이에 대해 설명한다.

e. 스프레드시트와 다른 계산 프로그램들을 이용해 수행된 계산 결과들이 정확하다는 것을 보증하기 위해 사용된 제어 장치들에 대해 기술한다. 계산 과정을 문서화하기 위해 유지 관리되는 기록물들을 확인한다.

f. 테스트 절차들, 테스트 케이스들, 예상되는 결과들, 테스트 데이터 및 실제 결과들의 관계들에 대해 기술한다.

바) 안전 소프트 품질 보증 요원들은 소프트웨어 안전 분석에 대한 친숙한 수준(familiarity-level)의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음에 제시된 각 계획의 수립 목적과 내용을 논의하고 각 계획의 중요성을 10 CFR 830 서브 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한

DOE G 414.1-4, 안전 소프트웨어 가이드, DOE O 414.1C, 품질 보증 문서 및 IEEE 1228, IEEE 소프트웨어 안전 계획 기준에 기술된 것 같이 소프트웨어 안전 분석과 관련 지어 설명한다.

- 소프트웨어 안전 계획
- 소프트웨어 안전 요구사항 분석
- 소프트웨어 안전 설계 분석
- 소프트웨어 안전 코드 분석
- 소프트웨어 안전 테스트 분석
- 소프트웨어 안전 변화 분석

사) 안전 소프트웨어 품질 보증 요원들은 안전 소프트웨어가 적절하게 유지 관리될 것이며 또한 의도된 대로 계속 운용될 것이라는 것을 보증하는 활동들에 대한 친숙한 수준(familiarity-level)의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음 개념들이 IEEE 1219, IEEE 소프트웨어 유지 관리 기준에 기술된 것 같이 안전 소프트웨어 유지 관리 및 운용과 관련 있는 경우, 그 개념들에 대해 논의한다:

- 소프트웨어 유지 관리 가능성(maintainability)
- 유지 관리 계획 수립
- 성능 모니터링
- 예방적 유지 관리

아) 안전 소프트웨어 품질 보증 요원들은 소프트웨어 형상 관리에 대해 실무 수준의 지식을 보유하고 있다는 것을 입증해야 한다.

뒷받침하는 지식 및/또는 기술들

a. 다음 개념들이 IEEE 828, IEEE 소프트웨어 형상 관리 기준에 기술된 것 같이 안전 소프트웨어 형상 관리와 관련된 경우 이 개념들에 대해 논의하고 각 개념이 적용되는 방식에 대해 설명한다:

- 소프트웨어 형상 관리 계획
- 형상 식별 (configuration identification)
- 형상 변화 제어
- 형상 상태 관리(configuration status accounting)
- 형상 감리 및 검토(configuration audits and reviews)
- 하도급업체와 판매업체 제어
- 소스 코드 제어 도구들을 포함해 소프트웨어 형상 관리 도구들

b. 제안된 변화나 잠재적 부족(potential inadequacy)을 포함해 소프트웨어 관련된 미-검토 안전 문제(unreviewed safety question, USQ)에 대한 결정 내용(determination)을 검토하고 평가한다.

c. 소프트웨어 형상 관리와 관련된 문제 보고, 수정 조치 제시 및 실행 과정에 대해 논의한다.

4) 안전 소프트웨어 평가

(1) 안전 소프트웨어 품질 보증 요원들은 적절한 DOE 지시들과 기준들 및 업계 기준들을 이용해 안전 소프트웨어 품질 평가를 수행할 능력을 가지고 있다는 것을 입증해야 한다.

필수 수행 활동들

a. DOE O 414.1C, 품질 보증과 DOE G 414.1-4, 10 CFR 830 하부 파트 A, 품질 보증 요구사항들과 함께 사용하기 위한 DOE G 414.1-4 안전 소프트웨어 가이드 및 DOE O 414.1C, 품질 보증 문서들에 기술된 것 같이 아래에서 선정된 SQA 실무 활동들을 충분히 평가할 수 있는 능력을 입증하는 한 가지 이상의 SQA 평가들에 참여하고 평가 과정과 결과를 문서로 기록해야 한다.

- 소프트웨어 프로젝트 관리 및 품질 유지 계획 수립
- 소프트웨어 위험 관리/ 소프트웨어 형상 관리
- 획득 & 납품업체 관리
- 소프트웨어 요구사항 확인 및 관리
- 소프트웨어 설계 및 실행
- 소프트웨어 안전, 확인 및 검증
- 문제 보고 및 수정 조치

- 안전 소프트웨어의 설계, 개발, 테스팅, 이용 및 평가 담당 요원들 훈련

b. 안전 소프트웨어 품질 보증 요원들은 한 평가를 통해 안전 기준과 관련해 안전 소프트웨어를 평가할 능력이 있다는 것을 입증하고 또한 안전 소프트웨어 기능에 영향을 미칠 수 있는 변화들을 제어하는 방식을 설명해야 한다. 다음에는 이 능력을 입증할 수 있는 활동들의 예들이 나와 있다:

- 안전 기반 문서에서 확인된 위험 요소들이나 사고들의 예방이나 경감(mitigation)과 관련된 안전 소프트웨어의 안전 기능을 식별하고 논의한다. 추가로, 위험 인식 능력을 가진 시스템 엔지니어와 함께 직접 돌아다니며 시설 장비 및/또는 시스템의 안전성을 확인한다.
- 기술 안전 요구사항(TSR) 문서나 안전 소프트웨어의 기능이나 또는 제어를 위한 기술 기준을 포함해 행정 차원의 제어 문서에 정의된 안전 기능을 제어하는 안전 소프트웨어의 기능을 검토한다.
- 안전 소프트웨어 형상과 실행의 변화들과 관련된 안전 시스템 디지털 계측 제어(I&C)에 가해진 영구적 또는 일시적 변화들을 검토하고, 그러한 변화들이 시설의 USQ 절차들에서 어떻게 처리되는지 평가하고, 또한 안전 소프트웨어에 가해진 변화들을 문서로 기록한 방식과 그러한 기록들이 타당한지 검토하고, 또한 안전 기준 문서화 과정이나 시설 절차들에 생긴 변화들이 실행된 방식에 대해 검토한다. 관찰 결과와 그로부터 얻은 통찰들과 더불어 검토 활동 내용을 요약한 문서를 작성한다.
- 한 안전 소프트웨어의 주요 입력 매개 변수들 중 일부를 확인하고 원치 않는 변화들을 막기 위해 그 매개 변수들을 제어하는 방식을 요약한 문서를 작성한다.

(5) 안전 소프트웨어 품질 보증 요원의 기술 역량 분석

안전 소프트웨어 품질 보증(SSQA) 요원의 필수 기술역량은 크게 품질보증, 안전 SW와 시스템 관계, SW 개발 유지관리, 안전 SW 평가로 나눌 수 있다. SW 개발 유지관리 역량은 SWEBOK의 역량과 유사하며, 품질보증 역할은 철도 분야 품질보증 관리자의 역할과 유사하다. 안전 SW 평가 역량은 철도 분야 평가자의 역할과 동일하지는 않으나, 포함되는 부분이 있다. 안전 SW와 시스템 관계의 관계는 앞의 두 개의 사례에서는 자세히 언급되는 않은 부분으로 소프트웨어와 하드웨어 통합의 중요성을 확인할 수 있는 부분이다.

<표 2-14> 안전 소프트웨어 품질 보증(SSQA) 요원 필수 기술 역량

대분류	중분류	역 량
품질 보증	DOE의 품질보증정책	DOE O 414.1C, 품질 보증 10 CFR 830, 서브 파트 A, “품질 보증”
	실무 수준 지식	SW프로젝트, 위험 관리, 안전 계획 SW 개발 계획, 형상 관리 계획, 통합 계획 SW 품질 보증, 테스트, 확인·검증 계획 SW 획득 및 납품업체 관리 계획 SW 문제 보고 및 수정 조치 계획 SW 유지 관리, 설치, 운영, 훈련, 퇴거 계획
안전 SW와 시스템 관계	안전 SW 유형과 등급 책정 수준	안전 SW의 일반 특징들, 적용 및 안전 중요성 안전 SW 관련 적용 가능한 SQA 작업 활동 안전 소프트웨어 유형의 기능
	SW, 시스템 인터페이스	안전 시스템 요구사항, 인적 배정 SW, 시스템 인터페이스 요구사항 정의 및 안전 기능
	안전 계획	안전성 분석이나 설계 코드의 개발 프로세스
	원자력 위험 요소 제어	안전 관리 SW 기능 요구사항, 원자력 위험요소 제어와 보호, 위험 관리 및 설계 제약 식별
SW 개발 유지관리	SW 생명주기	다양한 생명 주기 모델 및 각 모델의 단계 단계별 관련된 SQA 실무 활동, 품질 보증, 형상관리, 확인 및 검증 역할
	요구사항 식별·관리	요구사항 명세서 개발, 안전 SW 관련 사항 식별 요구사항 추적 기능, 성능, 안전 요구사항, 설계 제약 요구사항 등
	설계	모듈러 설계, 외부 인터페이스 안전 컴포넌트, 비안전 컴포넌트 간 인터페이스 인터페이스, 데이터 무결성(integrity) 흐름 제어(flow control), 예외와 오류 대처 소프트웨어 고장 모드 분석 등
	구현	안전 소프트웨어 코딩
	확인·검증	확인·검증 방법, 프로세스 설계, 소스코드, 테스팅 지식
	안전 분석	안전 계획, 요구사항, 설계, 코드, 테스트
	유지관리	유지관리, 성능 모니터링
	형상관리	형상관리 계획, 형상 감리, 형상 관리 도구
안전SW 평가	평가	선정된 SQA 실무 활동 평가 안전 SW 기능에 영향 미칠 수 있는 변화 관리

제4절 소프트웨어 안전 분야 재직자 역할과 필요 기술 현황 분석틀

1. 개요

원자력, 자동차, 의료, 항공, 철도 등 안전 필수 시스템 소프트웨어를 다루는 도메인들에서 소프트웨어 안전을 보장하기 위해 필요한 기술 집합에 공통점이 많기 때문에 본 보고서에서는 범 도메인적인 역할별 기술 집합을 설문 및 심층인터뷰의 분석틀로 사용하고자 한다.

안전 필수 소프트웨어 개발과 관련된 역할은 철도분야 IEC62279 표준 내 Annex B에 명시된 것처럼 요구사항 관리자, 설계자, 구현자, 테스터, 검증자, 통합자, 확인자, 평가자, 프로젝트 관리자, 형상 관리자, 품질보증 관리자, 검토자와 같이 여러 가지로 세분화되어 있다.

IET Competence Criteria for Safety related System Practitioners[14]를 살펴보면 다음과 같이 소프트웨어 안전 관련 역할을 구분하였다.

<표 2-15> 소프트웨어 안전 분야 재직자 역할별 기술 집합(IET 정의)

역할	설명
Corporate functional safety management	조직과 안전 규정에 맞는 안전문화 확보
Project safety assurance management	프로젝트 기간 동안 안전보장 수준 관리 및 시스템 안전성 보장을 위한 증거 수집 관리
Safety-related system maintenance/modification	시스템 사용기간 동안 시스템 운영 및 변경 시 적절한(협의된) 수준의 안전성 관리
Safety-related system or services procurement	안전 요구사항에 따라 공급자가 시스템을 공급하도록 명시하고 계약 관리
Independent safety assessment	시스템 설계, 개발, 운영자와 독립적으로 안전요구사항의 적절성 여부와 시스템이 안전 요구사항을 만족하는지 안전 평가
Safety hazard and risk analysis	예상되는 위해도를 파악하고 사고 위험을 평가. 위험분석은 시스템 설계 및 운영에 큰 영향을 주므로, 시스템 개발 초기부터 운영까지 시행하고

14) IET(2006), Competence Criteria for Safety related System Practitioners, Guidance provided by the IET in collaboration with the HSE and BCS

	관리
Safety requirements specification	안전관련 시스템의 안전 요구사항 작성
Safety validation	안전 관련 시스템이 안전 요구사항을 만족하고 위험 분석 가정이 맞는 지 여부 확인
Safety-related system architectural design	안전 요구사항을 만족하도록 시스템 구조 설계 서브시스템 간 기술한계 및 복잡도 수준을 고려하여 안전 요구사항 확보
Safety-related system hardware realization	하드웨어 기술 적용 후 시스템 안전 테스트를 통과 하도록 안전한 하드웨어 시스템 구현
Safety-related system software realization	소프트웨어 기술 적용 후 시스템 안전 테스트를 통과하도록 안전한 시스템 구현
Human factors safety engineering	시스템의 안전에 영향을 미치는 인적 요소 확인

IEC62279 표준 내 Annex B의 역할과 IET Competence Criteria for Safety related System Practitioners를 참고하고 전문가 자문을 통해 다음과 같이 본 연구에서 소프트웨어 안전 관련 업무의 역할을 정의하였다. 본 연구에서는 안전 시스템 소프트웨어 개발과 관련된 역할을 프로젝트 관리자, 소프트웨어 설계자, 소프트웨어 개발자, 소프트웨어 확인 및 검증자, 형상 관리자, 소프트웨어 품질 보증 관리자, 안전 관리자로 분류하였다. 위험 분석자, 요구사항 관리자, 설계자는 분리되어 있으나, 국내 현실을 반영하여 통합하였으며, 안전 평가자는 도메인 지식을 포함하여 고도의 안전 관련 지식과 경력이 필요하므로 본 연구 범위에서 제외하였다. IET의 안전 필수 소프트웨어 개발과 직접적인 연관이 없는 유지보수나 구매 업무는 고려하지 않았다.

<표 2-16> 소프트웨어 안전 분야 재직자 역할 정의

IET Competence criteria	IEC62279 역할	본 연구 역할 정의
	프로젝트 관리자	프로젝트 관리자
Corporate functional safety management		
Project safety assurance management		안전관리자
Safety-related system maintenance/modification		
Safety-related system or		

services procurement		
Independent safety assessment	평가자	* 중요한 역할이나 본 연구범위에서 제외
Safety hazard and risk analysis		소프트웨어 설계자 * 국내 현황 반영
Safety requirements specification	요구사항 관리자	
Safety-related system architectural design	설계자	
Safety-related system hardware realisation	구현자, 테스터, 통합자	
Safety-related system software realisation	구현자, 테스터, 통합자	소프트웨어 개발자
Safety validation	검증자, 확인자, 검토자	소프트웨어 확인 및 검증자
Human factors safety engineering.		
	형상관리자	형상관리자
	품질보증 관리자	소프트웨어 품질보증 관리자

2. 소프트웨어 안전 분야 재직자 역할별 필요 기술 현황 분석틀

IEC62279 표준 내 각 역할별 주요 기술 집합을 위에서 분류한 프로젝트 관리자, 소프트웨어 설계자, 소프트웨어 개발자, 소프트웨어 확인 및 검증자, 형상 관리자, 소프트웨어 품질 보증 관리자, 안전 관리자에 맞게 간략하게 정리하면 아래와 같으며, 현황 분석틀로 사용하고자 한다.

<표 2-17> 소프트웨어 안전 분야 재직자 역할별 기술 현황 분석틀

역할	기술 집합	필요 역량
프로젝트 관리자	프로젝트 관리 기법	◆ 각종 표준의 품질, 역량, 조직 및 관리 요구사항 이해 ◆ 안전 프로세스의 요구사항 이해 ◆ 각기 다른 선택사항들을 비교, 검토 가능, 선택된 사항이 안전 성능에 미치는 영향 ◆ 소프트웨어 개발 프로세스의 요구사항 이해
소프트웨어 설계자	소프트웨어 요구사항 분석 및 명세 기법	◆ 요구 공학 역량 ◆ 응용 프로그램의 도메인에 대한 경험 ◆ 응용 프로그램 도메인의 안전 특성에 대한 경험

		◆ 시스템의 전반적 역할, 응용 프로그램의 환경 이해 ◆ 분석적 기술과 결과 이해 ◆ 적용 가능한 규제 이해 ◆ 해당 표준의 요구사항 이해
	소프트웨어 설계 기법	◆ 적용 분야에 적합한 공학 역량 ◆ 안전 설계 원리에 대한 역량 ◆ 설계 분석, 테스트 방법론에 대한 역량 ◆ 주어진 환경에서 설계 제약사항 내에서 업무 가능 ◆ 문제 도메인을 이해할 수 있는 역량 ◆ 하드웨어 플랫폼, 운영체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항 이해 ◆ 해당 표준의 관련 조항/부조항 이해
소프트웨어 개발자	소프트웨어 구현/통합 기법	◆ 적용 분야에 적합한 공학 역량 ◆ 구현언어, 지원도구에 대한 역량 ◆ 지정된 코딩 표준과 프로그래밍 스타일 적용 가능 ◆ 하드웨어 플랫폼, 운영체제, 인터페이스 시스템에 의해 야기되는 모든 제약사항 이해 ◆ 컴포넌트 통합이 수행되는 도메인에 대한 역량(관련 프로그래밍 언어, 소프트웨어 인터페이스, 운영체제, 데이터, 플랫폼, 코드 등) ◆ 다양한 통합 접근법/방법론에 대한 역량 ◆ 주어진 상황에서 가장 적합한 방법/방법들의 조합 식별 가능 ◆ 다양한 중간 수준에서 요구되는 설계 및 기능 이해 ◆ 통합된 기능들의 집합으로부터 타입 도출 가능 ◆ 분석적 사고능력 ◆ 시스템 수준 관점에 도움이 되는 좋은 관찰 기술 ◆ 해당 표준의 관련 조항/부조항 이해
소프트웨어 확인 및 검증자	소프트웨어 테스트/검증 기법	◆ 테스팅/검증이 수행되는 도메인에 대한 역량 ◆ 다양한 테스팅/검증 접근법/방법론에 대한 역량 ◆ 주어진 상황에서 가장 적합한 방법 식별 가능 ◆ 주어진 명세로부터 테스트 케이스 및 검증의 타입 도출 ◆ 분석적 사고능력과 좋은 관찰 기술 ◆ 해동 표준의 관련 조항/부조항 이해
	소프트웨어 확인/평가 기법	◆ 확인/평가가 수행되는 도메인에 대한 역량 ◆ 응용 프로그램 도메인의 안전 특성에 대한 경험 ◆ 다양한 확인/평가 접근법/방법론에 대한 역량 ◆ 주어진 상황에서 가장 적합한 방법/방법들의 조합 식별 ◆ 의도된 활용을 명심하면서, 주어진 명세로부터 요구되는 확인 증거의 유형 도출 가능

		◆ 증거의 다른 소스 및 유형 결합 가능 ◆ 응용 프로그램의 목적, 제약사항 및 한계점에 대해 적합한지 전체적인 관점 종합 ◆ 분석적 사고능력과 좋은 관찰 기술 ◆ 전반적인 소프트웨어 이해 및 관점 ◆ 공인된 안전 기관으로부터의 승인/라이선스 ◆ 안전 원리 및 적용 도메인 내 원리의 적용에 있어서 충분한 수준의 경험을 지속적으로 얻기 위한 노력 ◆ 해당 표준의 요구사항을 만족시킴에 있어 관계가 있는 안전, 인적 자원, 기술 및 품질 관리 프로세스 이해 ◆ 모든 개발 프로세스의 충분성 판단 가능 ◆ 해당 표준의 요구사항 이해
형상관리자	소프트웨어 형상관리 기법	◆ 소프트웨어의 형상 관리 능력 ◆ 해당 표준의 요구사항 이해
소프트웨어 품질 보증 관리자	소프트웨어 품질관리 기법	◆ 소프트웨어의 품질 관리 능력 ◆ 주어진 프로젝트에서 품질 관리 이해 ◆ 주어진 프로젝트에서 품질 관리 조정 가능 ◆ 소프트웨어 품질 보증 활동에 대한 역량 ◆ 회사 품질 시스템에 대한 역량 ◆ 소프트웨어 프로세스 개선 방법에 대한 역량 ◆ 소프트웨어 형상관리, 문서관리 및 추적성 데이터 기록 및 분석에 대한 전반적 이해 ◆ 감사의 성능 관할 ◆ 소프트웨어 공학에 대한 전반적 이해 ◆ 프로젝트 관리에 대한 전반적 이해 ◆ 분석적 사고능력과 좋은 관찰 기술 ◆ 좋은 의사소통 능력 ◆ 해당 표준의 관련 조항/부조항 이해
안전관리자	기능안전관리 기법	◆ 각 단계별 모든 도메인에 대한 소프트웨어 안전 역량 ◆ 다양한 안전 기법에 대한 역량 ◆ 주어진 상황에서 가장 적합한 방법 식별 가능 ◆ 분석적 사고 능력 ◆ 시스템 수준 관점에 도움이 되는 좋은 관찰 기술 ◆ 해당 표준의 관련 조항/부조항 이해

위에서 정리한 기술 집합에서 SWEBOK 소프트웨어 지식 영역에서 명시한 소프트웨어 지식 영역 및 소프트웨어 안전 관련 전문가 집단의 의견을 참고하여 다음과 같은 하위 기술들을 생각해 볼 수 있다.

<표 2-18> 소프트웨어 안전기술

기술 집합	기술	지식 영역
공통 지식	◆ 소프트웨어 안전에 대한 정의 ◆ 소프트웨어 기능 안전과 소프트웨어 품질 ◆ 소프트웨어 안전사고 ◆ 소프트웨어 안전 관련 해외 사례	-
소프트웨어 안전 관련 국제 표준/규제	◆ IEC61508 ◆ ISO26262 ◆ DO-178C ◆ EN50128 ◆ IEC60880 ◆ MIL-STD-882	각종 국제 표준/규제
프로젝트 관리 기법	◆ Initiation and Scope Definition ◆ Software Project Planning ◆ Software Project Enactment ◆ Review and Evaluation ◆ Closure ◆ Software Engineering Measurement	SWEBOK Software Engineering Management [본문 2.2.8]
	◆ Safety Management Process ◆ Functional Safety Development Process ◆ Functional Safety Management ◆ Safety Process Tailoring	소프트웨어 안전 관련 산학연 추가 기술 집합
소프트웨어 요구사항 분석 및 명세 기법	◆ Software Requirements Fundamentals ◆ Requirements Process ◆ Requirements Elicitation ◆ Requirements Analysis ◆ Requirements Specification ◆ Practical Consideration	SWEBOK Software Requirements [본문 2.2.2]
	◆ Safety Case ◆ Safety Engineering Method (STAMP/STPA) ◆ System Engineering Process ◆ Software Engineering Process	소프트웨어 안전 관련 산학연 추가 기술 집합

	◆ Basic Safety Analysis Techniques ◆ Safety Goal Management ◆ Software Requirement Analysis ◆ Software Architecture and Design ◆ Change Impact Analysis	
소프트웨어 설계 기법	◆ Software Design Fundamentals ◆ Key Issues in Software Design ◆ Software Structure and Architecture ◆ User Interface Design ◆ Software Design Quality Analysis and Evacuation ◆ Software Design Notations ◆ Software Design Strategies and Methods	SWEBOK Software Design [본문 2.2.3]
	◆ Hazard and SIL(Safety Integrity Level) ◆ Functional Safety Mechanism ◆ Architecture-level FTA ◆ FMEDA, SW-FMEDA	소프트웨어 안전 관련 산학연 추가 기술 집합
소프트웨어 구현/통합 기법	◆ Software Construction Fundamentals ◆ Managing Construction ◆ Practical Considerations ◆ Contruction Technologies ◆ Software Construction Tools	SWEBOK Software Constructions [본문 2.2.4]
	◆ Coding Standard ◆ Code Refactoring ◆ Code-level FTA ◆ Static Analysis	소프트웨어 안전 관련 산학연 추가 기술 집합
소프트웨어 테스트/검증 기법	◆ Software Testing Fundamentals ◆ Test Levels ◆ Test Techniques ◆ Test-Related Measures ◆ Test Process	SWEBOK Software Testing [본문 2.2.5]
	◆ Software Testing in Different Phase ◆ Fault-injection Testing	소프트웨어 안전 관련 산학연 추가

	◆ Host-target Testing ◆ Test Case Management ◆ Static & Dynamic Testing ◆ Logical Proving ◆ Program Slicing	기술 집합
소프트웨어 확인/평가 기법	◆ Formal Verification ◆ HILS & SILS ◆ Traceability Analysis ◆ Reachability Analysis ◆ Predicate Calculus ◆ Modeling and Simulation ◆ Safety Audit	소프트웨어 안전 관련 산학연 추가 기술 집합
소프트웨어 형상관리 기법	◆ Management of the SCM Process ◆ Software Configuration Identifications ◆ Software Configuration Control ◆ Software Configuration Status Accounting ◆ Software Configuration Auditing ◆ Software Release Management and Delivery	SWEBOK Software Configuration Management [본문 2.2.7]
	◆ Configuration Management Planning ◆ Configuration Management System ◆ Defect Management ◆ Orthogonal Defect Analysis	소프트웨어 안전 관련 산학연 추가 기술 집합
소프트웨어 품질관리 기법	◆ Software Quality Fundamentals ◆ Software Quality Management Process ◆ Practical Consideration	SWEBOK Software Quality [본문 2.2.11]
기능안전관리 기법	◆ Hazard Analysis & Risk Assessment (HARA) ◆ HAZOP, FMEA, SW-FMEA ◆ FTA, SW-FTA	소프트웨어 안전 관련 산학연 추가 기술 집합

※ SWEBOK 지식 영역에 해당되는 부분 - 2.2. Software Engineering Body of Knowledge 참고, 그 외의 기술 정의 - A. 기술 용어 정리 참고

제3장 해외 교육 현황조사

제1절 개요

본 장에서는 해외 선진국에서 진행되고 있는 소프트웨어 안전 관련 교육 현황들에 대한 조사 결과를 설명한다. 이를 위하여 현재 미국 및 유럽에서 활발하게 수행되고 있는 소프트웨어 안전 관련 교육 활동들에 대한 조사를 실시하였다. 이러한 선진국의 소프트웨어 안전 관련 교육들은 산업 도메인별로 주도되는 교육, 대학 주도의 교육, 공공기관/인증기관 주도의 교육 차원에서 추진되고 있음을 확인할 수 있었다. 본 연구는 재직자 중심의 교육 개발을 목표하기 때문에 대학 교육 중 석·박사 중심의 정규 과정은 제외하였다. 또한 시스템 안전만을 다루는 과정이나, 소프트웨어 공학, 요구공학, 보안 등 광범위하게 다루는 교육, 도구 중심의 교육은 제외하였다. 교육 내용은 향후 교육커리큘럼 개발의 참고자료로 활용하기 위해 최대한 원문을 그대로 인용하였으며, 마지막 절에서 조사결과를 종합적으로 분석하여 대학에서 소프트웨어 안전교육, 대학 이외의 교육 기관에서 교육, 소프트웨어 안전 교육 내용의 특징을 정리하였다.

제2절 교육과정

1. 민간 전문기관

1) SOFTWARE SAFETY COURSE

교육명	Software Safety Course		
주최단체	HCRQ	기관분류	민간 전문기관
교육시간	4D	교육비용	$2,395
교육내용	소프트웨어 안전 개념, 소프트웨어 안전 표준, 소프트웨어 개발 단계별 안전 점검사항, 소프트웨어 검증 및 위해도 분석을 위한 다양한 기법(FMEA, FTA) 등		

(1) 개요

Software Safety Course는 미국의 HCRQ[15]에서 주최하는 소프트웨어 안전관련 강좌

15) http://www.hcrq.com/

이다. HCRQ는 1986년 설립된 시스템 및 소프트웨어 안전 컨설팅회사로 미군, Siemens, NASA 등의 다양한 회사들을 대상으로 시스템 및 소프트웨어 안전 컨설팅을 맡고 있다.[16] 강의는 최소 수강인원이 정해져 있다(미국 내: 5명, 미국 외: 10명).

(2) 교육내용

<표 3-20> HCRQ Software Safety Course 교육내용

- Software-Caused Accidents
- Software Safety Incentives
- Safety And Reliability Concepts
- Dependability Concepts
- Generic Integrity Levels
- Safety Integrity
- Expected Probability of Failure of Systems
- Risk Concepts
- Basic Approaches to Safe Design
- Software Safety Stds., Guidelines & Regulations
- Formal Methods
- Fault Tolerant Techniques
- Safe Design Techniques
- Safety Assurance Concepts
- Software Requirements Checklist
- Software Design Checklist
- Programming Languages
- System Safety Programs (SSP)
- System Safety Program Plans (SSPP)
- Software Safety Program Plans (SwSPP)
- Software Safety Working Group (SwSWG)
- Hazard Mitigation Precedence
- Hazard Tracking
- Preliminary Hazard Analysis (PHA)
- Functional Hazard Analysis (FHA)
- Determining/Lowering Software Criticality
- Degree Of Rigor In Software Development
- Subsystem Hazard Analysis (SSHA)
- System Hazard Analysis (SHA)
- Software Safety Analysis Process
- Tools
- Software FMEA
- Software FMECA?
- Fault Tree Analysis (FTA)
- Other Analysis Techniques
- Software Safety Cases
- Dealing with COTS Elements
- Safety Verification

강좌는 소프트웨어 안전에 대한 개념 강의로 시작해서, 소프트웨어 안전 표준, 소프트웨어 개발 단계별 안전 점검 사항, 소프트웨어 검증 및 위해도 분석을 위한 다양한 기법들에 대한 내용을 다룬다. 소프트웨어 안전 표준은 방위산업, 항공, 철도 등의 다

16) http://www.hcrq.com/client-list.html

양한 분야의 표준을 대상으로 강의를 진행한다. 소프트웨어 위해도 분석은 FMEA와 FTA와 같은 다양한 기술에 대하여 다룬다. 특히, Software FTA는 HCRQ가 가진 독자적인 기술이 있기 때문에 보다 심도 깊은 내용을 다룰 것으로 보인다.

다음은 소프트웨어 안전 표준/가이드라인 강의 세부 내용이다.

<표 3-1> HCRQ 소프트웨어 안전 표준/가이드라인 강의 세부 내용

국방

- Joint Services Software Safety Engineering Handbook
- MIL-STD-882E(System Safety)
- Relevance to Software Safety
- AMCOM 385-17
- AOP-52
- STANAG 4404

철도

- EN 50128
- IEEE 1483

항공

- NASA Software Safety Standard
- NASA Guidebook
- FAA System Safety Handbook
- SAE ARP4754A/4761
- Relevance to Software Safety
- RTCA DO-178
- Relevance to Software Safety
- ESARR 3, ESARR 4, ESARR 6
- ED-153

일반

- IEEE 1228 (Software Safety Plans)
- IEC 61508
- ISO/IEC 15026
- System & Software Integrity Levels
- UL 1998 (Safety-Related Software)
- MISRA Guidelines

몇몇 표준에 대한 별도의 강의가 존재한다. 위에 언급된 모든 표준을 하나의 코스에서 심도 깊게 배울 수는 없지만, 표준에서 다루는 안전관련 기법이나 기술들의 많은 부분을 Software Safety Course에서 다룬다고 볼 수 있다.

(3) 기타

HCRQ는 Software Safety Course와 별도로 온라인 교육(WEBINARS)을 통해 제공하는 소프트웨어 안전 교육도 제공한다. 소프트웨어 안전 관련 온라인 교육은 다음의 세 가지 주제가 있다.

- SOFTWARE SAFETY PROGRAM PLANS (SwSPPs)
- SOFTWARE FTA & SOFTWARE FMEA
- ESTIMATING PROBABILITIES

온라인 강좌의 내용은 Software Safety Course에서 다루는 내용의 일부분으로 1~2시간으로 구성되었다.

2) IEC 61508 Software Safety Training Course

교육명	IEC61508 Software Safety Training Course		
주최단체	Engineering Safety Consultants (영국)	기관분류	민간 전문기관
교육시간	1D ~ 3D	교육비용	미정
교육내용	IEC61508을 준수하는 안전 관련 소프트웨어 시스템의 명세, 설계, 개발 및 평가에 이르는 전반적인 기법 및 기술을 적용할 수 있는 전문지식		

(1) 개요

IEC 61508 Software Safety Training Course는 국제 표준인 IEC 61508을 안전 관련 소프트웨어 시스템 개발 및 평가에 적용할 수 있도록 교육하는 프로그램이다. 교육에서는 소프트웨어 안전 생명주기에 대한 내용과 안전 관련 소프트웨어 시스템 개발에 적용할 수 있는 방법을 다룬다. 또한 안전 관련 소프트웨어가 만족해야 할 안전 요구사항을 충족하는지 평가할 수 있는 방법에 대한 교육도 포함한다. 결과적으로 교육을 통해 IEC 61508을 준수하는 안전 관련 소프트웨어 시스템 명세부터 시작해 설계, 개발 및 평가에 이르는 전반적인 기법 및 기술을 적용할 수 있는 전문지식을 습득할 수 있다. 교육을 주관하는 기관은 영국의 Engineering Safety Consultants[17]이다.

(2) 교육내용

교육 프로그램은 1일 과정과 2일 과정이 있으며 1일 과정은 2일 과정의 몇몇 내용을 줄인 형태로 구성됐다. 2일 과정의 프로그램은 아래 표와 같다.

<표 3-3> IEC61508 Software Safety Training Course 교육내용

1일차

17) http://www.esc.uk.net/

Session	Title	Coverage
1	IEC 61508 Overview	IEC 61508 overall structure How different parts fit together
2	IEC 61508 Part 3 Overview	How does Part 3 fit in the overall IEC 61508 and E/E/PE system lifecycle Overview of the IEC 61508 Part 3 scope, structure and content Differences between software and hardware Compliance framework for software
3	IEC 61508 Part 3 Software requirements (generic requirements)	Software development lifecycle and safety lifecycle Requirements applicable throughout software development lifecycle Software configuration management Software forward and backward traceability Software verification and validation (V&V) Software modification, Software tool qualification Differences between system and application software

2일차

Session	Title	Coverage
4	IEC 61508 Part 3 Software requirements (specific requirements)	Requirements applicable to specific software development lifecycle stages Software safety requirements specification Software architecture, Software system design Module design, Coding, V&V
5	Software safety in the context of other related standards	Similarities and differences between IEC 61508 and other standards, including IEC 61511, EN 50128, Def Stan 00-55 DO-178, ARP4754, CAP 670
6	IEC 61508 Part 3 potential new developments	IEC 61508 maintenance committee activities Key software safety topics being discussed and debated Software lifecycle, Proven in use Tool qualification, Data safety

(3)기타

Engineering Safety Consultants는 Software Safety관련 교육뿐만 아니라 System Safety와 관련된 다양한 교육을 실시하고 있다:

- Safety Instrumented Systems – TÜV FS Eng
- Introduction to Functional Safety (IEC 61508) – 1 Day Training Course
- Introduction to Safety Instrumented Systems (IEC 61508/IEC 61511) – 3 Day Training Course
- Introduction to Safety Instrumented Systems (IEC 61508/IEC 61511) – 2 Day Training Course
- Introduction to Safety Instrumented Systems for Technicians (IEC 61508/IEC 61511) – 1 Day Training Course
- Failure Mode and Effects Analysis (FMEA) – E-Learning Course
- Introduction to SIL Determination – 1 Day Training Course
- Introduction to SIL Verification – 2 Day Training Course
- Introduction to HAZOP – 1 Day Training Course

3) Using PAScal Safety calculator Software Course[18)]

교육명	Using PAScal Safety calculator Software Course		
주최단체	PILZ	기관분류	민간 전문기관
교육시간	미정	교육비용	미정
교육내용	Safety Function Performance Level(EN ISO 1384901), Safety Integrity Level(EN 62061)		

(1) 개요

기계의 safety function을 디자인, 평가, 검증하기 위한 소프트웨어인 PAScal을 사용하는 실질적인 이해를 돕기 위한 교육과정이다. Pilz의 PAScal Safety Calulator software[19)]는 safety function들의 Performance Level (EN ISO 1384901)과 Safety Integrity Level (EN 62061)을 간단하고 빠르게 계산하고, 그 결과물은 해당 과정이 얼마나 효과적인지 평가하는데 사용될 수 있다.

18) https://www.pilz.com/en-GB/company/news/articles/073271
19) https://www.pilz.com/en-GB/eshop/00105002187038/PAScal-Safety-Calculator

(2) 대상 및 목적

이 강의의 목적은 PAScal 소프트웨어의 실용적으로 사용할 수 있도록 하는 것이다. 대상은 다음과 같다.

- 자동화, 전기, 유지보수, 프로젝트 엔지니어
- 자동화 관리자, 유지보수 관리자
- 안전성 전문가

(3) 교육내용

강의 내용은 미리 정의된 Safety function의 예를 이용하여 PAScal 소프트웨어를 충분히 다룰 수 있는 실용적인 사용방법을 제공한다. 아래 내용들을 포함한다.

- PAScal features and benefits
- Review of EN ISO 13849-1 and EN 62061
- Use of PAScal in detail using pre-defined safety function examples

4) Safety training[20)]

교육명	Safety Training		
주최단체	CRITICAL Software	기관분류	민간 전문기관
교육시간	미정	교육비용	미정
교육내용	DO-178C, DO254 & ED-80, DO-278A		

(1) 개요

강좌는 전문가들과 기술자들의 여러 해 동안의 경험적 재산으로부터 개설되었다. 각 코스들은 강의를 듣는 사람들이 인증된 안전 혹은 임무 필수(mission-critical) 시스템들을 제공하는데 필요한 핵심 기술과 이해를 개발할 수 있도록 설계되어 있다고 한다. 안전 표준과 관련된 가장 최근의 이슈들을 다시 공부하거나, 완전히 새로운 시장에 진입하거나 혹은 안전 필수 시스템(safety-critical system)들을 개발할 필요가 있는 엔지니어들에게 표준 중심의 교육을 제공한다.

20) http://www.criticalsoftware.com/en/what-we-do/safety-training

(2) 교육내용

<표 3-4> CRITICAL Software Safety Training 교육내용

항공	자동차
Applied DO-178C: 항공 인증 Applied DO-254 & ED-80: 항공 하드웨어 인증 DO-278A: 위성항행시스템(CNS/ATM Ground-Based Systems)	자동차 기능 안전
에너지	**의료**
기능 안전	의료기기 소프트웨어 개발: IEC 62304 모바일앱 소프트웨어 개발 : IEC 62304
철도	**우주**
철도 CENELEC[21] 표준 : EN 50126/8/9 ERTMS[22] 개요 철도 시스템 공학 철도 위험 평가 기술	소프트웨어 개발

- 항공 분야 안전 교육

교육과정은 참가자들에게 European Cooperation for Space Standardisation (ECSS)을 소개한다. ECSS의 조직과 Space Product Assurance와 Space Product Engineering branch들의 자세한 조사에 따른 규정들에 대한 개요를 제공한다.

- 자동차 분야 안전 교육

CRITICLA SOFTWARE의 ISO 26262 교육 과정은 ISO 26262에서 정한 개발수명주기 요구사항을 따라 개발되고 통합되어야 하는 차량 임베디드 시스템의 다양한 관점에 집중한다.

- 에너지 분야 안전 교육

엔지니어, 운영자, HSE advisor, 위험관리자 그리고 다른 의사결정자들을 위한 CIRITICAL SOFTWARE의 교육 과정은 어떻게 기능 안전 원리들을 IEC 61508 표준에 따라 안전 시스템의 개발 그리고 평가에 적용해야 할지 자세하게 알려준다.

---

21) 유럽전기기술표준화 위원회, European Committee for Electrotechnical Standardization
22) The European Railway Traffic Management System

- 의료 분야 안전 교육

의류 기구들의 소프트웨어를 개발하는 것은 매우 규제되어 있으며, 생산자에게 위험 관리, 생명주기 관리, V&V(validation and verification), 변화 관리에 대한 이해를 요구한다. 의료 기기 산업에서 위임된 소프트웨어 표준은 IEC 62304이다. 하지만 많은 표준들의 요구사항들은 규정에 의해 정의되어 있지 않아서, 의료기기 생산자들이 어떻게 표준을 가장 잘 해석할 수 있을지에 대한 문제가 남아 있다. CRITICAL SOFTWARE는 의료 산업을 주도하는 조직들에게 안전 관리 그리고 교육을 제공하여, 안전 필수(safety-critical) 소프트웨어와 하드웨어를 개발하고 테스팅하는데 있어 선두주자라고 하고 있다.

- 철도 분야 안전 교육

CRITICAL SOFTWARE는 열차와 철도 시스템들을 위한 CENELEC 표준들과 EN 50126, EN 50128, EN 50129, ERTMS/ECTS 교육을 제공하기 위해 현직에 종사하는 수석 항공 엔지니어들을 활용하고 있다. 게다가 교육 과정은 철도 부분에서 시스템 공학과 위험 평가 원리들도 다루고 있다.

5) Software Safety: Software Development for Safety-Related Systems[23)]

교육명	Software Development for Safety-Related System		
주최단체	Edif Group	기관분류	민간 전문기관
교육시간	2D	교육비용	$995[24)]
교육내용	Software Safety Concepts, Design, Testing, Analysis Technique, Requirements, Planning, Analysis & Arguments		

(1) 개요.

안전성과 관련된 복잡한 시스템들이 목적을 달성하기 위해 소프트웨어에 의존하는 점점 많이 의존하게 됨으로써, 구조화된 소프트웨어 위해도 분석(hazard analysis)는 소프트웨어를 개발하고 위험을 최소화하고 사람의 생명과 장비에 손해를 예방하는데 있어 필수적이 되었다. 민간 항공과 방위 환경에서 채택됨에 따라, Edif Group 자세한 과정은 잠재적인 소프트웨어 안전성 이슈들과 소프트웨어 안전 보증(safety arguments)를 만드는 기법들에 대한 소개를 제공한다.

강의에서는 소프트웨어 안전 개념(software safety concepts), 설계(design), 테스팅, 분

23) http://www.edifgroup.com/training/course/software-safety
24) £800.00, 2016-11-10 환율 기준

석 기술에 더하여 안전 요구사항, 계획, 분석 및 검증 방안을 소개할 것이다. 안전시스템에서 Programmable Logic Devices (PLDs)의 보증 요구사항(assurance requirement)들을 조사하고 안전 시스템에서의 Commercial Off-The-Shelf (COTS) 소프트웨어와 Software of Unknown Pedigree (SOUP) 사용 또한 학습한다.

(2) 교육내용

<표 3-5> Edif Group Software Development for Safety-Related System 교육내용

- Introduction
- Overview and software safety concepts
- Exercise: Software safety argument
- Safety requirements for software
- Group exercise: Software requirements
- Safety Integrity Levels (SILS) for software
- Safety planning for software
- Discussion: Use of software safety design techniques
- Software safety design
- Group exercise: Safety design
- Software testing
- Exercise: Test coverage
- Introduction
- Overview: safety analysis and argument
- Software safety analysis techniques
- Group exercise: Software safety analysis
- Programmable Logic Devices (PLDs)
- Discussion: Extent of use of PLDs
- Coding and anlysis
- Exercise: Code review
- Safety arguments for software
- Group exercise: Safety argument
- Designing a safety system with SOUP
- Group review: Proposed Off-The-Shelf (OST) solution

6) EEA Public Course: System Safety Engineering Management Master Class[25)]

교육명	System Safety Engineering Management Master Class		
주최단체	Engineering Education Australia Pty Ltd	기관분류	민간 전문기관
교육시간	5D	교육비용	$4,530
교육내용	System Safety Engineering/Management		

(1) 개요

현대 시스템에서 최신의 기법에 대한 수요와 효율성이 늘어남에 따라 새로운 디자인의 복잡성과 정교함이 상당한 증가하였다. 이는 복잡해지고 안전 필수 실패 상태(safety critical failure mode)들을 포함할 수 있는 제어 시스템이 소프트웨어에 크게 의존하는 증가되는 결과를 가져왔다. 국내외 관리, 규제 단체들은 안전 시스템들에서 고려되어 있는지 종합적이고 엄격한 증거를 요구하고 있다. 이 과정은 시스템 안전 공학과 관리에 중점을 두고 있다. 시스템 안전 공학의 핵심 분야들, safety case의 개발과 유지, 위해도(hazard) 인식 및 분석 , 위험 감소, 안전 관리 사례연구와 실무적인 문제해결, 실세계 예제들을 다루고 있다.

(2) 대상 및 목적

아래 공학 분야에서 4년 이상의 경험이 있는 전문가

- 항공
- 건설
- 컨설팅
- 국방
- 전기, 에너지
- 제조
- 운송

교육과정의 목적은 다음과 같다.

- Understand safety risk management as applied to engineering design of

25) https://www.engineersaustralia.org.au/portal/event/eea-public-course-system-safety-engineering-management-master-class

systems

- Establish a working knowledge of the relevant standards and guidelines for safety management
- Identify the critical elements of an effective system safety program
- Understand the Safety Case and how to construct an effective safety argument
- Appreciate the importance of maintaining the Safety Case argument
- Apply hazard identification and assessment techniques for safe system design
- Apply risk reduction strategies or safety critical systems
- Appreciate how the human factor applies to design and safety of systems
- Understand the importance of software safety management, software engineering and software related systems
- Apply system safety principles in engineering byway of a detailed practical worked example
- Identify and complete relevant project management documentation

(3) 교육내용

- Overview and comparison of specific military and commercial System Safety standards and requirements for safety management, from different industries such as aerospace, railway and maritime
- Some of the standards considered are: MIL STD-882, DEF STAN 00-56, AS61508, ARP4761
- Comparison of the different safety paradigms of System Safety, Risk Management, Reliability Engineering and OH&S concerns
- Introduction to the System Safety process and integration of the System Safety Program with the project life cycle
- Managing the interfaces to related Safety Programs, the role of constraints and assumptions
- Key elements of a Safety Case
- Building a Safety Case argument - introduction to GSN
- Safety Case maintenance
- Hazard identification, analysis and mitigation
- Qualitative and Quantitative methods of analysis techniques such as fault

tree analysis, event tree analysis, failure mode effects and criticality analysis and others

- Risk management - risk reduction strategies, risk assessment, monitoring and control
- Human factors - identification, modelling, quantification and understanding human error
- Software Safety engineering, software assurance, and software safety standards

2. 대학

1) SFT Course

교육명	SFT Course		
주최단체	University of Southern California	기관분류	대학
교육시간	4D	교육비용	$1,500
교육내용	소프트웨어 개발 및 분석에 대한 철학과 기법, 소프트웨어 안전 관리방법		

(1) 개요

SFT Course는 미국의 University of Southern California에서 개최하는 4일짜리 소프트웨어 안전 교육이다. 이 교육에서는 소프트웨어 개발 및 분석에 대한 철학과 기법에 대한 내용과 소프트웨어 안전을 관리할 수 있는 방법에 대한 내용을 포함하고 있다. 특히 소프트웨어 위해도 분석 기법으로 알려진 Fault Tree/Soft Tree와 Software Sneak Analysis, Petri Nets 기법에 대하여 평가해 보는 시간도 포함한다.

본 교육은 소프트웨어 위해도(Software Hazards), 위험 유발 원인과 여러 기법들에 대한 이해에 주목적을 두고 있다. 소프트웨어 안전 프로그램을 관리하고 확립시키기 위한 관리 방법이나 문서 등에 대한 교육도 포함하고 있다. 추가적으로 Safety Case 작성을 위한 증거도 제공한다.

(2) 교육내용

<표 3-6> USC SFT Course 교육내용

- Software
- Safety Overview
- Definitions and Concepts
- Design Requirements
- Software Regulations/References
- System Safety Team Organization
- Risk Processing/Management
- Risk by Agency
- Hazard and Security
- Catastrophic
- Probability of Occurrence
- Reliability Issues
- Probability
- Hazard Consideration/Analysis
- Risk Assessment and Risk Levels
- Program Documentation
- Software Reliability/Risk
- Software Engineering/Requirements
- Software Safety Life Cycle Goals
- Security Engineering
- VDHL Synthesis
- Error Classification and Types
- Software Safety Requirements Traceability
- Petri-Net Modeling
- Software Safety Checklist
- Preliminary Hazard Analysis
- Software Language Analysis
- Fault Tree Analysis
- Formal Mathematical Models
- Software Safety Testing
- Testing Schemes/Strategies
- Software Safety Reliability/Maintenance
- Risks: Cumulative Index of Software Engineering Terms
- Analyzing Safety and Fault Tolerance Using Time Petri-Nets
- Software Sneak Analysis (SSA) Fact Sheets

2) Engineering Safety- and Security-Related Requirements for Software-Intensive Systems[26)]

교육명	Engineering Safety and Security Related Requirements for Software-Intensive Systems		
주최단체	Carnegie Mellon University	기관분류	대학
교육시간	2D	교육비용	$2,100
교육내용	Safety, Security, Requirements engineering의 공통부분		

(1) 개요

이 이틀간의 교육과정은 안전, 보안 그리고 요구 공학의 공통부분을 다룬다. 안전과 보안은 위험(hazards and threats)으로 인한 허용하지 않는 피해로부터 가치 있는 자산을 지키기 위한 요구공학에서 자연스레 사용하는 위험 기반 접근 방식, 분석 기술과 많은 공통적인 부분을 갖는다.

