

글로벌 경쟁력 확보를 위한 소프트웨어 개발 프로세스 사례

안유환·최승일·박수연
(주)핸디소프트

목 차

1. 서 론
2. 소프트웨어 제품의 핵심 성공 전략
3. 제품 개발의 Unique 프로세스
4. 결 론

1. 서 론

하드웨어, 수동 업무 프로세스를 포함하는 시스템의 공급에 있어 소프트웨어의 비중이 점점 높아가고 있고(예를 들면 군용 비행기의 소프트웨어 기능 비중이 F-4에서는 8%이었으나, F-22에서는 80%에 이릅니다[7]), 세계적으로 정보통신산업에서 소프트웨어가 차지하는 비중이 22%를 넘어서고 있을 만큼 소프트웨어가 매우 중요한 요소로 대두되고 있다. 국내에서도 소프트웨어 산업이 고용 창출 효과가 큰 인적자원 중심의 산업이고(지식노동 집약적 산업으로서 직·간접 고용유발 효과가 매우 크며, 전체 비용의 60% 이상이 인건비인 산업으로, 고용유발계수(생산 10억원 당 직간접 고용 인원)가 소프트웨어 산업은 15.67명에 이릅니다[9]), 고부가가치의 지식산업이며, 단시간에 초고속 성장이 가능한 산업이며, 타 산업과의 연계가치가 높은 산업임을 고려하여 올해를 소프트웨어산업 발전의 원년으로 삼고 집중 육성하려고 하고 있는 상황이다.

그러나 전세계 소프트웨어 시장에서 국내 소프트웨어 시장이 차지하고 있는 비중이 1%에 불과하고, 주요 경쟁국들이 소프트웨어 분야에서 50위권 내의 세계적인 기업들을 다수 보유하고 있지만 세계 500위권 수준의 한국 소프트웨어 기업은 핸디소프트를 필두로 4개 업체 정도에 불과한 현실이다[4]. 이는 국내 소프트웨어 산업환경이 기업환경 측면에서는 국내 SW기업의 영세성, R&D 투자여력 부족, 국내시장에서의 낮은 점유율 등의 문제를 가지고 있고, 시장환경 측면에서는 국내기업 사이의 과당경쟁, 높은 SW 불법복제율, 자본차입의 어려움, M&A 시장의 부재 등의 문제를 가지고 있는 이유이기도 하겠으나[8], 기본적으로는 패키지 소프트웨어 공급 측면에서는 세계적으로 경쟁할 수 있는 소프트웨어 제품을 지속적으로 개발-공급할 능력을 가지지 못한 점이 문제이고, SI산업 측면에서는 SI 서비스 능력의 글로벌 경쟁력을 확보하지 못한 것이 가장 큰 원인으로 보인다.

세계시장을 주도하는 글로벌 소프트웨어 기

업으로 성장하기 위해서는 물론 전통적인 4P (product, price, place, promotion)가 모두 중요하지만, product의 측면에서 좋은 품질의 소프트웨어 제품을 지속적으로 공급하는 능력이 가장 중요하고, 프로세스의 품질이 향상되지 않는 한 제품의 품질은 보장할 수 없기 때문에 제품 개발 프로세스가 매우 중요하다. 그동안 국내 소프트웨어의 품질 수준을 보면, 국내 소프트웨어 산업체의 생산성(1인당 연간 개발 라인수: 12,700 라인)은 해외(1인당 연간 개발 라인수: 9,400 라인)보다 높은 편이나, 품질과 납기 준수 등 엄격한 기준으로 볼 때 국내 소프트웨어 개발 프로젝트 중 70% 가량이 실패라고 인식될 정도로 제품의 품질과 프로세스 능력이 낮은 점을 고려하면, 글로벌 경쟁력 확보를 위한 제품 개발 프로세스는 매우 중요하다.

본고에서는 2003년말 세계 306위의 소프트웨어 기업으로, 20여개 국가에 300여개의 글로벌 고객을 확보하고, 2005년 32억불 BPM 시장에서 세계 Top 5의 제품 기술력을 가진 핸드소프트웨어의 제품개발 프로세스를 살펴봄으로써, 향후 글로벌 경쟁력을 갖추고자 하는 소프트웨어 기업들이 효과적인 제품 개발의 best practice를 구축할 수 있도록 도움을 주고자 한다. 이를 위하여, 제 2장에서는 일반적으로 중요한 소프트웨어 관리원칙과 소프트웨어 프로젝트의 실패 원인을 간단히 살펴본 후에 핸드소프트웨어의 핵심 성공 전략을 제시하고, 3장에서는 핸드소프트웨어의 unique한 best practices들을 제시하고, 마지막으로 제 4장에서는 현재 열악한 중소 소프트웨어 기업체에서 이를 적용하기 위한 몇 가지 제언을 결론적으로 제시하고자 한다.

2. 소프트웨어 제품의 핵심 성공 전략

2.1 소프트웨어 관리 원칙

최근 수년간 소프트웨어 결함으로 인한 사망자 수가 4,000여 명에 이르고, 대형 프로젝트의 1%만이 품질, 비용, 일정의 목표를 달성하는 등으로 소프트웨어 프로젝트의 실패가 많은데, 그 원인을 보면 소프트웨어 프로젝트가 기술적 요인으로 실패하는 경우는 드물고, 문제는 바로 빈약한 관리에 있다는 것이다. 구체적인 원인은 다음과 같다[7].

■ 비현실적인 스케줄

Tight한 일정이 개발의 속도를 높이지 못함. 요구사항의 졸속 처리, 피상적인 설계, 코딩 질주 등의 개발팀의 불합리한 행동을 유발함. 무리한 일정으로 시작한 프로젝트는 합리적인 계획을 가진 것보다 항상 늦게 끝남.

■ 부적절한 인력 구성

적절한 수의 인원 할당, 방해와 간섭으로부터의 보호, 훈련이 필요함. 개발자를 여러 프로젝트에 할당하면 “경영진이 이 작업에 적극적인 관심을 보이지 않는데 왜 내가 적극적으로 해야 하는가?”라는 반응을 유발함.

■ 개발 중 요구사항의 변경

개발 초기 단계 이외의 요구사항 변경은 시간과 비용의 소모가 막대함. 따라서, 요구사항의 변경 검토가 매우 중요함. 특히 개발자들이 영향을 분석하고 계획을 세우고 일정에 동의해야 함.

