

임베디드 소프트웨어의 개발 프로세스와 품질평가방법

김금옥*, 양해솔**

The Embedded Software of Development Process & Quality Evaluation Method

Jin Jin Yu *, Yang Hae Sool **

요 약

최근 임베디드 소프트웨어가 다양한 분야에서의 개발이 증대되면서 요구사항도 변화되고 있다. 즉, 임베디드 시스템이 단순하고 독립적인 소형시스템에 정착되어 운영되었던 과거와는 달리 최근에는 초정밀의 동작제어와 함께 복잡하고 다양한 플랫폼 환경을 요구하거나 다수의 소프트웨어 통합하고 제어하는 시스템 중속적인 요구사항들이 많이 발생하고 있다. 현재 임베디드 소프트웨어 개발 분야의 가장 중요한 문제점은 개발 생산성의 향상과 품질의 확보라고 볼 수 있다. 본 연구에서는 컴포넌트 개발방법론을 토대로 개발 생산성과 품질 향상을 동시 확보할 수 있는 임베디드 소프트웨어 개발 프로세스를 구축하였다. 또한, 임베디드 소프트웨어 제품에 대한 품질 시험 체계를 확립하고 품질 시험을 위한 메트릭을 개발 /구축함으로써 품질 시험을 효과적으로 수행하여 경쟁력 있는 임베디드 소프트웨어 제품의 개발을 지원할 수 있는 방법론을 확립하였다.

Abstract

Recently use of the embedded software has been increased to different areas. The requirements and demands for the embedded software have also been altered. In the past, the embedded system was used in simple task and small portable devices but now, the usage of the embedded software has expanded to do much more complex and precise actions in a variety platform environment. The embedded software enables multiple softwares to be integrated into one and at the same time, control it. Currently the biggest challenges embedded software is facing during its development process is the improvement in product production and quality assurance. Our research team has developed an embedded software based on the component (technology or methodology) which both improves production capabilities as well as quality. Additionally, we also established and constructed a reliability test system which can effectively test the quality of the developed embedded software to further increase its competitiveness.

▶ Keyword : 임베디드 소프트웨어(Embedded software), 임베디드 소프트웨어 개발 프로세스(Embedded software development processor), 품질평가 모델(quality test model), 컴포넌트(Component)

• 제1저자 : 김금옥 교신저자 : 양해솔

• 접수일 : 2008. 9. 3, 심사일 : 2008. 9. 11, 심사완료일 : 2008. 11. 26.

* 호서대학교 벤처전문대학원 박사과정 **호서대학교 벤처전문대학원 교수

※ 본 연구는 지식경제부와 IITA의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2008- (C1090 -0801 - 0032))

I. 서론

다가오는 유비쿼터스 시대에는 자동차, 의료, 건설 등 다양한 산업에서 더욱 고도화된 지능형 시스템이 요구되며, 임베디드 소프트웨어 비중이 점차 커지면서 제품의 핵심경쟁력이 하드웨어 생산에서 소프트웨어 기술로 이동하고 있다[1].

임베디드 소프트웨어는 최종 제품의 기능과 밀접한 관계가 있으므로 임베디드 소프트웨어를 개발할 때에는 제품의 품질, 시장 적시성, 생산성이 고려돼야 한다. 이는 프로젝트 전체 생명주기에 대한 관리와 단계별 개발 과정간의 확실한 연계, 도구의 체계적인 지원으로 가능하다. 해외에서는 이와 같은 분야의 연구가 활발하게 진행 중에 있으나 현재 국내의 임베디드 소프트웨어 개발은 체계적인 개발방법 없이 주먹구구식으로 이뤄지고 있는 실정이다. 임베디드 소프트웨어를 현업에서 개발해 본 경험자나 개발을 시작한 국내 업체들은 개발에 필요한 체계가 존재하면 유용할 것이라는 견해를 공통적으로 나타내며 개발을 위한 지침이나 방향을 요구하고 있다[1, 2, 6]. 현재 국내에는 임베디드 소프트웨어를 경제·효율적으로 개발할 수 있는 개발방법론과 개발 지원도구가 부족할 뿐만 아니라 임베디드 소프트웨어 개발 필요한 표준화 작업은 미비하다.

본 연구에서는 이런 문제를 해결하기 위해 개발 생산성과 품질 향상을 동시 확보할 수 있는 임베디드 소프트웨어 개발 프로세스를 제시하고 임베디드 소프트웨어 제품에 대한 품질 시험 체계를 확립하고 품질 시험을 위한 메트릭을 개발/구축함으로써 품질 시험을 효과적으로 수행하여 경쟁력 있는 임베디드 소프트웨어 제품의 개발을 지원할 수 있는 방법론을 확립하였다.

II. 관련 연구

2.1 임베디드 시스템과 응용분야

임베디드 시스템이란 특정한 목적으로 수행하도록 만든 컴퓨터 하드웨어와 소프트웨어의 결합체이다. 즉, 임베디드 시스템에서 컴퓨터는 특정 목적을 위해 설계된 시스템의 한 구성요소이다. 임베디드 시스템 응용분야는 제어, 가전, 단말, 통신장비 등, 4개의 응용분야로 나눌 수 있다. 표 1에서는 각 응용분야와 그들의 특징들을 비교분석하고 있다[2].

