

임베디드 소프트웨어 개발자를 위한 표준 Skill Set 정의 및 활용

국민대학교 | 임성수* · 임은진* · 최준수**
한국정보산업연합회 | 김기철

1. 서론

정보통신부의 IT839 정책의 일환으로 2003년 이후 임베디드 소프트웨어 인력 양성을 위한 다양한 지원 및 교육 사업이 진행되어 왔다. 그 동안의 인력 양성 사업의 취지는 국가 기간산업을 이루고 있는 임베디드 시스템 분야에서 필요한 인력을 공급하기 위해 실제적인 응용 기술에 대한 교육을 강화하여 현장에서 당장 활용할 수 있는 인력을 양성하기 위함이었다. 임베디드 소프트웨어 교육을 위한 교재 및 교구가 다양해지고 전국적으로 많은 대학에 임베디드 소프트웨어 교육을 위한 시스템이 갖추어졌다는 데에 지난 인력 양성 사업의 성과가 있었다.

그러나, 여전히 산업 현장에서는 임베디드 소프트웨어 인력 부족을 호소하고 있으며 특히 각 기업에서 원하는 기술 분야 및 수준을 갖춘 인력을 확보하기가 어렵다는 데에 의견을 같이 하고 있다. 표 1은 2005년 직업능력개발원에서 분석한 우리나라의 IT 인력 부족 현황이다. 비록 절대적인 임베디드 소프트웨어 분야 인력 수의 부족이라는 결과를 보이고 있지만 그 내용을 분석하면 기업에서 원하는 기술 수준을 갖춘 인력이 부족하다는 현상이 통계적으로 절대적인 수의 부족을 나타내는 결과를 보이고 있음을 알 수 있다. 임베디드 소프트웨어 분야의 인력 수급의 어려움은 비단 교육 현장에서의 교육 내용이 산업 현장에서 요구하는 그것과 일치하지 않음으로 인해서 생기는 문

표 1 한국 IT인력 현황[출처:직업능력개발원,2005]

구분	초급	중급	고급	수급차
IT Service	1,800	-800	-200	800
임베디드 SW	-5,200	-4,300	-2,500	-12,000
패키지 SW	300	5,400	-1,800	3,900
디지털 콘텐츠	4,300	-2,300	-1,000	1,000

* 정회원

** 종신회원

제라기보다는 대학에서 교육받은 인력들이 제대로 자신들이 강점을 가지고 있는 기술 분야 및 수준을 알릴 수 있는 방법 및 기업들이 그러한 부분을 파악할 수 있는 방법이 없다는 데에 기인한다.

한국 정보산업연합회 산하 임베디드소프트웨어산업협의회에서는 2007년 임베디드소프트웨어 Skill Set 및 표준이력서 양식 정의를 위한 1차년도 사업을 진행하였다[2]. 이 사업의 목적은 학교에서 배출되는 인력의 보유 기술 종류 및 기술 수준을 보다 정확하게 표현한 개인별 자료를 만들고 산업 현장에서는 이 자료를 활용하여 필요한 인력을 빠른 시간 내에 확보할 수 있는 체계를 만드는 데에 있다. 필자는 2007년 이 사업에 참여하여 1차 표준 이력서 양식 및 시범 적용을 수행하였다. 본 글에서는 임베디드소프트웨어산업협의회에서 2007년 수행한 임베디드소프트웨어 Skill Set 및 표준 이력서 양식 정의 사업의 결과물을 정리하고 이를 교육 현장에 활용할 수 있는 방안에 대해 논의한다.

본 글의 구성은 다음과 같다. 제 2장에서는 본 연구와 관련이 있는 기존 연구를 소개하고 3장에서는 본 연구의 목적과 2007년 수행한 연구의 결과를 설명한다. 4장에서는 본 연구의 향후 계획과 함께 결론을 맺는다.

2. 관련 자료

산업 현장에서 필요한 인력의 직무 수행 능력을 분석하고 이를 표현하는 방법은 이미 2006년 한국직업능력개발원에서 정의한 'IT 스킬 체계에 따른 SW 개발자 직무수행능력개발'[1]이라는 연구에서 논의된 바 있다. 이 연구의 목적은 첫째, IT분야의 전문인력이 되기까지의 숙달정도, 직무 경험의 특징, 학습한 내용(교육 및 훈련) 및 자격 등을 조사하고 이를 수준별로 구체화한다. 둘째, 개발자 차원에서는 경력개발, 구직, 진로탐색 등을 효율화 할 수 있는 기초 정보로 활용