많은 소프트웨어 중요 시스템들은 안전과 보안의 중요한 세부사항들을 가지고 있으며 안전과 보안이 관련된 요구사항 속성들이 작성될 필요가 있다. 예를 들면, 몇몇 컨설턴트들, 연구자들, 작성자들이 말하길, 부적절한 요구사항들이 소프트웨어 중요 시스템들을 포함한 사고들(accidents)의 주된 원인이라고 한다. 아직도 실제로는, 요구사항과 안전, 보안 학문들에는 아주 적은 교차점만 있으며 그들 각각의 커뮤니티들에서도 약간의 공동연구만 진행되고 있다. 대부분의 요구사항 공학자들은 안전과 보안 공학에 대해 일부밖에 알지 못하며, 안전과 보안 공학자들도 요구 공학에 대해 잘 알지 못한다. 또한, 안전과 보안 공학은 전통적으로 요구사항보다 아키텍처들과 디자인에 집중한다. 왜냐하면, 위해도 분석은 전통적으로 사고(accident)를 발생시키고 성공적인 공격(attacks)을 할 수 있게 하는 실패(failure)나 약탈(exploitation)의 대상이 되는 취약한 하드웨어와 소프트웨어 컴포넌트를 정의하는 것에 의존하기 때문이다. 이는 안전과 보안 관련 요구사항들이 자주 모호하고, 불완전하며 심지어 누락되는 것을 야기한다.

(2) 대상 및 목적

이 교육의 예상 청중은 다음과 같다

- Requirements engineers who must collaborate with safety and security engineers to engineer the safety- and security-related requirements.
- Safety engineers who must perform the hazard and risk analysis that drives the safety-related requirements and who must collaborate with requirements

26) http://www.sei.cmu.edu/training/P64.cfm

engineers to engineer these requirements.

- Security engineers who must perform the threat and risk analysis that drives the security-related requirements and who must collaborate with requirements engineers to engineer these requirements.
- Stakeholders in the safety- and security-related requirements including subject matter experts, customer representatives, architects, software engineers, testers, and certifiers.

전체적인 목적은 교육 참가자들에게 소프트웨어 중요 시스템을 위한 안전 그리고 보안과 관련된 요구사항들을 어떻게 작성하는지 가르지는 것이다. 구체적인 목표들은 아래와 같은 내용을 배우는 것을 포함한다.

- Fundamental concepts underlying the engineering of safety- and security-related requirements.
- Different types of safety- and security-related requirements including their purpose and composition.
- Basic tasks of safety and security engineering that are related to engineering safety- and security-related requirements.
- Relationship between safety and security quality subfactors and the quality criteria specified in safety and security requirements.

(3) 교육내용

- Fundamental concepts underlying the engineering of safety- and security-related requirements.
- Different types of safety- and security- related requirements including their purpose and composition.
- Basic tasks of safety and security engineering that are related to engineering safety- and security- related requirements.
- Relationship between safety and security quality sub-factors and the quality criteria specified in safety and security requirements.

3) Systems Safety Engineering (ENGG4020)[27]

교육명	System Safety Engineering		
주최단체	The University of Queensland, Australia	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	HAZOP, CHAZOP, FFA, FTA, ETA, FMEA, FMECA, GSN 등		

(1) 개요

안전은 시스템의 모든 관점에서 하드웨어, 소프트웨어, 계획, 개발, 테스팅, 유지보수, 이관 그리고 다른 안전 프로그램에서 고려되는 다른 관점들까지 모든 생명 주기와 관련된 이슈이다. 안전 필수 시스템에서는 단순히 안전시스템을 개발하는 것으로는 부족하다. 해당 시스템은 안전하다는 것을 보여주어야 한다. Safety case의 개발은 이런 시스템에서 중요하다. 빠른 시기에 안전 이슈들을 정의하고 시스템의 안전을 평가하는 것은 뒤에 필요할 재작업을 방지할 수 있다.

(2) 대상 및 목적

대학생을 대상으로 하는 강의이며 강의의 목적은 다음과 같다.

- Explain basic system safety principles and the purpose and structure of safety cases and hazard logs
- Describe the role and responsibilities of Safety Engineers
- Carry out Preliminary Hazard Analysis for a system using techniques including Functional Failure Analysis and Event Tree Analysis
- Carry out System Hazard Analysis using techniques including CHAZOP, FTA and FMEA
- Identify and specify functional safety requirements for a given system or system design
- Give examples of system safety design techniques
- Interpret and apply key industry system safety standards

(3) 교육내용

강의에서 안전관리의 원리와 실습, 컴퓨터 기반 시스템들에 대해 설명한다. 내용은 안전 분석 기법들의 실무 경험과 관리 및 개발 문제에 대한 논의로 구성되어 있다. 포함하는 주제들은 다음과 같다.

27) https://www.uq.edu.au/study/course.html?course_code=ENGG4020

- Hazard identification and risk analysis, safe system design, safety analysis techniques, safe software engineering, system hazard analysis, safety cases, safety management and human factors.

기법들은 다음과 같다.

- Hazard and Operability Studies (HAZOP) and Computer Hazard and Operability Studies (CHAZOP), Functional Failure Analysis (FFA), Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Modes and Effects Analysis (FMEA) and Failure Modes Effects and Criticality Analysis (FMECA), and Goal Structured Notation (GSN).

4) Computers & Safety: University of York[28)]

교육명	Computers & Safety		
주최단체	University of York	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	Software intensive 시스템들의 Requirement Specification, Design Analysis		

(1) 개요

교육과정은 주로 시스템 안전성 공학자들에게 안전 필수 시스템이나 안전 관련 업무에서 사용되는 소프트웨어 기반 시스템들이 반드시 고려해야 할 이슈들을 소개한다. 과정은 기본 하드웨어 컴포넌트에서 소프트웨어까지 컴퓨터 시스템들이 어떻게 동작하는지에 대한 개요로 시작한다. 이 도입부를 통해 안전 공학자들이 잠재적으로 고려할 만한 부분들을 강조한다. 이후, 안전 필수 분야에서 소프트웨어 중요 시스템들의 개발에 중요한 요구사항 분석, 설계를 고려하여 좀 더 소프트웨어 개발 프로세스에 대한 좀 더 자세한 고찰로 이어진다. 또한, 교육과정은 software safety case의 증거를 수집하고 조직하는 것에 대해서도 다룬다.

28) https://www.cs.york.ac.uk/postgraduate/modules/casa.html

(2) 교육내용

<표 3-7> University of York Computers & Safety 교육내용

- Computer and Software Basics for the Systems Engineer 1
- Software Safety Principles
- Software Development in Context
- System & Software Requirements
- Adapting Analysis Techniques for S/W
- Software Architecture and Design
- Design Modelling
- Software Implementation
- Software Verification
- Handling Change
- Software Safety Standards
- Establishing Software Safety Arguments and Selecting Evidence
- Trends in Software

5) Systems and Software Safety[29)]

교육명	Systems and Software Safety		
주최단체	Australian National University	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	Safety Criticality 결정 및 관련 이슈, Hazardous Requirements 결정 기법 등		

(1) 개요

이 교육과정은 안전 중요도를 결정하는 것과 어떻게 그 결정이 다양한 시스템 그리고/혹은 소프트웨어 프로젝트 활동들과 관련된 주요 이슈들을 다룰 것이다. 위험 관련 요구사항을 결정하기 위한 기법들과 그것들을 어떻게 다룰지에 대해, 디자인과 코드에서 안전 실패를 설정하기 위한 기법들과 함께 설명되고 시연된다.

(2) 교육내용

Safety Criticality 결정 및 관련 이슈, Hazardous Requirements 결정 기법 등

29) http://programsandcourses.anu.edu.au/course/COMP8180

6) System Safety for Software-Intensive Systems[30)]

교육명	System Safety for Software-Intensive Systems		
주최단체	MIT Aeronautics and Astronautics Dept.	기관분류	대학
교육시간	5D	교육비용	$3,250
교육내용	System/Software Hazard Analysis, Software Requirement Specification/Modeling and Analysis, Principles of Safe Design, Verification and Validation of Safety		

(1) 개요

사고의 원인이 변화하고, 안전성을 보장하기 위해 엄청난 표준 접근법들에 의해 복잡성의 정도가 증가하는 시스템이 만들어 지고 있다. 이 과정은 소프트웨어와 사람 컴퓨터 관계와 같은 복잡한 시스템은 전통적인 시스템 안전 기법들로 다루기 어렵다는 점을 강조하며 안전성을 만들고 보장할 수 있는 기법들과 개념들을 다룰 것이다. 전통적인 시스템 안전성이 다루어질 것이며, STAMP 모델과 도구들을 포함한 위해도 분석, 위험관리를 위한 새롭고 독창적인 접근 방식들도 다룰 것이다. 오늘날의 프로젝트에 실제로 적용할 수 있는 절차들과 기법들도 중점적으로 볼 것이다. 이런 기법들이 서로 다른 분야에서 사용된 실 프로젝트 경험들이 설명되고 최근의 소프트웨어 관련 사고들도 검토되고 분석될 것이다. 이 수업의 목적은 안전성 표준을 어떻게 만족할 수 있는지를 배우는 것이 아니라, 프로젝트에 가장 효과적일 수 있는 잘 맞는 디자인을 할 수 있는 문제에 대한 깊은 이해를 얻는 것이다. 학생들은 작은 그룹들로 예제들을 공부할 것이다.

(2) 교육내용

- The Problem:
  - ■ Accident Causes
  - ■ Computers and Risk
  - ■ Safety vs. Reliability
- A New Holistic, Control-Based Approach to System Safety
- System Hazard Analysis for Complex, Software-Intensive Systems
- Software Hazard Analysis
- Software Requirements Specification/Modeling and Analysis

30) http://sunnyday.mit.edu/announce09.html

- Principles of safe design
  - ■ System and Software
  - ■ Human-Machine Interaction
- Verification and Validation of safety
- Organization and Management of Safety-Critical Projects

7) Techniques for System Safety Analysis Course

교육명	Techniques for System Safety Analysis Course		
주최단체	Technische Universität München	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	FTA, FMEA, STPA		

(1) 개요

Techniques for System Safety Analysis Course는 독일의 뮌헨 공과대학교 (Technische Universität München) 정보학부 (Fakultät für Informatik) Software & System Engineering Research Group[31]이 개설한 실습 강좌이다.

강좌는 시스템 안전성과 안전 공학에 대하여 소개하고, 실제 산업에서 자주 사용되는 기법인 FTA, FMEA, STPA 분석 기법에 대하여 소개한다. 또한 그룹 활동을 통하여 실제 시스템 개발 과정과 유사하게 요구사항을 명세하고 설계하는 과정에서 안전성 분석 기법을 적용하는 등의 활동을 수행한다.

(2) 교육내용

<표 3-8> Technische Universität München Techniques for System Safety Analysis Course 교육내용

- Home Preparation
  - FTA Basics
  - FMEA Basics
  - STPA Basics
  - References
  - Exercise
- Analysis Sessions
- Assurance Session
  - Safety cases
  - Assurance Reporting
  - Defect/Hazard Classification
  - SysML Reference Card
- Final Workshop

31) www4.in.tum.de/index.shtml

- Group Assignment
- Requirement Specification Template
- Analysis Reporting Template

• Group Interviews

(3) 기타

2016년 현재 Software & System Engineering Research Group의 이전 강좌/실습 강좌 및 세미나 목록[32]에서는 해당 실습강좌이외에 시스템 또는 소프트웨어 안전성에 대한 추가적인 강좌는 확인할 수 없다.

8) Software Safety Course

교육명	Software Safety Course		
주최단체	Johns Hopkins University	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	Safety-critical System, Safeware, System Hazard		

(1) 개요

Software Safety Course는 미국의 존스 홉킨스 대학교 (Johns Hopkins University) 화이팅 공과대학(Whiting School of Engineering)[33]에 개설된 강좌로, 컴퓨터 과학 (Computer Science) 석사 학위 과정의 소프트웨어 공학 트랙에 편성되어 있는 강좌이다.

강좌는 소프트웨어 안전에 대한 개념을 소개하고, 안전 필수 시스템 (Safety-critical system)에 사용되는 소프트웨어를 어떻게 개발하고 사용할 것인지에 대한 내용을 소개한다. 안전 소프트웨어 (safeware)를 개발하기 위한 시스템 공학 및 소프트웨어 공학 기법을 소개하고, 시스템 위해도의 개념과 이를 분석할 수 있는 기법들에 대하여 소개한다. 이에 따라 시스템 또는 소프트웨어의 심각한 고장 상황을 회피하기 위한 예제 활동을 수행한다. 또한 소프트웨어 공학의 위험 분석, 위해도 분석, fault tolerance, safety tradeoffs에 대한 패러다임이 포함된다.

32) www4.in.tum.de/lehre/index.shtml
33) http://ep.jhu.edu

(2) 교육내용

<표 3-9> Johns Hopkins University Software Safety Course 교육내용

- Introduction - Motivation
- Causes, Roles, Criteria
- System Safety
- System/Software Safety Process
- Hazard and Risk Analysis
- Software Safety Requirements Analysis
- Developing Safety-Critical Systems
- Software Safety Design
- Safety-Critical Hardware and Software
- Systems-Theoretic Accident Modeling and Processes based hazard analysis
- Testing

9) System Safety and Certification Course

교육명	System Safety and Certification Course		
주최단체	Embry-Riddle Aeronautical University	기관분류	대학
교육시간	3D	교육비용	$1,500
교육내용	Aviation Safety & Accident Investigation		

(1) 개요

System Safety and Certification Course는 미국 엠브리-리들 항공대학교 (Embry-Riddle Aeronautical University)에 개설된 강좌로, Aviation Safety & Accident Investigation Programs 중 하나이다.

강좌는 안전성 개념에 대하여 설명하고, 각 산업 분야 (regulated industries)에서의 안전 필수 시스템 (safety critical system)의 소프트웨어에 대한 요구사항 명세, 설계, 테스팅, 유지보수를 수행할 때의 이슈들에 대하여 소개하며, 또한 위해도 분석, FMEA, 실패 요인, fault-tolerant 등의 안전성 기법에 대하여 소개한다. 강좌는 또한 신뢰성 공학 (reliability engineering)의 기본적인 개념을 설명하고 안전성과 신뢰성의 관계에 대하여 소개한다. 소프트웨어의 테스팅, 검증, 인증에 대하여 각 산업 분야의 표준과의 이슈 또한 논한다.

(2) 교육내용

해당 강좌의 구체적인 커리큘럼은 공개되어있지 않다.

(3) 목표

- Describe the essential terms and concepts of system safety, reliability, and fail-safe operations
- Demonstrate and apply basic techniques of hazard and risk analysis
- Describe role of safety analyses as a part of high integrity software/system lifecycle
- Recognize selected tools supporting safety analysis study
- Describe concepts of fault tolerance
- Describe and classify safety standards and guidelines issued by regulatory and advisory groups
- Identify certification process, activities, and artifacts in the development lifecycle

(4) 기타

엠브리-리들 항공대학교에서는 해당 강좌 이외에도 항공 안전과 관련된 여러 강좌를 주최하고 있다. 다만 해당 강좌 이외에 소프트웨어의 안전성에 대한 강좌는 확인할 수 없다.

10) Software Safety Course

교육명	Software Safety Course		
주최단체	Åbo Akademi University	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	FMEA, FTA, FFA, HAZOP, SIL 개념 및 개발 절차 등		

(1) 개요

Software Safety Course는 핀란드의 오보 아카데미 (Åbo Akademi University)[34]에 2010년 가을학기에 개설되었던 강좌로, 강의 세션과 실습 세션으로 나누어 총 21회 (주 2회, 2시간)의 세션으로 구성되어 있다.

강좌는 안전 필수 시스템(Safety-critical Control system)에 대하여 설명하고, 안전 필수 시스템의 개발 및 분석에 관한 내용으로 시작한다. 강좌는 소프트웨어의 안전성에 관하여 폭넓게 다루고 있다. 강좌는 fault, error, failure의 개념, 시스템 및 소프트웨어의 safety, reliability, dependability의 개념, 시스템 및 소프트웨어의 안전성을 분석하기 위

34) http://www.abo.fi

한 FMEA, FTA, FFA, HAZOP 등의 기법에 대해서 다루며, SIL (Software Integrity Level) 개념과 안전 소프트웨어를 개발하기 위한 개발 절차에 대해서도 다루고 있다. 강좌는 특히 하드웨어와 밀접하게 연결되어 있는 임베디드 컨트롤 시스템을 중점적으로 다루는 것으로 생각되며 많은 예제가 임베디드 컨트롤 시스템을 대상으로 하고 있다.

(2) 교육내용

- Programming control systems at an application Level
- Simulation of the behavior of controlled processes
- Simulating behavior of faulty hardware
- Techniques for safety analysis: FMEA, FTA, FFA, HAZOP
- Deriving software requirements from safety analysis
- Modelling requirements using cases, state diagrams Allocating safety requirements
- Architecting safety critical systems, system partitioning
- Safety kernel
- Layered approach to architecting
- Verification: overview of static and dynamic testing
- Safety-critical systems development life-cycle
- Safety integrity levels
- Brief introduction to formal methods

11) Safety and Security of Software Products Course

교육명	Safety and Security of Software Products Course		
주최단체	Jönköping University	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	Safety/Security 개념 소개, 표준, 분석/평가 도구		

(1) 개요

Safety and Security of Software Product Course는 스웨덴의 Jönköping University[35] 의 공학 대학 (School of Engineering) 컴퓨터 과학 및 정보학부(Computer Science and

35) http://ju.se

Informatics)에 개설되어 있는 강좌로, Software Product Engineering 프로그램 (Product Development, Specialisation Software Product Engineering 석사 학위 과정)에 편성되어 있다.

강좌는 소프트웨어 분야에서의 안전과 보안 개념에 대하여 소개하고, 소프트웨어를 개발에서의 안전과 보안 표준에 대하여 설명한다. 또한 안전과 보안 특성을 분석하고 평가할 수 있는 몇몇 도구에 대해서도 소개하고 살펴보는 것으로 생각된다. 강좌는 안전과 보안의 공통점과 차이점을 파악하고, 이들의 원인 요소를 분석하는 방법에 대하여 논하는 것으로 생각되며, 또한 소프트웨어 안전과 보안의 상호 의존적인 특성에 대해서도 논하는 것으로 생각된다.

(2) 교육내용

- Safety and security failures of software systems
- The human and organizational aspects of software safety and security
- The role of safety and security standards
- Safety versus security, and crosscutting issues
- Assessing safety and security (including risk management and hazard analysis)
- Software dependability engineering (including availability, reliability, redundancy, recovery and survivability)
- Designing for safety and security, and defensive techniques
- Safety and security software assurance

12) Software Engineering Concepts Course

교육명	Software Engineering Concepts Course		
주최단체	Massachusetts Institute of Technology	기관분류	대학
교육시간	12 Weeks	교육비용	미정
교육내용	소프트웨어 공학, 시스템 안전, PHA, FTA, RA, STAMP방법론 및 STPA		

(1) 개요

Software Engineering Concepts Course는 MIT 공과대학 (Massachusetts Institute of Technology)[36]에서 제공하는 OCW (Open Course Ware)[37]에서 2005년 가을에 개설되

36) http://web.mit.edu

었던 강좌로, 총 12회의 (주 1회, 3시간) 세션으로 구성되어 있다.

강좌는 소프트웨어 공학 관점에서 폭넓은 주제를 다루고 있다. 소프트웨어 개발 과정에서 어떠한 문제가 발생하는지부터 시작하여, 소프트웨어 공학 관점에서의 소프트웨어 문제에 대한 이해, 소프트웨어 개발 모델에 대한 소개, 소프트웨어 요구사항 명세 및 설계 방법과 이미 개발된 소프트웨어의 재사용에 대한 이슈 등을 다룬다. 뿐만 아니라 소프트웨어의 품질을 나타낼 수 있는 지표(Software Metrics)와 이를 분석할 수 있는 여러 방법들, 프로그래밍 언어와 소프트웨어의 관계, 팀 개발 모델 등에 대해서도 다루고 있다. 또한 강좌는 소프트웨어 안전성에 대해서도 심도 있게 다루고 있으며, 시스템 안전성 개념에 대하여 소개하고, 안전성을 위한 접근 방법과 절차들에 대하여 소개한다. 여기에는 PHA, FTA, 요구 분석 등의 접근 방법과, STAMP 방법론 및 STPA가 포함되어 있다.

(2) 교육내용

<표 3-10> MIT Software Engineering Concepts Course 교육내용

- Introducing The Problem
- Process and Life Cycle Models
- Requirements and Specification
- Design
- COTS and Reuse
- Metrics and Reliability Assessment
- Building Confidence (Testing, Analysis, QA, Reviews)
- Selecting a Programming Language
- Team Organization and People Management
- Software and System Safety
- Putting It All Together

해당 강좌는 각 주제별로 관련 문헌 학습과 토의를 통해 진행된다.

(3) 다루는 문헌

<표 3-11> MIT Software Engineering Concepts Course에서 다루는 문헌

Lec #	Topics	Readings
1	Introducing The Problem	• Leveson, Nancy G. "The Therac-25 Accident." • Lewyn, Mark. "Flying in Place: The FAA's Air Control Fiasco." (April 26, 1998): 87-88. • Augustine, Norman R. "Yes, But Will It Work in Theory?" Norman

37) http://ocw.mit.edu

		Lockheed Martin Corporation. • Brooks, Frederick, Jr. “No Silver Bullet.“ In .EditedbyH.J.Kugler.North-holland,BV:ElseviaSciencePublishers,1986.ISBN:0444700773. • Davis, Alan. “Software Lemmingineering.“ (September 1993): 79-82. • Leveson, Nancy G. “The Role of Software in Spacecraft Accidents.“ • Ewusi-Mensah, Kweku. “Critical Issues in Abandoned Information Systems Development Projects.“ 50, no. 9 (September 1997).
2	Process and Life Cycle Models	• Spiral Model • Paulk, Mark C., Bill Curtis, Mary Beth Chrissis, and Charles V. Weber. “The Capability Maturity Model for Software.“ Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute. (Unpublished.) • Bach, James. “The Immaturity of CMM.“ (September 1994). (Courtesy of James Bach. Used with permission.) • ———. “Enough About Process: What we Need are Heroes.“ (March 1995): 96-98. • Gray, Lewis. “No Cowboy Programmers!“ (April 1998). • Mackey, Karen. “Mars versus Venus.“ (May/June 2000): 14-15. • McConnell, Steve. “Cargo Cult Software Engineering.“ (March/April 2000): 11-13. • Martin, Robert C. “eXtreme Programming.“ (July/August 2000): 12-13. • Highsmith, and Cockburn. “Agile Software Development: The Business of Innovation.“ (September2001):120-122. • ———. “Agile Software Development: The People Factor.“ (November 2001): 131-133. • Pressman. “Can Internet-Based Applications be Engineered?“ (September/October 1998): 104-110. • Bollinger, Terry. “The Interplay of Art and Science in Software.“ (October 1997): 128-131.
3	Requirements and Specification	• Lawrence, Brian, Karl Wiegers, and Christof Ebert. “The Top Risks of Requirements Engineering.“ (November/December 2001): 62-63. • Leishman and Cook. “Requirements Risks Can Drown Software

		Projects." (November 2001).
		• Wing, Jeannette M. "A Specifier's Introduction to Formal Methods." (September 1990): 9-23.
		• Leveson, Nancy G. "Intent Specifications: An Approach to Building Human-Centered Specifications."
4	Design	• Parnas, D. L. "On the Criteria To Be Used in Decomposing Systems into Modules." 15, no. 12 (December 1972): 1053-1058.
		• ———. "Designing Software for Ease of Extension and Contraction."
		• Wirth, Niklaus. "Program Development by Stepwise Refinement." 14, no. 4 (April 1971): 221-227.
		• Bergland, G. D. "A Guided Tour of Program Design Methodologies." (October 1981): 13-37.
5	Design (cont.)	• Garlan, David, and Mary Shaw. "An Introduction to Software Architecture."
		• Hatton, Les. "Does OO Sync with How We Think?" (May/June 1998): 46-54.
		• Meyer, Bertrand. "A Really Good Idea." (December 1999): 144-147.
		• Vessey, Iris, and Sue A. Conger. "Requirements Specification: Learning Object, Process, and Data Methodologies." 37, no. 5 (May 1994).
		• Ledgard, Henry F. "The Emperor with No Clothes." 44, no. 10 (October 2001).
		• Carpenter, Todd. "Avionics Integration for CNS/ATM." (December 1998): 124-126.
		• Budgen, David. "Software Design Methods: Life Belt or Leg Iron?" (September/October 1999): 136-139.
6	COTS and Reuse	• Lions, J. L. "Ariane 5 Accident Report." July 19, 1996.
		• Kruger, Charles W. "Software Reuse." 24, no. 2 (June 1992).
		• Weyuker, Elaine J. "Testing Components-Based Software: A Cautionary Tale." (September/October1998):54-59.
		• Glass, Robert L. "Reuse: What's Wrong with This Picture?" (March/April 1998): 57-59.
		• Leveson, Nancy G., and Kathryn Anne Weiss. "Making Embedded Software Reuse Practical and Safe."
7	Metrics	• Joseph K. Kearney, Robert L. Sedlmeyer, William B. Thompson,

	and Reliability Assessment	Michael A. Gray, and Michael A. Adler. "Software Complexity Measurement." 29, no. 11 (November 1986). • Armour, Phillip. "Ten Unmyths of Project Estimation." 45, no. 11 (November 2002).
8	Building Confidence (Testing, Analysis, QA, Reviews)	• Schaible, Dawn M., and Keith J. Britton. "Testing in NASA Human-Rated Spacecraft Programs: How Much is Just Enough?" SDM Master's Thesis. MIT, February 2003. (Only chapters 2, 4, and 5 are required reading.) • Yamaura, Tsuneo. "How to Design Practical Test Cases." (November/December 1998): 30-36. • ———. "Why Johnny Can't Test?" (March/April 1998): 113-115. • Hoare, C. A. R. "An Axiomatic Basis for Computer Programming." 12, no. 10 (October 1969). • Lipton, Richard J., Richard A. De Millo, and Alan J. Perlis. "Special Process and Proofs of Theorems and Programs." 22, no. 5 (March 1979). • Rothman, Johanna. "Of Crazy Numbers and Release Criteria." (December 1998): 127-128.
9	Selecting a Programming Language	• Naiditch, David. "Selecting a Programming Language for Your Project." (September1999):11-14. • Ziegler, Stephen F. "Comparing Development Costs of C and Ada." March 30, 1995. (Unpublished.) • Glass, Robert L. "One Giant Step Backward." 46, no. 5 (May 2003): 21-23.
10	Team Organization and People Management	• Kruper, K., Dr. Personality Types. (Additional class notes.) • Ferdinandi, Patricia L. "Facilitating Communication." (September/October 1998): 92-96. • McConnell, Steve. "Problem Programmers." (March/April 1998): 126-128. • Mantei, Marilyn. "The Effect of Programming Team Structures on Programming Tasks." 24, no. 3 (March 1981): 106-113. • Rettig, Marc. "Software Teams." 33, no. 10 (October 1990): 23-27. • Williams, Laurie, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. "Strengthening the Case for Pair Programming." (July/August 2000): 19-25. • Howard, Alan. "Software Project Management." 44, no. 5 (May 2001).

11	Software and System Safety	• Leveson, Nancy. “A New Accident Model for Engineering Safer Systems.“ • ———. “A Systems Theoretic Approach to Safety Engineering.“ (Read only the case study, not the introduction.) • ———. “A Systems-Theoretic Approach to Safety in Software-Intensive Systems.“ (Read only the case study, not the introduction.)
12	Putting It All Together	• Leveson, Nancy. “Software Engineering: A Look Back and A Path to the Future.“ December 14, 1996. • Shapiro, Stuart. “Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering.“ 19, no. 1 (1997): 20-54. • Wirth, Niklaus. “A Plea for Lean Software.“ (February 1995): 64-68. • Bach, James. “What Software Reality is Really About.“ (December 1999): 148-149.

13) SAFETY-CRITICAL SOFTWARE

교육명	Safety-critical Software		
주최단체	UCLA Extension	기관분류	대학
교육시간	3D	교육비용	미정
교육내용	임베디드 소프트웨어를 위한 Safety-critical System의 디자인 에러		

(1) 개요

Safety-critical software[38]는 미국의 UCLA Extension에서 주최하는 3일짜리 소프트웨어 안전관련 단기 강좌이다. UCLA Extension은 1917년에 설립된 평생 교육 기관으로 항공분야부터 에너지, 프로젝트 매니지먼트, 소프트웨어 공학까지 다양한 교육을 현재 진행 중이다.

해당 강좌에서는 안전 필수 시스템에서의 소프트웨어가 제공하는 실패의 유형과 이것들을 찾고 방어할 수 있는 기법들에 대해 강의한다. 강의는 잠재적으로 안전 필수 시스템이 될 임베디드 소프트웨어를 개발하고 관리하는 사람을 위한 강좌이며, 오로지 기본적인 프로그램 실력만 있다면 지원 가능하다. 해당 강좌는 안전 필수 시스템의 디자인 오류의 일반적인 형태에 대한 의식을 제고하며, 결과적으로 이러한 오류와 오류를 어떻게 방지할 수 있는지에 대해 강의가 진행된다.

38) http://shortcourses.uclaextension.edu/819-364/

(2) 교육내용

- Day 1
  - Software risks and vulnerabilities
  - Frequently occurring anomalies
- Day 2
  - Designing reliable systems from unreliable parts: the system view
  - Developing safety-critical code
  - Coding standards and compliance verification
  - Defensive coding techniques
  - System structure, redundancy, defense in depth
- Day 3
  - Design, development, and test methodologies
  - Modeling and analysis techniques
  - Tool-based code review processes
  - Software safety-related standards (NASA, IEEE, ARINC, DOD, etc.)

(3) 기타

강의를 신청하면 강의 자료를 무료로 제공하고 있다.

14) SOFTWARE SAFETY

교육명	Software Safety		
주최단체	Detroit University	기관분류	대학
교육시간	1 Semester	교육비용	미정
교육내용	소프트웨어 공학 측면에서 공정 및 품질, 안전계획 수립, 적용, 분석 등		

(1) 개요

Software safety는 미국의 Detroit 대학의 소프트웨어 안전 관련 강좌이다. 해당 강좌는 소프트웨어 공학의 대학원생을 위한 수업으로 소프트웨어공학 커리큘럼 중 한 과정인 3학점 강좌이다.

해당 강좌에서는 학생들에게 소프트웨어 공학 측면에서 공정 및 품질, 안전 계획 수립, 적용, 분석 등의 강의를 진행하고, 소프트웨어 공학에서의 과제를 해결하기 위한 적절한 해결책을 평가하고 평가할 수 있도록 강의가 진행된다. 팀 과제를 통하여 학생들이 팀에서 생산성과 효율성 측면에서의 능력을 배양하고, 석사과정 학생들이 소프트웨어 공학 분야의 이론과 실제 연구를 평가하는 능력을 배양하여 박사과정 학생이 될 수 있도록 지도한다.

(2) 입학 요구사항

이 수업을 듣기 전에 학생들은 최소 3.0 이상의 학사 학위를 보유해야 하며, C++ 이나 Java같은 현대적인 프로그래밍 언어 과정, 데이터 구조 및 알고리즘, 컴퓨터구조 미적분과 이산 수학 등의 과정을 수행이 요구된다.

(3) 기타

Detroit 대학의 소프트웨어 공학과의 정규편성 과목 중 하나이다.

3. 공공기관

1) Safety Critical Systems Course[39)]

교육명	Safety Critical Systems Course		
주최단체	IET(The Institution of Engineering and Technology)	기관분류	공공기관/인증기관
교육시간	미정	교육비용	미정
교육내용	IEC61508		

(1) 개요

교육과정에 참석하는 대표자들은 기업과 학교의 안전 필수 시스템 전문가들이 지도하는 강의, 실무 워크샵, 토론을 통해 IEC 61508을 어떻게 해석하고 적용할 수 있을지 배우게 된다. 이 교육 과정은 IEC 61508에 대한 소개와 위험 개념, IEC 61508표준, 하드웨어 아키텍처, 안전 보증 그리고 소프트웨어 안전을 포함하고 있다.

39) http://conferences.theiet.org/scs/about/index.cfm

⑵ 대상 및 목적

이 교육 과정을 통해 다음과 같은 것들이 가능할 것이다.

- Develop your understanding of the key aspects of IEC 61508
- Equip you with an appreciation of the various phases of the safety lifecycle in a systematic manner
- Prepare you to manage systems, hardware, software and human factor issues
- Help you to analyse risk based approaches to the development of the safety requirements

⑶ 교육내용

<표 3-12> IET Safety Critical Systems Course 교육내용

- Introduction to course
- Safety critical systems-legal implications
- Risk concepts and cost benefit analysis
- Introduction to IEC 61508
- Eurotunnel specific fire risk and drive towards mitigation measures
- Safety Integrity levels (SILs) and overall design framework
- SIL determination
- Work shop task 1
- Random hardware failures
- IEC 61508 - 2 structure and essentials
- Hardware architectures for meeting specified SILs
- Safety assurance - functional safety assessment
- Automotive safety integrity levels
- Workshop task 2
- Competence and competence management systems
- Operations and maintenance
- IEC 61508 and major hazards safety regulations
- Human factors
- Management of functional safety
- The deepwater horizon catastrophe - lessons learnt

2) ATM Software Safety Assessment Course

교육명	ATM Software Safety Assessment Course		
주최단체	Eurocontrol	기관분류	공공기관/인증기관
교육시간	5D	교육비용	$1,638.90[40]
교육내용	ATM(Air Traffic Management) 소프트웨어의 안전		

(1) 개요

ATM Software Safety Assessment Course는 Eurocontrol[41]에서 주최하는 강좌로, 항공 교통 관제(ATM: Air Traffic Management) 소프트웨어의 안전에 관한 5일 간의 프로그램으로 구성되어 있다. Eurocontrol은 유럽 전체의 항공 교통 관제를 위해 설립된 국제 민간기구로, 유럽 전체 항공 교통 관제의 계획과 관리를 주 업무로 하여 항법서비스 제공, 항공 교통 관제 관련 프로그램의 연구 및 개발, 항공 관제사 훈련 등의 업무를 수행하는 조직이다. 해당 강좌의 요금은 1인당 € 1,500 로 책정되어 있으며, 회원국의 인증된 기관 소속 인원은 Eurocontrol의 지원을 받아 무료로 참석할 수 있다.

강좌는 13개 항목으로 나뉘어져 있으며 소프트웨어 안전성과 관련된 사고 사례를 살펴보고 소프트웨어 안전성에 대한 중요성을 설명하는 것으로 시작한다. 강좌는 Software Assurance Level (SWAL) Concept에 대하여 소개하며, SWAL concept에 따른 소프트웨어 개발 과정 별 행동에 대한 예제를 수행하는 것으로 생각된다. 또한 관련 개발 표준 및 가이드라인을 살펴보는 것으로 생각된다.

(2) 교육내용

<표 3-13> Eurocontrol ATM Software Safety Assessment Course 교육내용

- Software and Software Safety
- Example Accidents
- Software Impact on Safety
- Software and Safety Lifecycles
- System and Software Requirements
- Assurance Levels
- Requirements – Exercise
- Software Safety Assurance
- EC 482 – SAM – SWAL
- Software Standards and Software Standards Exercises
- Software Supply Models and Contracting for Assurance
- Practicalities of a Project and exercises
- Case Study

40) €1,500, 2016-11-10 환율 기준
41) http://www.eurocontrol.int

(3) 필수 요건

해당 강좌는 소프트웨어 개발 단계 별로 SWAL concept에 따른 안전성 활동에 대하여 다루고 있으며, SWAL concept을 바탕으로 하는 개발 표준 및 가이드라인 (ED-12C, DO-178C, IEC 61508, ED-109A, ED-153) 등에 대하여 살펴본다. 이에 따라 해당 강좌는 마찬가지로 Eurocontrol에서 주최하는 Safety Assessment Methodology Course의 선행 참석을 강력하게 권장하고 있으며, ISO 12207, ED-109A, CMMI-Dev 등의 개발 표준 및 가이드라인 학습을 선행하도록 요구한다.

(4) 기타

Eurocontrol에서는 상기 강좌를 포함하여 항공 관제와 관련된 소프트웨어 안전성에 관련하여 다른 강좌도 개설하고 있다.[42)]

- Safety Assessment Methodology (SAF-SA1)
- Practical Safety Assessment (SAF-SA2)
- ATM Software Safety Assessment (SAF-SW)

3) Software System Safety Course

교육명	Software System Safety Course		
주최단체	IAASS(International Association for the Advancement of Space Safety)	기관분류	공공기관/인증기관
교육시간	3D	교육비용	$1,950
교육내용	안전성, PHA, FMECA, OSHA, FTA, ETA 등		

(1) 개요

Software System Safety Course는 IAASS (International Association for the Advancement of Space Safety)[43)]에서 주최하는 강좌로, 소프트웨어 안전성과 관련한 개발 프로세스 및 안전성 분석 기법에 강의이다. IAASS는 우주 분야의 안전성 발전을 목적으로 설립된 비영리 조직으로, 관련 산업계, 학계 및 정부조직에 속한 개별 인원들이 회원으로 참여하고 있다. IAASS는 이러한 목적에 따라 JSSE (The Journal of Space Safety Engineering) 및 각종 출판물 발간, 각종 강좌 및 워크샵 등을 주최하고 있으며 해당 강좌 또한 이러한 활동의 일환이다.

강좌는 소프트웨어 안전성에 대한 다양한 접근법과 분석 기법들에 대하여 소개하고

42) http://trainingzone.eurocontrol.int
43) http://iaass.space-safety.org

있다. 강좌는 PHA, FMECA, OSHA 등을 비롯한 여러 기법들에 대하여 다루고 있으며, FTA, ETA와 같은 기법에 대해서도 살펴본다. 강좌는 소프트웨어 개발 프로세스에 따른 안전성 관련 단계별 세부 활동 등에 대해서도 다루고 있다. Software System Safety Process (SwSS)를 설정하기 위한 위험 관리, 시스템 안전, 소프트웨어 개발에 대하여 소개하고 SwSS process 단계별 활동에 대해서도 논한다. 또한 SHCM과 LOR 개념 및 연관성에 대해서도 소개하고 있다. 강좌는 또한 예제에 대한 분석 기법을 적용하는 그룹 활동 등을 통하여 소개된 기법 및 관련 도구에 대한 이해 및 활용을 추구하는 것으로 생각된다.

(2) 교육내용

해당 안전한 소프트웨어를 개발하기 위한 개발 방법 설정 및 각 개발 단계별 활동에 대하여 논하며, 여러 안전성 분석 기법을 소개하고 예제에 적용해보는 활동을 수행하는 것으로 보인다. 그러나 해당 강좌에 대한 구체적인 커리큘럼은 공개되어 있지 않다.

(3) 기타

IAASS는 우주 분야의 다양한 안전성 관련 강좌, 워크샵 및 세미나를 주최하고 있으나, 소프트웨어의 안전성에 대한 강좌는 해당 강좌 이외에 확인할 수 없다.

4) Developing and Validating Software for the Medical Device Industry Course

교육명	Developing and Validating Software for the Medical Device Industry Course		
주최단체	AAMI(Association for the Advancement of Medical Instrumentation)	**기관분류**	공공기관/인증기관
교육시간	3D	**교육비용**	$2,435
교육내용	상업용 의료기기에 대한 표준 요구사항을 만족할 수 있는 소프트웨어 설계/검증		

(1) 개요

Developing and Validating Software for the Medical Device Industry Course는 미국 의료기기진흥협회 AAMI (Association for the Advancement of Medical Instrumentatio

n)[44][45]에서 주최하는 강좌이다. AMMI는 의료기기 분야의 기술 증진을 위해 만들어진 조직으로, 의료기기 및 기술에 대한 정보를 공유하고 표준을 개발하며, 의료기기 및 관련 기술사용 능력의 향상을 위해 각종 교육 및 인증 프로그램 또한 진행하고 있다. 해당 강좌는 3일 간으로 구성되어 있으며, 요금은 1인당 $2,435로 책정되어 있다. 단 AMMI 회원인 경우 $2,135, 정부기관 소속인 경우 $735로 책정되어 있다.

강좌는 상업용 의료기기에 대한 표준 요구사항을 만족할 수 있는 소프트웨어 설계 및 검증 계획 수행에 대한 내용을 담고 있으며, 임베디드 소프트웨어, 상업용 소프트웨어, 퀄리티 시스템 (non-device) 소프트웨어에 적용할 수 있는 각종 분석 기법과 도구들에 대한 예제 활동으로 구성되어 있다. 또한 예제를 가지고 추적성 분석 (traceability analysis), 요구분석 명세, 테스트 계획, 검증 계획 등의 검증 활동을 수행한다.

(2) 교육내용

<표 3-14> AAMI Developing and Validating SW for the Medical Device Industry 교육내용

- Intro to FDA and Software
- Verification and Validation Defined
- Intro to Lifecycles
- Risk Management
- Traceability and Configuration Management
- Device Concepts, Software Requirements
- Software Designs and Implementation Activities
- 62304 Update
- Software Testing
- Non-Device Validation
- Non-Device Software Lifecycle Planning
- Non-Device Intended Use and Requirements
- Non-Device Risk Management and CM
- Non-Device Validation Toolbox
- Non-Device Validation Toolbox - Part II
- Maintenance Validation Activities
- Streamlining Validation Processes

(3) 목표

- Design software validation plans that build confidence in the software and

44) http://www.aami.org
45) http://university.aami.org

comply with regulatory requirements for device, commercial off-the-shelf, and Quality System software

- Use risk management to focus validation activities to minimize risk
- Streamline elements of the Quality System for cost-efficient software development and validation
- Select appropriate lifecycle models and synchronize validation activities for all types of software
- Write unambiguous, testable requirements
- Integrate best development engineering practices to support validation efforts
- Organize test designs, test cases, and test procedures that effectively cover requirements being verified, and that provide opportunities for review and management of the process

(4) 기타

AAMI에서는 해당 강좌 외에도 의료기기 소프트웨어와 관련된 다른 강좌 및 온라인 강좌 (Webinar) 또한 제공하고 있다. 다음은 2016년 AAMI에서 주최하는 소프트웨어와 관련된 강좌이다.

- Design Control Requirements and Industry Practice
- Statistical Tools and Methods for a Quality System

4. 민간 인증기관

1) TÜV SÜD

교육명	TÜV SÜD Webinars		
주최단체	TÜV SÜD	기관분류	공공기관/인증기관
교육시간	미정	교육비용	미정
교육내용	SIL 인증, IEC61508, EN ISO 13849		

TÜV SÜD에서는 다양한 주제로 Webinars를 진행하고 있다[46]. TÜV SÜD는 시험, 인증, 검사, 교육 등 종합적인 기술 서비스를 제공하는 글로벌 기술 솔루션 기업으로 독일 뮌헨에 본사를 두고 있다. 1866년 1월 6일 독일 최초의 증기 보일러 검사 협회가 설립되었고 이 협회가 TÜV SÜD의 모태가 됐다.

46) http://www.tuv-sud-america.com/us-en/resource-center/training/webinars

TÜV SÜD의 모든 Webinars는 무료이고 TÜV SÜD 전문가가 강의를 진행한다. 최근에 진행될(2016.10.13 예정) webinar 에서는 “Introduction to Safety Integrity Level (SIL) Certification” 라는 타이틀로 SIL 인증과 IEC 61508과 EN ISO 13849에 대해서 강의를 진행할 예정이다. 강의 시간은 대략 1시간이고 사전 등록을 통해 신청 및 교육을 진행 할 수 있다. 다양한 교육 코스 및 자문 서비스를 지원하고 있지만, 특정 정규 교육 코스를 진행하고 있지는 않다.

TÜV SÜD는 소프트웨어 안전과 관련하여 다음과 같은 서비스를 제공하고 있다.

- Support of software development processes,
- Evaluation of the suitability of planned software architectures,
- Audits to determine the status quo,
- Introductory training courses and safety workshops,
- Analysis and derivation of safety requirements,
- Support in the creation of performance and requirements specifications,
- Checking of safety documentation and
- Software certifications.

이러한 서비스는 철도와 자동차, 항공기술과 같은 안전 관련 분야의 소프트웨어에 적용된다. 이러한 서비스를 통해 궁극적으로는 다음과 같은 표준 취득을 돕는다.

- IEC 61508, Part 3
- EN 50126, 50128, 50129
- ECSS-Q-80B
- DO-178B

2) DNV GL

교육명	DNV GL Webinars		
주최단체	DNV GL	**기관분류**	공공기관/인증기관
교육시간	미정	**교육비용**	미정
교육내용	IEC61508		

DNV GL 는 생명과 재산과 환경의 보호(To safeguard life, property and the

environment)를 목표로 1864 년 노르웨이에서 독립적인 재단법인으로 설립. 제품인증과 선급인증을 중심으로 진행하고 있는 회사이다. DNV GL의 모든 Webinars는 무료이고 사전 등록을 통해 신청 및 교육을 받을 수 있고 기존 강의는 비디오로 시청 가능하다[47]. 강좌 수는 다양하지 않은 편이지만 간단한 등록을 통해 과거 강좌의 비디오의 시청이 가능하다. 다양한 교육 코스 및 자문 서비스를 지원하고 있지만, 특정 정규 교육 코스를 진행하고 있지는 않다.

DVN GL은 화학연료나 에너지, 해양과 같은 분야의 소프트웨어 안전과 관련해서 다양한 서비스를 제공하고 있다. 소프트웨어 개발 및 검증, 테스팅에 대한 컨설팅 및 각종 도구 지원도 한다. 또한 DNV GL 자체적으로 개발한 표준이나 가이드라인을 활용한 인증도 실시하고 있다.

- Energy : IEC 61508 Verification
- Maritime : DVN GL Rules
- Oil & Gas : DNVGL-OS-D203 ISDS and IEC 61508/11

3) 기타 인증 기관 Webinar

UL[48], FCC[49], BSIAmerica[50], DoD[51]

47) https://www.dnvgl.com/software/video-webinar/webinar.html
48) http://industries.ul.com/tag/webinar
49) https://www.fcc.gov/news-events/events
50) http://www.bsigroup.com/en-US/Our-services/Events/On-demand-webinars/
51) http://www.dcoe.mil/Training/Monthly_Webinars/Archive.aspx

제3절 조사결과 분석

해외 교육 현황 조사결과 다음과 같은 특징들을 살펴볼 수 있었다.

- 소프트웨어 안전은 시스템 안전과 별개로 생각할 수 없기 때문에, 실제 산업에서는 산업 도메인별로 특화된 소프트웨어 안전 교육이 시스템 수준에서부터 하드웨어, 소프트웨어까지 포괄적으로 이루어지고 있음을 파악할 수 있다.
- 도메인별로 시스템의 안전, 요구 및 규제의 상황에 따라 교육의 활성도가 다름을 알 수 있다. 특이한 것은, 도메인이 다양하다 보니 도메인의 공통적 특성에 초점을 맞춘 IEC61508 중심의 교육들이 존재하는 것을 파악할 수 있었다.
- 기초적인 이론 중심의 교육들은 대학 중심으로 정규과정 혹은 비정규과정 형태로 진행되고 있다. 도메인 부분에서는 항공, 기술면에서는 분석기법 등 대학별로 전공 교수의 관심사에 따라 특화된 기술영역에 중점을 두는 추세를 파악할 수 있다. 또한 미국과 유럽(영국, 독일 등) 중심의 양대 연구 활동의 축이 형성되어 있었다.
- 해외의 소프트웨어 안전 교육들은 대부분이 소프트웨어 안전 관련 전문 회사 및 기관들에 의하여 소프트웨어 안전의 전반적이 내용이 일반 수강생을 위해 공개되어 제공되고 있었다. 도메인별 표준에 관련된 교육(항공, 의료 등)들은 인증기관이나 관련 표준기관에 의해 제공됨을 알 수 있다.

