

임베디드 소프트웨어의 개발 프로세스 구축

김금옥*, 이하용**, 양혜솔*

*호서대학교 혁신기술경영융합대학원

**서울벤처정보대학원대학교

e-mail:lhazby@hanmail.net, hsyang@office.hoseo.ac.kr

The Embedded Software of Development Process

Jin, Jin-yu*, Ha-Yong, Lee**, Hae-Sool, Yang*

*Graduate School of Multidisciplinary Technology and Management,
Hoseo University

**Seoul Univ. of Venture & Information

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 증속적인 요구사항들이 많이 발생하고 있다. 기존의 개발방법들은 하드웨어와의 밀접한 관계, 여러 유사 도메인에 대한 고수준의 재사용성 요구 등 임베디드 소프트웨어가 갖고 있는 특성들을 효과적으로 반영하지 못하고 있다. 본 연구에서는 개발 프로세스 혁신하고 오류를 최소화하기 위해서는 “계획⇒ 설계 ⇒ 구현 ⇒ 테스트” 단계와 같은 임베디드 소프트웨어 개발 프로세스를 정립하고 각 단계에서 필요한 최적의 품질관리 체계를 구축함으로써 경쟁력 있는 임베디드 소프트웨어 제품의 개발을 지원할 수 있는 방법론을 확립하였다.

1. 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.

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 나타내면 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. 임베디드 소프트웨어 개발이 기존의 전통적인 소프트웨어 개발과 확연히 다르다는 것을 고려했을 때 전통적인 개발방법을 그대로 적용하기에는 무리가 있다.

본 연구에서는 마르미-D와 컴포넌트 개발방법론을 토

대로 생산성과 품질 향상을 동시에 확보할 수 있는 임베디드 소프트웨어 개발 프로세스를 구축함으로써 제품개발활동을 효과적으로 수행하여 경쟁력 있는 임베디드 소프트웨어 제품의 개발을 지원할 수 있는 방법론을 확립하였다.

2. 관련연구

(1). 세계 현황

세계 임베디드 소프트웨어 시장은 '06년 1,148억 달러에서 '10년 1,305억 달러 규모로 연평균 3.9%, 국내는 '06년 84억 달러에서 '10년 100억 달러로 연평균 13.3%의 고성장이 전망된다. 세계 시장이 지속적으로 성장하는 가운데 미국 시장의 비중이 줄어드는 반면 EMEA(유럽&중동아시아) 및 AP(아태지역) 비중이 증가한다. Mobile Application OS 및 RTOS 시장의 경우 : 미국은 53.0%(2005) → 55.5%(2006), EMEA는 27%(2005) → 29%(2006), AP는 17.5% (2005) → 18%(2006)로 증가 및 2009년까지 16.6%의 가장 높은 성장률을 나타낼 것으로 전망된다. 한국의 임베디드 소프트웨어 시장규모는 미국, 유럽, 일본에 이어 4위로 중국과 비슷하다[3].

(2) 국내 현황

국내 임베디드 소프트웨어 시장은 '06년 84억 달러에서 '10년 100억 달러 규모로, 연평균 13.3%의 고성장이 전망된다. 하지만, 국내 시장은 임베디드시스템 개발 업체의 자체개발(in-house)이 87.5%를 차지하고, 임베디드 소프트웨어 또는 임베디드 모듈 개발 업체에 대한 용역개발은

10.5%, 라이선스/로열티 기반 개발은 2%에 불과하다. 국내 임베디드 소프트웨어 시장은 무선통신, 정보가전, 자동차 부문의 비중이 높다[6].