표 2 임베디드 SW 개발 직무의 경력경로 [출처:직업능력개발원,2005]

path	입사 전	Level 1	Level 2	Level 3
근속년수/ 학력연수	16.9년	입사 - 1년차	2년차 - 4년차	5년차 - 8년차
신분/직위	4년제 졸	사원 연구원	사원 → 대리/주임 연구원 → 전임연구원	대리/주임 → 과장 전임연구원 → 선임연구원
직무	<p><관련 학과 예시></p> <ul style="list-style-type: none"> - 정보통신 - 정보공학 - 제어공학 - 지리학 - 정밀기계공학 - 전산통계 - 전산학 - 컴퓨터공학 - 자동차공학 - 응용물리 - 전기공학 - 전자계산기공학 - 전자계산학 - 항공전자 - 전자공학 - 수학 - 경영정보 - 물리학 	<p><주요 직무이행 경로></p> <p>① 임베디드 SW 개발</p> <ul style="list-style-type: none"> - 설계 요소별 구현 보조 - 개발자 단위테스트 보조 <p><세부 보조 업무내용 예시></p> <ul style="list-style-type: none"> - F/W, 모듈 구현 보조 - 장애 방지를 위한 예외 처리 수행, 사업자 스펙 시험 - 스펙에 맞게 회로 및 SW설계(시스템 관련) - 기능 설계 및 SW설계(임베디드 관련) - 코딩 	<p><주요 직무이행 경로></p> <p>① 임베디드 SW 개발</p> <ul style="list-style-type: none"> - 시스템 요구사항 수집 - 단위 개발 공정 조정 및 수립 - 단위 기능 설계 - 개발환경 구축 - 설계 요소별 구현 - 개발자 단위테스트 - 프로그램 수정 <p>② 시스템 SW 개발</p> <p><이행 가능 직무 경로></p> <p>① 응용 SW 개발</p> <p>② 패키지 SW 개발</p> <p><직무관련 세부 업무내용 예시></p> <ul style="list-style-type: none"> - SW개발 환경 SETUP - RTOS개발 환경 구축 - 제품의 기본 요구 사항 분석 - 요구사항명세 파악(CPU Architecture, Linux 커널, 컴파일 구조 등) - 다중 처리, UI설계 - 멀티미디어 데이터 처리 프로세스 설계 - 코딩 - 개발자 단위 테스트 	<p><주요 직무이행 경로></p> <p>① 임베디드 SW 개발</p> <ul style="list-style-type: none"> - 시스템 요구사항 정의 - 후보 아키텍처 식별 - 기능 설계 - 객체 설계 - 개발환경 구축 - 설계 요소별 구현 - 하드웨어 오류수정 요청 - 소프트웨어 통합 테스트 <p>② 시스템 SW 개발</p> <p><이행 가능 직무 경로></p> <p>① 응용 SW 개발</p> <p>② 패키지 SW 개발</p> <ul style="list-style-type: none"> - 코드 최적화 작업 - 최적화 기법 적용, 모듈 단위 구현, VXWORKS/리눅스 OS 구조 분석 및 API활용 - 모듈 설계, 인터페이스 설계 - 디버깅 및 검증 - 임베디드콘트롤러 및 OS환경에 대한 이해와 프로그램 구현 - MS프로젝트와 비주얼 소스 세시프의 형성 및 공정 관리 - 디지털 스코프를 통한 회로 검증
교육훈련		<p><교육훈련 예시></p> <p>Window Devile Driver/ 네트워크 ADMIN/자바 프로그래밍에 대한 기초</p>	<p>임베디드 메모리 최적화 기법/ ASIC설계를 위한 기초 툴 교육 /DSP칩셋 활용을 위한 프로그램 기술교육/C/C++ 언어활용을 통한 프로젝트 실무교육/회로기초/파형분석(Timing 분석)</p>	<p>객체지향 분석설계/형상관리/ UML/OOP/임베디드 리눅스 설계과정/임베디드 프로그래밍/임베디드 운영체제 개론/SW 테스트 기법</p>
기타조건 (자격 및 역량)		<p><자격증 예시></p> <p>MCP</p>	<p>리눅스마스터 1급 MCSE MCP</p>	<p>CISA UML 전문가</p>

한다. 한편, 기업들의 입장에서는 신규인력을 채용하거나, 기존인력의 배치할 수 있는 기본 골격 등의 인사 관리에서 활용한다. 마지막으로 교육, 훈련기관에

서는 교육 커리큘럼을 개발할 수 있는 자료로 활용한다. 표 2는 한국직업능력개발원의 연구 결과로 도출된 소프트웨어 개발자들의 경력 개발 경로를 보인다.