표 1. 응용과 특성비교
Table 1. Application and design trend comparative

	실시간성	입출력	GUI	CPU 처리	소비전력 요구
제어	크다	다양	불필요	보통	보통
가전	작다	단순	대형LCD	높다	작다
단말	작다	단순	중소형LCD	보통	크다
통신장비	보통	다양	불필요	보통	보통

2.2 임베디드 소프트웨어 표준화 동향

다양한 컨버전스 서비스를 지원할 수 있는 컴포넌트 구조 기반의 서비스 지향 모바일 미들웨어 플랫폼 기술 개발이 확산되는 추세이다. 각종 모바일 서비스 제공에 필수적인 요소들을 포함하는 공통 플랫폼은 없으며, 다양한 기능을 탑재해 단말이 제공하는 서비스에 특화된 플랫폼들이 존재한다. LiPS, OMA, OMTP 등 국제 표준화 기구를 중심으로 모바일 환경을 위한 플랫폼 표준화 활동이 이루어지고 있으나, 다양한 서비스를 위한 공통 컴퓨터 플랫폼 관련 표준화는 진행되고 있지 않다. 국내에서는 무선인터넷 서비스를 위한 미들웨어 플랫폼으로 무선인터넷표준플랫폼(WIPI) 규격이 제정되어 상용화 되고 있으며, 3G 등 변화된 시장에 보다 경쟁력을 갖기 위한 새로운 버전의 규격 제정을 위해 활동 중이다. 모바일 중심의 컨버전스 시장에 대한 경쟁력 선점을 위한 노력으로 다양한 기기와 서비스를 확장성 있게 제공할 수 있는 통합 플랫폼 개발 방안 모색중이다. 센서네트워크와 IPv6 네트워크를 직접 연동하는 기술인 6LoWPAN이 최근 IETF에서 표준화 진행하고 있다[4].

III. 임베디드 소프트웨어 개발 프로세스 구축

기존의 개발방법들은 하드웨어와의 밀접한 관계, 여러 유사 도메인에 대한 고수준의 재사용성 요구 등 임베디드 소프트웨어가 갖고 있는 특성들을 효과적으로 반영하지 못하고 있다. 그리고 국내 업체들의 프로세스 성숙도 수준은 아직 발전 초기단계 상태이다. 국내 환경에서 그 동안의 단계적 개발생명주기에 익숙한 개발자들이 CBD의 반복적 개발생명주기를 완전하게 적용하는데는 현실적인 한계가 있다. 이러한 현황을 감안하여 반복적 개발생명주기 적용의 현실적인 한계와 정보공학의 장점인 데이터 중심 모델링 기법 및 객체기술을 적절

히 조합하여 컴포넌트 기반 개발방법론을 토대로 개발 프로세스를 구축하여야 한다. 따라서 본 연구에서는 “계획⇒설계⇒구현⇒테스팅” 단계와 같은 임베디드 소프트웨어 개발 프로세스를 정립하고 각 단계에서 필요한 최적의 품질관리 체계를 갖춘 컴포넌트 기반 임베디드 소프트웨어 개발 프로세스 제안하고 적합한 품질평가 체계를 구축하였다.

3.1 임베디드 시스템 개발 생명주기

임베디드 시스템 개발 프로젝트의 생명주기는 그림 2와 같이 크게 6단계로 나눌 수 있으며 각 단계에 대응하는 검증 절차들을 가진다.

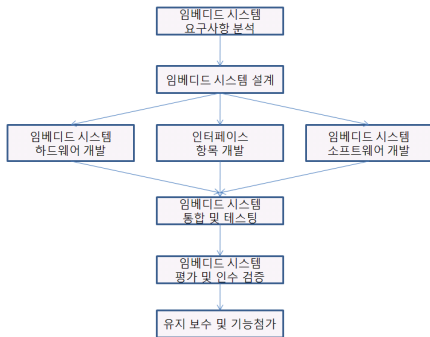


그림 1. 임베디드 시스템 개발 생명주기
Fig. 1. Development life cycle for embedded S/W

3.2 컴포넌트 기반 임베디드 소프트웨어 개발 프로세스

본 연구의 개발방법론은 우선 계획단계, 설계단계, 구현단계, 인도단계의 네 단계로 구분된다. 그리고 각 단계는 명확한 목표설정과 이의 달성을 위한 실행 그리고 평가를 포함하는 작은 프로젝트 수준으로 정의한다. 각 단계는 프로젝트 일정 수립과 진척관리 과정에서 주요관리이징표(Major Milestone)로 활용하게 되며, 국내 개발자들의 문화를 고려하여 각 주요관리이징표는 계약 및 기성고 산정과 연계하여 관리하게 된다.

다음은 본 연구의 임베디드 소프트웨어 개발 프로세스 각 단계에 대한 설명과 작업흐름도이다.

계획단계 그림 3에서는 개발 할 시스템의 비전, 목표 및 범위를 결정하는 작업을 수행한다. 이를 위해 사용자 요구사항을 수집/분석하고 기존 구조적 분석 방법론의 컨텍스트 다이어그램을 이용하여 외부 환경과 타깃 시스템 간의 관계를

분명히 설정하고, 전체 시스템에 대한 컨텍스트 다이어그램의 다양한 지침에 따라 부시스템으로 만들고, 각각 부 시스템에 존재하는 객체와 함수를 세분화한다. 타깃 시스템 전체의 동적 행동 특성을 표현하기 위해 상태전이 다이어그램을 사용한다. 그 다음에 시스템 요구분석 및 설계 바탕으로 하드웨어와 소프트웨어 트랙을 구분한다.

