

MBASE 기반의 결함 및 진척도 관리를 위한 OT설계

신승민⁰ 이경환

중앙대학교 컴퓨터공학부 소프트웨어공학 연구실
{smshin⁰, kwlee}@object.cse.cau.ac.kr

A Study on Opportunity Tree Framework to Manage the Software Defects and Monitoring

Sol-min Shin⁰, Kyung-wan Lee

Software Engineering Lab. School of Computer Science & Engineering, Chung-Ang University

요 약

본 논문은 소프트웨어 공정 시 발생하는 결함을 찾아내고 원인을 분석하여 제거하는 동시에 제품 개발의 진척도를 관리하여 결함을 미리 예측하고 재 발생을 예방한다. 이를 통해 소프트웨어 생산성을 높이며 개발주기를 줄이는 동시에 높은 수준의 제품출하를 이루어 타 회사보다 앞선 경쟁력과 Speed 경영을 이룰 수 있는 방법을 제시한다. 이를 위해 MBASE 기반의 소프트웨어 프로젝트 관리 모형과 함께 OT를 설계하고 구현하여 제품의 결함을 줄이고 공정의 진행사항을 쉽게 파악할 수 있게 구현하였다.

1. 서 론

본 논문은 신뢰성 있는 소프트웨어를 생산하기 위하여 결함을 식별하고 연관성을 분석하여 소프트웨어 개발의 진척도를 얻을 수 있는 Opportunity Tree의 설계를 제안한다. 급속히 줄어들고 있는 제품생산의 속도는 경쟁력 있는 기업의 Speed 경영에 필수적인 요소로 자리잡았다. 하지만, 제품의 빠른 공정과 더불어 여러 요소에서 발생하는 결함의 증가는 또 다른 소프트웨어 개발의 문제로 자리잡고 있으며, 신뢰성 있는 소프트웨어를 개발하기 위해서 소프트웨어의 개발 과정에 발생하는 결함을 찾아내고 이를 관리하는 것이 중요한 요인이 된다. 또한, 이를 통해 제품의 개발공정에 대한 진척도 관리를 수행하여 높은 수준의 소프트웨어를 빠른 시간 내에 개발할 수 있다.

2. 기반연구

본 연구에서는 소프트웨어 프로젝트 관리모형에 대한 기반으로 MBASE(Model Based Architecting and Software Engineering)와 OT(Opportunity Tree)를 제안하였다.

2.1 MBASE

MBASE는 소프트웨어 시스템을 4가지 모델을 기반으로 구축하고, 이들 모델 간의 충돌을 해결함으로써 프로젝트를 성공으로 이끌기 위한 프레임워크이다. 그림1은 프로젝트의 성공 모델 (Success Model), 프로덕트 모델 Product Model), 프로세스 모델(Process Model) 그리고 프로퍼티 모델(Property Model)이 일관되고 서로 잘 통합될 수 있도록 하기 위해 MBASE 접근방법에서 사용되고 있는 전체 프레임워크의 그림이다. 그림의 상단은 다양한 모델을 나타내고 있으며, 그것은 우선순위와 일관성이 가장 먼저 고려되어야 한다. 성공모델은 프로덕트 모델에 대한 많은 요소와 프로세스 모델 그리고 다양한 프로퍼티 모델을 정의하고 결정한다. 따라서 성공 모델에 의해 나머지 세가지 모델이 변경될 수 있다. 비용-일정 예측 모델은 어떠한 프로젝트가 성공 모델을 위해 충분한지를 결정하기 위해, 다양한 프로덕트 특성(컴포넌트 크기, 재사용, 프로덕트 복잡성), 프로세스 특성(개발 인원의 능력과 경험, 도구의 지원, 프로세스 성숙도), 그리고 프로퍼티 특성 (요구되는 신뢰성, 비용 제약사항)과 관련될 수 있다.