본 연구에서는 IT분야의 전문 인력의 직무를 중심으로 개발자 능력을 측정하고 있기 때문에 기업에서 활용하기에 부족한 측면이 있다. 예를 들면, 기업에서는 인력 채용 시 개발자의 능력을 검증하기 위한 기준 및 이에 따른 능력 수준을 평가하고 싶어 하는데 본 연구 결과는 개발자를 주로 경력 년수를 기준으로 능력 수준을 구분하고 있어서 현실성이 떨어진다. 또한, 측정 항목이 기술적 항목이 아닌 다소 추상적인 항목이기 때문에 현장에서 실제 활용되기에 무리가 따른다.

해외 유사 연구 결과로는 일본의 ETSS(Embedded Technology Skill Set)를 들 수 있다. ETSS는 임베디드 시스템 관련 기술 보유 수준을 대략적으로 분류하여 직무 능력을 파악하기 위한 연구 결과로서 현재 일본 내 직무 능력 표준화 추진 단계에 있다. ETSS는 각 보유 기술을 세부적으로 파악하기 보다는 전체적인 기술 적용 능력을 파악하는 데 초점을 맞추고 있으며 특히 기술적 능력보다는 개발 프로세스의 경험 수준을 평가하는 데 더 큰 비중을 할애한다.

본 글의 바탕이 되는 2007년 임베디드소프트웨어 산업협의회에서 수행한 ESSS(Embedded Software Skill-set Standard)는 임베디드 소프트웨어를 활용하는 각 산업 분야에서 필요한 엔지니어의 보유 기술을 보다 세부적으로 분류하고 각 기술의 수준 평가를 하기 위한 기준을 마련하는데 초점을 두고 있다는 점이 위에서 언급한 한국직업능력개발원의 연구 결과와 일본의 ETSS와 차별화된다.

Skill Set에 대한 연구는 영국에서 먼저 시작되었다. 애초의 연구는 영국에서 동남아, 인도 등의 저급 인력의 활용을 위해 도입했으며, 표준 임금 가이드 역할을 하였다. 영국의 경우 중소기업의 인력은 대개 3~4개 정도의 기술 수준으로 분류하고, 대기업의 인력은 7~9개 정도의 기술 수준으로 분류하고 있다. 영국에서는 이러한 영국의 기술표준을 유럽의 표준으로 추진하기 위한 노력이 진행되고 있다.

3. 임베디드 소프트웨어 Skill-set Standard

3.1 연구 목적

기존 직무 중심의 연구의 단점을 극복하고자 제안된 ESSS의 연구 목적은 다음 세 가지이다.

가. 임베디드SW 인력의 보유 기술 분야와 기술 수준을 파악하는 기준 마련

임베디드 소프트웨어 관련 인력의 공급과 수요가 적절히 연결되지 않고 있어서 각 기업에서 필요한 인력

을 적절한 시점에 확보하기 어려운 현실이다. 특히, 재교육을 위한 투자 여력이 많지 않은 중소기업의 경우 당장 현장 투입이 가능한 기술 수준을 보유한 인력을 원하지만 채용 단계에서 대상 인력의 보유 기술 수준을 정확하게 파악하는 것은 거의 불가능하다. 따라서, 취업 지원자들과 기업 양자에게 취업 과정에서 참고할 만한 공통의 자료가 필요하며 기업의 수요 기술 종류 및 수준에 따라서 대학의 교육 내용도 그에 맞추어 적용해 나갈 수 있는 체계가 필요하다.

나. 체계적인 임베디드SW 인력 양성의 기틀 마련

임베디드 소프트웨어 분야는 전통적인 소프트웨어 분야와 달리 하드웨어 기술 및 소프트웨어 구현 기술을 동시에 보유하고 있어야 경쟁력을 발휘할 수 있는 분야이다. 따라서, 전통적인 소프트웨어 분야 인력의 보유 기술 수준을 파악하는 방법과 다른 방법이 필요하다. 인력 양성의 관점에서 하드웨어와 소프트웨어 분야의 융합적 교육이 필요하며 어떤 수준의 하드웨어 기술과 소프트웨어 기술의 교육 내용이 필요한지에 대한 정보는 해당 인력을 활용하는 산업 현장으로부터 도출되어야 한다. 따라서, 임베디드 소프트웨어 인력의 보유 기술 종류와 기술 수준을 파악할 수 있는 표준 이력서 양식을 정의하는 것은 체계적인 인력 양성 시스템을 만드는 데 도움을 줄 수 있을 것으로 보인다.

또한, 대학에서 관련 분야의 교육을 받는 학생들은 이러한 표준적인 이력서 양식 및 기술 수준 평가 자료를 바탕으로 자신의 경력 개발 경로를 계획하고 해당 경력 개발을 위한 교육 시스템 및 내용을 아울러 구축할 수 있을 것으로 보인다.

