

효율적인 소프트웨어 개발 방법에 관한 연구

권 영직 (대구대학교 전산정보학과 교수)
이 영달 (제일정보통신 소프트웨어 개발부)
손 관호 (박사과정 수료, 대구대학교 강사)
장 명희 (대구대학교 강사)

제 1 장 서 론

소프트웨어 개발 수명 주기는 크게 시스템 개발 수명 주기(System development life cycle : SDLC)와 프로토타이핑 개발 수명 주기(Prototyping development life cycle : PDLC)방법으로 대별할 수 있는데, SDLC는 소프트웨어 개발 단계에 대해 확실한 지침을 제공해 줌으로써 상당히 논리적이긴 하지만 이론을 중심으로 하고 있어 사용자들에게 컴퓨터 시스템의 감각을 알기위한 사전 테스트를 허용하지 못했고, 또한 여기에 대한 경험을 누적하고 운영하는 과정에서 사용자들의 요구를 만족시켜 주지 못함으로써 사용자들의 기대에 못미쳤다. 이러한 미비점을 보완하여 사용자의 시각에서 컴퓨터 시스템을 설계할 수 있게 한 방법이 PDLC이다.

따라서 본 연구의 목적은 사용자 자신의 요구 사항 도출과 신속하고 저렴한 비용으로 보고서를 산출하려는 Rapid prototyping 개발 방법을 응용하여 효과적인 사용자 요구사항 파악, 요구변화에 신속한 대처, 그리고 요구사항의 조기 검증을 가능하게 하는 효율적인 소프트웨어 개발 방법론을 제시하고자 하는 데 있다.

제 2 장 소프트웨어 개발 수명 주기

1. SDLC

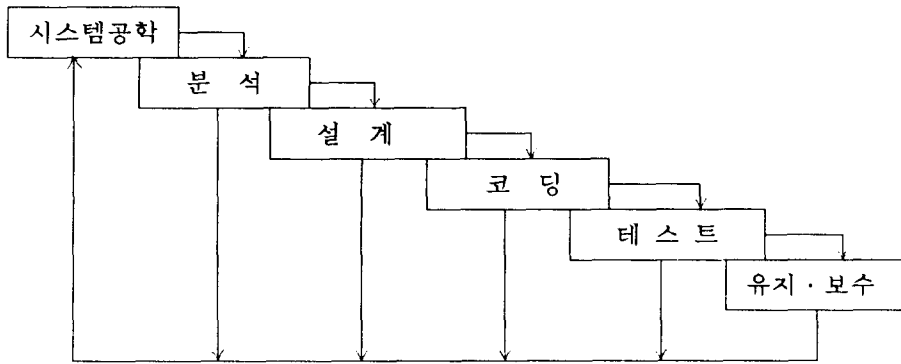
SDLC는 아래 <그림 1>에서 처럼 Waterfall model을 사용하여 설명할 수 있다. 이것은 요구 사항의 분석, 설계, 코딩, 시험단계를 거쳐 유지·보수 단계로 수명 주기를 이루고 있다. 기본 모델은 각 단계마다 시험 과정을 거치고 검증 결과를 현 단계에서 선행 단계로 반영시키는 Feedback으로 이루어져 있다.

이 모델은 70년대에 제시된 것으로 가장 오래 되었으며, 소프트웨어를 개발하는 조직은 조직의 개발 환경에 맞도록 기본 모델을 변경하여 사용하기 때문에 아직도 소프트웨어 공학용으로 널리 사용되고 있다. 그러나 사용자의 요구와 관련된 환경적 특성이 보편화됨에 따라 다음과 같은 많은 문제점들이 지적되고 있다. [Harrison, 1985]

- (1) 실제의 프로젝트는 모델이 제안한 순서의 흐름을 지키기 힘들며, 반복해야 할 상황이 항상 발생하고 그 적용에 문제점이 따른다.
- (2) 초기 단계에서 고객이 모든 요구 사항을 언급하기가 어렵다.
- (3) 프로그램의 작성은 프로젝트가 완료 되어야만 사용자가 이용할 수 있으며 작업 프로그램이 재 검토될 때까지 검출되지 않은 주요 실책은 소프트웨어 시스템의 큰 재해가 될 수 있다.
- (4) 사용자들이 개발된 시스템을 검토하여 받아들이기 이전에는 시스템에 대해 경험해 보는 것이 불가능하다.

- (5) 사용자들이 시스템을 사용케 하면서 얻은 경험을 바탕으로 하여 시스템의 요구 사항을 재 정의하는 것이 불가능하다.
- (6) 사용자가 개발하고자 하는 시스템의 최종적인, 모든 요구 사항을 다 표현하기는 극히 어렵다.
- (7) 그래프나 작성된 문서에 의해 개발자와 사용자간에 정보를 전달하는 것이 최적의 방법이라 할 수 없으며 경우에 따라서는 정보가 잘못 전달될 수 있다.

<그림 1 > SDLC 개발 주기



2. PDLC

Weiser는 시스템에서의 프로토타입을 “시스템의 기능을 정확하고 빨리 체크하는데 효과적인 시스템의 모델이며, 프로토타입은 다른 방법으로는 정확하고 완벽하게 표현할 수 없는 것을 표현할 수 있는 모델의 일종”[Weiser, 1982]이라고 하였다. 이와 같은 프로토타이핑은 신속한 모델(Rapid throw-away models)이면서 주기적 반복적 모델(Cyclic iterative models)이라는 속성을 갖고 있다. 프로토타이핑에 대한 구체적인 내용을 아래에 고찰해 두었다.

1) 프로토타이핑의 특성

프로토타이핑은 개발자가 소프트웨어를 구축하기 위한 모델을 생성하는 과정이며, 이경우 고려되는 모델에는 다음과 같은 유형이 있다.

- (1) 사용자가 인간과 기계와의 상호 작용(Man-machine interaction)을 이해하기 쉽게 하는 Paper 프로토타이핑.
- (2) 소프트웨어의 요구 기능의 부분 집합으로 구현된 Working 프로토타이핑
- (3) 새로운 개발 노력을 감소시키는 현존 프로그램 프로토타이핑 등이 있다. [Pressman, 1987]

프로토타이핑은 요구자 자신의 의도를 잘 반영시키기 위한 방법으로써 빠른 접근과

주기적, 반복적 속성을 실현하기 위해서는 필요한 자원과 프로토타이핑을 적용하기에 적절한 상황 등이 필요하다.

(1) 프로토타이핑에 필요한 자원 [Naumann & Jenkins, 1982]

프로토타이핑을 구현하기 위해 필요한 자원들은 아래와 같다.

(1) 대화식 시스템(Interactive systems)

대화식 시스템은 컴퓨터 자원의 능력을 확장시켜 사용자가 필요로 할 때 즉시 자원을 사용할 수 있는 환경을 제공함으로써 처리 시간이 지연되는 것을 방지할 수 있고 사용자의 시스템 제어 능력을 증대시킨다.

(2) 데이터베이스 관리 시스템(Data Base Management Systems ; DBMS)

대화식 시스템에서는 DBMS의 질의어에 바탕을 둔 여러가지 자연어들이 자료를 관리하는 보완적인 기능을 수행할 수 있다. DBMS의 지원에 의하여 자료의 생성과 수정을 신속하고 용이하게 할 수 있고 자료를 입력하고 추가 변경하는 절차와 보고서를 작성하는 기능을 수행할 수 있게 된다.