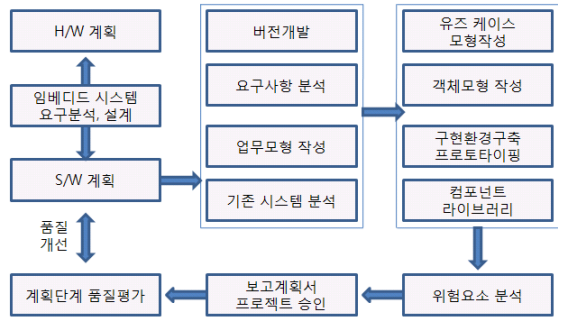


그림 2. 계획단계 개발 프로세스
Fig. 2. Development process for plan staircase

설계단계 그림 4에서는 요구사항 정련을 통해 아키텍처 관련 요건을 파악하고, 이를 토대로 구현 가능한 응용 아키텍처와 재사용성 높은 비즈니스 아키텍처를 포함하는 시스템의 아키텍처를 결정한다. 또한 다양한 기법을 통해 비즈니스 컴포넌트를 식별하고, 응용아키텍처를 지원하는 응용 컴포넌트를 결정 후 최종 컴포넌트 명세를 작성하여 컴포넌트를 도출한다.

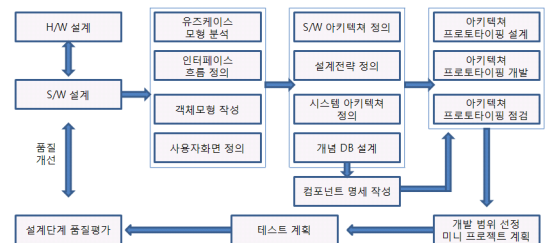


그림 3. 설계단계 개발 프로세스
Fig. 3. Development process for design staircase

아울러 아키텍처 프로토타입을 통해 아키텍처적인 위험요소를 발견하고, 이를 반영하는 시스템 아키텍처를 결정하여 향후 구현 과정에서 발생 가능한 아키텍처 위험요소를 제거한다.

구현단계 그림 5는 개발해야 할 시스템을 유스케이스를 기준으로 컴포넌트 단위로 분할하여 개발한다. 컴포넌트로 구축할 유스케이스를 구현 관점에서 보완하고, 이를 바탕으로 아키텍처를 보완한다. 식별된 컴포넌트의 내부를 설계하고, 데이터

베이스, 사용자 인터페이스 등을 설계하고 구축한다. 개발된 컴포넌트 대해 인터페이스 체대로 구현되었는지 컴포넌트가 통합테스트를 수행하여 실제 자동환경에서 통합하게 된다.

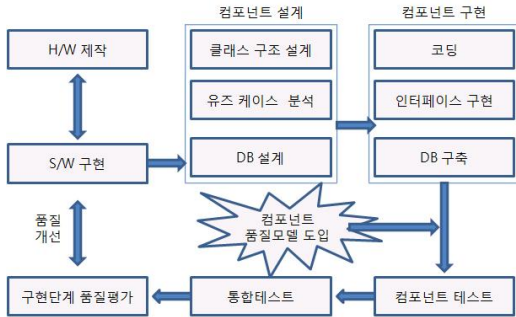


그림 5. 구현단계 개발 프로세스
Fig. 5. Development process for implementation staircase

컴포넌트 테스트 단계에서 소프트웨어 컴포넌트 품질모델을 적용하여 컴포넌트의 품질특성인 재사용성 및 준수성을 초점을 맞춰 확인한다.

ISO/IEC 9126에서 제시한 범용적 품질모델을 기반으로 하여 소프트웨어 컴포넌트 특성을 분석하고 그를 근거로 컴포넌트의 품질모델을 도출하여 그의 적용함으로써 컴포넌트의 품질을 보증한다[7].

표 2. 소프트웨어 컴포넌트 품질모델
Table 2. Quality model for S/W component

Quality Model	소프트웨어 품질 모델
Factors	재사용성, 준수성
Criteria	결합성, 범용성, 특화성, 도메인 준수성, 플랫폼 준수성
Metrics	Generic Metrics for Componenets

인도단계 그림 6에서는 개발자 환경에서 개발된 소프트웨어와 하드웨어를 통합한다. 여러 개 부시스템이 있는 경우 각 시스템 간에 연동시킨다. 기존에 운영되고 있는 시스템이 있을 경우 신규 시스템으로 전환하여 원활 한 운영이 가능하도록 한다. 품질평가를 통해 대상 제품에 대한 품질 수준을 파악하고 문제점을 개선을 통한 품질수준을 향상시킨다. 개발된 시스템에 대하여 품질평가의 결과를 근거로 사용자 요구사항과의 일치 여부에 대하여 승인을 얻고 인도하게 된다.

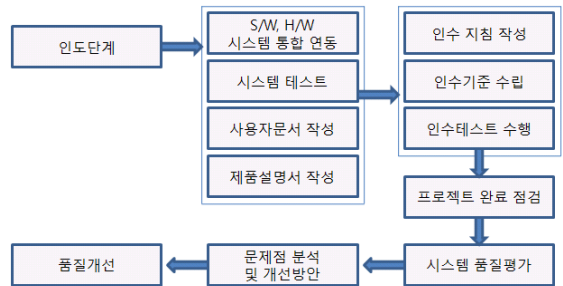


그림 6. 인도단계 개발 프로세스
Fig. 6. Development process for delivery staircase

본 연구에서 병행 개발 유형에 따른 개발 프로세스는 전통적인 하드웨어와 소프트웨어를 순차적으로 개발 유형이 아니라 하드웨어와 소프트웨어가 동시에 개발해야 하기 때문에 하드웨어 개발 방법과 결합되어서 접근해야 하며 서로 협력적인 하드웨어와 소프트웨어 기능들을 대상으로 한다. 이런 개발의 중간 단계에서 하드웨어와 소프트웨어의 동시 검증능력이 올바른 설계를 위한 필수적인 요건이 되며, 각 영역의 단계별로 일관성과 정확성을 가지고 지속적인 통제를 요구한다.