다. 현실 적용 가능한 기준 마련

기존 소프트웨어 인력 기술 수준 파악을 위한 연구의 결과는 연구 과정에서 기업들의 참여를 유도하지 못했고, 결과물 도출 후 기업들과의 피드백을 받지 못하였기에 그 결과물이 활발하게 적용되지 못했다. 이번 연구에서는 시장에서 인력 수요가 가장 큰 대기업을 비롯한 각 분야의 중견 기업들을 참여시켜 현실 적용 가능한 임베디드SW 산업의 인력 양성 기준을 제정하는 것이 목적이며, 기업의 의견 수렴 과정 및 중간 결과물에 기업 의견 반영 등의 절차 등을 통해 현실 적용 가능한 기준을 제시할 예정이다. 2007년 1차 연구 과정에는 대기업 및 중소기업에서 15명 이상의 전문가들이 참여하였다.

3.2 연구 목표

본 연구는 아래 열거된 세 가지 목표를 달성하기

위해 수행되었다. 2007년 수행한 연구 결과를 바탕으로 보다 발전된 연구 결과를 도출하기 위한 지속적인 연구를 수행할 계획이다.

1) 임베디드 소프트웨어 분야에서 필요한 기술표준을 정의한다: 임베디드 소프트웨어 분야에서 인재육성 및 인재활용을 위한 지표로서 사용될 수 있도록 임베디드 소프트웨어 관련 개발 기술을 체계적으로 정리한다.

2) 경력 기준을 마련한다: 임베디드 소프트웨어 개발에 관련된 직종과 구체적인 개발자의 능력 단계를 정의한다. 이러한 단계는 보유 기술 수준과 해당 분야에서의 경력 기간에 따라 결정된다.

3) 기술 수요에 맞는 교육과정을 제안한다: 임베디드 소프트웨어 분야의 기술 수요에 맞춰 각 분야에 맞는 교육과정을 제안하고 이를 교육기관에서 참고할 수 있도록 지원한다.

3.3 연구 결과

가. 임베디드 소프트웨어 개발자 표준 이력서의 구조

임베디드 소프트웨어 개발자 표준 이력서는 그림 1과 같은 구조로 이루어져 있다. 대분류 항목은 임베디드 소프트웨어 분야에서 필요한 기술을 분야별로 분류한 항목이고 소분류는 각 대분류 내에서 요구되는 세부 기술 항목이다. 각 소분류 기술 항목별로 기술 수준 평가를 통해 레벨1에서 레벨5까지 표기하도록 되어 있다. '비고'란에는 각 기술 수준에 대한 평가

결과와 함께 각 소분류 기술 항목에 대해 습득이나 구현 경험이 있는 기술 종류를 직접 나열한다.

표준이력서는 전반적으로 4~5레벨로 구성되어 있으며, 5레벨까지 정의된 소분류에서 평균적으로 3레벨이 학부를 졸업한 우수한 학생이 가지고 있어야 하는 수준을 나타내며, 5레벨은 석사를 졸업한 우수한 학생 또는 경력 1~2년차의 경력자가 가지고 있어야 하는 수준을 나타낸다. 표준 이력서의 대분류 항목은 현재 총 7개이며 이러한 대분류 항목은 향후 연구를 수행하면서 확대할 예정이다. 각 대분류에 대한 정의는 아래와 같다.

1) Programming Languages: 각 프로그래밍 언어에 대한 기술 보유 수준을 평가한다. 모든 소분류 항목별로 각 언어 활용 난이도 수준에 따라 기술 수준을 평가한다.

2) Development Tools: 각종 소프트웨어 개발 도구의 활용 수준을 평가한다. 각 개발 도구에 대한 활용 경험에 따라 기술 보유 수준 평가를 한다.

3) System Programming: 각 운영체제의 API를 활용한 시스템 소프트웨어 개발 기술 보유 수준을 평가한다. RTOS-based System Programming 소분류 항목에 대해서는 개발 경험이 있는 RTOS를 비교란에 명시한다.

4) OS Kernel Programming: 운영체제 커널 내의 일부를 구현한 경험 및 보유 기술 수준을 평가한다. Linux Device Driver, Windows CE Device Driver, RTOS

