

국방 소프트웨어산업 활성화 방안



朴 允 鎬

국방품질관리연구소 소프트웨어 실장,
공학 박사

현대 무기체계는 과학기술의 발전으로 인해 고도화·첨단화 되어 가고 있는 추세이며, 특히 최근의 전쟁양상을 보면 첨단고도 정밀무기, 전자전무기 등과 같은 소프트웨어 집약형 무기체계를 구성하는 핵심인 소프트웨어의 중요성이 날로 증대되고 있다. 또한 과거에는 무기체계의 하드웨어가 전체 체계에서 차지하는 비중이 높았으나, 이제는 소프트웨어로 인해 소요되는 수명주기 비용과 중요도가 하드웨어를 이미 능가하고 있다.

21 세기 세계경제는 경쟁의 세계화와 정보통신기술(IT)의 발전으로 창의적 지식경제 인 프라 보유여하에 따라 흥망성쇠가 결정된다고 할 수 있다.

따라서 기업 및 국가의 경쟁력이 과거 노동, 자원중심에서 지식·정보·기술력 등에 의해 결정되는 지식기반경제로 급속히 전환되고 있으며, 선진 각국 및 OECD·세계은행 등 국제기구들은 지식을 통해 국가경쟁력을 제고하기 위한 지식기반경제 발전전략을 수립하여 시행 중에 있다.

우리 나라 또한 정부주도의 양적 성장전략이 90년대 이후 세계경제환경 변화로 인하여 한계에 직면하

게 되어 새로운 성장 원천을 '지식과 정보'에서 모색하는 국가전략을 수립하고 있다.

이러한 국내의 경제 환경에서 소프트웨어 산업은 경제사회전반에 걸쳐 지식이 효율적으로 창출·확산·활용될 수 있는 기반을 구축함으로써 국가경쟁력을 제고시키는 신산업 혁명의 인프라인 동시에 전통산업 경쟁력 강화의 기반이 되는 산업이라고 볼 수 있다.

이와 같은 사실은 방산 부문에 있어서도 크게 다르지 않다. 현대 무기체계는 과학기술의 발전으로 인해 고도화·첨단화 되어 가고 있는 추세이며, 특히 최근의 전쟁양상을 보면 첨단고도 정밀무기, 전자전무기 등과 같은 소프트웨어 집약형 무기체계를 구성하는 핵심인 소프트웨어의 중요성이 날로 증대되고 있다.

또한 과거에는 무기체계의 하드웨어가 전체 체계에서 차지하는 비중이 높았으나, 이제는 소프트웨어로 인해 소요되는 수명주기 비용과 중요도가 하드웨어를 이미 능가하고 있다.

그러나 소프트웨어를 재산으로 인식하는 마인드가 정부와 업체 모두에게 부족하고, 무기체계 획득의 기본 정책이 하드웨어 위주로만 수립, 시행되고 있어서 민수 분야에 비하여 소프트웨어 개발 자체가 체계적으로 수행되지 않고 있는 실정으로 방산업체의 소프트웨어 산업 환경은 대단히 취약한 실정이라고 할 수 있다.

따라서 이 글에서는 우리 나라 무기체계 획득 전반에 걸쳐 소프트웨어에 관한 문제점을 분석하고, 외국의 사례를 참조하여 우리 나라 실정에 맞는 무기체계 소프트웨어의 획득 및 유지보수에 관한 제도와 정책 수립에 기초자료로 제공될 수 있는 연구 결과를 제시하고자 한다.

이 글을 통하여 정부는 품질이 좋은 무기체계 소프트웨어를 획득함으로써 군의 전투력 향상을 도모하는 한편, 경제적인 소프트웨어 유지보수 기반을 구축하여 국방비 절감에도 기여할 수 있을 것으로 판단되며, 또한 방산업체는 소프트웨어에 대한 원가 보상을



품관소는 지난해 10월 창설 20주년 기념행사로 국방분야 최초로 소프트웨어 관련 심포지엄을 개최하였다.