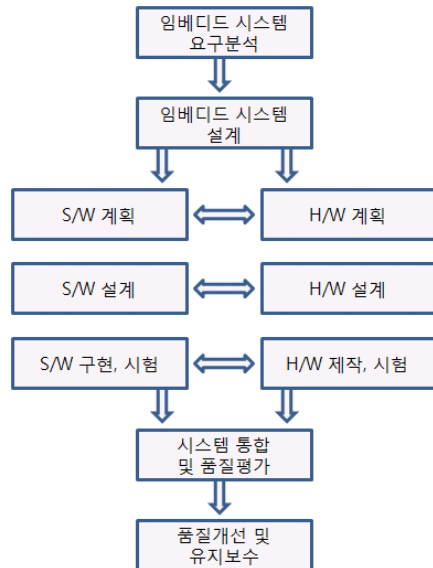


그림 7. 임베디드 소프트웨어 병행개발 프로세스
Fig. 7. Parallelism development process for embedded S/W

IV. 임베디드 소프트웨어에 품질 요구사항과 품질평가모델

본 연구에서는 임베디드 소프트웨어에 대한 품질평가 모델을 구축하기 위해 ISO/IEC 12119, ISO/IEC 9126과 ISO/IEC 14598-5을 도입하여 임베디드 소프트웨어의 품질 평가에 적합한 품질평가모델을 구축하고자 한다(11, 12, 16).

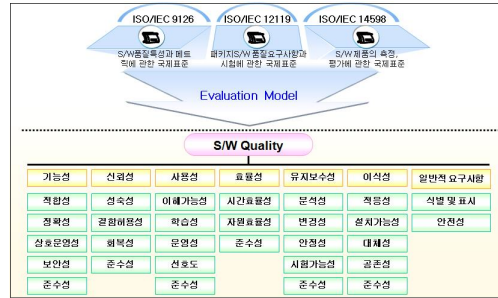


그림 8. 통합적인 품질모델
Fig. 8. Integrated quality model

4.1 임베디드 소프트웨어의 요구사항

(1) 임베디드 소프트웨어 요구사항

임베디드 소프트웨어의 특성(8, 9, 10)에 따른 요구사항을 살펴보면 다음과 같다.

- ① 신뢰성
- ② 경제적인 원가
- ③ 하드웨어 플랫폼이 쉽게 변경될 수 있는 이식성 요구
- ④ 소프트웨어 재사용 요구 강화
- ⑤ 소규모 메모리
- ⑥ 실시간 지원
- ⑦ 소프트웨어 업그레이드 쉽게 할 수 있도록 기능 확장이 용이성 요구
- ⑧ 다양한 소프트웨어 기능의 PnP와 재구성
- ⑨ 임베디드 소프트웨어는 하드웨어와 동시개발
- ⑩ 하드웨어와 소프트웨어 부분의 조화
- ⑪ 최소 비용에 최대의 효과
- ⑫ 거친 환경에서 강한 내구성 요구
- ⑬ 초소형/초경량 시스템의 일반화
- ⑭ 신기능의 추가나 이미 설계된 모듈과의 연결 용이
- ⑮ 사용편의성 요구
- ⑯ 임베디드 시스템 간 상호운용성을 요구

4.2 품질평가모델의 구성

임베디드 소프트웨어의 품질평가 요소에는 제품설명서, 사용자문서, 실행 소프트웨어와 관련 데이터가 있다. 임베디드 소프트웨어에 대한 시험 프로세스를 구축하기 위해 소프트웨어 품질평가에 관한 국제표준인 ISO/IEC 9126, 패키지 소프트웨어 평가 프로세스에 관한 국제 표준인 ISO/IEC 14598-5을 도입하였다. 통합적인 품질모델은 그림 8과 같다.

임베디드 소프트웨어의 품질시험을 위한 프로세스는 그림 9와 같다.

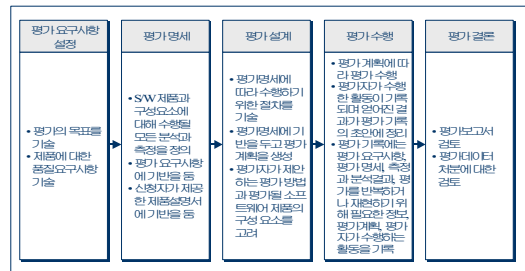


그림 9. 임베디드 소프트웨어의 품질시험 프로세스
Fig. 9. Quality test process for embedded S/W

4.3 임베디드 소프트웨어를 위한 품질평가 모듈

임베디드 소프트웨어는 일반 소프트웨어와는 다른 여러 특성들을 가지기 때문에 임베디드 소프트웨어 품질시험 및 평가를 위해서는 이러한 차이점을 명확히 이해하고 수용하여 평가 모듈의 개발에 적용해야 한다.

(1) 품질측정 항목의 설정

본 연구에서 임베디드 소프트웨어의 품질 평가모델의 전반적인 체계는 ISO/IEC 14598-5에 의해 정의된 평가모델을 토대로 하며 각 평가항목에 대해 다음과 같이 구성되어 있다.