대분류	소분류	L1	L2	L3	L4	L5	비고	대분류	소분류	L1	L2	L3	L4	L5	비고
Programming Languages	C programming language							Middleware & Application Programming	GUI 모듈웨어						
	C++ programming language								각종 표준 모듈웨어						
	Java programming language								Windows Programming						
	Assembly programming language								Web Programming						
Development Tools	MS Visual Studio							Hardware Control Programming	Mobile Programming						
	GCC								Application Programming						
	Java Development Tools								Firmware Programming						
	Debugger								VHDL/VeriLog Programming						
System Programming	ICD/JCE							Software Engineering/Development Process	계측장비 활용기술						
	Linux/LINUX System Programming								하드웨어 설계 도구 활용 기술						
	Windows System Programming								하드웨어 구조 이해 능력						
	Socket programming								Development Process						
OS Kernel Programming	RTOS-based System Programming							Development Process	실제/모의형 방법론						
	Linux Device Driver Programming								Test/QA						
	Windows Device Driver Programming								시스템 구성 기술						
	Windows CE Device Driver Programming								System Optimization						
	RTOS Device Driver Programming								Documentation						
	Linux Kernel Implementation														

그림 1 임베디드 소프트웨어 기술 수준 파악을 위한 표준 이력서 양식

Device Driver 소분류 항목에 대해서는 구현 경험이 있는 디바이스 드라이버의 종류를 '비고'란에 명시한다.

5) Middleware & Application Programming: GUI와 각종 표준 미들웨어 구현 경험, 그리고 미들웨어를 활용한 응용 소프트웨어 구현 경험 및 기술 보유 수준을 평가한다. '각종 표준 미들웨어' 소분류 항목에 대해서는 보유 기술 수준 평가 외에 구현 경험이 있는 표준 미들웨어를 '비고'란에 명시한다.

6) Hardware Control Programming: 임베디드 시스템 소프트웨어 개발을 위해 필요한 하드웨어 제어 기술 보유 수준을 평가한다. 일부 하드웨어 설계를 위한 기술 보유 수준도 평가한다.

7) Software Engineering/Development Process: 구현 기술 외에 비 기술적 경험 수준을 평가한다. 개발 프로세스, 모델링, 그리고 문서 작업 수준 등을 평가하여 관리자 능력 보유 수준을 평가한다.

Embedded Software Skills Standard Masking Sheet (Linux Device Driver Engineer)																
대분류	소분류	L1	L2	L3	L4	L5	비고	대분류	소분류	L1	L2	L3	L4	L5	비고	
Programming Languages	C programming language							Middleware & Application Programming	GUI 미들웨어							
	C++ programming language								각종 표준 미들웨어							
	Java programming language								Windows Programming							
	Assembly programming language								Web Programming							
Development Tools	MS Visual Studio							Hardware Control Programming	Mobile Programming							
	gcc								Application Programming							
	Java Development Tools								Firmware Programming							
	Debugger								VHDL/Verilog Programming							
System Programming	ICD/JCE							Software Engineering/Development Process	계량장비 활용기술							
	Linux/UNIX System Programming								하드웨어 설계 도구 활용 기술							
	Windows System Programming								하드웨어 구조 이해 능력							
	Socket programming								Development Process							
OS Kernel Programming	RTOS-based System Programming							Software Engineering/Development Process	설계/모델링 방법론							
	Linux Device Driver Programming								Test/QA							
	Windows Device Driver Programming								시스템 구성 기술							
	Windows CE Device Driver Programming								System Optimization							
	RTOS Device Driver Programming								Documentation							
Linux Kernel Implementation																

그림 2 리눅스 디바이스 드라이버 개발자를 위한 Masking Sheet의 예

Embedded Software Skills Standard Masking Sheet (CDMA/GSM Phone Software Engineer)																
대분류	소분류	L1	L2	L3	L4	L5	비고	대분류	소분류	L1	L2	L3	L4	L5	비고	
Programming Languages	C programming language							Middleware & Application Programming	GUI 미들웨어							
	C++ programming language								각종 표준 미들웨어							
	Java programming language								Windows Programming							
	Assembly programming language								Web Programming							
Development Tools	MS Visual Studio							Hardware Control Programming	Mobile Programming							
	gcc								Application Programming							
	Java Development Tools								Firmware Programming							
	Debugger								VHDL/Verilog Programming							
System Programming	ICD/JCE							Software Engineering/Development Process	계량장비 활용기술							
	Linux/UNIX System Programming								하드웨어 설계 도구 활용 기술							
	Windows System Programming								하드웨어 구조 이해 능력							
	Socket programming								Development Process							
OS Kernel Programming	RTOS-based System Programming							Software Engineering/Development Process	설계/모델링 방법론							
	Linux Device Driver Programming								Test/QA							
	Windows Device Driver Programming								시스템 구성 기술							
	Windows CE Device Driver Programming								System Optimization							
	RTOS Device Driver Programming								Documentation							
Linux Kernel Implementation																

그림 3 휴대폰 개발자를 위한 Masking Sheet의 예

나. 표준 이력서 사용 방법 및 활용도

ESSS 표준 이력서는 현재는 일단 자가 진단 방식을 통해 개발자 스스로 보유 기술의 기술 수준을 평가한다. 향후 표준적인 평가 기준 및 평가 방법을 마련할 예정이다. 개발자는 각 대분류 내 소분류 항목에 대해 기술 수준 가이드라인에 설명된 기술 수준 요구 조건을 확인하여 어느 기술 수준에 해당하는지 표기한다. ‘비고’란에는 기술 수준에 대한 단계 표기 외에 보유 기술의 보다 정확한 내용을 전달하기 위해 기술의 종류를 기입한다.