(3) 범용 입·출력 소프트웨어

보고서를 작성해서 출력시키는 소프트웨어와 다양한 기능을 갖춘 질의어들이 DBMS와 연결되어 사용되고 있다.

(4) 초 고급 언어(Very high level language)

프로토타입을 개발할 때는 DBMS를 관리하는 자원들과 연결되어 사용되어 질 수 있고, 보고서 작성과 출력 처리 등을 신속하고 용이하게 해결할 수 있는 언어들이 유용하게 사용되어질 수 있다.

(5) 의사 결정 모형

인간의 의사 결정을 지원해 주는 모델들은 많은 시스템에서 필요 불가결하다. 유용한 모델을 모아 놓은 모델 뱅크는 프로토타이핑의 중요한 자원 중 하나이다.

(2) 프로토타이핑에 적합한 상황 [Burns & Dennis, 1985]

프로토타이핑은 불확실성의 정도가 큰 프로젝트일 경우 불확실한 사용자들의 요구사항을 다 단계에 걸쳐 정의해 내는데 적합하다고 할 수 있다. 패키지의 규모가 크고 동시에 불확실성의 정도가 높을 경우에는 SDLC에 프로토타이핑을 결합시킨 개발방법이 적합하다. 여기에서는 프로토타이핑을 시스템개발의 첫 단계로 도입하고 있다. 프로토타입을 개발하여 사용자들에게 인도하고 사용자들은 프로토타입을 사용해 보고 자신들의 진정한 필요가 무엇인지를 시스템 개발자들에게 명시한다. 사용자들의 요구사항을 Feedback하여 시스템을 개선하고 사용자들이 만족하면 이 프로토타입을 사용자들의 요구사항을 정의하는 수단으로 사용하는 것이다. 프로젝트의 불확실성이 이러한 프로토타입 단계에서 많이 감소된다.

이상의 결과를 요약하면 <표 1>과 같다.

<표 1> 프로토타이핑에 적합한 상황

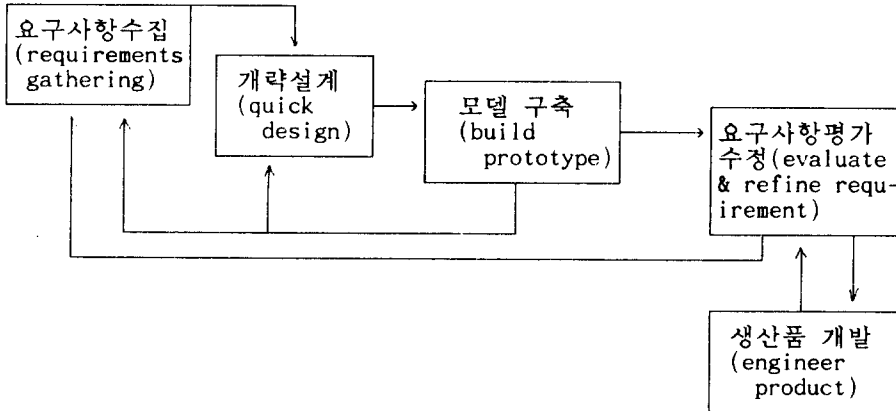
프로젝트의 복잡성	고	S D L C	Mixed Methodology
	저	Prototyping	Prototyping

저 고
프로젝트의 불확실성

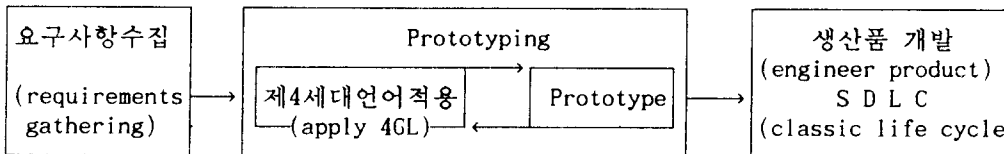
(3) 프로토타이핑의 잠정적인 개발 주기

PDLC는 <그림 2>에 요약하여 두었고, 새롭고 강력한 도구의 출현은 <그림 3>처럼 새로운 조합형 개발 주기를 가능케 한다.

<그림 2> P D L C



<그림 3> 조합형 개발 주기



<표 2>는 잠정적인 프로토타이핑 개발 단계별 변화를 보여주는 요약표이다. 단계가 변화함에 따라 기능과 목표들이 변한다. 사용하는 도구도 다양하며 도구와 목표가 변화 함에 따라 시스템에 관한 관점도 변화되어야 한다.

<표 2> 잠정적인 프로토타이핑 개발 단계 [Zahniser, 1983]

추상의 기준	기준에 대한 설명	수명주기별단계	단계의 목표
0 해설 (nirvana: 열반)	가상 사용자의 마음 속에 존재, 다른 방법으로 정의될 수 없음	유지 및 평가	문제인식과 정의
1 기능	기능적 관점에서 사용자에 의해 정의됨	정 의	사용자, 개발자 간의 의사소통
2 자료	자료의 흐름처럼 자료의 관점에서 정의되거나 변화, 저장됨	논리적 설계	설계의 무한성
3 상자	기능과 변화가 존재하는 프로그램과 하드웨어 상자 관점에서 정의됨	물리적 설계	경제적인 설계의 적용
4 통제	앞에서 정의된 개별기능 / module을 조화시키는 실제적인 통제 논리의 관점에서 정의됨, 실제 code가 정의	구현적 (constructive)	적용의 수정
5 구축	코드화된 모듈은 사용자들에게 보여줄 시스템의 작업 version내에서 통합된다.	구 현 (construction)	시스템 운과을 통제
6 문제 해결	시스템은 사용자 환경에서 정의되고 사용자들의 문제 집합체를 해결해 준다.	운 용	문제 해결을 위해 사용가능한 시스템

2) 프로토타이핑 개발 방법

(1) 프로토타이핑 개발 접근 방법

프로토타이핑의 사용은 사용자들이 프로토타이핑을 이용하여 소프트웨어를 이해하도록 하는 방법이다. 프로토타이핑은 사용자들을 위하여 수행하는 기능이 무엇이며, 어떻게 수행하는 가를 보여주기 때문에 사용자들의 요구사항을 정의하는 데 많은 도움을 준다. 또한 개발자에게는 예상되는 개발 위험 부담을 예측할 수 있는 기회로도 이용할 수 있다. [박진영, 1989]

이와 같은 목적을 달성하기 위해 프로토타이핑 개발 접근에 있어서도 체계적이고 단계적인 방법이 요구되는데, 아래<표 3>은 프로토타이핑 개발 접근방법에 대한 여러 학자들의 견해를 요약하여 보여주고 있다.