통해 적절한 인력, 자원, 기술 등 기본적인 인프라에 대한 투자를 유도함으로써 방산 부문의 소프트웨어 산업기반의 활성화는 물론 국제적인 산업 경쟁력을 갖출 수 있을 것으로 생각된다.

소프트웨어의 획득 및 연구개발 실태

● 민수분야의 실태와 향후 동향

우리 나라의 민수분야 소프트웨어 산업은 1990년대 중반 이후 지식정보사회로의 진입에 따라 신산업이 출현하고 소프트웨어 분야에 대한 투자증대로 '95~'99년동안 연평균 13.3%의 고성장을 지속하고 있으며, 2000년 국내 S/W산업은 전년 대비 41%의 매출증가를 기록하며 9조 2천억원을 넘을 것으로 전망하고 있다.

그러나 과거 5년간 정부의 소프트웨어 산업육성정책, 벤처창업법, B2B전자상거래의 확대 등으로 국내 소프트웨어 산업은 견실한 성장세를 유지해 왔음에도 불구하고, 세계적 수준의 경쟁력 확보는 미흡한 실정이다.

이는 주로 제품개발 초기단계부터 글로벌시장을 겨냥한 상품개발 보다는 내수시장을 목표로 소프트웨어를 개발하는 경우가 많음으로 인하여 틈새시장

공략, 수출전략품목 증점지원 등의 공격적 해외진출 전략이 부족한 것에 기인한다.

또한 소프트웨어를 재산으로 인식하는 마인드가 부족하여 불법복제율이 여전히 높은 수준이며 전산 시스템 외주(Outsourcing)가 활성화되지 않았고, Home Market용 소프트웨어 판매점이 부족하며, 시장 확대에 따른 인력수요에 비해 공급시스템이 경직되어 있어 빠르게 변화하는 기술추세에 부합하는 인력 양성이 미흡하여 XML(eXtensible Markup Language), JAVA 등 고부가가치 창출을 주도하는 고급 기술인력이 태부족하다는 것도 중요한 요인 중 하나이다.

그리고 산업전반에 과급효과가 큰 대형 소프트웨어 개발 프로젝트보다는 평균 10억원 규모의 중·소형 연구개발이 많아서 세계 시장에서 경쟁력을 확보할 기반기술이 취약하고, 소프트웨어 벤처창업에 필요한 물적 인프라 구축에만 치중하고 창업 이후에도 지속적으로 성장할 수 있도록 법률·회계·마케팅활동 지원서비스와 시장·기술정보 제공이 부족하며 수도권권을 제외한 지방의 소프트웨어 산업 성장기반이 취약한 것도 사실이다.

이에 대한 정부의 정책 기본 방침은 지식기반경제의 핵심 신산업인 소프트웨어 산업을 『섬유 → 철강 → 반도체』의 뒤를 잇는 수출전략산업화를 추진한다는 전략을 가지고, 2005년까지 소프트웨어 수출 30억 달러(세계 7위 수준), 소프트웨어 생산액 세계시장 2.6% 차지, GDP대비 소프트웨어 생산비중 5%로 제고 등 구체적인 실천 목표를 설정하여 추진 중에 있다.

이에 따른 소프트웨어 산업 육성의 기본 방향을 세계경제의 글로벌화에 대응할 수 있는 수출경쟁력을

확보 할 수 있도록 해외시장 거점구축 및 해외마케팅을 적극 지원하고, 산업계 수요에 부응할 창의적 전문인력을 집중 양성하는 한편, 인터넷기반·게임·보안·컴포넌트·리눅스 등 핵심 소프트웨어 기술개발에 선택적 집중화 정책을 실시하려 하고 있다.