그러므로 비용/일정 프로퍼티 모델은 시스템의 프로덕트, 프로세스, 성공 모델간의 일관성 분석 및 평가를 위해 사용될 수 있다. 또한, 성공 모델은 프로세스 모델과 프로덕트 모델을 만들어 낼 수 있다. IKIWISI(I'll Know it when I see it) 성공 모델은 프로토타이핑 및 진화적 개발 프로세스 모델을 초기에 설정할 수 있게 하고, 프로덕트 요소와 프로퍼티 레벨에 대한 대부분을 프로세스에 의해 결정되도록 한다. 프로덕트 라인 개발에 초점이 맞추어진 성공 모델은 비즈니스 분석을 수행하기 위해 부수적으로 조사될 수 있는 프로세스 모델 및 프로퍼티 모델과 함께 도메인 분석 및 프로덕트 라인 아키텍처를 포함하고 있는 프로덕트 모델을 강조하게 된다. 프로젝트 성공 모델을 어떻게 만드느냐에 따라 다양한 프로덕트/프로세스/프로퍼티 모델이 설정될 수 있으며, 그것 간의 충돌을 해결하고 어떻게 잘 통합하느냐에 따라 프로젝트의 성과가 결정된다.

2.2 Opportunity Tree

실제 프로젝트 수행 시에, 관리자들은 결함을 관리한다. 그리고 개발일정에 맞추기 위하여 많은 수의 결함들을 완전히 조치를 취하지 못하고 다음 개발 단계로 진행되기도 한다. 이와 같은 현상은 많은 프로젝트에서 발생하고 있다. 그 이유는 관리자의 경험부족과 일정상의 여유가 없기 때문인 경우가 대다수이다. Opportunity Tree(이하 OT)는 해결하려는 목표를 설정하게 되고, 전체목표를 해결하기 위하여 요구되는 하위 목표를 결정하게 된다. 즉, 하위목표를 달성하게 되면 상위 목표가 달성될 수 있게 된다. 그리고 목표 설정과 함께 목표를 해결하기 위하여 요구되는 문서와 전문가의 노하우를 OT화 하여 사용자에게 지침을 제공하고자 하는 것이 OT의 목적이 된다. 따라서, 본 논문에서는 결함관리와 함께 Life Cycle의 진척도 관리를 할 수 있도록 하는 것이 최종 목표가 된다. 이를 통해 개발과정의 발전 정도와 함께 개선점을 예측하고 예방할 수 있다.

3. 본 론

3.1 APM (Anchor Point Milestones)

MBASE의 4가지 모델간의 통합과 조화를 위해 두 가지의 이정표(Milestone)가 사용된다. 그림2는 두 가지 이정표 즉, LCO와 LCA에 대한 기준내용으로 올바른 프로젝트의 수행여부의 진행사항에 대한 진척도를 관리할 수 있다.

LCO, LCA 내용

이전과 동일	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
요구 사항 분석 (OCB)	<ul style="list-style-type: none"> • 고객 요구사항의 시나리오 및 범위 • 운영적/비 운영적 시나리오 • 운영 및 유지보수 시나리오 	<ul style="list-style-type: none"> • 구조적 요구 사항 상세화 • 운영적/비 운영적 상세화
요구사항 분석 및 요구사항 관리 (SRD)	<ul style="list-style-type: none"> • 고객 요구사항의 기준, 인터페이스, 조건 • Stakeholder에 의해 	<ul style="list-style-type: none"> • 기능, 인터페이스, 조건, 구조에 상세화 • Stakeholder의 요구사항 및 고객과의 논의
요구사항 분석 및 요구사항 관리 (SSAD)	<ul style="list-style-type: none"> • 고객 요구사항의 구현을 위한 요구사항에 대한 상세화 • 고객 요구사항에 대한 상세화 • 고객 요구사항에 대한 상세화 	<ul style="list-style-type: none"> • 고객 요구사항의 상세화 • 고객 요구사항에 대한 상세화
요구사항 분석 및 요구사항 관리 (SAD)	<ul style="list-style-type: none"> • Stakeholder에 의해 • 고객 요구사항에 대한 상세화 • 고객 요구사항에 대한 상세화 	<ul style="list-style-type: none"> • LCO의 요구사항을 WWWWHH 상세화 • IOC, Initial Operational Concept, 이진 파일 (문서)
요구사항 분석 및 요구사항 관리 (SRD)	<ul style="list-style-type: none"> • 고객 요구사항에 대한 상세화 • 고객 요구사항에 대한 상세화 	<ul style="list-style-type: none"> • 고객 요구사항에 대한 상세화 • 고객 요구사항에 대한 상세화

그림 1. LCO, LCA의 내용

첫 번째 Milestone은 LCO(Life Cycle Objectives)이며, 이것은 적어도 하나의 아키텍처 생성 단계를 통해 생성된 비즈니스 결과물에 대한 기초를 검증한다. 즉, 이것은 어떠한 프로젝트가 주어진 예산과 일정 내에서 구현되고, 핵심 Stakeholder의 승리조건(Win Condition)을 만족시키고, 실행 가능한 투자 비즈니스 사례를 생성할 수 있음을 보여준다. 즉 프로젝트에 대한 타당성을 보여줄 수 있는 적어도 하나의 시스템 아키텍처와 COTS/재사용 컴포넌트의 선택이 존재한다는 것을 검증해준다.