■ 낮은 품질의 작업

대충의 설계와 코딩, 품질 검토와 검사 (inspection)의 부재 → 기록적인 시간에 테스트 돌입 → 테스트 기간의 장기화, 재작업 비용 증대, 유지보수 비용의 증대.

■ 기적을 바라는 마음

품질 프로세스를 따라서 제품을 개발했다면 대부분의 경우에 고품질의 제품을 얻을 수 있거나 그러지 않고 기적을 바랍.

이와 같은 문제점을 극복하는 관리 원칙은 여러 가지가 있을 수 있으나 다음의 두 가지가 가장 중요한 것으로 보인다.

■ 품질이야말로 가장 중요한 것이다

1987년 Lotus, MS에 이어 세계 3번째 소프트웨어 회사인 Aston Tate사의 Dbase IV가 Boland사로 인수된 이유는 제품결합이 너무 많았던 품질의 문제임. 시장에서의 적기 진출(Time to Market)도 중요하지만 품질을 무시한 Quick and dirty는 비용을 더 많이 들게 함.

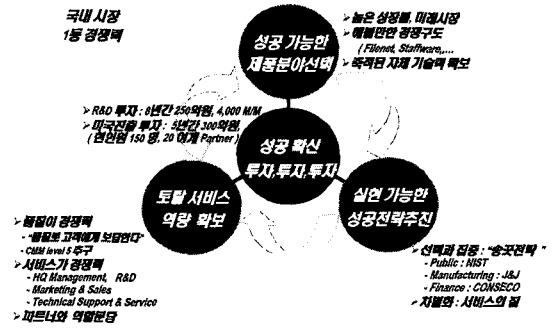
■ 연마(Disciplined)되고 의욕적인 사람이 고품질의 소프트웨어를 개발한다

품질관리 방법을 훈련받지 못한 관리자와 개발자는 품질이 중요하다는 것을 믿지 않음. 소프트웨어 엔지니어는 품질에 집중하지 않는다면 그 이유는 개발자가 게을러서나 의욕이 없어서가 아니라, 그렇게 훈련하고 관리하였기 때문임.

2.2 소프트웨어 핵심 성공 전략

핸디소프트가 글로벌 시장에서 절반의 성공을 거두어 해외에 3,500만불의 누적 수출 실적을 기록한 배경은 (그림 1)과 같이 국내시장 점유율 1위의 시장 경쟁력을 바탕으로, 글로벌 성공을 확신하고 투자를 집중한 것이라고 말할 수 있다. 이를 위하여 성공 가능한 제품 분야를 성장성, 경쟁구도, 확보된 기술력을 바탕으로 선정한 후에, 이 제품의 개발 및 서비스를 위해서는 “품질로 고객에게 보답한다”라는 품질 최우선 정책을 바탕으로 CMM level 5를 목표로 제

품과 서비스의 품질을 높이기 위한 노력을 끊임 없이 해왔다. 마지막으로 시장에서는 각 산업분야별로 주요 시장을 차별화된 제품과 서비스를 바탕으로 집중 공략하는 전략을 활용하여 왔다.



(그림 1) 소프트웨어 성공 핵심 전략

이러한 성공 전략 중에서 역량 확보 측면에서 핸디소프트는 2.1절에서 설명한 소프트웨어 관리 원칙을 매우 중요시 하여 품질 최우선의 정책과 연마된(disciplined) 개발조직을 가장 중요시 하고 있다.

핸디소프트는 (그림 2)와 같이 CMM level 5를 목표로 프로세스를 중심으로 인력과 기술을 포함하는 소프트웨어 프로세스와 프로세스 자산을 지속적으로 개선하고 있으며, 다음과 같은 점에 특히 주력하고 있다.

■ 3단계 품질 검수 체계 운영

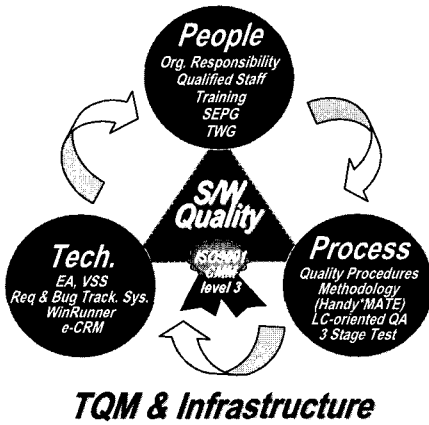
개발팀의 단위테스트 후에는 통합테스트 과 정부터 시스템 테스트, 설치테스트까지를 별도의 테스트팀인 제품관리팀이 독립적으로 테스트를 수행하고 이를 다시 품질경영실 또는 릴리즈 검토회의를 통하여 독립적으로 검토함.

■ 개발자에 대비한 설계 및 품질관리자의 수를 1:1로 유지하는 것을 목표로 함.

현재는 2:1 정도의 비율로 유지하고 있음.

■ CMM level 5의 프로세스 능력을 목표로 함
 2000년 ISO 9001 인증 획득, 2001년 CMM Level 2, 2002년 CMM level 3 획득에 이어, 2004년에는 소프트웨어 업계 최초로 신품질혁신상을 수상하였음. 2005년에 CMM level 5에 도전할 예정임.

“품질로 고객에게 보답한다”
Competitive Advantage



(그림 2) 품질최우선 정책

개발조직은 품질경영 교육을 비롯한 인스펙션 교육을 매년 정기적으로 받은 훈련된 인력이 미국과 한국의 협업하여 공동 개발하는 글로벌 개발조직을 가지고 있다. (그림 3)은 Global PARK 라는 개념에서 필수적인 개발자 역할 및 자산을 보여 주고 있으며, 그 특징은 다음과 같다.

