

장해관리를 중심으로 한 품질관리 지원 시스템의 구현

양 해 술[†] · 이 하 용^{††} · 안 유 환^{†††}

요 약

소프트웨어의 무형성, 비가시성으로 인해 체계적인 품질관리를 실시하는데 어려움이 있으나 구성관리를 통해 소프트웨어를 하드웨어 생산과 같이 일괄적인 품질관리의 시도와 함께 소프트웨어 품질관리 표준요강을 제정하여 이 표준지침에 따라 소프트웨어의 품질을 관리함으로써 효율적인 품질관리를 실시하고 있다. 본 연구에서는 오류의 입력감소 및 조기발견이라는 기본 방침하에 이러한 품질관리활동을 지원하기 위한 소프트웨어 품질관리 지원시스템을 개발하려는 시도로서 소프트웨어 품질관리 시스템을 장해관리 지원기능을 중심으로 살펴보고 장해관리 지원기능을 보조할 수 있는 각종 시각적인 출력 요소들을 정리하였다. 또한, 이러한 소프트웨어 품질관리 시스템을 구축하게 된 배경을 살펴봄으로써 품질관리 표준화에 따르는 문제점 및 기존의 품질관리 도구에서 제기되는 문제점을 고려하여 시스템 구축을 위한 과제와 그 대책을 수립하였으며 끝으로 장해관리 지원 기능을 이용했을 때의 효과를 기술하였다.

Implementation of Software Error Management Supporting System

Hae-Sool Yang[†] · Ha-Yong Lee^{††} · You-Whoan Ahn^{†††}

ABSTRACT

Because of formlessness and sightlessness of software, systematic quality management is hard to implement. But we attempted collective quality management as hardware production by configuration management and settled on software quality management standard outlines. We are implementing efficient quality management by software quality management according as these standard guidelines. In this study, we study error management supporting functions as an attempt of development of software quality management supporting system which can support quality management activity by reduction and early finding of error input, and represent various visual output elements which can help error management supporting functions. And we represent the background of construction of this software quality management system and searched effects which we can get when we use the error management supporting function.

1. 서 론

최근 소프트웨어 개발 영역이 계속 확장됨에 따라

고품질 소프트웨어를 생산할 수 있는 기반 기술인 품질관리는 더욱 중요하게 부각되고 있다. 이와 같은 이유는 소프트웨어가 대규모화 복잡화되기 때문에 종래와 같은 개발중심의 소프트웨어 개발은 소프트웨어의 성능과 품질을 저하시키는 요인이 되기 때문이다 [1, 7, 8]. 따라서, 소프트웨어 생명주기 전단계에 걸쳐 각 단계별로 소프트웨어의 품질을 평가하여 이를 만

† 종신회원: 한국소프트웨어품질연구소 소장

†† 종신회원: 한국소프트웨어품질연구소 선임연구원

††† 정 회 원: 시스템공학연구소 품질관리연구실장

논문접수: 1996년 6월 18일, 심사완료: 1997년 6월 25일

축할만한 수준으로 끌어올릴 수 있는 방법 즉, 품질 관리를 위한 방법론 및 지원도구의 개발이 요구되고 있다. 고품질의 소프트웨어를 개발하기 위해서는 생명주기 초기단계부터 전단계에 걸쳐 소프트웨어의 유지보수성을 고려한 체계적인 품질관리를 통해 소프트웨어의 품질을 객관적으로 측정할 수 있어야 하며, 촉박한 개발 일정을 지원할 수 있는 품질관리 지원시스템의 개발 또한 절실히 필요하다[9, 10].

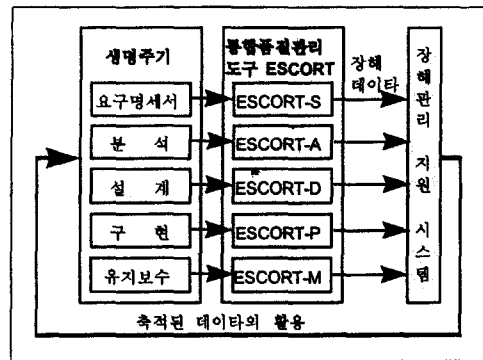
소프트웨어의 품질관리를 실시할 때에는 하드웨어를 제조할 때와 마찬가지로 공산품처럼 일괄적인 품질관리 제도에서 생산한다는 기본적인 사고방식을 가지고, 이 사고방식에 기반을 둔 소프트웨어 품질관리 표준지침을 제정하고 개량함과 동시에 표준지침에 따라 소프트웨어의 품질관리를 실시해야 한다. 아울러 오류 입력을 감소시키고 오류를 조기에 발견한다는 기본방침에 따라 소프트웨어의 품질을 높이려는 목적으로 일관된 품질보증을 구현하기 위한 품질관리제도를 확립해야 한다. 또한, 변경방침관리의 추진에 의해 품질지침에 기초한 품질목표를 철저히 관리함을 기본으로 한 방침의 설정 및 전개를 실시할 필요가 있다[14]. 본 연구에서는 이와 같은 품질관리활동을 효과적으로 지원하기 위해서 생명주기 전 단계의 품질평가를 지원하는 통합품질관리도구 ESCORT (Evaluation Software for Computerized Integrated Quality Management)를 구축하고 통합품질관리도구의 각 구성 도구로부터 전달되는 장해데이터를 통합하여 처리할 수 있는 장해관리 지원 시스템을 구축하여 품질관리의 효율화와 질적향상을 도모하고자 소프트웨어 품질관리지원 시스템의 개요와 각각의 프로젝트에 유효하게 활용해 온 효과를 장해 관리의 측면에서 소개하고자 한다.

본 연구의 2장에서는 개발한 품질관리지원 시스템의 전체적인 기본구성과 장해관리 지원기능의 개념 및 기능에 대해 기술하였으며, 3장에서는 지금까지 품질관리에서 발생한 문제점들을 기술함으로써 장해관리 중심의 품질관리지원 시스템에 대한 개발 배경을 살펴보고, 시스템 구축시 해결해야 할 과제와 그 대책을 수립하였으며, 4장에서는 장해관리 지원기능의 구현에 대해 기술하였다. 그리고 5장에서는 품질관리 지원시스템을 실제 프로젝트의 품질관리에 적용한 사례를 제시하였으며 품질관리지원 시스템을

적용했을 때 얻을 수 있었던 효과를 기술하였다. 끝으로 결론 및 향후 연구과제를 제시하였다.