<표 3-15> 해외현황 조사결과

구분	주최단체	구분	특징적 내용	시간
민간	HCRQ	일반/분석기법	FMEA, FTA	4D
	Engineering Safety Consultants	표준	IEC61508	1~3D
	PILZ	도구	EN ISO 1384901, EN62061	-
	CRITICAL Software	표준	DO-178C, DO-254, DO-278A, ISO26262, EN50126/8/9	-
	Edif Group	일반	PLDs, SILS, SOUP	2D
	Engineering Education Australia	일반/관리	HARA, Safeware	5D
대학	University of Southern California	분석기법	분석, 안전 관리방법	4D
	Carnegie Mellon University	일반	Safety/Security, Hazard	2D
	The University of Queensland	일반/분석기법	HAZOP, CHAZOP, FFA, FTA, ETA, FMEA, FMECA, GSN	S
	University of York	일반/SW공학	안전 요구사항 분석	S
	Australian National University	일반/SW공학	안전 요구사항 분석, 기법	S
	MIT Aeronautics & Astronautics Dept.	분석기법/항공	위험 분석, 안전 설계	5D
	Technische Universität München	분석기법	FMEA, FTA, STPA	S
	Johns Hopkins University	일반/SW공학	HARA, Safeware	S
	Embry-Riddle Aeronautical University	분석기법/항공	HARA, FMEA, Testing, VV	3D
	Åbo Akademi University	분석기법/절차	SE, FMEA, FTA, HAZOP, SIL	S
	Jönköping University	일반/분석기법	Safety/Security	S
	Massachusetts Institute of Technology	일반/분석기법	분석, PHA, FTA, RA, STPA	S
	UCLA Extension	일반/임베디드	Embedded-safety 디자인	3D
	Detroit University	일반/SW공학	Safety 소프트웨어공학	S
공공	IET	표준	IEC61508	-
	Eurocontrol	항공 표준	ATM	5D
	IAASS	일반/분석기법	PHA, FMECA, OSHA, FTA, ETA	3D
	AAMI	의료 표준	분석, IEC62304	3D
인증	TÜV SÜD	표준/분석기법	SIL, IEC61508, EN ISO 13849	-
	DNV GL	의료 표준	IEC61508	-

(D:일, S:학기, - :미정)

제4장 국내교육 현황조사

제1절 개요

본 장에서는 국내에서 현재 진행되고 있는 소프트웨어 안전 관련 교육 현황들에 대한 조사 결과를 설명한다. 국내에서는 주로 자동차 분야에서 ISO26262 관련 소프트웨어 안전 교육이 활발히 진행되고 있으며, 그 외에 산업군에서는 국소적으로 진행되고 있는 것으로 파악된다. 대학에서는 소프트웨어 안전 관련 전공 교수진이 있는 대학에서 대학원 과정의 교과과정으로 부분적으로 개설되고 있으며, 또한 소프트웨어 안전 관련 ITRC 지정 대학에서도 수행되고 있다.

제2절 교육과목

1. 민간 전문기관

1) Training Courses for Software Engineering

교육명	Training Courses for Software Engineering		
주최단체	SOLUTIONLINK	기관분류	민간 전문기관
교육시간	1D / 2D	교육비용	미정
교육내용	소프트웨어 공학, 요구공학, 소프트웨어 안전 설계, FMEA, 기능안전(ISO26262) 테스팅		

(1) 개요

Training courses for software engineering은 solution link[52]에서 주최하는 소프트웨어 공학에 대한 전반적인 내용을 담고 있는 강좌이다.

강좌는 크게 6 가지의 주제로 이루어져 있으며, 소프트웨어 공학 개요, 요구사항 분석, 설계, 개발, V&V(verification & validation), 유지보수 순으로 구성되어 소프트웨어 생명 주기 별로 전반적인 사항들을 다룬다. 각 파트별로 주제에 맞는 다양한 강좌들을 구성하고 있으며, 소프트웨어의 안전과 관련해서 요구사항 파트에서 기능안전 요구사

52) http://sol-link.com/neo/kr/academy/training_01.php

항 명세에 대한 부분을, 설계 파트에서 안전 설계 및 분석에 대한 강좌와 개발 파트에서 안전 메커니즘 구현에 대한 강의를 진행하며 V&V 파트에서 기능안전 테스팅에 대한 강좌를 진행 한다. 소프트웨어 설계 파트에서 위해도 분석을 위해 SW FMEA 에 대한 강좌를 개설하고 있는 특징이 있다.

(2) 교육내용

<표 4-1> SOLUTIONLINK Training Courses for Software Engineering 교육내용

Software Engineering	SWE-00. 소프트웨어 공학 개요
Software Requirements	SWR-01. 요구공학
	SWR-02. 기능안전 요구사양 명세
Software Design	SWD-01. 객체지향 분석/설계
	SWD-02. 구조적 분석/설계
	SWD-03. 소프트웨어 디자인 패턴
	SWD-04. 소프트웨어 안전 설계
	SWD-05. SW FMEA
	SWD-06. 소프트웨어 안전설계 및 분석
Software Construction	SWC-01. 소프트웨어 안전 메커니즘 구현
	SWC-02. 자동통합 환경 구축
Software Verification & Validation	SWV-01. 소프트웨어 V&V
	SWV-02. 도구 기반 단위 테스트
	SWV-03. 기능안전 테스팅
Software Maintenance	SWM-01. 소프트웨어 역공학
	SWM-02. C/C++ 코드 리팩토링

(3) 소프트웨어 안전 표준/가이드라인 강의 세부 내용

각 파트별 소프트웨어 안전과 관련된 강좌들의 상세한 강의 내용은 다음과 같다.

<표 4-2> SOLUTIONLINK 기능 안전 요구사양 명세

교육 시간	1일 (7시간)
개요	기능안전 요구사항 기술을 위한 준정형 명세 방법을 습득, 실습을 통해 실무 적용 능력 배양
목표	기능안전 요구사항 개발에 대한 이해 준정형 요구사항 명세 방법 습득 실습을 통한 요구사항 명세 및 검증 능력 배양
교육 내용	안전 요구사항 명세 및 관리 개요

	· 안전 요구사항의 단계별(기술/하드웨어/소프트웨어) 구분 · 안전 요구사항 도출 절차 및 단계별 도출 방법 · 안전 요구사항 단계별 명세 요소 · 안전 요구사항 추적 관리 안전 요구사항 명세 패턴 · 안전 요구사항 패턴 적용 전략 및 패턴 구성 요소 · 기술안전, 하드웨어, 소프트웨어 요구사항 패턴 안전 요구사항 명세 예시 , 실습

<표 4-3> SOLUTIONLINK SW 안전설계

교육 시간	2일 (14시간)
개요	ISO 26262를 준수하는 SW 아키텍처 및 상세 설계를 예제를 통해서 습득, SW에 대한 주요 안전 메커니즘 이해
목표	SW 개발 수준에서 요구되는 기능안전 요구사항 이해 기능안전 요구사항을 만족하는 임베디드 소프트웨어 설계 방법 습득 및 실무 적용 능력 배양
교육 내용	SW 설계 개요 · SW 개발을 위한 기능안전 요구사항, SW 설계 방안 SW 아키텍처 설계 절차, 기법 SW 상세 설계 원칙, 문서 SW 안전설계 메커니즘 : 센서, 엑츄에이터, 제어기, 메모리, 통신 인터페이스에 적용되는 기능안전 컨셉 SW 설계 예제 : E-GAS 시스템 예제

<표 4-4> SOLUTIONLINK FMEA

교육 시간	2일 (14시간)
개요	SW 수준에서 기능안전 분석 기법인 SW FMEA 수행 방법을 습득, 실습을 통해 실무 적용 역량 배양
목표	SW 개발 수준에서 요구되는 기능안전 요구사항 이해 기능안전 요구사항을 만족하는 임베디드 소프트웨어 설계 방법 습득 및 실무 적용 능력 배양
교육 내용	SW Failure Mode · 오류 유형, SW 수행형태에 따른 오류 유형 해석 · SW Failure Mode 정의 SW FMEA 수행 방안

	· VDA 기반의 SW FMEA 절차 · 구조, 기능, 경함, 위험 분석 · 개선 최적화 SW FMEA 실습

<표 4-5> SOLUTIONLINK 안전 매커니즘 구현

교육 시간	2일 (14시간)
개요	SW 안전 메커니즘 설계의 개요 숙지, 주요 SW 안전 메커니즘 구현 상세 기법을 습득
목표	SW 안전 메커니즘 설계의 이해 사례를 통한 SW 안전 메커니즘 구현 상세 기법 습득
교육 내용	SW 안전 아키텍처 설계 · SW 아키텍처 설계의 안전 체계를 설계하는 과정 SW 안전 메커니즘 개요 · 전통적인 전장 SW의 아키텍처 · 주요한 SW 안전 메커니즘 목록 SW 안전 메커니즘 구현 상세 · 센서의 모니터링 결과에 대해 정합성 확인을 위한 안전 메커니즘의 코드 사례 · 액추에이터의 모니터링 결과에 대해 정합성 확인을 위한 안전 메커니즘의 코드 사례 · 감시기(watchdog), 프로그램 흐름 모니터링 등 기능의 수행 전반을 모니터링하는 안전 메커니즘의 코드 사례 · 액추에이터의 모니터링 결과에 대해 정합성 확인을 위한 안전 메커니즘의 코드 사례 SW 구현 실습 환경 구성, SW 안전 메커니즘 구현 실습

<표 4-6> SOLUTIONLINK 기능 안전 테스팅

교육 시간	1일 (7시간)
개요	ISO26262에서 정의하는 V&V에 대한 수행 요건 이해, 단계별 테스트 기법 적용 방안을 습득
목표	ISO26262의 V&V 수행 요건, 테스팅 이해 단계별 테스트 기법 적용 방안 습득 안전 메커니즘 검증 방안 습득

교육 내용	ISO26262의 V&V 요구사항 테스팅 기법 소개 단계별 테스트 기법 적용 방안 · 하드웨어, 소프트웨어 단위/통합 테스트, 시스템 통합 테스트 안전 메커니즘 검증

2) NAVITHES ISO 26262 교육

교육명	NAVITHES ISO 26262 교육		
주최단체	NAVITHES	**기관분류**	민간 전문기관
교육시간	1D / 3D	**교육비용**	미정
교육내용	ISO26262		

(1) 개요

해당 과정은 NAVITHES[53]에서 진행하는 ISO 26262 교육으로 1 ~ 3일 간으로 진행되는 교육이다. NAVITHES는 2011년 설립된 차량 전장부품 및 의료기기 전문 개발 업체로, ISO 26262 실무지도와 AUTOSAR 개발 및 교육을 서비스하고 있다. 주로 ISO 26262에 대한 실무 지도 및 컨설팅, 업무지원 등의 사업을 진행하고 있다.

ISO 26262 교육은 1일 코스와 3일 코스 2 종류의 ISO 26262 교육을 제공하고 있고, 1일 코스는 ISO 26262의 기본적인 개념과 각 장의 설명 및 개요에 대한 내용이고, 3일 코스의 교육은 실무와 관련된 내용이 포함되어 있으며, FMEA와 FTA 에 대한 내용을 교육 내용에 포함하고 있다.

(2) 교육내용

NAVITHES 에서 진행하는 ISO 26262 교육은 1일 과정과 3일 과정 두 종류로 구분되어 있고, 각 과정에 대한 상세한 커리큘럼은 다음 표와 같다. 1일 과정의 경우 ISO 26262 기능안전성 표준의 기본적인 내용 및 각 파트별 설명으로 구성 되어 있고, 3일 과정엔 추가적으로 시스템/소프트웨어 개발 단계별 설명 및 테스팅에 관련된 내용을 포함하고 있다.

<표 4-7> NAVITHES ISO 26262 교육내용

내용 (1일 과정)	비고
◆기능안전성 ISO 26262 의 개념 ◆ISO 26262 정식판 6장의 설명	일본 및 유럽 사례 포함.

53) https://sites.google.com/a/navithes.com/nt/gyoyug/iso26262gyoyug

-기본적인 개념	-SW 안전 요구 사항의 정의 -SW 설계 -SW 테스트 -SW의 Verification과 Validation	
◆ISO 26262 정식판 1장, 2장, 8장의 설명 -용어 -OEM과 Supplier의 관계 -DIA계약서 -기능안전 계획, 기능안전 요구사항	◆ISO 26262 정식판 3장,4장의 설명 -Item, Safety case -Hazard 분석 및 리스크 평가 -기능안전 개념 -기술적 안전 개념 -시스템 설계의 개발 -시스템 설계의 테스트 -FMEA · FTA set	
◆ISO 26262 정식판 5장, 9장의 설명 -HW 안전 요구사항의 정의 -HW 설계 -HW 테스트 -Random HW 고장의 평가 -FMEA와 FTA Set -ASIL 평가방법론 -ASIL 분석의 사례	◆ISO 26262 정식판 7장 `10장의 설명 -생산과 운영 단계 -ISO 26262 10장의 가이드라인	

일정	교육 내용	비고
1일 기본 개념	기본 개념 및 구상 단계	개념 설명
	-유럽, 일본, 국내 자동차업계의 기능안전	
	-기능안전 ISO26262의 전반적인 개요	
	-OEM과 Supplier관계	
	-DIA 계약서	
	- BMS의 ASIL 할당 및 분할	
	-용어 설명	
	-아이템 정의	
	-안전생명주기 준비 및 시작	

		-위험 분석 및 리스크 평가 -기능안전 구상	
2일	ISO 26262 실무 (Part 2 ~ 9)	**시스템 개발 단계** -시스템 개발 준비 및 시작 -기술적인 안전 요구 정의 -시스템 설계 (S/W, H/W 개발) -아이템의 통합 테스트 -안전 타당성 확인 -기능안전 평가 - FMEA, FTA -생산 릴리즈 **소프트웨어 개발 단계 (1/2)** -소프트웨어 개발 준비 및 시작 -소프트웨어 안전 요구 정의 -소프트웨어 아키텍처 설계	ISO 26262 **실습**
3일		**소프트웨어개발 단계 (2/2)** -소프트웨어 유닛 설계 및 구현 -소프트웨어 유닛 테스트 -소프트웨어 통합 테스트 -소프트웨어 안전 요구 검증 **하드웨어 개발 단계** -하드웨어 개발 준비 및 시작 -하드웨어 안전 요구 정의 -하드웨어 설계 -임의의 고장의 평가 - FMEA, FTA -하드웨어 통합 테스트	

(3) 기타

ISO 26262이외에도 의료기기에 관련된 IEC 62304 실무자 교육 과정을 운영하고 있다. 주로 표준 개요 및 의료용 소프트웨어 위험(RISK, 리스크) 관리에 대한 교육을 진행하고 있다.

<표 4-8> IEC 62304 실무자 교육 과정

일정		교육 내용	비고
1일	ISO 14971	**기본 개념**	개념 및 실습
	리스크 관리	리스크 관리 평가방법	
	교육 과정	리스크 분석	
		리스크 평가	
		리스크 컨트롤	
		잔여 리스크 평가 및 리스크 시각화 방법	
		리스크 평가로 본 각국의 리콜 사례	
2일		**실습 과정**	
		리스크 계획서 설명과 해설	
		리스크 계획서 작성 실습	
		리스크 보고서 설명과 해설	
		리스크 보고서 작성 실습	
		FMEA 및 FTA 설명과 적용	
3일	IEC 62304	**기본 개념 및 실습 과정**	개념 및 실습
	교육 과정	IEC 62304 개요 및 개발 경위와 규제동향	
		S/W 개발 및 유지보수 프로세스	
		S/W 리스크 관리, 구성관리, 문제해결 프로세스	
		의료용 소프트웨어 심사예제	

3) 안전성 확보를 위한 철도 SW 개발 프로세스 소개 (EN50128)

교육명	안전성 확보를 위한 철도 SW 개발 프로세스 소개 (EN50128)		
주최단체	모아소프트	**기관분류**	민간 전문기관
교육시간	2D	**교육비용**	미정
교육내용	EN50128		

(1) 개요

EN50128 교육은 모아소프트[54]에서 진행하는 2일 과정의 교육이다. 모아소프트는 1998년 신뢰성 분석기술을 시작으로 설립된 회사로, 항공, 철도, 자동차, 해양 등의 분야에 대해 개발 프로세스 지원 등의 소프트웨어 솔루션을 제공 하고 있으며, 이외에도 소프트웨어 신뢰성 시험, 종합 군수지원 솔루션 등 여러 솔루션과 DO-178C, RAMS 분

54) http://www.moasoftware.co.kr/company/greeting.asp

석, 소프트웨어 검증 등의 컨설팅을 하고 있다. 원자력, 군수 사업, 자동차, 철도 등의 분야에서 다양한 기술 지원 및 신뢰성 시험 등의 프로젝트를 수행하였다.

이외에도 모아소프트에서는 솔루션제공, 컨설팅 등의 업무 외에도 시스템의 신뢰도, 테스팅, SW 개발 프로세스 등에 관련된 다양한 교육 세미나를 제공하고 있다. 철도 SW 개발 프로세스 교육은 이 중의 하나로 일반적인 철도분야의 SW 개발 프로세스에 대한 설명과 각 프로세스 별 산출물에 대해 설명하고, EN50127에 대한 내용을 다룬다. 또한 추가적으로 신뢰성 시험에 대한 소개 및 사례에 대해 소개하고 있다. 해당 교육에서는 다음과 같은 내용에 대해 소개 및 교육을 목표로 한다.

- 타 산업분야의 소프트웨어 프로세스를 기반으로 철도분야의 SW개발 프로세스를 이해한다.
- 소프트웨어 개발 단계별 산출물을 이해한다.
- 철도 분야의 SW 안전성을 위한 SW개발 프로세스를 이해한다.
- EN50128에서 요구하는 Objective와 Requirement를 이해한다.
- 안전성 확보를 위한 신뢰성 시험 프로세스를 이해한다.

(2) 교육내용

<표 4-9> 안전성 확보를 위한 철도 SW 개발 프로세스 소개 (EN50128) 교육 내용

일정	내용	세부내용
1 일차	EN50128 SW개발 프로세스 소개	개요 및 교육생 인사
		EN50128 개요 및 소개 (산출물 소개)
		점심 식사
		EN50128 Objectives & Requirements 소개 -개발 프로세스 산출물 Review
2 일차	안전성 확보를 위한 신뢰성 시험 프로세스 소개	EN50128 개발 프로세스 Review
		신뢰성 시험 프로세스 소개
		점심 식사
		개발 프로세스 산출물 사례 소개 신뢰성 시험 사례 소개 Q&A

4) 감항인증을 위한 항공용 SW 개발 프로세스(RTCA DO-178C)

교육명	감항인증을 위한 항공용 SW 개발 프로세스 (RTCA DO-178C)		
주최단체	모아소프트	기관분류	민간 전문기관
교육시간	3D	교육비용	미정
교육내용	RTCA DO-178C		

(1) 개요

DO-178C를 기반으로 한 SW 개발 프로세스에 대한 교육으로 모아소프트에서 진행한다. 해당 교육의 개요는 다음과 같다.

"DO-178C는 DO-178C는 항공기 소프트웨어 안전성을 위한 국제적인 감항인증 규격으로, 미국연방항공국(FAA)에서 항공용 제품에 대한 승인 시 SW 부분에 적용하고 있습니다. 이에 따라 SW 안전성 레벨(Level A, B, C, D)에 따른 DO-178C의 Objectives, Activities, Outputs의 이해를 돕고, 시스템 요구사항으로부터 할당된 SW 요구사항 구현의 완전성, 정확성, 추적성 보증을 위한 문서화 방안을 제시합니다. 이와 더불어 DO-331(모델기반 개발 규격)과 DO-278A(항공 지상 장비 개발 규격)을 함께 교육합니다."

해당 교육의 목표는 다음과 같다.

- RTCA DO-178C 개요
- SW 안전성 레벨에 따른 DO-178C의 Objectives, Activities, Outputs 이해
- 시스템 요구사항으로부터 할당된 SW 요구사항 구현의 완전성, 정확성, 추적성 보증을 위한 문서화 방안 이해
- DO-331 Model Based Development(Supplements to DO-178C) 및 DO-278A 소개

(2) 교육내용

<표 4-10> 감항인증을 위한 항공용 SW 개발 프로세스 (RTCA DO-178C) 교육 내용

일정	교육 내용
1일차	1. Course Introduction
	2. RTCA DO-178C Overview
	3. System Aspects Relating to Software Development
	4. Software Life Cycle
	5. Software Planning Process
	6. Software Development Process

2일차	7. Software Verification Process - Software Reviews and Analyses - Software Testing 8. Software Configuration Management Process 9. Software Quality Assurance 10. Certification Liaison Process and Certification Overview Process
3일차	11. Software Life Cycle Data 12. Additional Considerations 13. Supplements to DO-178C - DO-331 Model Based Development 14. DO-278A: Communication, Navigation, Surveillance and Air Traffic Management System

5) 자동차 기능안전 (ISO 26262) 전문가 과정 (FSCP) 교육

교육명	자동차 기능안전 (ISO26262) 전문가 과정 (FSCP) 교육		
주최단체	TUV SUD Korea	기관분류	민간 전문기관
교육시간	3D	교육비용	385만원
교육내용	ISO26262		

(1) 개요

TUV SUD Korea[55]에서 진행하는 자동차 기능안전 전문가 과정 교육이다. TUV SUD는 1994년 한국 법인을 설립 하였고, 제품 시험 인증 및 경영 시스템 인증을 비롯하여, 산업 검사 및 관련 보고서, 교육 및 세미나, 전문 지식 및 정보 제공 서비스 등을 제공하는 회사로 각각의 서비스들은 전기, 전자, 통신, 의료기기, 자동차, 철도, 승강기, 놀이공원 및 놀이기구, 원자력 발전소 등의 다양한 분야에 제공하고 있다. 3일 강의의 비용은 1인 385 만원 이며, 6년 이상의 산업의 업무 경험 및 2건 이상의 기능안전 프로젝트 경험 보유자만 참여 할 수 있다.

강좌는 3일로 구성되어 있으며, 기능 안전 관리 강의부터 시작하여 안전 관련 하드웨어 개발, 안전 관련 소프트웨어 개발에 관련된 내용을 중점적으로 다룬다. 또한 안전 생명주기별 FMEA 및 FTA 분석에 대한 내용도 교육 내용에 포함되어 있고 주된

55) http://www.tuv-sud.kr/kr-kr/about-tuev-sued/tuev-sued-in-korea/about-us-in-korea

내용은 ISO 26262의 요구사항에 맞추어 진행 한다. 특이 사항으로는 교육 마지막에 시험을 통해 TUV SUD 인증서 (Certificate of “Functional Safety Professional regarding ISO 26262”)를 수여 하고 있다.

(2) 교육내용

<표 4-11> 자동차 기능안전 (ISO26262) 전문가 과정 (FSCP) 교육 내용

일정	교육내용
1일차	Functional Safety Management - Organizational structures, Task profiles, Safety lifecycle From concept phase to system design - Hazards Analysis with Risk Assessment - Derivation of a Functional Safety Concept - Derivation of the Technical Safety Requirements and Creation of the Technical Safety Concept Safety relevant hardware development Part I - Hardware design, Architectural metrics, Hardware testing
2일차	Safety related software development, Part I - Software safety requirements, Software development - Software modification, Software validation FMEA and FTA analysis for the Safety lifecycle - Decomposition, ASIL Tailoring, Analytical methods
3일차	Introduction of Examination Examination

6) STA 테스팅

교육명	STA 테스팅		
주최단체	STA 테스팅컨설팅	기관분류	민간 전문기관
교육시간	1~3D	교육비용	미정
교육내용	SW Testing, SW 기능안전성 보증, ISO26262		

(1) 개요

STA 테스팅컨설팅[56]은 2002년 STEN (Software Test Engineers Network) 커뮤니티

56) http://www.sta.co.kr/

설립을 시작으로 시작 되었으며, 인천공항, 두산중공업, 현대오토에버 등 다양한 기업들을 대상으로 SW 품질 관련 컨설팅을 진행하고 있다. 주로 SW 테스팅 관련 교육들을 진행하고 있으며, 테스팅 교육은 foundation, test manager, test analyst 과정으로 나누어 진행하고 있다[57].

Foundation 과정에서는 소프트웨어 테스팅의 전반적인 개념에 대한 설명과, 테스트 케이스 작성, 테스팅 프로세스 등에 대한 기초적인 내용을 강의하고, test manager 과정에서는 테스트 조직관리, 규모 산정, 도구 선정, 전략 수립 등 테스팅을 수행하는 매니저의 입장의 강의를 진행 한다. 추가적으로 특화교육으로 안전필수 분야 SW 기능안전성 보증, ISO 26262 기능안전성 과정 교육을 진행 한다.

(2) 교육내용

<표 4-12> STA 테스팅 교육내용

교육 과정 명	일수	시간
TMMi 중심의 테스트 프로세스 구축 및 개선교육	2	14
탐색적 테스팅(Exploratory Testing) 교육	2	14
테스터도 알아야 할 화이트박스 테스트 & SW 테스트 자동화 교육	2	14
요구사항 관리 및 요구공학 기초과정	3	21
스마트폰 어플리케이션 테스팅 교육(안드로이드 중심)	2	14
소프트웨어 결함 찾는 법	1	4
임베디드 SW 테스팅 실무교육	3	21
안전필수 분야 SW 기능 안전성 보증(기초과정)	1	7
ISO FDIS 26262 기능안전성(Functional Safety)과정	2	14
Practical Test Management - 리스크 기반 접근 전략 (ISO 29119 중심)	3	22
개발 환경에서의 단위 테스트 교육	3	21
실무 중심의 테스트 기초 과정(KSTQB Basic Level 자격 기반)	1	7
Automotive SW Testing	3	21
애자일 방법론과 애자일 테스팅 교육	2.5	20
영향도표(Influence Diagrams)를 활용한 SW 테스팅 이해	1	7
테스팅 설계 교육 (Testing Techniques in Practice)	2	14
Jmeter 성능테스트 이론과 실무	2	14
탐색적 테스팅(Exploratory Testing) 교육(1일)	1	8
SW테스트 설계 향상(명세 기반 기법) 교육	2	14

57) www.sten.or.kr/bbs/board.php?bo_table=training&sca=FL%2FAL+%B1%B3%C0%B0&type=1

(3) 소프트웨어 안전/기능안전성 교육 내용

<표 4-13> STA 안전 필수분야 SW기능 안전성 보증(기초 과정)

교육 시간	1일 (7시간)
개요	안전필수 분야 SW에 대한 국제 표준 보증 요구사항 위험분석 기법을 학습 안전 수명주기 및 SIL 요구 기법 및 수단 이해 IEC 61508(국제 기능 안전 표준) 이해 테스트 설계 기법 이론 학습 및 실습(MC/DC) SW 검증 사례 발표를 통해 SW 기능 안전 학습
목표	안전필수 분야 전체 프로세스를 이해 SW 안전성을 보증하는 절차 및 방법 이해 안전관련 소프트웨어와 IEC 61508-3을 이해 단계별 요구사항에 대한 개략적 이해를 통해 효율적인 실무 적용 가능 실무적 기능 안전성 평가 및 보증 사례 발표를 통한 실무 경험 공유 가능
교육 내용	기능 안전성 개요 위험 분석 이론 및 실습 SIL 결정 방법 안전 SW 개발 프로세스 - SW 안전 요구사항 명세, SW 안전 검증 계획 - SW 설계 및 개발, HW + SW 통합, SW 운영 및 변경 - SW 안전 검증(Validaion), SW 확인(Verification) - 형상 관리 - 기능 안정성 평가

<표 4-14> ISO FDIS 26262 기능 안전성 (Functional Safety)과정

교육 시간	2일 (14시간)
개요	IEC 61508, ISO FDIS 26262를 통한 기능안전성 소개 ISO FDIS 26262 표준 논리 및 구조 이해 ASIL 및 기능안전성 생명주기 각 단계에 대한 이해
목표	자동차 분야의 ISO 26262 표준의 명확한 이해 기능안전성 관리 업무에 필요한 요구사항 이해 ASIL targeting 실습을 통한 ASIL 개념 정립 ASIL 확인검토/심사/평가/인증에 대한 이해 및 효율적인 실무적용 방안 습득

교육 내용	1일차	기능안전성 표준 및 범위, 관리체계 개념 구상 및 리스크 평가+실습 형상관리, 변경관리, 문서관리, 소프트웨어 도구 검증 시스템 설계
	2일차	하드웨어 개발, 소프트웨어 개발, 생산, 운영, 지원 ASIL-oriented analysis

7) SPID Academy 교육

교육명	SPID Academy 교육		
주최단체	SPID	기관분류	민간 전문기관
교육시간	1~3D	교육비용	미정
교육내용	기능안전(ISO26262), CMMI, SW공학, 품질기준 등 / FTA 방법론, FTA++ 솔루션 소개		

(1) 개요

SPID에서는 기능안전, CMMI, SW 공학, 품질기준 등에 대한 교육을 위해 ACADEMY를 운영하고 있다. 각 교육별로 일정과 교육비가 책정되어 운영되고 있다. 교육의 내용으로는 표준 소개 및 이론부터, 기능안전 실무자 교육과 같이 적용을 위한 교육도 제공하고 있다. 특히 기능안전을 위해 ISO 26262 기능안전 실무자 교육, 기능안전 구현을 위한 시스템 모델링과 모델링 언어, ISO 26262 Professional Engineering 자격 인증 과정이 있으며, 신뢰성 분석을 위한 FTA 방법론 및 FTA++ 솔루션 소개 강좌도 존재한다.

(2) 교육내용

<표 4-15> SPID Academy 교육 내용

교육과정	기간(일)	시간	교육비
ISO 26262 기능안전 실무자 교육	3	21	80
하드웨어 부품 고장률 기반의 FMEDA (Failure Modes Effects and Diagnostic Analysis) 실무과정	2	14	70
VDA/AIAG 표준기반, 공정상 발생 오류 가능성 최소화를 위한 PFMEA 이론 및 SW(IQ-FMEA RM PRO) 활용 실습	2	14	55
기능안전 구현을 위한 시스템 모델링과 모델링 언어	2	14	55
신뢰성 분석을 위한 FTA(Fault Tree Analysis) 실무 방법론 및 Isograph RWB Fault Tree+ 활용 실습	2	14	55
ISO 26262 Professional Engineering 자격 인증과정	5	31	270
Automotive SPICE® Model and Provisional Assessor Training	5	35	미정
글로벌 기업의 사례를 통해 본 소프트웨어테스팅 이론과 실제	3	21	55

(3) 소프트웨어 안전/기능안전성 교육 내용

<표 4-16> SPID 기능 안전 구현을 위한 시스템 모델링과 모델링 언어

교육 시간	2일 (14시간)	
개요	System engineering 기반의 시스템 설계 기법을 기초로하여, 시스템 설계 영역에 대해 모델기반 접근을 통한 설계의 이해를 교육한다.	
목표	모델기반 접근을 통한 시스템 설계 수행의 이해 SysML을 활용법 교육	
교육 내용	1일차	ISO 26262에서의 시스템 시스템, 시스템 공학 이해 모델기반 시스템 설계 시스템 아키텍처, 미국 설계표준과 설계 사양서 SysML 기반 설계 프로세스, SysML 모델링 언어 개요
	2일차	SysML을 통한 요구사항 분석 및 생성, 관리, 구조 분석

<표 4-17> SPID ISO 26262 Professional Engineering 자격 인증 과정

교육 시간	5일 (31시간)	
개요	TUV Nord System의 자동차 분야의 기능안전전문가로서의 자격 검증(FSCAE)을 위한 교육 과정	
목표	자동차 기능안전 표준에 대한 이해	
교육 내용	1일차	Functional Safety Management Training
	2일차	System and Hardware Level Implementation
	3일차	System and Hardware Level Implementation
	4일차	Software Level Implementation
	5일차	Qualifying Exam for FSCAE

8) SGS Academy 교육

교육명	SGS Academy 교육		
주최단체	SGS	**기관분류**	민간 전문기관
교육시간	5D	**교육비용**	미정
교육내용	ISO26262, IEC61508		

(1) 개요

SGS 는 검사, 시험, 검정 및 인증서비스 분야의 선도 기업으로서 농산물, 광물, 소비자 상품, 식품, 환경, 오일 및 가스, 화학물질, 제조업 및 산업기계, 건설부문까지 폭

넓은 분야의 서비스를 제공하고 있다. SGS academy 에서는 SGS가 제공하는 분야에 대해서 각 회사 별로 전문적인 개발 교육 및 맞춤식 교육 솔루션을 제공한다.

(2) 교육내용

SGS에서 제공하는 교육 커리큘럼은 다음과 같이 다양한 분류별, 주제별로 여러 교육 솔루션을 제공하고 있다.

<표 4-18> SGS Academy 교육 내용

분류	주제	비고
산업별 교육	항공우주, 농업, 자동차, 화학물질, 소비자물품 에너지관리, 파이낸스, 정보 기술, 오일 및 가스	각 분류 - 주제별 다양한 교육 프로그램 운영
재료 시험	비파괴검사 교육 기능안전	
환경	기후 변화, 에너지 관리, 환경 관리 시스템	
리더십 및 매니지먼트	개발 관리	
매니지먼트 시스템 및 표준	비즈니스 연속성 관리 에너지 관리, 환경 관리 시스템 식품, 정보 기술, 품질 경영 시스템	
프로세스 개선	기능 안전 린과 6시그마	
위험 및 보안 관리	위험 관리	
공급망 및 제조	공급망 및 제조	
지속 가능성	사회 책임 지속 가능한 이벤트 경영	

(3) 소프트웨어 안전/기능안전성 교육

<표 4-19> SGS Academy IEC 61508교육

개요	SGS의 IEC 61508 기능 안전 교육에서는 안전 시스템의 산업용 개발 과정에서 이러한 표준을 구현하는 방법에 대해 학습합니다.
교육 내용	IEC 61508 위임 및 해당 요구 사항 소개 실제 사례 조사 및 연구를 통해 이러한 요구 사항을 대상 중심으로 구현하는 방법 안내 다양한 교육 모듈 제공

<표 4-19> SGS Academy ISO 26262 자동차 기능 안전 훈련

교육 시간	5일
개요	ISO 26262는 전체 제품 수명 주기에 영향을 미치는 신규 자동차 적용 기준입니다. 그러나 기준을 효과적으로 시행하는 것은 복잡할 수 있습니다. 그러므로 직원들이 제대로 된 훈련을 받도록 해야 합니다. 기능 안전 기관인 당사의 ISO 26262 자동차 기능 안전 훈련을 받으면 귀사의 법적 책임, 안전 과정 그리고 ISO 26262 요건 준수 방법을 배우실 수 있습니다.
목표	ISO 26262 요건 자동차 개발상의 기능 안전 난제 형식적 안전 관리 과정과 추천받은 지원 과정 위험 분석과 기능 안전 개념 기술 안전 개념 및 시스템 설계 안전 지향 하드웨어 및 소프트웨어 개발 안전 분석에 대한 방법적 접근
교육 내용	1일차 SAFETY MANAGEMENT AND SUPPORTING PROCESSES
	2일차 FROM THE RISK ANALYSIS TO THE FUNCTIONAL SAFETY CONCEPT
	3일차 TECHNICAL SAFETY CONCEPT AND SYSTEM DESIGN
	4일차 SAFETY-ORIENTATED HARDWARE DEVELOPMENT
	5일차 SAFETY-ORIENTATED SOFTWARE DEVELOPMENT

2. 대학

1) 상명대

교육명	상명대		
주최단체	상명대학교	**기관분류**	대학
교육시간	미정	**교육비용**	미정
교육내용	안전성 분석 및 보증 프로세스, 소스코드 품질향상, SW/안전성 Verification		

(1) 개요

국내의 경우 미국, 유럽 등 소프트웨어 안전과 관련된 연구를 활발히 진행하고 있는 선진국에 비해 소프트웨어 안전 관련 대학 교육 및 연구가 거의 진행되고 있지 않은 실정이다. 현재 국내에서는 상명대학교에서 소프트웨어 안전과 관련된 체계적인 커리큘럼을 선보이고 있어 이에 대한 소개를 하고자 한다.

(2) 교육내용

<표 4-20> 상명대 교육 내용

분류	내용	상세 내용	구분	시간
개론	SW 개발에서의 안전성 보증	• SW 개발 특징 • SW 안전성 보증 활동의 필요성	석사/박사	2/2
	임베디드 SW 연구/개발 프로세스 이해	• 임베디드 SW 개발/연구 프로세스 • 임베디드 SW 개발 사례	석사/박사	4/4
안전성 분석	[개론] SW 안전성 분석을 위한 Hazard Analysis 이해	• SW와 안전성 • Safety Engineering (안전성공학) • SW 안전성 분석 프로세스	석사/박사	1/1
	Accident Casualty Model의 이해	• Accident Analysis 및 Hazard Analysis의 이해 • Traiditional Accident Model • System Theory 기반의 Accident Model	석사/박사	2/2
	Chain of Event Model 기반의 Safety Analysis	• HAZOP (Hazard and Operability Studies) • FTA (Fault Tree Analysis) • FMEA (Failure Mode Effects Analysis)	석사/박사	2/2
		• FMEA 실습	박사	2
	System Theory 기반의 Safety Analysis: STAMP	• System Theory 기반의 Process Model • Safety Requirement 및 Constraints • Safety Control Structure	석사/박사	2/2
	System Theory 기반의 Hazard Analysis: STPA	• STPA Process의 이해 • Hazard Analysis 실습	박사	2
	Safety Requirement 도구 실습	• UML 기반의 Safety Requirement 도출 방안	석사/박사	2/2
		• Safety Requirement 도구 실습	박사	2
안전성 보증 프로세스	[개론] SW 안전성 확보를 위한 SW 개발 프로세스	• SW 개발 프로세스 특징 • SW 안전성과 SW 개발 프로세스	석사/박사	1/1
	SW 안전성 확보를 위한 프로세스 정의	• 안전성 확보를 위한 SW 설계 기법 • 결합도/응집도를 고려한 SW 설계	석사/박사	2/2
	ISO26262의 이해	• ISO26262 표준의 이해 및 프로세스 정의 사례	(석사)/박사	(4)/8

	A-SPICE의 이해	• A-SPICE 이해 및 프로세스 정의 사례	(석사)/박사	(4)/8
	CMMI의 이해	• CMMI 이해 및 프로세스 정의 사례	(석사)/박사	(4)/8
소스코드 품질향상	[개론] 코딩 표준 및 리팩토링	• SW 품질 향상을 위한 코딩 표준 및 리팩토링의 이해	석사/박사	1/1
	소스코드 리팩토링 기법	• SW 안전성과 관련한 냄새나는 소스코드 식별 • 안전하게 소스코드를 개선하는 기법	석사/박사	4/4
		• 리팩토링 실습	석사/박사	3/3
	코딩 표준의 이해	• 안전성 확보를 위한 코딩 표준 • MISRA C의 이해	석사/박사	2/2
안전성 Gate	[개론] SW 안전성 확보를 위한 Gate	• SW 품질 점검을 위한 Gate 활동	석사/박사	1/1
	안전성 Gate별 Checklist 및 검사 기법	• 개발 단계 별 안전성 Gate 활동 • Checklist 및 검사 기법의 이해 • Checklist 개발 실습	박사	6
SW Visualization	[개론] SW 품질 수준 가시화와 Tool Chain	• SW Visualization 목적 및 필요성 • 가시화 적용 분야 및 도구	석사/박사	1/1
	형상관리/이슈관리 이론 및 도구 실습	• 형상관리 이론 및 Git/SVN활용 • 이슈관리 이론 및 Redmine 활용	석사/박사	2/2
	정적분석 이론 및 도구 실습	• 소스코드 정적 분석 목적 및 필요성 • 기능안전 표준과의 관계	석사/박사	2/2
		• 도구 활용 및 결과 생성 실습	박사	3
	도구 실습: 도구 간 연계 및 결과 분석	• Jenkins 기반의 도구 연계 기법 • 각 도구 분석 결과의 통합 분석	박사	3
안전성 Verification	[개론] SW 안전성 확보를 위한 테스팅	• SW 안전성과 테스팅 • V모델과 V&V 역할	석사/박사	1/1
	SW 테스팅 기법의 이해	• 테스팅 단계 및 기법 • 테스팅 종류 및 커버리지 • 테스팅 자동화 도구 이해 및 실습	석사/박사	6/6
	휴먼 에러를 고려한 결함/문제 예방 기법	• 결함과 휴먼 에러의 관계 • 결함/문제 예방을 위한 데이터 측정 • 결함/문제 데이터 분석 기법	박사	2

제3절 조사결과 분석

<표 4-21> 국내 현황 조사 결과

구분	주최단체	구분	내용	시간
민간	SOLUTIONLINK	SW공학/표준	SE, SR, FMEA, ISO26262	1~2D
	NAVITHES	표준	ISO26262	1~3D
	모아소프트	표준	EN50128, RTCA DO-178C	2D
	TUV SUD Korea	표준	ISO26262	3D
	STA 테스팅컨설팅	표준	ISO26262, Testing	3D
	SPID	표준/분석기법	ISO26262, CMMI, FTA	1~3D
	SGS	표준	ISO26262, IEC61508	5D
대학	상명대	일반/SW공학/표준	SE, ISO26262, A-SPICE, CMMI, STAMP, STPA	S

(D:일, S:학기)

국내 교육 현황 조사결과 다음과 같은 특징들을 살펴볼 수 있었다.

- 대학 중심의 교육은 연구 중심으로 특정 대학들에서 대학원 과정의 해당 전공자들에게 제공되고 있으나, 해외의 사례와 같이 포괄적이고 심도 깊게 제공되고 있지는 않은 것으로 보인다. 현재 상명대에서 소프트웨어 안전 관련 교과과정이 개설되고 있으며, 그 외 카이스트와 고려대, 건국대 등에서 특정 실험실 내의 특화과정으로 운영되고 있음이 파악되었다.
- 도메인별 교육은 자동차 산업 분야가 가장 활발하게 이루어지고 있으며, 이는 자동차 산업 분야가 수출 산업으로서 소프트웨어 안전 관련 표준인 ISO26262와 관련된 수출 요건의 강화와 잠재적인 결함으로 인한 리콜 예방 차원의 수요 때문인 것으로 파악된다. 대부분의 도메인 교육은 해당 도메인의 전문 컨설팅 회사를 중심으로 이루어지고 있다. 그 외 산업분야(철도, 원자력 등)에서는 필요에 따른 간헐적인 교육이 일어나고 있었던 것으로 파악되나, 공식적인 정규과정들이 정기적으로 개설되고 운영되지는 않고 있다.
- 표준 중심의 교육들은 관련 컨설팅 업체와 국내에서 사업 중인 해외 인증서비스 제공업체에서 수행되고 있음을 파악할 수 있었다.
- 해외선진국의 소프트웨어 안전 전반에 대한 3~5일 정도의 교육 과정을 국내에서는 찾아보기 어려웠다.

제5장 설문조사 및 분석

제1절 개요

안전 필수 소프트웨어 제공자 및 사용자를 위한 필요 교육에 대한 수요 조사는 먼저 소프트웨어 안전 분야의 재직자를 대상으로 소프트웨어 안전 교육에 대한 현황 및 수요에 대한 설문조사를 진행한 후, 안전 인증 경험이 있는 선도 업체들을 대상으로 심층 인터뷰를 실시하였다.

소프트웨어 안전 교육에 대한 설문조사는 현재 현황 및 앞으로의 수요에 대하여 조사를 중점적으로 진행하였다. 국내 여건상 소프트웨어 안전공학 기술에 대한 기초적인 이해 자체가 미흡한 상태여서 기술적으로 심도 있는 질의보다는 소프트웨어 안전에 대한 이해 및 현재 교육현황에 주안점을 두었으며, 향후 필요한 소프트웨어 안전 관련 교육에 대한 기대사항을 파악하는 것으로 초점을 맞추었다. 또한, 보다 구체적인 기술적 수요 파악을 위하여 대표적인 소프트웨어 안전 관련 과정에 대한 수요조사를 실시하였다.

심층 인터뷰는 소프트웨어 안전 관련 공식 혹은 비공식 인증 경험이 있는 선도 업체들을 대상으로 진행하였으며, 안전 필수 소프트웨어 사업현황, 기술역량, 인력역량 등을 기본적으로 파악한 후, IEC61508 표준에 의거한 관련 기술과 관리 방법의 실제 역량 확보 여부를 판단하는데 주안점을 두었다. 또한 정부 차원에서 그리고 범국가적으로 소프트웨어 안전 관련 역량 증진을 위한 기대사항 및 건의사항들도 수집하였다.

제2절 재직자 수요조사

재직자 수요조사의 조사 모집단은 주요 산업군 중 안전 소프트웨어 사용 비중이 높은 원자력, 철도/지하철, 항공/공항/국방, 의료, 미래형자동차, 로봇, 승강기 등의 분야를 우선으로 구축하였으며, 소프트웨어 안전성 전문가 포럼(Software & Safety Experts Forum, SSEF) 회원 기관 및 산업별 안전 규제 기관을 참조하였다. 또한 한국표준산업분류, 특수 분류 산업 중 소프트웨어를 사용/제작하는 업체, 주요 정보통신기반시설 중 안전 관련 시스템을 사용하는 기관 등을 참조하여 다양한 산업군에 대해 조사할 수 있도록 하였다. 이렇게 추출된 표본에 대하여 산학연 전문가 그룹이 검토, 주관기관과

조사 대상 기관을 협의하여 설문조사를 진행하였다.

1. 응답자 특성

수요조사의 대상 업종으로는 원자력, 항공, 자동차, 의료 등 안전 소프트웨어 사용 비중이 높은 분야에서 총 151명을 최대한 골고루 선정하여 진행하였으며, 이 중에서 설문조사의 신뢰성을 높이기 위하여 소프트웨어 안전성 확보가 불필요하다고 응답한 11명을 제외한 총 140명을 대상으로 통계를 정리하였다. 설문조사에 참여한 업종과 각 업종별 응답자 수는 다음과 같다.

<표 5-1> 업종별 설문조사 참여 응답자

업종	응답자 수	비중(%)	업종	응답자 수	비중(%)
원자력	8	5.7	전력 및 수력	11	7.9
기타 에너지	7	5.0	철도 및 지하철	8	5.7
스크린도어	4	2.9	공항	4	2.9
항공	11	7.9	국방	10	7.1
항만	11	7.9	로봇	9	6.4
자동차 및 미래자동차	12	8.6	IoT	12	8.6
의료	9	6.4	엘리베이터	9	6.4
기타기반시설	8	5.7	기타	7	5.0
			총합	140	100.0

[그림5-1] 기업규모별 재직자 수요조사 응답자

[그림5-2] 기업 내 역할별 재직자 수요조사 응답자

기업규모는 중소기업이 가장 많이 응답하였으며, 중견기업 및 대기업, 공기업과 공공기업, 그리고 사단법인이 있었으며, 응답자의 기업 내 역할은 실무 개발자보다는 관리자/경영층이 약 두 배 정도 많았다.

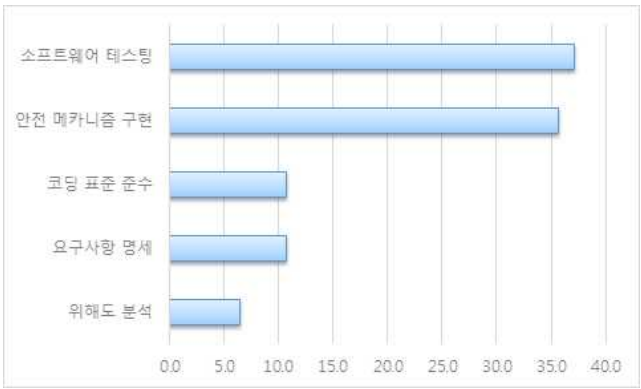
2. 소프트웨어 안전에 대한 인식 및 필요성

[문1] 소프트웨어 안전성 확보를 위하여 가장 중요한 기술적 활동은 무엇이라고 생각하십니까?

① 소프트웨어 테스팅 ② 코딩 표준 준수
③ 위해도 분석 ④ 요구사항 명세
⑤ 안전 메커니즘 구현 ⑥ 기타

[조사 결과]

[그림5-3] 소프트웨어 안전성 확보를 위한 가장 중요한 기술적 활동



◆ 소프트웨어 안전성 확보를 위한 기술적 활동으로는 소프트웨어 테스팅과 안전 메커니즘 구현이 각각 37.1%와 35.7%로 가장 많은 비중을 차지하였다. 코딩 표준 준수와 요구사항 명세는 각각 10.7%를 차지하였으며, 위해도 분석은 6.4%에 그쳤다.