■ PARK 개발 체계

- 다음의 핵심 역할자 양성 및 할당
- 제품관리자(Product Manager)
 - 품질보증 담당자(QA Leader)
 - 아키텍트(Tech. Architect)
 - 지식 재활용(Knowledge Recycle Process(필요 지식 정의, 생성, 평가, 활용 사이클)을 위한 자산 및 전문가(Code Guru)

■ 분야별 전문 TA(technical Architect) 그룹

■ 전문개발자 경력 관리

Manager, Consultant, Architect, Application Engineer, Programmer (Senior, Junior), QAM(QA manager), Test Manager, Tester

■ PM 인증 제도

PM 교육/시험/인터뷰를 통한 사내 인증 제도, PM에 수당 지급, 1년 이내에 PMP 자격증 취득 조건



(그림 3) G-PARK: 선진화된 개발 조직

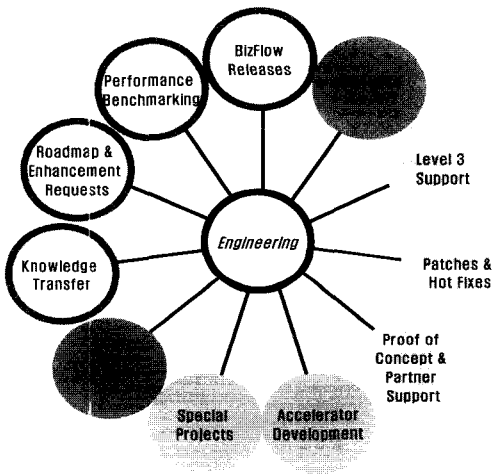
3. 제품 개발의 Unique 프로세스

핸디소프트의 제품 개발 공학은 다음과 같은 철학을 가지고 있으며, (그림 4)와 같이 단순한 제품의 설계 및 개발 외에 유지보수의 3단계 지원, 특별 프로젝트 수행, 제품 로드맵 개발 등의 여러 업무를 복합적으로 수행하여 영업/마케팅, 서비스 수행 조직과 밀접하게 협력하고 있다.

- 제품은 고객과 협력사들에게 잘 받아들여져야 함
 - 부가 수익을 유도할 수 있어야 함.
 - 향후 영업을 위한 레퍼런스를 확보할 수 있어야 함.

- 제품은 견고하여야 함
 - 유지보수에 드는 비용이 수익을 악화시키지 않아야 함.
 - 고객에게 약속한 제품을 제공하고 고객 기대에 부응함.
 - 부가 수익을 유도하기 위해 고객과 협력사들에게 잘 받아들여져야 함.

■ 제품의 각 릴리즈는 각각의 생명주기를 가짐



(그림 4) 제품공학의 범위

3.1 글로벌 요구공학 프로세스

핸디소프트는 글로벌 제품을 공급하기 위하여 미국법인/한국법인이 협업하는 글로벌 요구공학 프로세스를 가지고 있으며, 요구사항 개발 프로세스는 아래와 같은 특징을 가지고 있다.

- 로드맵과 개선의 입력 메커니즘
 - 월별 International Meeting 및 North America Meeting
 - 요구사항 제안자는 다음과 같은 Business Case 및 자세한 사항을 제공함: 잠재적 수익, 가능한 시장점유율 개선 사항, 가능한 고객 구매의 확대 사항, 고객 계획 및 개선

의 장점 등의 구체사항

■ 요구사항의 원천

- International: 한국/일본 법인의 마케팅/영업, 한국 법인 R&D, International 영업
- 북미: 미국법인의 제품/솔루션 마케팅, 영업 및 영업 지원팀, 서비스(고객지원, 전문 서비스, 교육), 공학팀(솔루션 설계팀, 설계 및 공학팀, 개발팀, 품질보증팀)
- 고객 및 협력사

■ 요구사항의 다양한 분석

- 마케팅팀: 경쟁 및 요구의 우선 순위 분석, 시장 및 제품 방향에 대한 전략 명세화
- 설계 및 공학팀: 요구의 우선순위에 대한 심사, 전략 수립(기술 혁신, 품질 보장, 유지보수 및 localization 용이, OEM 및 제품 컴포넌트 제공 전략을 포함)

■ 릴리즈 개요 설정

- 설계 및 공학팀: 릴리즈의 목표일자를 마케팅과 협업, 릴리즈를 위한 'Top 10' 요구사항을 설정하기 위하여 마케팅/한국법인/일본법인과 협업

■ 로드맵 개발

- 설계 및 공학팀: 요구사항의 우선 순위 확정, major 개선 사항에 대한 목표 일정 수립, 내부/외부 개발 로드맵 설정, major 릴리즈에 대한 theme 설정 및 각 개선사항을 목표 릴리즈에 할당

■ 분기별 경영층 검토회의: 로드맵의 검토

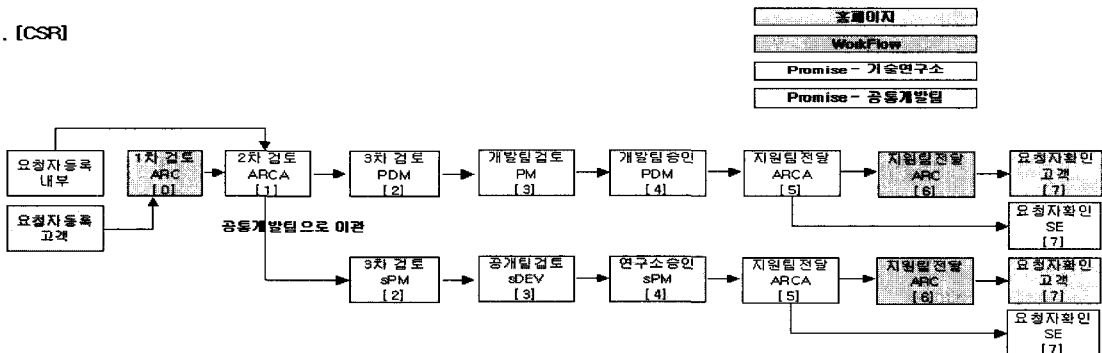
그리고, 요구사항의 관리 측면에서는 각 제품 개발 프로젝트에서 다음 사항이 발생하는지를 중점적으로 검토하여 요구사항을 명확하게 유

지하고 무제한으로 변경되거나 확장되지 않도록 통제한다.

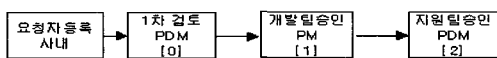
- 프로젝트의 비전과 범위가 한번도 명확하게 정의되지 않았다.
- 고객들이 너무 바빠서 요구사항에 대하여 분석가 또는 개발자들과 협의하지 않았다.
- 제품 관리자, 개발 관리자, 사용자 관리자, 마케터 등과 같이 사용자의 위임을 받은 사람들이 사용자를 대변한다고 주장하지만, 사용자의 요구를 정확하게 대표하지 않았다.
- 고객들이 모든 요구사항이 중요하기 때문에 우선순위를 결정하지 못했다.
- 개발자들이 코딩을 할 때 잘못된 정보 또는 애매모호함 때문에 추측을 해야만 했다.