구분	생산액 (억 원)	임베디드 SW 시장규모 (억 원)	임베디드 SW 시장비중 (%)	산업별 임베디드 SW 비중 (%)	생산액 증가율 (%)
전 계	308,903,445	53,336,747	17.3%	2.69%	129.747
무선통신	5,624,251	1,183,251	1.42%	52.7%	119,251
무선가전	61,498,126	5,049,138	36.57%	4.90%	1,049,118
정보가전	112,185,093	1,672,233	20.06%	1.49%	1,672,233
자동차	62,091,407	2,294,614	27.52%	3.66%	2,294,614
산업자동화	67,373,269	1,138,701	13.66%	1.69%	1,138,701
의료장비	1,882,729	33,937	0.41%	1.80%	33,937
무주행교	704,620	29,894	0.35%	4.29%	29,894

(그림 1) 임베디드 소프트웨어 산업별 비중

(3) 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, OMTF 등 국제 표준화 기구를 중심으로 모바일 환경을 위한 플랫폼 표준화 활동이 이루어지고 있으나, 다양한 서비스를 위한 공통 컴퓨팅 플랫폼 관련 표준화는 진행되고 있지 않다. 국내에서는 무선인터넷 서비스를 위한 미들웨어 플랫폼으로 무선인터넷표준플랫폼(WIFI) 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. 센서 노드들이 IPv6 스택을 보유하는 방식으로 USN 사용자가 어디서든 IPv6 통해 통신 가능하도록 센서네트워크와 IPv6 네트워크를 직접 연동하는 기술인 6LoWPAN이 최근 IETF에서 표준화 진행하고 있다.

3. 임베디드 소프트웨어 개발 프로세스 구축

소프트웨어 개발단계에서 오류를 최소화하여 품질과 신뢰성을 확보하는 것이 제품 경쟁력의 핵심이라는 업계의 인식이 확산되고 있다. 컴포넌트 기반의 임베디드 소프트웨어 개발은 산업계에서 성장하고 있는 중요한 분야중의 하나이다. 재사용성과 유지보수성이 우수한 컴포넌트는 e-Business 가 활성화 되면서 급속하게 변화하는 복잡한 비즈니스 로직 구현 등의 요구사항을 만족 시키줄 수 있는 대안으로 산업계에서 활성화되어 있어야 한다. 기존의 개발방법들은 하드웨어와의 밀접한 관계, 여러 유사 도메인에 대한 고수준의 재사용성 요구 등 임베디드 소프트웨어가 갖고 있는 특성들을 효과적으로 반영하지 못하고 있다. 개발 프로세스 혁신하고 오류를 최소화하여 임베디드 소프트웨어 개발 생산성과 품질을 동시 제고하여야 한다. 개발 생산성과 제품의 품질을 향상을 위해서는 “계획⇒설계 ⇒ 구현 ⇒ 테스트” 단계와 같은 임베디드 소프트웨

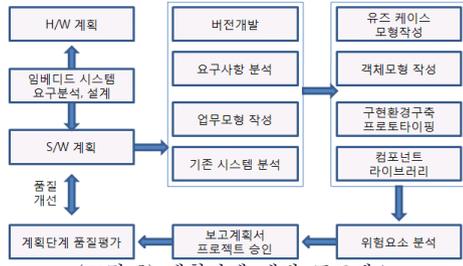
어 개발 프로세스를 정립하고 각 단계에서 필요한 최적의 품질관리 체계를 구축하여야 한다. 그러나 국내 업체들의 프로세스 성숙도 수준은 아직 발전 초기단계 상태이다. 국내 환경에서 그 동안의 단계적 개발생명주기에 익숙한 개발자들이 CBD의 반복적 개발생명주기를 완전하게 적용하는데는 현실적인 한계가 있다. 이러한 현황을 감안하여 반복적 개발생명주기 적용의 현실적인 한계와 정보공학의 장점인 테이타 중심 모델링 기법 및 객체기술을 적절히 조합하여 컴포넌트 기반 개발방법론을 토대로 개발 프로세스를 구축하였다.

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. 이는 국내 개발자들의 문화가 단계적 개발생명주기에 익숙한 점을 감안하여 프로젝트 관리 및 계약 등은 단계적인 접근으로 수행되되 내부적으로 반복적 개발방법을 접근할 것이다. 각 단계의 기준은 다음과 같다.