◆ 규모별로는 대기업은 안전 메커니즘 구현을 응답한 비중이 85.7%로 가장 높았지만, 공기업 및 공공기관에서는 코딩 표준 준수를 가장 많이 응답하여 차이를 보였다.

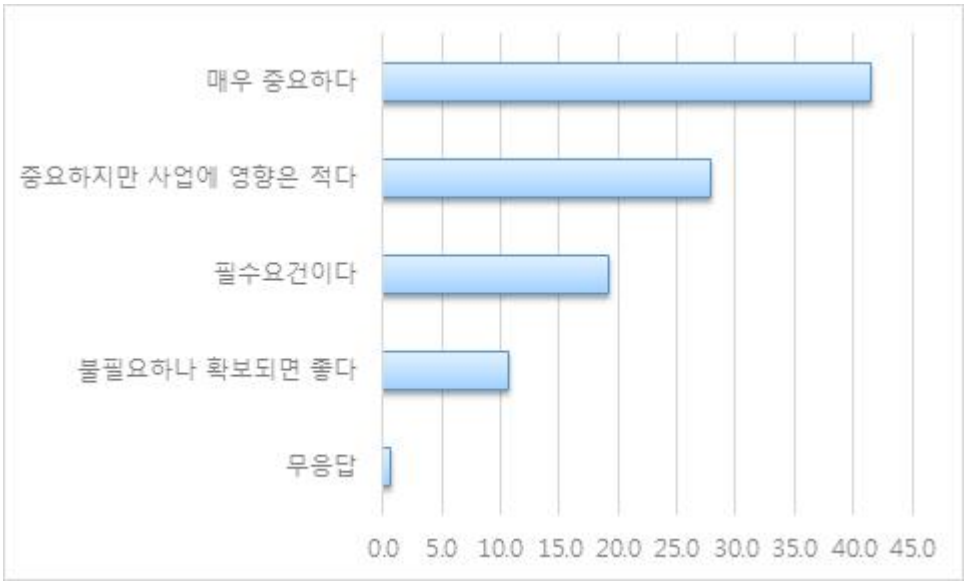
◆ 실제 소프트웨어 안전성 확보를 위하여 갖추어야 할 중요한 기술적 활동은 안전 메커니즘 구현과 위해도 분석이다. 하지만 위해도 분석에 대한 응답이 가장 낮은 결과에서 아직 소프트웨어 안전성에 대한 기술적인 인식이 미흡하다는 것을 유추해볼 수 있다.

[문2] 귀사가 개발하는 제품/시스템의 소프트웨어 안전성 확보가 중요합니까?

① 불필요 ② 불필요하나 확보되면 좋다
③ 중요하지만 사업에 영향은 적다 ④ 매우 중요하다
⑤ 필수요건이다

[조사 결과]

[그림5-4] 개발하는 제품/시스템의 소프트웨어 안전성 확보의 중요성



* “불필요” 응답은 분석 대상에서 제외

◆ 개발중인 제품/시스템의 소프트웨어 안전성 확보의 필요성에 대한 응답으로는 ‘매우 중요하다’가 41.4%로 가장 많았으며, ‘중요하지만 사업에 영향은 적다’가 27.9%, ‘필수요건이다’가 19.3%, ‘불필요하나 확보되면 좋다’가 10.7%를 차지하였다.

<표 5-2>분야별 안전성 중요도

분야(총 응답 수)	불필요하나 확보되면 좋다	중요하지만 사업에 영향은 적다	매우 중요하다	필수요건이다
원자력(8)	25.0	25.0	37.5	12.5
전력 및 수력(11)	18.2	18.2	36.4	27.3
기타 에너지(7)	0.0	14.3	71.4	14.3
철도 및 지하철(8)	0.0	50.0	50.0	0.0
스크린도어(4)	0.0	0.0	50.0	50.0
공항(4)	0.0	0.0	75.0	25.0
항공(11)	9.1	27.3	36.4	27.3
국방(10)	10.0	30.0	40.0	20.0
항만(11)	27.3	9.1	27.3	36.4
로봇(9)	0.0	66.7	22.2	11.1
자동차/미래자동차(12)	8.3	33.3	50.0	8.3
IoT(12)	25.0	41.7	25.0	8.3
의료(9)	0.0	11.1	77.8	11.1
엘리베이터(9)	22.2	0.0	44.4	33.3
기타기반시설(8)	0.0	37.5	50.0	12.5
기타(7)	0.0	57.1	0.0	28.6

산업 분야별 안전성 확보의 중요도 차이를 비교해보기 위하여 분야별 응답 비율을 살펴보았다. (무응답 제외, 단위 %)

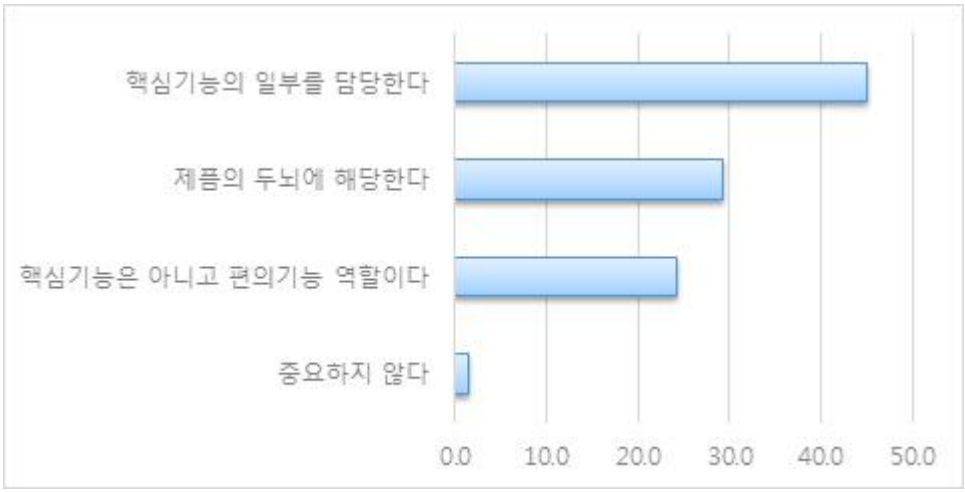
◆ 실제 소프트웨어 안전성이 중요한 산업 분야에서 모두 응답하였기 때문에 중요도의 차이가 분야별로 크게 차이가 나지는 않았다. 원자력, 항만, IoT 및 엘리베이터 분야에서 다른 분야에 비해 불필요하다는 응답의 비중이 약간 높은 것을 알 수 있었다.

[문3] 귀사가 개발하는 제품/시스템에서 소프트웨어 중요성(역할적 비중)은 어떠합니까?

① 중요하지 않다. ② 핵심기능은 아니고 편의기능 역할이다.
③ 핵심기능의 일부를 담당한다. ④ 제품의 두뇌에 해당한다.

[조사 결과]

[그림5-5] 개발하는 제품/시스템에서 소프트웨어 중요성의 비중



◆ 개발 중인 제품/시스템의 소프트웨어 중요성(비중)에 대한 응답으로는 '핵심 기능의 일부를 담당한다.' 는 응답이 45.0%로 가장 많은 비중을 차지하였다. 그 다음으로는 '제품의 두뇌에 해당한다.' 는 응답이 29.3%, '핵심기능은 아니고 편의 기능 역할이다' 는 응답이 24.3%를 차지하였고, '중요하지 않다' 는 응답은 1.4%에 그쳤다.

[문4] 귀사는 소프트웨어 안전 관련 요건이 귀사 제품의 인허가 혹은 수출 요건에 필수적입니까?

① 예 ② 아니오

[조사 결과]

[그림5-6] 소프트웨어 안전 관련 요건의 필수 여부

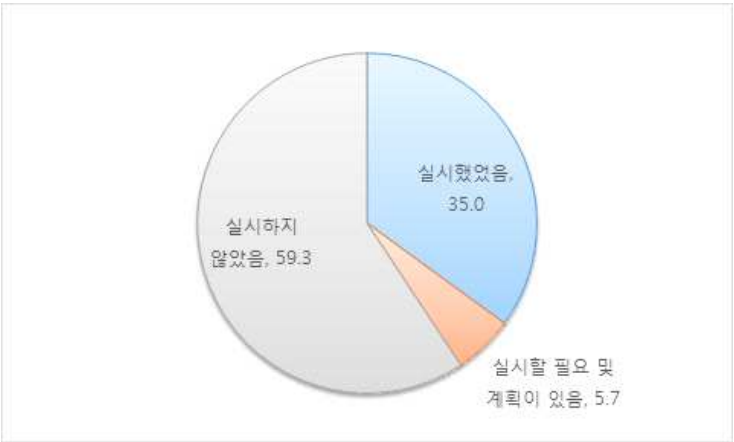
◆ 소프트웨어 안전 관련 요건이 제품의 인허가 혹은 수출 요건에 필수적인가에 대한 응답으로는 '예'가 44.3%, '아니오'가 55.0%, 그리고 무응답이 0.7%를 각각 차지하였다.

[문5] 귀 부서에서는 2010년 1월 이후 소프트웨어 안전 관련 교육을 실시한 적이 있습니까?

① 실시했었음 ② 실시할 필요 및 계획이 있음
③ 실시하지 않았음

[조사 결과]

[그림 5-7] 소프트웨어 안전 관련 교육 실시 여부



◆ 소프트웨어 안전 관련 교육 실시 여부에 대한 응답으로는 ‘실시하지 않았음’이 59.3%로 절반을 넘었다. 소프트웨어 안전 관련 교육을 실시했던 업체는 35.0%, 실시할 필요 및 계획이 있는 업체는 5.7%를 차지하였다.

◆ 앞서 소프트웨어 안전이 개발 중인 제품 및 시스템에서 중요하다고 응답한 비중이 매우 높음에도 불구하고 소프트웨어 안전 교육을 실시하지 않은 업체가 과반수이상을 차지한 것에서 현재 소프트웨어 안전 교육이 원활하게 이루어지지 않고 있음을 유추해볼 수 있다.

<표 5-3> 소프트웨어 중요도(문3)에 따른 교육 실시(문5)

문3 \ 문5	실시했거나 실시할 계획이 있다	실시하지 않았다
제품의 두뇌에 해당	25(61.0%)	16(39.0%)
핵심기능의 일부	26(41.3%)	37(58.7%)
편의기능 역할	6(17.6%)	28(82.4%)
중요하지 않다	0(0.0%)	2(100.0%)

( ) 안은 문3의 각 응답에 대한 문5의 응답 비율

이후의 응답(문5-1부터 문5-9까지)은 본 문항에서 소프트웨어 안전 관련 교육을 실시하거나 실시할 계획이 있는 업체들을 대상으로 조사하였다.

[문5-1] 귀 부서에서 교육수요가 발생하는 원인을 2가지만 응답해 주십시오.

① 소프트웨어 안전사고 방지를 위해서 ② 발주자의 요청 때문에
③ 안전관련 법/제도/표준의 준수요건 때문에 ④ 내부 역량 확보 차원에서
⑤ 수출 요건 때문에 ⑥ 기타

[조사 결과]

[그림5-8] 교육수요의 발생원인

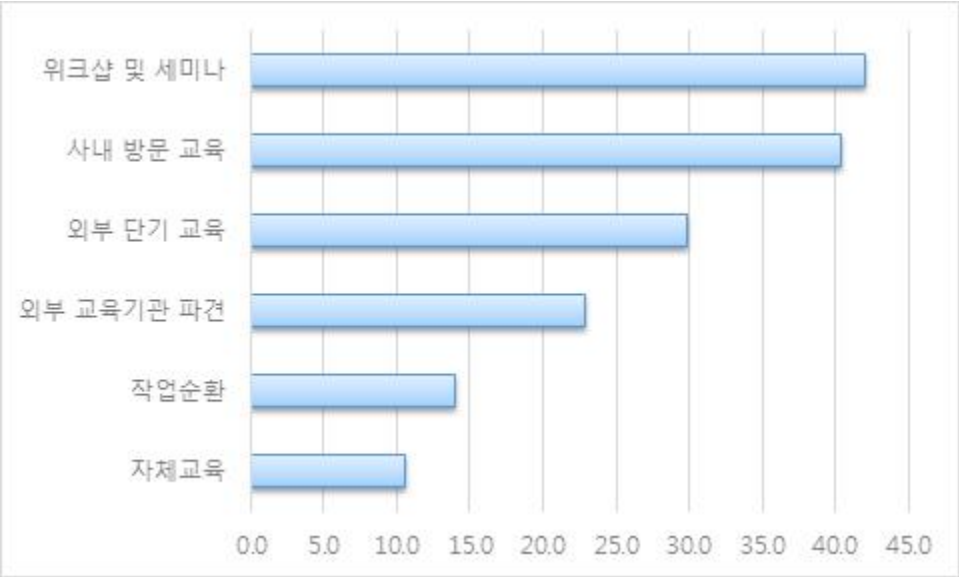
- 교육수요의 발생 원인으로는 ‘소프트웨어 안전사고 방지를 위하여’를 선택한 응답자가 64.9%, ‘안전관련 법/제도/표준의 준수요건 때문에’를 선택한 응답자가 57.9%로 많은 부분을 차지하였다. ‘내부 역량 확보 차원에서’라는 응답이 31.6%, ‘발주자의 요청 때문에’라는 응답이 15.8%, ‘수출 요건 때문에’라는 응답이 5.3%를 각각 차지하였다.
- 교육은 규제와 외적인 요인보다는 제품 자체의 안전 중요도 인식에서 수요가 발생하는 것으로 추측된다.

[문5-2] 귀 부서에서 역점을 두고 시행하는 교육방식을 2가지만 응답해 주십시오.

① 사내 방문 교육 ② 외부 교육기관 파견
③ 외부 단기 교육 ④ 워크샵 및 세미나
⑤ 작업순환 ⑥ 기타

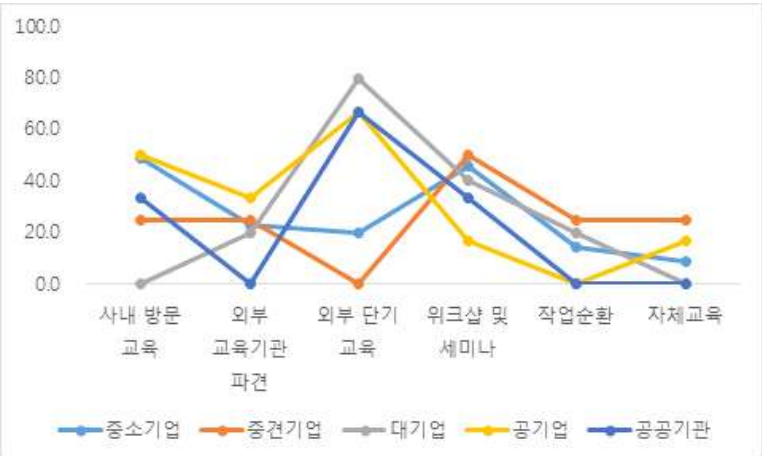
[조사 결과]

[그림5-9] 부서에서 시행하는 교육방식



- 교육방식으로는 '워크샵 및 세미나'가 42.1%로 가장 많았으며, '사내 방문 교육'도 40.4%로 많은 비중을 차지하였다. 그 외에도 '외부 단기 교육'이 29.8%, '외부 교육기관 파견'이 22.8%, '작업순환'을 통한 교육이 14.0%, 자체 교육을 실시한 업체가 10.5%를 각각 차지하였다.
- 주로 외부에서 진행되는 교육보다는 사내에서 자체적으로 또는 초청 강사를 통한 교육을 좀 더 선호하는 것으로 나타났다.
- 기업 규모에 따른 교육방식의 선호 차이를 알아보기 위하여 규모별 응답 비율을 살펴보았다. (무응답 제외, 단위 %)

[그림5-10] 기업규모에 따른 교육방식 선호도 비교



<표 5-4> 기업규모에 따른 교육방식 선호도

기업규모	사내 방문 교육	외부 교육 기관 파견	외부 단기 교육	워크샵 및 세미나	작업순환	자체교육
중소기업	48.6	22.9	20.0	45.7	14.3	8.6
중견기업	25.0	25.0	0.0	50.0	25.0	25.0
대기업	0.0	20.0	80.0	40.0	20.0	0.0
공기업	50.0	33.3	66.7	16.7	0.0	16.7
공공기관	33.3	0.0	66.7	33.3	0.0	0.0

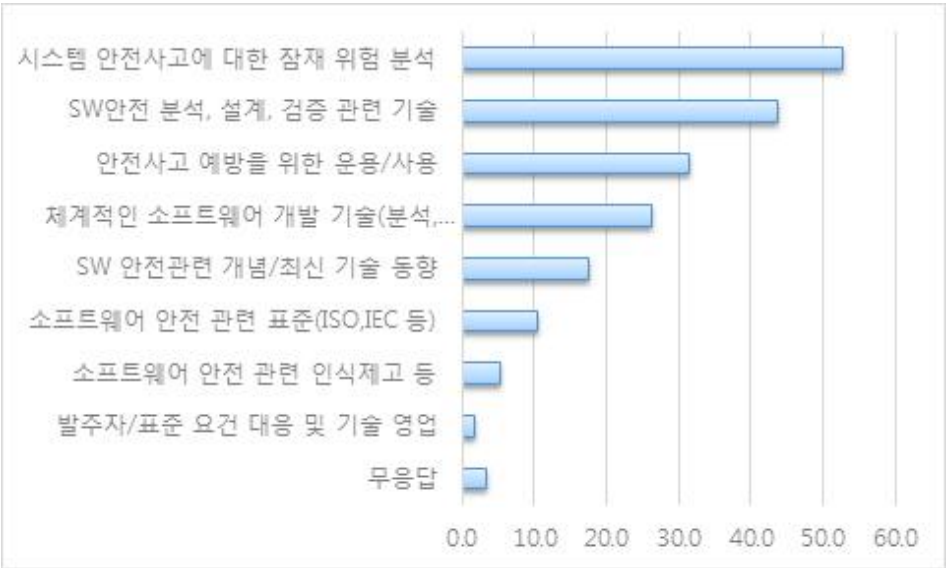
◆ 중소기업에서는 사내 방문 교육을 주로 선호하는데 반해(48.6%) 대기업 및 공기업에서는 외부 단기 교육을 더욱 선호하는 것으로 드러나(80.0%/66.7%) 규모에 따른 차이를 보였다. 이는 아무래도 기업 규모가 클수록 교육 예상 인원이 많아 사내에서 교육하기에 어려움이 있기 때문인 것으로 유추해볼 수 있다.

[문5-3] 귀 부서에서 교육을 실시하는데 있어서 중요하게 생각하는 내용은 무엇인지 2가지만 응답해 주십시오.

① SW안전관련 개념 / 최신 기술 동향
② 시스템 안전사고에 대한 잠재위험 분석
③ SW안전 분석, 설계, 검증 관련 기술
④ 안전사고 예방을 위한 운용/사용
⑤ 소프트웨어 안전 관련 표준(ISO, IEC등)
⑥ 발주자/표준 요건 대응 및 기술 영업
⑦ 소프트웨어 안전 관련 인식제고
⑧ 체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)
⑨ 기타

[조사 결과]

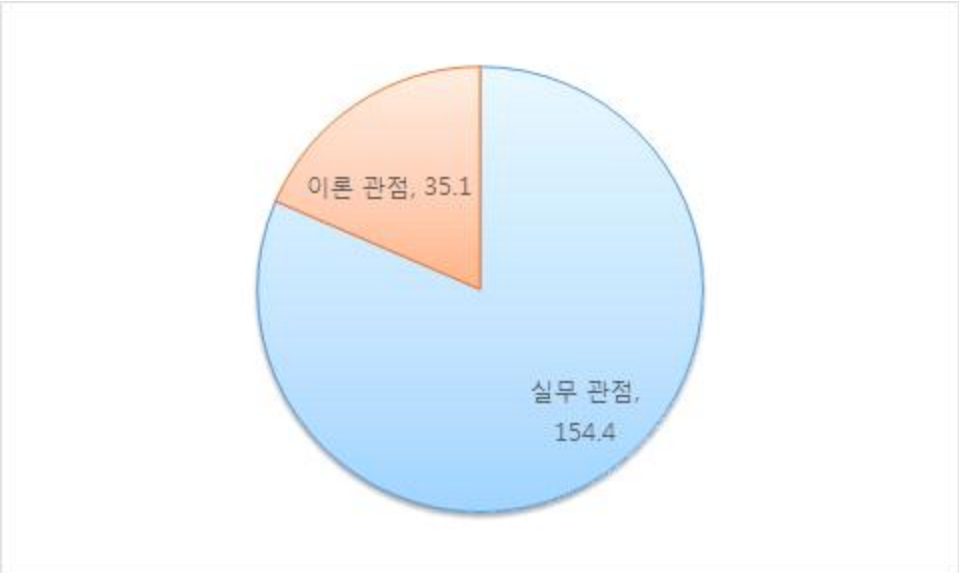
[그림5-11] 교육 시 중요하게 생각하는 내용



◆ 교육내용으로는 ‘시스템 안전사고에 대한 잠재 위험 분석’이 52.6%로 가장 많았으며, ‘SW안전 분석, 설계, 검증 관련 기술’이 43.9%, ‘안전사고 예방을 위한 운용/사용’이 31.6%로 많은 비중을 차지하였다. ‘체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)’을 답한 응답자가 26.3%, ‘SW안전관련 개념/최신 기술 동향’을 답한 응답자가 17.5%이며, ‘소프트웨어 안전 관련 표준(ISO, IEC 등)’, ‘소프트웨어 안전 관련 인식제고’, ‘발주자/표준요건 대응 및 기술 영업’을 답한 응답자는 각각 10.5%, 5.3%, 1.8%에 그쳤다.

◆ 보기에 제시된 내용들을 실무 관점과 이론 관점으로 분류하여 그 선호도의 차이를 살펴보기로 하였다. ‘시스템 안전사고에 대한 잠재위험 분석’, ‘SW안전 분석, 설계, 검증 관련 기술’, ‘안전사고 예방을 위한 운용/사용’, ‘체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)’을 실무 관점의 주제, ‘SW안전관련 개념 / 최신 기술 동향’, ‘소프트웨어 안전 관련 표준(ISO, IEC등)’, ‘발주자/표준 요건 대응 및 기술 영업’, ‘소프트웨어 안전 관련 인식제고’를 이론 관점으로 분류하여 그 비중을 합산한 결과는 다음과 같다.

[그림 5-12] 교육시 중요하게 생각하는 내용: 실무/이론 관점에서



◆ 위 결과에서 실무와 관련이 있는 주제에 대한 선호도가 기초적인 이론에 대한 교육보다 선호도가 매우 높음을 알 수 있다.

◆ 실제 교육에서는 기본 개념 및 표준에 대한 이론적인 뒷받침이 있어야 실무와 관련된 소프트웨어 안전 분석, 설계, 검증 관련 기술이나 체계적인 소프트웨어 개발 기술에 대한 교육을 진행할 수 있지만, 위와 같이 뒷받침이 되어야 할 교육에 대한 실질적인 수요가 크게 없고, 교육 여건상 이론적인 내용을 별도로 진행하기가

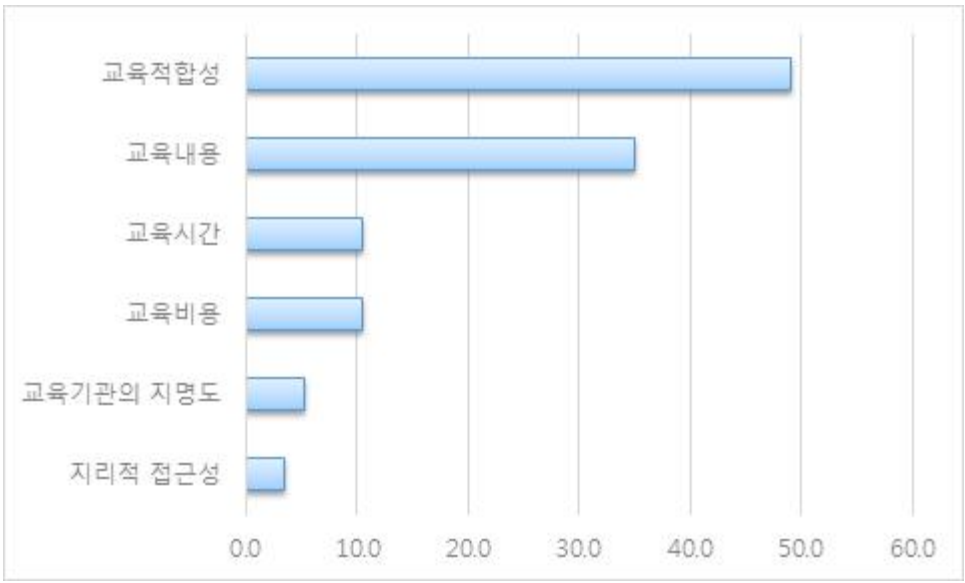
쉽지 않은 것이 현실이다. 따라서 실무 관점에서의 교육 위주로 진행함과 동시에, 교육의 진행에 앞서 필요한 배경 지식 정도를 교육 과정에 추가하는 것도 고려해 볼 수 있겠다.

[문5-4] 귀 부서에서 교육과정 및 기관 선정시 가장 중요한 기준은 무엇입니까?

① 교육시간
② 교육비용
③ 교육기관의 지명도
④ 지리적 접근성
⑤ 교육적합성
⑥ 교육내용
⑦ 기타

[조사 결과]

[그림5-13] 교육과정/기관 선정 시 중요한 기준



- 교육과정 및 기관 선정시 가장 중요한 기준에 대한 응답으로는 '교육적합성'이 49.1%로 가장 많았으며, '교육내용'도 35.1%로 상당수를 차지하였다. '교육시간', '교육비용', '교육기관의 지명도', '지리적 접근성'은 각각 10.5%, 10.5%, 5.3%, 3.5%에 그쳤다.
- 보기에 제시된 기준들을 실제 교육과 관련된 기준과 교육 외적인 기준으로 분류하여 그 중요성의 차이를 비교해 보기로 하였다. '교육적합성'과 '교육내용'은 실제 교육과 관련이 있기에 실제 교육과 관련된 기준으로 분류하고, 나머지 '교육시간', '교육비용', '교육기관의 지명도', '지리적 접근성' 등은 교육 외

적인 기준으로 분류하여 그 비중을 합산한 결과는 다음과 같다.

[그림5-14] 교육과정/기관 선정 시 중요한 기준: 교육과의 관련성 관점에서

◆ 위 결과에서 교육 외적인 부분보다는 교육과 업무의 연관성, 그리고 교육의 실제 내용이 중요하다는 것을 알 수 있었다.

[문5-5] 귀 부서에서 소프트웨어 안전 혹은 품질관련 교육을 받은 기관은 어디이며, 교육에 대한 만족도는 어느 정도입니까? 해당란에 √ 표기해 주시기 바랍니다.

교육기관	교육 참여여부		교육 만족도		
	참여	미참여	만족	보통	불만족
회사내부 자체적으로 실시	①	②	①	②	③
학교(대학, 전문대학 등)	①	②	①	②	③
공공기관/협회	①	②	①	②	③
SW 안전분야 관련 전문 기업	①	②	①	②	③
외부 전문가 초청 내부 교육	①	②	①	②	③
기타	①	②	①	②	③

[조사 결과]

◆ 아래는 소프트웨어 안전 또는 품질관련 교육을 실시한 기관 및 기관별 만족도에 관한 결과다. 교육기관으로는 ‘회사내부 자체적으로 실시’, ‘학교(대학, 전문대학 등)’, ‘공공기관/협회’, ‘SW 안전분야 관련 전문 기업’, ‘외부 전문가 초청 내부 교육’ 등이 있었다.

① 회사내부 자체적으로 실시

[그림5-15] 회사내부 교육 참여 여부, 만족도

◆ 회사내부 자체적으로 실시한 교육에서는 응답자 중 73.3%가 참여하였으며, 이 중 63.6%의 응답자가 만족, 33.3%의 응답자가 보통의 만족도를 보였다.

② 학교

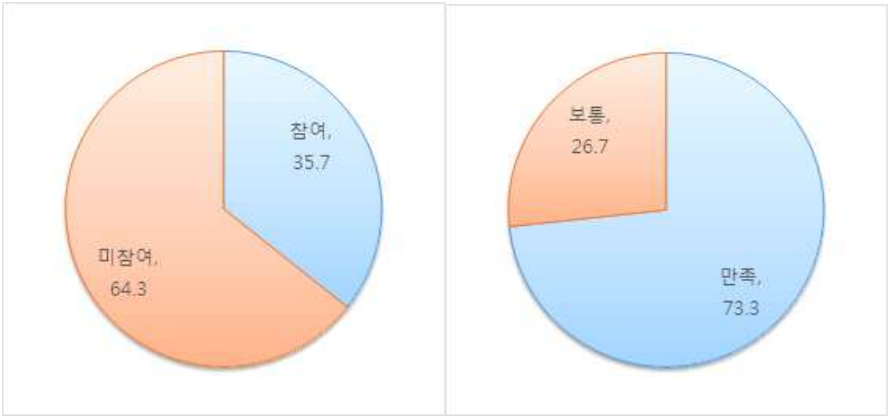
[그림5-16] 학교 교육의 참여 여부

◆ 대학교, 전문대학 등의 학교 교육은 국방 관련 업체 한 곳에서만 참여하고 나머지 업체에서는 참여하지 않은 것으로 응답하였다. 학교 교육에 참여한 업체의 교육 만족도는 '보통'이었다.

◆ 다른 컴퓨터 공학 분야에 비해 소프트웨어 안전 분야는 기업과 대학의 연계교육은 거의 일어나지 않는 것으로 보여 진다.

③ 공공기관/협회

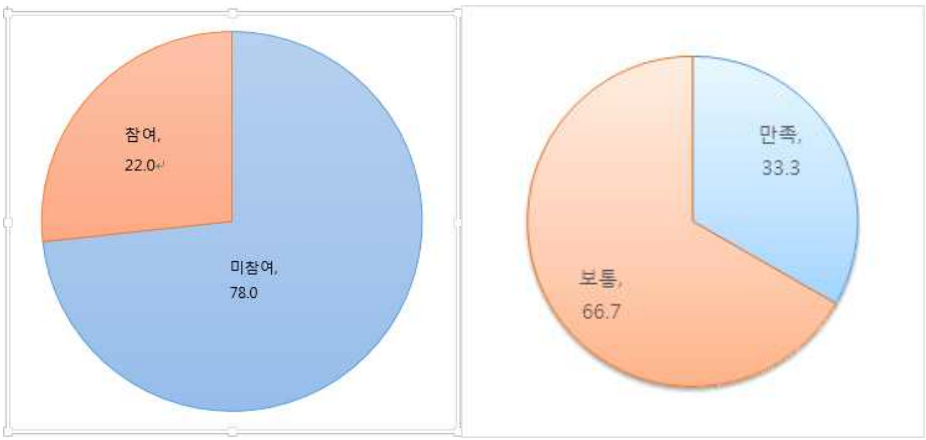
[그림5-17] 공공기관/협회 교육의 참여 여부, 만족도



◆ 공공기관/협회에서 주최하는 교육은 응답자 중 35.7%가 참여하였으며, 이 중 73.3%의 응답자가 만족, 26.7%의 응답자가 보통의 만족도를 보였다.

④ SW 안전분야 관련 전문 기업

[그림5-18] 전문 기업 교육의 참여 여부, 만족도



◆ SW 안전 분야 전문기업에서 주최하는 교육은 응답자 중 22.0%가 참여하였으며, 이 중 33.3%의 응답자가 만족, 66.7%의 응답자가 보통의 만족도를 보였다.

◆ 이는 한정된 필수 교육 수요자가 참여했으며, SW 안전 분야 전문기업에서 주최하는 교육은 교육 수요자의 다른 교육 과정에 비해 기대에 미치지 못한 것으로 판단된다.

⑤ 외부 전문가 초청 내부 교육

[그림5-19] 외부 전문가 초청 교육의 참여 여부, 만족도

◆ 외부 전문가 초청 내부 교육은 응답자 중 23.1%가 참여하였으며, 이 중 55.6%의 응답자가 만족, 22.2%의 응답자가 보통, 22.2%의 응답자가 불만족하였다.

⑥ 기타

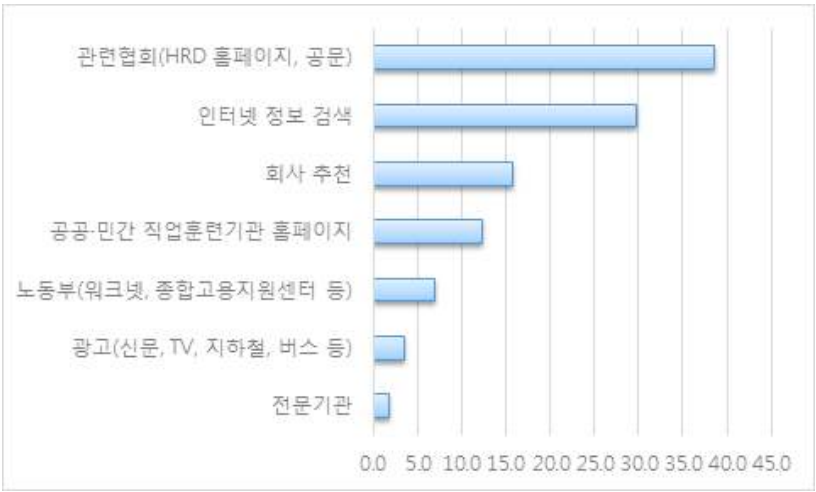
◆ 기타 의료 업계 관련 두 업체에서 의료기기 안정기관에서 주최한 교육에 참석하여 모두 만족하였다.

[문5-6] 귀 부서에서는 교육 관련 정보를 주로 어디서 얻습니까?

① 회사 추천
② 인터넷 정보 검색
③ 관련협회(HRD 홈페이지, 공문)
④ 노동부(워크넷, 종합고용지원센터 등)
⑤ 공공/민간 직업훈련기관 홈페이지
⑥ 광고(신문, TV, 지하철, 버스 등)
⑦ 기타

[조사 결과]

[그림5-20] 교육 관련 정보 획득 경로



◆ 교육 관련정보 획득 경로로는 관련협회(HRD 홈페이지, 공문)를 통한 정보 획득이 38.6%로 가장 많았으며, 인터넷 정보 검색이 29.8%, 회사 추천이 15.8%, 공공/민간 직업훈련기관을 통한 정보 획득이 12.3%로 뒤를 이었으며, 노동부(워크넷, 종합고용지원센터 등)를 통한 정보 획득은 7.0%, 광고(신문, TV, 지하철, 버스 등)가 3.5%, 전문기관을 통한 정보 획득은 1.8%에 그쳤다.

[문5-7] 귀 부서의 교육활동 수준이 적절하다고 보십니까?

① 적당함 ② 많음
③ 부족함 ④ 모르겠음

[조사 결과]

[그림5-21] 부서의 교육활동 수준

◆ 교육활동의 수준에 대한 응답으로는 적당함이 63.2%, 부족함이 26.3%, 많음이 7.0%, 모르겠음이 3.5%를 차지하였다.

◆ 교육활동의 수준에 대한 응답은 기업규모에 따라 다른 반응을 보였다. 다음은 기업규모별 교육활동의 수준에 대한 응답이다. (단위 %)

<표 5-5> 기업규모별 교육활동의 수준

기업규모	적당함	많음	부족함	모르겠음
중소기업	65.7	5.7	25.7	2.9
중견기업	62.5	0.0	37.5	0.0
대기업	60.0	40.0	0.0	0.0
공기업	83.3	0.0	16.7	0.0
공공기관	0.0	0.0	66.7	33.3

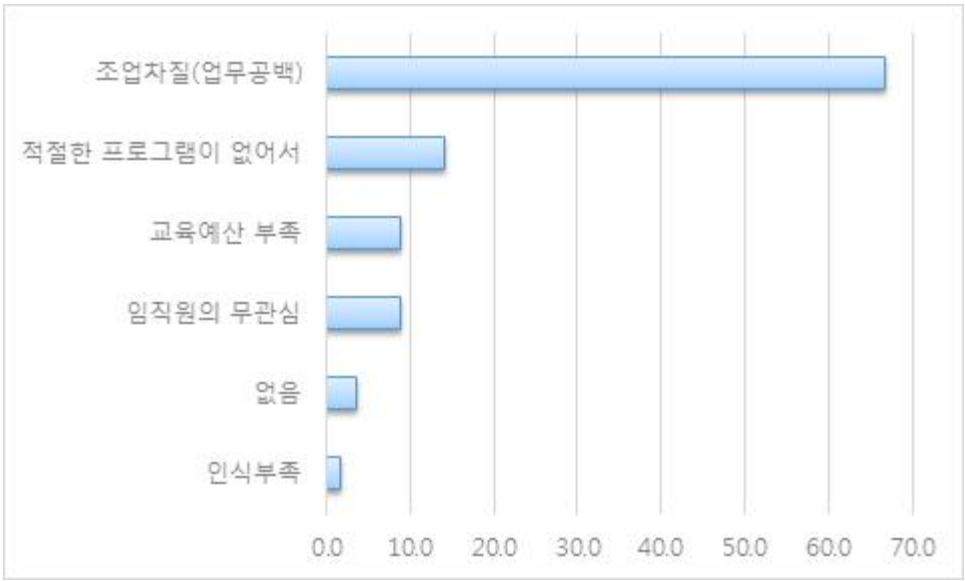
◆ 위 결과에서 중소기업과 중견기업은 교육활동의 수준이 부족하다는 응답이 각각 25.7%와 37.5%로 적지 않은 비중을 차지하고 있지만, 대기업의 경우 오히려 많다는 응답이 40.0%이고 부족하다는 응답은 없어 차이를 보였다. 대기업의 경우 주로 표준 관련 요건들이 필수적으로 들어가고, 따라서 이에 대한 교육이 컨설팅 회사를 통해 어느 정도 활발하게 이루어지고 있기 때문인 것으로 추측된다.

[문5-8] 귀 부서에서 교육을 실시하는데 있어서 가장 큰 걸림돌은 무엇입니까?

① 조업차질(업무공백)　② 적절한 프로그램이 없어서
③ 교육예산 부족　④ 임직원의 무관심
⑤ 기타　⑥ 없음

[조사 결과]

[그림5-22] 교육을 실시하는데 있어 차지하는 걸림돌



◆ 교육을 실시하는데 있어서 가장 큰 걸림돌에 대한 응답으로 '조업차질(업무공백)'이 66.7%로 과반수이상을 차지하였다. 그 외에 '적절한 프로그램이 없어서'가 14.0%, '교육예산 부족'과 '임직원의 무관심'이 각각 8.8%을 차지하였다. 걸림돌이 없다는 의견은 3.5%가 있었으며, 기타 의견으로 '인식부족'을 꼽는 응답이 있었다.

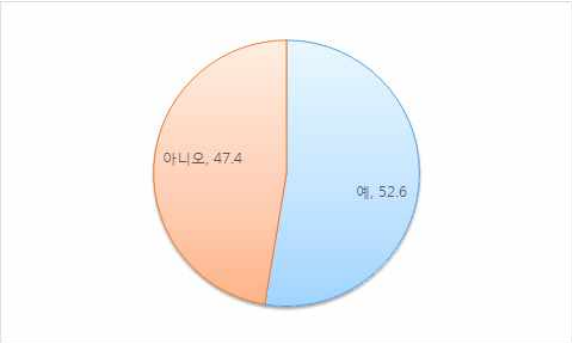
◆ 교육을 실시하는데 있어서 가장 중요한 것은 업무공백을 최소화하는 것으로 교육시간 확보 및 환경 구축이 가장 중요하다는 의견이 주를 이루었다. 업무 공백 최소화는 교육 제공 시 해결해야 하는 가장 중요한 문제점 중의 하나로 판단된다.

[문5-9] 귀 기관에는 소프트웨어 안전분석/위험분석 도구(FMEA, FTA 지원 도구를 의미함)을 보유 및 사용하고 계십니까?

① 예 ② 아니오

[조사 결과]

[그림5-23] SW 안전분석/위험분석 도구 보유/사용 여부



- SW 안전 분석/위험분석 도구 보유 및 사용 여부에 대한 응답은 전체 응답자의 52.6%가 ‘보유 및 사용하고 있다’, 나머지 47.6%의 응답자가 ‘사용하고 있지 않다’라고 응답하였다.
- 다음은 기업규모별 SW 안전 분석/위험분석 도구 보유 및 사용 여부에 대한 응답이다.

<표 5-6> 기업규모별 SW 안전 분석/위험분석 도구 보유 및 사용 여부

기업규모	예	아니오
중소기업	45.7	54.3
중견기업	100.0	0.0
대기업	80.0	20.0
공기업	33.3	66.7
공공기관	0.0	100.0

- 위 결과에서 중견기업 및 대기업의 경우에는 대부분의 업체가 안전분석/위험분석 도구를 보유하고 사용 중이지만, 중소기업의 경우 절반 이상의 업체에서 도구를 보유하고 있지 않은 것을 살펴볼 수 있다. 이는 SW 안전에 대한 인식이 부족하거나, SW 안전 분석/위험분석 도구의 사용방법을 숙지하지 못하였기 때문에, 또는 어느 정도 SW 안전에 대한 인식이 있더라도 기업 규모가 작아 비용이 부담되는

SW 안전 분석/위험분석 도구를 갖추지 못한 경우일 것으로 해석될 수 있다.

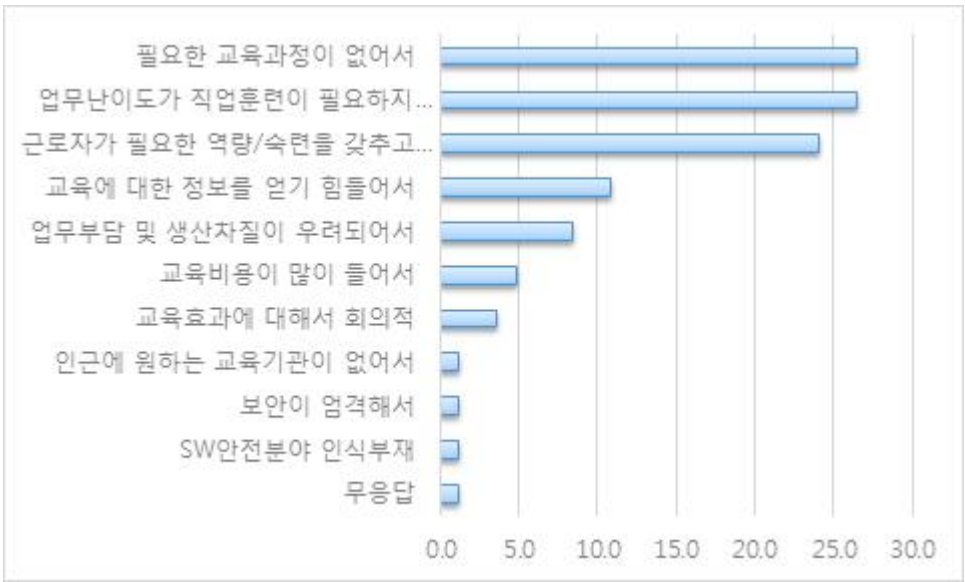
◆ 공기업 및 공공기관에서는 중소기업보다 더 나은 여건을 가지고 있음에도 불구하고 SW 안전 분석/위험분석 도구의 보유 및 사용 여부가 가장 낮은 것으로 드러났다. 이는 현재 공기업 및 공공기관의 SW 안전에 대한 인식이 매우 부족함을 나타내는 것이라고 볼 수 있다.

[문5-10] 귀 부서에서 2010년 1월 이후 교육을 실시하지 않은 이유는 무엇입니까?

① 근로자가 필요한 역량/숙련을 갖추고 있어서 ② 필요한 교육과정이 없어서
③ 교육비용이 많이 들어서 ④ 업무부담 및 생산차질이 우려되어서
⑤ 업무난이도가 직업훈련이 필요하지 않아서 ⑥ 교육효과에 대해서 회의적
⑦ 인근에 원하는 교육기관이 없어서 ⑧ 교육에 대한 정보를 얻기 힘들어서
⑨ 기타

[조사 결과]

[그림5-24] 교육을 실시하지 않은 이유



◆ 문5-10은 문5-1에서 소프트웨어 안전 관련 교육을 실시하지 않은 업체들을 대상으로 하여 그 이유에 대해 질문하였다. 이에 대한 응답으로는 '필요한 교육이 없어서'가 26.5%, '업무난이도가 직업훈련이 필요하지 않아서'가 26.5%, '근로자가 필요한 역량/숙련을 갖추고 있어서'가 24.1%로 많았으며, 그 외에도 '교육에 대한 정보를 얻기 힘들어서' 10.8%, '업무분담 및 생산차질이 우려되어서' 8.4%, '교육비용이 많이 들어서' 4.8%, '교육효과에 대해서 회의적' 3.6%, '인근에

원하는 교육기관이 없어서’ 1.2%, ‘보안이 엄격해서’ 1.2%, SW 안전분야 인식 부재’ 1.2% 등의 응답이 있었다.

- 교육 환경을 개선하여 소프트웨어 안전 교육을 활성화하기 위해서는 좀 더 다양한 교육 과목을 개설하여 근무자의 필요를 충족시켜줄 필요가 있다고 보여진다. 그 외에 교육에 대한 홍보와 업무공백을 최소화할 수 있는 환경개선 등이 필요할 것으로 생각된다.

- 실제로 현업에서 업무의 종류에 따라 SW 안전이 필요한 도메인이라도 SW 안전과는 무관한 업무일 경우가 있어 SW 안전 관련 교육이 필요하지 않을 수도 있다. ‘업무난이도가 직업훈련이 필요하지 않아서’에 대한 응답이 두 번째로 많은 것은 이에 대한 반증이라고 유추해볼 수 있다.

[문6] 귀사에서 소프트웨어 안전 기술 관련 교육을 실시한다면 예상 교육 인원은 몇 명입니까? (　　명)

[조사 결과]

[그림5-25] 예상 교육 인원

- 소프트웨어 안전 기술 관련 교육 실시 시 예상 교육 인원에 대한 응답은 10명 이하, 11~50명, 51명 이상으로 나누어 통계를 내었다. 전체 응답자의 3/4에 해당하는 75.7%의 응답자가 10명 이하라고 응답하였으며, 11~50명이 16.4%, 51명 이상이 6.4%를 각각 차지하였고, 무응답이 1.4% 있었다.

- 기업 규모에 따른 예상 교육 인원은 다음과 같다.

[그림5-26] 중소기업, 중견기업/대기업의 예상 교육 인원

[그림5-27] 공기업/공공기관의 예상 교육 인원

◆ 위 기업 규모에 따른 예상 교육 인원을 살펴보면 기업의 인원수가 상대적으로 적은 중소기업의 경우 10명 이하로 응답한 비중이(83.5%) 중견기업 및 대기업(58.3%)보다 많았으며, 기업의 인원수가 많은 대기업에서는 반대로 51명 이상을 응답한 비중이 중소기업(2.1%)보다 많아(25.0%) 기업 규모에 따른 차이를 살펴볼 수 있었다.

◆ 다음은 전체 SW 개발 인력 대비 예상 교육 인원의 비율을 조사한 결과이다.

<표 5-7> 전체 SW 개발 인력 대비 예상 교육 인원의 비율

예상 교육 인원 / 전체 SW 개발 인력	100% 초과	100%	50% ~ 100%	50% 미만	무응답
비율(%)	14.3%	43.6%	18.6%	15.0%	8.5%

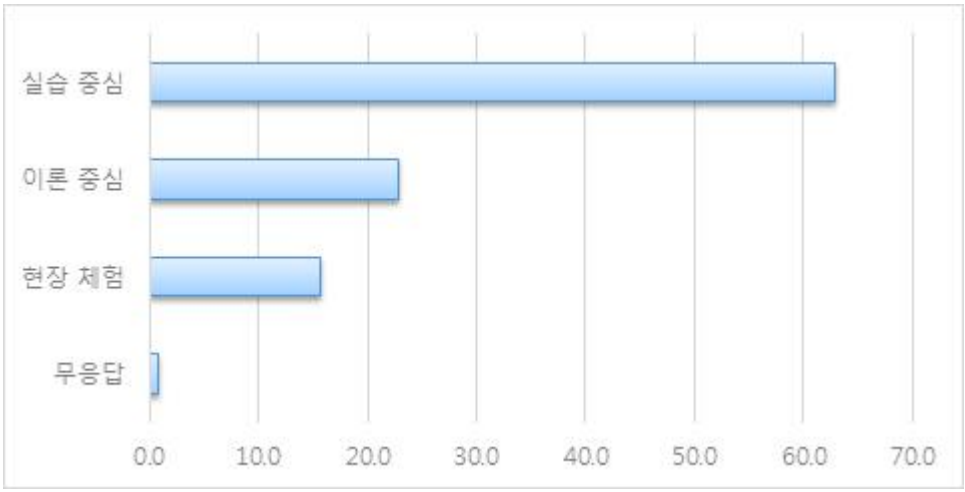
◆ 절반에 가까운 43.6%의 응답자가 전체 SW 개발 인력이 곧 예상 교육 인원이라고 응답하였다. 14.3%의 응답자는 오히려 전체 SW 개발 인력보다 더 많은 인원을 예

상 교육 인원으로 응답하였는데, 이는 SW 개발 인력이 아니더라도 교육 대상자가 될 수 있음을 의미한다.