각 기업에서는 개발자가 표기한 표준 이력서 양식을 보고 해당 기업에서 필요한 기술을 보유하고 있으며 해당 기업에서 활용 가능한 기술 수준을 갖추고 있는지 판단한다. 이 때 보다 체계적으로 수월하게 개발자의 기술 수준을 파악하기 위해 각 기업에서는 산업 분야별 ‘Masking Sheet’를 활용할 수 있다. ‘Mask-

ing Sheet’란 각 산업 분야 별로 필요한 Skill Set과 그렇지 않은 Skill Set을 분류하여 필요한 Skill Set에 대한 기술 보유 수준만을 쉽게 평가하기 위한 평가 보조 도구이다. 그림 2와 그림 3에는 리눅스 디바이스 드라이버 개발자와 휴대폰 개발자를 위한 ‘Masking Sheet’의 가상의 예를 보인다. 그림에서 보이는 바와 같이 특정 산업 분야에서 필요 없는 기술 항목은 평가 대상에서 제외하여 각 산업 분야에서 필요한 기술 수준에 대해서만 평가할 수 있도록 가려져 있는 양식을 제공한다. 예를 들면, Linux Device Driver Engineer를 위한 Masking Sheet에서는 C 외의 고급 프로그래밍 언어 및 Windows Programming 관련 기술들이 Masking되어 있다.

대학 등 교육 기관에서는 특성화하고자 하는 분야의 Masking Sheet를 참고하거나 필요한 Masking Sheet를 정의하여 특성화 분야를 정하고 그에 맞는 교육

표 3 Programming Languages 기술 분야의 기술 수준 파악을 위한 가이드라인

소분류	Level	정의
C	1	1차원 배열, 2차원 배열을 이용한 문제해결 함수 구현이 가능하다.
	2	포인터를 사용한 다양한 스트링 조작 함수 구현이 가능하다. Command line argument 처리가 가능하다.
	3	이중 포인터를 이용한 문제해결 함수 구현 가능, 이진파일에 데이터를 읽거나, 쓰거나, 조작하는 기능 구현이 가능하다.
	4	C소스파일이 10개 이상이며 5,000 라인 이상의 소규모 프로그램 개발 경험이 있다.
	5	C소스파일이 20개 이상이며 20,000 라인 이상의 프로그램 개발, 개발 결과물을 오픈소스 소프트웨어, 쉘어웨어, 혹은 상용 소프트웨어로 다수가 사용할 수 있도록 릴리즈한 경험이 있다.
C++	1	C++ class 를 이용한 간단한 객체를 조작하는 프로그램 구현이 가능하다.
	2	Class의 operator overloading 를 이용한 새로운 객체의 연산자 구현 가능 Template를 사용한 class 구현이 가능하다.
	3	Inheritance, virtual function 등을 사용하는 계층구조의 여러 class 구현이 가능하다.
	4	STL에서 제공하는 다양한 자료구조를 자유롭게 사용 가능, C++소스파일이 10개 이상이며 5,000 라인 이상의 소규모 프로그램 개발 경험이 있다.
	5	C++ 언어를 사용한 객체 지향 구조를 갖춘 소프트웨어를 개발하여 오픈소스, 쉘어웨어, 혹은 상용 소프트웨어로 다수가 사용할 수 있도록 릴리즈한 경험이 있다.
Java	1	Java class를 이용한 간단한 객체를 조작하는 프로그램 구현이 가능하다.
	2	Inheritance, interface 등을 사용하는 계층구조의 여러 class 구현이 가능하다.
	3	AWT, Applet, Swing 등과 thread 프로그래밍이 가능하다.
	4	JDBC, Sublet, JSP 프로그래밍이 가능. Java 소스파일이 10개 이상이며 5,000 라인 이상의 소규모 프로그램 개발 경험이 있다.
	5	Java 언어를 사용한 객체 지향 구조를 갖춘 소프트웨어를 개발하여 오픈소스, 쉘어웨어, 혹은 상용 소프트웨어로 다수가 사용할 수 있도록 릴리즈한 경험이 있다.
Assembly Language	1	간단한 procedure 구현 가능. Inline assembly 프로그래밍이 가능하다.
	2	레지스터 접근을 중심으로 한 하드웨어 제어 코드 구현, Interrupt 처리, 파일 입출력 프로그램 구현이 가능하다.
	3	Assembly 코드로 이루어진 함수와 C 혹은 C++ 코드로 이루어진 함수 사이의 상호 호출을 이용한 10개 이상의 함수로 이루어진 소프트웨어 구현 경험이 있다.
	4	Assembly 코드와 C언어로 이루어진 부트로더를 구현 가능하다.
	5	Assembly 코드로 이루어진 운영체제의 하드웨어 초기화 코드를 완성하여 운영체제를 포팅 가능하다.