2. 품질관리지원 시스템의 구성과 구축 배경

소프트웨어 품질관리활동을 효과적으로 실시하기 위해서는 개발완료 전후에 품질 데이터를 일괄적으로 축적 및 관리하고 이들을 다각적으로 집계, 분석, 평가, 기록하고 피드백하는 것이 중요하다. 이를 위해 생명주기 단계별 품질평가를 지원할 수 있는 소프트웨어 품질관리 지원시스템을 (그림 1)과 같은 체계로 구축하였다.



(그림 1) 소프트웨어 품질관리 지원시스템
(Fig. 1) Software quality management supporting system

2.1 시스템의 기본 구조

1) 요구명세서의 품질평가 지원

소프트웨어 명세서를 기술하기 위한 설계방법과 기술방법이 제안되어 있으나 실제로 대부분의 명세서는 자연어로 자유롭게 기술하는 경우가 대부분이다. 요구명세서의 품질평가 도구는 명세서의 구성과 각 명세서의 서식을 설정하여 구축함으로써 서식에 따라 명세서를 기술하고 기술된 명세서에 대한 품질평가를 실시하여 문제점을 도출하여 개선지침을 개발자에게 피드백하여 효율적인 명세서 작성을 지원함으로써 요구명세서에 대한 품질향상을 지원한다.

2) 분석단계 산출물의 품질평가 지원

분석단계 산출물에 대한 품질평가를 위해 ISO/IEC 9126의 품질특성 6항목을 품질특성으로 도입하여 품

질부특성 및 품질내부특성을 개발하여 구축하고 내부특성에 대한 매트릭스와 계산식 및 계산식을 구성하는 요소데이터를 개발하여 이들을 자동적으로 계산할 수 있는 평가 지원도구를 구축하고 있다.

3) 설계단계 산출물의 품질평가 지원

설계단계 산출물에 대한 품질평가를 위해 ISO/IEC 9126의 품질특성 6항목을 품질특성으로 도입하여 품질부특성 및 품질내부특성을 개발하여 구축하고 내부특성에 대한 매트릭스와 계산식 및 계산식을 구성하는 요소데이터를 개발하여 이들을 자동적으로 계산할 수 있는 평가 지원도구를 구현하였다.

4) 구현단계 산출물의 품질평가 지원

구현단계 산출물에 대한 품질평가는 프로그램 명세서를 구성하는 산출물에 대한 평가와 개발된 소스코드 자체에 대한 평가로 구분할 수 있다. 프로그램 명세서를 구성하는 산출물에 대한 평가를 위해 품질특성, 부특성, 내부특성을 구축하고 매트릭스와 계산식 및 요소데이터를 개발하여 도구화하였으며 소스코드에 대한 품질평가를 위해서는 기능사이즈 모델, 이해용이성 모델, 복잡성 모델, 구조화 모델, 객체지향 모델 등의 품질평가 모델을 구축하고 각 모델에 대한 매트릭스 및 계산식을 개발하여 도구화함으로써 화일로 이루어진 소스코드를 입력으로 하여 신속한 품질평가를 통해 결과를 개발자에게 피드백할 수 있도록 구성하였다.

5) 유지보수단계 산출물의 품질평가 지원

유지보수단계 산출물에 대한 품질평가를 위해 ISO/IEC 9126의 품질특성 6항목을 품질특성으로 도입하여 품질부특성 및 품질내부특성을 개발하여 구축하고 내부특성에 대한 매트릭스와 계산식 및 계산식을 구성하는 요소데이터를 개발하여 이들을 자동적으로 계산할 수 있는 평가 도구를 설계하였다.

이와 같이 통합품질관리 도구는 생명주기 단계별로 요구명세서 작성단계, 분석단계, 설계단계, 구현단계 및 유지보수단계의 품질평가를 지원할 수 있도록 구성되어 있으며 각 단계에서 개발되는 산출물에 대한 품질평가 체계와 매트릭스를 개발하였고 이들을 도구로 구현하여 각 단계의 품질평가를 담당할 수 있도록 하였다.

각 단계에서는 품질평가 결과 발생된 장해데이터들을 장해관리 지원시스템에 전달하며 단계별로 통합 관리된다. 현재의 품질평가 정보는 데이터베이스에 저장된 과거의 품질평가 정보와 비교되어 품질평가한 시스템에 대한 수준을 파악할 수 있다. 따라서, 본 연구에서는 생명주기 단계를 완료하기 전에 각 단계에 따른 정량적인 품질 데이터에 기초하여 품질관리를 효과적으로 실시하기 위한 장해관리 지원기능에 대해 기술하기로 한다.

2.2 장해관리 지원 시스템의 구축 배경

시스템을 구축하려는 배경으로는 첫째, 품질관리의 표준화 둘째, 기존의 지원 도구 셋째, 품질관리 보급 및 정착 활동의 3가지 관점에서 다음에 기술한 문제점을 해결할 필요가 있다.

1) 품질관리 표준화의 문제점

표준적인 관리항목을 설정하고 그것에 대응하는 관리장표를 정하여 운용하고 있지만 대규모 프로젝트에서 소규모 프로젝트까지 동일한 관리 레벨이기 때문에 대규모 프로젝트에서는 관리항목이 부족할 수도 있으며 소규모 프로젝트에서는 관리항목이 많아서 관리하기 어려운 경우가 발생하고 있다.

2) 기존 지원도구의 문제점

기존의 지원도구에서 나타나는 문제점으로는 첫째, 개발시험과 적용시험 사이에서 시험환경에 있는 서로 다른 지원 도구는 데이터베이스 구조의 차이때문에 연속된 장해 데이터를 볼 수 없고, 개발부서에서 현장에 이르기까지 일괄적인 관리를 할 수 없는 상황이다.

둘째, 각각의 생명주기 단계별로 장해 데이터의 평가 및 분석을 하고 싶지만 지금까지는 생명주기 전단계를 지원하는 지원도구의 설계 및 구현에 관한 연구가 미흡하여 요구에 적절히 대처하지 못하고 있는 상황이다.

셋째, 정형적인 형태의 관리장표밖에 출력할 수 없기 때문에 원하는대로 평가하거나 분석할 수 없는 상황이다.

넷째, 생명주기 단계별로 지원되는 품질평가 결과를 통합한 집계, 문제점 분석이 용이하지 않다.