[문7] 귀사에서 소프트웨어 안전 기술 관련 교육시 선호하는 교육방법은 무엇입니까?	
① 이론 중심	② 실습 중심
③ 현장 체험	④ 기타

[조사 결과]

[그림5-28] 선호하는 교육방법

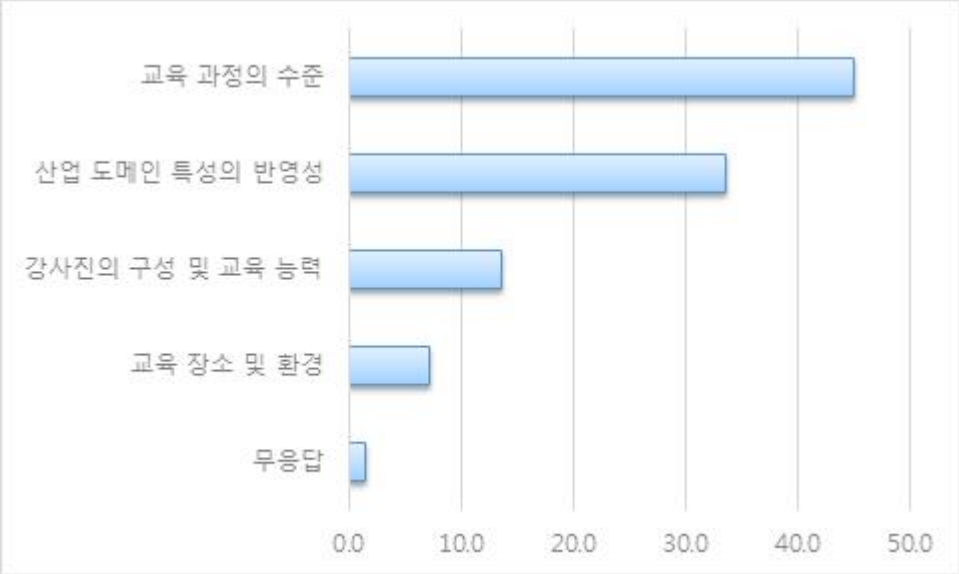


◆ 소프트웨어 안전 기술 관련 교육 시 선호하는 교육방법으로는 '실습 중심'이 62.9%로 가장 많았으며, '이론 중심'이 22.9%, '현장 체험'이 15.7%를 각각 차지하여 이론 중심의 교육보다는 실습 중심의 교육을 선호한다고 응답하였다. 이는 실무에 활용할 수 있는 실질적인 교육을 원하는 것으로 유추된다.

[문8] 귀사에서 소프트웨어 안전 기술 관련 교육 참여시 다음 항목 중 가장 중요하게 생각하는 항목은 무엇입니까?	
① 교육 과정의 수준	② 강사진의 구성 및 교육 능력
③ 산업 도메인 특성의 반영성	④ 교육 장소 및 환경
⑤ 기타	

[조사 결과]

[그림5-29] SW안전 교육 참여 시 중요하게 생각하는 항목

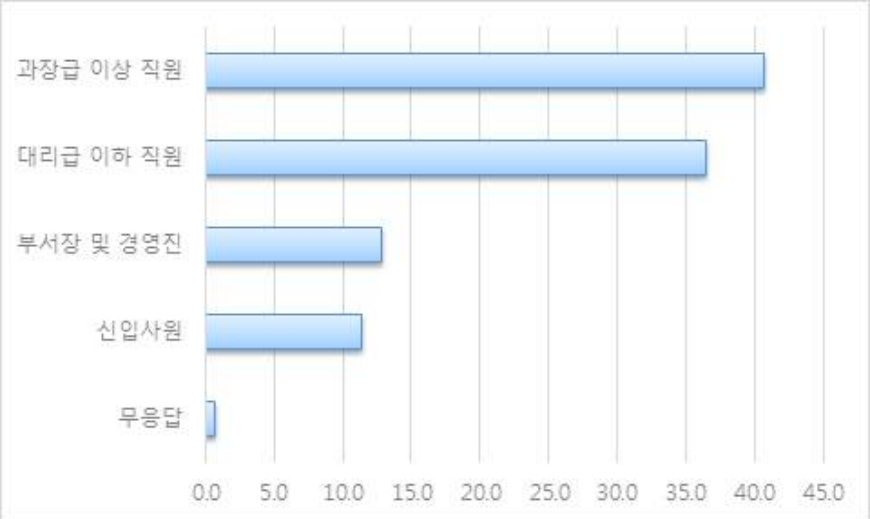


- 소프트웨어 안전 기술 관련 교육 참여시 중요하게 생각하는 것으로는 ‘교육 과정의 수준’이 45.0%로 가장 많이 차지하였으며, ‘산업 도메인 특성의 반영성’이 33.6%, ‘강사진의 구성 및 교육 능력’이 13.6%, ‘교육 장소 및 환경’이 7.1%로 각각 뒤를 이었다.
- 교육과 직접 관련된 ‘교육 과정의 수준’과 ‘산업 도메인 특성의 반영성’이 교육 외적인 부분인 ‘강사진의 구성 및 교육 능력’, ‘교육 장소 및 환경’보다 높은 응답률을 보인 것을 확인할 수 있다.
- 문5-4에서와 마찬가지로 실제 교육과 관련된 부분과 교육 외적인 부분으로 나누어서 살펴보면 78.6%(교육 과정의 수준 + 산업 도메인 특성의 반영성)와 20.7%(강사진의 구성 및 교육 능력 + 교육 장소 및 환경)로 문5-4에서와 비슷한 양상을 보여, 재직자들이 교육 외적인 부분보다는 교육과 업무의 연관성, 그리고 교육의 실제 내용을 중요하게 여긴다는 것을 알 수 있다.

[문9] 귀 부서에서 교육을 가장 필요로 하는 직급은 무엇입니까?	
① 신입사원	② 대리급 이하 직원
③ 과장급 이상 직원	④ 부서장 및 경영진

[조사 결과]

[그림5-30] SW안전 교육을 필요로 하는 직급



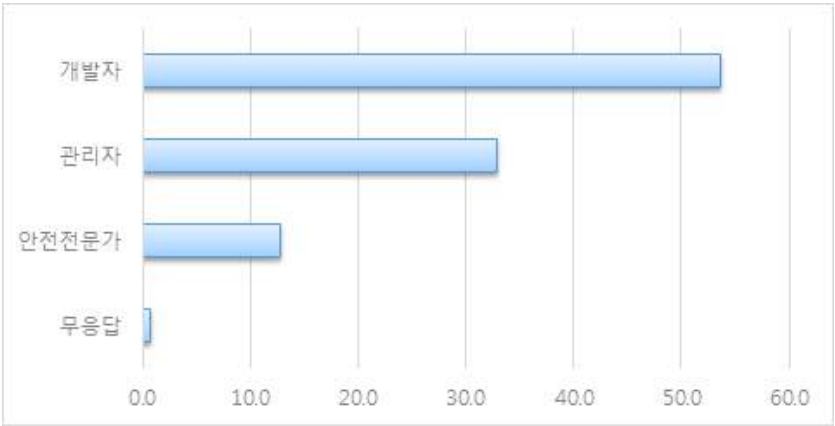
◆ 소프트웨어 안전 기술 관련 교육을 가장 필요로 하는 직급에 대한 응답으로는 과장급 이상 직원이 40.7%, 대리급 이하 직원이 36.4%로 대다수를 차지했으며, 부서장 및 경영진이 필요하다는 응답이 12.9%, 신입사원이 필요하다는 응답이 11.4%로 나머지를 차지하였다. 이 결과에서 어느 정도 업무능력을 갖춘 실무자 중심의 교육 수요가 많다는 것을 유추할 수 있다.

[문10] 귀사에서 교육을 가장 필요로 하는 직무는 무엇입니까?

① 관리자 ② 개발자
③ 도메인 ④ 안전전문가
⑤ 기타

[조사 결과]

[그림5-31] SW안전 교육을 필요로 하는 직무



◆ 소프트웨어 안전 기술 관련 교육을 가장 필요로 하는 직무에 대한 응답으로는 개발자가 53.6%로 가장 많았으며, 관리자가 필요하다는 응답이 32.9%, 그리고 안전전문가가 필요하다는 응답은 12.9%에 그쳤다.

◆ 교육을 가장 필요로 하는 직무에 대한 응답은 기업 규모별로 차이를 보였다. 다음은 기업 규모별 교육을 가장 필요로 하는 직무에 대한 응답이다. (무응답 제외, 단위 %)

<표 5-8> 기업 규모별 교육을 가장 필요로 하는 직무

기업규모	관리자	개발자	안전전문가
중소기업	34.0	50.5	15.5
중견기업	11.8	70.6	17.6
대기업	0.0	100.0	0.0
공기업	88.9	11.1	0.0
공공기관	33.3	55.6	0.0

◆ 위 결과에서 기업 규모가 커질수록, 즉 중소기업보다는 대기업에서 관리자보다는 개발자라고 응답한 비중이 높았으며, 반대로 발주기관에 주로 해당되는 공기업의 경우에는 관리자라고 응답한 비중이 일반 기업에 비해 크게 높음을 알 수 있다.

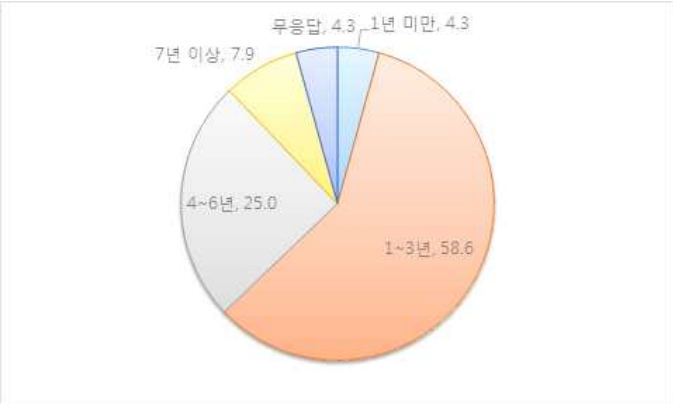
◆ 실제 소프트웨어 안전 기술을 담당해야 할 안전전문가에 대한 응답은 중소기업 및 중견기업에서만 조금 찾아볼 수 있었으며, 주요 발주기관인 공기업 및 대기업에서는 찾아볼 수 없었다. 이는 현재 산업 전반에서 개발자와 안전전문가에 대한 구별이 잘 되어있지 않으며, 아직까지는 안전전문가에 대한 인식이 미흡하다는 반증이다.

[문11] 귀 부서에 신입사원이 입사하여 SW안전 관련 전문 인력이 될 때까지의 소요 기간은 어느 정도 된다고 생각하십니까?

(년)

[조사 결과]

[그림5-32] SW안전 관련 전문 인력이 될 때까지의 소요기간



- 신입사원이 소프트웨어 안전 관련 전문 인력이 될 때까지의 소요기간에 대한 응답은 1년 이하, 1~3년, 4~6년, 7년 이상으로 나누어 통계를 내었다. 전체 응답자의 58.6%가 1~3년, 25.0%가 4~6년이라고 대답했다.
- 해외 전문가의 의견에 따르면 실제 소프트웨어 안전 관련 전문 인력이 되기 위해서는 평균적으로 7년 이상의 오랜 기간이 필요하다고 한다. 아직 국내의 소프트웨어 안전 관련 인식이 상대적으로 부족하여 1~3년과 4~6년에 대한 응답의 비중이 높은 것으로 추측된다.

[문12] 귀 부서에서는 2017년에 교육을 실시할 계획을 가지고 있습니까?
① 있다 ② 없다

[조사 결과]

[그림5-33] 2017년 교육 실시 계획 여부

◆ 2017년 교육 실시 여부에 대한 응답으로는 교육을 실시할 계획이 없다는 응답이 58.6%로 있다고 응답한 40.0%의 응답자보다 많았다.

◆ 다음은 앞서 응답한 소프트웨어 안전 관련 교육 실시 여부와 본 문항과의 관계를 나타낸 것이다.

〈표 5-9〉 소프트웨어 안전 관련 교육 실시 여부와 본 문항과의 관계

문5 / 문12	실시했거나 실시할 계획이 있다	실시하지 않았다
실시할 계획이 있다	37	19
실시할 계획이 없다	18	64

◆ 위 결과에서 교육을 실시했던 업체의 1/3 가량이 올해 교육을 다시 실시할 계획이 없음을 확인할 수 있다. 이는 교육에 대한 만족도가 업체의 기대에 미치지 못하였거나, 지속적인 교육의 필요성을 느끼지 못하였기 때문인 것으로 생각되며, 현재보다 좀 더 양질의 체계적인 교육 커리큘럼을 마련하여 재직자들이 지속적인 교육을 받을 수 있도록 노력해야 할 것이다.

◆ 또한 응답자의 절반 가까이가 교육을 받은 적도 없고, 2017년에도 교육을 실시할 계획이 없다고 밝혔다. 이는 소프트웨어 교육의 인프라가 그만큼 부족하다는 것을 의미하며, 이를 개선하기 위해 소프트웨어 교육의 필요성을 알리고 좀 더 다양한 소프트웨어 교육을 실시하여 재직자들에게 소프트웨어 교육의 기회를 늘리는 것이 필요하다고 생각된다.

[문13] 귀 부서의 업무형태 및 교육의 성과를 고려하였을 때, 참여 가능한 시간대와 적정 교육기간은 어느 정도라고 생각하십니까? (각각 응답해 주십시오)

(1) 교육참여 가능 시간대	(2) 적정 교육기간
① 평일 오전 ② 평일 오후 ③ 평일 종일 ④ 주말 ⑤ 기타	① 1일 ② 2~3일 ③ 4~5일 ④ 6~10일 ⑤ 기타

[조사 결과]

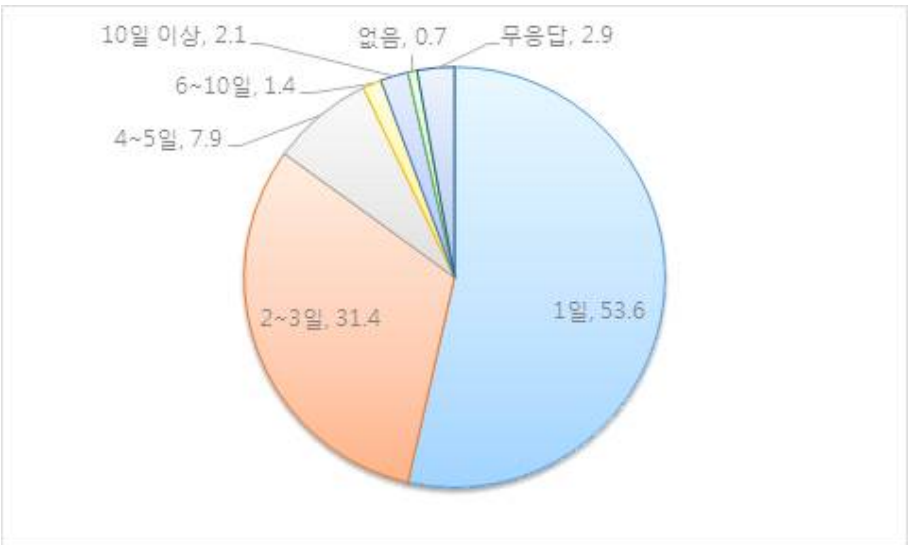
① 교육 참여 가능 시간대

[그림5-34] 참여 가능한 교육 시간대

- 교육 참여 가능 시간대로는 평일 종일이 50.0%로 가장 많았으며, 평일 오후가 25.0%, 평일 오전이 17.9%를 차지하였다. 주말이라고 응답한 응답자는 4.3%에 불과하였다.
- 주말보다는 평일을 선호하며, 오전/오후 한정적으로 이루어지는 반나절 교육보다는 하루를 온전히 투자하여 진행되는 교육을 좀 더 선호하는 것으로 나타났다.

② 적정 교육기간

[그림5-35] 적정 교육 기간



◆ 적정 교육기간으로는 1일이 53.6%로 가장 많았으며, 2~3일이 31.4%, 4~5일이 7.9%, 6~10일이 3.5%를 차지하였다.

◆ 아래는 앞서 살펴본 문5-8의 결과이다. 아래 결과에서 알 수 있듯이 교육을 실시함에 있어 가장 큰 고려사항으로 업무공백을 꼽고 있으며, 교육기간이 길어질수록 업무공백과 관련하여 차질이 발생할 수 있기에 3일 이내의 단기교육을 더욱 선호하는 것으로 생각해볼 수 있다.

[문14] 정부 주도의 소프트웨어 안전 교육과정을 개설한다면 교육 제도를 다음 중 어떤 방식으로 운영하는 것이 좋다고 생각하십니까?

① 안전관련 사업 시 필수 의무 교육 ② 선택 교육

③ 자격증(예. SW안전전문가)과 연계된 교육

[조사 결과]

[그림5-36] 정보 주도의 SW안전 교육과정의 선호하는 운영방식

◆ 정부 주도의 소프트웨어 안전 교육과정 개설 시 선호하는 운영방식에 대해서는 ‘자격증과 연계된 교육’이 40.0%, ‘선택 교육’이 35.7%, ‘안전관련 사업 시 필수 의무 교육’이 23.6%로 각각 응답하였다.

◆ 자격증과 연계하여 간접적으로 소프트웨어 안전 기술을 강제하는 교육이 선택 교육을 통해 업체의 자율에 맡기는 것 보다 선호되는 것으로 보인다. 필수 의무 교

육을 통한 소프트웨어 안전 기술의 강제는 오히려 선호도가 떨어지는 것으로 알 수 있었다.

3. SW 안전 교육과정 선호도

[문15] SW 안전 관련 교육과정 선호도를 파악하기 위해 아래 각 코스별 선호도와 시급성을 상, 중, 하로 응답해 주십시오. (잘 모를 경우 모름에 체크)

1) 교육과정 선호도 조사방법

- 설문지에 제안한 교육과정으로는 '소프트웨어 안전 인식 제고', '소프트웨어 안전 개념 및 기초 이론', '소프트웨어 안전 공학 방법론 개론', '시스템 안전 분석', '시스템 안전 설계', '소프트웨어 안전 분석', '소프트웨어 안전 설계 및 구현', '소프트웨어 코드 안전 분석', '소프트웨어 안전성 테스팅 기법', '고안전성 시스템들을 위한 정형 기법', '소프트웨어 안전 관련 국제 표준/규제', '안전표준 및 규제에 따른 안전공학기법 이론 교육', '소프트웨어 안전성 관리', '형상관리 및 결함관리', '안전 품질 및 측정관리', '소프트웨어 안전 Audit' 등이 있으며 각각 필요성과 시급성에 대해 상, 중, 하로 측정하도록 하였다.
- 가중치는 상에 5, 중에 3, 하에 1을 곱하여 각각의 합으로 계산하였다. 무응답 비율은 가중치 계산에서 제외하였다.

2) 교육과정 선호도 조사결과 개요

- 필요성이 높다고 응답된 과정 순으로 나열하면 아래와 같다. (단위 %, 무응답은 제외)

<표 5-10> 교육과정 선호도 조사 결과 (필요성 순서)

안전 기술분야	상	중	하	가중치
소프트웨어 안전 인식 제고	51.4	34.3	9.3	369.2
소프트웨어 안전성 테스팅 기법	49.3	37.9	7.9	368.1
소프트웨어 안전성 관리	47.9	39.3	7.9	365.3
형상관리 및 결함관리	45.7	42.1	7.1	361.9
안전 품질 및 측정관리	43.6	43.6	7.9	356.7
시스템 안전 설계	47.1	36.4	9.3	354.0
소프트웨어 안전 설계 및 구현	45.0	37.9	7.9	346.6

소프트웨어 안전 분석	41.4	42.9	7.9	343.6
소프트웨어 코드 안전 분석	40.7	42.9	10.0	342.2
소프트웨어 안전 개념 및 기초 이론	36.4	46.4	11.4	332.6
소프트웨어 안전 관련 국제 표준/규제	39.3	40.0	14.3	330.8
시스템 안전 분석	36.4	44.3	10.0	324.9
안전표준/규제에 따른 안전공학기법 이론	28.6	50.0	12.9	305.9
소프트웨어 안전 Audit	25.7	54.3	10.0	301.4
소프트웨어 안전 공학 방법론 개론	22.1	57.1	13.6	295.4
고안전성 시스템들을 위한 정형 기법	27.1	45.0	12.1	282.6

◆ 시급성이 높다고 응답된 과정 순으로 나열하면 아래와 같다. (단위 %, 무응답은 제외)

<표 5-11> 교육과정 선호도 조사 결과 (시급성 순서)

안전 기술분야	상	중	하	가중치
형상관리 및 결함관리	42.1	47.4	9.8	362.5
소프트웨어 안전성 테스팅 기법	42.1	46.6	9.8	360.1
시스템 안전 설계	40.0	48.5	10.8	356.3
소프트웨어 안전성 관리	39.8	47.4	12.0	353.2
소프트웨어 안전 설계 및 구현	37.0	51.2	10.2	348.8
안전 품질 및 측정관리	36.8	50.4	12.0	347.2
소프트웨어 코드 안전 분석	30.5	55.0	13.7	331.2
소프트웨어 안전 분석	31.8	52.7	14.0	331.1
소프트웨어 안전 인식 제고	33.8	48.1	17.3	330.6
시스템 안전 분석	29.9	53.5	14.2	324.2
소프트웨어 안전 개념 및 기초 이론	29.5	50.8	17.4	317.3
고안전성 시스템들을 위한 정형 기법	28.0	53.4	16.9	317.1
소프트웨어 안전 Audit	22.2	63.5	13.5	315
소프트웨어 안전 관련 국제 표준/규제	25.2	54.2	19.1	307.7
안전표준/규제에 따른 안전공학기법 이론	21.1	59.4	18.8	302.5
소프트웨어 안전 공학 방법론 개론	20.0	56.9	20.8	291.5

◆ 전반적으로 시급성이 높다고 생각되는 과정들이 필요성도 높다고 응답하였다. 단, 예외적으로 ‘소프트웨어 안전 인식 제고’ 과정은 전반적인 인식에 관한 문제라고 생각되어서인지 중간 정도의 시급성으로 응답하였지만 필요성에 대해서는 가장 필요한 과정이라고 응답하였다.

◆ 필요성/시급성이 높다고 응답된 과정은 주로 ‘소프트웨어 안전성 테스팅 기법’, ‘소프트웨어 안전성 관리’, ‘시스템 안전 설계’, ‘형상관리 및 결함관리’, ‘소프트웨어 안전 설계 및 구현’ 등 주로 실무와 연관성이 높다고 판단되는 과정들이 많이 선정되었다.

◆ 테스팅 과정이 필요성 및 시급성 상위에 있고, 소프트웨어 안전 분석 과정이 필요성 및 시급성이 높지 않은 결과에서 수요자들이 소프트웨어 안전에 대한 지식이 아직 부족한 것을 알 수 있다.

3) 기업규모별 선호도 조사결과 및 차이점

◆ 다음은 기업규모별 교육과정의 필요성 및 시급성에 대한 차이를 알아보기 위한 결과이다. 교육과정의 필요성에 대한 위 응답 중 중소기업 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-12> 중소기업에서 교육과정 필요성에 대한 응답

안전 기술 분야	상	중	하	가중치
소프트웨어 안전성 테스팅 기법	45.4	42.3	8.2	361.9
소프트웨어 안전성 관리	44.3	42.3	9.3	357.7
소프트웨어 안전 인식 제고	42.3	43.3	10.3	351.5
형상관리 및 결함관리	39.2	48.5	8.2	349.5
안전 품질 및 측정관리	39.2	47.4	9.3	347.4
시스템 안전 설계	40.2	42.3	10.3	338.1
소프트웨어 코드 안전 분석	36.1	46.4	11.3	330.9
소프트웨어 안전 분석	36.1	46.4	9.3	328.9
소프트웨어 안전 설계 및 구현	37.1	43.3	9.3	324.7
소프트웨어 안전 관련 국제 표준/규제	30.9	47.4	16.5	313.4
소프트웨어 안전 개념 및 기초 이론	27.8	52.6	14.4	311.3
시스템 안전 분석	27.8	50.5	13.4	304.1
안전표준/규제에 따른 안전공학기법 이론	18.6	56.7	16.5	279.4
소프트웨어 안전 공학 방법론 개론	15.5	60.8	17.5	277.3
소프트웨어 안전 Audit	17.5	58.8	12.4	276.3
고안전성 시스템들을 위한 정형 기법	19.6	45.4	15.5	249.5

◆ 교육과정의 시급성에 대한 위 응답 중 중소기업 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-13> 중소기업에서 교육과정 시급성에 대한 응답

안전 기술분야	상	중	하	가중치
소프트웨어 안전성 테스팅 기법	36.6	50.5	10.8	345.2
소프트웨어 안전성 관리	35.5	50.5	12.9	341.9
시스템 안전 설계	35.6	50.0	13.3	341.1
형상관리 및 결함관리	33.3	53.8	11.8	339.8
안전 품질 및 측정관리	33.3	52.7	12.9	337.6
소프트웨어 안전 설계 및 구현	32.2	54.0	11.5	334.5
소프트웨어 코드 안전 분석	27.5	57.1	14.3	323.1
소프트웨어 안전 분석	27.0	57.3	13.5	320.2
소프트웨어 안전 인식 제고	28.0	52.7	18.3	316.1
시스템 안전 분석	23.6	56.2	16.9	303.4
소프트웨어 안전 개념 및 기초 이론	23.9	55.4	17.4	303.3
고안전성 시스템들을 위한 정형 기법	21.8	56.4	19.2	297.4
소프트웨어 안전 관련 국제 표준/규제	19.6	57.6	20.7	291.3
소프트웨어 안전 Audit	12.8	69.8	16.3	289.5
안전표준/규제에 따른 안전공학기법 이론	15.7	62.9	20.2	287.6
소프트웨어 안전 공학 방법론 개론	14.3	59.3	23.1	272.5

◆ 교육과정의 필요성에 대한 위 응답 중 중견기업 및 대기업 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-14> 중견기업/ 대기업에서 교육과정 필요성에 대한 응답

안전 기술분야	상	중	하	가중치
소프트웨어 안전 인식 제고	75.0	16.7	4.2	429.2
시스템 안전 설계	66.7	29.2	0.0	420.8
소프트웨어 안전 설계 및 구현	66.7	29.2	0.0	420.8
시스템 안전 분석	58.3	37.5	0.0	404.2
소프트웨어 안전 분석	58.3	37.5	0.0	404.2
소프트웨어 안전성 테스팅 기법	58.3	37.5	0.0	404.2
소프트웨어 안전성 관리	58.3	37.5	0.0	404.2
형상관리 및 결함관리	58.3	37.5	0.0	404.2
소프트웨어 안전 개념 및 기초 이론	58.3	33.3	4.2	395.8
소프트웨어 코드 안전 분석	54.2	41.7	0.0	395.8
안전표준/규제에 따른 안전공학기법 이론	54.2	37.5	4.2	387.5
안전 품질 및 측정관리	50.0	45.8	0.0	387.5
소프트웨어 안전 Audit	45.8	50.0	0.0	379.2
고안전성 시스템들을 위한 정형 기법	41.7	54.2	0.0	370.8
소프트웨어 안전 관련 국제 표준/규제	54.2	29.2	8.3	366.7
소프트웨어 안전 공학 방법론 개론	29.2	66.7	0.0	345.8

◆ 교육과정의 시급성에 대한 위 응답 중 중견기업 및 대기업 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-15> 중견기업/ 대기업에서 교육과정 시급성에 대한 응답

안전 기술분야	상	중	하	가중치
시스템 안전 설계	61.8	38.2	0.0	423.6
형상관리 및 결함관리	57.6	42.4	0.0	415.3
소프트웨어 안전성 테스팅 기법	53.5	46.5	0.0	406.9
시스템 안전 분석	49.3	50.7	0.0	398.6
소프트웨어 안전 설계 및 구현	49.3	50.7	0.0	398.6
소프트웨어 안전성 관리	45.1	50.7	4.2	381.9
소프트웨어 안전 Audit	41.0	59.0	0.0	381.9
소프트웨어 안전 분석	45.1	46.5	8.3	373.6
소프트웨어 안전 인식 제고	43.8	47.9	8.3	370.8
소프트웨어 코드 안전 분석	36.8	59.0	4.2	365.3
고안전성 시스템들을 위한 정형 기법	36.8	59.0	4.2	365.3
안전 품질 및 측정관리	36.1	59.7	4.2	363.9
소프트웨어 안전 개념 및 기초 이론	36.1	51.4	12.5	347.2
소프트웨어 안전 공학 방법론 개론	22.9	72.9	4.2	337.5
소프트웨어 안전 관련 국제 표준/규제	28.7	58.0	13.3	330.9
안전표준/규제에 따른 안전공학기법 이론	27.8	59.7	12.5	330.6

◆ 교육과정의 필요성에 대한 위 응답 중 공기업 및 공공기관 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-16> 공기업/ 공공기관에서 교육과정 필요성에 대한 응답

안전 기술분야	상	중	하	가중치
소프트웨어 안전 인식 제고	66.1	11.1	11.1	377.8
소프트웨어 안전 관련 국제 표준/규제	61.1	16.7	11.1	366.7
형상관리 및 결함관리	61.1	16.7	11.1	366.7
소프트웨어 안전 개념 및 기초 이론	55.6	27.8	5.6	366.7
소프트웨어 안전 설계 및 구현	55.6	22.2	11.1	355.6
안전 품질 및 측정관리	55.6	22.2	11.1	355.6
시스템 안전 설계	55.6	16.7	16.7	344.4
소프트웨어 안전성 테스팅 기법	55.6	16.7	16.7	344.4
소프트웨어 안전성 관리	50.0	27.8	11.1	344.4
소프트웨어 안전 분석	44.4	33.3	11.1	333.3
고안전성 시스템들을 위한 정형 기법	44.4	33.3	11.1	333.3
안전표준/규제에 따른 안전공학기법 이론	44.4	33.3	5.6	327.8
시스템 안전 분석	50.0	22.2	5.6	322.2

소프트웨어 코드 안전 분석	44.4	27.8	16.7	322.2
소프트웨어 안전 Audit	38.9	38.9	11.1	322.2
소프트웨어 안전 공학 방법론 개론	44.4	27.8	11.1	316.7

◆ 교육과정의 시급성에 대한 위 응답 중 공기업 및 공공기관 재직자들로 응답자를 한정시킨 결과는 아래와 같다. (단위 %, 무응답은 제외)

<표 5-17> 공기업/ 공공기관에서 교육과정 시급성에 대한 응답

안전 기술분야	상	중	하	가중치
형상관리 및 결함관리	67.5	19.8	12.7	409.5
소프트웨어 안전성 테스팅 기법	56.3	23.8	19.8	373.0
안전 품질 및 측정관리	54.8	25.4	19.8	369.8
소프트웨어 안전성 관리	53.2	27.0	19.8	366.7
소프트웨어 안전 관련 국제 표준/규제	47.6	32.5	19.8	355.6
소프트웨어 안전 인식 제고	53.2	19.8	27.0	352.4
소프트웨어 안전 개념 및 기초 이론	53.2	19.8	27.0	352.4
소프트웨어 안전 Audit	43.7	36.5	19.8	347.6
소프트웨어 안전 설계 및 구현	42.1	39.7	18.3	347.6
시스템 안전 설계	29.4	57.9	12.7	333.3
고안전성 시스템들을 위한 정형 기법	42.1	31.0	27.0	330.2
안전표준/규제에 따른 안전공학기법 이론	36.1	41.7	22.2	327.8
시스템 안전 분석	32.2	46.7	21.1	322.2
소프트웨어 안전 분석	36.5	36.5	27.0	319.0
소프트웨어 코드 안전 분석	34.9	38.1	27.0	315.9
소프트웨어 안전 공학 방법론 개론	41.7	19.4	38.9	305.6

◆ 중소기업 재직자들의 응답 결과는 전체 응답 결과와 큰 차이가 없었다. 다만, 전체 결과에서 필요성이 가장 높게 나온 ‘소프트웨어 안전 인식 제고’에 대한 응답이 상대적으로 조금 낮게 나와, 실무자 위주의 교육을 원하는 것으로 유추되나, 중소기업 재직자들에도 ‘소프트웨어 안전 인식 제고’ 과정 교육이 필요하다고 판단된다.

◆ 중소기업 재직자들의 교육과정의 시급성에 대한 응답 결과 또한 전체 응답 결과에서 시급성이 가장 높게 나온 ‘형상관리 및 결함관리’에 대한 응답이 상대적으로 낮게 나와 실무에서 형상관리 및 결함관리 등의 관리 관련 교육보다는 실질적인 개발과 관련된 교육이 좀 더 시급하다고 응답한 것을 알 수 있었다.

◆ 대기업 재직자들의 응답 결과는 전체 결과와는 다르게 기법에 대한 필요성보다는 설계 및 분석에 대한 필요성을 좀 더 높게 대답하여 소프트웨어 안전에 대한 지식이 좀 더 견고하다는 것을 알 수 있다. 그 외에는 이론 및 국제 표준 등에 대한

교육의 필요성이 낮게 나온 것이 전체 응답 결과와 비슷하였다.

- 대기업 재직자들의 교육과정의 시급성에 대한 응답 결과 전체 응답 결과와 큰 차이가 없었는데, ‘안전 품질 및 측정관리’에 대한 응답이 상대적으로 많이 낮게 나와 차이점을 보였으며, 이는 대기업 재직자들은 안전에 대한 지식을 다른 기업보다는 습득하고 있다고 생각하는 것으로 유추된다.

- 공공기관 재직자들의 응답 결과는 앞서 살펴본 중소기업 및 대기업 재직자들의 응답결과와는 다르게 국제 표준 및 규제, 개념 및 기초 이론에 대한 교육의 필요성 및 시급성이 높게 나와 차이점을 보였다. 반대로 소프트웨어 안전 분석 및 코드 안전 분석 등 분석과 관련된 교육의 필요성 및 시급성을 낮게 측정하였다. 그러나 발주 역량 강화를 위해서는 소프트웨어 안전 분석 등의 실무 역량 강화가 필요하리라 생각된다.

- 가중치로 판단해 볼 때 중소기업이나 공기업/공공기관보다는 중견기업/대기업에서 소프트웨어 안전교육이 더 필요하다고 응답한 것으로 확인되며, 이는 중견기업/대기업이 소프트웨어 안전에 대한 인식이 더 높다고 판단된다.

[문16] 실질적으로 귀 부서에서 필요로 하는 교육프로그램으로 정부에서 추가적으로 개발, 운영해야 할 교육프로그램은 무엇이라고 생각하십니까? (주관식 응답)

4) 기타 의견

- 위 설문에서 제시된 교육과정 외에 추가 의견으로 ‘SW 프로젝트 관리자 관련 자격증 중요성 증대’, ‘교육/컨설팅 과정이 너무 형식적임’, ‘소프트웨어 안전에 관한 많은 홍보가 시급함’, ‘테스트 위주의 교육과정 필요’, ‘SW 안전에 대한 백신 지원’, ‘안전관리에 대한 체계적인 시스템 필요’ 등의 소수 의견이 있었다.

- 기타 의견에서 유추되는 사실은 소프트웨어 안전에 대한 홍보가 부족하고, 아직 소프트웨어 안전 개념이 정립되지 않았다는 것이다.

- 이번 설문의 효과는 소프트웨어 안전에 대한 홍보도 한 축이라고 판단된다.

제3절 심층인터뷰

심층인터뷰는 주요 산업 분야에서 국내 소프트웨어 안전 기술을 선도하는 주요 업체들을 대상으로 심층적인 소프트웨어 안전 교육 수요조사를 위해 별도의 질문으로 진행하였다. 안전 필수 소프트웨어 사업현황, 기술역량, 인력역량 등을 기본적으로 파악한 후, IEC61508 표준에 의거한 관련 기술과 관리 방법의 실제 역량 확보 여부를 판단하는데 주안점을 두었다.

심층 인터뷰는는 원자력, 자동차, 항공, 철도, 의료기기 등 소프트웨어 안전과 관련된 분야의 주요 기업 21곳을 대상으로 진행하였으며, 다양한 분야의 주요 업체들과 골고루 인터뷰할 수 있도록 노력하였다. 대상 업체의 업종별 및 규모별 응답자 수는 다음과 같다.

<표 5-18> 업종별 및 규모별 응답자 수

업종	응답자 수	업종	응답자 수
원자력	3	항공	5
자동차/미래차	5	의료	2
휴대폰 제조사	2	철도/지하철	1
SW/IoT	1	기계/로봇	1
드론	1	**총합**	21

[그림5-37] 기업규모별 심층인터뷰 참여 기업 수

1. 소프트웨어 안전문화 및 환경, 경영 조사

[문2-1] 귀하가 속한 조직에서 기능안전(Functional Safety)에 대하여 범조직차원에서 이해/숙지하고 있습니까? 있다면 이를 위해 어떠한 활동(사내 세미나 등)이 정기적으로 진행되고 있습니까?

[조사 결과]

<표 5-19> 업종별 조직의 기능안전(functional Safety)에 대한 이해/숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	3	5	1	2	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력과 항공, 자동차 및 휴대폰 분야에서는 기능안전 관련 국제 표준이 존재하거나, 발주기관에서 기능안전에 관련된 요구사항이 있기 때문에 이에 대한 이해/숙지를 위해 주기적인 사내 교육 및 세미나가 진행되고 있다고 응답하였다.
- 반대로 기계/로봇, IoT, 철도 분야에서는 기능안전과 관련된 인식이 미흡하거나, 있더라도 기능안전의 이해/숙지를 위한 활동이 부족(비정기적이거나 다른 전공자가 공부하여 전파 등)하다고 응답하였다. - “인증기관을 통한 비정기적인 교육을 진행하고 있으나, 소프트웨어에 대한 전문성이 부족하다.” (의료분야 중소기업, 소프트웨어 엔지니어)

[문2-2] 귀하는 기능안전에 대해 충분히 이해하고 있습니까?

[조사 결과]

<표 5-20> 업종별 개인의 기능안전(Functional Safety)에 대한 이해/숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	5	5	0	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력과 항공, 자동차 및 철도 분야 종사자들은 모두 충분히 이해하고 있거나, 어느 정도는 이해하고 있다고 응답하였다.
- 기계/로봇, IoT, 드론, 의료기기 종사자들은 인증에 필요한 최소한의 이해 또는 이해하고 있지 않다고 응답하였다. 특히 휴대폰 분야의 업체에서는 기능안전의 범위 및 역할이 확대되고 복잡해지고 있어 충분한 이해가 힘들다고 응답하였다. - “계

속해서 기능안전의 범위와 역할이 확대되고 복잡해지고 있어 충분한 이해가 부족하다.” (휴대폰분야 대기업, 실무개발자)

[문2-3] 귀하가 속한 산업군의 기능안전 관련하여 안전 표준/규제가 존재합니까?

[조사 결과]

<표 5-21> 업종별 기능안전과 관련된 안전 표준/규제 존재 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	5	5	2	2	0	0	0	1
전체수	3	5	5	2	2	1	1	1	1

- 원자력 분야의 경우 IEEE7432, IEC, IAEA, ASME, KEPIC-QAP(전력산업기술기준 품질 보증) 등 관련된 방대한 표준이 존재한다고 응답하였다. - “원자력 관련하여 방대한 표준이 존재한다.” (원자력분야 대기업, 프로젝트 매니저)
- 자동차 분야의 경우 관련 표준으로 모두 ISO26262를 응답하였다.
- 의료 분야의 경우 관련 표준으로 IEC62304, ISO14971, IEC60601, IEC62366 등을 언급하였으며, 추가로 KFDA, FDA, CE 등 각국의 관리 규제를 같이 언급하였다.
- 그 외에 항공, 휴대폰, 드론 분야에서도 존재한다고 응답하였으며, 기계/로봇, IoT, 분야에서는 존재하지 않는다고 응답하였다. 철도 업체에서는 안전법의 형식승인기준에 일부 포함되어 있지만 전반적인 기능안전성에는 미치지 않는다고 응답하였다.

[문2-4] 귀 기관의 상급기관(발주기관, 감독기관)에서 사업 발주/감독 시, 기능안전에 관련하여 개발요건에서 명시하고 있습니까?

[조사 결과]

<표 5-22> 사업 발주/감독 시, 기능안전 관련 문항 명시 여부(업종별)

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	2	5	2	1	1	0	0	0
전체수	3	3	5	2	2	1	1	1	1

- 다음은 기능안전 관련 표준/규제의 존재 여부와 본 질문과의 관계를 나타낸 것이다.

<표 5-23> 기능안전 관련 표준/규제의 존재 여부(문2-3)와 개발 요건 명시 관계

문2-4 \ 문2-3	존재한다	존재하지 않는다
명시한다	13	1 (철도1)
명시하지 않는다	3 (항공1, 휴대폰1, 드론1)	2
해당사항 없음	2 (항공2)	-

- 대체적으로 문2-3에서 기능안전 관련 표준/규제가 존재한다고 응답한 원자력, 항공, 자동차, 의료, 휴대폰 등 분야의 업체에서는 개발요건에 명시하고 있다고 응답하였고, 존재하지 않는다고 응답한 IoT 및 로봇 분야의 업체는 개발요건에 명시하고 있지 않다고 응답하였다. 단, 일부 항공 업체의 경우 해당 기관이 최상위기관이기 때문에 해당사항이 없다고 답하였다.
- 드론 업체의 경우 기능안전 관련 표준이 존재하지만, 이에 대한 보완이 필요하며 개발요건에서 따로 명시하고 있지는 않다고 응답하였다. - “국토부에서 규제/법규에 대한 보완이 필요하다.” (드론분야 중소기업, 프로젝트 관리자)

[문2-5] 귀 기관에서는 국제 표준에 준하는 소프트웨어 개발 프로세스가 사내 표준으로 정립되어 있습니까? 정립되어 있다면 적절히 사용되고 있습니까?

[조사 결과]

<표 5-24> 국제 표준에 준하는 소프트웨어 개발 프로세스의 사내 표준 정립여부(업종별)

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	4	3	1	0	1	1	0	0
전체수	3	5	5	2	2	1	1	1	1

- 다음은 기능안전 관련 표준/규제의 존재 여부와 본 질문과의 관계를 나타낸 것이다.

<표 5-25> 기능안전 관련 표준/규제의 존재 여부(문2-3)와 개발프로세스 관계

문2-5 \ 문2-3	존재한다	존재하지 않는다
사용하고 있다.	12	1(IoT)
사용하지 않는다	6 (항공1, 자동차2, 의료1, 드론1)	2

- 문2-4와 같이 기능안전 관련 표준/규제의 존재 유무에 따라 사내 표준 역시 정립되고 있음을 알 수 있었다. 특히 원자력 분야의 경우 프로젝트 별로 표준이 정립되어 매우 철저히 사용하고 있다고 응답하였다. - “대형 프로젝트 별로 정립되어

있고, 매우 철저히 사용되고 있다.” (원자력분야 공공기관, 프로젝트/안전 관리자)

- 철도 관련 업체는 사내 표준으로 정립되어 있지는 않고 연구자 개인에 따라 정립 및 사용되고 있다고 응답하였고, 일부 자동차, 의료 업체의 경우 기능안전 관련 표준/규제가 존재하지만 사내 표준으로 직접 정립되어있지는 않다고 응답하였다. - “RAMS에 대한 요구사항이 포함되어 있다.” (철도분야 공공기관, 시스템 엔지니어)

[문2-6] 귀 기관에서는 전장부품/제어소프트웨어 개발 시 기능안전 관련 표준 프로세스가 정립되어 있습니까? 정립되어 있다면 잘 사용되고 있습니까?

[조사 결과]

<표 5-26> 전장부품/제어소프트웨어 개발 시 기능안전 표준 프로세스의 정립여부(업종별)

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	2	1	2	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 전장부품/제어소프트웨어 개발상의 기능안전 관련 표준 프로세스에 대한 응답은 앞서 다른 질문들에 대한 응답보다는 ‘그렇다’의 비중이 낮았다. 원자력, 항공, 자동차 분야의 업체에서도 몇몇에서는 표준 프로세스가 정립되어 있지 않거나, 정립되어 있지만 잘 사용하고 있지 않다고 응답하였다.
- 업계의 표준 프로세스가 존재하고, 그에 따라 사내 소프트웨어 개발 프로세스가 정립되어 있는 경우에도 기능안전에 대한 프로세스는 아직 상대적으로 미흡함을 살펴볼 수 있었다.

[문2-7] 귀 기관에서는 소프트웨어 개발 프로세스, 전장부품/제어소프트웨어 개발 시 기능안전 관련 표준 프로세스를 재정/개정/배포/관리하는 전담 부서가 존재합니까?

[조사 결과]

<표 5-27> 업종별 표준 프로세스 전담 부서 존재 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	2	2	2	1	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

◆ 다음은 문2-6에서 기능안전 관련 표준 프로세스의 정립에 대한 응답과 본 질문과의 관계를 나타낸 것이다.

<표 5-28> 기능안전 관련 표준 프로세스(문2-6)와 전담부서와의 관계

문2-7 \ 문2-6	존재한다	존재하지 않는다
존재한다	6	3 (원자력1, 항공1, 의료1)
존재하지 않는다	3 (항공2, 휴대폰1)	9

◆ 전담 부서의 존재는 대체로 기능안전 관련 사내 표준 프로세스의 유무와 관련 있음을 알 수 있었다. 사내 표준 프로세스가 존재하는 업체 대부분이 이와 관련된 전담 부서도 존재하고 있다고 응답하였는데, 일부 업체에서는 QA팀 또는 전략그룹을 전담 부서로 응답하기도 하였다. - “당사에는 품질 및 규제 관련 업무를 전담하는 팀(QA/RA팀)이 존재하며, 해당 팀에서 모든 프로세스 및 규제 관련 문서를 제정/개정/배포하고 관리 감독한다.” (의료분야 중소기업, 소프트웨어 엔지니어)

[문2-8] 귀 기관에서는 기능안전/소프트웨어 안전 분야 관리자(Safety Manager)가 존재합니까?

[조사 결과]

<표 5-29> 업종별 안전 분야 관리자 존재 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	2	1	1	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

◆ 다음은 문2-7에서 기능안전 관련 표준 프로세스 전담 부서의 존재에 대한 응답과 본 질문과의 관계를 나타낸 것이다.

<표 5-30> 기능안전 관련 표준 프로세스 전담 부서(문2-7)와 안전관리자 관계

문2-8 \ 문2-7	존재한다	존재하지 않는다
존재한다	6	2 (항공2)
존재하지 않는다	3 (원자력1, 항공1, 의료1)	10

◆ 기능안전 관련 표준 프로세스 전담 부서가 존재하는 업체는 대부분 안전 분야 관리자도 존재한다고 응답하였다. 앞서 기능안전 관련 표준 프로세스 전담 부서가

존재한다고 응답한 몇몇 원자력 업체나 의료 업체의 경우 QA관리자 또는 위험관리 업무 담당자는 존재하지만 전문적인 안전 분야 관리자는 존재하지 않는다고 응답하였고, 항공 분야의 업체의 경우 전담 부서는 존재하지 않지만 기능안전/소프트웨어 안전 관리자는 별도로 존재한다고 응답한 경우도 있었다. - “프로세스 전담 부서가 대체할 수 있다고 생각한다.” (항공분야 공공기관, 시스템 엔지니어)

[문2-9] 귀 기관에서는 기능안전/소프트웨어 안전에 대하여 관련 지식 필요 시, 지식 습득이 원활합니까? 원활하지 않을 경우 가장 중요한 원인이 무엇이라고 생각하십니까?