- 계획단계: 개발대상 시스템의 명확한 범위를 확정하며, 이를 검증하기 위한 UI수준의 구현물을 개발함.
- 설계단계: 시스템 구현을 위해 모든 아키텍처적인 위험 요소가 제거되어 최종 시스템 아키텍처를 확정하고 각 컴포넌트 도출을 완료함.
- 구현단계: 모든 컴포넌트 들이 개발되고 통합 테스트가 지 마무리 하였으며, 아키텍처 정의단계에서 정의된 테스트 기준을 통과함.
- 인도단계: 실제 사용자가 사용하도록 교육과 설치작업을 실시하기 위한 모든 작업이 완료되었으며, 시스템테스트와 인수테스트 기준을 만족시킴.

각 단계는 프로젝트 일정 수립과 진척관리 과정에서 주요 관리이정표(Major Milestone)로 활용하게 되며, 국내 개발자들의 문화를 고려하여 각 주요관리이정표는 계약 및 기성고 산정과 연계하여 관리하게 된다. 이외에 개발단계에서 본 임베디드 시스템의 소프트웨어를 점진적(Incremental), 반복적(Iterative)으로 개발 대상 컴포넌트를 분할하여 시스템을 개발하게 된다. 컴포넌트는 보조관리 이정표로 관리하게 된다. 다음은 본 연구의 임베디드 소프트웨어 개발 프로세스 각 단계에 대한 설명과 작업흐름도 이다.

계획단계 (그림 2)에서는 개발 할 시스템의 비전, 목표 및 범위를 결정하는 작업을 수행한다. 이를 위해 사용자 요구사항을 수집/분석하고 기존 구조적 분석 방법론의 콘텍스트 다이어그램을 이용하여 외부 환경과 타깃 시스템 간의 관계를 분명히 설정하고, 전체 시스템에 대한 콘텍스트 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.

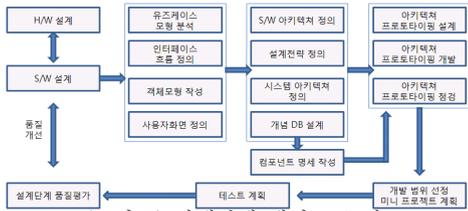


(그림 2) 계획단계 개발 프로세스

설계단계(그림 3)에서는 요구사항 정리를 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 결정한 후 최종 컴포넌트 명세를 작성하여 컴포넌트를 도출한다.

아울러 아키텍처 프로토타입을 통해 아키텍처적인 위험요소를 발견하고, 이를 반영하는 시스템 아키텍처를 결정하여 향후 구현 과정에서 발생 가능한 아키텍처 위험요소를 제거한다. 그 외에 점진적으로 시스템을 개발하기 위한 계획을 수립하여 다음 단계를 위한 반복계획을 수립한다. 이 단계에서는 반드시 독립적으로 실행 가능한 기능을 구현하여 사용자가 시스템 구현에 적극적으로 참여하여 요구사항을 제시하려는 분위기를 유도해야만 실질적인 구현상의 위험요소를 완화할 수 있게 된다.

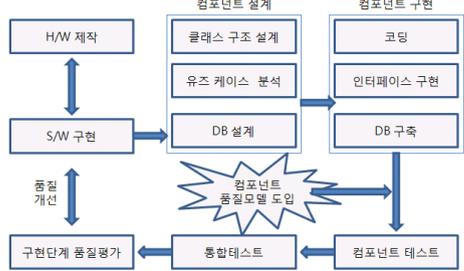
비즈니스 컴포넌트를 도출하는 가장 효율적인 방법은 유즈케이스를 중심으로 비즈니스 컴포넌트의 크기를 결정하고 지속성 계층(Persistency Layer), 즉 데이터 관점의 클래스를 중심으로 인터페이스를 식별하는 것이 적절하다.