3) 품질관리보급과 정착활동의 문제점

현재 소프트웨어 품질관리 방법론이 명확히 구축되어 있지 못하며 더구나 품질관리는 생명주기 전단계에 걸쳐 실시되어야 하고 이에 따라 각 단계별 품질특성 및 매트릭스가 구축되어야 하지만 이들 또한 미흡한 실정이다. 품질관리 체계와 방법론이 미흡하므로 품질관리의 대상이 되는 생명주기 각 단계의 산출물 또한 개발자에 따라 내역이나 서식이 달라 평가에 곤란이 따르며 품질관리 방법론과 체계를 표준화하여 보급 정착시키는 과정에서도 대폭 수정이 예상된다. 또한 평가 매트릭스 및 각 생명주기 단계의 품질관리에 대한 의식은 정확히 무엇을 어떻게 관리하면 좋은가라는 관리방법이 명확히 정의되지 않은 상황에 있다. 이 때문에 장해처리 상황을 수작업으로 관리하고 있는 레벨에 있으며 신뢰성 평가, 품질분석, 품질목표 및 실적관리까지 실시하지 못하고 있다.

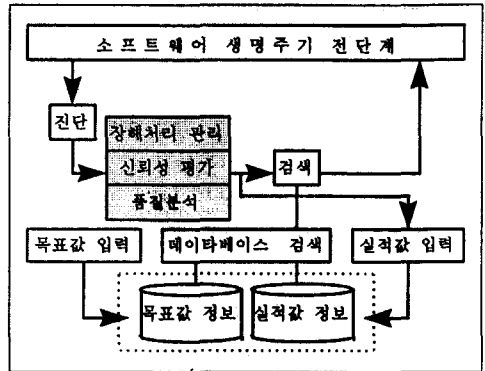
3. 장해관리의 지원기능과 대책

장해관리 지원기능은 생명주기 각 단계에서 발생하는 장해정보를 집계하고 분석하며 장표를 작성하는 등의 일과 관련된 소프트웨어 품질관리 업무를 효과적으로 지원하는 기능으로 (그림 2)에 기본적인 개요를 도시하였다. 또한, 본 시스템은 같은 그룹내 LAN으로 구축할 수 있으며 소프트웨어 개발자는 근처에 있는 워크스테이션에서 서버가 일괄적으로 관리하는 장해정보를 수시로 추출하여 정량적 데이터에 기초한 품질 평가 및 분석을 할 수 있도록 설계하였다.

3.1 장해관리 지원의 목표

생명주기 각 단계의 품질평가를 지원하는 도구는 각 단계의 산출물을 평가하기에 적합한 독자적인 매트릭스를 가지고 있다. 그러나 생명주기 각 단계의 매트릭스는 품질특성, 부특성, 내부특성의 체계에 따르고 있으므로 단계별로 관련된 품질특성에 대한 매트릭스에 대해서 통합적인 집계 및 분석을 실시할 필요성이 있으며 이러한 작업을 장해관리 지원도구에서 실시할 수 있다. 장해관리 지원의 목표는 첫째, 소프트웨어 품질관리 업무의 효율화와 둘째, 소프트웨어 품질관리의 질적 향상을 기하는데 있다. 본 연구에서는 품질관리 업무의 효율화와 품질관리의 질적

향상을 위해 통합품질관리도구를 구성하는 각 단계별 품질평가 도구로부터 발생하는 장해 데이터를 장해관리지원 시스템을 이용하여 통합관리를 목적으로 한다.



(그림 2) 장해관리 지원기능의 개요
(Fig. 2) Outline of error management supporting function

3.2 장해관리 지원의 기능

생명주기 각 단계의 품질평가 도구로부터 전달된 장해데이터는 장해관리 지원도구에서 통합 관리하며 장해데이터를 이용하여 다양한 품질관련 업무를 지원하게 된다. <표 1>에 지원하는 품질관련 업무에 대해 기술하였다.

<표 1> 지원가능한 품질관련 업무
<Table 1> Supportable quality related works

품질관련 업무	내 용
품질목표 관리와 실적관리의 지원	장해데이터를 바탕으로 한 품질목표값과 측정값을 비교·평가할 수 있도록 축적된 품질평가 정보의 검색 및 참조를 위한 데이터베이스 기능 지원
장해 복구 관리지원	장해발생과 복구상황을 파악하고 남기확보를 관리하고 지원하기 위해 데이터(건명, 발생일, 담당자, 복구일, 원인, 발생장소 등) 입력기능, 일람표, 각종 그래프 출력기능을 지원
신뢰성 평가지원	신뢰성을 예측하고 평가(잠재오류수의 추정, 수축시기의 예측)할 수 있도록 지원
품질분석 지원	중점적으로 품질을 강화해야 할 약점부분 및 장해원인의 분석을 지원
품질진단 지원	품질을 평가하고 진단하기 위한 각 처리장표 검색결과에 대해 이상값이 있는 경우 알람을 울리거나 경고 메시지를 출력하는 기능을 지원

3.3 장해관리 지원의 특징

① 레이아웃과 데이터 항목의 설정: 각 생명주기 단계별로 관리하기 쉽도록 임의의 관리장표 레이아웃과 데이터 항목을 설정할 수 있다.

② 데이터 교환: 각 단계별로 개발부서내 시험에서 현장적용시험에까지 일괄적인 관리가 가능하도록 적용시험용의 휴대용 개인용 컴퓨터와 데이터 교환이 가능하다.

③ 이용의 편리성: 각 단계를 이용하고 싶을 때(장소, 시간대)에 언제라도 이용할 수 있도록 근처의 워크스테이션에서 이용할 수 있다.

④ 대행서비스 운용: 각 단계의 장해 데이터 입력 및 관리장표의 검색과 출력의 작업 부하를 경감할 수 있도록 대행 서비스(집중운용 서비스)를 운용할 수 있다.

3.4 시스템 구축 과제와 대책

본 장에서는 2장에서 기술한 문제점에 대해 시스템 구축 과제를 설정하고 조기구축과 운용이라는 기본 방침에 기초하여 수립한 대책을 기술하였다.

3.4.1 품질관리의 표준화

품질관리를 위해 관리항목을 표준화하더라도 프로젝트의 규모에 따라 표준 관리항목이 적합하지 않을 수도 있다. 즉, 표준 관리항목의 규모가 일정하므로 프로젝트의 규모에 따라 신축성있게 대처하기 곤란하다. 이러한 문제점에 따라 발생하는 과제와 대책을 <표 2>에 기술하였다.