- 개발자와 고객 간의 대화가 사용자가 소프트웨어를 사용하여 하는 일이 아닌, 사용자 인터페이스 디스플레이에 집중되어 있다.
- 사용자가 요구사항에 승인한 후에 이것을 끊임없이 변경했다.
- 요구사항 변경에 따라 프로젝트 범위가 넓어지는 반면에, 추가적인 지원이 없거나 이전의 기능을 모두 구현해야 하기 때문에 일정이 지연되었다.
- 요청된 요구사항 변경이 어디론가 사라져버렸고, 여러분과 고객들은 모든 변경 요청의 현재 상태에 대해 알지 못했다.
- 고객이 특정 기능을 요청했고, 개발자들은 이것을 구현했지만 아무도 사용하지 않았다.
- 명세가 충족되었지만 고객은 만족하지 않았다.

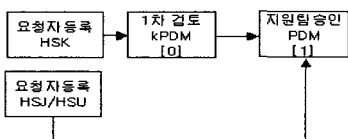
1. [CSR]



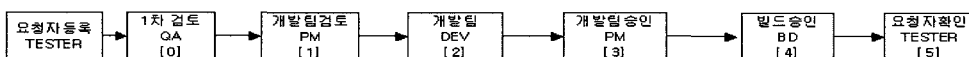
2. [RMT]



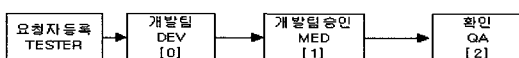
* WF Product 선택시 RMT



3. [DMT]



4. [Inspection]



(그림 5) Promise 시스템의 주요 프로세스

3.2 고객 요구 및 결함의 추적 시스템

소프트웨어의 품질을 관리하기 위한 기본적인 품질 목표로서 활용되는 것이 릴리즈된 제품의 결함 수 및 심각도, 서비스 기간 중의 고객 불만의 수 및 심각도 등이다[2]. 릴리즈 후 또는 서비스 중인 소프트웨어의 결함 및 고객 불만을 줄이기 위해서는 개발 과정에서 결함을 발견하고 제거하는 일이 매우 중요하다. 개발과정에서 50%의 결함 감소를 보이도록 결함 예방조치를 시행한 어떤 프로젝트에서는 릴리즈 이후에 78%의 결함 감소효과를 보이는 사례도 있다는 점을 보면 이를 잘 알 수 있다[3]. 개발 과정에서 결함을 제거하기 위한 테스트와 인스펙션을 효과적으로 통제하기 위해서는 가장 중요한 요소가 바로 오류/장애/실패(통칭하여 결함(defects))를 효과적으로 추적하는 것이다[3][6].

3.2.1 Promise 시스템의 주요 기능

Promise 시스템은 고객의 불만 및 요구사항, 인스펙션 및 테스트 시의 결함에 대하여 각 담당자별로 업무를 할당, 추적할 수 있게 하고, 결함 및 요구사항에 관한 각종 상태정보와 통계정보를 제공하여, 각 개발 및 테스트 단계별로 결함 제거 목표의 수립, 정량적 통제, 결함 원인 분석을 통한 프로세스 개선 등을 지원하는 시스템이다.

Promise 시스템에서 처리되는 요청서의 경로는 eCRM, CSD, Promise 자체 등록의 3가지의 Process가 있어, 고객 요청관리 시스템, 통합고객관리 시스템으로부터 개발이 필요한 요구사항 및 결함보고를 넘겨받아 처리하고 그 결과를 해당 시스템으로 돌려주는 역할을 한다. 또한 Promise 시스템의 내부적으로는 (그림 5)에서 보는 바와 같이 아래의 주요 서브시스템으로 구분되어 요구사항관리와 결함관리(인스펙션으로부터의 결함보고 포함) 기능을 제공한다.

- CSR: 릴리즈 이후에 발생하는 사이트 요구사항과 결함 등을 처리
- RMT: 개발 단계의 제품에 대한 요구사항을 관리
- DMT: 개발 단계의 제품에 대한 결함을 등록하여 처리
- Inspection: 개발 단계의 제품에 대한 동료 검토 결과를 등록하여 관리

Promise 시스템은 다음과 같은 주요 기능들을 가지고 있다[1]:

〈표 1〉 Promise 시스템의 기본 통계 정보 항목

| 항 목 | |
|-------|----------------------------|
| 등록 현황 | 날짜별 요청서 타입 현황 |
| | 날짜별 요청구분 현황 |
| | 날짜별 결함심각도 현황 |
| | 기능별 요청구분 현황 |
| | 기능별 우선순위 현황 |
| | 기능별 결함심각도 현황 |
| | 기능별 역할 현황 |
| | 개발자별 결함심각도 현황 |
| | 사이트별 요청구분 현황 |
| 진행 현황 | 기능별 진행상태 현황 |
| | CSR MyToDoList 현황 |
| | DMT MyToDoList 현황 |
| | 개발자별 진행상태 현황 |
| | 요청자별 진행상태 현황 |
| | 결함심각도별 진행상태 현황 |
| 주요 통계 | 결함유입별 결함발견 현황 |
| | 버전별 릴리즈 이후 현황 |
| | 개발자별 평균처리시간 현황 |
| | 날짜별 반송처리 현황 |
| | 단계별 결함유형 현황 |
| | 동료검토 보고서 Inspection 효과성 |

1. 사용자 로그인: 사내 그룹웨어를 통하여 접속 가능
2. MyToDoList: MyToDoList는 현재 자신이

처리해야 하는 일들을 보여주며(요청서 ID, 사이트/요청자, 제목, 요청일자, 상태, 우선순위, 발견버전, 릴리즈계획 등의 항목) 검색과 이메일 노티의 설정 등이 가능함. 목록에서 제목을 클릭하면 자신의 역할에 따라 요청 처리화면이 나타남.