그리고 벤처기업 성장기반 구축 및 지방의 소프트웨어 산업 활성화를 통해 내수시장을 확대할 수 있도록 관련 법·제도를 정비하려는 계획도 가지고 있다.

이를 위한 구체적인 추진전략으로는 인력양성, 핵심 기술개발 보급 등의 기반조성을 통하여 「인터넷·게임·보안·컴포넌트·리눅스기반 소프트웨어」를 수출전략산업으로 육성하고, 수도권 이외 지방의 소프트웨어 산업 활성화 및 성장기반을 구축하기 위해 주요 거점도시를 중심으로 게임, 디지털컨텐츠 등 지역별 특성에 맞는 「소프트타운」을 조성, 전국적 S/W 산업육성 기반을 구축하려는 계획을 가지고 있다.

또한 창의적인 소프트웨어 전문인력 양성 및 차세대 소프트웨어 기술개발을 강화하기 위하여 JAVA, 컴포넌트 등 시대흐름에 특화된 전문인력을 집중양성하고, 무선인터넷 플랫폼, 음성인식 및 합성기술, 전자상거래기술 등 차세대 소프트웨어 기반기술 개발을 추진하고 있다.

이와 더불어 공공부문 수요창출, 법·제도의 개선 등을 통하여 소프트웨어 제품의 건전한 유통 및 내수 시장 활성화를 추진하고, 우수 소프트웨어 제품에 대한 품질인증제를 시행함과 동시에, 환경변화에 따라 소프트웨어 구매제도 등 관련 제도를 개선·보완함으로써 시장 활성화 기반을 조성하려고 한다.

이상에서 살펴본 민수분야 소프트웨어 산업의 주

민수분야의 소프트웨어 산업 주요 지표

항 목	2000년	2005년
소프트웨어 생산액	8,048백만 달러	33,804백만 달러
소프트웨어 수출액	182백만 달러	3,344백만 달러
개 발 인 력	8만명	22만명
1 인 당 생 산 액	100.6천 달러	153.6천 달러
수 출 액 순 위	23위	7위권

민수분야 소프트웨어 시장 전망

(단위 : 백만달러)

항 목	2001년	2002년	2003년	2004년	2005년	'01~'05 평균
생 산	11,354	16,196	21,382	26,958	33,804	29.9%
수 출	306	607	1,169	2,087	3,344	81.8%
수 입	491	586	688	796	928	17.2%
무역수지	-185	21	481	1,291	2,416	-

요지표와 시장 전망을 p.58 아래 표와 위의 표에 나타내었다.

방산분야의 실태

과거의 무기체계가 비교적 단순 기능을 가졌던 탓에 대부분 하드웨어 위주로 구현되고 획득 절차도 하드웨어 중심으로 수행되어 왔다. 이러한 획득 관행은 무기체계가 첨단화/복잡화되어 가고 사실상 성능의 핵심을 소프트웨어가 차지해 가고 있는 현재까지도 개선되고 있지 않은 상태이다.

이렇게 된 원인에는 소프트웨어를 하드웨어의 종속품 정도로만 인식하여 체계적인 개발 활동과 개발 관리를 수행하지 않고 있는 것이 가장 큰 문제점으로 지적될 수 있다.

즉, 소프트웨어에 대한 요구사항 분석, 설계, 구현, 품질보증, 시험평가 등 개발 단계별로 수행하여야 할 활동들이 제대로 이루어지지 않거나, 극단적인 경우에는 아예 무시된 채 요구되는 기능의 구현에만 노력을 집중하고 있는 형편으로서, 개발된 소프트웨어의 품질과 신뢰성을 확인하는 활동은 대단히 미흡한 형편이다.