<표 3> 프로토타이핑 개발 접근 방법에 대한 비교

학 자	단 계	세 부 내 용
Naumann & Jenkins[1982]	4단계	①식별단계 ②개발단계 ③시행 및 사용단계 ④수정 및 보완 단계
Edelman[1981]	4단계	①사용자는 일반 명세서 준비 ②설계자는 프로토타이핑 개발 ③사용자는 테스트 ④설계자는 상세 명세를 위해 변경
Boar[1986]	6단계	①기본 요구 사항 식별 ②기초 모델 개발 ③모델 내용의 시범 ④필요성 및 확장된 요구의 분석 ⑤사건 명세서의 세부사항 등에 대한 준비 ⑥수정 보완된 사항의 시행
Harrison[1985]	3단계	①On-line 화면의 구성 ②데이터 베이스 환경 구축 ③시스템의 모형 활용
Pressman[1987]	6단계	①프로토타이핑 개발 여부 평가 ②요구사항을 요약 ③설계명세서 작성 ④테스팅 ⑤의뢰자가 운영 ⑥위의 ①~⑤ 단계를 반복

(2) 프로토타이핑 개발에 활용되는 언어

지난 30년동안 프로그래밍 언어의 발전은 매우 느렸다. 즉 FORTRAN, COBOL, PL/1, PASCAL 등의 언어 구조와 개념을 기본으로 하여 구분만 다르게 설계된 셈이다. 그러나 최근 자동처리의 생산성을 극적으로 증가시켜 주는 새로운 언어와 도구들이 등장하기 시작하게 되어 프로토타이핑의 소프트웨어 환경을 마련할 수 있게 되었다. 이러한 새로운 도구들은 '제4세대 언어', '고 생산성 언어', '비절차적 언어', '응용 개발 생성기'로써 컴퓨터 사용 및 소프트웨어 공학에 있어 귀로의 계기를 제공하게 된 것이다.

프로토타이핑을 이용한 개발에는 주로 제4세대 언어가 이용되는데, <표 4>와 <표 5>는 제4세대 언어의 종류 및 사용 영역을 보여주고 있다.

<표 4> 국내 사용 제4세대 언어 종류

응용 개발 생성기		의사결정 지원 시스템	질의어 및 보고서 작성기	프로그램 자동생성기
사용자용	자료처리			
RAMIS-2 MAPPER SAS UFO FOCUS MUMPS CLIPPER	MANTIS IDEL MARK V CPS FOCUS MUMPS CLIPPER	EMPIRE FOCUS MUMPS CLIPPER	ANSWER/DB ADRS-2 RPG-Ⅲ QMF SQL/DS FOCUS MUMPS CLIPPER	FORCE ASG FOCUS MUMPS CLIPPER

<표 5> 제4세대 언어의 영역

프로그램 개발언어	응용 개발 지원 시스템	PC용 프로그램 개발 도구
JASPOL JASMAL Q STAMPS-2 UFO NATURAL DYL-260/280 QUIKOB	PRO-V RAMIS-2 SAS ADS/ONLINE SUPAR JUSPREC ACCELL MODEL 204 UL/204 MARK V FOCUS	PC/SAS MODULE-2/86 UNIHOPE YAC-2 BASIC/I PROLESS HCP PADET/C. F PC-JASPOL

한편, Taylor와 Standish는 프로토타이핑 개발 기법에 사용되는 언어의 필요 조건을 아래와 같이 제시하였다. [Taylor & Standish, 1982]

- (1) 강력한 확장성
- (2) Program text in data
- (3) 번역기가 있고, 기계와 사용자간의 상호작용의 용이성
- (4) 프로그래밍 언어 특유의 자료구조는 처리속도, 성능을 만족시켜야 한다.
- (5) 자체 교환 호출 및 값 교환 호출이 가능
- (6) 명시적 표기법의 배제
- (7) 개념 축소
- (8) 범위의 제한 배제
- (9) 프로그램 작성시 오류를 진단하여 해결책을 제시할 수 있어야 한다.
- (10) 사용법은 간단하고 기능이 다양
- (11) 자료확장의 특성
- (12) 값이 저장되어 있는 주소를 수식으로 연산 가능

- (13) 확장된 제어구조 가능
- (14) 다양한 사용자 서비스
- (15) 문제의 유형에 대한 보다 유력한 개념 설정
- (16) 인공지능 변형 기법
- (17) 프로그램 변환이 용이

3) 구조적 개발 기법과 프로토타이핑 개발 기법간의 비교분석

구조적 개발기법과 프로토타이핑 개발기법과를 비교하여 아래 <표 6>에 요약해 두었다.

<표 6> 구조적 개발기법과 프로토타이핑 개발기법과의 비교 분석

항목 / 구분	구조적 개발기법	프로토타이핑
대상업무	대규모 정형업무 위주	사용자 많고 비정형업무위주
사용자 참여도	대부분 요구분석 단계에 편중	전 단계에 참여
방법론	단계적 선형 개발	신속한 구현, 반복, 수정개발
개발환경	비교적 독립	비교적 종속
의사전달	다양화	표준화

아울러 프로토타이핑의 주요 장, 단점을 파악하여 <표 7>에 정리하여 두었다.

<표 7> 프로토타이핑 개발기법의 장, 단점 [Carpenter, 1982 ; Taylor & Standish, 1982]

구분	내용
장점	사용자의 참여로 정확한 요구사항의 제시 및 반영, 초기에 오류 수정 →비용의 효과 개발 지원자가 개발 진척도를 파악가능 →신뢰성 증대 개발 주기에 사용자 항상 참여 → 품질 향상, 위험 배제 도구의 사용과 의사소통의 표준화 → 개발속도 향상
단점	사용자의 자질과 참여도의 의문 성능상의 문제점 자원의 환경 관리기법 및 능력

제 3장 실험

1. 연구의 개요

프로토타이핑 개발 방법론이 아직까지는 명확히 정착되어 있지는 못하며 프로토타이핑의 용어 조차 상황에 따라 다르게 정의되고 있고 응용 범위도 사람과 적용업무에 따라 변하고 있는 상황에서 본 연구에서는 실험을 통하여 효율적인 소프트웨어 개발 방법론을 도출하고자 한다.

1) 대상업무 : 시한성 교환 품목 관리제도(TCI : Time change item management system)

2) 구현환경

(1) HARDWARE

가. CPU : PRIME P6350-GSC

나. Main memory : 64MB

다. 처리속도 : 11.8MIS

라. Disk : Model 4835 (9,600MB)

마. Tape device : Model 4592 * 2

바. Line printer : Model BP-1500 * 2 (1500LPM BAND 방법)

사. Terminal : LYNX-3300

아. Dot printer : EXPRESS-4500

(2) SOFTWARE

가. Operating System : PRIMOSTTM (22.1.0)

나. S/W : INFORMATIONTM

INFORMATION(INFO)은 PRIME사에서 제공하는 Package로서 고도로 개발된 Subsystem을 이용하여 경영 자료를 쉽게 처리할 수 있게 한 RDBMS (Relational data base management system)이다.

Subsystem은 INFOR/DMS, INFORM, ENTRO, PREFORM 그리고 경영 자료 처리용 Language 인 INFO/BASIC으로 되어 있다.

특징으로는 File dictionary를 이용하여 자료의 논리적 정의와 물리적 자료 구조를 분리한 것과 가변적 크기의 Record를 이용할 수 있는 신축성 있는 Database 구조 등을 들 수 있다.

3) 개발 방법론

프로토타이핑 기법 을 분석 단계 및 테스트 단계에 적용하였다.