- 개요: 메트릭의 개념, 측정목적, 메트릭의 범주, 용어 설명 등을 기술
- 적용범위: 적용대상 및 필요자원, 적용가능한 시험기법, 적용시 고려사항 등을 기술
- 참조문서: 메트릭이 도출된 관련문서를 기술
- 메트릭: 측정항목, 측정방법, 계산식을 기술

- 적용절차: 메트릭 적용을 위한 상세절차 기술
- 결과해석 및 보고: 측정치의 매핑, 측정결과의 해석, 보고서항 등에 대해 기술

표 3. 임베디드 소프트웨어의 평가항목
Table 3. Quality evaluation items for embedded S/W

품질특성	부특성	평가항목
일반적 요구사항	식별 및 표시	제품정보 제공, 바이러스 감염 여부
가능성	적합성	기능 정보 제공, 데이터 정보 제공, 사용 환경 명세 제공, 기능 구현 완전성, 기능 충분성, 기능 구분 적절성, 경계값 정보 제공, 경계값 처리율
	정확성	기능 분류 명확성, 기능 구현 정확성 정보 제공, 기능 구현 정확성
	상호운용성	연결가능성, 데이터 교환 정보 제공 수준, 데이터 교환성
	보안성	접근 통제 정보 제공, 접근 통제 정보제공, 접근 통제 가능성(시스템)
	준수성	기능 표준 준수 정보 제공, 기능 표준 준수율
신뢰성
사용성
효율성
유지 보수성
이식성

(2) 평가모듈 개발 내역

표 4. 평가모듈 개발 내역
Table 4. Development list for evaluation module

특성	부특성	평가모듈수	계
일반적 요구사항	식별 및 표시	제품정보 제공... 2	2
가능성	적합성	기능 정보 제공... 8	21
	정확성	기능 분류 명확성... 3	
	상호운용성	연결 가능성... 3	
	보안성	접근 통제 정보 제공... 2	
	준수성	기능 표준 준수 정보 제공... 2	
신뢰성	성숙성	문제해결 입력정보 제공... 3	13
	오류허용성	다운 회피율... 3	
	회복성	데이터 회복 정보 제공... 3	
	준수성	신뢰성 수준 정보 제공... 2	

특성	부특성	평가모듈수	계
효율성	시간효율성	처리기한 명세... 3	7
	자원효율성	메모리 사용률... 2	
	준수성	효율성 표준 준수 정보 제공... 2	
사용성	이해가능성	예비지식 정보 제공... 6	18
	학습성	기능학습 용이성... 2	
	운영성	오류복구 용이성... 4	
	선호도	인터페이스 조정 가능성... 2	
유지보수성	준수성	사용성 표준 준수 정보 제공... 2	14
	분석성	진단 기능 정보 제공... 5	
	변경성	환경설정 변경 정보 제공... 2	
	안정성	환경설정 변경 안정성 정보 제공... 1	
	시험가능성	내장형 시험기능 정보 제공... 2	
이식성	준수성	유지보수 표준 준수 정보 제공... 2	9
	적응성	이식 편리성... 1	
	설치가능성	설치 정보 제공... 2	
	대체성	데이터 지속 정보 제공... 2	
	공존성	공존 가능 정보 제공... 2	
	준수성	이식 표준 준수 정보 제공... 2	
계	28		68

본 연구를 통해 표 4과 같이 일반적 요구사항, 가능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성에 대한 부특성 28 개에 대해 총 68개의 메트릭을 개발하였다.

또한, 임베디드 소프트웨어의 시험항목, 시험할 특성별로 품질평가 판정하는 기준을 표 5과 같이 설정하였다.

표 5. 합격 판정기준 설정
Table 5. Success set up examination standard

시험 대상	시험할 특성 (메트릭)	합격 기준
일반적 요구사항	제품정보 제공	식별 정보의 90% 이상을 제공
	바이러스 감염 여부	바이러스에 감염되어 있지 않음
가능성	제품정보 제공	식별 정보의 90% 이상을 제공
	바이러스 감염 여부	바이러스에 감염되어 있지 않음

신뢰성	문제해결 이력 정보 제공	소프트웨어 제품의 이전 버전 또는 이전 릴리즈에서 발견된 문제의 해결 정보 제공

시험 대상	시험할 특성 (메트릭)	합격 기준
효율성	처리 기한 명세	임베디드 시스템의 각 과제별 처리기한 명세
사용성	예비지식 정보 제공	제품 사용시 요구되는 예비지식 정보의 제공

유지 보수성	진단 가능 정보 제공	진단기능에 관한 정보 제공

이식성	이식 편리성	자신의 환경에 쉽게 적응시킬 수 있는 수준이 규정값을 준수함

(3) 품질검사표

품질검사표는 품질시험을 수행하는 과정에서 활용할 수 있는 자료로서 품질시험원에게 필요한 최소 필요사항을 포함하여 테이블의 형태로 구성하였다. 품질검사표의 예는 표 6과 같다.

표 6. 품질검사표 예
Table 6. A example of quality inspection table

메트릭명	임베디드 시스템이 제한 받고 있는 경계값의 범위를 벗어난 입력에 대한 예외처리를 하고 있습니까?	
경계값 처리율	A	경계값 확인 대상 항목 수
	B	- 사용자 문서에 명세된 경계값에 대한 테스트케이스를 작성
추정 항목	A	각 항목별 테스트케이스 성공률의 합
	B	- 테스트케이스에 따라 테스트를 수행하여 명세와 일치하는 경우를 체크
계산식	- 경계값 처리율 (BEC) = B/A $B = \frac{\sum_{i=1}^A \text{Success_TC}_i}{\sum_{i=1}^A \text{Total_TC}_i}$ - Success_TC : i 번째 경계값 처리 기능 확인을 위해 수행한 테스트케이스 중 성공한 건 수 - Total_TC : i 번째 경계값 처리 기능 확인을 위해 수행한 테스트케이스 수	
	결과 영역	0 ≤ 경계값 처리율(BEC) ≤ 1 결과값
문제점		

(4) 점검표

점검표는 품질 검사표를 이용하여 측정항목에 대한 측정을 수행하기 위해 작성된 테스트 케이스의 시험 목록이다. 예를 들어, 측정항목의 점검표는 표 7와 같이 작성될 수 있다.

