

대규모 소프트웨어 개발 사업에서의 품질 보증을 위한 개발방법론

윤석환 · 박지은

한국전자통신연구소

신용백

아주대학교 산업공학과

A Development Methodology for Quality Assurance System in Large-Scale Software Development Project

Seokhwan Yoon · Jieun Park

Electronics and Telecommunications Research Institute

Yongback Shin

Dept. of Industrial Engineering, Ajou University

Abstract

To successfully carry out large scale research projects while supporting quality assurance of research output, effective and systematic management through utilization of resources such as manpower, time and cost as well as engineering techniques such as component technology and design methodology is required.

It is necessary to establish development methodology to support quality assurance. The development methodology covers the contents and procedures of the project, such as division of the project into independently executable units, allocation of resources including researchers, component technology, related know-how and equipment, deployment of research units and integration of the project at the end.

In this paper we present systematic development methodology for quality assurance in large scale software development projects by analyzing the contents of the methodology applied to the Gigabit Information-processing And Networking Technology development project(GIANT).

1. 서론

GIANT 개발사업은 정부에서 추진하는 초고속정보통신망 구축 사업과 초고속정보통신 이용기술 개발사업을 연결하는 맥락에서 생성된 개발사업이다. 즉, 초고속정보통신망이 구축되어 이용기술이 활성화되기 위해서는 초고속정보통신망에 존재하는 정보를 컴퓨터를 통하여 처리할 수 있어야 한다. 이러한 컴퓨터들은 서버-클라이언트의 형태로 망에 접속되어 기존의 정보를 이용하거나, 새로운 서비스를 제공하거나 혹은 새로운 서비스를 생성할 수 있는 소프트웨어 플랫폼의 역할을 하여야 할 것이다. 이러한 초고속정보통신망과 이용기술을 연결하는 초고속정보통신망에서의 소프트웨어 플랫폼을 개발하려는 것이 GIANT 개발사업의 목표이다.

이 개발사업은 3년에 걸쳐, 90명의 연구원과 103억원의 예산이 소요되는 대규모 소프트웨어 개발 프로젝트로서, 이러한 사업을 성공적으로 수행하기 위해서는 많은 인력, 시간, 비용 등의 자원과 요소기술 및 설계 방법론 등의 공학적 기술이 상호 유기적인 결합을 통해 효과적이고 종합적으로 관리되어야 할 것이다. 특히 시스템적 분석을 통한 절차나 방법론이 정립되지 않은 상태에서 대규모 연구 프로젝트를 수행한다는 것은 위험한 발상이며, 이러한 관점에서 볼 때 개발사업의 시작부터 종료시까지의 과정에 품질공학적 접근 방법의 도입은 필수적인 것이며, 선진국의 경우 각자의 고유한 방법을 적용하고 있다.

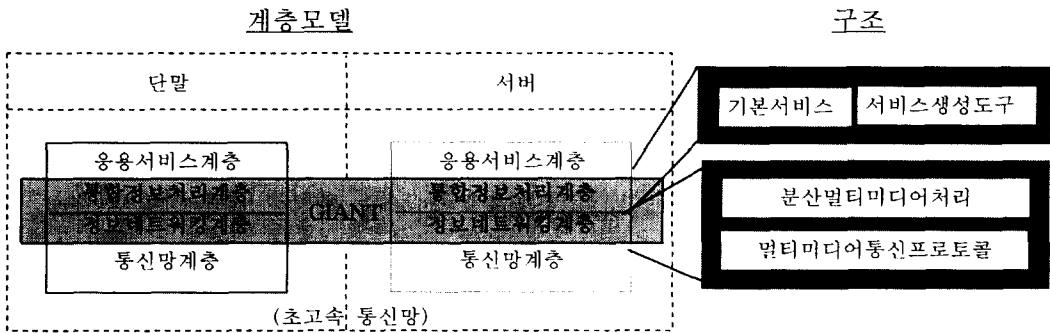
그러나 우리나라의 경우 연구개발의 측면에서 보면, 대규모 연구개발 사업을 수행한 경험이 많지 않기 때문에 이에 대한 품질공학적 기술 축적이 미진하며, 소규모 연구 개발사업의 수행시에도 개발 환경이나 문화가 다른 선진국의 사례만을 적용하였기에 이에 따른 시행 착오와 어려움이 많았다[천유식, 1991]. GIANT 개발사업은 사용자 요구사항이 분명하지 않은 시스템을 개발하는 사업이어서 개발방법론을 사업초기에 정립하는 것이 매우 곤란하였다. 이러한 연구 목표가 분명하지 않은 시스템을 개발하는 경우의 개발방법론을 제안하려는 것이 본 논문의 목적이다.