- (1) 업무개요 및 현업의 양식을 통한 자료 수집
- (2) INFO의 기능인 INFORM을 통한 온라인 화면 모형
- (3) PERFORM을 이용한 Input과 Output의 정의
- (4) INFORM의 기능을 이용하여 초기 동작 모델 구축

2. 연구의 대상

항공기에 장착되어 작동되는 모든 부분품중 주요 부품을 선별하여 각각의 수명주기를 선정 기록 유지하고 적시에 교환할 수 있도록 하여 치명적인 비행 사고 예방과 원만한 항공 정비지원 체계를 유지하며 항공기 호수별로 시한성 교환 품목의 잔여시간 및 예상 교환시기 등을 수시 및 주기적으로 지원하는 시스템을 대상으로 하였다.

3. 현 시스템의 문제점

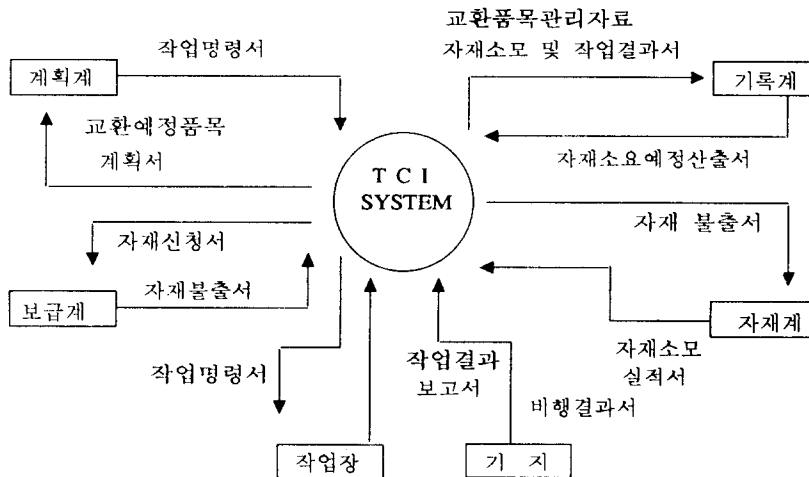
현 시스템의 문제점을 도출해 보면 다음과 같다.

- (1) 품목 관리 자료량 및 자료 산출 주기에 의거 인력낭비
- (2) 정확한 자료 산출 및 적기 지원 곤란
- (3) 자료 관리 및 보관의 어려움 등이 있다.

4. 초기 모델 구현

1) 초기 동작 모델 (VERSION 0)

(1) 시스템 배경도



(2) 자료 분석

가. 자료 사전(Data dictionary)

작업명령서 = (작업명령번호 + 기종일련번호 + 기종 + 작업내용)

교환예정품목계획서 = (부분품번호 + 부품일련번호 + 기종 + 기종일련번호 + 교환예정일자 + 수량 + 주기)

자재신청서 = (부분품번호 + 부품일련번호 + 수량 + 신청일자 + 증빙서번호)

비행결과서 = (항공기기종 + 항공기 호수 + 비행시간 + 소티수 + 사이클 + 일자)

작업결과보고서 = (작업명령번호 + 작업장 + 기종일련번호 + 기종 + 장탈부분품번호 + 장탈부분품일련번호 + 작동시간 + 장착부분품번호 + 장착부분품일련번호 + 작업단순부호 + 대치번호 + 장탈일자 + 장착일자 + 기타)

자재소모실적서 = /*교환예정품목계획서와 동일*/

교환품목관리자료=(부분품번호 + 부품일련번호 + 물품번호 + 기종 + 기종일련번호 + 장착일자 + 장탈일자 + 교환예정일자 + 장착수 + 교환주기 + 상태 + 이전작동주기 + 작업단순부호 + 기타)

자재소모 및 작업결과서 = /*작업결과보고서와 동일*/

나. 정규화 (Normalization)

가) 제3정규형(TNF : Third normal form)의 추출절차

(가) 자료사전 작성 및 키 항목 설정

(나) 제1정규형 제출 : 내부 반복 그룹의 제거

- 내부 반복 그룹을 별개의 자료군으로 분리

- 조합키(Composite key) 및 외래키(Foreign key) 개념 이용

(다) 제2정규형 추출 : 부분적 종속성(Partial dependence)의 제거

(라) 제3정규형 추출 : 이행적 종속성(Transitive dependence)의 제거

(마) 자료군의 통합 : 같은 자료군을 갖는 자료군에 대해서 하나로 통합

나) 제3정규형 추출 결과

1.1 시한성 교환 자료철

= (항공기기종 + 항공기호수 + 부품번호 + 부품일련번호 + 제작일 + 장착일 + 장착시간 + 이전작동일 + 교환예정일 + 기타 + 상태)

1.1.1 항공기호수 = /*항공기 기종별 고유번호*/

1.1.2 제작일 = /*부품일련번호별 제작일*/

1.1.3 장착일 = /*부품별 항공기에 장착한 일자*/

1.1.4 이전작동일 = /*부품별 항공기에 창작성까지의 작동 일자*/

1.1.5 교환예정일 = /*항공기에 장착된 부품별 교환 예정 일자*/

1.1.6 상태 = /*항공기에 장착 혹은 예비부품이나를 구분*/

= [장착 : A ; 예비 : S]

1.2 시한성 부품철

= (부품번호 + 물품번호 + 품명 + 위치 ; 장착수 + 교환주기 + 관련근거 + 적용기종 + 작업단순부호)

1.2.1 물품번호 = /*부품번호와 비슷한 기능이나 자재신청시에는 부품번호에 해당하는 물품번호로 신청하게됨*/ X(12)

1.2.2 위치 = /*항공기에 장착되어 있는 위치를 표시하며 특히 같은 부품이
 들이상 장착되어 있을시는 각각 다른 위치로 표시됨*/ X(6)

1.2.3 장착수 = /*한 항공기에 장착될 수 있는 동일 부품 수 */ 9(3)

1.2.4 교환주기 = /*조작 및 장착, 사용후 교환해야할 일수*/

1.2.5 작업단순부호 = /*부품이 사용되는 계통*/
 = [엔진 ; 전기 ; 유압 ; 통신 ; 항전 ; 조종 ; 등 20여종

1.3 기체 시간철

= (항공기기종 + 항공기호수 + 현기체시간)

1.3.1 현기체시간 = /*항공기 총비행시간*/
 = 시간(9(5)) + 분(9(2))

1.4 월작동 주기철

= (항공기기종 + 월시간)

1.4.1 월시간 = /*기종별 한대당 월 평균 비행시간, 교환 예정일 산출에 관
 여*/

다. 자료 정의 (Data Definition)