표 7. 기능 점검표의 예
Table 7. A example of function examination table

순번	기능명	2.2 정확성		비고
		정확성		
		기능구현 정확성 정보 제공(D) Y/N	기능구현 정확성(P) Y/N	
1	소프트웨어 개요 설명	Y	Y	
2	사용허가	Y	Y	
3	패킷의 전송간격	N	NA	
4	패킷의 최대크기	N	Y	
5	...			

예를 들어, "기능구현 정확성" 메트릭의 경우 테스트 대상인 기능에 대해 문서상으로 언급되어 있는 경우 'Y'로 기입하여 전체 기능 수에 대해 언급된 기능의 수를 계산함으로써 결과값을 얻게 된다.

(5) 시험결과서

시험결과를 품질특성, 부특성, 메트릭의 분류체계에 따라 정리하여 표 8과 같다.

표 8. 시험 결과서의 예
Table 8. The example of test result

제품설명서 및 사용자 취급 설명서				
품질특성	부특성	메트릭	기준값	측정값
1. 일반적 요구사항	1.1 식별 및 표시	제품정보제공(PIP)	Scale	0.80
		기능정보제공(FDI)	Scale	0.92
2. 기능성	2.1 적합성	기능구현 완전성(FIC)	Scale	0.95
		경계값 정보 제공(BSI)	Scale	0.20
		경계값 처리율(BEC)	Scale	1.00
		기능구현 정확성 정보제공(AIP)	Scale	0.77
	2.2 정확성	기능구현 정확성(ADF)	Scale	0.92
	
3. 신뢰성	3.1 성숙성	문제해결 이력 정보 제공(PRI)	Y/N	NA
	

본연구의 품질평가 모델의 평가절차는 정리해 보면 그림 10과 같다.

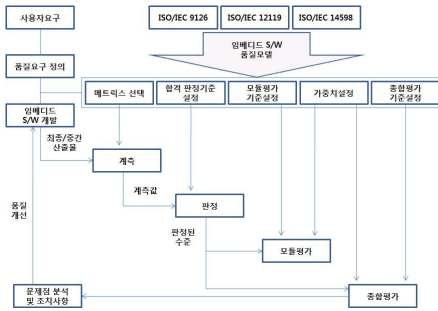


그림 10. 품질평가 절차
Fig. 10. The quality evaluation procedure

V. 임베디드 소프트웨어 평가적용과 분석

본 연구에서는 임베디드 소프트웨어 중 S사의 PDA 소프트웨어를 대상으로 하여 평가를 수행하였으며 품질을 측정하고 평가한 사례를 통해 문제점과 개선방안에 대하여 기술하였다.

(1) 메트릭에 대한 시험결과

임베디드 소프트웨어의 평가모듈에 관한 기본사항으로서 표 9은 평가모듈에서 사용하는 평가유형의 종류를 나타낸 것이며, 표 10는 평가모듈에서 사용하는 측정유형의 종류를 나타낸 것이다.

표 9. 임베디드 소프트웨어의 평가유형
Table 9. Evaluation type for embedded S/W

측정유형	측정단위	표시기호
측정유형 1	Y : 만족함 N : 만족하지 않음 NA : 적용 불가능	(Y/N/NA)
측정유형 2	비율	Scale

표 10. 임베디드 소프트웨어의 측정유형
Table 10. Measurement type for embedded S/W

측정유형	측정단위	표시기호
측정유형 1	Y : 만족함 N : 만족하지 않음 NA : 적용 불가능	(Y/N/NA)
측정유형 2	비율	Scale
측정유형 3	숫자	Number
측정유형 4	시간	Time

소프트웨어의 일반적 품질 요구사항에 대한 국제 표준인 ISO/IEC 12119와 소프트웨어 품질평가에 대한 정량적인 평가방법중 하나인 ISO 9126에 근거한 품질특성 6가지(기능성, 신뢰성, 효율성, 사용성, 유지보수성, 이식성)를 바탕으로 하여 품질평가를 수행하였다.

평가과정에서 우선 각 메트릭의 계산식을 구성하는 측정항목에 대한 값을 도출하기 위해 점검표를 이용하게 된다. 표 11에 '기능정보 제공' 메트릭에 대한 점검표를 나타내었다. 점검표에서 "기능 정보 제공" 메트릭의 경우, 나열된 기능요소에 대해 기능정보가 사용자 문서에 제공된 경우는 'Y', 제공되지 않은 경우는 'N'으로 기록하고 전체 기능의 수에 대해 기능정보가 사용자 문서에 제공되어 점검표에 'Y'로 기록된 항목의 수에 대한 비율을 산출하면 이 값이 '기능정보 제공' 메트릭에 대한 결과값이 된다.

표 11. 점검표의 예
Table 11. A example of examination table

순번	기능명	1.1	비고
		적합성 기능정보 제공(D)(Y/N)	
1	소프트웨어 개요 설명	Y	
2	사용하가	Y	
3	패킷의 전송간격	N	
4	패킷의 최대크기	N	
5	
합계	Y의 갯수	33	
	N의 갯수	1	
결과(계산식 = (Y/(Y+N)))		33/34 = 0.97	

표 12에서는 기능성에 대한 측정 결과를 나타내었다. 각 해당 점검표를 통해 메트릭의 값이 도출되는데 평가 대상이 되는 소프트웨어의 특성에 따라 메트릭의 적용이 적절하지 않은 경우나 적용 대상이 미비한 경우 'N/A(Not Applicable)'로 기록될 수 있다. 'N/A'로 기록된 경우에는 평가 대상에서 제외하여 결과 산출에 영향을 미치지 않을 수도 있고, 평가에 반드시 필요한 요소이면서 평가 대상 소프트웨어의 특성상 미비되어 있을 이유가 없는 경우에는 평가 대상에 포함한다.