<표 2> 품질관리의 표준화에 관한 문제점 및 대책
(Table 2) Problem and counterplan about standardization of quality management

구분	내용
문제점	표준화된 관리항목이 프로젝트의 규모에 적합한 관리 레벨로 되어 있지 않다.
시스템 구축 과제	프로젝트의 규모에 맞는 관리 레벨의 설정
대책	프로젝트의 제품규모·신규도·복잡도를 고려한 관리 레벨을 3가지 레벨(S·M·L) 설정하고 집중통용 서비스의 관리자출력대행 서비스 메뉴로서 함께 투입한다.

3.4.2 기존의 지원도구의 문제점

개발부서와 현장사이의 데이터 교환이 용이하지 않을 경우 장해데이터의 연속성이 보호되지 않는 문제점이 있으며 이용환경에도 제한이 발생할 수 있다. 이러한 문제점들에 대한 문제해결 과제를 설정하고 대책을 수립하여 <표 3>에 기술하였다.

<표 3> 기존의 지원도구에 관한 문제점 및 대책
(Table 3) Problem and counterplan about existing supporting tool

구분	내용
문제점	a. 개발부서에서 현장까지의 일괄관리 가능하지 않다(장해데이터의 연속성이 보호되지 않음). b. 이용환경의 제한이 많다. c. 정형적인 관리장표밖에 출력되지 않는다.
시스템 구축 과제	a. 개발부서에서 현장까지의 일괄적인 관리화(장해 데이터의 연속성 유지) b. 이용환경의 유연화 c. 관리장표의 Customize의 용이성(부품화)
대책	① 프로젝트의 제품규모·신규도·복잡도를 고려한 관리 레벨을 3가지 레벨(S·M·L)로 설정하고 집중통용 서비스의 관리자출력대행 서비스 메뉴로 함께 투입한다. ② 현장시험에서의 장해관리가 가능한 휴대용 개인용 컴퓨터와의 데이터 교환이 가능하도록 데이터 변환기능을 개발한다. ③ 각 프로젝트 가까이 있고, 이용시간도 제한되지 않는 사내 통신 LAN상의 워크스테이션에 구축한다. ④ 운용관리자 부재중에도 시스템을 운용 및 유지할 수 있도록 보안과 데이터 보전성을 시스템에 고려한다. ⑤ 관리장표를 부품화하여 제공하고 각 프로젝트에서 초기에 생성하는 레이아웃과 관리항목의 설정 등이 임의로 가능하도록 한다.

3.4.3 품질관리 보급과 정착 활동

품질관리 체계 및 방법론이 현재 명확히 확립되어 있지 못하며 각 개발자들마다 생명주기 단계별로 생성되는 산출물에 대한 구성이나 서식을 달리하고 있는 등 낮은 형태로 사용하는 경우가 많다. 그러므로 품질관리 체계와 방법론에 대한 표준을 확립하고 보

급 정착시킬 때 많은 어려움이 뒤따른다. 이러한 문제점에 대한 해결과제와 대책을 <표 4>에 기술하였다.

<표 4> 보급과 정착활동에 관한 문제점 및 대책
 (Table 4) Problem and counterplan about supply and fixation activity

구분	내용
문제점	품질관리의 관리방법이 이해되어 있지 않다.
시스템 구축과제	품질관리 방법의 교육(보급, 정착)
대책	① 품질관리의 사고방식, 품질관리(특히 품질평가방법)를 실시하기 위한 구체적 방법과 지원시스템의 이용방법을 포함시킨 "품질관리 지침서"를 작성 및 배포하고 프로젝트에 대한 설명회 및 데모를 실시한다. ② 설문조사를 실시하여 사용자, 요구와 문제점을 반영하고 개선한다. (①~⑤의 대책 등) ③ 각 프로젝트로부터의 요구는 우선적으로 받아들여 "적극 협력해 준다!" 는 이미지를 부여하고 기능적으로 지원할 수 없는 부분은 운용에서 커버할 수 있는 방안을 세운다.

4. 장애관리 지원기능의 구현

장애관리 지원기능은 통합품질관리도구의 각 단위 도구로부터 구해진 매트릭스 값을 전달받아 처리하게 된다. 처리 절차에 따라 실적값이 저장되어 기존의 축적된 결과에 따라 정해진 목표값과 비교하여 결과를 피드백하게 된다.

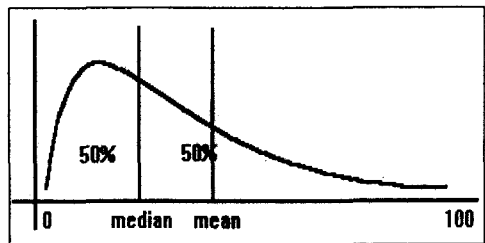
4.1 장애의 진단

생명주기 각 단계의 품질평가 도구는 단계별 산출물로부터 매트릭스에 의해 품질을 측정하고 품질평가 결과를 산출하게 된다. 품질평가의 최소단위는 요소데이터로서 결국 품질평가 결과가 좋지 않은 경우는 그 원인을 요소데이터에서 찾을 수 있다. 평가 결과가 좋지 않은 요소데이터를 장애데이터로 취급하여 생명주기 단계별 품질평가 도구는 품질평가 결과를 통해 장애데이터를 장애관리 지원도구로 전달하게 된다. 즉, 장애의 진단은 통합품질관리도구에 속한

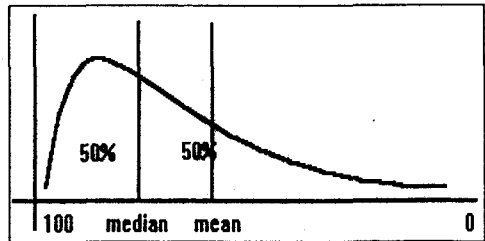
각 툴킷에서 실시하며 그 결과를 장애관리 지원 시스템에 전달해 준다.

4.2 목표값 입력

목표값은 과거에 실시한 품질평가 결과를 바탕으로 결정된다. 축적되어 있는 품질평가 결과들에 대한 중위수(M_e), 최대값(M_u), 최소값(M_l)을 구하고 이들 값을 득점값을 구할 때 반영한다.