가) 시한성 교환 자료철

field name	type	location	conversion	format	S/M	assoc- iation
TC1-KEY	D	0		8L	M	
제작일	D	1		6L	S	
장착일	D	2		6L	S	
장착시간	D	3		8R2	S	
이전작동일	D	4		6R	S	
교환예정일	D	5		6R	S	
기지	D	6		1L	S	
상태	D	7		1L	S	

field name	type	location	conversion	format	S/M	association
장착후주기	I	(교환예정일[1.2] - 장착일[1.2]) * 12 + 교환예정일 [3.2] - 장착일[3.2]		7R###	S	
제작후주기	I	(교환예정일[1.2] - 제작일[1.2]) * 12 + 교환예정일[3.2] - 제작일[3.2]		7R###	S	
잔여시간	I	(교환예정일[1.2] - 현년) * 12 + 교환예정일[3.2] - 현월		7R###	S	
장착이후시간	I	(현년 - 제작일[1.2] * 12 + 현월 - 장착일[3.2])		7R###	S	
현월	I	@MONTH		2L	S	
현년	I	@YEAR		2L	S	
항공기기종	I	EXTRACT(TCI-KEY, 1, 1, 0)		3L	S	
항공기호수	I	EXTRACT(TCI-KEY, 1, 2, 0)		7L	S	
부품번호	I	EXTRACT(TCI-KEY, 1, 3, 0)		15L	S	
부품일련번호	I	EXTRACT(TCI-KEY, 1, 4, 0)		10L	S	
관련근거	I	TRANS(PTCI. FILE, 부품번호, 관련근 거, "X")		22L	S	
물품번호	I	TRANS(PTCI. FILE, 부품번호, 물품 번호, "X")		13L	S	
위치	I	TRANS(PTCI. FILE, 부품번호, 위치, "X")		6L	S	
장착수	I	TRANS(PTCI. FILE, 부품번호, 장착 수, "X")		3R	S	
주기	I	TRANS(PTCI. FILE, 부품번호, 주기, "X")		7R	S	
품명	I	TRANS(PTCI. FILE, 부품번호, 품명, "X")		16L	S	
교환예정년	I	교환예정일[1.2]		2R	S	
교환예정월	I	교환예정일[3.2]		2R	S	
@ENTRO	PH	TCI-KEY 장착일 이전작동일 기지	제작일 장착시간 교환예정일 상태			

나) 시한성 부품철

field name	type	location	conversion	format	S/M	association
부품번호	D	0		15L	S	
물품번호	D	1		12L	S	
품명	D	2		16L	S	
위치	D	3		6L	S	
장착수	D	4		12L	S	
교환주기	D	5		16L	S	
관련근거	D	6		6L	S	
적용기종	D	7		12L	S	
작업단위부호	D	8		16L	S	
@ ENTRO	PH	부품번호 위치 교환주기 적용기종	물품위치 장착수 관련근거 작업단위부호			

다) 기체 시간철

field name	type	location	conversion	format	S/M	association
ACT-KEY	D	0		10L	?	
현기체시간	D	1		8RZ##### :##	S	
항공기기종	I	ACT-KEY[1.3]		3L	S	
항공기호수	I	ACT-KEY[4.7]		7L	S	
@ ENTRO	PH	ACT-KEY	현기체 시간		S	

라) 월작동 주기철

field name	type	location	conversion	format	S/M	association
항공기기종	D	0		3L	S	
월시간	D	1		6RZ	S	
@ ENTRO	PH	항공기기종 월시간				

* TYPE

D-Type : 실제로 Data가 들어갈 Field를 나타낸다.

I-Type : Data field나 다른 File 또는 System에서 제공하는 Information 등을 Define한 Field이다.

PHrase : INFORM command를 사용해야 할 Field를 나타낸다. INFORM verb. 는 나타나지 않는다.

X-type : File상에 Data가 아니고 User가 기억시킨 값이다.

이 값은 INFORM에서는 사용할 수 없고, I-type이나 INFO/BASIC에서만 쓸 수 있다.

* FORMAT

L은 Left-justified

R은 Right-justified

T는 Text-justified를 나타낸다.

* S/M

Singl(S) 또는 Multiple(M) value를 나타낸다.

(3) 입 · 출력 화면 설계 (Input output screen design)

시 한 성 교 환 품 목 관 리

- 1 = TCI품목 등록 및 변경
- 2 = TCI품목 일련번호 와 등록 및 변경
- 3 = 항공기 기체시간 변경
- 4 = 월 작동주기 입력 및 변경
- 5 = 기종별 시한성 교환 품목 현황
- 6 = CAD/PAD품목 _년도 월별 소요량
- 7 = 기종별 TCI품목 현황
- 8 = 기종별 일련번호별 TCI품목 현황
- 9 = 기종별 호수별 TCI품목 _개월 이내 교환예정 현황
- 10 = 품목별 TCI품목 _개월 이내 교환예정 현황
- 11 = 품목별 년간 및 월별 교환 예정 현황

Which would you like ? (1 - 11) =

가. 입력 설계

가) TCI품목 등록 및 변경

```

TCI.FILE-Screen  1-FIRST SCREEN      16:21:26 04 APR 1990
1 TCI-KEY         A_5
2 장착일자       880423
3 장착시간       324
4 이전작동       547
5 교환예정       901223
6 기지           A
7 상태           B
8 제작일자       850708
SI = FIRST SCREEN

```

S2 = TCI-KEY

나) TCI품목 - 연 번호별 등록 및 변경

PTCI.FILE-Screen 1-16:23:41 04 APR 1990

1 부분품 번호 11111
2 물품번호 111111111111
3 품명 POMP
4 위치 L/H
5 장착수 002
6 주기 0000048
7 관련근거 TO A-1-1
8 적용 기종 A-1
9 작업 단위부 2

CHANGE=

다) 항공기 기체 시간 변경

ACT.FILE-Screen 1-16:24:03 04 APR 1990

1 ACT-KEY 1111
2 현 기체 시간 234

CHANGE =

라) 월 작동 주기 입력 및 변경

AET.FILE-Screen 1-16:24:30 04 APR 1990

1 항공기종 A_5
2 월 시간 432

CHANGE =

나. 출력 설계

가) 기종별제한성 교환 품목 현황

LIST TCI.FILE 항공기종 품명 부분품번호 장착수 주기 위치 관련근거 WIGH항

공기종 = A_1 16:03:09 04-04-90 PAGE 1

TCI.FILE ... A_1
. 43216
. 11111
. 65863

항공기종 ... A_1
품명 PUMP
부분품번호 . 11111
장착수 002
주기 0000048
위치 L/H
관련근거 ... TO A-11

TCI.FILE ... A_1
. 39406
. 11111
. 32876

항공기종 ... A_1
품명 PUMP
부분품번호 . 11111
장착수 002
주기 0000048
위치 L/H

관련근거 ... TO A-1-1
2 record listed.

나) CAD/PAD품목 _년도 월별 소요량

해당 년도 _ = 92
LIST TCI.FILE 부분품번호 물품번호 품명 제작후주기 장착후주기 항공기종
BREAK.ON 교환예정일 WITH교환예정년 = 92 16:04:26 04-04-90 PAGE 1
TCI.FILE A_1
 . 39406
 . 11111
 . 32876
부분품번호 ... 11111
물품번호 111111111111
품명 PUMP
제작후주기 ... 130개월
장착후주기 ... 78개월
항공기종 A_1
교환 예정일 .. 03
TCI.FILE A_2
 . 76592
 . 22222
 . 35390
부분품번호 ... 22222
물품번호 222222222222
품명 VALVE
제작후주기 ... 152개월
장착후주기 ... 66개월
항공기종 A_2
교환예정일 ... 11
2record listed.