표 4 Development Tools 기술 분야의 기술 수준 파악을 위한 가이드라인

소분류	Level	정의
Visual Studio	1	Visual Studio 환경에서 C/C++ Project(혹은 Workspace)를 만들어 S/W 개발 가능하다.
	2	간단한 debugging 툴을 사용하여 프로그램의 tracing 가능하다.
	3	최적화, 사용자 라이브러리 링킹 등과 같은 여러가지 Setting 기능을 사용가능하다.
	4	Win32 dll(Dynamic-Link Library)을 만들 수 있다.
	5	Visual Studio Analyzer을 활용하여 개발되고 있는 S/W의 성능을 분석하는데 활용할 수 있다. Visual SourceSafe 를 이용한 S/W 개발 경험이 있다.
GCC	1	여러 개의 파일로 개발된 C/C++ 프로그램을 실행파일로 만들 수 있다.
	2	gdb와 같은 간단한 debugging 툴을 사용하여 프로그램의 tracing 가능하다.
	3	make 의 다양한 기능을 활용할 수 있다.
	4	Profiling 기능을 사용할 수 있다. Target Machine에 따른 machine code를 만들기 위한 기능을 사용할 수 있다.
	5	CVS(Concurrent Versions System)을 이용한 S/W 개발 경험이 있다.
Java Development Tools	1	Eclipse와 같은 IDE(Integrated Development Environment)를 설치하여, Java 프로그램 개발환경을 구축할 수 있다.
	2	Java interpreter, compiler, archiver, document generator 등을 사용하여 java 프로그램을 개발할 수 있다.
	3	Java Debugger 를 활용하여 Java 프로그램을 debugging가능하다.
	4	Java 통합 개발 환경을 이용하여 Java 응용 소프트웨어를 5개 이상 구현한 경험이 있다.
Debugger (Visual Studio, gdb 등)	1	프로그램 tracing 기능을 사용할 수 있다. (VS, gdb)
	2	콜 스택(VS), 스택 프레임(gdb)을 활용한 debugging 이 가능하다.
	3	변수값을 수정하는 등의 방법을 통하여 실행하는 과정을 변경하는 하여 debugging 할 수 있다 (VS, gdb)
	4	사용자 명령어를 만들 수 있으며, 파일에 저장된 명령어를 활용할 수 있다 (gdb)
	5	디버깅 타겟을 지정하여 debugging 할 수 있다(gdb)
ICD (In Circuit Debugger)	1	JTAG 인터페이스의 용도를 이해하고 ICD 장비의 목적을 이해하고 있다. 직접 ICD를 사용해서 디버깅한 경험은 없다.
	2	JTAG 인터페이스를 통해 하드웨어에 부트로더를 다운로드할 수 있다.
	3	ICD와 해당 ICD에서 제공하는 디버거를 이용하여 운영체제의 동작을 디버깅할 수 있다. 운영체제의 심볼을 읽어 운영체제 내의 함수 단위 디버깅을 할 수 있다.
	4	ICD와 운영체제를 연동하여 OS-aware 디버깅을 할 수 있다. (REX, Linux 등)
	5	ICD와 운영체제의 연동을 위해 커널 내의 디버거와 연동을 위한 코드를 작성할 수 있다.

커리큘럼을 만들어 대분류와 소분류에 나타난 항목에 대한 집중 교육을 할 수 있다. 특성화 산업 분야 혹은 대상 기업과 연계한 교육 체계를 만들 경우 도움이 될 것이다. 각 Masking Sheet 별로 개발자들의 보유 기술을 점수화할 수 있는 장치를 마련하면 개발자들은 특정 산업 분야의 Masking Sheet를 활용하여 자신이 해당 산업 분야에 적합한 기술을 어느 수준으로 보유하고 있는 지에 대해 정량화된 평가를 할 수 있을 것이다.