이를 위하여 본 논문에서는 GIANT 개발 사업을 수행하며 습득한 경험과 공학적 기술을 토대로 연구개발 사업에서의 개발방법론에 대한 품질공학적 요소를 분석하여 미비점을 도출한 후 이를 보완한 형태의 대규모 소프트웨어 사업을 위한 개발방법론을 제시한다. 이를 위하여 1장의 서론을 거쳐, 2장에서는 GIANT의 개념 및 GIANT 개발사업의 특징을 서술하였으며, 3장에서는 GIANT 개발사업의 개발방법론을 서술하였다. 4장에서는 GIANT 개발방법론을 품질공학적 측면에서 분석하였으며, 5장에서는 GIANT 개발방법론의 보완을 위한 형상관리 체계를 서술하였으며, 6장에서는 결론과 향후 연구과제를 제시한다.

2. GIANT 개발 사업

2.1 GIANT 설계 개념 및 규격

GIANT 개발사업은 초고속정보통신망 환경에서 많은 활용이 예상되는 기본적인 정보통신 서비스를 제공하고, 누구라도 새로운 정보와 서비스를 쉽게 만들어서 이용할 수 있도록 도와주는 초고속정보통신 시스템 소프트웨어를 개발하는 것이 최종 목표로서, 초고속정보통신망 환경에서 GIANT의 위치는 사용자가 이용하게 되는 응용 서비스 계층과 다양한 망환경을 제공하는 통신망 계층 사이에 존재하며, GIANT 내부는 통합정보처리 계층과 정보네트워킹 계층으로 구성된다. 통합정보처리 계층은 응용 서비스 개발에 필요한 공통적인 서비스 및 기능 요소들로 구성하였으며, 정보네트워킹 계층은 분산된 객체를 탐색하고 멀티미디어 스트림의 분배 및 동기처리 등을 수행하며, 이질적인 네트워크 상에서의 멀티미디어 데이터를 정확하게 송수신하는 기능으로 구성하였다. 이들의 관계를 <그림 1>로 나타내었다.



< 그림1 > GIANT의 계층 모델 및 구조

2.1.1 기본서비스 블록

기본 서비스 블록은 특정한 응용 분야에 국한되지 않고 공통적으로 이용되거나, 최종 사용자에게 기본적인 정보통신 서비스를 제공하는 블록으로서 세부 기능 규격은 다음과 같다.

- o 원거리에 위치한 다수의 사용자가 동시에 음성, 영상, 그래픽 등을 이용하여 회의 를 하는 기능
- o 원거리에 위치한 다수의 사용자가 멀티미디어 문서를 공동으로 편집하는 기능
- o 임의의 사용자가 원하는 상대방에게 멀티미디어 메일을 전송하는 기능
- o 멀티미디어 정보를 구축 또는 탐색하는 기능
- o 임의의 사용자가 1명 또는 다수의 사용자 그룹에게 원격으로 발표하는 기능
- o 통신망 환경에서 서비스를 탐색하고 사용자 환경에 적합한 서비스를 제공하는 기능

2.1.2 서비스 생성도구 블럭

사용자가 보유하고 있는 정보 혹은 새로운 아이디어를 서비스화 할 수 있는 환경을 제공하는 블럭으로서 세부 기능 규격은 다음과 같다.

- o 서비스 생성을 위해 필요한 환경 설정 및 서비스 창출 도구 기능
- o 서비스 창출 도구로는 불충분한 처리를 직접 명시할 수 있는 프로그래밍 기능
- o 서비스 라이브러리 기능

2.1.3 분산 멀티미디어 처리 블럭

분산 환경에서 통신망을 의식하지 않고 응용간 세션의 설정이나 정보 객체의 탐색, 멀티미디어 스트림을 상호 교환할 수 있는 환경을 지원하는 블럭으로서 세부 기능 규격은 다음과 같다.

- o 응용간 세션의 설정 및 공동작업 환경을 관리하는 기능
- o 멀티미디어 정보 객체에 대한 위치 등 속성을 조회하는 기능
- o 분산 환경에서 멀티미디어 서비스를 제공하기 위하여 최적의 자원 및 환경을 이용하도록 지원하는 기능
- o 멀티미디어 스트림의 분산화, 동기화, 합성 및 분배를 처리하는 기능

2.1.4 멀티미디어 통신 프로토콜 블럭

통신망을 이용하는 응용 프로그램이 요구하는 서비스 품질이나 통신망 성능, 종단 시스템의 상황을 고려하여 최적의 멀티미디어 정보를 상호 교환해 주는 블럭으로서 세부 기능 규격은 다음과 같다.