(그림 3) 설계단계 개발 프로세스

구현단계 (그림 4)는 개발해야 할 시스템을 유즈케이스를 기준으로 컴포넌트 단위로 분할하여 개발한다. 컴포넌트로 구축할 유즈케이스를 구현 관점에서 보완하고, 이를 바탕으로 아키텍처를 보완한다. 식별된 컴포넌트의 내부를 설계하고, 데이터베이스, 사용자 인터페이스 등을 설계하고 구축한다. 개발된 컴포넌트 대해 인터페이스 제대로 구현되었는지 검사하기 위해 단위테스트를 수행하고, 컴포넌트가 통합테스트를 수행하여 실제 자동환경에서 통합하게 된다. 컴포넌트 테스트 단계에서 소프트웨어 컴포넌트 품질모델을 적용하여 컴포넌트의 품질특성인 재사용성 및 준수성을 조점을 맞춰 확인한다. 각 미니프로젝트가 끝난

후, 개발된 시스템의 기능성 및 성능이 사용자의 요구를 만족하는지 확인하기 위해 시스템테스트를 실시한다.



(그림 4) 구현단계 개발 프로세스

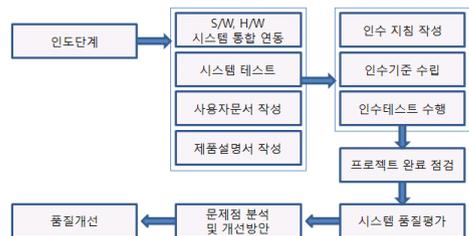
임베디드 소프트웨어 컴포넌트의 품질보증을 위해서는 개발 단계에서 컴포넌트는 ISO/IEC 9126에서 정의한 소프트웨어 품질특성 밖에 또한 컴포넌트의 여러 가지 특징을 고려하여야 한다. 소프트웨어 컴포넌트의 품질특성을 근거로 컴포넌트의 품질모델을 도출하여 그의 적용함으로써 컴포넌트의 품질을 보증한다. 소프트웨어 컴포넌트의 특징은 다음과 같다[9].

- 도메인 공통기능 평가
- 특화를 위한 Require 인터페이스 평가
- 컴포넌트 인터페이스 및 Granularity 평가
- 이진 파일 수준의 조합 및 대체성 평가
- 컴포넌트 참조 모델 준수성

<표 1> 소프트웨어 컴포넌트 품질모델

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

인도단계(그림 5)에서는 개발자 환경에서 개발된 소프트웨어와 하드웨어를 실제 시스템이 운영될 사용자 환경에 설치한다. 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 5) 인도단계 개발 프로세스

본 연구의 임베디드 소프트웨어 개발 프로세스는 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. 병행 개발 유형(그림2, 3, 4, 5)과 같이 하드웨어와 소프트웨어가 동시에 개발되는 프로세스로 시스템 설계가 마무리되기 전 하드웨어와 소프트웨어 설계가 시작되며 각 단계별 검증 작업은 하드웨어와 소프트웨어에서 동시에 수행되어야 한다. 이런 개발의 중간 단계에서 하드웨어와 소프트웨어의 동시 검증능력이 올바른 설계를 위한 필수적인 요건이 되며, 각 영역의 단계별로 일관성과 정확성을 가지고 지속적인 통제를 요구한다.

4. 결론

본 연구에서는 마르미-D와 컴포넌트 개발방법론을 토대로 생산성과 품질 향상을 동시 확보할 수 있는 임베디드 소프트웨어 개발 프로세스를 구축하였다. 개발 프로세스 혁신하고 오류를 최소화하기 위해서는 “계획⇒ 설계 ⇒ 구현 ⇒ 테스트” 단계와 같은 임베디드 소프트웨어 개발 프로세스를 정립하고 각 단계에서 필요한 최적의 품질 검증단계를 구축하였다. 향후 연구에서는 본연구에 제시한 임베디드 소프트웨어의 개발 프로세스 개선 및 혁신과 실질적인 활용을 통해 국내 임베디드 소프트웨어의 생산성 향상을 시켜 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] 양해술, “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