다) 기종별 TCI품목 현황

항공기종 = A_1

부분품번호 = 11111

LIST TCI.FILE 항공기종 부분품번호 품명 호수 부품일련번호 위치 주기 이전
작동 장착시간 장착이후시간 잔여시간 교환예정 WITH 항공기종 = A_1 WITH 부
분품번호 11111 16:05:13 04-04-90 PAGE 1

TCI.FILE A_1
 . 43216
 . 11111
 . 65863
항공기종 A_1
부분품번호 ... 11111
품명 PUMP
호수 43216
부품일련번호 . 65863
위치 L/H
주기 0000048

이전작동 643
 장착시간 2395.00
 장착이후시간 . 57개월
 잔여시간 13개월
 교환예정 910514
 TCI.FILE A_1
 . 39406
 . 11111
 . 32876
 항공기종 A_1
 부분품번호 ... 11111
 품명 PUMP
 호수 39406
 부품일련번호 . 32876
 위치 L/H
 주기 0000048
 이전작동 3214
 장착시간 12345.00
 장착후시간 ... 55개월
 잔여시간 23개월
 교환예정 920324
 2 record listed

라) 기종별 - 연번호별 TCI품목 현황

항공기종 = A_1

항공기 호수 = 43216

LISTED TCI.FILE 항공기종 호수 품명 물품번호 주기 이전작동 장착시간 장착이후
 시간 잔여시간 교환예정 WITH 항공기종 = A_1 WHTH 호수 = 43216 16:06:34

04-04-90 PAGE 1

TCI.FILE A_1
 . 43216
 . 11111
 . 65863

항공기종 A_1
 호수 43216
 품명 PUMP
 물품번호 111111111111
 주기 0000048
 이전작동 643
 장착시간 2395.00
 장착이후시간 . 57개월
 잔여시간 13개월
 교환예정 910514
 1 record listed

마) 기종별 호수별 TCI품목 _ 개월이내 교환 예정 현황

_개월 이내 교환 = 32

LIST TCI. FILE 항공기종 호수 품명 부품일련번호 주기 장착시간 잔여시간
교환예정 WITH 잔여시간 <= 32 16:11:25 04-04-90 PAGE 1

TCI. FILE A_1
 . 43216
 . 11111
 . 65863
항공기종 A_1
호수 43216
품명 PUMP
부품일련번호 . 65863
주기 0000048
장착시간 2395.00
잔여시간 13개월
교환예정 910514

TCI. FILE A_1
 . 39406
 . 11111
 . 32876
항공기종 A_1
호수 39046
품명 PUMP
부품일련번호 . 32876
주기 0000048
장착시간 12345.00
교환예정 920324
2 record listed

바) 품목별 TCI품목 _개월이내 교환예정 현황
_개월이내 교환 = 28

LIST TCI. FILE 품명 부품번호 부품일련번호 항공기종 호수 주기 장착시간 잔여시
간 교환예정 WITH 잔여시간 <= 28 16:14:19 04-04-90 PAGE 1

TCI. FILE A_1
 . 43216
 . 11111
 . 65863
품명 PUMP
부분품번호 ... 11111
부품일련번호 . 65863
항공기종 A_1
호수 43216
주기 0000048
장착시간 2395.00
잔여시간 13개월
교환예정 910514

TCI. FILE A_1
 . 39406

. 11111
 . 32876
 품명..... PUMP
 부분품번호 ... 11111
 부품일련번호 . 32876
 항공기종 A_1
 호수 39046
 주기 0000048
 장착시간 12345.00
 잔여시간 23개월
 교환예정 920324

TCI.FILE AA
 . 11111
 . 11111
 . 11111
 품명..... PUMP
 부분품번호 ... 11111
 부품일련번호 . 11111
 항공기종 AA
 호수 11111
 주기 0000048
 장착시간 AA
 잔여시간 84개월
 교환예정 AA
 3 record listed

2) 최종 동작 모델(Prototyping output)

(1) 자료 정의 (Data definition)

시한성 교환철 = (항공기기종 + 항공기호수 + 부품번호 + 부품일련번호 + 제작일자
 + 장착일 + 장착시간 + 구분 이전 작동현황 + 교환예정 + 기타 + 상태)

* 구분 = [시간 | 월수 | 소티 | 사이클]

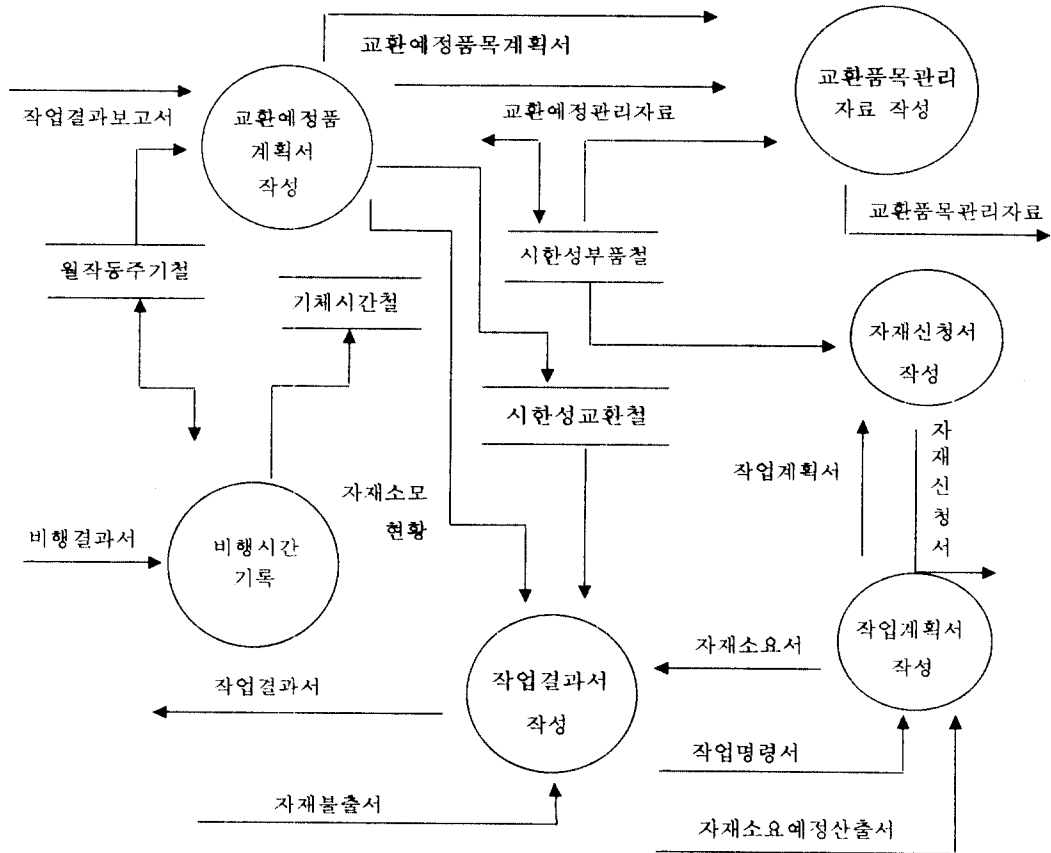
/* 이전작동시간과 교환예정일자에 대한 구분*/

시한성 부품철 = (부품번호 + 물품번호 + 품명 + 장착위치 + 장착수 + 구분 + 교환
 주기 + 관련근거 + 적용기종 + 작업단위부호)