측정결과를 통해 각 메트릭에 대한 결과를 알 수 있고, 상대적으로 취약한 품질특성을 파악할 수 있다. 기능정보 제공, 기능구현 완전성, 기능구현 정확성 정보제공 등이 우수한 결

과를 나타냈으며, 경계값 정보제공, 경계값 처리율 등이 낮은 값을 나타냈다. 또한, Y/N로 측정되는 메트릭은 'Y'를 1로, 'N'을 0으로 하였으며, N/A로 측정된 메트릭은 관련된 정보를 제공하지 않는 것으로 나타났다.

표 12. 기능성에 대한 측정 사례
Table 12. Functionality measurement case

특성	부특성	메트릭	결과	등급
기능성	적합성	기능정보제공	0.96	
		데이터 정보 제공	N/A	
		사용 환경 명세 제공	Y	
		기능구현 완전성	0.95	
		기능충분성	0.92	
		기능구분 적절성	0.77	
		경계값 정보제공	0.58	
		경계값 처리율	0.61	
	정확성	기능분류 명확성	0.96	
		기능구현 정확성 정보제공	0.95	
		기능구현 정확성	0.87	
	상호 운영성	연결가능성	0.96	
		데이터 교환 정보 제공 수준	0.84	
		데이터 교환성	0.91	
	보안성	접근통제 정보제공	0.77	
		접근통제 기능성(소프트웨어)	0.93	
	준수성	기능표준 준수 정보제공	0.60	
		기능표준 준수율	0.97	

(2) 품질특성과 부특성의 결과 집계

표 13은 품질 부특성에 대한 집계 결과를 나타낸 것으로 각 메트릭 결과로부터 부특성에 대한 메트릭값의 합계를 평균한 것이다. 결과를 통해 각 품질특성별로 취약한 결과를 보이고 있는 부특성들을 확인할 수 있다.

표 16은 표 15를 정리하여 품질특성에 대해 백분율로 나타낸 결과로서 100점 기준으로 획득된 점수를 나타낸다. 결과적으로 사용성과 이식성에서 상대적으로 낮은 점수를 나타내고 있으며 보완이 요구된다고 할 수 있다.

표 13. 품질 부특성에 대한 집계표
Table 13. The tabulation of quality subordination speciality

특성	부특성 결과				
	식별 및 표시				
일반적 요구사항	0.95				
기능성	적합성	정확성	상호 운영성	보안성	준수성
	0.79	0.93	0.90	0.85	0.78
신뢰성	성숙성	오류 허용성	회복성	준수성	
	0.79	0.92	0.84	0.76	
사용성	이해 가능성	학습성	운영성	선호도	준수성
	0.64	0.79	0.69	0.73	0.73
효율성	시간 효율성	자원 효율성	준수성		
	0.89	0.92	0.69		
유지 보수성	분석성	변경성	안정성	시험 가능성	준수성
	0.97	0.88	0.79	0.87	0.66
이식성	적응성	설치 가능성	대체성	공존성	준수성
	0.82	0.94	0.96	0.73	0.56

표 14. 품질특성에 대한 집계표
Table 14. The tabulation of quality speciality

특성	일반적 요구 사항	기능성	신뢰성	사용성	효율성	유지 보수성	이식성
결과값	95	85	83	72	83	83	80
평균	83						

(3) 문제점의 제시 및 개선방향

품질평가는 결과를 산출하는 것뿐만 아니라 결과에 대한 문제점을 분석하고 개발자에게 제시함으로써 고품질 소프트웨어 개발을 목적으로 한다. 표 15은 평가 대상 소프트웨어에서 나타난 품질특성별 문제점을 나타낸 것이다. 여기에서는 ISO/IEC 9126과 12119의 품질특성 수준에서 문제점을 제시하였다. 품질평가의 목적은 평가를 통해 대상 소프트웨어

제품에 대한 품질 수준을 파악하고 문제점을 발견하여 개선 방안을 모색하고 개선을 통해 품질수준을 향상시키는 것을 목적으로 한다. 따라서 필요한 경우, 부특성 수준 또는 매트릭 수준에서 문제점을 제시함으로써 품질평가에 목적에 따라 대응할 수 있다.

표 15. 문제점 예시의 일부
Table 15. A part of problem exemplification

시험결과 내역	
시험대상: 제품설명서 및 사용자 문서, PDA 소프트웨어	
일반적 요구사항	<ul style="list-style-type: none"> 현재 프로그램의 버전에 관한 정보 부재 인식 정보에 대한 상세수준까지 설명하지 않고 있음
가능성	<ul style="list-style-type: none"> 세부적인 기능 수행방법에 대한 정보 부족 보안성에 대한 비밀번호 설정기능 부재 자동수신 거부기능 부재
신뢰성	<ul style="list-style-type: none"> 발생되는 문제점에 대해 제공되는 정보가 부족
사용성	<ul style="list-style-type: none"> 도움말이 없어 사용하면서 발생하는 문제점에 대한 즉각적 조치 불가능 인터페이스에서 제공하는 피드백 부재
효율성	<ul style="list-style-type: none"> 특정 환경 설정시 데이터의 평균처리량이나 처리속도에 관한 정보 부족
유지 보수성	<ul style="list-style-type: none"> 문제점 해결방안에 대한 정보 부족 -사용시 빈번하게 발생할 수 있는 문제점 해결 방법 -운영과정에서 발생할 수 있는 오류 진단 방법 •환경설정에서 경계값 설정에 대한 오류를 알려주는 메시지를 제공하지 않음
이식성	<ul style="list-style-type: none"> •설치와 관련한 상세한 설명 부족 -설치시 발생할 수 있는 문제점 -프로그램 제거 방법