(그림 3) Positive 매트릭스 값 분포 형태
 (Fig. 3) Distribution form of positive metrics values



(그림 4) Negative 매트릭스 값 분포 형태
 (Fig. 4) Distribution form of negative metrics values

<표 5> 매트릭스의 유형에 따른 득점 계산
 (Table 5) Point calculation for type of metrics

매트릭스의 유형	득점 계산식
Positive 매트릭스	$M > M_e$ $EV = 50 + \frac{50 * (M - M_e)}{(M_u - M_e)}$
	$M < M_e$ $EV = \frac{50 * (M - M_l)}{(M_e - M_l)}$
Negative 매트릭스	$M > M_e$ $EV = 50 - \frac{50 * (M - M_e)}{(M_u - M_e)}$
	$M < M_e$ $EV = 100 - \frac{50 * (M - M_l)}{(M_e - M_l)}$

메트릭스의 값이 클수록 양호한 값이 되는 경우에는 Positive 메트릭스라 하고 작을수록 양호한 값이 되는 경우에는 Negative 메트릭스라 한다. 이와 같이 메트릭스의 계산값은 Positive와 Negative로 분류되며 측정값을 분석해본 결과 값의 분포가 정규성을 따르지 않고 한 방향으로 강하게 치우쳐 있으므로 평균값을 이용해 득점을 구하기 보다는 중위수를 이용하여 득점을 구한다. (그림 3)에 Positive 메트릭스 값의 분포 형태를 나타내었고 (그림 4)에 Negative 메트릭스의 분포형태를 나타내었다. 메트릭스의 유형에 따라 득점을 구하는 식은 <표 5>에 기술하였다. 표에서 M은 해당 메트릭스에 대한 측정값을 의미한다.

<표 5>의 식에 따르면 메트릭스의 값이 축적된 데이터 중 최대치와 같으면 100점이 된다. 이점을 고려하여 목표 점수를 결정하여 입력한다. 동일한 목표 점수일 때에라도 데이터가 계속 축적되어 감에 따라 그 값의 수준은 달라지게 된다.

4.3 실적값 입력

실적값은 통합품질관리지원도구의 각 틀킷으로부터 전달된 값이다. 실적값은 데이터베이스에 저장되며 목표값과 비교되어 득점이 산출된다. 실적값은 필요할 때 검색할 수 있으며 축적된 정보로서 남게되어 차후 품질평가지 목표값을 새롭게 산출하는데 사용된다.

4.4 데이터베이스 검색

데이터베이스 검색은 첫째, 진단결과와 목표값 정보를 비교할 때, 둘째, 목표값을 산출하기 위해 과거에 축적된 실적값들을 참조할 때에 실시한다. 목표값 정보와 실적값 정보에 대한 데이터베이스는 (그림 5)과 같은 구조를 갖는다.

```

struct OBJECT_INFO {
    char    *info_code;
    int     value;
};

struct OBJECT_INFO
    object_info[NO_OF_ITEM];
    
```

(그림 5) 목표값 정보에 대한 자료구조
(Fig. 5) Data structure for object value infomation

목표값 정보는 메트릭스의 각 항목들을 구별하기 위해 코드화하여 구성하였으며 실적값 정보와 비교하기 위해 코드를 통해 동일한 항목을 찾게 된다.

4.5 장해데이터 축적을 위한 자료구조

품질평가 결과인 실적값 중에서 취약한 결과를 보인 장해데이터에 대해서는 그 결과를 축적하여 차후에 장해데이터에 대한 분석을 위해 사용하게 된다. 장해데이터의 축적을 위해 (그림 6)과 같은 자료구조를 구성하였다. 장해데이터의 목록은 평가 결과가 축적됨에 따라 증가될 수 있으며 각 장해데이터에 대한 점수 또한 계속해서 증가되기 때문에 포인터를 이용하여 장해데이터의 목록과 축적된 점수가 저장될 자료구조를 정의하였다.

```

struct err_data {
    char    *err_data;
    struct err_data *next_d;
};

struct err_grade {
    int     grade[ACCU_NO];
    struct err_grade *next_g;
}

struct err_data  err_dat;
struct err_grade err_grad;
    
```

(그림 6) 장해데이터의 축적을 위한 자료구조
(Fig. 6) Data structure for error data accumulation

5. 적용 사례 및 효과

5.1 지원 시스템 적용 사례

본 장에서는 장해관리 지원 시스템을 구현단계 산출물에 대한 평가에 적용한 사례를 소개하기로 한다. 평가는 K사의 종합정보시스템에 대한 프로그램 기술서를 대상으로 하였다.

5.1.1 종합정보시스템의 구성 업무

K사의 종합정보시스템은 정보통신, 무역정보, 경영정보, 사무자동화 업무로 구성되어 있다. 이들 업무가 다시 부수적인 업무들로 구성되어 있으며 구성 내용은 아래와 같다.

- 정보통신
E-MAIL, 문서정보(전자결재 & 문서관리)
- 무역정보
무역정보구축(VAN), 고객정보
- 경영정보
예산관리, 회계관리, 고정자산관리, 급여관리, 복지후생관리, 사업관리, 경영자정보
- 사무자동화
자료관리(지침서, 규정집 등), 조직도관리

5.1.2 장해정보의 축적

통합품질관리도구를 이용하여 생명주기 각 단계별 산출물에 대한 품질평가가 끝나면 정해진 합격수준에 이르지 못한 품질특성, 품질부특성, 내부특성에 대한 항목을 알 수 있다. <표 6>은 구현단계 산출물에 대한 품질평가를 위해 ESCORT-C(Coding)를 이용하여 품질평가한 결과 중 내부특성의 평가결과를 나타내고 있으며 합격선으로 설정한 80점에 이르지 못한 내부특성을 쉽게 파악할 수 있으며 합격선에 이르지