- o 정보 전달을 위한 연결 설정 및 해제 기능
- o 흐름 및 오류 제어 기능
- o 다자간 통신 기능
- o 성능, 지연, 지연전이 등 QoS(Quality of Service) 통제 기능
- o 시스템 주소 처리 기능

2.2 GIANT 개발 사업의 특징

GIANT 개발사업은 정부에서 추진하는 초고속정보통신망 구축사업과 초고속정보통신 이용기술을 연결하는 맥락에서 생성된 개발사업이다. 초고속정보통신망이 구축되어 초고속정보통신 이용기술이 활성화 되기 위해서는 초고속정보통신망에 존재하는 정보를 컴퓨터를 통하여 처리할 수 있어야 한다. 이러한 컴퓨터들은 서버-클라이언트의 형태로 망에 접속되어 기존의 정보를 이용하거나, 새로운 서비스를 제공하거나 혹은 새로운 서비스를 생성할 수 있는 소프트웨어 플랫폼의 역할을 하여야 할 것이다. 이러한 초고속정보통신망과 이용기술을 연결하는 초고속정보통신망에서의 소프트웨어 플랫폼을 개발하려는 것이 GIANT 개발사업의 목표이다,

이러한 목표를 갖는 GIANT 개발사업의 특징으로는 다음의 세가지를 들 수 있다. 첫째, 정부주도형 공익사업으로서 미국의 HPCC(high performance computing and

communication), 일본의 정보고속도로(information super highway), 유럽 공동체(EC)의 RACE(R&D in advanced communication in Europe), 싱가포르의 IT 2000 (information technology 2000)등의 정보 인프라(information infrastructure) 구축 사업에 대응하는 사회 간접 자본의 성격을 갖는다. 둘째, 개발해야 할 연구 목표를 구체화하기가 곤란하다는 점이다. 대부분의 연구개발 사업은 요구사항 정의 단계 - 설계단계(시스템, 서브 시스템, 블록) - 구현단계 - 통합 단계- 시스템(통합)시험 단계 - 개발완료 단계 등의 과정을 거쳐 수행되며, 특히 요구사항 정의 단계에서는 사용자의 요구사항 또는 불특정 다수의 사용자를 가정한 요구사항을 기술적 언어로 정의한다. 본 GIANT 개발사업은 불특정 다수의 사용자가 서비스를 이용하거나 생성할 수 있는 소프트웨어 플랫폼을 개발하는 것으로서 기술적 언어로 요구사항을 정의하는 것이 곤란하였다. 셋째, 개발 목표에 대응하는 참조 모형(reference model)이 없다는 점이다.

3. GIANT 개발방법론

앞에서 설명한 특징을 갖는 GIANT 개발사업을 효율적으로 수행하기 위한 개발 방법론을 정립하기 위하여 기존에 제시된 모형을 조사하였다. GIANT 개발사업에 적용 가능한 모델로서는 폭포수 모델(waterfall model)[Ramamoorthy, 1984], 시제품 모델(prototyping model)[Agresti, 1986], 나선형 모델(spiral model)[Blanchard, 1990] 등을 고려하였다. 이중 폭포수 모델은 소프트웨어 개발사업에서 적용하는 전형적인 모델로서 사용자 요구사항 즉, 연구 목표가 명확한 경우에 효과적인 방법이다. 시제품 모델은 비교적 소규모이면서 기술 수준이 높은 첨단 소프트웨어 개발시 적용하는 방법이며, 나선형 모델은 시제품 제작 과정을 여러번 반복할 수 있으며 연구 목표를 수정할 수 있을 경우에 적용하는 방법으로서 이들은 우리나라에서는 채택해 본 경험이 없는 방법들이다.

앞에서 지적한 대로 본 개발사업은 개발 목표를 기술적 측면에서 정확히 정의하기가 불가능했기에, 개발방법론 정립시 선행시제품(rapid prototyping) 개념을 도입하였다. 즉, 시스템 설계 단계를 시스템 개략 설계와 시스템 상세 설계로 구분하고, 시스템 개략 설계 단계를 선행시제품을 확인하는 시스템 검증모델 구현 단계로 정의하였다.

GIANT 개발사업에서는 앞에서 서술한 개발사업의 특성을 고려하여 폭포수 모델과 시제품 모델을 결합시킨 형태의 순기 모형을 정의하였다. 이렇게 한 이유는 규모가 큰 연구개발 사업을 수행한 경험이 많지 않은 우리나라 실정상 개발 환경이나 문화가 상이하고, 연구 개발 및 관리 수준이 높은 선진국에서 제시한 모델을 그대로 적용하기에는 많은 혼란과 무리가 있을 것으로 판단되었기 때문이다. 또한 본 사업이 정부 주도형이기 때문에 당초 정해진 기간 내에 완료하여야 한다는 점도 고려하였다.