기체 시간철 = (항공기기종 + 항공기호수 + 기체총시간 + 기체총소티 + 검사주
 기)

기종 부호철 = (항공기기종 + 월평균가동시간 + 월평균소티 + 월사이클 + 격렬도)

(2) 동작모델 자료 흐름도 (Data flow diagram)

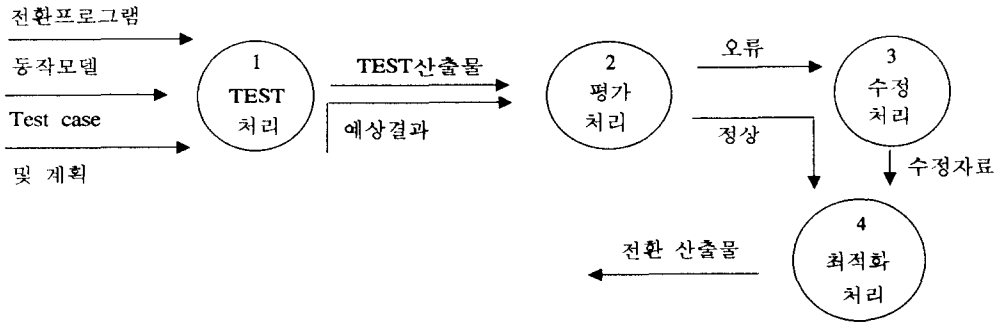


3) 테스트 모델 구현

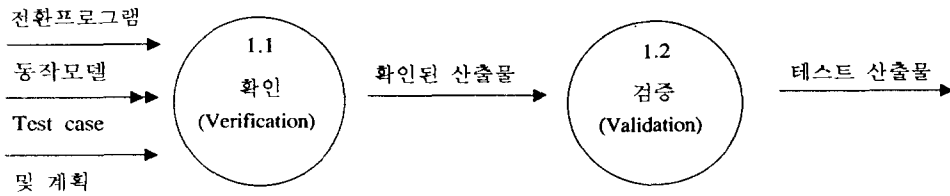
동작모델로 요구사항이 정확히 파악되었다고 해도 설계시 자료구조의 변경과 모듈의 통합 및 전환에 대한 또다른 오류와 요구사항의 변화에 대응하기 위하여 테스트 단계는 수행되어야 하며 이때의 테스트 단계는 구조적 개발 방법론과 비슷한 절차를 거치게 된다. 그러나 동작 모델의 이용 및 응용으로 테스트를 효과적으로 할 수 있다.

전반적인 테스트의 절차는 <그림 4>로 나타내었으며 <그림 5>는 테스트 기능을 세분하여 구현 명세서와 구현 시스템과의 일치여부를 테스트하는 확인단계(Verification)와 사용자 요구사항과 컴퓨터 시스템과 일치하는 가를 테스트하는 검증단계로 구분하여 나타내었다.

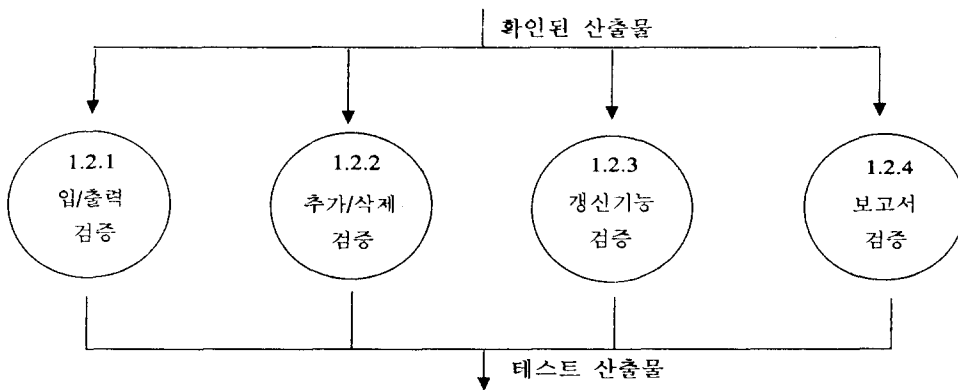
<그림 4> 테스트 흐름도



<그림 5> 세분화된 테스트 처리도



<그림 6> 프로토타입의 주요 기능



또한 검증 단계를 세분화하여 실제 사용자와 프로토타이핑을 하기 위한 프로토타입 기능을 <그림 6>으로 나타내고 이들중 갱신 기능에 대해 오류 수정을 실시하기 위한 갱신 선택 화면 및 처리 기능 재정의 화면에 대해 <표 8>과 <표 9>와 같이 설계하였다.

<표 8> 갱신 선택 화면

```

                갱신 선택(TYPE별)

1 = D TYPE : 입력 자료(Field)재정의
2 = I TYPE : 처리 기능 재정의
3 = PH TYPE : 파라메타 재정의
4 = X TYPE  : 초기값 재정의

                Which would you like ? (1-4) =
    
```

<표 9> 처리 기능 재정의 화면

```

LIST DICT TCI,FILE : I-TYPE 17:21:59 06-05-90 PAGE 1
FIELD NAME.... LOCATION ..... CONV DISPLAY NAME FORMAT SM ASSOC..

장착후주기 (교환예정 [1.2]-          장착후주기 7R### S
              장착일자[1.2])*12
              +교환예정[3.2]-
              장착일자[3.2]
잔여시간 (교환예정[1.2]-          잔여시간 7R### S
           현년)*12+교환예정
           [3.2]
장착이후시간 (현년-장착일자[1.2])      장착이후시간 7R### S
              *12+현월-장착시간[3.2]
품명 TRANS(PTCI,FILE.          품명 16L S
           부분품번호, 품명, "X"
           ..... 20 records listed
    
```

제 4 장 분석 및 결과

1. 프로토타이핑 기법 활용에 대한 분석

프로토타이핑 기법을 분석 단계에서 실제 활용한 결과 다음과 같은 고려요소가 도출되었다.

1) 업무 특성 고려

- (1) 개발 대상 업무의 범위를 명확히 할 수 있는가.
- (2) 복잡도와 크기는 어느 정도이고 자료량과 사용 회수는 빈번한가.
- (3) 시스템 사용자 수는 어느 정도되고 주로 사용하는 부류는 어떤 부류인가.
- (4) 개발 기간과 개발 자원은 가능한가 등이 있다.

2) 개발자 특성 고려

- (1) 개발 의뢰자의 직위와 자질 및 의욕과 진취성은 어느정도인가.
- (2) 대상 업무에 대한 지식과 경험은 어느정도인가 등이다.