<표 6> 업무별 내부특성의 평가 결과
<Table 6> Evaluation result of internal characteristics

내부 특성	E-Mail	무역 정보	예산 관리	회계 관리	고정 자산	급여 관리	복지 후생	사업 관리	조직 관리
완전성	85.6	73.6	73.2	74.8	86.4	76.8	85.6	76.4	86.0
추적 가능성	73.0	83.0	86.0	73.0	77.0	77.0	76.0	83.0	83.0
일관성	81.0	69.8	72.2	81.0	69.8	80.2	71.4	72.2	81.6
자기 기술성	69.0	87.8	64.8	66.4	88.4	82.8	67.8	66.6	66.4
무모순성	72.0	75.0	80.0	70.0	75.0	75.0	84.0	87.0	78.0
통신질차 공통성	81.0	81.0	70.0	81.0	86.0	89.0	69.0	68.0	66.0
액세스 가능성	77.0	76.0	87.0	78.5	83.5	77.0	86.5	86.5	78.5
액세스 제어성	66.5	67.9	66.5	84.9	67.9	62.6	85.3	66.5	71.4
액세스 감시성	63.0	82.8	62.0	64.6	64.6	68.2	68.2	61.8	87.2
견고성	84.2	67.0	86.2	67.0	82.8	82.8	85.8	80.0	83.0
무결성	67.9	65.1	67.9	87.9	67.0	67.9	60.0	84.9	64.9
모듈성	67.9	84.4	77.2	77.0	83.6	75.2	81.4	83.6	74.6
단순성	72.0	70.0	72.4	69.6	68.4	76.0	66.0	72.4	86.0
계속성	80.0	70.0	60.0	80.0	64.0	60.0	70.0	60.0	64.0
통일성	78.8	76.0	86.4	76.4	85.2	82.4	86.4	74.0	76.4
표현성	78.9	88.2	71.9	89.3	73.9	80.9	69.8	70.9	81.8
계속성	85.8	71.2	73.0	73.0	77.0	77.6	71.2	84.8	86.4
설명성	66.0	64.0	84.0	62.0	68.0	82.0	62.0	62.0	66.0
주목성	72.6	81.2	74.2	74.2	82.4	72.4	89.4	72.4	73.2
간결성	84.0	71.2	72.8	66.8	65.2	74.0	74.0	84.0	81.2
선택성	67.0	80.0	83.0	63.0	83.0	73.0	80.0	73.0	70.0
평 균	74.9	75.5	74.8	74.3	76.1	75.8	75.7	74.8	76.5

못한 내부특성의 값이 합격선에 이르지 못한 원인이 되는 요소데이터를 장해 데이터로 취급한다.

5.1.3 장해데이터의 선정

품질평가 결과 합격선에 이르지 못했거나 합격선을 넘었어도 품질평가의 최소 단위인 요소데이터의 경우에는 평가결과가 좋지 않을 수 있다. 이러한 요소 데이터들을 장해데이터로 선정하여 요소데이터별로 평가가 반복될 때마다 평가결과가 축적된다.

<표 7> 장해데이터와 축적된 점수
<Table 7> Error data and accumulated grade

장해 데이터	점 수					
	65.2	72.1	67.5	69.4	61.5	...
총실효 구현된 기능의 수	65.2	72.1	67.5	69.4	61.5	...
실제서에서 추적할 수 있는 기능의 수	63.5	68.5	72.3	77.5	65.7	...
표준규약을 일관성있게 지원하는 기능의 수	62.7	67.6	77.2	71.2	68.5	...
오류메시지를 일관성있게 제공하는 기능의 수	61.3	71.2	66.7	72.6	72.4	...
표준/지침을 준수하는 주석의 수	69.2	71.8	66.8	69.8	72.3	...
	:	:	:	:	:	:

장해데이터의 축적결과를 통해 일반적으로 개발자들이 각 생명주기 단계의 산출물들을 작성하는 과정에서 미흡한 부분들을 파악할 수 있으며 프로젝트 생명주기 각 단계가 시작되기 전에 개발자들에게 제시하는 품질평가 기준과 평가 지침서를 작성할 때 중점적으로 지침을 줄 수 있을 것이다.

5.1.4 문제점 분석과 개선방안

각 생명주기 단계로부터 전달받은 장해데이터에 대해 구체적인 문제점을 제시하고 이들 문제점들에 대한 개선방안을 작성하여 개발자에게 제시한다. <표 8>에 장해데이터의 예와 문제점 및 개선방안의 일부를 기술하였다.

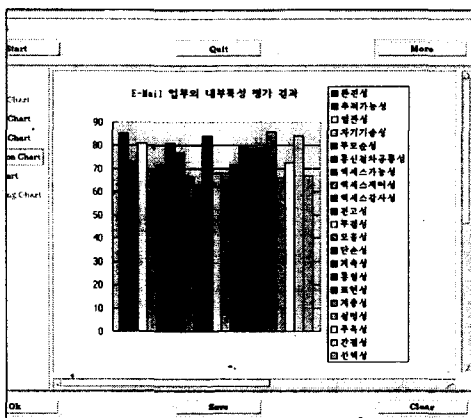
5.1.5 결과의 가시화

〈표 8〉 장해데이터의 문제점 및 개선 방안
(Table 8) Problem and enhancement way of error data

장해 데이터	문제점	개선지침의 예
충실히 구현된 기능의 수	구현된 모든 기능 중 충실히 구현된 기능이 미흡하다.	구현된 기능중 사용자의 요구에 부적합한 기능에 대해 보완한다.
설계서에서 추적할 수 있는 기능의 수	설계서에 정의된 기능으로부터 구현 사항을 추적하기 어렵다.	생명주기 단계의 관련 산출물들을 참조하기 쉽도록 일관성있게 순서를 부여한다.
표준규약을 일관성있게 지원하는 기능의 수	코딩에 관련된 표준규약을 정확히 준수하고 있지 못하다.	구현된 각 모듈별로 코딩 규약에 위배된 사항을 수정한다.
오류 메시지를 일관성있게 제공하는 기능의 수	오류발생시 오류메시지의 형식이 통일되어 있지 못하다.	오류메시지의 제공형식이나 용어 등을 통일한다.
표준/지침을 준수하는 주식의 수	주식의 위치, 주식량 등에 대한 지침을 명확히 준수하고 있지 못하다.	소스 코드에서 주식 위치를 통일하고 충분한 내용을 기술한다.

내부특성의 평가결과와 장해 데이터에 대한 축적된 점수를 가시화하여 그래프로 나타낼 수 있다.