① 근로자가 필요한 역량/숙련을 갖추고 있어서
② 필요한 교육과정이 없어서
③ 교육비용이 많이 들어서
④ 업무부담 및 생산차질이 우려되어서
⑤ 업무난이도가 직업훈련이 필요하지 않아서
⑥ 교육효과에 대해서 회의적
⑦ 인근에 원하는 교육기관이 없어서
⑧ 교육에 대한 정보를 얻기 힘들어서
⑨ 기타

[조사 결과]

[그림5-38] SW안전 지식 습득이 원활하지 않은 원인

- 두 업체를 제외한 나머지 업체에서 기능안전/소프트웨어 안전에 관련된 지식 습득이 원활하지 않다고 응답하였다. 그 원인으로는 ② ‘필요한 교육과정이 없어서’와 ④ ‘업무부담 및 생산차질이 우려되어서’를 많이 선택하였으며, ⑦ ‘인근에 원하는 교육기관이 없어서’나 ⑧ ‘교육에 대한 정보를 얻기 힘들어서’ 등 교육 여건이 부족하다는 의견도 다수 있었다.

- 기타 의견으로는 ‘필요성을 인지하지 못하고 있었다’, ‘많은 직원들을 동시에 교육하기 어렵다’, ‘국내 기능안전 여건이 성숙하지 않아서’ 등이 있었다.

- 아래는 재직자 수요조사의 문5-10의 결과이다.

[그림5-39] 교육을 실시하지 않는 이유: 재직자 수요조사

- 공통적으로는 ‘필요한 교육과정이 없어서’, ‘교육에 대한 정보를 얻기가 힘들어서’, ‘업무부담 및 생산차질이 우려되어서’ 3가지 항목이 많은 응답을 차지한 것을 확인할 수 있다. 반면에 전문가 심층인터뷰에서는 재직자 수요조사에서 많은 비중을 차지했던 ‘업무난이도가 직업훈련이 필요하지 않아서’, ‘근로자가 필요한 역량/숙련을 갖추고 있어서’에 대한 응답이 상대적으로 낮게 나온 것을 확인할 수 있다. 이는 심층인터뷰에 응답한 전문가의 경우 관련업무가 소프트웨어 안전과 직접적인 연관이 있을 것이고, 따라서 소프트웨어 안전에 대한 필요성 및 중요성을 앞선 온라인 수요조사의 응답자보다 크게 느꼈기 때문이라고 유추해볼 수 있다.

[문2-10] 상기 2번 항목들 중 애로사항이 있다면 어떠한 것들이고, 정부의 지원 요청이 있다면 어떠한 것들이 있습니까? (예. 교육 지원, 교육 의무화, 자격증 제도화 등)

[조사 결과]

- ‘연차별 필수교육과정 제도화’, ‘법규 및 자격인증 강제’, ‘일정 교육시간 의무화’ 등 기능안전 관련 교육에 대한 제도적인 법제화가 필요하다는 의견이 많이 있었다.
- ‘산학연 전문가들과의 연계를 통한 교육 지원’, ‘전문 강사 양성’, ‘온라인 교육 지원’ 등 교육 지원과 관련한 의견도 상당수 있었다.

2. 기능안전 관리지식 수요조사

[문3-1] 프로젝트 수행 시 Safety Manager가 존재하여 PM과 함께 기능안전에 대한 분석/설계/구현/검증 계획에 참여합니까?

[조사 결과]

〈표 5-31〉 프로젝트 수행 시 안전 관리자 존재(업종별)

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	4	0	0	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력과 자동차, 그리고 일부 항공 분야의 업체에서는 Safety Manager가 존재하여 PM과 함께 기능안전에 대한 분석/설계/구현/검증 계획에 참여한다고 응답하였다. 그 외의 기계/로봇, IoT, 의료, 철도, 드론 등의 업체에서는 Safety Manager가 존재하지 않는다고 응답하였다.
- Safety Manager는 존재하지 않지만 기능안전에 대한 분석/설계/구현/검증 계획에 위험 관리 담당자가 일부 참여하고 있다고 응답한 업체도 있었다. - “별도의 Safety Manager가 없지만, 앞서 언급한 위험 관리 담당자가 일부 참여하는 과정은 존재한다.” (의료분야 중소기업, 소프트웨어 엔지니어)
- Safety Manager는 기능안전의 전반적인 책임을 지는 중요한 요소 중 하나임에도

불구하고 기능안전이 중요한 각 분야에서 Safety Manager를 별도로 두지 않고 PM에게 대부분의 책임을 넘기고 있어 실질적인 기능안전에 대한 인식이 부족함을 알 수 있다.

[문3-2] 프로젝트 계획과 별도로 기능안전계획(Functional Safety Plan)을 수립하는 방법을 알고 있습니까?

[조사 결과]

- Safety Manager가 존재한다고 응답한 업체들은 자동차 분야의 한 업체를 제외한 나머지 모든 업체에서 기능안전계획을 수립하는 방법을 알고 있다고 응답하였다. Safety Manager가 존재하지 않는다고 응답한 업체들은 대체로 기능안전계획을 수립하는 방법을 모른다고 하였지만, 항공, 의료, 자동차, 철도 등의 몇몇 업체에서는 Safety Manager는 없지만 기능안전계획을 수립하는 방법에 대해서는 어느 정도 알고 있다고 응답하였다.

- Safety Manager가 존재하지 않고 별도의 기능안전계획을 수립하는 방법은 모르지만 업무 처리를 위하여 외부 지원을 받아 진행중이라고 응답한 업체도 있었다. - “외부 지원 받아서 하고 있다.” (항공분야 중소기업, 프로젝트 관리자)

[문3-3] 제품에 대한 Safety Goal이 식별되고 이에 대한 Safety Case를 전개하는 방법을 알고 있습니까?

[조사 결과]

<표 5-32> 업종별 안전 목표 및 안전 케이스 식별

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	1	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 문3-2와 동일한 응답을 보였다. 다시 말해서 별도의 기능안전계획을 수립하는 방법을 알고 있는 경우 Safety Goal 식별 및 Safety Case를 전개하는 방법을 알고 있다고 응답하였다.

[문3-4] Safety Audit 활동 방법을 알고 있습니까? 알고 있다면 적용하고 계십니까?

[조사 결과]

<표 5-33> 업종별 Safety Audit 이해

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	0	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 문3-2, 문3-3과 거의 동일한 응답을 보였다. 의료 분야 한 곳에서는 별도의 기능안전계획을 수립하는 방법과 Safety Goal 식별 및 Safety Case를 전개하는 방법을 알고 있다고 응답하였지만 Safety Audit 활동 방법에 대해서는 모른다고 응답하였다.
- Safety Audit 활동 방법을 알고 있다고 응답한 몇몇 업체에서는 방법에 대해서는 알고 있지만 이제 막 시작한 단계이고 내재화가 필요하다고 응답하였다.

[문3-5] 외주 개발 발주의 경우 프로젝트 요구사항에 SIL 레벨 할당과 Safety Requirements를 적절히 기술하고 있습니까?

[조사 결과]

<표 5-34> 업종별 발주 시 SIL 레벨 할당과 Safety Requirements 기술 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	3	0	0	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- SIL 레벨 할당 및 Safety Requirements 기술에 대한 응답은 문3-1에서 응답한 Safety Manager의 존재에 따라 대부분 실시하고 있다고 응답하였다. 자동차 분야의 한 업체에서만 Safety Manager가 존재하지만 SIL 레벨 할당 및 Safety Requirement 기술을 하지 않는 것으로 밝혔다.
- 자동차 분야의 경우 ASIL이 잘 정의되어 있어 이를 적절히 따르고 있는 것을 알 수 있었다. - “ASIL에 따라 수행중이다.” (자동차/미래차분야 대기업, 품질 보증/안전 관리자)
- SIL은 Safety-Critical 기능들에 대한 위험의 정도를 측정하는 지표로써 기능안전의

중요한 요소 중 하나이지만, 아직 국내의 기능안전에 대한 인식 및 필요성이 부족한 현실을 결과에서 살펴볼 수 있었다.

[문3-6] 귀사가 개발 또는 사용하는 제품의 Safety Critical 필드 오류의 발생 원인이 하드웨어(전장회로)와 소프트웨어 간 어떠한 비중을 가지고 있습니까? 하드웨어 ( )%, 소프트웨어 ( )%, 기타 ( )%

[조사 결과]

- 원자력 업체의 경우 하드웨어와 소프트웨어의 오류 비율이 1:1이라고 응답하였다.
- 자동차 및 항공 업체의 경우 하드웨어와 소프트웨어의 오류 비율을 3:7 또는 4:6 정도로 응답하여 소프트웨어의 오류 비율이 조금 더 높다고 밝혔다. 다만, 위성체 관련 업체의 경우 실 결함이 극히 적어 응답하기 힘들다고 하였다. - “위성체는 실 결함이 극히 적다.” (항공분야 공공기관, 소프트웨어 엔지니어)
- 의료 업체의 경우 오류 발생 건수가 높지 않아 통계를 내기 어렵다고 하였다. - “아직까지 Safety Critical 오류 발생 건수가 높지 않아 통계를 낼 정도는 안 되며, 하드웨어 건수가 소프트웨어 건수보다 많다고 판단된다.” (의료분야 중소기업, 소프트웨어 엔지니어)
- 드론의 경우 하드웨어의 오류는 극히 적고 대부분 소프트웨어의 오류라고 응답하였다.
- 대부분의 분야에서 소프트웨어 오류의 비율이 하드웨어 오류의 비율보다 높았으며, 이는 곧 소프트웨어 안전의 중요성을 시사하는 점이라 볼 수 있다.

3. 기능안전 구현 기초지식 수요조사

[문4-1] 기존 제품의 변경 프로젝트의 경우 안전성 측면에서의 변경영향 분석 방법을 충분히 숙지하고 계십니까?

[조사 결과]

<표 5-35> 업종별 안전성 측면에서의 변경영향 분석 방법 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	0	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력, 자동차, 항공, 철도 업체에서는 안전성 측면에서의 변경영향 분석 방법을 알고 있다고 응답하였으며, 그 외의 대부분의 업체에서는 이에 대해 잘 숙지하고 있지 않다고 응답하였다.

- 변경영향 분석은 안전성 관점에서 특히 중요하다. 안전성이 중요한 주요 분야에서 위와 같이 변경영향 분석 방법에 대한 숙지 정도가 부족한 결과에서 기능안전에 대한 교육의 필요성이 절실함을 살펴볼 수 있다.

[문4-2] 개발 초기 단계에 제품 수준에서의 통계적인 Hazard Analysis와 Risk Assessment 방법을 충분히 알고 계십니까?

[조사 결과]

<표 5-36> 업종별 Hazard Analysis와 Risk Assessment 방법 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	5	1	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 문4-1과 마찬가지로 원자력, 자동차, 항공, 철도 업체 위주로 이에 대해 알고 있으며 어느 정도 수행하고 있다고 응답하였다.

- 자동차 분야의 경우 발주처에서 HARA(Hazard Analysis and Risk Analysis)에 대한 명시를 분명히 하고 있어 모든 업체가 이에 대한 방법을 알고 있음을 확인할 수 있다.

- 앞서 안정성 측면에서의 변경영향 분석 방법을 숙지하고 있지 않다고 밝힌 의료기기 업체 한 곳에서도 ISO14971에 명시된 것을 바탕으로 알고 있다고 응답하였다. - “ISO14971에서 요구하는 수준으로 이해하고 실행하고 있다.” (의료업체 중소기업, 소프트웨어 엔지니어)

[문4-3] SIL(Safety Integrity Level)을 알고 계십니까? 알고 계신다면 어떤 방법으로 부품/컴포넌트 별로 SIL을 정하고 계십니까?

[조사 결과]

<표 5-37> 업종별 SIL(Safety Integrity Level) 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	4	4	0	0	1	0	0	1
전체수	3	5	5	2	2	1	1	1	1

- 문4-1에서 안전성 측면에서의 변경영향 분석 방법을 숙지하고 있다고 밝힌 업체에서 대부분 SIL을 알고 있다고 응답하였다.
- 앞에서와 마찬가지로 변경영향 분석 방법, HARA, SIL 등 기능안전에 대한 기법은 현재 수행중이라고 응답한 업체가 일정함을 살펴볼 수 있다. 이는 기능안전에 대한 인식이 부족하거나 기능안전에 대한 교육이 부족하여 그 기법에 대한 숙지가 부족하기 때문이라고 추측해볼 수 있다. - “잘 알지 못하며, 적용하지 않고 있다.” (의료분야 중소기업, 소프트웨어 엔지니어)

[문4-4] 제품 SIL에 따른 부품별 SIL Decomposition 방법을 충분히 알고 계십니까? 알고 계시다면 실제 적용하고 계십니까?

[조사 결과]

<표 5-38> 업종별 부품 SIL Decomposition 방법 인식

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	1	1	1	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- SIL 레벨을 알고 있다고 응답한 다수의 업체에서도 부품별 SIL Decomposition 방법에 대해서는 모르고 있거나, 실제 적용하고 있지는 않다고 응답하였다.
- 부품별 SIL Decomposition에 대해서는 타사와 의견 충돌이 있다고 밝힌 응답도 있었다. - “이 부분에 대해서는 A사와 의견 충돌이 있다.” (원자력분야 대기업, 프로젝트 관리자)

[문4-5] 안전 목표에 따른 Safety Concept 도출 방법을 알고 계십니까?

[조사 결과]

<표 5-39> 업종별 Safety Concept 도출 방법 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	5	0	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- Safety Concept 도출 방법에 대한 응답은 문4-2에서 응답한 HARA 방법을 알고 있는가에 대한 응답과 비슷한 모습을 보였다.

[문4-6] 제품의 안전 요구사항을 하드웨어 컴포넌트, 소프트웨어 컴포넌트로 체계적으로 분할 할당하여 구현하고 있습니까?

[조사 결과]

<표 5-40> 안전 요구사항 체계적 분할 구현 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	5	1	2	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력, 항공, 자동차, 의료, 휴대폰, 철도 등 대부분의 분야에서 안전 요구사항을 하드웨어/소프트웨어로 분할 및 할당하여 구현하고 있다고 응답하였다. 하지만 IoT, 로봇 분야를 포함한 몇몇 업체에서는 요구사항이 형식적이거나 분할 및 할당이 체계적이지는 않다고 답하였다.

- 원자력 분야의 업체에서는 하드웨어 컴포넌트와 소프트웨어 컴포넌트뿐만 아니라 휴먼으로도 분류하여 할당한다고 밝혔다. - “Task Analysis에 의해 하드웨어, 소프트웨어, 휴먼으로 할당한다.” (원자력분야 공공기관, 프로젝트/안전 관리자)

- 드론 업체의 경우 일반 요구사항에 대해서는 하드웨어 컴포넌트와 소프트웨어 컴포넌트로 각각 분할 할당하여 구현하고 있지만 안전 요구사항에 대해서는 아니라고 밝혔다. - “요구사항에 대해서는 맞는데, 안전 요구사항은 아니다.” (드론분야 중소기업, 프로젝트 관리자)

[문4-7] 다양한 기능안전 요구사항에 대한 검증 방법들에 대하여 충분한 지식을 가지고 계십니까?

[조사 결과]

<표 5-41> 업종별 요구사항 검증 지식 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	0	0	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력, 자동차, 항공, 철도 업체 위주로 기능안전 요구사항에 대한 검증 방법들에 대하여 지식을 가지고 있다고 응답하였다. 자동차와 드론 분야의 몇몇 업체에서는 지식이 있으나 충분하지 않다고도 응답하였으며, 소프트웨어 안전 관련 인식이 부족한 업체에서는 Unit Test와 혼동하고 있는 답변도 살펴볼 수 있었다.

- 특히 자동차 업체의 경우 기능안전별 테스트케이스를 별도로 관리하고 있다고 밝혔다. - “기능안전 요구사항에 대한 검증 방법으로 기능안전별 테스트케이스를 별도로 관리하고 있다.” (자동차업체 대기업, 안전 관리자)

4. 기능안전 구현

[문5-1] 제품의 기능안전 요구사항을 반영하여 시스템의 안전 아키텍처를 설계하는 방법을 충분히 보유하고 계십니까? (System Modeling, Domain Reference Arch., etc…)

[조사 결과]

<표 5-42> 업종별 안전 아키텍처를 설계 방법 보유 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	3	0	0	0	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- 원자력, 자동차, 항공 업체 위주로 시스템의 안전 아키텍처 설계 방법을 보유하고 있다고 응답하였다. 철도와 의료기기, 기계 등의 업체에서는 어느 정도는 보유하고 있지만 미흡하거나, 기본적인 방법 정도만 알고 있는 정도라고 응답하였다. - “기본적인 수준에서 안전 아키텍처를 사용하고 있으나, 설계 방법론을 충분히 보유하

고 있지는 않다.” (의료분야 중소기업, 소프트웨어 엔지니어)

[문5-2] 시스템 안전 분석 시 어떠한 방법을 사용하고 계십니까? (FMEA, FTA)

[조사 결과]

<표 5-43> 업종별 안전 분석 사용여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	5	2	1	1	0	0	0
전체수	3	5	5	2	2	1	1	1	1

- IoT, 로봇, 드론 분야의 업체를 제외한 나머지 대부분의 기업이 시스템 안전 분석을 하고 있다고 응답하였다. 사용하고 있는 시스템 안전 분석 방법으로는 대부분 FMEA를 언급하였으며, 자동차 업체의 경우 FTA를 추가로 사용하고 있다고 하였다. 원자력 및 철도 업체에서는 FMEA, FTA 외에도 HAZOP, STPA 등의 방법도 사용하고 있는 것으로 밝혔다.
- 도메인에 상관없이 공통적으로 FMEA를 사용하고 있어 범 도메인 차원에서의 FMEA에 대한 교육을 생각해 볼 수 있다.

[문5-3] 시스템 분석 시 안전 메커니즘을 정의하는 담당자가 하드웨어(전자회로) 안전 메커니즘과 소프트웨어 안전 메커니즘에 대한 지식을 충분히 숙지하고 있습니까?

[조사 결과]

<표 5-44> 업종별 안전 메커니즘 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	3	4	0	1	1	0	1	0
전체수	3	5	5	2	2	1	1	1	1

- 자동차, 항공 분야 외에도 철도와 로봇 등의 분야에서 안전 메커니즘을 정의하는 담당자가 존재하여 하드웨어/소프트웨어 안전 메커니즘에 대한 지식을 숙지하고 있다고 응답하였다.
- 몇몇 분야의 업체에서는 전체를 완전히 이해하는 전문가가 부족하거나, 해당 담당

자의 지식 수준이 미흡하다고 응답하여 충분히 숙지하고 있지는 않은 것으로 밝혔다. - “전반적으로 미흡하며, 전체를 이해하는 전문가가 부족하다.” (원자력분야 공공기관, 프로젝트/안전 관리자)

[문5-4] 소프트웨어 담당자는 안전 메커니즘 구현시 하드웨어 엔지니어, 시스템 엔지니어와 지속적인 협의를 하고 있습니까?

[조사 결과]

<표 5-45> 업종별 안전 메커니즘 구현 시 협의 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	3	4	2	1	-	0	1	0
전체수	3	5	5	2	2	-	1	1	1

- IoT, 드론 분야의 업체를 제외한 나머지 대부분의 업체에서 지속적인 협의를 하고 있다고 답하였다. 하지만 지속적인 협의를 하고 있다고 밝힌 몇몇 업체에서는 발주기관의 담당자를 통한 간접 협의를 하고 있다고 밝히거나, 협의는 하고 있지만 충분하지는 않다고 하기도 하였다. - “발주기관의 담당자를 통하여 간접 협의하고, 다른 협력 업체와의 직접적 미팅은 원활하지는 않다.” (원자력분야 중소기업, 소프트웨어 엔지니어) / “하드웨어 및 소프트웨어 개발자들은 설계와 구현이 진행됨에 따라 위험을 줄이기 위한 협의를 지속적으로 하고 있으나, 충분하지는 않다.” (의료분야 중소기업, 소프트웨어 엔지니어)

[문5-5] 소프트웨어 담당자는 소프트웨어 안전분석 방법을 충분히 숙지하고 있습니까? (SW-FMEA, SW-FTA)

[조사 결과]

<표 5-46> 업종별 소프트웨어 안전분석 방법 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	3	4	1	0	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 원자력, 항공, 자동차, 의료 분야의 일부 업체에서 소프트웨어 안전분석 방법을 숙지하고 있다고 응답하였다.

◆ 자동차 분야의 업체가 소프트웨어 안전분석 방법을 숙지하고 있다고 응답한 비율이 상대적으로 높았지만, 일부만 숙지하고 있거나 막 시작하는 단계라고 응답하여 소프트웨어 안전분석 방법의 숙련도는 높지 않은 것으로 추측된다.

[문5-6] 소프트웨어 담당자는 소프트웨어 안전 메커니즘에 대한 구현 방법을 충분히 숙지하고 있습니까? (Diversity, Program Flow Monitering, Message Checking, etc…)

[조사 결과]

<표 5-47> 업종별 소프트웨어 안전 메커니즘에 대한 구현 방법 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	3	3	1	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 원자력, 항공, 자동차, 의료, 휴대폰 분야의 일부 업체에서 소프트웨어 안전 메커니즘에 대한 구현 방법을 숙지하고 있다고 응답하였다.

◆ 소프트웨어 안전 메커니즘에 대한 구현 방법은 알고 있지만 기본적인 방법만 알고 있어 충분하지 못하다는 응답이 다수 있어 실제 적용과는 아직 거리가 있을 것으로 생각된다. - “기본적인 방법만 알고 있고, 다양한 방법론에 대한 체계적인 지식을 갖추고 있지는 않다.” (의료분야 중소기업, 소프트웨어 엔지니어) / “어느 정도는 숙지하고 있지만, 더 많은 파악이 필요하다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문5-7] 주로 사용하는 소프트웨어 안전 메커니즘은 어떠한 것들입니까?

[조사 결과]

◆ 원자력 업체의 경우 Hazard Identification, Hazard Elimination, Safety Testing, Diversity Redundancy, Periodic Testing 등을 주로 사용하였다.

◆ 휴대폰 업체의 경우 설계 단계에서 Memory Protection, Fault Detection, Error Correction 등을 사용하고, 검증 단계에서 Abort Mode, System Error, Fault Containment 등을 사용하여 검증을 진행하였다.

◆ 그 외의 업체에서는 대부분 모니터링을 통한 소프트웨어 검증이나 CRC 체크, 논리적 신호판단 등을 사용하고 있다고 응답하였다.

[문5-8] 소프트웨어 개발 단계별(분석, 설계, 구현, 테스팅) 생명주기(폭포수, 반복적, Agile 등)를 정하고 이에 따라 구현하고 있습니까?

[조사 결과]

<표 5-48> 업종별 소프트웨어 생명주기 구현 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	5	3	2	2	-	1	0	1
전체수	3	5	5	2	2	-	1	1	1

◆ 자동차 분야의 일부 업체와 기계/로봇 분야의 업체를 제외한 대부분의 업체가 폭포수 모델이나 V-cycle을 사용하여 구현하고 있다고 응답하였다.

◆ 도메인에 상관없이 공통적으로 폭포수 모델과 V-cycle을 사용하고 있어 범 도메인 차원에서의 소프트웨어 생명주기와 소프트웨어/기능 안전 관련 교육을 생각해 볼 수 있다.

[문5-9] 소프트웨어 개발 각 단계(분석, 설계, 구현, 테스팅)에서 수행하는 기능 안전에 대한 활동이 별도로 정의되어 사용하고 있습니까?

[조사 결과]

<표 5-49> 업종별 기능 안전에 대한 활동 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	2	0	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 원자력, 항공, 자동차, 휴대폰 분야의 일부 업체에서 소프트웨어 개발 단계별로 기능 안전에 대한 활동을 각각 정의하여 사용하고 있다고 응답하였다. 특히 자동차 분야의 경우 A-spice 모델 및 ISO26262 표준에 잘 정의가 되어 있어서 이를 참고하여 정의하고 있다고 밝혔다.

◆ 현재 소프트웨어 개발 단계별 기능 안전에 대한 활동을 별도로 정의하여 사용하고

있지 않다고 밝힌 자동차 업체에서도 모두 이와 관련된 프로세스를 정립중이거나 구축중인 것으로 밝혔다. 이는 자동차 분야의 국제 표준인 ISO26262에 이와 관련된 기능안전 프로세스가 정의되어 있고, 해외 OEM에서 이러한 프로세스에 대한 요구가 있기 때문인 것으로 생각된다.

[문5-10] 소프트웨어 개발 단계별 기능안전과 관련된 도구/개발환경 등을 구축하고 사용하고 계십니까?

[조사 결과]

<표 5-50> 업종별 기능 안전 도구 사용여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	3	0	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 원자력 업체에서는 Statechart와 UML, SCADE, NuSCR 등의 도구를 사용하고 있다고 응답하였다.
- 그 외의 업체에서는 대부분 별도의 도구를 사용하고 있지 않거나, 현재 도입을 추진 중이라고 밝혔다.

[문5-11] 정형화된 소프트웨어 모델링 기법(SysML, UML, Statechart 등)을 사용하고 있습니까?

[조사 결과]

<표 5-51> 업종별 소프트웨어 모델링 기법을 사용 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	3	2	2	-	1	0	1
전체수	3	5	5	2	2	-	1	1	1

- 많은 업체에서 UML과 Statechart를 사용하고 있다고 응답하였다.
- 원자력 업체에서는 Scade, Statemate 등을 사용한다고 밝혔고, 한 자동차 업체에서는 Simulink를 모델링에도 사용한다고 답하였다.
- 시스템 소프트웨어를 개발하는 데에는 정형화된 모델링 기법이 적합하지 않다는

의견도 있었다. - “시스템 소프트웨어를 개발하는 데 적합하지 않다.” (항공분야 공공기관, 시스템 엔지니어)

- 정형화된 소프트웨어 모델링 기법으로 널리 사용되고 있는 UML 및 Statechart와 소프트웨어/기능 안전과 관련된 교육이 도메인과 관계없이 도움이 될 것으로 생각된다.

[문5-12] 소프트웨어 Formal Modeling, Formal Verification 기법들을 알고 있습니까?

[조사 결과]

<표 5-52> 업종별 정형검증 숙지 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	4	2	0	1	-	0	1	1
전체수	3	5	5	2	2	-	1	1	1

- 원자력, 항공, 자동차, 휴대폰 분야의 일부 업체에서 해당 기법들에 대하여 알고 있다고 응답하였다. 기계/로봇 및 드론 분야의 업체에서도 위 기법들에 대해서는 알고 있다고 답하였다.

- 위 기법들에 대하여 알고 있다고 응답한 업체에서도 현재 실무에서 적용하고 있지 않거나, 잘 활용하고 있지는 않다고 밝혔다. - “기법들을 알고 있지만 장점을 최대한 활용하고 있지는 못하다.” (원자력분야 대기업, 프로젝트 관리자)

[문5-13] 프로그래밍 언어 선정 시 런타임 에러 핸들링 등에 대한 런타임 잠재 오류에 대한 고려가 이루어지고 있습니까? (C 이외의 다른 언어 사용시)

[조사 결과]

[그림5-40] 런타임 잠재 오류에 대한 고려 여부

<표 5-53> 업종별 런타임 잠재 오류에 대한 고려 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	1	3	1	1	-	1	0	0
전체수	2(1)	1(4)	5	2	2	-	1	1	1

(C언어만 사용 시 ( ) 안에 해당, 전체 업체 수에서 제외함)

- 원자력 및 항공 분야에서는 C언어를 주로 사용하고 있어서 본 질문에 대한 답변으로 해당사항 없다는 의견이 주를 이루었고, 그 외의 경우에는 자동차, 의료, 휴대폰 등의 분야에서 다수의 업체가 런타임 잠재 오류에 대한 고려가 이루어지고 있다고 응답하였다.

- 몇몇 업체에서는 제한적인 경우에만 고려하거나, 컴파일러의 표준 준수 여부 정도만 고려한다고 밝혔다. - “제어 SW는 gcc, Windows 기반은 Visual Studio를 사용하는 등 표준을 따르는 컴파일러냐 아니냐의 정도만을 따른다.” (드론분야 중소기업, 프로젝트 관리자)

[문5-14] 소프트웨어 기능안전 요구사항에 대한 명세 활동이 진행됩니까?

[조사 결과]

<표 5-54> 업종별 소프트웨어 기능안전 요구사항 명세 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	4	4	2	1	-	0	0	1
전체수	3	5	5	2	2	-	1	1	1

- 항공과 자동차 각각 한 군데, IoT 및 로봇 분야의 업체를 제외하고, 대부분의 업체에서 기능안전 요구사항에 대한 명세 활동이 진행된다고 응답하였다. 일부 업체에서는 기능안전은 아니지만 테스트 케이스나 검증 절차로 분류하는 활동으로 대체하고 있다고 밝혔다.

[문5-15] 소프트웨어 아키텍처 단계의 기능 안전성 분석이 이루어지고 있습니까? (설계단계 SW-FMEA, SW-FTA)

[조사 결과]

<표 5-55> 업종별 소프트웨어 아키텍처 단계의 기능 안전성 분석 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	2	4	1	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 설계 단계에서 SW-FMEA 또는 SW-FTA와 관련된 기능 안전성 분석에 대해서는 원자력, 항공, 자동차, 의료, 휴대폰 분야의 일부 업체에서 하고 있다고 응답하였다. 그 외의 업체에서는 현재 진행 예정이거나 진행하지 않고 있다고 밝혔다.

- 특히 휴대폰 관련 업체의 경우 COTS 활용과 관련하여 중요한 고려 요소이기 때문에 기능 안전성 분석이 잘 이루어지고 있다고 응답하였다. - “COTS를 활용하는 경우 COTS 공급사에서 설계 시 고려하고 있고, 자사 솔루션의 경우 별도로 기능 안전을 설계 단계부터 고려하고 있다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문5-16] 소프트웨어 구현 단계에서 기능 안전성 분석이 이루어지고 있습니까? (구현단계 SW-FMEA, SW-FTA)

[조사 결과]

<표 5-56> 업종별 소프트웨어 구현 단계에서 기능 안전성 분석 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	2	4	1	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 구현 단계에서 SW-FMEA 또는 SW-FTA와 관련된 기능 안전성 분석에 대해서는 문5-15와 동일한 답변을 보였다.

[문5-17] 재사용, 변경되는 소프트웨어 모듈의 경우 어떠한 방법으로 기능안전을 확보하십니까? (Component Qualification, Prove-in-use)

[조사 결과]

<표 5-58> 업종별 재사용, 변경되는 소프트웨어 기능 안전 확보 방법 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	0	2	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 원자력, 항공, 자동차, 휴대폰 분야의 업체에서 재사용 및 변경 모듈에 대한 기능 안전을 확보하고 있다고 응답하였다. 그 방법으로는 주로 Regression Test, Prove-in-use, Component Qualification 등을 사용한다고 밝혔다.

◆ 그 외의 업체에서는 컴포넌트 재사용이 없거나 단순한 변경 관리 정도만 하고 있다고 밝혔다.

[문5-18] 소프트웨어 개발 단계별로 기능안전 수준에 따른 검증 활동/리뷰가 이루어지고 있습니까?

[조사 결과]

<표 5-59> 업종별 소프트웨어 개발 단계별 검증 활동 및 리뷰 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	4	2	0	2	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 원자력, 항공, 휴대폰 분야와 자동차 분야의 일부 업체에서 단계별 검증 활동 및 리뷰가 이루어지고 있다고 응답하였다. 특히 휴대폰 업체에서는 기능안전 수준에 따른 검증이 구현단계에서 매우 중요한 고려 요소라고 답하였다. - “기능 안전성 분석은 구현 단계에서 매우 중요한 고려 요소이다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

◆ 그 외의 대부분의 업체에서는 단계별 검증 자체가 진행되고 있지 않거나, 현재 개선중이라고 밝혔다.

[문5-19] 소프트웨어 테스팅 단계에서 기능안전에 대한 특화된 검증 기법을 충분히 확보하고 있습니까? (Fault-injection Testing)

[조사 결과]

<표 5-60> 업종별 테스팅 단계 검증 기법 확보 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	2	2	0	2	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 자동차 업체에서는 ISO26262의 요구사항에 따라 HILS를 사용하는 등의 기능안전 검증 기법을 확보하고 있다고 한 업체도 있지만, 본 질문에 응답한 많은 업체들이 특화된 검증 기법을 확보하고 있지는 않다고 응답하였다.

[문5-20] 소프트웨어의 실시간성 분석, 리소스 충돌 분석, 통신 자원 분석 등이 이루어지고 있습니까? (Worst-case Execution, Time Analysis, etc…)

[조사 결과]

<표 5-61> 업종별 소프트웨어의 실시간성 분석, 리소스 충돌 분석, 통신 자원 분석 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	4	1	2	-	0	1	1
전체수	3	5	5	2	2	-	1	1	1

◆ 자동차와 항공 업체의 경우 상용화를 하기 위한 필수 요건이기 때문에 소프트웨어의 실시간성 분석이나 리소스 충돌 분석, 통신 자원 분석 등이 이루어지고 있다고 응답하였다. - “하지 않으면 상용화가 안되기 때문에 기능안전의 대부분이 기존에 해오던 활동이었다.” (자동차분야 대기업, 안전관리자)

◆ 그 외의 분야에서는 원자력, 의료, 휴대폰 업체에서 어느 정도 위 사항들에 대한 분석이 이루어지고 있고, 나머지 분야의 업체에서는 해당 분석들이 현재 이루어지고 있지 않다고 밝혔다.

[문5-21] 소프트웨어 일반 기능 외 기능안전 요구사항에 대한 추적활동이 이루어지고 있습니까?

[조사 결과]

<표 5-62> 업종별 기능안전 요구사항에 대한 추적활동 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	3	3	1	1	-	0	1	0
전체수	3	5	5	2	2	-	1	1	1

◆ 기능안전 요구사항에 대한 추적활동에 대해서는 원자력, 항공, 자동차 업체에서 주로 이루어지고 있다고 응답하였다.

◆ 대부분의 업체에서는 형식적인 수준에서 일반 요구사항에 대한 추적활동이 이루어

지고 있다고 밝혔고, 로그 기록 정도만 한다는 업체도 있는 반면에, 다양한 한계 시험이나 예외 케이스 검증에 대한 추적활동이 철저히 이루어지고 있다고 응답한 업체도 있었다. - “형식적인 수준에서 추적하며, SRS에 Risk 관련 항목을 별도의 Section에 기록하고 있다.” (의료분야 중소기업, 소프트웨어 엔지니어) / “다양한 한계 시험이나 예외 케이스 검증을 통하여 추가 기능안전 요구사항을 확인하고 있다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문5-22] 소프트웨어 단위 시험/통합 시험/시스템 시험이 구분되어 이루어지고 있습니까?

[조사 결과]

- 대부분의 업체에서 단위시험, 통합시험, 시스템시험을 각각 구분하여 수행하고 있다고 응답하였다. 하지만 항공이나 자동차 분야의 몇몇 업체에서는 구분하고 있지 않거나, 구분할 예정이라고 밝혔다.

[문5-23] 소프트웨어 단위 시험/통합 시험/시스템 시험 시 기능안전 요건에 대한 시험이 이루어지고 있습니까?

[조사 결과]

<표 5-63> 업종별 소프트웨어 시험 시 기능안전 시험 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	3	2	2	1	-	0	1	0
전체수	3	5	5	2	2	-	1	1	1

- 앞서 단위시험, 통합시험, 시스템시험을 각각 구분하여 수행하고 있다고 응답한 업체들 대부분이 각각의 테스트에서 기능안전 요건에 대한 시험이 이루어지고 있다고 응답한 반면, 원자력이나 자동차 분야의 일부 업체에서는 기능안전 요건에 대한 시험은 이루어지지 않는다고 응답하였다.

[문5-24] 소프트웨어에 대한 정적 테스팅을 하고 계십니까?

[조사 결과]

<표 5-64> 업종별 소프트웨어 정적 테스팅 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	4	4	2	2	-	0	1	0
전체수	3	5	5	2	2	-	1	1	1

- 형식적인 테스팅만 하고 있다고 밝힌 몇몇 업체들을 제외한 대부분의 업체에서 정적 테스팅을 하고 있다고 응답하였다.

[문5-25] 소프트웨어 시스템 시험 시 목표 하드웨어 상에서 충분히 시험되고 있습니까? 환경 상 어려우면 이유가 무엇입니까?

[조사 결과]

<표 5-65> 업종별 하드웨어 상 시험 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	5	5	2	2	-	0	1	0
전체수	3	5	5	2	2	-	1	1	1

- 목표 하드웨어 상에서의 소프트웨어 시스템 시험은 제품 상용화의 중요한 요소이기 때문에 IoT와 드론 업체를 제외한 나머지 모든 업체에서 충분히 시험되고 있다고 응답하였다.
- IoT 관련 업체는 하드웨어가 부족하여 테스팅이 힘들다고 밝혔고, 드론 업체의 경우 하드웨어 제작이 늦어질 경우 충분한 시험이 어렵다고 밝혔다.

[문5-26] 소프트웨어 개발에 사용되는 도구(사용 및 내부 개발)들에 대한 안전성 검증이 이루어지고 있습니까?

[조사 결과]

<표 5-66> 업종별 도구 안전성 검증 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	3	4	3	0	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 소프트웨어 개발에 사용되는 도구들에 대한 안전성 검증은 원자력, 항공, 자동차 분야의 업체에서 이루어지고 있다고 응답하였다. 하지만 도구 검증 방법에 대해서는 상위 기관(발주처)에서 검증을 한다든지, 담당 부서에서 할 것이라고 응답하는 등 모호한 응답들이 다수 있었다. - “여기서 하고 있지는 않다. XX에서 하고 있다.” (원자력분야 중소기업, 소프트웨어 품질보증) / “개발 툴에 대한 안전성 검증은 개발팀에서는 진행하지 않지만, 담당팀에서는 검토할 것으로 생각된다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

5. 양산/구축, 이관 및 운영 조사

[문6-1] 양산/구축 과정 중 제품 요구사항 변경이 발생할 경우 기능안전 관점에서 요구사항에 대한 변경 영향 분석과 관리가 이루어지고 있습니까?

[조사 결과]

<표 5-67> 업종별 요구사항에 대한 변경 영향 분석과 관리 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	2	3	1	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

◆ 양산/구축 과정 중에서의 변경 영향 분석 및 관리에 대해서는 원자력과 항공, 자동차 등의 업체에서 일부 이루어지고 있다고 응답하였다. 일부 업체들은 필요하나 아직 이루어지지 않고 있어 이에 대한 대책을 계획중이라고 답하였고, 몇몇 업체는 아직 양산/구축 과정 전이라고 밝혔다.

◆ 휴대폰 분야의 한 업체에서는 모든 과정에서 기능안전 관점에서의 영향분석 및 관리가 이루어지고 있다고 답하였다. - “요구사항 변경이 발생하면 어느 과정에서는 기능안전 관점에서의 영향분석 및 관리가 이루어지고 있다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문6-2] 운영자들을 위한 Safety Manual이 개발되고, 교육되고 있습니까?

[조사 결과]

<표 5-68> 업종별 안전 매뉴얼 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	1	3	2	1	1	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 운영자들을 위한 Safety Manual에 대해서는 원자력과 항공, 자동차 등의 일부 업체에서 개발 및 교육되고 있다고 응답하였다. 의료 및 드론 분야의 업체에서는 운영자들을 위한 안전 교육은 진행되고 있지만 실제 Safety Manual은 존재하지 않는다고 밝혔고, 반대로 휴대폰 업체에서는 Safety Manual의 역할을 하는 Checklist가 있을 것으로 생각되지만 직접 관련 교육을 받지는 못했다고 답하였다. - "양산 단계에서 기능안전에 대한 Checklist가 존재할 것으로 생각된다. 직접 관련 교육을 받지는 못하였다." (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문6-3] 필드 결함 발생 시, Defect Analysis 활동이 충분히 이루어지고 있습니까? 이루어지고 있다면 어떠한 Defect 분류체계를 가지고 있습니까?

[조사 결과]

<표 5-69> 업종별 결함 분석 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	0	2	2	2	2	-	1	0	0
전체수	3	5	5	2	2	-	1	1	1

- 의료 및 휴대폰, IoT 업체와 항공, 자동차 분야의 일부 업체에서 결함에 대한 분석 활동이 충분히 이루어지고 있다고 응답하였다.

- 분류체계에 대해서는 의료 업체의 경우 결함에 대한 분석 활동은 이루어지고 있으나 그에 대한 분류체계가 정형화되어 있지는 않다고 답하였고, 휴대폰 및 IoT 업체에서는 발생한 결함의 심각도에 따라 분류한다고 하였다. - "상세한 분석 활동을 수행하지만, Formal한 분류 체계나 체계화된 분석 시스템을 갖추고 있지는 않다." (의료분야 중소기업, 소프트웨어 엔지니어) / "필드에서 발생하는 거의 모든 Defect에 대해서는 Report되는대로 분석이 진행된다. 결함이 사용자나 운영 인프라에 미치는 심각도에 따라 분류되며, 기능안전 관련 결함은 심각도 중 上에 해당하

게 된다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

[문6-4] 동일 결함 재발 방지를 위한 변경관리, 유사 파생 제품의 변경 파급 관리가 적절히 이루어지고 있습니까?

[조사 결과]

<표 5-70> 업종별 유사 파생 제품의 변경 파급 관리 여부

업종	원자력	항공	자동차	의료	휴대폰	철도	IoT	로봇	드론
긍정	2	2	3	2	2	-	0	0	0
전체수	3	5	5	2	2	-	1	1	1

- 원자력과 의료, 휴대폰 업체, 그리고 항공과 자동차 분야의 일부 업체에서 변경관리 및 파급 관리가 적절히 이루어지고 있다고 응답하였다.
- 관리가 이루어지고 있다고 응답한 업체 중에선 원인, 분석, 해결방안에 대한 수평 전개가 이루어지고 있으나, 해당 관리가 적절한 수준인지에 대한 판단은 어렵다는 의견이 몇몇 있었고, 단순히 변경 및 파급 관리에 대한 절차만 마련된 업체도 있었다. - “수평전개를 행하고 있으나, ‘적절히’의 수준은 측정하기 어렵다.” (자동차분야 대기업, 안전 관리자) / “이슈 상황의 원인, 분석, 해결방안 등에 대한 수평 전개를 하고 있으며, 신규 프로젝트에서 재발하지 않도록 하는 시스템도 확보하고 있다.” (휴대폰분야 대기업, 소프트웨어 엔지니어)

6. 교육과정에 대한 일반 건의

[문7-1] Safety Critical SW의 기능안전교육이 필요하다고 생각하십니까? 필요하다면 그렇게 생각한 이유가 무엇입니까?

① 소프트웨어 안전 사고 방지를 위하여
② 발주자의 요청 때문에
③ 안전관련 법/제도/표준의 준수요건 때문에
④ 내부 역량 확보 차원에서
⑤ 수출 요건 때문에

[조사 결과]

[그림5-41] SW안전 교육의 필요 원인

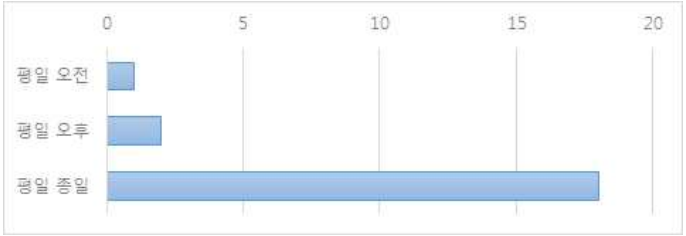
◆ 기능안전교육이 필요한 이유로는 ‘소프트웨어 안전 사고 방지를 위하여’를 가장 많이 꼽았다. 중복 응답이 가능하여 대부분의 응답자가 기능안전교육의 전제가 될 수 있는 소프트웨어 안전 사고 방지를 선택한 것으로 보인다. 그 외에 ‘내부 역량 확보 차원에서’를 선택한 업체도 8곳 있었으며, 각 업체별 요구사항이 될 수 있는 ‘안전관련 법/제도/표준의 준수요건 때문에’, ‘발주자의 요청 때문에’, ‘수출 요건 때문에’ 등의 응답도 몇몇 확인할 수 있었다.

[문7-2] 적절한 교육 시간은 어느 정도라고 생각하십니까?

(1) 교육 참여 가능 시간대	(2) 적정 교육시간
① 평일 오전 ② 평일 오후 ③ 평일 종일 ④ 주말 ⑤ 기타	① 1일 ② 2~3일 ③ 4~5일 ④ 6~10일 ⑤ 기타

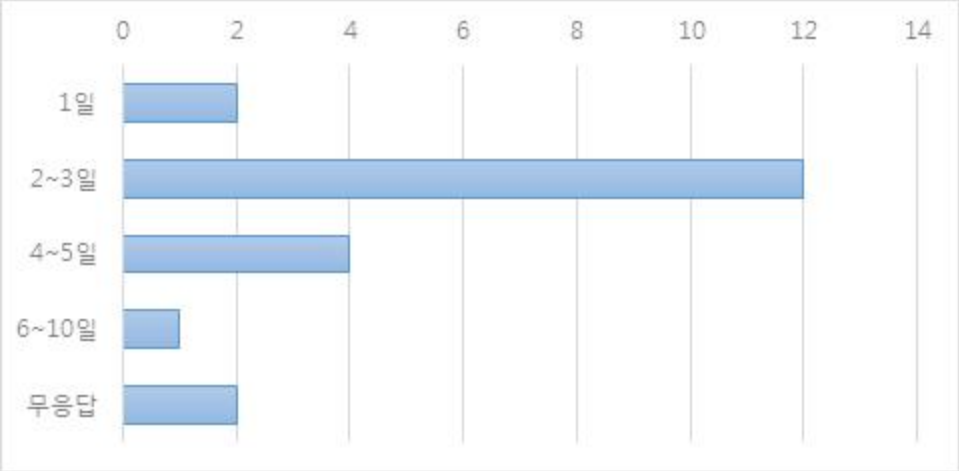
[조사 결과]

[그림5-42] 참여 가능한 교육 시간대



◆ 교육 참여 가능 시간대에 대한 응답으로는 업무 여건상 평일 종일을 택할 수 없는 몇몇 업체를 제외하고는 대부분 평일 종일을 선택하였다. 주말을 선택한 업체는 없는 것으로 보아 대부분의 업체가 평일 교육을 선호하는 것으로 알 수 있었다.

[그림5-43] 적정 교육기간



◆ 적정 교육기간에 대한 응답으로는 대부분의 업체가 2~3일을 응답하였다. 업무 여건상 며칠을 연속으로 교육할 수 없는 업체에서는 1일을 선택하기도 하였고, 1주 이상의 긴 기간을 투자하여 교육하여야 한다고 응답한 업체도 있었다.

[문7-3] 적절한 교육 시기는 언제라고 생각하십니까? 그리고 그 이유는 무엇입니까?

[조사 결과]

◆ 적절한 교육 시기로는 모든 응답자가 일이 바쁘지 않을 때, 또는 큰 프로젝트가 시작하기 전으로 응답하였다. 그 시기에 대해서는 업체별로 업무가 주로 진행되는 시기가 달라서 큰 공통점을 찾기가 어려웠다.

◆ 적절한 교육 시기가 업체별로 다 다르기 때문에 현재 국내의 교육환경으로는 교육에 대한 수요를 충족하기가 힘들 것으로 생각된다. 따라서 정부 주도의 범 도메인 차원의 교육이 지속적으로 실시되어 이러한 수요를 충족시켜줄 필요가 있을 것이다.

[문7-4] 교육에서 가장 중요하게 다루어야 할 과목이 무엇이라고 생각하십니까?