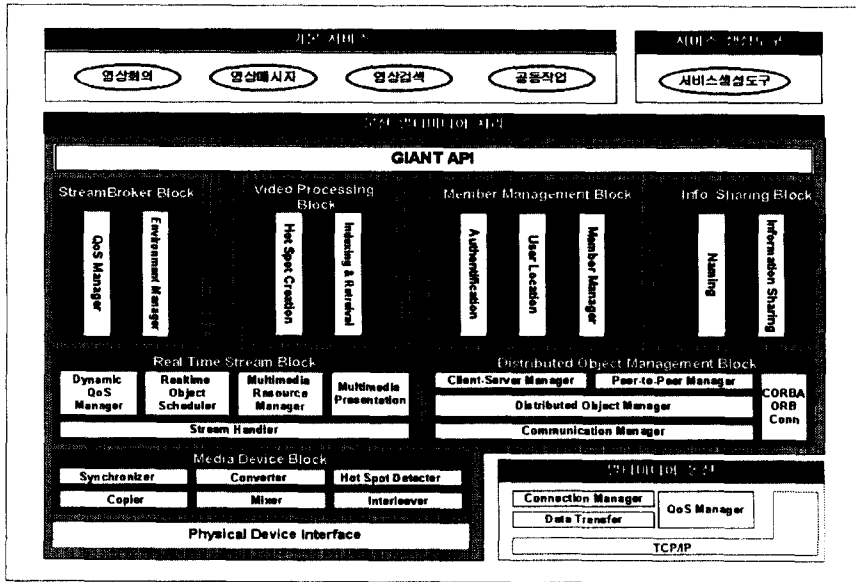
이렇게 정립한 개발방법론은 기존의 폭포수 모델과 시제품 모델의 장점을 취한 것으로서 GIANT처럼 연구 목표가 불명확하고 정해진 기간 내에 완료해야 하는 개발사

업의 경우에 적합하다고 판단된다. 그리고 개발하고자 하는 전체 시스템을 4개의 개발 단위로 분해하여 각각을 개별적, 독립적으로 구현한 후, 이들을 통합하는 방법을 채택하였다. 본 논문에서는 기가비트 정보통신 시스템 소프트웨어 개발 체계(V.2.1)에서 정한 GIANT 순기체계가 내포하고 있는 2가지 철학 즉, 시스템의 수직적 분해(요소별 분해)와 수평적 분해(단계별 수행)에 대하여 살펴본다.

3.1 시스템의 요소별 분해

GIANT 개발사업을 본격적으로 추진할 때 유사한 사업 추진 내용을 파악하기 위하여 외국의 사례 조사를 수행하였다. 이와 동시에 국내,외의 대규모 연구 개발 사업의 개발 방법론을 조사하였다. 유사한 사업 내용을 파악하기 위하여 미국의 HPCC, 일본의 정보 고속도로, 유럽 공동체(EC)의 RACE, 싱가포르의 IT 2000 등의 관련 자료를 분석하였다.

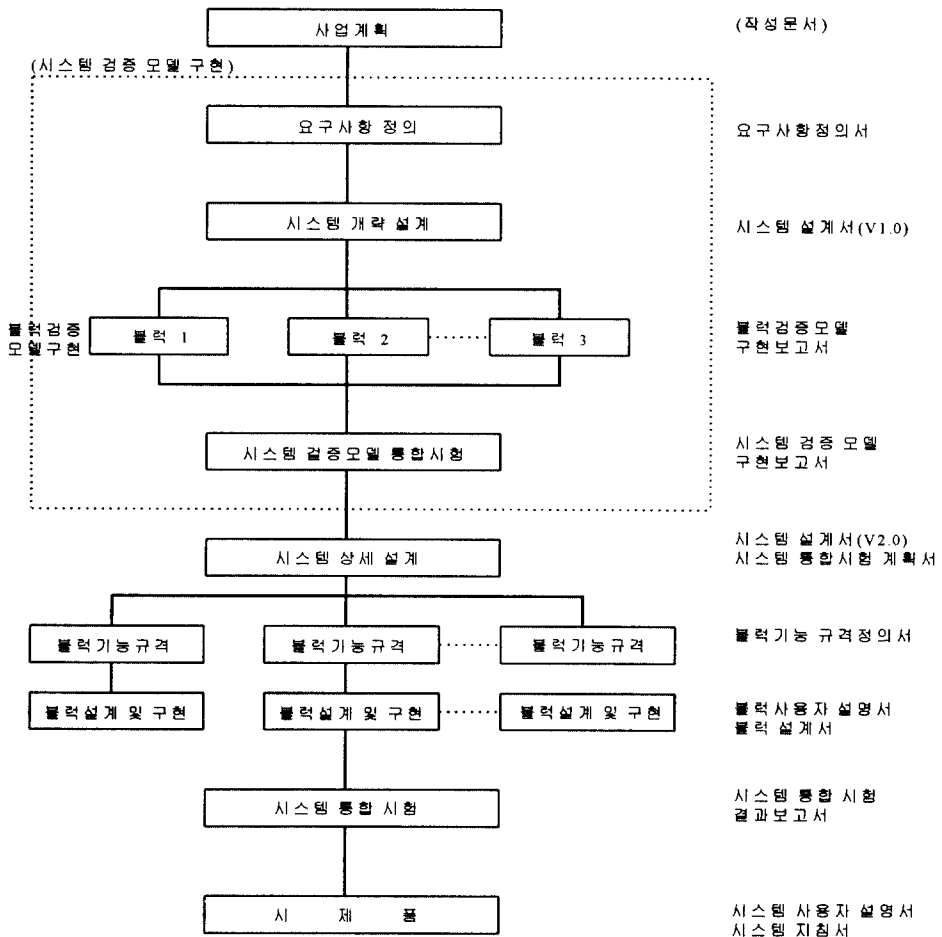
GIANT 개발사업은 대규모의 소프트웨어 개발사업이기 때문에 전체 시스템을 독립적인 개발성 및 조직의 단위를 기준으로 블록(block)으로 분해하였다. 따라서 각 블록은 다른 블록과 사용자 요구사항, 구현 및 시험 측면에서 거의 독립적이다. 이에 각 블록은 시스템으로 통합되기 이전에는 상호 관련이 있는 인터페이스 부분만이 공유되어 관리될 뿐 서로 독립적으로 구현되어 시험된다. 블록내에서의 구현의 편이상 여러 개의 기능 모듈로 세분화되기도 하지만 이는 시스템 수준에서는 고려하지 않는다. 이러한 시스템의 요소별 분해 결과는 2.1절에서 서술한 바와 같으며, 이를 그림으로 나타내면 <그림2>와 같다.