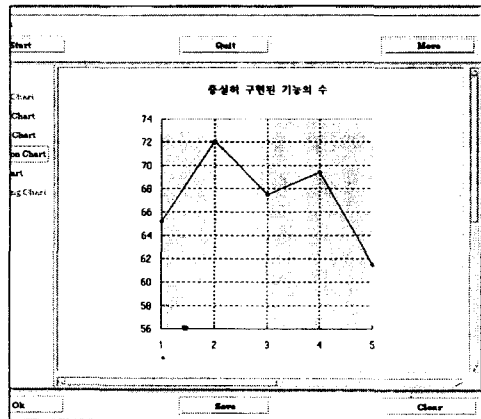
(그림 7)은 E-Mail 업무에 대한 내부특성의 품질평가 결과를 막대그래프로 나타내고 있다. 그림에서 추적가능성 73.0, 자기기술성 69.0, 무모순성 72.0, 액세스가능성 77.0 등으로 합격선을 80점으로 했을 때 내부특성의 값이 합격선에 이르지 못한 업무와 다른 업무



(그림 7) 결과에 대한 가시화의 예(막대 그래프)
(Fig. 7) Examples of visualization for result(bar graph)

무에 비해 상대적으로 취약한 점수를 획득한 업무를 쉽게 파악할 수 있다.

(그림 8)은 '충실히 구현된 기능수'에 대한 장해데이터의 축적값을 축적된 순서대로 꺾은선 그래프를 이용하여 나타내고 있다. '충실히 구현된 기능수'에 대한 장해데이터의 값은 65.2, 72.1, 67.5, 69.4, 61.5의 순으로 축적되어 있으며 평가 결과의 차이를 쉽게 파악할 수 있다.



(그림 8) 결과에 대한 가시화의 예(꺾은선 그래프)
(Fig. 8) Examples of visualization for result(line graph)

5.2 지원기능의 효과

실제의 프로젝트에 적용한 결과 다음과 같은 운영면에서의 효과를 얻을 수 있었다.

5.2.1 실제 프로젝트에 적용한 효과

(1) 다면적 평가와 종합적 평가를 했을 때의 효과
종래에는 출하전의 각 프로젝트 및 기종과 분야에서의 품질평가와 분석을 위한 것을 카탈로그 프로젝트화하여 관리자료를 검색하고 출력하는 조작을 간편하게 하려고 했었다. 그렇지만 출하후 현장의 적용시험에서의 장해 데이터와 현장 정보를 합하여 일괄적으로 보는 다면적 평가와 종합 평가를 실시해본다 각 데이터베이스의 링크가 용이하지 않았다. 이번 개발에 의해서 각 데이터베이스와의 링크를 실현하고 출하전부터 출하후까지를 일괄적으로 보는 다면적 평가와 종합평가의 관리자료를 검색출력 조작을 카탈로그화함으로써 수집계로 분류하던 작업이 소멸되었다.

(2) LAN으로 연결된 워크스테이션을 이용한 효과 종래에는 각 프로젝트 자체에서 임의로 장해 데이터를 추출하여 평가와 분석을 하고 싶어도 단말기가 없었기 때문에 대형 서비스에 의해 얻어지는 정형장표 대로의 평가 분석에 머물고 있었다.

이번의 개발에 의해 근거리에 배치되어 있는 워크스테이션에서

- ① 목적에 부합한 집계와 검색을 할 수 있기 때문에 데이터를 여러 각도에서 잘 볼 수 있게 되었다. (다면적 평가)
- ② 필요한 때에 정확한 정보를 시기적절하게 추출할 수 있도록 하였다.
- ③ 다른 프로젝트(유사 프로젝트, 이전 버전의 프로젝트)에서 축적된 데이터를 활용(시험계획, 비교 분석)할 수 있도록 하였다.

(3) 개발부서와 현장의 적용 시험 사이의 장해 데이터 링크에 의한 효과

종래에는 개발부서시험과 현장적용시험간에 시험 환경에 있던 지원 도구가 사용되고 있는 서로 다른 데이터베이스 구조때문에 현지시험 장해데이터를 공장시험 장해데이터로부터의 연속된 데이터로 볼 수가 없었다. 이번의 개발에 의해 연속 데이터로 보는 것이 가능하고 일괄적으로 총합평가하는 것이 가능해졌다.

(4) 품질목표 및 실적 관리기능 지원에 의한 효과

종래에는 개발부서와 협력회사간의 대비를 실적치만에 의한 관리장표로 행하고 있었기 때문에 목표에 대해서 어떻게 진행되고 있는가를 파악하기 곤란하였으며 협력회사에 대한 관리도 용이하지 못했다. 이번의 개발에 의해 실적값과 목표를 대비할 수 있는 관리장표로 협력회사간의 대비가 행해지도록 하였고, 협력회사에 서로의 품질목표달성이 가능하도록 하였다.

5.2.2 운용면에서의 효과

지금까지는 장해 데이터를 기종과 분야라고 하는 단위의 데이터베이스로 운용하고 있었지만 해마다 생산량이 증가함에 따라 한 데이터베이스의 용량이 커지고 데이터 검색 성능이 현저히 저하하고 있다. 이번 프로젝트 단위의 데이터베이스를 구축함으로써

프로젝트 단위의 데이터베이스 구축이 용이하고 검색성능이 대폭 향상되어 운용효율이 증대되었다.

6. 결 론

소프트웨어의 품질향상은 소프트웨어 생명주기 전 단계에 걸친 품질평가를 통해 문제점을 발견하여 개발자에게 피드백하는 과정을 반복함으로써 이루어질 수 있다. 소프트웨어 생명주기의 각 단계는 품질특성에 따른 각각의 매트릭스와 이들 매트릭스를 구현한 각각의 툴킷으로 구성되어 있으며 자동적으로 품질평가 결과를 산출할 수 있다. 비록 생명주기 각 단계별로 매트릭스들이 독자적으로 구축되어 있지만 매트릭스들은 ISO/IEC 9126에 기초한 품질특성을 바탕으로 품질부특성과 품질내부특성의 체계에 따라 개발하였으므로 생명주기에서 다른 단계간에 관련된 매트릭스가 존재하며 이들에 대한 품질평가 결과의 통합집계, 분석이 필요하게 될 것이며 이러한 기능을 담당하는 것이 본 논문에서 개발한 장해관리 지원도구이다. 지금까지 이러한 장해관리 중심의 소프트웨어 품질관리 지원시스템에 대해 소개하였다. 본 장해관리 중심 소프트웨어 품질관리 지원시스템은 현재 개발중인 통합품질관리시스템 ESCORT의 기능을 지원할 수 있는 도구로 통합하였다. ESCORT는 요구명세서, 분석·설계·구현·유지보수 단계의 산출물들의 품질을 평가하여 개발자에게 피드백함으로써 생명주기 각 단계의 고품질화를 목표로 하는 통합품질관리 도구이며 이들 도구들로부터 전달되는 장해데이터를 통합하여 처리할 수 있는 도구로서 본 장해관리중심 소프트웨어 품질관리 지원시스템과의 통합을 통해 품질평가 결과에 따른 장해데이터의 효율적인 관리와 원활한 유지보수에 큰 도움을 줄 것으로 기대한다. 본 장해관리중심 소프트웨어 품질관리 지원시스템의 운용에 의해서 소프트웨어 품질관리 업무의 효율화 및 질적인 향상이라는 목표에 도움이 되었고 소프트웨어 개발에 따른 유지보수 비용의 절감과 고품질화에 도움을 줄 수 있다고 본다.