본 연구에서 S사의 임베디드 소프트웨어에 평가결과에 대해 그림 11과 같이 종합 판정할 수 있으며 그에 따라 향후 고 품질 임베디드 소프트웨어 개발 위한 품질향상 차원에서 필요한 조치를 취해야 할 것이다.

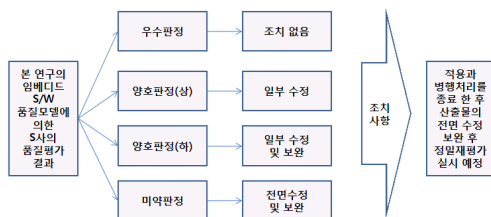


그림 11. 평가결과의 종합 판정과 조치사항
Fig. 11. The total decision and correction for evaluation result

VI. 결론

임베디드 소프트웨어는 갈수록 복잡화되고 복합적인 다양한 기능을 요구하게 됨에 따라 임베디드 소프트웨어의 생산성과 품질 향상의 중요성이 높아지고 있다. 따라서, 국내 실정에 적합한 임베디드 소프트웨어 제품의 개발프로세스와 품질 시험 체계의 구축이 매우 시급한 문제가 되고 있다.

따라서, 본 연구에서는 임베디드 소프트웨어의 개발 프로세스와 품질평가 모델을 개발하기 위한 것으로서 다음과 같은 사항에 중점을 두고 연구를 진행하였다. 임베디드 소프트웨어 개발 프로세스 구축, 유형별 품질 요구사항의 분석, 시험 절차의 확립, 품질평가 대상 및 항목에 대한 조사/연구, 품질평가 기준 연구, 품질평가 결과의 해석 및 판정 방법 확립 등에 관한 연구를 수행하였다.

임베디드 소프트웨어에 대한 개발 프로세스 구축 및 품질평가 모델 개발과 향후 실질적인 활용을 통해 국제표준을 수용한 소프트웨어 품질시험 측정 방법 구축, 국내 임베디드 S/W의 품질수준 향상에 기여, 국내 임베디드 소프트웨어의 생산성과 품질 향상을 통해 소프트웨어 유통시장의 안정성 확보, 고품질 임베디드 소프트웨어 개발 의욕을 제고하여 고부가가치 임베디드 소프트웨어 개발 등 측면에서 국제경쟁력을 확보할 수 있을 것으로 기대한다.

참고문헌

- [1] “국내 개발체계 기반의 임베디드 소프트웨어 발전방향”, ETRI, 소프트웨어 산업 부문별 시장/기술 전망 세미나, 2004.
- [2] 양혜술, “Embedded S/W의 품질평가 모델 개발 연구” 한국정보통신기술협회 위탁과제, 최종보고서, 2002.11.
- [3] Gartner, 2006. www.thegartnergroup.com
- [4] “임베디드 소프트웨어 산업 실태조사”(KESIC), 2007.
- [5] 오영배, 나희동, 박준성, 백두권, “컴포넌트 기반 소프트웨어 개발 프로세스”, 한국정보과학회, 제20권 제3호, ISSN 1229-6821, 2002.3.
- [6] 임형택, 임채덕, 박승민, 김홍남, “임베디드 소프트웨어 개발도구 동향”, IT정보, www.naver.com
- [7] 김남희, 신석규, “소프트웨어 컴포넌트 시험기술 동향” TTA저널 제88호

- [8] 김행곤, "임베디드 소프트웨어 공학", 그린출판사, 2006.
- [9] 김정국, 손재기, 박상현, "임베디드 리눅스의 이해" 상학당, 2005.9.5.
- [10] 남시병, 이해진, 이호근, "임베디드 시스템", 두양사, 2005.
- [11] ISO/IEC 9126, "Information Technology - Software Quality Characteristics and metrics- Part" 1, 2, 3.
- [12] ISO/IEC 14598, "Information Technology - Software product evaluation - Part 1, 2, 3, 4, 5, 6".
- [13] BS7925-1, "Glossary of terms used in software testing, British Computer Society Specialist Interest Group in Software Testing".
- [14] BS7925-2, "Standard for software component testing, British Computer Society Specialist Interest Group in Software Testing".
- [15] Binder, R. V., "Testing Object-oriented System: Models, Patterns, and Tools", Addison-Wesley, 2000.
- [16] ISO/IEC 12119, "Information Technology - Software Package - Quality requirement and testing".

저 자 소 개



김 금 옥

2005년 호서대학교 벤처전문대학원 졸업 (경영학석사)
 2008년 호서대학교 벤처전문대학원 정보경영학과 (박사과정)
 <관심분야> 소프트웨어 품질관리, 소프트웨어 프로세스 개선



양 해 술

1999년 일본오사카대학 정보공학과 (공학박사)
 1999년 호서대학교 벤처전문대학원 교수
 <관심분야> 소프트웨어공학(특히, S/W 품질보증과 평가, 품질검리와 컨설팅, COA/ OOD/ OOP, CASE, SI), 프로젝트 관리, CBD 기반기술, IT품질경영