3. 상세 MyToDoList: 현재 자신이 처리해야 하는 일들을 상세하게 보여줌(요청내용과 답변내용, 히스토리까지 상세한 정보).
4. MyDoneList: 자신이 등록했거나 처리한 모든 사항에 대한 요청서 목록을 보여주며, 목록 중에 현재 처리되고 있는 진행 상태를 확인하고 싶을 경우 해당 요청의 [상태]를 클릭하면 현재 진행 중인 상태를 확인 가능.
5. 등록 및 처리 : CSR, RMT, DMT, Inspection 요청서를 등록하고 처리하는 기능으로 각종 해당 담당자가 각종 상태정보와 처리 내역을 역할별로 입력. 처리함: CSR 등록, CSR 처리, RMT 등록, RMT 처리, DMT 등록, DMT 처리, Inspection 등록, Inspection 처리.
6. 빌드 승인: 각 제품의 릴리즈별 빌더의 역할로 지정된 사람이 빌드 상태의 요청서를 한꺼번에 빌드 승인하는 기능이며, 모든 사용자가 [빌드승인] 메뉴를 통해 현재 빌드 승인 상태의 요청서 목록과 내용을 확인할 수 있으며 빌더만이 [approve] 기능을 수행할 수 있음.
7. 검색: 빠른검색, 상세보고서, 고급검색의 세 가지 주요 기능으로 강력한 검색 기능 제공.
8. 통계: 다양한 통계보고 기능을 제공한다. 등록현황, 진행현황, 주요통계를 제공한다.
9. FAQ: 요청서의 내용과 답변을 FAQ 게시판으로 등록하여 자주 문의되는 오류 및 요청내용과 그에 대한 답변을 공유 가능함.
10. 시스템 관리: 사용되는 제품(Product)과

각각의 버전(Version), 기능(Component), 사용자(User) 그룹, 시스템에서 사용하는 코드(Code), 제품이 설치된 사이트(Site) 등의 정보를 등록하고 관리하는 역할.

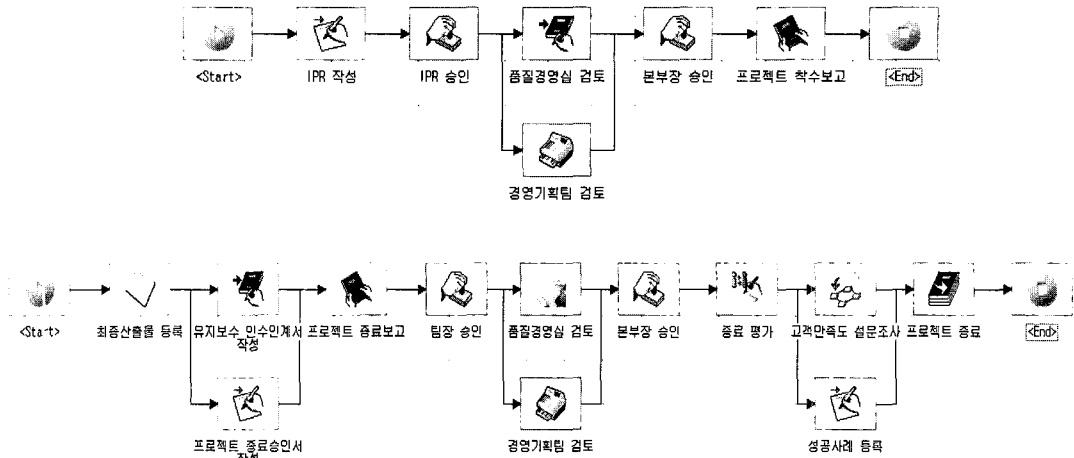
11. 기타 릴리즈 정보 제공 및 도움말

각 제품별, 릴리즈별, 기간별, 요청 구분별, 개발자별, 결과상태별로 등록현황, 진행현황, 주요통계 등이 제공되며, 그 세부 기능들은 <표 1>과 같다. 또한 화면으로 제공되는 모든 결과들은 Excel file로 export가 가능하여 필요한 통계자료의 추가 가공 등이 가능하도록 한다.

3.3 프로젝트 관리의 unique한 실무활동

프로젝트 관리에 있어서도 다음과 같이 몇 가지 unique한 훌륭한 실무활동들을 가지고 있다.

- Daily Tea Meeting: 개발팀 일일 업무 시작회의
- Time card 작성(시스템으로 비용/공수 추적)
- 주간 보고
 - PM 및 본부장 관리 사항: 진척도, 위험, 이슈 관리
 - 경영분석회의: 주요 프로젝트의 위험 및 이슈 사항, 프로젝트 진척도, Inspection 현황, 계약 이슈 현황, 결함 보고 및 Promise 지연 현황
- 월간보고
 - 연구소 관리 사항: 릴리즈 시의 제품 결함률도, 릴리즈 후의 결함 보고 추이, 프로젝트 공수 투입 현황(billable, non-billable 등을 몇가지로 구분), 프로젝트 비용 집행 현황(IPR 대비), 프로젝트 진척도 및 이슈, 사이트, 주요 릴리즈 계획 변경
 - 서비스 팀 관리사항: 제안 및 계약 현황, 프로젝트 진척도, Cost recovery 현황, 가동률 등
- 분기별 보고 : 분기별 종합 보고, 분기별 실행 계획



(그림 6) 프로젝트 착수 및 종료 프로세스

| 영역 | 단계 | 제품 계획 단계 | 제품 정립 단계 | 릴리즈별 반복 개발 | | | 유지보수 단계 |
|-------------|-----------------------------------|-----------------------------------|--|---|---|--|--------------------------------|
| | | | | 잠정적 설계 및 코딩 단계 | Freezing 단계 | 릴리즈 단계 | |
| 제품 문서 개발 | 기술관경 기능 정적행위 동적행위 테스트 | 제품비전 정의 | 시제품 개발 | 제품 문서 및 교육자료 개발 - 사용자 매뉴얼 - 운영자 매뉴얼 - 유체해결집 - 설치매뉴얼 | | S/E 교육 | 수정사항 구명 및 시험 유지보수 요구사항 및 분석 |
| | | | 제품요구서 위험분석 | 제품개발계획 수립 | 요구정제 및 설계 - 사용자 요구 정제 - 클래스 정제 - DB 설계 - UI 설계 - 아키텍처 정제 | 코딩 및 단위 테스트 - 코딩 단위 테스트 - 단위 테스트 | |
| 프로젝트 관리 | 계획단계 준비 프로젝트 계획 수립 | 정립단계 준비 | 시브프로젝트 준비 | 진척관리 | | 제품 릴리즈 | 유지보수 계획 수립 |
| | | | 작업계획 | 문서관리 | | | |
| Infra구축 | 제품계획 | 품질계획 | 행서관리 | | | 시험 테스트 | 형상관리 |
| | | | 품질평가 | | | | |
| 품질보증 | 계획단계 점검 | 정립단계 점검 | 감도회의 | | 코드검사 | 릴리즈 점검 | |
| | | | | | | | |
| 관련 Infra/표준 | - 제품기능 명세서 - 프로젝트 계획서 | - 제품명세서 - 제품개발 계획서 - 제품개발체제 | - 작업계획 - 테스트계획 - Comm. 절차/도구 - 설계명세서 - 설계검토 지침 | - 코딩 지침 - 코드인스펙션 지침 | - 시험시나리오/ - 테스트리스트 - 시험case/도구 | - 릴리즈 note - 설치매뉴얼 - 사용자매뉴얼 | - Patch 발송기록 - 변경관리 기록 |