< 그림 2 > GIANT시스템의 요소별 분해

3.2 개발 과정의 단계별 수행

GIANT 개발사업에서는 top-down 방식의 설계, 블럭간의 상호 독립적인 구현, bottom-up 방식의 통합 및 시험의 3가지 원칙하에 개발방법론을 정립하였다. 이러한 배경에서 개발 과정을 앞에서 서술한 바와 같이 순기 모형화 하고 각 단계별 주요 결과물을 정의하여 이를 <그림3>에 나타내었다.



< 그림 3 > GIANT 개발 단계

4. GIANT 개발방법론 분석

GIANT 개발방법론을 품질공학적인 측면에서 분석하면 초고속정보통신망이 갖는 자체적인 특징을 바탕으로 한 선행시제품 전략과 개발 요소별 자원 할당 전략을 특징으로 파악할 수 있다.

4.1 선행 시제품 전략

전체적인 관점에서 볼 때 앞절에서 서술한 개발 단계에 따라 개발 과정이 진행되고 있기는 하지만 내부적으로 보면 여러 번의 변경 과정이 있었다. 초고속정보통신망에서 수많은 불특정 다수의 사용자를 가정하여 만드는 소프트웨어이기 때문에 소위 말하는 분할정복(divide-and-conquer)형태의 전략을 선택하였다. 이에 의하여 목표로 하는 규격의 소프트웨어 개발에 이르는 과정 중에 블럭별 선행시제품을 개발하여 이를 평가한 후, 규격을 재조정하는 방식을 채택하였다. 이렇게 한 또 다른 이유는 멀티미디어 처리에 관한 기술 및 상품 수준이 매우 빠르게 변하고 있기 때문에 이를 동적으로 반영하기 위함이었다.

정해진 기간내에 가시적인 결과를 반드시 제시하여야 하는 정부출연 연구개발 사업의 특성과 목표 규격을 정의하기가 분명치 않은 특성이 있는 첨단 기술 개발사업의 경우에는 이러한 형태의 개발 전략을 고려하는 것도 하나의 방법이 될 것이다.

4.2 개발 요소별 자원 할당

목표 규격을 초기 단계에 설정할 수 있는 일반적인 개발사업의 경우에는 자원의 할당 문제 또한 비교적 정형화하여 접근할 수 있다. 그러나, GIANT 개발사업의 경우와 같이 목표 규격이 모호할 경우의 자원 할당 문제는 우선 추정하여 할당한 후 개발 사업을 진행하면서 신속적으로 대응하여야 하며, 연구 조직도 조직인 만큼 구성원의 행동 과학에 근거하여 접근하여야 한다. 개발사업에서의 가장 중요한 자원은 연구원이다. 이들이 상호 신뢰하는 분위기에서 개발 업무를 수행할 수 있을 때 연구 생산성이 향상될 것이다.

GIANT 개발사업의 경우에는 몇번의 조직 변경 및 자원 할당 과정이 있었다. 장단점이 있지만 연구원들의 의욕이 저하되는 현상이 나타나 이를 해소시키는 데에 많은 시간이 필요하였다.

GIANT 시스템은 서비스 플랫폼 분야, 멀티미디어 처리를 담당하는 미들웨어 분야, 이기종간 신뢰성 있는 통신을 위한 멀티미디어 통신 프로토콜 분야로 구분하였으며, 이들을 통합하고, 시험하고, 중간 결과물을 상호 조정할 수 있도록 하는 시스템 업무 분야를 정하였다. GIANT 개발사업에서는 이들간의 연구인력 비율을 1:1:1:0.5의 비율로 유지하고 있는 바, 이에서 서비스 플랫폼 분야와 시스템 분야를 합쳐서 보면 1.5:1:1의 비율을 볼 수 있다. 대부분의 시스템 개발사업에서 할당하고 있는 시스템:하드웨어:소프트웨어의 비율이 1:1:2인 점[박진원, 1991]과 비교해 보면 어느 정도 유사한 면이 있다고 할 수 있다.