또 한가지 문제점은 소프트웨어의 개발단계에서 반드시 작성되어야 하는 관련 기술자료가 생성되지 않거나, 개발자만이 알 수 있는 상태로 존재하고 있는 점인데, 이로 인하여 개발된 소프트웨어의 실체를 사실상 파악할 수 없고, 소프트웨어에 대한 품질보증, 형상관리 등이 근본적으로 제한을 받고 있는 실정이며, 해당 무기체계가 전력화된 이후에 발생하는 오류의 수정이나 수요군의 요구 사항을 반영하기 위한 개

선 활동도 엄청나게 비경제적으로 이루어질 수밖에 없는 문제점을 가지고 있다.

결국 무기체계에 내장되는 소프트웨어는 국방 자산으로 효율적인 관리가 되지 않는 상태라고 볼 수 있다. 여기에서는 실제 무기체계 획득사례를 토대로 국내 무기체계 소프트웨어의 현실태와 문제점을 분석하였다.

* 요구사항의 불분명으로 인한 개발 시행착오 과다

소프트웨어의 개발과정은 상당히 체계적인 분석과 검토의 과정을 거치면서 개발되어야만 요구되는 품질, 비용, 일정을 만족시킬 수 있다. 특히 개발의 가장 초기단계인 수요군의 요구사항을 정의하고 분석하는 단계는 전체 개발과정 중에서 가장 중요한 단계라고 할 수 있다.

소프트웨어의 개발과정에서 최초로 탄생하는 것이 소프트웨어 요구사항이다. 이러한 요구사항을 정의하는 1차적인 목적은 수요군이 ROC라는 형태로 제시한 포괄적인 요구를 개발자가 사용자의 관점으로 구체적이고 명확하게 정의함으로써, 개발과정 중에 발생할 수 있는 군과 개발자 간의 불확실한 사항을 조기에 발견하기 위함이다.

요구사항의 초기확정은 안정적이며 성공적인 소프트웨어를 만드는데 결정적인 요소이므로 모든 노력을 경주하여 수행하여야 한다.

그러나 우리의 현실은 수요군이 요구하는 성능과 기능이 무엇인지 구체적으로 결정되고, 확인하는 과정이 없이 소프트웨어를 설계하는 단계로 이어지는 것이 대부분이다. 요구사항을 제대로 만들지 않거나

전혀 만들지 않고 시작한 소프트웨어 개발 프로젝트가 실패한 예는 많이 찾아볼 수 있다.

첫번째 범주의 예는 정확한 요구사항을 만들어 사용자와 합의하지 않은 경우로서, 계속적인 사용자의 새로운 요구와 변경 요구로 인하여 개발 중에 있는 소프트웨어가 중심을 잡지 못하고 흔들리는 경우이다. 이런 경우에 있어서 소프트웨어의 품질이 좋을 걸로 기대하는 것은 무리이다.

두번째는 요구사항이 확정되어야 이것에 맞추어 개발될 소프트웨어의 시험계획을 미리 만들어 놓을 수 있는데, 만들어진 소프트웨어를 보고 나서야 시험계획을 만들게 되므로 막대한 시간의 손실은 물론, 시험의 독립성이 없어서 결과적으로는 시험을 제대로 못하게 되는 것이다. 세번째는 사용자가 개발된 소프트웨어를 인수하려는 시점까지도 요구사항을 확정하지 않은 경우로, 이 때의 개발자는 사용자의 의지 내에서 사용자의 뜻대로 움직일 수밖에 없게 된다.

이상에서 살펴본 경우들은 현재까지 우리의 무기체계 개발과정에서 쉽게 찾아볼 수 있는 사례들이다. 다시 말하면 실질적인 개발이 종료되었음에도 불구하고 소요군의 추가 요구사항 또는 요구사항의 변경으로 인하여 추가적인 개발활동을 다시 수행하는 사례가 빈번하게 발생하고 있다.

이에 대한 사례로서, 2001년 업체주도로 개발된 ○○○사격지휘 장비를 들 수 있다.