(그림 7) Handy*MATE-PD의 viewpoint

3.3.1 프로젝트 착수 프로세스

위 사항 외에 프로젝트의 착수 시에는 다음과 같이 프로젝트 계획서와 함께 IPR (Initial Project Report)를 작성하여 초기에 비용과 일정, 자원을 명확히 정의하도록 하고 단계적으로 시스템을 통하여 승인과정을 거친다(그림 6 참조). 프로젝트의 진행 중에는 이를 기반으로 비용과 일정, 자원 들을 통제한다.

1. PM이 릴리즈를 위한 Master Requirements

document를 작성: 개선 요구사항과 새로운 feature에 대한 요구사항을 grouping → 주요 마케팅 또는 구조적 주제와 맞는지 분석 → 요구사항에 우선순위 부여 → 요구사항이 서브 릴리즈로 지정됨.

- PM이 Draft Project Plan을 수립: 서브릴리즈의 수, 반복(Iteration)의 수, 주요 마일스톤 등.
- PM과 D&E가 프로세스를 선택 및 조정(방

법론/도구/환경의 조정).

4. PM이 지원되는 추천 구성(Configuration)을 수립: PM과 Software Architect가 지원 버전을 설정하고, PM이 릴리즈에 대한 추천 구성을 선택.
5. PM이 Architect 및 개발팀, PQM팀과 협력하여 필요한 공수 산정: 버그 수정 대비 새로운 기능의 비율, 아웃소싱의 비율, 참여 개발자의 경험 수준 등을 고려.
6. PM이 릴리즈에 대한 산출물 및 목표를 수립하여 공표: PQM팀, 고개지원팀, 프로젝트 수행팀, 마케팅, 사업본부에 공지하고 피드백.

3.3.2 프로젝트 종료 프로세스

프로젝트 종료시점에서는 프로젝트의 산출물과 데이터, 교훈들을 재활용할 수 있도록 다음과 같은 프로세스를 따른다(그림 6 참조).

- 프로젝트 PM은 최종 산출물을 등록한다.
- 프로젝트 PM은 유지보수 인수인계서를 작성하여 고객지원팀에게 전달한다.
- 프로젝트 종료승인서를 작성하여 품질경영실에 전달한다.
- 프로젝트 PM은 프로젝트 종료보고를 수행한다.
- PM의 소속 팀장은 프로젝트 종료를 승인한다.
- 품질경영실과 경영기획팀이 프로젝트 종료에 대한 검토를 수행한다.
- 소속 본부장이 프로젝트 종료를 승인한다.
- 프로젝트에 대한 종료 평가가 수행된다.
- 프로젝트 고객만족도 설문조사를 수행한다.
- 우수한 프로젝트를 성공사례로 등록한다.
- 프로젝트를 종료 처리한다.

3.4 개발방법론 및 개발 프로세스

3.4.1 개발방법론

소프트웨어 개발을 위해서 핸디소프트는 아래

와 같은 Handy*MATE(Methodology Accompanied with Technology Enhancement) 시리즈라는 제품별 표준 개발방법론을 개발하여 적용하고 있다.

- Handy*MATE-PD: 패키지 제품 개발용
- Handy*MATE-BPM: BPM기반 시스템 개발용
- Handy*MATE-KD: 지식관리 시스템 구축용
- Handy*MATE-GD: 그룹웨어 기반의 시스템 개발용
- Handy*MATE-Web: 웹 애플리케이션 개발용
- Handy*MATE-PRAD: 패키지의 신속 개발용

Handy*MATE는 아래와 같은 방향으로 개발되어 실무에 잘 조정하여 적용할 수 있도록 하고 있다.

- 현행 practice의 약점을 보완: 현행 practice를 최대한 활용
- 최소한의 산출물 위주로 정리: 유지보수용 최소 문서(설계서, 코드의 주석문, 코딩 표준의 준수) 및 필수 제품 문서(사용자 매뉴얼, S/E 매뉴얼) 위주
- 산출물의 구체적 작성 지침 제공: 산출물별 필수 항목 및 작성 요령 정리, 기법에 dependent한 항목의 작성방법은 별도의 지침서로 정리
- 업무에 필요한 필수지침 위주로 정리: Inspection 지침, 테스트 지침/체크리스트
- 개발자의 관점 외에 관리/지원 활동과의 조화: PM, QA, 개발자 그룹의 업무 관점 포함
- 개발 작업과 유기적으로 연결되도록 활동 및 작업 정의
- 관련 표준에의 적합성 검토
- 실제 프로젝트에 적용 및 유효성 확인

패키지용 제품 개발에 적용하는 방법론인 Handy*MATE-PD는 (그림 7)에서 보는 바와 같이 필수적인 영역의 관점을 모두 포함하면서 다

음의 특징을 가지고 있다.

- 작업의 주기적 동기화, 반복적 증가 병렬 개발
- 단계별 출시(Phased Release) 반영: 개발과 유지보수 중첩
- 객체지향 개발방법의 적용: UML 및 아키텍처 중심의 개발, RAD 및 진화적 개발 지원
- 구축단계의 오류 수정/방지 강화: 디버깅, 코딩 지침의 준수, 코드 인스펙션
- 시스템 테스트의 강화: 제품관리팀에 의한 시스템 수준시험(기능시험 중심)과 성능테스트, 설치티스트, 환경테스트

적용되는 방법론은 Handy QMS라고 하는 웹기반의 프로세스 관리시스템에서 절차 및 산출물 템플리트를 쉽게 액세스할 수 있게 하고 있다.

3.4.2 개발 실무 지침

개발 과정에서 여러 가지 사항이 중요하지만 특히 아래와 같은 사항을 중점적으로 검토한다.