4.3 개발체계상의 보완점

시스템 및 소프트웨어 개발과정에서 발생하는 모든 결과물 및 기술 문서는 개발 과정이 진행됨에 따라 어떤 형태의 변경이 있게 마련이다. GIANT 개발사업과 같은 소프트웨어 개발사업의 품질보증을 위해서는 이러한 형태의 변경 내용 및 내역을 통제하고 기록, 보관하여 관련되는 다른 개발 단위에 통보해 주어야 하는 등의 업무가 필요하다. 이러한 형상관리 업무가 앞에서 설명한 개발체계에 미비되어 있어 GIANT형상관리 체계를 정립하여 이를 적용하고 있다.

5. 품질보증을 위한 형상관리 체계도입

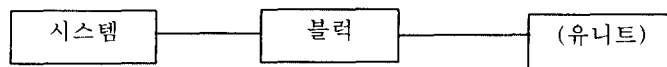
연구개발 과정에서의 품질보증을 위해서는 연구개발 과정에서 발생하는 결과물 및 기술 문서의 기능적, 물리적 특성을 식별, 구분하여 문서화하고 이들 특성의 변경을 통제하며 현황 및 변경 절차를 기록, 보관하는 등의 관리 업무가 필요하다. 이에 관련된 일련의 활동, 즉 하나의 시스템 라이프 사이클 전반에 걸쳐 정의되는 형상 항목과 그에 관련되어 생성되는 형상물을 종합하여 시스템 형상을 형성하고 이들에 대한 변경을 체계적으로 관리함으로써 개발되는 결과물의 품질을 보증할 수 있도록 하는 업무를 형상관리 업무라고 하며[박진원, 1991], 이를 위한 체계를 형상관리 체계라 한다. 이는 개발 과정중의 결과물을 최종 목표 시스템에 적합시켜 궁극적으로는 시스템의 품질 유지 및 품질 향상을 가능토록 하며, 고장처리/수정/개선에 따른 중복 비용을 줄이기 위한 것이다.

5.1 GIANT 형상관리 체계

GIANT 형상관리 체계는 형상체계, 형상관리 절차, 형상 번호부여 체계의 세부사항으로 구성하였다.

5.1.1 형상체계

GIANT 형상 체계는 <그림 4>에서의 해당 계층별로 형상항목을 정의하며 또한 라이프사이클의 각 단계별로 생성되는 형상물을 정의함으로써 형상인식이 가능토록 하였다



< 그림 4 > GIANT 형상 체계

5.1.2 형상관리 절차

형상관리 활동을 구체적으로 추진하기 위하여 다음의 업무를 정의하고 각각의 절차를 확립하였다.

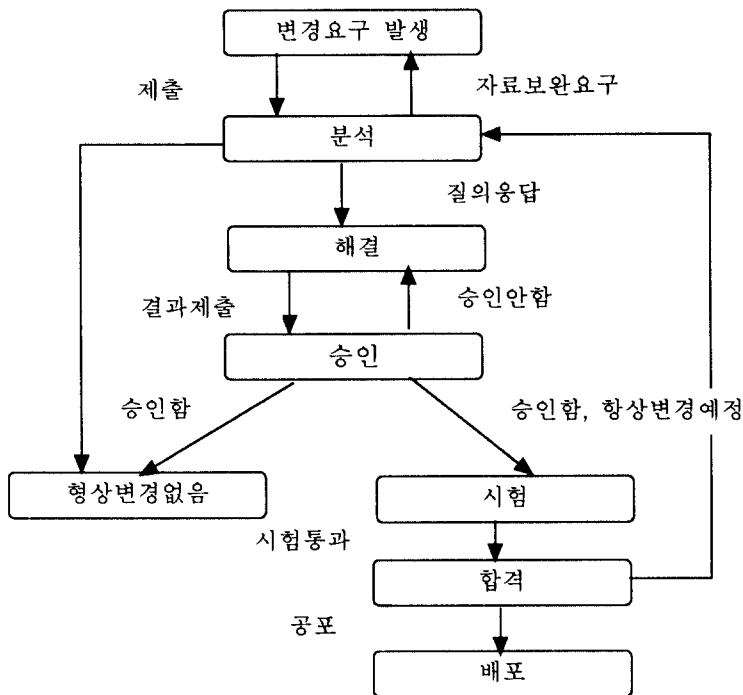
(1) 형상 식별(configuration Identification)

형상 식별은 시스템 개발 시작 시점부터 개발 완료 시점까지의 과정 중 연구원에 의하여 생성되는 형상 항목과 형상물을 종합하여 관리하는 활동으로서, 이 단계에서는 중간 결과물에 대한 수정을 수시로 할 수 있도록 함으로서 인력 및 시간의 소모를 줄이고자 하였다. 형상 식별 단계에서의 주요 활동은 형상 항목 관리, 문서 관리, 소프트웨어 결과물 관리가 있으며, 발생하는 모든 형상물에 대하여 고유한 형상 식별자를 부여한다.

하나의 형상 항목은 한 개 이상의 형상물로 구성되며, 형상물은 각 단계별로 구현된다. 구현 단위인 블록이 실현되면 화일(file)의 형태를 취하는데, 이 화일과 관련 기술 문서를 합하여 하나의 형상물이라고 정의하였다.