이 장비는 군에서 요구하는 성능과 기능의 대부분이 소프트웨어로 구현되는 장비로서, 초기에 군의 요구사항이 무엇인지를 구체적으로 확인하는 과정이 개발단계에서 가장 중요한 부분이지만, 군과 업체 모두 설계와 구현에만 관심을 집중하였고, 장비가 실질적인 개발이 종료되고 난 운용시험 단계에 가서야 미확정 되었던 요구사항들을 군에서 추가로 제기함으로써 인하여 개발업체는 비용 추가, 사업 지연은 물론, 소프트웨어 자체의 품질에도 결코 좋은 결과를 가져올 수가 없었다.

결국 요구사항을 제대로 정의하지 않고 소프트웨어를 개발하는 행위로 인하여 군은 개발사업이 계획된 일정 내에 수행되지 못하거나, 전체 개발 활동 중 후반부에 수행해야 할 활동들을 남은 일정에 제한을 받음으로 인하여 제대로 수행되지 못하게 되는 결과를 가져오게 된다. 또 개발자인 업체의 입장에서라도 예상치 못했던 추가적인 개발로 인하여 비용 증가 및 사업 일정 지연 등 많은 문제점을 유발하고 있다.

소프트웨어 요구사항을 정의한다는 것은 소프트웨어에 할당된 기능들과 성능들을 바탕으로 소프트웨어의 기능, 성능, 인터페이스, 제약사항 등을 문서로 표현한다는 것을 의미한다.

따라서 요구사항은 이와 같이 소프트웨어가 무엇을 할 것인지를 완벽하게 정의하는 것이며, 이것은 개발단계에서 어떻게 개발할 것인지에 대한 완벽한 기준이 되는 것이므로 될 수 있는 한 초기에 완벽한 요구사항명세서를 만드는 것이 군과 업체 모두에게 성공적인 개발활동을 보장하는 관건이 된다는 사실을 분명히 인지하여야 한다.

* 체계적인 개발활동 미흡

소프트웨어를 탑재하여 구성되는 내장형 시스템(embedded systems)은 일반적으로 마이크로프로세서와 저장매체(메모리) 그리고 외부장비들과 상호작용하기 위한 표준 직렬 포트나 네트워크 인터페이스 등으로 구성되며, 통상적으로 키보드나 스크린, 디스크 드라이브들을 가지지 않는다.

여기에서 내장형 소프트웨어란 부팅시(전원인가) 소프트웨어가 저장매체에서 주기억장치로 이동하여 CPU상에서 동작하거나 또는 고정된 저장매체에서 작동하는 소프트웨어를 말한다. 내장형 소프트웨어는 데스크탑이나 메인프레임 컴퓨터용 소프트웨어에서 발견되는 것보다 많은 문제들을 다루어야 한다.

대부분의 내장 소프트웨어들은 대개 여러 기능을 동시에 수행하는데, 사람의 개입 없이 외부 이벤트들에 대한 응답을 해야 하며, 이때 반응시간에 종속적인

실시간 처리(hard real-time) 특성을 지닌다.

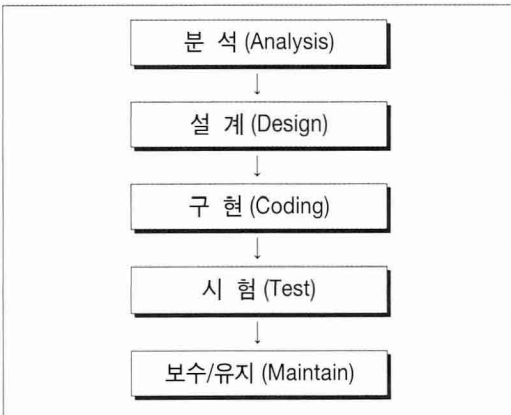
따라서 이러한 내장 소프트웨어의 오동작이나 고장은 물리적, 인적, 환경적으로 막대한 손실을 초래할 수 있으므로 이를 사전에 예방하거나 최소화하기 위해서는 소프트웨어 개발 초기단계부터 요구기능/성능 도출, 설계 및 구현에 이르는 전 과정에 걸쳐 철저한 검증 및 확인이 수반되어야 한다.