- 개발자들은 상세설계서를 보고 코딩하는가?
 - 개발 과정 중에 산출물의 일관성, 추적성은 확보되어야 함.
 - 요구사항의 변경에 따라 다른 산출물 (설계서, 코드, 테스트 시나리오)은 일관성 있게 변경 관리되어야 함.
 - 명세서는 일관성과 정확성이 최우선 사항임 (완전성의 범위는 고려 필요).
- 표준 방법론을 프로젝트에 조정하여 사용하는가?: 최소한의 산출물/deliverables이 결정되었는가?
- 소프트웨어 요구사항명세서(SRS)가 다음을 포함하는가?
 - 새로운 features, enhancements, and architectural updates에 대하여 Engineering Requirements가 명세화 되는가?
 - high level Use Cases를 포함하고 있는가?
 - 이전 버전과의 호환성, 업그레이드 목표가 검토되는가?
- 각 단계별 공식검토가 수행되는가? 명세서의 개발, 코딩 과정에서 Walkthrough 및 Inspection

〈표 2〉 인스펙션 기초 데이터

| No | ID | 부서명 | 제품 | 버전 | 제목 | 월 | 회의일자 | 단 기 | 참 가 자 | 준 비 기 | 검토 크 기 | 총 준 비 시간 | 검토 비 율 | 결 합 비 율 | 제 거 비 율 | 재작 업 비 율 | 총 준 비 시간 | 평 균 준 비 시간 | 검토 시간 | 총 시 간 | 준 비 속 도 | 검토 속 도 | 결 함 비 율 |
|----|-----|-------|------|--------------|--|----|-------|----------|-------------|-------------|--------------|-------------------|--------------|------------------|------------------|-------------------|-------------------|------------------------|----------|-------------|------------------|--------------|------------------|
| 1 | ### | 공인개발팀 | PPG | RPG 1.0 | [Groupware RPG1.0] mGlobal.bas 2004-08-01 | 6월 | ##### | 코드 검토 | 3 | 508 | 508 | 120 | 180 | 3 | 2 | 20 | 2 | 0.67 | 3 | 11 | 758 | 169 | 0.59 |
| 2 | ### | OM개발팀 | EDMS | EDMS 2.0 | [EDMS 2.0 기능정의서] 2004-06-01 | 6월 | ##### | 상계 검토 | 5 | 6 | 6 | 60 | 60 | 4 | 4 | 1081 | 1 | 0.2 | 1 | 6 | 30 | 6 | 666.67 |
| 3 | ### | 공인개발팀 | RTM | RTM 1.1 | [Groupware RTM 1.1] SaveProcData.asp 2004-06-03 | 6월 | ##### | 코드 검토 | 3 | 1419 | 1419 | 80 | 120 | 4 | 4 | 45 | 1.33 | 0.44 | 2 | 7.32 | 3225 | 710 | 0.28 |
| 4 | ### | GTEC팀 | GTEC | GTEC 1.0 | [GTEC 1.0] GuestServerShutDown.spc.doc 2004-06-03 | 6월 | ##### | 상계 검토 | 6 | 3 | 3 | 30 | 30 | 2 | 2 | 0 | 0.5 | 0.08 | 0.5 | 3.48 | 38 | 6 | 66.67 |
| 5 | ### | OM개발팀 | EDMS | EDMS 2.0 | [EDMS 2.0 기능정의서] 2004-06-07 | 6월 | ##### | 상계 검토 | 5 | 60 | 60 | 60 | 60 | 3 | 3 | 1442 | 1 | 0.2 | 1 | 6 | 300 | 60 | 50 |
| 6 | ### | 공인개발팀 | PPG | RPG 1.0 | [Groupware RPG1.0] fmOrganizationIm 2004-06-08 | 6월 | ##### | 코드 검토 | 3 | 357 | 357 | 90 | 80 | 3 | 0 | 180 | 1.5 | 0.5 | 1.33 | 5.49 | 714 | 268 | 0.84 |
| 7 | ### | GTEC팀 | GTEC | GTEC 1.0 | [GTEC 1.0] EasierPrint and Click - Visible/Enable By Credit 2004-06-10 | 6월 | ##### | 상계 검토 | 6 | 1 | 1 | 30 | 40 | 6 | 6 | 0 | 0.5 | 0.08 | 0.67 | 4.5 | 13 | 1 | 4020 |
| 8 | ### | 공인개발팀 | RTM | RTM 1.1 | [Groupware RTM 1.1] GetServiceBus().asp 2004-06-11 | 6월 | ##### | 코드 검토 | 3 | 431 | 431 | 180 | 80 | 4 | 4 | 270 | 3 | 1 | 1.33 | 6.99 | 431 | 324 | 0.93 |
| 9 | ### | OM개발팀 | 자료관 | Archieve 2.0 | [Archive 2.0 개발소스 (승격중기)반형] 2004-06-11 | 6월 | ##### | 코드 검토 | 8 | 1860 | 1860 | 60 | 2 | 9 | 9 | 1820 | 1 | 0.13 | 0.03 | 1.28 | 14308 | 62000 | 0.48 |
| 10 | ### | 공인개발팀 | RTM | RTM 1.1 | [Groupware RTM 1.1] MonitoringFormatorB assuaa 2004-06-16 | 6월 | ##### | 코드 검토 | 3 | 728 | 728 | 120 | 60 | 1 | 1 | 30 | 2 | 0.67 | 1 | 5.01 | 1087 | 728 | 0.14 |

이 수행되는가?

-기능적 요구사항 외의 비기능적 요구사항이 포함되는가?

- 구조설계가 다음을 포함하는가?: 전체적인 architectural strategy가 포함되어 있는가?
- 로컬라이즈, 유지보수 지원, 제품의 컴포넌트화 등에 대한 지침이 포함되는가?
- 주요 기술적 개념에 대한 프로토타이핑이 이루어졌는가?
- 다음의 표준 및 지침이 준비되고 이것을 따르고 있음이 검토되는가?: 사용자 인터페이스 지침, 설계 표준, 설계 검토 지침, 코딩 표준 (성능 관점 코딩 표준, 언어별 코딩 표준)

3.5 인스펙션 및 테스트

현재 제품의 품질을 높이기 위한 노력으로 집중적으로 개선되고 있는 부분이 인스펙션(Inspection)의 효율성 제고 노력이다. 특히 제품 개발을 하는 연구소에서는 이를 필수적으로 시행하여 준비시간(preparation time), 결함 유형, 인스펙션 속도 및 효과성 중심의 정량적 데이터의 수집·보고 체계를 가동 중이다. <표 2>는 기초 데이터 수집을 위한 데이터시트를 보여주고 있다. 수집된 데이터를 분석하여 주별/월별/분기별로 품질경영실이 경영분석회의에 보고하고 개선하고 있다. 인스펙션의 결과로 발견된 결함은 Promise시스템을 통하여 해결될 때까지 추적 관리되고 분석을 위한 통계정보도 제공된다.