(2) 형상 통제(configuration control)

형상이 형성된 이후에 발생하는 모든 형상 변경(기술적 개량, 보완 등)에 대한 처리 절차를 정의하였다. 형상의 변경은 변경 요구(CR: change request)를 매체로 하여 그 상태를 제어함으로서 실현된다. 변경 요구의 상태 전이도는 <그림 5>와 같으며, 변경 요구의 처리 흐름도는 <그림 6>과 같다.



< 그림 5 > 변경요구 상태 천이도

단 계	업 무 흐 름	업 무 내 용
ENTER (CR 발생)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">CR 생성</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">CR 등록</div> </div>	<ul style="list-style-type: none"> - 발의 - CR 작성 - CR 등록
ANALYSIS (분 석)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 100px; margin: 0 auto;">검토 및 분석</div> <div style="border: 1px solid black; padding: 2px; width: 100px; margin: 0 auto;">분석 및 승인</div> </div>	<ul style="list-style-type: none"> - 기술검토위원회구성 - CR 검토 - CR 검토보고서 작성 - 해결 요청서작성 - 해결 담당팀에 해결요청서 전달
SOLVE (해 결)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">해 결</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">검토 및 승인</div> </div>	<ul style="list-style-type: none"> - 해결 결과보고서작성 - 변경 형상물 가등록 - CR 해결 검토 및 승인
APPROVE (승 인)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">승 인</div> </div>	<ul style="list-style-type: none"> - CR 해결 최종 승인
TEST (시험)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">시험요청</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">시 청</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">검 토</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">승 인</div> </div>	<ul style="list-style-type: none"> - 시험요청서작성 및 전달 - 시험수행 - 시험결과 보고서 작성 - 시험결과보고서 검토 - 시험결과 승인
PASS (합 격)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">최종승인</div> </div>	<ul style="list-style-type: none"> - 형상관리위원회구성 - 형상변경승인 및 배포결정 - 최종승인
ISSUE (배 포)	<div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">정 리</div> <div style="border: 1px solid black; padding: 2px; width: 80px; margin: 0 auto;">배 포</div> </div>	<ul style="list-style-type: none"> - 문서 정리 및 보관 - 배포 및 상황판작성

< 그림 6 > 변경요구 처리 흐름도

(3) 형상 정보 관리

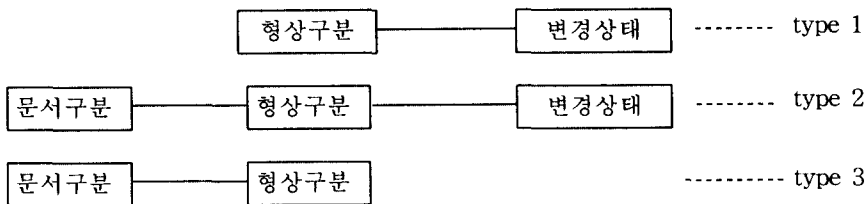
시스템을 구성하는 각 결과물들의 형상 상태를 기록하고 그 상태의 추적을 위한 이력을 유지 관리함으로써 사용자가 필요로 하는 정보를 효과적으로 제공하기 위한 활동으로서, 각 형상 항목 및 형상물에서 발생하는 기능 변경 및 확장에 따른 각종 변경 요구들을 종합적이고 체계적으로 관리함으로써 시스템의 기능 향상 및 유지 보수를 효율적이고 용이하게 수행하기 위함이다.

관리를 위한 원칙으로서, 형상 변경 단계에서 형상 대상 정보의 수정 및 변경은 변경 요구를 통해서만 이루어지도록 하고, 자료 기록 및 변경 정보 관리는 GIANT 형상 관리 시스템(GCMS: GIANT configuration management system)을 개발하여 적용하고 있다.

5.1.3 형상 번호부여 체계

시스템 개발중 혹은 시험/유지보수 기간중 결과물의 정보를 참조하거나 변경할 필요가 있을 때 가장 먼저 해야 할 일은 관련 정보를 추적하는 것이다. 추적의 수단으로서는 결과물의 형상과 문서의 식별 코드를 이용한다. 형상 번호부여 체계에서는 GIANT의 형상 항목과 형상물에 식별 코드를 부여하는 방법을 정의한다. 번호 부여는 GIANT 형상 체계에서 정의하는 형상 항목과 구현되는 결과물 즉, 시스템 및 블럭 수준의 결과물과 관련 문서를 대상으로 한다.

GIANT 개발에 사용되는 모든 형상물의 번호부여 체계는 문서 구분, 형상 구분 및 변경 상태를 나타내는 3부분으로 구성한다(<그림 7>). 문서 구분은 특정 형상에 대한 문서의 종류를 식별하기 위하여 표기하는 것으로, 각각의 문서에는 4자리의 십진수로 고유한 번호를 부여한다. 형상 구분은 시스템 형상 체계상의 시스템-블럭-(유니트) 단계의 해당 형상을 의미한다.