앞에서도 언급한 사실이지만, 소프트웨어의 개발과정은 상당히 체계적인 분석, 검토, 설계, 구현, 시험 등 일련의 단계들을 거치면서 개발되어야 요구되는 품질, 비용, 일정을 충족시킬 수 있다. (아래 그림 참조)

이러한 체계적인 활동을 수행하기 위해서는 시스템 엔지니어링, 소프트웨어 엔지니어링 등 공학적인 바탕 위에서만 가능하다. 그러나 불행하게도 우리의 방산 환경, 특히 무기체계의 개발에 있어서는 그렇지 못하다.

무기체계 전체의 요구사항이 정해지면 그에 따라서 소프트웨어로 구현해야 할 요구사항을 정의하고, 분석하는 단계, 이를 토대로 소프트웨어를 설계하는 단계, 구현하는 단계, 시험하는 단계 등 각각의 단계별로 수행하여야 할 활동들이 제대로 이루어지지 않거나, 극단적인 경우에는 아예 무시된 채, 요구되는 기능의 구현에만 노력을 집중하고 있는 형편으로서, 개발된 소프트웨어의 품질과 신뢰성을 확인하는 활동은 대단히 미흡한 형편이다.

소프트웨어의 개발 단계



소프트웨어에 대한 체계적인 개발활동을 수행하지 않으면 각 단계에서 산출되어야 할 기술자료도 제대로 나오지 않는다는 것을 의미하며, 이는 요구사항에서부터, 설계, 시험, 최종 프로그램에 이르기까지의 품질을 보증할 수 있는 자료가 나올 수 없다는 결과를 가져오며, 결국 개발된 소프트웨어의 품질보증과 전력화 이후의 유지보수에도 많은 제한 사항으로 작용하게 된다. 더불어 초기에 예상했던 계획보다 비용, 일정에도 추가의 소요를 유발시키게 된다.

소프트웨어의 개발이 이렇게 진행되는 원인으로는 무기체계 획득절차가 하드웨어 위주로만 수행되고 있는데 기인한다. 국내 무기체계의 획득절차를 명시하고 있는 국방획득관리규정은 훈령676호로 개정되어 2001년 1월 1일 부로 시행하기 전까지 무기체계의 소프트웨어 관리에 관한 내용이 전혀 없었다.

즉, 지난 2000년까지 개발된 무기체계의 경우에는 소프트웨어의 체계적인 개발관리 활동이 될 수 있는 정책적인 장치가 없었던 것이 가장 큰 원인으로 볼 수 있다. 결국 훈령 제676호부터 무기체계 소프트웨어의 관리를 명시하기는 하였으나, 세부적인 시행지침 등은 아직도 확정되지 않고 있는 상태이다.

아울러, 적용표준 및 관리지침 측면에서 보면 연구 개발조직인 국과연 및 개발업체에서는 소프트웨어 관련 국제 품질표준인 ISO 9000-3, 9126, 14598 등에 의한 개발관리 및 표준화가 미흡하고, 특히 개발단계 주요 산출물에 대한 관리 표준인 MIL-STD-498, IEEE/EIA(ISO/IEC) 12207 표준과 같은 체계화된 개발 프로세스 관리 표준이 부재한 실정이다.

따라서 소프트웨어 개발시 개발 단계마다 산출되는 설계자료 및 산출물에 대한 표준화된 문서화 및 개발자료 작성체계가 부재한 상태에서 양산이 되고, 전장환경의 변화에 따른 소요군 요구사항 변경시 소프트웨어 변경관리 및 성능개선에 관한 조치가 효과적으로 또는 경제적으로 수행되지 못하고 있는 결과를 가져오고 있다. (다음호에 계속)