① SW 안전관련 개념/최신 기술 동향
② 시스템 안전사고에 대한 잠재 위험 분석
③ SW 안전 분석, 설계, 검증 관련 기술
④ 안전사고 예방을 위한 운용/사용
⑤ 소프트웨어 안전 관련 표준(ISO, IEC 등)
⑥ 발주자/표준 요건 대응 및 기술 영업
⑦ 소프트웨어 안전 관련 인식제고
⑧ 체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)
⑨ 기타

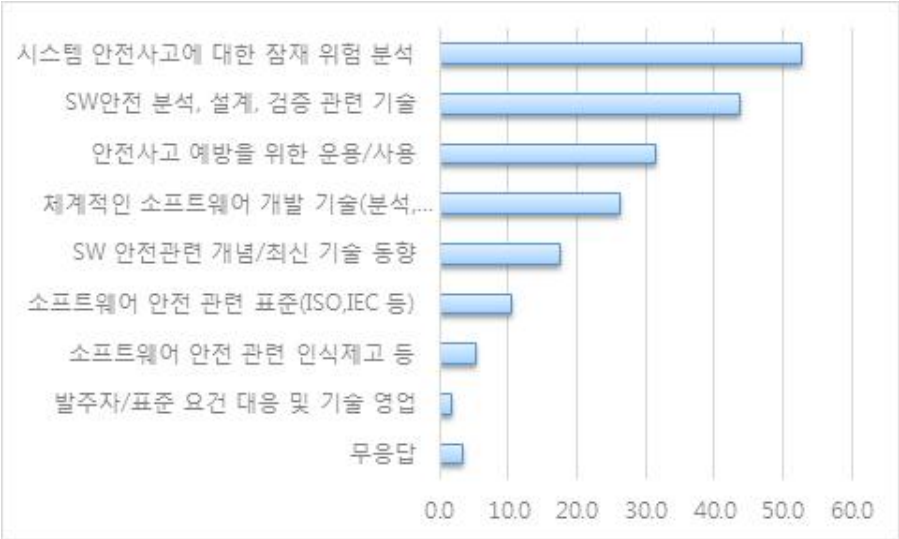
[조사 결과]

[그림5-44] 교육에서 가장 중요하게 다루어야 할 부분

◆ 중요하게 다루어야 할 과목으로는 'SW 안전 분석, 설계, 검증 관련 기술'에 대한 응답이 가장 많았으며, '체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)'이 그 다음으로 많았다. 주로 최신 동향이나 인식제고, 표준 등에 관련된 교육보다는 실제 업무에서 사용되는 기술과 관련된 교육을 더욱 중요하게 생각하는 것으로 보인다.

◆ 아래는 재직자 수요조사의 문5-3의 결과이다.

[그림5-45] 교육시 중요하게 생각하는 내용: 재직자 수요조사



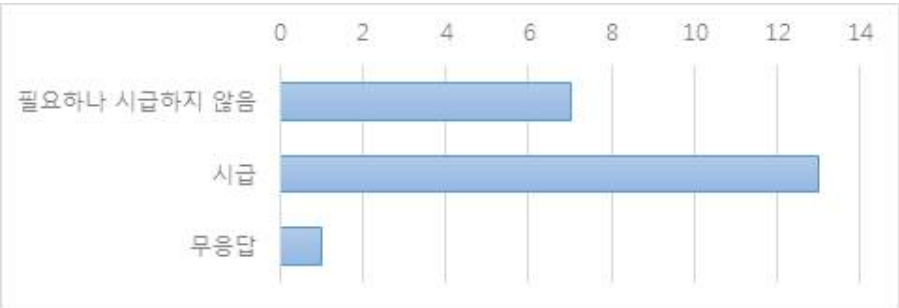
◆ 공통적으로는 ‘SW안전 분석, 설계, 검증 관련 기술’에 대한 중요도가 높았고, ‘발주자/표준 요건 대응 및 기술 영업’에 대한 중요도가 낮았다. 반면에 앞에서는 가장 중요하다고 생각되었던 ‘시스템 안전사고에 대한 잠재 위험 분석’에 대한 중요도가 중간 정도를 나타내었고, ‘안전사고 예방을 위한 운용/사용’에 대한 중요도는 앞선 온라인 수요조사 결과에 비해 상당히 낮은 결과를 보였다.

[문7-5] 상기 교육의 필요 부분에 대한 시급성은 어떠합니까?

필요하나 시급하지 않음 ( )	시급 ( )	매우 시급 ( )

[조사 결과]

[그림5-46] 교육의 시급성



◆ 기능안전교육에 대한 시급성으로는 시급하다는 응답이 필요하지만 시급하지 않다는 응답의 두 배 정도를 차지하였다.

[문7-6] 상기 교육이 귀하가 개발/운영하는 제품의 인허가 요건상에서도 필요합니까?

[조사 결과]

[그림5-47] 제품의 인허가 요건상 교육의 필요 여부

◆ 실제 인허가 요건상에서 교육의 필요성에 대한 응답으로는 필요하다는 응답보다 필요하지 않다는 응답이 조금 더 많은 비중을 차지하였다.

[문7-7] 기능안전 교육의 진행 방식이 어떤 형태로 운영되길 원하십니까?

① 사내 방문 교육 ② 외부 교육기관 파견 ③ 외부 단기교육
④ 워크샵 및 세미나 ⑤ 작업순환 ⑥ 기타

[조사 결과]

[그림5-48] SW안전 교육의 선호 진행방식

◆ 기능안전교육의 진행방식으로는 외부 단기교육과 사내 방문 교육을 많이 선호하는 것으로 나타났다. 외부 교육기관 파견은 많은 시간이 소모되어 업무진행에 영향을 줄 수 있어 다른 방식에 비해서는 조금 덜 선호되는 것으로 보였다. 그 외에도 워크샵 및 세미나, 작업순환 등과 같이 외부 강사 없이 진행되는 교육에 대해서도 사내 방문 교육이나 외부 단기교육 등에 비해서는 선호도가 떨어지는 것으로 나타났다.

제4절 수요조사 결과 분석

설문조사는 소프트웨어 안전성 확보가 중요하고 소프트웨어 중요성이 큰 항공, 자동차, 철도, 에너지, 의료 등 각 산업군에서 대기업/중견기업, 중소기업, 공공기관을 대상으로 관리자, 실무개발자 대상으로 시행하였다. 안전에 대한 인식 조사에서 테스팅이나 코딩 표준 준수가 소프트웨어 안전성 확보에 가장 중요한 요소라고 보는 기업이 상당히 많아, 소프트웨어 안전성에 대한 기술적 인식이 미흡하다는 사실이 유추되었다.

교육이 필요한 이유로는 소프트웨어 안전사고 방지 및 안전 관련 표준 준수 요건으로 응답하여 교육 수요는 규제나 외적인 요인 보다는 제품 자체의 안전 중요도 인식에서 발생한다고 판단된다.

기업 규모에 따라 사내방문교육, 외부 단기교육, 세미나 등 선호도가 달라 기업 규모, 비용, 근무환경에 따른 맞춤교육이 필요하다고 분석된다. 또한 대학 교육과 기업체와의 연관성이 부족하여, 대학 교육과정에서 재직자를 위한 교육 체계를 마련하는 것이 필요하리라 본다.

교육 내용에 있어서는 실무에 관련된 내용이 이론보다 선호도가 높아 교육이 필요한 이유인 제품 자체의 안전사고 방지를 위한 실질적인 기술을 원하는 것과 일맥상통한다. 그러나 안전사고 방지를 위해서는 실무도 중요하나, 소프트웨어 안전 인식 제고 및 안전 개념 학습이 우선적으로 필요하다는 사실을 기업에 홍보하는 것도 교육 정책 마련 시 고려해야 한다.

교육 과정 선정 시 가장 중요하게 생각하는 것은 비용이나 교육 기관 등 외적인 요인보다는 교육 적합성과 내용을 중요 시 여기는 것으로 조사되어 교육 정책 마련 시 비용절감 등의 방안보다는 소프트웨어 안전 교육 품질을 높이는데 목표를 두어야 한다고 생각된다.

중소기업에 비해 대기업은 교육활동 수준이 많다는 응답자가 많아, 교육의 양은 많으나, 소프트웨어 안전 개별 과정의 필요성 조사에서는 대기업이 가중치로 판단할 때 중소기업이나 공공기관보다 높은 가중치를 보여 소프트웨어 안전 교육의 질은 교육의 양만큼 충분하지 않음이 확인되었다.

교육을 실시하는 데 가장 큰 걸림돌은 업무공백이라고 조사되어 교육 정책 마련 시 업무공백에 대한 문제를 해결하는데 주력해야 한다는 결과가 나왔다. 그러나 교육 참

여 가능 시간대에 대해서는 주말이라고 대답한 응답자는 4.3%로 조사되어 근무 시간 외 근무는 선호되지 않으며, 기업차원에서는 초과 근무 수당을 제공하지 않으려는 의지가 보인다. 적정 교육 기간으로는 1일이 가장 많아 소프트웨어 안전 교육을 하기는 충분치 않은 것으로 판단되며, 2~3일 정도로 각 과정을 구성하여 단계별 교육 계획을 마련하는 것이 필요하리라 본다.

예상 교육 인원에 대해서는 전체 기업을 조사한 것이 아니라, 전체 교육 수요를 유추하기는 쉽지 않았으며, 이를 계산하기 위해서는 산업군별 소프트웨어 안전 인력의 통계 자료가 지원되어야 한다. 개별 기업의 전체 소프트웨어 인력 대비 예상 교육 인원 비율 확인으로, 조사 대상의 절반이 넘는 기업이 모든 소프트웨어 개발자가 소프트웨어 안전 교육을 받아야 한다고 인식하는 것으로 조사되었다.

교육이 필요로 하는 직무는 개발자로 조사되었으며, 안전전문가가 필요하다는 응답은 10%대로 아직 안전전문가에 대한 인식이 미약하다고 판단되어 국내에서 안전전문가의 역할과 중요성에 대한 홍보가 필요하다고 판단된다.

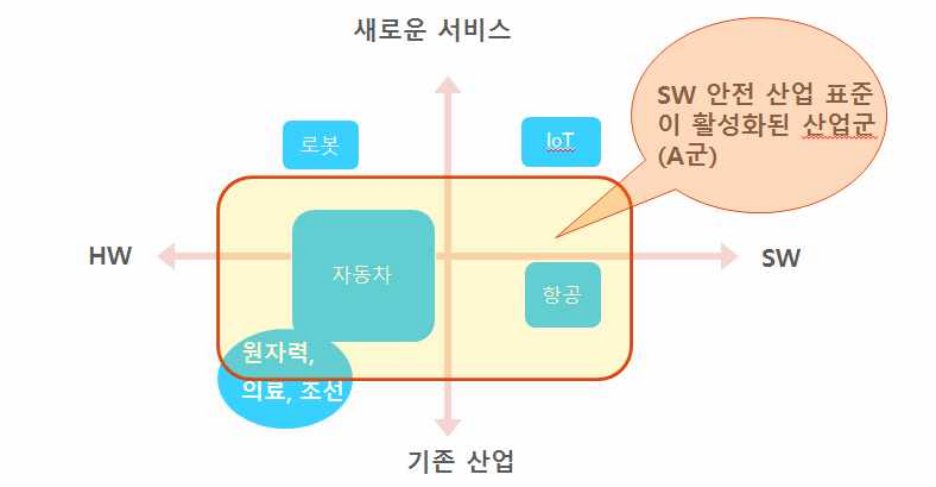
소프트웨어 안전관련 전문 인력이 되는데 소요기간은 1~3년이라는 응답이 가장 많아, 7년 이상 오랜 기간이 필요하다는 해외 전문가의 의견과는 큰 차이가 보였다. 또한 철도나 항공 등의 도메인 전문 소프트웨어 안전 평가자를 위한 최소요건에 20년 정도의 경력이 필요한 외국의 사례를 볼 때 국내에서의 소프트웨어 안전 역량을 높이기 위한 인식 및 노력이 더 필요하다고 생각된다.

교육과정 선호도 조사 결과는 기업 규모별로 정리하였으며, 기업 규모별로 필요한 과정과 시급한 과정이 달라 기업의 수요에 가능한 맞추되 소프트웨어 안전 확보를 위한 최소 교육 과정은 기업에 권장하는 것도 필요하다고 본다.

소프트웨어 안전 전문가 심층인터뷰는 원자력, 항공, 자동차, 의료 등 안전의 중요성이 높은 분야 위주로 하되, 인증 경험이 있어나 비교적 SW 안전에 대한 의식이 높다고 판단되는 업체의 종사자를 대상으로 진행하였다.

조사한 산업군을 다음과 같이 나누어 A군(자동차, 항공, 원자력 등)과 다른 산업군과 비교 분석하였다. 가로축은 재직자 및 사내에서 중요하게 인식하는 것을 기준으로 왼쪽은 하드웨어, 오른쪽은 소프트웨어 중심으로 배열하였다. 세로축은 소프트웨어 및 산업군과의 융합정도를 기준으로 융합정도가 높은 산업군을 세로축의 위쪽에 배치하였다.

[그림5-49] 심층인터뷰 산업군 분류



A군은 국제표준 존재하여 소프트웨어 안전을 구현하고 공공기관과 대기업은 안전 관리자가 존재하여 개발 전체 단계에 참여하나 중소기업은 안전 관리자 및 체계적인 안전 관리 대책이 미비하였다. 또한 변경영향 분석, 위험 분석 및 측정 방법 등에 대한 지식을 가지고, 안전 요구사항을 하드웨어/소프트웨어 구성요소 별로 체계적으로 분할 할당하여 구현하고 있었다. 그러나 모든 산업군에서 교육 과정 부재로 소프트웨어 안전에 대한 지식 습득이 어렵다고 했으며, 소프트웨어 안전 필요성을 인식시키기 위한 제도적 법제화 및 자격 인증 등 정부의 지원을 요청하였다.

기계/로봇, IoT, 의료 분야 등 기업 안전 관리자 및 체계적인 안전 관리 대책이 미비하여, 신기술 및 융합 분야의 소프트웨어 안전 관련 교육 시행을 교육 정책 마련 시 고려해야 한다.

소프트웨어 안전 구현 면에서는 A군은 분석 및 설계 단계에서 소프트웨어 안전에 대한 메커니즘을 설정하고 이에 따른 구현 및 검증을 수행하고 있었다. 일부 중소기업은 테스팅 단계에서 소프트웨어 안전에 대한 검증에 대한 방법 습득 및 도구에 어려움을 토로했으며, 대부분의 업체는 시스템 안전 분석에 대한 방법으로 FMEA를 사용하고 있었다. 해외 교육 동향이나 전문가 심층 인터뷰 결과로 분석 방식으로 FMEA를 가장 많이 사용하는 것으로 보이며, 소프트웨어 안전 분석관련 방법은 아직 도출되지 않은 것으로 보인다. FMEA의 경우는 하드웨어 위험 분석 방식으로 소프트웨어에 적용하기는 어려움이 있으나 현재로는 다른 대안이 있어 보인지는 않는다. 의료분야는

표준이 존재하나, 안전 분석이나 테스팅 등 일부 단계만을 제한적으로 수행하고 있어, 의료 분야는 헬스케어서비스 등 인간의 안전에 가장 중요한 역할을 할 소프트웨어 안전에 대한 교육이 가장 필요한 분야라고 할 수 있다.

조사 대상 대부분의 업체가 테스팅은 어느 정도 체계적으로 진행하고 있어 설문조사 결과와는 대조적인 결과를 보였으며, 소프트웨어 안전에 대한 중요시 여기는 업체는 테스팅에 대한 중요성부터 강조함을 역설적으로 확인하게 되었다.

주요 요청 교육과정은 소프트웨어 안전 분석, 설계, 검증 관련 기술과 체계적인 소프트웨어 개발 기술이며, 그 이외에 소프트웨어 안전 표준, 소프트웨어 안전 관련 개념/ 최신 기술 동향이 있어, 설문 조사와는 달리(테스트 과정 등) 소프트웨어 안전에 중요한 과정에 대한 교육이 더 필요하다고 인식하고 있었다.

교육의 적절한 시간으로는 대부분의 업체가 평일 2~3일 교육을 원해 설문조사 결과와 유사한 결과가 도출되었으며, 교육 시기로는 대부분 업무 비수기를 원했으나, 각 업체별 비수기가 일정치 않거나 예측이 어려운 업체가 많았다. 비수기 교육은 교육에 가장 걸림돌로 조사되었던 업무 공백 문제를 해결하기 위한 방안이라고 보이나, 비수기가 예측하기 어려운 점은 시행이 쉽지는 않을 것으로 예상된다.

제6장 문제점 분석 및 개선 방안

제1절 문제점 분석

1. 부족한 소프트웨어 안전 인식 및 환경

설문조사에 참여한 업체 대부분이 소프트웨어 안전성이 필요한 분야임에도 불구하고 기능안전교육을 실시한 업체보다 실시하지 않은 업체가 더욱 많은 것으로 나타났다. 재직자 수요조사의 세부 내용에서도, 교육을 가장 필요로 하는 직무로 안전전문가에 대한 응답률이 가장 낮게 나오고 소프트웨어 안전성 확보를 위한 기술적 활동으로 실제 가장 중요한 요소 중 하나인 위해도 분석에 대한 응답률이 가장 낮게 나오는 등 소프트웨어 안전 관련 인식이 업계 전반적으로 부족한 것을 확인할 수 있었다.

특히 안전 관련 기술을 선도하는 주요 업체들의 전문가들을 대상으로 한 심층인터뷰에서도 기능안전과 일반적인 테스팅을 혼돈하거나, SIL(Safety Integrity Level)이나 FMEA/FTA 등에 대한 지식이 부족한 것으로 나타나 소프트웨어 안전 관련 인식제고가 시급한 것으로 보인다.

기능안전교육을 실시하고자 하는 업체도 다수 있었지만, 교육환경이 부족하거나 교육 관련 정보를 얻지 못하여 교육을 실시하지 못하는 경우도 다수 있었다. 기능안전 교육을 진행하는 기관이 많지 않고, 그 정보가 매우 부족하여 적절한 교육이 제대로 이루어지고 있지 않으며, 해외 인증 및 컨설팅 기관에서 진행하는 교육들이 있지만 가격이 비싸고 국제 표준을 단순히 읽어주는 수준에 그쳐 교육에 대한 만족도가 크지 않는 것으로 보인다. 대기업과 공공기관 중심으로 소프트웨어 안전에 중요성을 실감하고 시스템 안전사고에 대한 잠재 위험 분석, 소프트웨어 안전 분석, 설계, 검증 관련 기술 등 실질적 소프트웨어 안전 교육을 시행하려고 하는 움직임이 증가되고 있으나, 안전 과정 부족 및 교육에 대한 정보 수집의 어려움을 느끼고 있다. 특히 설문조사 결과와는 달리 심층인터뷰 결과 안전 관련 소프트웨어를 개발하는 중소기업은 아직 소프트웨어 안전 중요성을 인지하지 못하고 있기 때문에 안전 인

식을 제고할 수 있는 교육이 대기업보다 필요한 것으로 나타나고 있다.

소프트웨어 안전공학 기술에 대한 이해 수준의 관점에서 볼 때, 국내에서는 발주기관에 해당하는 공공기관의 경우 이해 수준이 높지 않은 것으로 파악되며, 소프트웨어 안전과 일반적인 품질 향상과의 기술적인 구분을 명확히 하지 못하는 것으로 보인다. 예를 들자면, 공공기관의 경우 필요교육의 우선순위로써 테스팅과 국제 표준 및 규제에 주안점을 두었는데, 소프트웨어 안전은 안전 메커니즘의 분석과 설계가 가장 중요한 핵심 기술이며, 이러한 안전 메커니즘에 대한 요건은 발주자의 발주 요건상에도 분명히 명시되어 있어야 하는 내용이다.

소프트웨어 안전에 관련된 교육의 필요성과 시급성에 대한 인식 관점에서 볼 때, 중소기업은 그 중요성을 크게 인식하고 있지 않는 것으로 추정이 된다. 예를 들면, 제품에 대한 수출, 사용, 운영을 책임지는 공공기관과 대기업의 경우 대부분의 안전성 관련 교육의 필요성과 시급성을 높게 인지하고 있는 반면에, 중소기업의 경우는 중요하다고 생각하나 필요성과 시급성을 그다지 높게 인지하고 있지는 않았다.

교육 대상자 선정을 위해서는 소프트웨어 안전 관련 직무가 명확히 정의되어야 한다. 국내에서는 NCS(National Competency Standards, 국가직무능력표준)에도 소프트웨어 안전 역량은 찾아보기 어려우며, 소프트웨어 안전 관련 직군이 정의되어 있지 않다. 본 연구에서는 철도 분야 IEC62279 표준과 전문가들의 자문으로 소프트웨어 안전 관련 역할을 정의하였다.

2. 시스템 중심의 안전성 교육

여러 기관이나 기업체에서 주최하는 교육들은 대부분 일반 기능안전성의 국제표준인 IEC61508을 중심으로 각 도메인별로 국제 표준에 맞는 시스템을 대상으로 하는 안전성 교육이 대다수였다. 해외의 경우에는 일반적인 소프트웨어 안전에 대한 교육을 실시하는 민간 기업을 다수 확인할 수 있었으며, 대학 및 대학원 수업에서도 소프트웨어 안전을 주제로 다수의 과정이 진행되고 있음을 확인할 수 있었다. 반면에 국내의 경우에는 상명대 전문 과정을 제외하면 모두 표준 위주의 과정으로 전문적인 소프트웨어 대상 교육을 찾아보기가 매우 힘들었다.

국내의 안전성 교육의 진행 형태는 대부분 기능안전성 기준표준(IEC61508)으로

시작해서, 각 응용 도메인별로 특화된 표준(자동차의 경우 ISO26262, 항공의 경우 DO-178C, 철도의 경우 EN50128, 원자력의 경우 IEC60880 등)을 설명하는 교육이 다수였다. 특히 소프트웨어 안전성은 시스템 안전성으로부터 시작되는 한 부분으로 인식되고 있어 교육 말미에 간단하게 소개하는 정도로만 진행되고 있다.

3. 연속성이 없는 표준 위주의 안전성 교육

기업은 주로 소프트웨어/시스템 안전에 대한 전반적인 내용을 설명한 후, 기능안전성 중심으로 각 도메인 특화된 설명 및 실습을 제공한다. 그렇기 때문에 교육들이 모두 체계적이지 못하고 산발적으로 진행되고 있으며, 수요가 많은 도메인(주로 자동차)에 특화된 교육에 치우쳐져 있다.

협회는 표준 자체를 설명하거나, 해당 도메인에 필요한 내용만을 중점적으로 다루는 특화된 교육과정을 제공한다. 특히 협회에서 주최하는 교육의 경우 대부분 표준의 개정 시기에 맞춰서 진행되거나 각 도메인의 교육에 대한 수요가 많을 때 한정적으로 진행되기 때문에 교육이 주기적으로 진행되지 않고 시스템에서 소프트웨어 안전성으로 연결되는 연속성이 없다.

대학은 일부 항공 등 특정 도메인에 특화된 학과를 제외하면, 대부분 산업계에서 바로 적용 가능한 내용보다는 보다 넓고 실험적인 내용으로 구성된 교육과정을 제공한다. 따라서 소프트웨어 안전성에 대한 보다 원론적인 교육을 진행하고 있지만, 기업 및 협회에서 진행하는 교육과 차이가 커 교육의 연속성에 문제가 있어 보인다.

기능안전과 관련된 분석, 설계, 구현, 검증 등 실무와 관련된 기술 중심의 교육에 대한 선호도가 기초 이론 및 관련 국제 표준(IEC61508, ISO26262, DO-178C 등) 등 표준/규제 중심의 교육에 대한 선호도보다 높게 나타났다. 이는 아직까지도 국내에서는 위험분석 및 안전 메커니즘 설계기법 등의 심도 있는 전문화된 교육은 제대로 이루어지지 않는 것으로 파악된다.

소프트웨어 안전 인식제고에 대한 필요성은 다른 기술 중심의 교육이나 이론 중심의 교육보다 높게 나타나 현업 종사자들도 현재 소프트웨어 안전에 대한 인식제고가 필요하다고 생각하는 것을 알 수 있었다.

제2절 개선방안

수요 조사 결과 산업별 소프트웨어 안전 인식 및 역량이 달라, 인식 및 역량 수준에 따른 교육 과정 개발이 필요하다. 신기술 분야에 비교해서는 기존 산업 분야가 안전에 대한 인식 수준이 높고 안전 기술 역량을 보유하고 있으나, 전체 산업에 산업 특성 및 소프트웨어 안전 기술 수준에 따라 교육 계획을 세우고 소프트웨어 안전 교육을 실시하는 것이 필요하다. 중소기업은 교육으로 인한 업무 부담을 최소한으로 할 수 있는 방안 마련하여 소프트웨어 안전 인식에 대한 교육부터 필요하며, 대기업은 실무교육 위주의 교육이 필요하리라 본다. 소프트웨어 안전 인식 및 역량에 따라 교육 방향을 다음과 같이 분류하였다.

[그림6-1] 소프트웨어 안전 인식 및 역량에 따른 교육 방향

자동차, 항공, 원자력 등 기능 안전 관련 국제 표준이 존재하며 소프트웨어에 대한 인식 수준과 역량이 모두 높은 산업군은 국제 표준에 따른 실무 능력 강화 교육이 요구되며, 소프트웨어 안전 기술을 타 분야에 확산하는 것이 필요하다. 소프트웨어 안전 표준이 존재하거나 제작 중인 의료, 조선 분야는 인식 수준은 높으나, 소프트웨어 안전 기술 수준이 낮아 안전 기초 교육 및 표준 교육이 필요하다. 소프트웨어 안전 인식 및 역량 수준이 낮으나, 안전이 보다 중요 시 되는 소프트웨어 융합 분야는 소프트웨어 안전 표준 정립 및 교육이 필요하다. 수요조사와 전문가의 자문결과 산업별 전문 교육은 산업 도메인 자체에서 전문가를 양성하여 교육하는 것이 역량 향상에 효율적이며, 정부

는 소프트웨어 안전 인식 제고 및 소프트웨어 안전 전반의 교육을 담당하는 것이 바람직하다고 본다.

1. 소프트웨어 안전 인식 및 환경 개선

현재 특정 도메인의 소프트웨어 안전을 중요시 여기는 소수의 업체를 제외하고는 소프트웨어 안전에 대한 인식이 매우 부족한 것으로 보인다. 기본적으로 일반적인 소프트웨어 품질의 높은 수준이 안전이라는 오해 혹은 잘못된 인식들이 있는 것으로 추정된다. 소프트웨어 안전이란, 기본적으로 소프트웨어에는 잠재적 결함이 존재하는 것으로 가정하며, 이러한 예상치 못한 잠재적 결함이 시스템의 오류를 발생시킬 경우 시스템을 안전한 상태로 유지하는 메커니즘에 대한 설계와 구현에 그 초점을 맞추는 것이다.

소프트웨어 안전에 대한 이와 같은 낮은 인식 및 부족한 교육 환경을 개선하기 위하여 정부 차원에서 소프트웨어 안전과 관련된 캠페인을 실시하고, 현재 산업체에서 진행되고 있는 도메인별로 특화된 기능안전교육 외에도 **범 도메인 차원의 소프트웨어 안전에 관련된 기초 개념 교육**을 마련하여 이를 적극적으로 홍보, 실시하여 소프트웨어 안전의 필요성을 점차 확대해야 할 것이다.

또한 시급성에 따라 산업 도메인별로 **국제 수준에 준하는 소프트웨어 안전과 관련된 법규 및 자격인증**을 강화하고, 정부 주도의 소프트웨어 안전 관련 세미나를 개설하고 일정 교육시간을 의무화하는 등 **일정부분 소프트웨어 안전 관련 교육을 필수화**하여 시행하는 것도 하나의 방법으로 보인다. 수요조사에서도 소프트웨어 안전 교육을 자율화하는 것보다는 필수 교육이나 자격증과 연계한 부분적 필수 교육에 대한 응답이 65% 가량으로 조사되어 이는 수요자의 기대에도 부응하는 것이다.

국제 수준에 준하는 소프트웨어 안전 관련 자격증의 시행을 위해서는 소프트웨어 안전 관련 직무의 정의가 필요하며, 국가직무능력표준(NCS) 등에서 제도화하여 교육 대상자를 선택하고 집중하여 지원하는 것이 필요하다. 또한 실질적인 소프트웨어 안전 교육을 위해서는 소프트웨어 안전 교육 대상자 현황 파악과 선정을 위한 통계 등도 구축되어 지원되어야 한다.

2. 소프트웨어 안전성 교육의 확대

현재 시행되고 있는 안전성 교육은 소프트웨어에 초점을 맞춘 교육보다는 시스템 안전성 교육 위주로 진행되고 있다. 소프트웨어 안전성 또한 시스템 안전성 못지않게 중요한 만큼 소프트웨어 안전성 교육을 개설 및 확대할 필요가 있다.

이론적 배경에서 살펴본 본 것과 같이 소프트웨어 안전 기술의 많은 부분이 소프트웨어 공학 기술과 동일하게 사용하기 때문에, 대학 소프트웨어 공학과정에 소프트웨어 안전 기초 개념을 도입해야 한다. 이는 미래 소프트웨어 개발자들에게 소프트웨어 안전의 중요성을 인식시키고, 소프트웨어 안전 분야 재직자들에게 대학과정과 연계한 소프트웨어 안전 전문가 교육 체계를 지원할 것이다.

산업 도메인별 안전성 교육은 해당 도메인의 대기업과 공공기관이 시장 특성에 따라 자발적으로 진행하고 있고, 또 진행하는 것이 바람직한 것으로 판단된다. 반면 중소기업의 경우, 교육환경과 교육비용 자체에 대한 부담과 더불어 소프트웨어 안전에 대한 인식 자체도 대기업이나 관련 공기업에 비하여 현저하게 낮으므로, **범 도메인 차원의 공통적인 소프트웨어 안전 교육 과정**을 정부에서 개설하여 중소기업을 대상으로 운영하는 것이 필요하다고 판단된다.

3. 체계화된 교육 방식 수립

현재 국내에서 시행되고 있는 안전성 교육은 각 산업 도메인별 국제 표준에 맞춘 교육이 대다수이다. 이러한 방식의 안전성 교육은 필요할 때에 단발성으로 시행되는 경우가 많아 근본적인 소프트웨어 안전 교육에 대한 해답이 될 수 없으므로, 소프트웨어 안전 일반 국제 표준인 IEC61508을 포함하는 범 도메인 차원의 공통적인 소프트웨어 안전 교육 과정을 마련하여 이를 기본적으로 시행하고, 이를 수료한 재직자들을 대상으로 각 도메인별 안전 표준에 대한 교육을 각각 실시하는 것이 바람직하다고 판단된다. 다만, 도메인별 안전 표준에 대한 교육은 현재 대기업이나 각 컨설팅 업체에서 필요에 따라 잘 진행되고 있기 때문에, 정부에서는 이에 대한 지원 및 홍보를 통하여 중소기업에서도 이러한 교육 활동이 활발히 이루어질 수 있도록 힘써야 할 것이다.

교육 방식에 있어서는, 수요조사 결과에 따라 중소기업이 필요한 교육과정은 **5일 이내의 단기 강좌**들이 적절하며, 대학이나 인증기관의 교육보다는 **우수한 강사진을 확보하여 정부차원에서 제공**하는 것이 바람직하다고 판단된다. 소프트웨어 안전 교육을

위한 교수자 확보를 위해서는 소프트웨어 안전 구현에 대한 체계가 비교적 잘 마련되고 지켜지고 있는 항공, 원자력 등 분야의 전문가로 재직자 소프트웨어 안전 교육을 위한 전문가 풀을 구축하는 것이 필요하다.

또한 비용 차원에서는 **무료교육보다는 유료교육이 적합**하다는 전문가들의 의견이 있다. 이는 무료교육일 경우 교육 참여생의 적극성 자체가 떨어지는 부작용이 있으며, 동시에 운영기관의 예산상의 한계 또한 있게 되어 훌륭한 강사진 섭외가 불가능하게 된다. 질 좋은 교육은 좋은 교육내용과 가장 중요한 것은 강사진의 지식과 역량이다.

제3절 커리큘럼 제안

앞서 2장에서 제안한 안전 소프트웨어 개발을 위한 예상 기술 집합과 이어 실시한 재직자 수요조사 및 소프트웨어 안전 전문가 심층인터뷰의 결과를 바탕으로 본 연구에서는 다음과 같은 소프트웨어 안전 분야의 예상 교육과목들을 제안한다.

각 과정은 2장에서 정의한 7가지의 역할(프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자)을 기반으로 교육 대상을 정했으며, 권장되는 교육 시간을 명시한다. 과정 학습에 필요한 선수 지식을 정의하여 교육을 체계적으로 구성하였다.

다음은 역할별 이수가 권장되는 과목이다. 본 표는 과정명을 간단히 기입했으며, 과정에 대한 상세한 설명은 다음 표로 정리하였다.

<표 6-1> 역할별 권장 과정

	프로젝트 관리자	설계자	개발자	확인/ 검증자	형상 관리자	품질보증 관리자	안전 관리자
1)안전인식	o	o	o	o	o	o	o
2)개념	o	o	o	o	o	o	o
3)안전공학	o	o	o	o	o	o	o
4)시스템분석	o	o	o			o	o
5)시스템설계	o	o	o			o	o
6)SW분석	o	o	o			o	o
7)설계	o	o	o			o	o
8)코드분석	o		o	o		o	o
9)테스팅	o	o	o	o	o	o	o
10)정형기법		o	o			o	o
11)표준	o	o	o	o	o	o	o
12)표준이론	o	o	o	o	o	o	o
13)안전관리	o						o
14)형상관리	o	o	o		o		o

다음은 각 과정의 선수과목(prerequisite subject, 선행 과목)과의 관계를 정리한 것이다. 상위의 과정이 선수과목이 된다. 역할별로 특징을 살펴보면 첫 번째, 두 번째 군이 프로젝트 관리자, SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자를 위한 커리큘럼이며, 세 번째 군은 모든 역할 담당자에게 속하며, 마지막 군은 형상관리자를 위한 것이다. 확인 및 검증자는 두 번째 군의 일부 과정을 선별하여 학습할 수 있다.

<표 6-2> 소프트웨어 안전 커리큘럼 제안

<table>
<tr><td colspan="4">소프트웨어 안전 인식 선진화</td></tr>
<tr><td colspan="4">소프트웨어 안전개념 및 기초 이론</td></tr>
<tr><td colspan="4">소프트웨어 안전공학 방법론 개론</td></tr>
<tr><td>시스템 안전분석</td><td colspan="2">소프트웨어 안전 분석</td><td>안전표준 및 규제에 따른 안전공학기법 이론 교육</td><td>형상관리/ 결함관리</td></tr>
<tr><td>시스템안전설계</td><td colspan="2">소프트웨어 안전 설계 및 구현</td><td></td><td></td></tr>
<tr><td></td><td colspan="2">소프트웨어 코드 안전 분석</td><td></td><td></td></tr>
<tr><td></td><td colspan="2">소프트웨어 안전성 테스팅기법</td><td></td><td></td></tr>
<tr><td></td><td>고안전성시스템을 위한 정형기법</td><td>소프트웨어 안전성관리</td><td></td><td></td></tr>
<tr><td>1 군</td><td colspan="2">2 군</td><td>3 군</td><td>4 군</td></tr>
</table>

1) 소프트웨어 안전 인식 선진화

과정명	소프트웨어 안전 인식 선진화	교육시간	0.5일
과정개요	소프트웨어 안전 관련 최신 동향 및 인식을 제고하고, 중요성을 습득한다.		
교육대상	모든 직군		
교육목표	소프트웨어 안전에 대한 올바른 개념을 배운다. 소프트웨어 안전과 일반 품질과의 관계를 학습한다. 소프트웨어 안전에 대한 대표적인 국제 표준들을 이해한다. 소프트웨어 안전과 관련된 해외 동향을 파악한다.		
선수지식	없음		
교육내용	1. 소프트웨어 안전에 대한 정의 2. 소프트웨어 기능 안전과 소프트웨어 품질 3. IEC61508과 관련 파생 표준 4. 소프트웨어 안전 사고 및 소프트웨어 안전 관련 해외 사례		

2) 소프트웨어 안전 개념 및 기초 이론

과정명	소프트웨어 안전 개념 및 기초 이론	교육시간	1일
과정개요	소프트웨어 안전성에 대한 개념, 위험, 안전 등급 등에 대한 기본 이론을 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자		
교육목표	소프트웨어 안전성에 대한 기술적 개념 이해 소프트웨어의 위험과 위험 분석에 대한 이해 및 안전 등급에 대한 기본 이론 습득		
선수지식	소프트웨어 안전 인식 선진화		
교육내용	1. 소프트웨어 기능 안전과 소프트웨어 품질 2. IEC61508과 관련 파생 표준 3. Safety Management Process 4. Functional Safety Development Process 5. Hazard and SIL(Safety Integrity Level)		

3) 소프트웨어 안전 공학 방법론 개론

과정명	소프트웨어 안전 공학 방법론 개론	교육시간	2일
과정개요	안전성 추론 방법, 주요 소프트웨어 안전 공학 방법론(STAMP/STPA 등) 등에 대한 전반적인 이해를 한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자		
교육목표	안전성 추론 방법 습득 주요 소프트웨어 안전 공학 방법론인 STAMP, STPA에 대한 이해 시스템 엔지니어링 기초 이론 습득		
선수지식	소프트웨어 안전 개념 및 기초 이론		
교육내용	1. Safety Case 2. Safety Engineering Method (STAMP/STPA) 3. System Engineering Process 4. Software Engineering Process 5. Basic Safety Analysis Techniques		

4) 시스템 안전 분석

과정명	시스템 안전 분석	교육시간	1일
과정개요	해저드(Hazard)의 식별 및 평가를 통해 안전목표(Safety Goal)를 도출하기 위한 전반적인 절차(FMEA, FTA 등)에 대해 배운다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자		
교육목표	해저드(Hazard)의 식별 및 평가 방법 습득 안전목표(Safety Goal)를 도출하기 위한 전반적인 절차인 FMEA, FTA 습득		
선수지식	소프트웨어 안전 공학 방법론 개론		
교육내용	1. Hazard Analysis & Risk Assessment 2. Safety Goal Management 3. FMEA, FTA		

5)시스템 안전 설계

과정명	시스템 안전 설계	교육시간	1일
과정개요	기능안전 요구사항에 대한 이해를 바탕으로 이를 만족하는 시스템 설계 방법(FTA/FMEDA)에 대해 배운다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자		
교육목표	기능안전 요구사항에 대한 기본 개념 이해 기능안전 요구사항을 만족하는 시스템 설계 방법인 FTA 및 FMEDA습득		
선수지식	시스템 안전 분석		
교육내용	1. Functional Safety Mechanism 2. Architecture-level FTA 3. FMEDA		

6) 소프트웨어 안전 분석

과정명	소프트웨어 안전 분석	교육시간	1일
과정개요	SW 개발 수준에서 요구되는 기능안전 분석 요건의 이해 및 기능안전 분석 요건을 만족하는 임베디드 소프트웨어 분석 방법(SW-FMEA)을 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자		
교육목표	SW 개발 수준에서의 기능안전 분석 요건 이해 임베디드 소프트웨어 분석 방법 습득		
선수지식	소프트웨어 안전 공학 방법론 개론		
교육내용	1. Software Requirement Analysis 2. Software Functional Safety Analysis (SW-FMEA, SW-FTA) 3. Software Safety Analysis Workshop		

7) 소프트웨어 안전 설계 및 구현

과정명	소프트웨어 안전 설계 및 구현	교육시간	1일
과정개요	기능안전 분석 요건을 만족하는 임베디드 소프트웨어 분석 방법(SW-FTA)을 습득하고, SW 안전 메커니즘 설계에 대해 이해한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자		
교육목표	소프트웨어 설계 방법 습득 SW 안전 메커니즘 설계 이해		
선수지식	소프트웨어 안전 분석		
교육내용	1. Software Architecturing and Design 2. Architecture-level SW-FTA 3. SW-FMEDA 4. Change Impact Analysis 5. Software Safety Mechanism Design Workshop		

8) 소프트웨어 코드 안전 분석

과정명	소프트웨어 코드 안전 분석	교육시간	1일
과정개요	코드레벨의 소프트웨어 안전 메커니즘을 이해하고, 안전 코딩 표준 및 정적 테스팅을 숙지한다.		
교육대상	프로젝트 관리자, SW 개발자, SW 확인 및 검증자, SW 품질보증 관리자, 안전 관리자		
교육목표	코드레벨에서의 소프트웨어 안전 메커니즘 이해 안전 코딩 표준 습득 정적 테스팅의 기본 및 그 방법 습득		
선수지식	소프트웨어 안전 설계 및 구현		
교육내용	1. Coding Standard 2. Code Refactoring 3. Code-level FTA 4. Static Analysis		

9) 소프트웨어 안전성 테스팅 기법

과정명	소프트웨어 안전성 테스팅 기법	교육시간	1일
과정개요	Fault-injection Testing 등 안전 메커니즘 검증 방안을 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자		
교육목표	코드레벨 안전 메커니즘 검증 방안 습득		
선수지식	소프트웨어 코드 안전 분석		
교육내용	1. Software Testing in Different Phase 2. Fault-injection Testing 3. Host-target Testing 4. Test Case Management		

10) 고안전성 시스템들을 위한 정형 기법

과정명	고안전성 시스템들을 위한 정형 기법	교육시간	3일
과정개요	고안전성 수준을 요하는 소프트웨어에 대한 정형 기법을 습득한다.		
교육대상	SW 설계자, SW 개발자, SW 품질보증 관리자, 안전 관리자		
교육목표	고안전성 수준의 소프트웨어에 대한 Formal Method 습득		
선수지식	1~9 과정		
교육내용	1. FTA, FMEA, FMEDA & Logical Proving 2. Program Slicing 3. Reachability Analysis 4. Predicate Calculus 5. Modeling and Simulation 6. Models and Their Syntax & Semantics		

11) 소프트웨어 안전 관련 국제 표준/규제

과정명	소프트웨어 안전 관련 국제 표준/규제	교육시간	1일
과정개요	국방(MIL-STD-882), 항공(DO-178C), 원자력(IEC60880), 철도(EN50128), 자동차(ISO26262), 일반(IEC61508) 등 각 도메인별 소프트웨어 안전 관련 국제 표준 및 규제를 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자		
교육목표	도메인별 안전 관련 국제 표준 살펴보기 일반 국제 표준인 IEC61508에 대한 기본 이해 및 습득		
선수지식	소프트웨어 안전 인식 선진화		
교육내용	IEC61508, ISO26262, DO-178C, EN50128, IEC60880, MIL-STD-882		

12) 안전표준 및 규제에 따른 안전공학기법 이론 교육

과정명	안전표준 및 규제에 따른 안전공학기법 이론 교육	교육시간	2일
과정개요	FTA, FMEA, HAZOP, HARA, Formal Verification, 테스팅, HILS & SILS, Traceability Analysis 등 각종 안전공학기법을 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, SW 확인 및 검증자, 형상 관리자, SW 품질보증 관리자, 안전 관리자		
교육목표	각종 안전공학기법 살펴보기 상황별 사용할 수 있는 안전공학기법 선택 및 기본이론과 사용방법 습득		
선수지식	소프트웨어 안전 공학 방법론 개론		
교육내용	1. HARA, HAZOP 2. FTA, SW-FTA 3. FMEA, SW-FMEA 4. Formal Verification 5. Static & Dynamic Testing 6. HILS & SILS 7. Traceability Analysis		

13) 소프트웨어 안전성 관리

과정명	소프트웨어 안전성 관리	교육시간	1일
과정개요	고안전성 소프트웨어 프로젝트 관리 및 안전 관리자 역할과 책임에 대해 배운다.		
교육대상	프로젝트 관리자, 안전 관리자		
교육목표	고안전성 소프트웨어 관련 프로젝트 관리 이해 안전 관리자의 역할과 책임 이해		
선수지식	1~9 과정		
교육내용	1. Functional Safety Management 2. Safety Manager R&R 3. Safety Process Tailoring 4. Safety Audit		

14) 형상관리 및 결함 관리

과정명	형상관리 및 결함 관리	교육시간	1일
과정개요	소프트웨어 형상관리와 체계적인 결함 분석 및 관리 기법을 습득한다.		
교육대상	프로젝트 관리자, SW 설계자, SW 개발자, 형상 관리자, 안전 관리자		
교육목표	소프트웨어 형상관리 이해 체계적인 결함 분석 및 관리 기법 습득		
선수지식	소프트웨어 안전 공학 방법론 개론		
교육내용	1. Configuration Management Planning 2. Configuration Management System 3. Defect Management 4. Orthogonal Defect Analysis		

제7장 결론

제1절 연구의 요약

본 연구는 소프트웨어 안전 교육 정책 마련을 위해 소프트웨어 안전 이론을 기반으로 소프트웨어 안전 관련 업체 설문조사와 심층인터뷰를 시행한 최초의 시도이다.

국내외 다양한 시스템 및 소프트웨어 안전성 교육을 살펴본 결과, 이론적이고 실험적인 주제로 교육을 진행하는 대학 수업을 제외하면 대부분의 기업 및 협회에서는 기능안전 일반 국제표준인 IEC61508을 중심으로 각 도메인별로 국제 표준에 맞는 시스템을 대상으로 하는 안전성 교육이 대다수였다. 따라서 기능안전 교육들이 산발적으로 진행되고 있어 시스템 안전 및 소프트웨어 안전성에 대한 연속성이 부족한 실정이다.

소프트웨어 안전성과 관련된 인식 또한 설문조사를 살펴본 결과, 일반적인 테스팅과 기능안전을 혼동하거나, SIL(Safety Integrity Level)이나 FMEA/FTA 등에 대한 지식이 부족하고, 특히 국내 주요 산업의 발주기관에 해당하는 공공기관에서 필요교육의 우선순위로 안전 메커니즘의 가장 중요한 핵심 중 하나인 분석과 설계를 낮게 측정하는 것으로 나타나 소프트웨어 안전 관련 인식이 부족한 것으로 드러났다. 또한 기능안전이 필요한 도메인의 대부분의 업체에서 소프트웨어 안전성이 필요함에도 불구하고 교육을 진행하거나 진행할 계획이 없으며, 교육을 진행하고자 하더라도 기능안전에 대한 교육의 정보가 매우 부족하고, 실시되고 있는 교육 자체 또한 부족한 실정이다. 그나마도 현재 해외 인증 및 컨설팅 기관에서 진행하고 있는 교육들은 가격이 비싸고 국제 표준을 단순히 읽어주는 수준에 그쳐 교육에 대한 만족도 또한 크지 않다.

제2절 시사점 및 향후 연구

본 보고서에서는 소프트웨어 안전성과 관련된 인식을 우선 개선하기 위하여 정부 차원에서 소프트웨어 안전과 관련된 캠페인을 실시하고, 특히 교육환경과 교육비용 자체에 대한 부담과 더불어 소프트웨어 안전에 대한 인식도 공공기관 및 대기업에 비해 현저히 낮은 중소기업을 대상으로 하는 **범 도메인 차원의 소프트웨어 안전에 관련된 기초 개념 교육**을 마련하여 이를 통해 소프트웨어 안전 인식을 개선하고, 각 도메인별로 진행되는 국제 표준에 대한 교육에 앞서 대학 및 대학원에서 진행되는 이론 교육

과의 연결고리를 만족시키는 것이 중요하다.

또한 시급성에 따라 산업 도메인별로 **국제수준에 준하는 소프트웨어 안전과 관련된 법규 및 자격인증**을 강화하고 정부 주도의 소프트웨어 안전 관련 세미나를 개설하여 소프트웨어 안전 인식을 개선함과 동시에 일정 교육시간을 의무화하여 **일정부분 소프트웨어 안전 관련 교육을 필수화**하는 등 시스템 및 소프트웨어 안전 관련 교육의 필요성을 높이는 것 또한 필요하다. 그리고 이러한 인식 개선 및 필요성 증가에 대비하여 다양한 소프트웨어 안전성 교육을 개설 및 확대하고, 이에 대한 홍보를 강화하여 기능안전이 필요한 각 도메인에서의 교육이 수월하게 진행될 수 있도록 정부와 각 기관에서의 협조가 필요할 것이다.

교육의 진행 방식으로는 재직자 수요조사 결과에 따라 **5일 이내의 단기강좌**들을 개설하는 것이 적절하며, 대학이나 인증기관의 교육보다는 **우수한 강사진을 확보하여 정부 차원에서 제공**하는 것이 바람직하다고 판단된다. 또한 교육 참여생의 적극성을 장려하고, 훌륭한 강사진의 원활한 섭외를 위한 **적당한 비용의 유료교육으로 진행**하는 것이 적합하다.