앞으로의 연구과제로서 첫째, 아직도 불충분하다고 생각되는 보급, 정착활동과 둘째, 축적 데이터의 효율적인 활용(관리기준에의 반영)과 셋째, 품질평가와 진단제도의 향상을 과제로하여 본 시스템을 보다 실

용적인 시스템으로 보완하는 것이다.

참 고 문 헌

[1] Boehm, B. W. et al, "Quantitative Evaluation of Software Quality", Proc. of 2nd ICSE, Vol. 2, 1976.

[2] DeMarco, T, "Controlling Software Projects: Management, Measurement and Estimation", Englewood Cliffs, Prentice Hall, 1982.

[3] Walters, G. F., and McCall, J. A., "Software Quality Metrics For Life Cycle Cost Reduction", IEEE Transaction on Reliability, Vol. R-28, No. 3, pp. 212-220, IEEE, 1979.

[4] Perlis, A., Sayward, F. and Shaw M., "Software Metrics: An Analysis and Evaluation", The MIT Press, 1981.

[5] Thadhani, A. J., "Factors Affecting Programmer Productivity during Application Development", IBM Systems Journal, Vol. 23, No. 1, pp. 19-35, IBM, 1984.

[6] Li, H. F. and Cheung, W. K., "An Empirical Study of Software Metrics", IEEE Transaction on SE., Vol. SE-13, No. 13, pp. 697-708, June 1987.

[7] 森口繁一, "ソフトウェア品質管理ガイドブック", 日本規格協會, 1990.

[8] 菅野文友, "ソフトウェアの品質管理", 日科技連出版社, 1986.

[9] 石井康雄, "ソフトウェアの検査と品質保証", 日科技連出版社, 1986.

[10] IEEE, "手引ソフトウェア品質保證計劃", ANSI/IEEE, Std983, 1986.

[11] 小林 博, "基本ソフトウェア開發プロジェクトにおけるソフトウェアの品質管理の事例", 情報處理學會第30回全國大會 1985.

[12] 양해술, "KOTRA 종합정보 시스템의 품질보증과 품질평가", 한국컴퓨터(주), 품질평가보고서, 1995.

[13] 양해술, 이용근, "소프트웨어 요구명세서 품질평가 지원도구의 설계 및 구현", 한국정보과학회논문지(C), 제3권 제2호, pp. 152-163, 1997. 4.

[14] 양해술, "한진해운 재무정보시스템의 품질감리 및 평가", 구현단계 품질감리 및 평가보고서, 한국소프트웨어품질연구소, 1997. 5.

[15] 양해술, 이하용, "설계단계에서의 품질평가 도구(ESCORT-D)의 설계 및 구현", 한국정보과학회 논문지(C), 제3권 제3호, pp. 262-274, 1997. 6.

[16] 양해술, "소프트웨어 품질측정 기록 및 지원 툴킷의 개발", 시스템공학연구소 위탁연구과제, 제1차년도 중간보고서, 1997. 6.

[17] 양해술, "분석단계 산출물에 대한 품질평가툴킷의 설계 및 구현", 한국정보처리학회논문지, 제4권 제7호, 1997. 7.

[18] 양해술, "한진해운 신정보(영업 및 물류) 시스템의 품질감리 및 평가", 제1차 품질감리보고서, 한국소프트웨어 품질연구소, 1997. 7.

[19] 양해술, "품질관리방법론 및 지원도구의 개발", STEP2000 위탁연구과제, 3차년도 최종보고서, 과학기술처, 1997. 7.

[20] 양해술, "한남대학교 종합정보시스템의 품질감리 및 평가", 설계단계 품질감리 및 평가보고서, 한국소프트웨어품질연구소, 1997. 7.



양 해 술

1975년 홍익대학교 공과대학 전 기공학과 졸업(학사)

1878년 성균관대학교 정보처리학과 정보처리 전공(석사)

1991년 일본 오사카대학교 기초 공학부 정보공학과 소프트웨어공학 전공(공학 박사)

1975년~1979년 육군중앙경리단 전자계산실 시스템 분석장교 근무

1986년~1987년 일본 오사카대학교 객원연구원

1980년~1995년 강원대학교 전자계산학과 교수

1993년~1994년 한국정보과학회 학회지 편집부위원장

1994년~1995년 한국정보처리학회 논문지편집위원장

1994년~현재 한국산업표준원(KISI) 이사

1995년~현재 한국소프트웨어품질연구소(INSQ) 소장

관심분야: 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 품질컨설팅, OOA/OOD/OOP, CASE, SI), 소프트웨어 프로젝트관리



이 하 용

- 1993년 강원대학교 전자계산학과 졸업(학사)
- 1995년 강원대학교 대학원 전자계산학과 소프트웨어공학 전공(이학석사)
- 1996년~현재 경희대학교, 경원대학교 공과대학

전자계산공학과 강사

1995년~현재 한국소프트웨어품질연구소(INSQ) 선임연구원

관심분야: 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 객체지향 프로그래밍, 객체지향 분석과 설계, CASE)



안 유 환

- 1984년 서울대학교 산업공학과 졸업(학사)
- 1986년 한국과학기술원 경영과학과 졸업(석사)
- 1990년~현재 한국과학기술원 경영과학과 박사과정
- 1986년~현재 시스템공학연구소 선임연구원

1996년~현재 시스템공학연구소 품질관리연구실장

관심분야: S/W 품질보증, S/W 품질평가, S/W 프로세스 관리 및 자동화, S/W 프로세스 개선, 개발/관리 방법론, Client/Server 시스템 개발 방법론