테스트는 2장에서 설명된 바와 같이 제품 개발팀에 의한 단위테스트, 제품관리팀에 의한 통합 및 시스템 시험, 설치시험 등을 거쳐 최종적으로 릴리즈 검토회의에서 독립적으로 검토되어 최종적으로 릴리즈가 결정된다. 릴리즈 검토회의에서는 요구사항 목록에 대한 확인 결과, 각종 결함 현황(일자별, 기능별, 처리 현황별) 등이 보고되고 보고되는 결함 현황은 Promise

시스템의 통계 결과를 활용한다.

4. 결론

이제까지 소프트웨어 제품의 글로벌 경쟁력을 확보하기 위해서는 제품의 품질을 보장하고 생산성을 높이기 위한 여러 실무활동들이 필요함을 설명하고 그 중 몇 가지 실제로 적용되고 있는 실무활동 및 프로세스를 사례로 설명하였다.

글로벌 경쟁력을 확보하기 위해서는 프로세스가 지속적이며 주기를 따르며 개선되어야 한다. 하지만 국내 소프트웨어 산업의 대부분을 차지하고 있는 중소 소프트웨어 업체에서는 제품 품질의 현상태에 불만족하여 품질개선과 프로세스 개선은 분명 도움이 될 것이라는 생각을 가지고 있는 반면에 필요한 자원과 역할에 따른 인력의 할당이 어렵기 때문에 개선에 소요되는 비용과 자원을 감당할 수 있느냐? 일부만만 한다면 무엇부터 해야 하느냐? 등의 고민을 가지고 있다.

이를 해결하기 위한 방법은 첫째로, 중소 규모 조직에서의 프로세스 개선은 사업에 초점을 맞추고(business-focus), 적고(less) 더 좋은(better) 프로세스를 요구한다는 것이다. 즉, 대규모 조직에 적용되는 대부분의 프로세스가 요구하는 실무활동들이 조정되어야 한다는 것이며, 개선의 우선순위를 잘 파악하여 개선하여야 한다는 것이다. 물론, 조정의 범위를 보면 대부분의 핵심 실무 절차를 생략하는 것이 아니라 작업 산출물의 상세도와 핵심 실무 절차의 수행 엄밀성을 조정해야 한다는 것이다.

둘째로, 일반적으로 인정되는 우선순위를 적용하여 검증(Verification)(동료 검토 포함), 위험 관리, 교육훈련, 프로세스 정의, 품질보증 등의 순서로 우선 적용을 하는 방법이 있다.

셋째로는, 개선을 위한 기반구조가 없으면 체계적인 진행이 안되므로 반드시 이를 조기에 구

축하도록 한다는 것이다. 조직적인 측면에서는 개선을 주도하는 그룹(Management Steering Committee, SEPG 그룹, 프로세스 개선 그룹(Process Improvement Team) 등)이 반드시 필요하며, 소규모 조직에서는 이들 조직을 part-time 참여자로 할 수 있다는 것이다. 기술적인 측면의 기반구조인 프로세스 정의도 없는 것을 개선할 수는 없기 때문에 있어야 하나, 프로세스 문서는 간결하게 유지하도록 한다.

위와 같은 방법을 통하여 점진적으로 프로세스를 개선하기 시작하고 개선 과정에서 일부 내재화 되는 과정에서의 생산성 저하 등으로 발생되는 문제에 실망하지 않고 단계적이고 점진적으로 프로세스를 개선하면 글로벌 경쟁력을 갖춘 제품과 개발 능력을 확보할 수 있을 것이다.

참고문헌

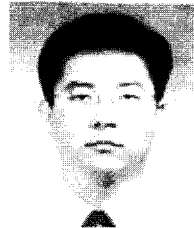
- [1] Promise 2.5 사용자 매뉴얼, 핸디소프트, 2004
- [2] Bill Hetzel, "The Complete Guide to software Testing" 2nd Ed. John Wiley & Sons, Inc. 1988.
- [3] <http://testingfaqs.org/t-track.html>
- [4] IDC 2004, "Worldwide Software 2004-2008 Forecast Summary."
- [5] Mark C. Paulk and et al., "The Capability Maturity Model: Guidelines for Improving the Software Process," Addison-Wesley Publishing Company, 1994.
- [6] Watts S. Humphrey, "Managing the Software Process", Addison-Wesley Publishing Company, 1989.
- [7] Watts S. Humphrey, "Winning with Software: An executive Strategy", 2001.
- [8] 소프트웨어진흥원, "정책연구 보고서", 2003.
- [9] 프로그램 심의조정위원회, "산업연관분석 연구보고서" 2004.12.

저자약력



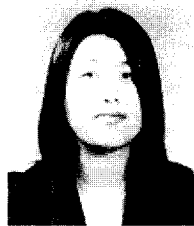
안 유 환

1984년 서울대학교 산업공학과(학사)
 1986년 한국과학기술원 경영학과(석사)
 2000년 한국과학기술원 테크노경영대학원 경영공학전공(박사)
 1986년~2000년 한국전자통신연구원 소프트웨어 품질보증연구팀장
 2000년~현재 ㈜핸디소프트 품질경영실장, 연구본부장, EKP사업본부장/상무이사
 E-mail: ywahn@handysoft.co.kr



최 승 일

1992년 한양대학교 물리학과(학사)
 1992년~1995년 ㈜제이씨이 개발팀
 1995년~1996년 ㈜녹십자 전산실 1996~2001 ㈜동양시스 템즈 RM팀
 2001년~현재 ㈜핸디소프트 품질경영실장
 E-mail: sichoi@handysoft.co.kr



박 수 연

2000년 전주대학교 경영정보학과(학사)
 2002년 고려대학교 경영학과 MIS전공(석사)
 2002년~현재 ㈜핸디소프트 품질경영실, 책임연구위원
 E-mail: sypark@handysoft.co.kr