표 3과 표 4는 표준 이력서의 대분류 및 소분류에 대한 기술 수준을 평가하기 위한 가이드라인의 예를 보인다. 표 3은 Programming Languages 기술 분야에서 기술 수준 파악을 위한 가이드라인이다. 현재는 C 언어, C++ 언어, Java 언어, 어셈블리 언어 기술 분야에 대한 가이드라인만 제공하고 있는데 향후 추가 연구를 통해 다른 언어에 대한 기술 수준 파악 자료

를 추가할 예정이다. 컴퓨터 관련 학과 학부 수준의 학생의 경우 대부분의 커리큘럼이 프로그래밍 언어 활용 교육에 상당한 비중을 두고 있기 때문에 많은 학생들이 상당히 높은 수준의 기술 수준을 보유하고 있는 것으로 파악된다. 상대적으로 저수준 프로그래밍 언어이자 하드웨어에 대한 이해가 필요한 어셈블리 언어의 경우에는 대부분의 컴퓨터 관련 학과 학생들이 레벨 1이나 레벨 2 수준의 기술을 보유한 것으로 파악되었다.

표 4는 Development Tools 기술 분야의 기술 수준 파악을 위한 가이드라인을 보인다. 컴퓨터 관련 학과의 학부생을 대상으로 시범 적용을 한 결과 마이크로소프트 윈도우즈 기반 윈도우즈 프로그래밍 도구를 활용한 소프트웨어 개발 경험에 상당히 치중되어 있고 리눅스 기반 개발 도구인 GCC나 디버깅 관련 개발 도구에 대한 경험은 상대적으로 미약하였다. 현

장에서 활용되는 기술 중 디버깅 기술이 상당한 비중을 차지하는 것으로 볼 때 디버깅 관련한 실무 교육이 더 강화되어야 할 것으로 보인다.

4. 결론

ESSS는 임베디드 소프트웨어 분야의 인력 양성의 효율성을 높이고 기업에서 필요한 기술 수준을 보유한 인력을 확보하는 데 도움을 주고자 임베디드소프트웨어산업협회를 중심으로 수행된 연구 결과이다. ESSS는 기존 직무 및 경력 개발 관련 연구 결과에서 다루지 않았던 개별 개발자들에 대해 구체적인 기술 분류에 따른 보유 기술 수준을 파악하기 위한 연구이다. 이러한 연구는 개발자들 스스로의 경력 개발 및 인력 수요자인 기업들의 필요 인력 확보에 도움이 될 것이다.

연구 작업의 첫 단계로 2007년에는 7개 기술 분류에 따른 기술 수준에 대한 평가 가이드라인을 정의하였으며 특정 산업 분야의 필요 기술에 따른 Masking Sheet의 예를 구성하였다. 이러한 연구 결과를 바탕으로 2008년에는 기술 분류를 확대하고 각 기술 분류별 기술 수준을 현재의 5단계에서 7단계 이상으로 확장하여 보다 정밀한 기술 수준 파악이 가능하도록 추진할 예정이다. 또한, 본 연구에 참여하는 기업체 담당자를 통해 시범 적용을 하여 그 결과를 표준 이력서 양식 개선에 활용할 예정이다.

장기적으로 본 연구는 개발자들의 기술 수준을 파악하고 산업 현장에서 요구되는 기술을 쉽게 파악하기 위한 표준 양식으로 활용되도록 표준화 작업에 초점을 둘 예정이다.

참고문헌

- [1] 한국직업능력개발원, "IT 스킬 체계에 따른 SW 개발자 직무수행능력개발", 한국직업능력개발원 보고서, 2006.
- [2] 임베디드소프트웨어산업협회, "Embedded Software Skill-set Standard 구축 방법에 관한 연구", 최종보고서, 12월 2007.



임 성 수

1993 서울대학교 컴퓨터공학 학사
1995 서울대학교 컴퓨터공학 석사
2002 서울대학교 전기컴퓨터공학 박사
2000~2004 팜팜테크(주) 기술총괄이사
2004~현재 국민대학교 컴퓨터학부 조교수
E-mail : sslim@kookmin.ac.kr



임 은 진

1991 서울대 컴퓨터공학과 학사
1993 서울대 컴퓨터공학과 석사
2000 U.C.Berkeley Dept. of EECS 박사
2001~현재 국민대학교 전자정보통신대학 컴퓨터공학부 부교수
E-mail : ejim@kookmin.ac.kr



최 준 수

1984 서울대학교 전기공학 학사
1986 한국과학기술원 전산학 석사
1995 뉴욕대학교 전산학 박사
1986~1996 한국통신 연구개발원(선임연구원)
1996~현재 국민대학교 컴퓨터공학부(교수)
E-mail : jschoi@kookmin.ac.kr



김 기 철

1985 서울대학교 무역학과 경제학 학사
1987 건국대학교 경영학 석사
2004 삼보컴퓨터 이사
2004~현재 한국정보산업연합회 기획실장/상무
임베디드소프트웨어산업협회 사무국장
E-mail : kck@fkii.org