3) 도구 특성 고려

- (1) 한글 지원이 가능하며 한글과 영문자와 혼합하여 사용함에 용이한가.
- (2) 기본기능(추가, 제거, 수정) 및 질의형 언어(Query language)기능 제공과 사용자가 사용하기 쉬운가.
- (3) 화일 구조의 변경이 자유로운가 : Data base file 또는 Tool에서 제공하는 화일과 기존화일(SAM, ISAM, DAM)상호간 변환이 자유로워야 다음 단계로의 전환이 용이하 고 타 시스템과 유기적으로 구축할 수 있다.
- (4) 다른 언어와 호환성은 어떤가.
- (5) 데이터 량을 고려한 성능은 어떤가.
- (6) 조합키(Composite key)에 대한 재 정의 기능은 어떤가.
- (7) 적어도 개발자는 도구에 대해 상당 부분까지 알고 있는가 등이었다.

2. 효과 측정

실험결과에 대한 비용대 효과 측정을 인 시수(Man-hour)로 환산하여 분석하였으며 그 결과 아래 <표 10>과 같다.

요약하면, 비용면에서는 수작업비용에서 12,480 만원이 소요되고, 전산 개발시 비용은 6,540 만원, 전산 운영시 운영비는 10,800 만원이 소요된다. 따라서 수작업 대비 전산화 운영시 기대 효과는 1,680 만원이고, 순비용 대 효과의 회수 기간은 약 4년의 기간이 소요된다.

정성적인 평가는 정확히 파악하기 힘들지만 프로토타입 Working model 운영중 소티의 개념에 의한 교환주기 고려, 격렬도에 따른 관리자료 산출등 애매한 기능의 초기 파악이 상당히 효과적으로 개선되었다고 볼 수 있으며 사용자도 실제 작동해 봄으로써 추후 개발참여에 자신감을 갖고 적극적으로 동참할 수 있다는 컴퓨터 마인드를 형성할 수 있었다.

<표 10> 비용 대 효과 비교

1. 수작업 비용

구 분	단위작업시간	년간 인시수	비 고
수시 자료 산정	2	1,800	300 * 2 * 3(1일 3회)
주간 자료 산정	15	750	50 * 15
월간 자료 산정	75	900	12 * 75
분기 자료 산정	75	300	4 * 75
연간 자료 산정	150	150	1 * 150
계		4,000M-H/year	
연 평균 소요 인원 $\approx 13 \div (4000 \div 300)$ 연간 총 비용 = 12,480만원(13 * 80 * 12) * 연간 작업기준 : 300일, 50주 * 평균 임금(수당포함) : 80만원			

2. 전산화 개발비

구 분	총 액	비 고
인 건 비	3,600 만원	5(명) * 80 * 9
장 비 사 용 료	2,940 만원	420,000 * 0.007
계	6,540 만원	

3. 전산화 운영비

구 분	총 액	비 고
연 평균 인건비	3,480 만원	4(명) * 80 * 12
연간 장비 사용료	6,300 만원	42,000 * 0.015
연간 운영 유지비	160 만원	5(일/월) * 12
연간 소 모 품	500 만원	
계	10,800 만원	

4. 순비용/효과 면에서 회수 기간은 4년 미만이다.

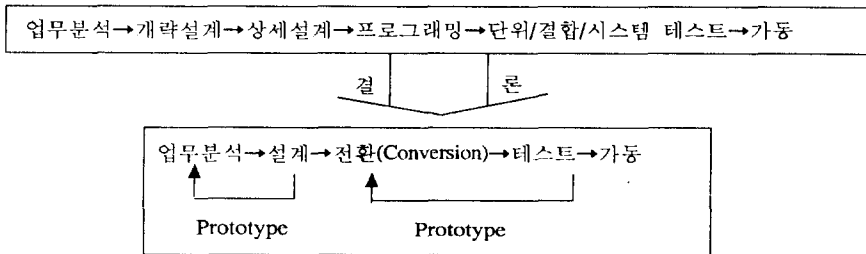
제 5 장 결 론

이상 고찰한 바와같이 항공기 부품중 시한성 교환 품목 관리를 위한 제도를 개발하는 과정에서 분석 업무에 대해 프로토타이핑을 적용하여 보았으며 DBMS를 이용한 프로토타이핑을 적용시켜 보았다. 현재의 개발환경에서는 전반적인 개발주기는 SDLC를 근간으로 하여 분석단계에서는 필수적으로 PDLC를 적용하고 업무 성격과 규모에 따라 PDLC적용이 설계단계까지를 포함할 수 있겠다.

구조적 분석 설계기법을 활용함에 있어서 사용자의 참여는 분석 단계 및 테스트 단계 과정 중 검증은 하기위해 참여하는 반면 프로토타이핑은 사용자 중심의 개발기법으로써 분석 및 설계 단계시 개략의 동작 모델을 사용자에게 보여줌으로써 사용자의 관심을 유발시키고 요구사항을 충분히 반영하게 되어 설계상의 미비점을 보완하게 된다. 결국 시스템이 구체화 될때까지 사용자와의 단절을 지양함으로써 효과적이라 하겠다.

본 연구의 결과로는 <그림 7>과 같이 SDLC상에서 분석 및 개략설계 시점과 테스트 단계에서 프로토타입을 적용하는 것이 효과적일 것으로 사료된다.

<그림 7> 효과적인 Prototype 주기



본 논문에서는 데이터의 구조를 논리적으로 구현하고, 수학적이며, Pseudocode와 같은 형식 명세서(Formal spec)의 활용으로 자연스럽게 동작 모형으로 전환될 수 있는 자동 코드 생성 등의 기능 대신 자료 흐름도, 자료 사전만의 활용으로 프로토타이핑 효과 측정이 제한되었다.

그러나 향후 Computer aided prototyping system 및 자동화된 지원도구의 개발은 프로토타이핑 응용에 있어서 문제점을 해소하게 될 것이며 프로토타이핑 응용 범위와 가치를 높일 것이다.

참 고 문 헌

- 박진영, "정보시스템 개발을 위한 Prototyping 연구", 영진전문대학 논문집, 제11호, 1989.
- Burns, R. N. & Dennis, A. R., "Selecting the Appropriate Application Development Methodology", Data Base, Fall 1985.
- Boar, B., "Application Prototyping : A Life Cycle Perspective", Journal of Systems Management, February 1986.
- Carpenter, R. A., "Designing and Developing Adaptive Information System", Computer Technology Review, Spring-Summer 1982.
- Edelman, F., "Mangers Computers and Productivity", MIS Quarterly, Vol. 5, No. 3, September 1981.
- Harrison, R., "Prototyping and the System Development Life Cycle", Journal of Systems Management, August 1985.
- Harrison, R., op. cit.
- Naumann, J. D. & Jenkins, A. M., " Prototyping : The New Paradigm for Systems Development", MIS Quarterly, September 1982.
- Naumann, J. D. & Jenkins, A. M., op. cit.
- Pressman, R. S., "Software Engineering", McGraw-Hill, 1987.
- Pressman, R. S., op. cit.
- Taylor, T. & Standish, T. A., "Initial Thoughts on Rapid Prototyping Techniques", ACM Sigsoft, Software Engineering Notes, Vol. 7, No. 5, December 1982.
- Taylor, T. & Standish, T. A., op. cit.
- Weiser, M., "Scale Models and Rapid Prototyping", ACM Sigsoft, Software Engineering Notes, Vol. 7, No. 5, Decemle 1982.
- Zahniser, R. A., " Levels of Abstraction in the System Life Cycle", ACM Sigsoft, Software Engineering Notes, Vol. 8, No. 1, January 1983.