향 후 연구에서는 재직자의 지속적인 소프트웨어 안전 역량 강화를 위해 본 연구에서 제외하였던 석・박사 과정의 소프트웨어 안전 교육을 조사하고 기업과의 연계방안 연구와 소프트웨어 안전 직무 정의를 위한 역할에 대한 연구가 추가로 필요하리라 본다. 본 연구는 공개된 해외 자료만을 활용했으며, 해외 우수 교육 기관을 방문해 공개되지 않은 소프트웨어 안전 관련 교육을 조사하고 국내에 적용하는 연구도 필요하다. 또한 도메인별로 특화된 기술을 조사하고 교육하는 방안을 마련하며, 현재 자율자동차의 개발 추세인 항공, 철도 등의 안전 기술 차용을 비취어 볼 때 교육을 통한 도메인 간의 성숙된 소프트웨어 안전 기술 활용 방안 연구도 필요하리라 본다.

제3절 연구의 한계

본 연구는 과정 개발 연구에 제한되어 있어 소프트웨어 안전 교육의 시행을 위해서는 교육 운영에 대한 연구도 필요하다. 교육의 3대 요소인 교육내용, 교수자, 교육 대상자 면에서 살펴보면, 교수자 양성 방안과 통계를 불확실성으로 확인이 어려웠던 교육 대상자 선정에 관한 연구도 필요하다. 특히 소프트웨어 안전 교육을 품질을 높이고 교육의 걸림돌로 지적되었던, 업무 공백의 문제도 해결되어야 하겠다.

참 고 문 헌

국내 문헌

관계부처 합동(2014), 소프트웨어중심사회 실현 전략

관계부처 합동(2015), 안전혁신 마스터플랜

모아소프트, 안전성 확보를 위한 철도 SW 개발 프로세스 소개 (EN50128).
http://www.moasoftware.co.kr/company/greeting.asp

NAVITHES, NAVITHES ISO 26262 교육,
https://sites.google.com/a/navithes.com/nt/gyoyug/iso26262gyoyug

SOLUTIONLINK, Training Courses for Software Engineering,
http://sol-link.com/neo/kr/academy/training_01.php

SPRi(2015), 국내 소프트웨어 안전 산업동향 조사

SPRi(2015), SW안전 체계 확보와 중점 추진과제

STA 테스팅컨설팅, STA 테스팅,
www.sten.or.kr/bbs/board.php?bo_table=training&sca=FL%2FAL+%B1%B3%C0%B0&type=1

TUV SUD Korea, 자동차 기능안전 (ISO26262) 전문가 과정 (FSCP) 교육,
http://www.tuv-sud.kr/kr-kr/about-tuev-sued/tuev-sued-in-korea/about-us-in-korea

해외 문헌

Australian National University, Systems and Software Safety,
http://programsandcourses.anu.edu.au/course/COMP8180

Carnegie Mellon University, Engineering Safety and Security Related Requirements for Software-Intensive Systems, http://www.sei.cmu.edu/training/P64.cfm

CRITICAL Software, Safety Training,
http://www.criticalsoftware.com/en/what-we-do/safety-training

Department of Energy(2011), DOE STD-1172-2011, Safety Software Quality Assurance Functional Area Qualification Standard

Edif Group, Software Development for Safety-Related System,
http://www.edifgroup.com/training/course/software-safety

Engineering Education Australia Pty Ltd, System Safety Engineering Management Master Class,
https://www.engineersaustralia.org.au/portal/event/eea-public-course-system-safety-engineering-management-master-class

Engineering Safety Consultants, IEC 61508 Software Safety Training Course ,
http://www.esc.uk.net/

HCRQ, Software Safety Course, http://www.hcrq.com/

IEC 62279 Edition 2.0(2015), Railway applications - Communication, signalling and processing

systems - Software for railway control and protection systems
IET(2006), Competence Criteria for Safety related System Practitioners, Guidance provided by the IET in collaboration with the HSE and BCS
IET, Safety Critical Systems Course, http://conferences.theiet.org/scs/about/index.cfm
MIT Aeronautics and Astronautics Dept., System Safety for Software-Intensive Systems, http://sunnyday.mit.edu/announce09.html
P. Bourque and R.E. Fairley(2014), eds.,IEEEComputerSociet;www.swebok.org.
PILZ, Using PAScal Safety calculator Software Course, https://www.pilz.com/en-GB/company/news/articles/073271
Technische Universität München, Techniques for System Safety Analysis Course, www4.in.tum.de/index.shtml
The University of Queensland, Australia, System Safety Engineering, https://www.uq.edu.au/study/course.html?course_code=ENGG4020
UCLA Extension, Safety-critical Software, http://shortcourses.uclaextension.edu/819-364/
University of York, Computers & Safety: University of York https://www.cs.york.ac.uk/postgraduate/modules/casa.html

부록1 : 기술용어정리

- IEC 61508 제목은 “전기/전자/프로그램 가능한 전자 안전 관리 시스템의 기능 안전(Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES))” 으로, 산업에 적용되는 IEC 국제표준이다.
- ISO 26262 제목은 “자동차 기능 안전(Road vehicles – Functional safety)“이며 자동차에 탑재되는 E/E (Electric & Electronic) 시스템의 오류로 인한 사고방지를 위해 ISO에서 제정한 자동차 기능 안전 국제 규격이다.
- DO-178C 제목은 “항공 시스템 및 장비 인증에 대한 소프트웨어 고려 사항(Software Considerations in Airborne Systems and Equipment Certification)” 으로 항공기, 엔진, 프로펠러, 보조 파워 장비 등과 같은 항공용 시스템과 장비에 활용되는 소프트웨어의 인증을 위한 문서이다.
- EN 50128 제목은 “철도 응용프로그램 – 통신, 신호 및 처리 시스템 – 철도 제어 및 보호 시스템을 위한 소프트웨어 (Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems)” 으로 철도 산업 기능 안전 표준이다.
- IEC 60880 제목은 “원자력 발전소 – 안전에 중요한 계측 및 제어 시스템 – 범주 A 기능을 수행하는 컴퓨터 기반 시스템의 소프트웨어 측면(Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions)” 으로 원자력 분야 계측 제어 시스템 안전에 관한 IEC 국제 표준이다.
- MIL-STD-882 제목은 “미국방부 표준 실무 시스템 안전성 (Department of Defense Standard Practice System Safety)” 으로 미국방부의 모든 군부서와 방위기관이 사용할 수 있도록 승인된 문서이다.
- Safety Management Process (안전 관리 프로세스): 기능 안전을 관리하는 프로세스를 말한다.
- Functional Safety Development Process (기능 안전 개발 프로세스): 기능 안전을

구현하기 위한 개발 프로세스이다.

- **Functional Safety Management** (기능 안전 관리): 기능 안전을 관리하는 것으로, IEC 61508 표준은 기능 안전 관리가 정착되어 있을 것을 요구한다. (IEC 61508 Part 1 Clause 6에 기능 안전 관리 요구사항들이 기술되어 있다.) 기능 안전 관리 원칙은 생명과 재산을 보호하는 임무를 수행하는 소프트웨어 및 전자 시스템이 자신의 임무를 안정적으로 수행하도록 보장한다.

- **Safety Process Tailoring** 안전 프로세스를 안전 무결성 레벨 및 회사 별 관행이나 팀에 따라 조정하는 것을 말한다.

- **Safety Case** (안전 진술): 특정 응용시스템이 특정 운용 환경에서, 수용될 만큼 안전한지를(acceptably safe) 보이기 위한 의도로 작성되는, 증거(evidence)에 의해 지지되는 구조화된 논증(argument)이다.

- **Safety Engineering Method (STAMP/STPA)** 안전 공학 기법 중 하나인 STAMP(System-Theoretic Accident Model and Process)는 사고의 인과 관계를 분석하는 모델로, 안전 제약사항(Safety Constraints)과 계층형 제어 구조(A Hierarchical Safety Control Structure), 프로세스 모델(Process Models)이라는 세 가지 개념에 기반을 둔다. STPA(System-Theoretic Process Analysis)는 STAMP 모델의 안전성 분석을 위해 제시된 기법으로, 시스템을 위험으로 이끄는 원인을 밝혀내는 기법이다.

- **System Engineering Process** (시스템 공학 프로세스): 포괄적이고 반복적인 귀납적(recursive) 문제 해결 프로세스로서 통합된 팀에 의해 위에서 아래로 순서대로 적용된다. 요구사항 분석, 시스템 분석 및 제어, 기능 분석/할당, 설계 합성 등의 단계로 구성된다.

- **Software Engineering Process** (소프트웨어 공학 프로세스): 초기 고객 시작부터 완제품 출시까지 소프트웨어 작성을 관리하기 위해 선택된 모델이다. 선택된 프로세스는 일반적으로 분석, 설계, 코딩, 테스트 및 유지 보수와 같은 기법들을 포함한다.

- **Safety Analysis Techniques** (안전성 분석 기술): 시스템의 안전성을 분석하는 기술로 시스템 수준의 위험 요소(hazard)와 개별 구성 요소의 오류(failure) 간의 인과 관계를 찾는 것을 목표로 한다. 안전성 분석 기술의 예로 FMEA, FTA 기법 등이 있다.

- **Safety Goal Management** (안전 목표 관리): 하나 또는 그 이상의 위험한 사건(event)의 위험을 허용될 수 있는 수준까지 줄이기 위한 목적으로, 시스템에 할당된 최상위 수준의 안전 요구사항인 안전 목표들을 관리하는 것을 의미한다.

- **Software Requirement Analysis** (소프트웨어 요구사항 분석): 다양한 이해관계자의 상충할 수도 있는 요구사항을 고려하여 새롭거나 또는 변경된 소프트웨어에 부합하는 요구와 조건을 결정하고, 소프트웨어 요구사항을 분석, 문서화, 검증 및 관리하는 작업을 포함한다.

- **Software Architecture and Design** 에이전트가 원시적(primitive) 컴포넌트 집합을 사용하고 제약 조건을 따르면서, 목표를 달성하기 위한 소프트웨어 산출물의 명세를 작성하는 프로세스이다. 소프트웨어 설계는 '복잡한 시스템의 개념화, 구성, 구현, 위임 및 수정과 관련된 모든 활동' 또는 '소프트웨어 공학 프로세스에서 요구사항 명세 이후, 프로그래밍 전에 수행하는 활동'으로도 정의될 수 있다.

- **Change Impact Analysis** (변경 영향 분석) 변화의 잠재적 결과를 식별하거나 변화를 달성하기 위해 무엇을 수정해야 하는지를 추정하는 것이다.

- **SIL (Safety Integrity Level)** 안전 기능(Safety Function)의 안전 무결성 기준(SIL: Safety Integrity Level)은 제품의 안전 기능에 요구되는 신뢰도 수준이다. SIL은 안전 기능의 가장 낮은 신뢰도 수준인 SIL 1에서부터 가장 높은 신뢰도 수준인 SIL 4까지 4단계로 구분한다. 비교적 낮은 임계의 안전 기능에는 SIL 1이 적합하고, 임계가 매우 높은 안전 기능의 경우 SIL 3 또는 SIL 4가 요구된다.

- **Functional Safety Mechanism** (기능 안전 메커니즘): 안전 상태(safe state)로 도달하거나 유지하기 위해 결함(faults) 검출 또는 고장(failure) 제어를 수행하는 전기/전자장치의 기능이나 엘리먼트(elements) 또는 기타 기술로 구현된 기술적 해결책(technical solution)이다. (ISO 26262 내 정의)

- **Architecture-level FTA** FTA(Fault Tree Analysis)는 고장을 구체적으로 정의하고 그러한 사건을 발생시킨 가능한 모든 원인을 결정하는 체계적이고 연역적인 방법론으로, Architecture-level FTA는 아키텍처 수준에서 수행한 FTA를 말한다.

- **FMEDA** (Failure Modes, Effects, and Diagnostic Analysis, 고장 모드, 영향 및 진단 분석): 하위 시스템/제품 수준 고장률, 고장 모드 및 진단 기능을 얻기 위한 체계적

인 분석 기법이다.

- **SW-FMEDA** (Software Failure Modes, Effects, and Diagnostic Analysis): 소프트웨어 대상 FMEDA 기법이다.
- **Coding Standard** (코딩 표준) 해당 언어로 작성된 프로그램의 각 측면에 대한 프로그래밍 스타일, 방법 및 방법을 권장하는 특정 프로그래밍 언어에 대한 일련의 지침이다.
- **Code Refactoring** (코드 리팩토링) 외부 행동이나 결과의 변경 없이 코드의 구조를 재조정하는 것을 말한다. 주로 가독성을 높이고 유지보수를 편하게 한다. 버그를 없애거나 새로운 기능을 추가하는 행위는 아니다.
- **Code-level FTA** FTA(Fault Tree Analysis)는 고장을 구체적으로 정의하고 그러한 사건을 발생시킨 가능한 모든 원인을 결정하는 체계적이고 연역적인 방법론으로, Code-level FTA 는 코드 수준에서 수행한 FTA를 말한다.
- **Static Analysis** (정적 분석) 실제 실행 없이 컴퓨터 소프트웨어를 분석하는 것을 말한다.
- **Software Testing in Different Phase** 다른 개발 단계에서의 소프트웨어 테스팅을 의미한다.
- **Fault-injection Testing** (오류-주입 테스팅) 커버리지를 개선하기 위해 코드에 오류를 주입하는 소프트웨어 테스팅 기술로서, 개발된 소프트웨어의 강건성(robustness)을 위해 보통 스트레스 테스트와 함께 사용된다.
- **Host-target Testing** (호스트-타겟 테스팅): 테스트 대상 응용프로그램이 정상적인 타겟 환경에서 실행되고 생성된 테스트 결과가 별도의 호스트 플랫폼에서 평가 및 분석되는 테스팅이다. 호스트-타겟 테스팅은 호스트 플랫폼에서 개발되고 크로스 컴파일 되고 특정 타겟 프로세서로 다운로드 되는 임베디드 시스템에 대한 테스트를 용이하게 한다.
- **Test Case Management** (테스트 케이스 관리): 테스트 결과 및 테스트 문서와 같은 테스트 자산 및 산출물들을 체계화하여 손쉬운 접근과 사용을 가능하게 하는 방법이다.

- Static Testing (정적 테스팅): 프로그램의 코드와 관련 문서를 조사하는 것을 포함하지만 프로그램을 실행하지 않아도 되는 소프트웨어 테스트 방법이다. 수동으로 수행하거나 다양한 소프트웨어 테스트 도구를 사용하여 수행 할 수 있다. 코드 분석, 인스펙션(inspection), 코드 검토(review) 및 워크 스루(walk through) 등의 기법이 정적 테스팅에 해당한다.

- Dynamic Testing (동적 테스팅): 코드의 동적인 행위에 대한 테스팅 기법으로, 동적 테스팅에서 소프트웨어는 실제로 컴파일 되고 실행되어야 한다. 수동으로 또는 자동화 된 프로세스를 사용하여 특정 테스트 케이스를 실행하여 입력 값을 제공하고 출력 결과가 예상대로 되는지 확인하는 작업이 소프트웨어와 함께 수행된다.

- Logical Proving (논리적 증명): 정형 증명(formal proof)은 (정형 언어의 경우 well-formed formula라고 불리는) 공식의 유한한 시퀀스이며, 각 공식은 공리(axiom) 또는 가정(assumption)이거나 추론 규칙에 따라 시퀀스 앞에 있는 공식을 뒤따른다.

- Program Slicing (프로그램 슬라이싱): 슬라이싱 기준이라 불리는, 어떤 관심 지점에서의 값에 영향을 줄 수 있는 프로그램 구문들의 집합(즉, 프로그램 슬라이스)을 계산하는 것을 말한다.

- Formal Verification (정형 검증): 수학적 방법을 사용하여 특정 정형 명세 또는 속성에 대하여 시스템의 기초가 되는 의도된 알고리즘의 정확성을 입증하거나 반증하는 행위이다.

- HILS (Hardware-in-the-Loop (HIL) simulation) 복잡한 실시간 임베디드 시스템의 개발 및 테스트에 사용되는 기술로, HIL 시뮬레이션은 제어 하에 있는 플랜트의 복잡성을 테스트 플랫폼에 추가함으로써 효과적인 플랫폼을 제공한다. 제어 하에 있는 플랜트의 복잡성은 모든 관련된 동적 시스템의 수학적 표현을 추가함으로써 테스트 및 개발에 포함된다. 이러한 수학적 표현을 '플랜트 시뮬레이션'이라고 하며, 테스트 할 임베디드 시스템은 이 플랜트 시뮬레이션과 상호 작용한다.

- SILS (Software-in-the-Loop (SIL) simulation) 임베디드 소프트웨어 개발 및 테스트에 사용되는 기술로, SIL 시뮬레이션은 컴파일된 산출물 소프트웨어 코드를 시뮬레이션 모델에 포함하는 것을 말한다.

- Traceability Analysis (추적성 분석) 시스템 구성 요소와 계층상 상위 또는 하위에 있는 다른 시스템 요소간의 상호 관계 네트워크를 통해 앞뒤로 추적하는 프로세스입니다.

- Reachability Analysis (도달성 분석) 허용된 규칙 또는 변환을 가지는 (잠재적으로 무한한 상태) 계산 시스템이 주어졌을 때, 시스템의 특정 상태가 시스템의 주어진 초기 상태에서 도달 가능한지 여부를 분석하는 것을 말한다.

- Predicate Calculus (술어 계산) 수학, 철학, 언어학 및 컴퓨터 과학에서 사용되는 정형 시스템 모음으로, 비논리적(non-logical) 객체에 대해 정량화된 변수를 (quantified variables) 허용하고 그러한 변수를 포함하는 문장의 사용을 허용한다.

- Modeling and Simulation (모델링 및 시뮬레이션) 모델(시스템, 엔티티, 현상 또는 프로세스에 대한 물리적, 수학적 또는 다른 논리적 표현)을 시뮬레이션(모델을 정적으로나 시간에 따라서 구현하기 위한 방법)의 기초로 사용하는 것을 말한다. 관리적 또는 기술적 의사 결정을 한 기초 자료로 활용된다.

- Safety Audit (안전 감사) 감사는 구현된 프로세스에 대한 조사/검사를 의미한다. ISO 26262에서는 기능 안전 감사(functional safety audit)를 위해 1 명 또는 그 이상의 인원이 1회 이상의 기능 안전 감사를 수행하도록 임명되어야 한다. 임명된 사람들은 기능적 안전에 필요한 프로세스의 실행에 대한 평가를 포함하는 보고서를 제공해야 한다.

- Configuration Management Planning (형상 관리 계획) 형상 관리란 형상 항목을 식별하여 그 기능적 물리적 특성을 문서화하고, 그러한 특성에 대한 변경을 제어하고, 변경 처리 상태를 기록 및 보고하고, 명시된 요구사항에 부합하는지 확인하는 기술적이고 관리적인 감독, 감시 활동으로, 형상 관리 계획은 이러한 활동들에 대한 계획을 세우는 것을 말한다.

- Configuration Management System (형상 관리 시스템) 형상 관리를 지원하는 시스템을 말한다.

- Defect Management (결함 관리) 결함 예방, 결함을 가능한 빨리 발견, 결함의 영향력을 최소화 하는 것을 목표로, 결함을 관리하는 것을 말한다. 결함은 submitted, opened, assigned, resolved, closed, reopened, duplicate, postponed 등의 상태를 거

치게 된다.

- **Orthogonal Defect Analysis ODC** (Orthogonal Defect Classification)는 각 소프트웨어 결함의 의미를 신속하게 포착하는 체계로, ODC 데이터 분석은 소프트웨어 수명주기(설계, 개발, 테스트 및 서비스)의 다양한 단계와 제품의 성숙도를 평가하는 데 유용한 진단 방법을 제공한다.

- **Hazard Analysis & Risk Assessment** (HARA, 위해도 분석 및 위험 평가) 시스템의 오작동에 의해 발생할 수 있는 위험 사건(Hazardous Event)들을 분석하고, 위험으로 인한 잠재적인 인명 피해, 신체 상해, 경제적 상해 및 재산 피해를 측정하는 프로세스이다.

- **HAZOP** (Hazard & Operability Study, 위험 및 운용성 연구) 인력 또는 장비에 위험을 초래할 수 있는 문제를 식별하고 평가하기 위한 목적으로 수행하는, 계획되거나 존재하는 프로세스 또는 작업에 대한 구조화되고 체계적인 검사이다.

- **FMEA** (Failure Mode & Effect Analysis, 고장 모드 및 영향 분석) 제품, 공정의 잠재적 고장과 그 고장의 영향을 인식 평가하고 잠재적 고장 발생의 기회를 제거하거나 줄일 수 있는 조치를 파악하기 위한 방법론이다.

- **SW-FMEA** (Software Failure Mode & Effect Analysis) 소프트웨어 대상 FMEA이다.

- **FTA** (Fault Tree Analysis, 결함수 해석) 고장을 구체적으로 정의하고 그러한 사건을 발생 시킨 가능한 모든 원인을 결정하는 체계적이고 연역적인 방법론이다.

- **SW-FTA** (Software Fault Tree Analysis) 소프트웨어에 대한 FTA이다.

부록2 : 설문지

통계법 제33조(비밀의 보호)에 의해 본 조사에서 개인의 비밀에 속하는 사항은 엄격히 보호됩니다

ID			

SW 안전 분야 재직자 교육 수요조사

귀 기관의 무궁한 발전을 기원합니다.

저희 소프트웨어정책연구소(SPRI)는 SW안전에 필요한 역량을 조사하고 실수요자의 기대사항 조사를 통해 재직자 중심 SW 안전의 역량 강화를 위한 정책 기초자료로 활용하고자 재직자 교육 수요조사를 수행하고 있습니다.

응답해주신 내용이 소중한 정책 자료로 반영될 수 있도록 바쁘시더라도 잠시만 시간을 내서 조사에 협조해 주실 것을 부탁합니다.

귀 기관의 성의 있는 답변은 SW 안전 분야 재직자 역량 제고를 위한 중요한 기초자료로 활용될 것입니다. 본 조사의 결과는 통계법 제8조에 의거하여 비밀이 보장되며, 설문에 대한 모든 응답과 개인적인 사항은 철저히 비밀과 무기명으로 처리되고 통계분석 목적 외에는 절대 사용되지 않습니다.

여러 가지 업무로 바쁘시겠지만 소중한 의견 개진을 부탁드립니다.

■ 주관기관 : 소프트웨어정책연구소, 솔루션링크

■ 조사기관 : (주)리서치랩

SPRi 소프트웨어정책연구소

기업현황	기업명		종업원 수	전체	명
				SW 개발 인력	명
	기업 규모	①중소기업 ②중견기업 ③대기업 ④공기업 ⑤공공기관			
	주력산업분야	①원자력 ②전력 및 수력 ③기타 에너지 ④철도 및 지하철 ⑤스크린도어 ⑥공항 ⑦항공 ⑧국방 ⑨항만 ⑩로봇 ⑪자동차 및 미래자동차 ⑫IoT ⑬의료 ⑭엘리베이터 ⑮기타기반시설 ⑯기타( )			
	기업내 역할	①실무 개발자 ②관리자/경영층			

작 성 책임자	이 름		전화번호	
	소속부서		직 위	
	이메일			

소프트웨어 안전성 확보	▣ 소프트웨어가 내장/연계되어 작동하는 전체 시스템/제품의 안전사고를 방지하기 위하여 해당 소프트웨어를 개발하는 것을 의미합니다. 이를 위해서는 시스템 수준의 잠재 사고 위험에 대한 위험 분석, 작동 환경/시스템/인적 위험 오류 발생시의 시스템을 안전 상태로 전이하는 소프트웨어 메커니즘을 의미합니다. 이를 통하여 소프트웨어 안전이 확보됩니다.

[문1] 소프트웨어 안전성 확보를 위하여 가장 중요한 기술적 활동은 무엇이라고 생각하십니까 ?

① 소프트웨어 테스팅 ② 코딩 표준 준수 ③ 위해도 분석
④ 요구사항 명세 ⑤ 안전 메카니즘 구현 ⑥ 기타( )

[문2] 귀사가 개발 하는 제품/시스템의 소프트웨어 안전성 확보가 중요합니까?

① 불필요 ② 불필요하나 확보되면 좋다 ③ 중요하지만 사업에 영향은 적다
④ 매우 중요하다 ⑤ 필수요건이다

[문3] 귀사가 개발 하는 제품/시스템에서 소프트웨어 중요성(역할적 비중)은 어떠합니까 ?

① 중요하지 않다 ② 핵심 기능은 아니고 편의 기능 역할이다 ③ 핵심 기능의 일부를 담당 한다
④ 제품의 두뇌에 해당한다

[문4] 귀사는 소프트웨어 안전 관련 요건이 귀사 제품의 인.허가 혹은 수출 요건에 필수적입니까 ?

① 예 ② 아니오

[문5] 귀 부서에서는 2010년 1월 이후 소프트웨어 안전 관련 교육을 실시한 적이 있습니까?

① 실시했었음 ☞ 문 5-1으로 ② 실시할 필요 및 계획이 있음 ☞ 문 5-1으로
③ 실시하지 않았음 ☞ 문 5-10으로

문5-1. (문5에서 ①혹은 ②번 응답자) 귀 부서에서 교육수요가 발생하는 원인을 2가지만 응답해 주십시오.

① 소프트웨어 안전 사고 방지를 위해서 ② 발주자의 요청 때문에
③ 안전관련 법/제도/표준의 준수요건때문에 ④ 내부 역량 확보 차원에서
⑤ 수출 요건 때문에 ⑥ 기타(기재:)

문5-2. (문5에서 ①혹은 ②번 응답자) 귀 부서에서 역점을 두고 시행하는 교육방식을 2가지만 응답해 주십시오.

① 사내 방문 교육 ② 외부 교육기관 파견 ③ 외부 단기 교육
④ 워크샵 및 세미나 ⑤ 작업순환 ⑥ 기타(기재 :)

문5-3. (문5에서 ①혹은 ②번 응답자) 귀 부서에서 교육을 실시하는데 있어서 중요하게 생각하는 내용은 무엇인지 2가지만 응답해 주십시오.

① SW안전관련 개념/최신 기술 동향 ② 시스템 안전사고에 대한 잠재 위험 분석
③ SW안전 분석, 설계, 검증 관련 기술 ④ 안전사고 예방을 위한 운용/사용
⑤ 소프트웨어 안전 관련 표준(ISO,IEC 등) ⑥ 발주자/표준 요건 대응 및 기술 영업
⑦ 소프트웨어 안전 관련 인식제고 등
⑧ 체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)
⑨ 기타(기재 :)

문5-4. (문5에서 ①혹은 ②번 응답자) 귀 부서에서 교육과정 및 기관 선정시 가장 중요한 기준은 무엇입니까?

① 교육시간 ② 교육비용 ③ 교육기관의 지명도
④ 지리적 근접성 ⑤ 교육적합성 ⑥ 교육내용
⑦ 기타(기재 :)

문5-5. (문5에서 ①혹은 ②번응답자) 귀 부서에서 소프트웨어 안전 혹은 품질관련 교육을 받은 기관은 어디이며, 교육에 대한 만족도는 어느 정도입니까? 해당란에 √ 표기해 주시기 바랍니다.

교육기관	교육 참여여부		교육 만족도		
	참여	미참여	만족	보통	불만족
회사내부 자체적으로 실시	①	②	①	②	③
학교(대학, 전문대학 등)	①	②	①	②	③
공공기관/협회	①	②	①	②	③
SW 안전분야 관련 전문 기업	①	②	①	②	③
외부 전문가 초청 내부 교육	①	②	①	②	③
기타(기재 :)	①	②	①	②	③
기타(기재 :)	①	②	①	②	③

문5-6. (문5에서 ①혹은 ②번 응답자) 귀 부서에서는 교육 관련 정보를 주로 어디서 얻습니까?

① 회사 추천 ② 인터넷 정보 검색
③ 관련협회(HRD 홈페이지, 공문) ④ 노동부(워크넷, 종합고용지원센터 등)
⑤ 공공.민간 직업훈련기관 홈페이지 ⑥ 광고(신문, TV, 지하철, 버스 등)
⑦ 기타(기재 :)

문5-7. (문5에서 ①혹은 ②번 응답자) 귀 부서의 교육활동 수준이 적절하다고 보십니까?

① 적당함 ② 많음 ③ 부족함 ④ 모르겠음

문5-8. (문5에서 ①혹은 ②번 응답자) 귀 부서에서 교육을 실시하는데 있어서 가장 큰 걸림돌은 무엇입니까?

① 조업차질(업무공백) ② 적절한 프로그램이 없어서 ③ 교육예산 부족
④ 임직원의 무관심 ⑤ 기타(기재 :) ⑥ 없음

문5-9. (문5에서 ①혹은 ②번 응답자) 귀기관에는 소프트웨어 안전 분석/위험분석 도구(FMEA, FTA지원 도구를 의미함)을 보유 및 사용하고 계십니까?

① 예 ② 아니오

문5-10 (문5에서 ③번 응답자) 귀 부서에서 '10년 1월 이후 교육을 실시하지 않은 이유는 무엇입니까?

① 근로자가 필요한 역량/숙련을 갖추고 있어서 ② 필요한 교육과정이 없어서
③ 교육비용이 많이 들어서 ④ 업무부담 및 생산차질이 우려되어서
⑤ 업무난이도가 직업훈련이 필요하지 않아서 ⑥ 교육효과에 대해서 회의적
⑦ 인근에 원하는 교육기관이 없어서 ⑧ 교육에 대한 정보를 얻기 힘들어서
⑨ 기타(기재 :)

[문6] 귀사에서 소프트웨어 안전 기술 관련 교육을 실시 한다면 예상 교육 인원은 몇 명 입니까?

(명)

[문7] 귀사에서 소프트웨어 안전 기술 관련 교육시 선호하는 교육방법은 무엇입니까?

① 이론 중심 ② 실습 중심 ③ 현장 체험

④ 기타(기재 :)

[문8] 귀사에서 소프트웨어 안전 기술 관련 교육 참여시 다음 항목 중 가장 중요하게 생각하는 항목은 무엇입니까?

① 교육 과정의 수준 ② 강사진의 구성 및 교육 능력 ③ 산업 도메인 특성의 반영성

④ 교육 장소 및 환경 ⑤ 기타(기재 :)

[문9] 귀 부서에서 교육이 가장 필요로 하는 직급은 누구입니까?

① 신입사원 ② 대리급 이하 직원 ③ 과장급 이상 직원 ④ 부서장 및 경영진

[문10] 귀사에서 교육이 가장 필요로 하는 직무은 누구입니까?

① 관리자 ② 개발자 ③ 도메인 ④ 안전전문가

⑤ 기타(기재 :)

[문11] 귀 부서에 신입사원이 입사하여 SW안전 관련 전문인력이 될 때까지 소요기간은 어느 정도 된다고 생각하십니까? (년)

[문12] 귀 부서에서는 2017년에 교육을 실시할 계획을 가지고 있습니까?

① 있다 ② 없다

[문13] 귀 부서의 업무형태 및 교육의 성과를 고려하였을 때, 참여 가능한 시간대와 적정 교육기간은 어느 정도라고 생각하십니까? (각각 응답해 주십시오)

(1) 교육참여 가능 시간대	(2) 적정 교육기간
① 평일 오전	① 1일
② 평일 오후	② 2~3일
③ 평일 종일	③ 4~5일
④ 주말	④ 6~10일
⑤ 기타( )	⑤ 기타( )

[문14] 정부 주도의 소프트웨어 안전 교육과정을 개설한다면 교육 제도를 다음 중 어떤 방식으로 운영하는 것을 좋다고 생각하십니까?

① 안전관련 사업시 필수 의무 교육 ② 선택 교육 ③ 자격증(예. SW안전전문가)과 연계된 교육

[문15] SW 안전 관련 교육과정 선호도를 파악하기 위해 아래 각 코스별 선호도와 시급성을 상, 중, 하로 응답해 주십시오(잘 모를 경우 모름에 체크)

안전 기술분야	주요 내용	필요성			시급성			잘 모름
		상	중	하	상	중	하	
소프트웨어안전 인식제고	소프트웨어 안전 관련 최신 동향 및 인식 제고, 중요성 습득							
소프트웨어 안전 개념 및 기초 이론	소프트웨어 안전성에 대한 개념, 위험, 해저드(Hazard), 안전 등급 등에 대한 기본 이론 습득							
소프트웨어 안전 공학 방법론 개론	안전성 추론 방법, 주요 소프트웨어 안전 공학 방법론(STAMP/STPA등) 전반적 설명							
시스템 안전 분석	해저드(Hazard)의 식별 및 평가를 통해 안전목표(Safety Goal)를 도출하기 위한 전반적인 절차, FMEA, FTA							
시스템 안전 설계	기능안전 요구사항에 대한 이해를 바탕으로 이를 만족하는 시스템 설계 방법, FTA							
소프트웨어 안전 분석	SW 개발 수준에서 요구되는 기능안전 분석 요건의 이해 기능안전 분석 요건을 만족하는 임베디드 소프트웨어 분석 방법(SW FMEA)을 습득							
소프트웨어 안전 설계 및 구현	기능안전 분석 요건을 만족하는 임베디드 소프트웨어 분석 방법(SW FTA)을 습득. SW 안전 메커니즘 설계의 이해							
소프트웨어 코드 안전 분석	코드레벨의 소프트웨어 안전 메커니즘 이해 안전 코딩 표준 및 정적 테스팅							
소프트웨어 안전성 테스팅 기법	안전 메커니즘 검증 방안 습득. fault-injection testing							
고안전성 시스템들 위한 정형 기법	고안전성 수준을 요하는 소프트웨어에 대한 정형 기법(formal method)							
소프트웨어 안전 관련 국제 표준/규제	Defense (MIL-STD-882), Aerospace (DO-178C), Nuclear (IEC 60880), Railway (EN 50128), Automobile (ISO 26262), General (IEC 61508)							
안전표준 및 규제에 따른 안전공학기법 이론 교육	FTA, FMEA, HAZOP, HARA, Formal Verification, 테스팅, HILS & SILS, Traceability Analysis							
소프트웨어 안전성 관리	고안전성 소프트웨어 프로젝트 관리 및 안전 관리자 역할과 책임							
형상관리 및 결함 관리	소프트웨어 형상관리와 체계적인 결함 분석 및 관리 기법							
안전 품질 및 측정 관리	안전성 관련 품질 모니터링 방법과 이를 위한 각종 척도 및 측정 기법							
소프트웨어 안전 Audit	국제 표준에 의거한 소프트웨어 안전성 audit 방법 및 선진 사례							

[문16] 실질적으로 귀 부서에서 필요로 하는 교육프로그램으로 정부에서 추가적으로 개발, 운영해야 할 교육프로그램은 무엇이라고 생각하십니까? (주관식 응답)

♠ 귀사의 사업이 번창하시길 바라며, 본 조사에 협조하여 주심에 깊이 감사드립니다. ♠

부록3 : 심층인터뷰 질문지

SW 안전분야 재직자 역량 제고를 위한 교육(필요 기술) 수요조사

주관기관 : 소프트웨어정책연구소

귀 기관의 무궁한 발전을 기원합니다. 저희 소프트웨어정책연구소(SPRI)는SW안전에 필요한 역량을 조사하고 실수요자의 기대사항 조사를 통해 재직자 중심SW 안전의 역량 강화를 위한 정책 기초자료로 활용하고자 재직자 교육 수요조사를 수행하고 있습니다. 응답해주신 내용이 소중한 정책 자료로 반영될 수 있도록 바쁘시더라도 잠시만 시간을 내서 조사에 협조해 주실 것을 부탁합니다. 귀 기관의 성의 있는 답변은 SW 안전 분야 재직자 역량 제고를 위한 중요한 기초자료로 활용될 것 입니다. 본 조사의 결과는 통계법 제8조에 의거하여 비밀이 보장되며, 설문에 대한 모든 응답과 개인적인사항은 철저히 비밀과 무기명으로 처리되고 통계분석 목적 외에는 절대 사용되지 않습니다.

1. 기관 일반 현황

1-1 기관 현황

기업명		종업원수	전체	명
			SW개발인력	명
기업규모	중소기업( ) 중견기업( ) 대기업( ) 공기업( ) 공공기관( )			
주산업분야	원자력( ) 자동차/미래차( ) 철도/지하철( ) 항공( ) 드론( ) 기계/로봇( ) 의료( ) 기타( )			
기업내 역할	실무 개발자( ) 관리자/경영층( ) 이름:			

1-2 기업내에서 귀하의 주요 담당업무는 무엇입니까? (중복 선택 가능)

[개발기관의 경우]

소프트웨어 엔지니어( ) 하드웨어 엔지니어( ) 시스템 엔지니어( ) 프로젝트 관리자( ) 테스팅( ) 품질 보증( ) 안전 관리자(Safety Manager) ( ) 관리자/경영층( )

[발주/사용기관의 경우]

개발 담당자( ) 구매 담당자( ) 사용자( ) 운영자( ) 유지보수 담당자( )

품질보증 담당자( ) 관리자/경영층( )

2. 소프트웨어 안전 문화 및 환경 경영 조사

2-1 귀하가 속한 조직에서, 기능 안전(functional Safety)에 대하여 범조직차원에서 이해/숙지하고 있습니까? 있다면 이를 위해 어떠한 활동(사내 세미나 등)이 정기적으로 진행되고 있습니까?

2-2 귀하는 기능 안전에 대해 충분히 이해하고 있으십니까?

2-3 귀하의 기관이 속한 산업군의 기능안전 관련하여 안전 표준/규제가 존재 합니까?

2-4 귀 기관의 상급기관(발주기관, 감독기관)에서 사업 발주/감독 시, 기능 안전에 관련하여 개발요건에 명시하고 있습니까

2-5 귀 기관에서는 국제 표준에 준하는 소프트웨어 개발 프로세스가 사내 표준으로 정립되어 있습니까? 정립되어 있다면 적절히 사용되고 있습니까?

2-6 귀 기관에서는 전장부품/제어소프트웨어 개발 시 기능안전관련 표준 프로세스가 정립되어 있습니까? 정립되어 있다면 잘 사용되고 있습니까?

2-7 귀 기관에서는 소프트웨어 개발 프로세스, 전장부품/제어소프트웨어 개발 시 기능안전 관련 표준 프로세스를 재정/개정/배포 관리하는 전담 부서가 존재합니까?

2-8 귀 기관에서는 기능안전/소프트웨어 안전 분야 관리자(Safety Manager)가 존재 합니까?

2-9 귀 기관에서는 기능안전/소프트웨어 안전에 대하여 관련 지식 필요 시, 지식 습득이 원할 합니까? 원할하지 않을 경우 가장 중요한 원인이 무엇이라고 생각하십니까?

① 근로자가 필요한 역량/숙련을 갖추고 있어서	② 필요한 교육과정이 없어서
③ 교육비용이 많이 들어서	④ 업무부담 및 생산차질이 우려되어서
⑤ 업무난이도가 직업훈련이 필요하지 않아서	⑥ 교육효과에 대해서 회의적
⑦ 인근에 원하는 교육기관이 없어서	⑧ 교육에 대한 정보를 얻기 힘들어서
⑨ 기타(기재:)	

2-10 상기2번 항목들 중 애로사항이 있다면 어떠한 것들이고, 정부의 지원 요청이 있다면 어떠한 것들이 있습니까? (예. 교육 지원, 교육 의무화, 자격증 제도화, 등)

3. 기능 안전 관리 지식 수요 조사

3-1 프로젝트 수행시 Safety Manager가 존재하여, PM 과 함께 기능안전에 대한 분석/설계/구현/검증 계획에 참여 합니까?

3-2 프로젝트 계획과 별도로, 기능안전계획(Functional Safety Plan)을 수립하는 방법을 알고 있습니까?

3-3 제품에 대한 Safety Goal 이 식별되고 이에 대한 Safety Case를 전개하는 방법을 알고 있습니까?

3-4 Safety Audit 활동 방법을 알고 있습니까? 알고 있다면 적용 하고 계십니까?

3-5 외주 개발 발주의 경우 프로젝트 요구사항에 SIL레벨 할당과 Safety Requirements를 적절이 기술하고 있습니까?

3-6 귀사가 개발 또는 사용하는 제품의 Safety Critical 필드 오류의 발생 원인이 하드웨어(전장회로)와 소프트웨어 간 어떠한 비중을 가지고 있습니까?

하드웨어( ) %, 소프트웨어( )% 기타_____ ( )%

4. 기능 안전 구현 기초 지식 수요 조사

4-1 기존 제품의 변경 프로젝트의 경우 안전성 측면에서의 변경 영향분석 방법을 충분히 숙지하고 계십니까?

4-2 개발 초기 단계에 제품 수준에서의 통계적인Hazard Analysis와 Risk Assessment 방법을 충분히 알고 계십니까?

4-3 SIL (Safety Integrity Level) 알고 계십니까? 알고 계신다면 어떤 방법으로 부품/컴포넌트 별로 SIL을 정하고 개발 하십니까?

4-4 제품SIL에 따른 부품별 SIL Decomposition방법을 충분히 알고 계십니까? 알고 계시다면 실제 적용하고 있습니까?

4-5 안전 목표에 따른 Safety Concept 도출 방법을 알고 계십니까?

4-6 제품의 안전 요구사항을 하드웨어 컴포넌트, 소프트웨어 컴포넌트로 체계적으로 분할 할당하여 구현하고 있습니까?

4-7 다양한 기능 안전 요구사항에 대한 검증 방법들에 대하여 충분한 지식을 가지고 계십니까?

5. 기능 안전 구현

5-1 제품의 기능 안전 요구사항을 반영하여 시스템의 안전 아키텍쳐를 설계하는 방법을 충분히 보유하고 계십니까?(System Modeling, Domain Reference Arch. Etc)

5-2 시스템 안전 분석 시 어떠한 방법을 사용하고 계십니까?(FMEA, FTA)

5-3 시스템 분석시 안전 메카니즘을 정의하는 담당자가 하드웨어(전자 회로) 안전 메카니즘과 소프트웨어 안전 메카니즘에 대한 지식을 충분히 숙지하고 있습니까?

5-4 소프트웨어 담당자는 안전 메카니즘 구현시 하드웨어 엔지니어, 시스템 엔지니어와 지속적인 협의를 하고 있습니까?

5-5 소프트웨어 담당자는 소프트웨어 안전 분석 방법을 충분히 숙지하고 있습니까?(SW-FMEA, SW-FTA)

5-6소프트웨어 담당자는 소프트웨어 안전 메카니즘에 대한 구현 방법을 충분히 숙지하고 있습니까? (diversity, program flow monitoring, message checking, etc...)

5-7 주로 사용하는 소프트웨어 안전 메카니즘은 어떠한 것들입니까?

5-8 소프트웨어 개발 단계별(분석, 설계, 구현, 테스팅) 생명주기(폭포수, 반복적, agile등)를 정하고 이에 따라 구현하고 있습니까?

5-9 소프트웨어 개발 각 단계(분석, 설계, 구현, 테스팅)에서 수행하는 기능 안전에 대한 활동이 별도로 정의되어 사용하고 있습니까?

5-10 소프트웨어 개발 단계별 기능 안전과 관련된 도구/개발 환경 등을 구축하고 사용하고 계십니까?

5-11 정형화된 소프트웨어 모델링 기법(SysML, UML, Statechart등) 을 사용하고 있습니까?

5-12 소프트웨어 formal modeling, formal verification기법들을 알고 있으십니까?

5-13 프로그래밍 언어 선정 시 런타임 에러 핸들링 등에 대한 런타임 잠재 오류에 대한 고려가 이루어지고 있습니까?(C 이외의 다른 언어 사용시)

5-14 소프트웨어 기능 안전 요구사항에 대한 명세 활동이 진행됩니까?

5-15 소프트웨어 아키텍쳐 단계의 기능 안전성 분석이 이루어지고 있습니까?(설계 단계 SW-FMEA, SW-FTA)

5-16 소프트웨어 구현 단계에서 기능 안전성 분석이 이루어지고 있습니까?(구현단계 SW-FMEA, SW-FTA)

5-17 재사용, 변경되는 소프트웨어 모듈의 경우 어떠한 방법으로 기능 안전을 확보하십니까?(Component Qualification, Prove-in-Use)

5-18 소프트웨어 개발 단계별로 기능안전 수준에 따른 검증 활동/리뷰이 이루어지고 있습니까?

5-19 소프트웨어 테스팅 단계에서 기능 안전에 대한 특화된 검증 기법을 충분히 확보하고 있습니까?(Fault-Injection Testing)

5-20 소프트웨어의 실시간성 분석, 리소스 충돌 분석, 통신 자원 분석 등이 이루어지고 있습니까? (Worst-case execution time analysis, etc)

5-21 소프트웨어 일반 기능 외 기능안전 요구사항에 대한 추적활동이 이루어지고 있습니까?

5-22 소프트웨어 단위 시험/통합 시험/시스템 시험이 구분되어 이루어 지고 있습니까?

5-23 소프트웨어 단위 시험/통합 시험/시스템 시험 시 기능 안전 요건에 대한 시험이 이루어 지고 있습니까?

5-24 소프트웨어에 대한 정적 테스팅을 하고 계십니까?

5-25 소프트웨어 시스템 시험 시 목표 하드웨어 상에서 충분히 시험이 되고 있습니까? 환경상 어려우면 이유가 무엇입니까?

5-26 소프트웨어 개발에 사용되는 도구(사용 및 내부 개발)들에 대한 안전성 검증이 이루어지고 있습니까?

6. 양산/구축, 이관 및 운영

6-1 양산/구축 과정 중 제품 요구사항 변경이 발생할 경우 기능안전 관점에서 요구사항에 대한 변경 영향 분석과 관리가 이루어 지고 있습니까?

6-2 운영자들을 위한 Safety Manual이 개발되고, 교육되고 있습니까?

6-3 필드 결함 발생시, Defect Analysis활동이 충분이 이루어지고 있습니까? 이루어지고 있다면 어떠한 Defect 분류체계를 가지고 있습니까?

6-4 동일 결함 재발 방지를 위한 변경관리, 유사 파생 제품의 변경 파급 관리가 적절히 이루어지고 있습니까?

7. **교육 과정에 대한 일반 건의**

7-1 Safety Critical SW의 기능안전교육이 필요하다고 생각하십니까? ----------(O , X)
그렇게 생각한 이유가 무엇입니까?

① 소프트웨어 안전 사고 방지를 위해서	② 발주자의 요청 때문에
③안전관련 법/제도/표준의 준수요건때문에	④ 내부 역량 확보 차원에서
⑤ 수출 요건 때문에	

7-2 적절한 교육 시간은 어느 정도라고 생각하십니까?

(1) 교육참여 가능 시간대	(2) 적정 교육기간
① 평일 오전	①1일
② 평일 오후	②2~3일
③평일 종일	③4~5일
④ 주말	④6~10일
⑤ 기타( )	⑤ 기타( )

7-3 적절한 교육 시기는 언제라고 생각하십니까? 그리고 그 이유가 무엇입니까?

7-4 교육에서 가장 중요하게 다루어야 할 과목이 무엇이라고 생각하십니까?

①SW안전관련 개념/최신 기술 동향	② 시스템 안전사고에 대한 잠재 위험 분석
③SW안전 분석, 설계, 검증 관련 기술	④ 안전사고 예방을 위한 운용/사용
⑤ 소프트웨어 안전 관련 표준(ISO,IEC 등)	⑥ 발주자/표준 요건 대응 및 기술 영업
⑦ 소프트웨어 안전 관련 인식제고 등	⑧ 체계적인 소프트웨어 개발 기술(분석, 설계, 구현, 검증)
⑨ 기타	

7-5 상기 교육의 필요 부분에 대한 시급성이 어떠합니까?

필요하나 시급하지 않음( ) 시급( )매우 시급( )

7-6 상기 교육이 귀하가 개발/운영하는 제품의 안전 인.허가 요건상에서도 필요합니까

7-7 기능 안전 교육의 진행 방식이 어떤 형태로 운영되길 원하십니까?

① 사내 방문 교육	② 외부 교육기관 파견	③ 외부 단기 교육
④ 워크샵 및 세미나	⑤ 작업순환	⑥ 기타

연구보고서 2016-019

소프트웨어 안전 분야 재직자 역량 제고를 위한 교육 커리큘럼 개발에 관한 연구

2017년 05월 인쇄
2017년 04월 발행

발행처 정보통신산업진흥원 부설 소프트웨어정책연구소
경기도 성남시 분당구 대왕판교로712번길22 A동 4층
Homepage: www.spri.kr

ISBN : 978-89-6108-380-5

주 의

1. 이 보고서는 소프트웨어정책연구소에서 수행한 연구보고서입니다.
2. 이 보고서의 내용을 발표할 때에는 반드시 소프트웨어정책연구소에서 수행한 연구결과임을 밝혀야 합니다.

ISBN : 978-89-6108-380-5