< 그림 7 > 형상 번호부여 체계

형상 구분의 표시방법은 3개의 영문자와 1개의 영문자 및 2개의 숫자로 표기한다. 3개의 영문자는 각각의 형상을 나타내는 식별 코드이며, 1개의 영문자와 2개의 숫자는 해당 형상을 보다 계층적으로 식별할 수 있도록 첨가하는 보조 코드이다. 변경 상태는 형상 혹은 문서의 변경 상태를 나타내는 것으로 3자리의 십진수로 나타낸다. 첫 번째 숫자는 버전(version) 번호를, 두 번째 숫자는 릴리즈(release) 번호를 나타내며, 마지막 숫자는 고정(fix) 번호를 의미한다.

형상 번호 체계는 문서 구분, 형상 구분 및 변경 상태를 조합하여 크게 3가지의 번호 형태(type)를 정의하였다. 시스템의 특정 형상을 식별하기 위하여는 형상 구분과 변경 상태 코드를 이용한다(type 1). 특정 형상에 관한 문서를 식별하기 위해서는 문서구분, 형상구분 및 변경상태 코드를 이용한다(type 2). 이 경우 특정 형상에 속한 문서를 문서의 변경 상태를 나타내지 않고 표시할 경우에는 문서 구분과 형상 구분 코드를 이용한다(type 3).

6. 결론

본 논문에서 제시하는 개발방법론은 개발 목표가 비교적 불명확하고 정해진 기간 내에 완료해야 하는 소프트웨어 개발사업의 경우에 효과적이다. 이를 GIANT 개발사업에 적용한 결과는 다음과 같다. 우선 총 개발 기간 3년 중 1년을 개발 목표 설정에 할애함으로써 선행시제품 구현을 통한 목표 설정이 가능하였다는 점이다. 이는 남은 2년의 기간을 명확한 목표를 가지고 개발할 수 있도록 하였다. 둘째는 불특정 다수의 사용자를 대상으로 한 서비스 소프트웨어 플랫폼을 선행시제품 과정을 통하여 구현한다는 점과 서비스 소프트웨어 플랫폼 및 시스템 분야, 미들웨어 분야, 통신 프로토콜 분야의 연구 인력 구성 비율이 1.5:1:1인 점을 들 수 있다. 이 비율은 다른 연구 개발사업의 사례를 분석해서 일반화 시킬 수도 있다고 본다. 그리고 GIANT 개발사업처럼 개발 목표가 기술적 측면에서 모호하며 정해진 기간내에 완료해야 하는 대규모 소프트웨어 개발사업의 경우 부분적으로 분할정복 전략을 도입하여 문제점을 해결하는 방식은 유사한 개발사업에서 원용할 수 있는 개념의 적용사례를 제시한다고 본다. 아울러, 개발 목표가 불확실한 경우의 개발 체계에 형상관리 체계를 도입함으로써 연구 결과물의 품질 보증이 가능하다고 판단된다. 즉, 정해진 기간내에 정해진 목표의 개발을 완료하기 위해서는 개발 단위인 소프트웨어 모듈별 기능 및 인터페이스 방법 등에 대한 구체적 형상이 체계적으로 관리되어 필요할 때에 활용될 수 있도록 하여야 하는데 이는 통합 시험을 위한 기본 자료로 활용되며, 통합시의 문제점 해결을 위한 중요 수단이 된다.

마지막으로 대규모이면서도 첨단 기술 개발의 경우에는 가장 중요한 것이 조직의 구성 및 연구원간의 인간적인 측면이다. 앞전에서 언급했다시피 조직의 재구성 또는 통합 업무 수행시에는 자칫 연구팀간의 불협화음을 유발할 수 있다. 이러한 경우에는 연구원간의 행동 과학적 측면을 다시 한번 고려하여, 이에 따른 마이너스적 요소를 최소화하도록 하여야 할 것이다. 정보통신 분야에서의 연구개발 사업이 점차 전문화, 복합화 되어 가는 경향이다. 이러한 차원에서 수행되는 여러 개발사업에서의 품질공학적 사례를 수집하여 공통적으로 적용할 수 있는 부분을 정립함으로써 향후 개발사업에 적용토록 함이 바람직할 것이다.

참고문헌

- [1] 천유식(1991), 「시스템 개발 방법론」, 컴퓨터 월드, pp. 243-261.
- [2] 박진원(1991), "TDX와 TICOM의 개발체계 비교," 「전자통신」, 제14권, 제4호, pp. 88-96.
- [3] Agresti, W.W.(1986), *New Paradigms for Software Development*, IEEE Computer Society Press.

- [4] Blanchard, B.S., Fabrycky, W.J.(1990), *System Engineering and Analysis*, 2nd Ed., Englewood Cliffs, N.J., Prentice-Hall, pp. 16-33.
- [5] Ramamoorthy, C.V.(1984), "Software Engineering: Problems and Perspectives," *Computer*, pp.191-209.
- [6] Rock, P.(1986), "Controlling Software Projects," *Software Engineering Journal*, pp.108-117.