두 번째 Milestone은 LCA(Life Cycle Architecture)이며, 이것은 프로덕트 개발에 대한 견고함(결정된)결과물에 대한 기초를 검증한다. AT&T/Lucent Architecture Review Board기법은 LCO/LCA를 포함시킨 관리 검증 접근 방법으로 유명하며, Rational사의 Objector Process 즉, RUP에서도 LCO와 LCA를 핵심 Milestone으로 사용하고 있다.

3.2 Model 충돌

소프트웨어 프로덕트 간의 모델 충돌에 대한 개념은 여러 형태로 나타난다. 예를 들어, 프로젝트 입력과 출력 사이의 구조적 충돌이나, 프로덕트의 요구사항, 설계, 구현 간의 추적성 충돌, 상업적인 COTS 컴포넌트 혹은 재사용 소프트웨어 컴포넌트를 통합하는데 있어서의 아키텍처 충돌 등이 존재한다. 또한 시스템의 프로덕트 모델과 개발 프로세스 모델 간에도 충돌이 발생할 수 있다. 예를 들면, 프로덕트 COTS 기반으로 개발하고, 개발 프로세스는 폭포수(Waterfall)모델을 쓸 경우이다. 프로덕트 모델과 프로퍼티 모델 사이에서도 충돌이 발생할 수 있으며, 또한 프로덕트 모델과 성공 모델 사이에서도 충돌이 발생할 수 있다. 그림 3은 프로덕트, 프로세스, 성공, 프로퍼티 모델들의 통합과정에서 발생할 수 있는 심각한 모델 충돌을 보여주고 있는 모델-충돌 매트릭스이다. 이를 통해 각 모델간의 충돌을 피하고 OT설계를 통해 그 해결점을 찾을 수 있는 경로(Route)의 제시한다.

3.3 MBASE 프레임워크

MBASE 접근 방법에 대한 전체적인 프레임워크를 제공한다. 어떠한 시스템의 특징을 위한 주된 유도자는 그 시스템의 핵심 Stakeholder가 된다. 핵심 Stakeholder란 일반적으로 시스템 개발 프로젝트에 참여하고 이를 수행하는 사람으로서, 개발자(Developer), 사용자(User), 고객(Customer), 유지보수자(Maintainer) 등이 된다. 또한 Stakeholder에는 마케팅 관리자, 운영자 등도 포함될 수 있다. 먼저, 이러한 Stakeholder들이 가장 관심을 갖고 있는 사항들에 대해서 우선순위를 결정하고, 이들을 성공 모델 내에서 정의되도록 한다. 즉, 시스템에 대한 성공 모델을 Stakeholder들의 성공조건, 운영적 효율성 모델, IKWISI등을 반영하게 된다.

도메인/환경 모델을 기반으로 개념적인 프로덕트 모델을 생성하고, 개념적인 프로덕트 모델에 따라 적절한 프로세스 모델을 결정한다. 이렇게 결정된 프로세스 모델 또한 프로퍼티 모델에 반영이 되어 전체 시스템 비용에 영향을 미치게 된다. 도메인 모델은 여러 번 반복 과정을 거쳐 완전한 프로덕트 모델로 완성된다.

4. OT프레임워크 설계

4.1 영역분석

OT 프레임워크의 설계 시, 선행되어야 할 작업으로는 OT 프레임워크를 적용할 도메인 선정과 설계될 상세한 세부주제이다. 본 논문에서는 OT 프레임워크 설계 시 디지털 시스템 분야를 도메인으로 선정하였고, 상세한 OT 프레임워크의 내용으로 회사에서의 위험요소로 전이될 수 있는 결함을 대상으로 하였다. 결함을 대상으로 한 이유는 결함을 관리하지 않고서는 산출물에 대한 품질을 보장할 수 없게 된다. 또한 결함이 위험요소로 전이될 때, 비용, 스케줄, 품질관점에서 관리를 하지 않으면 산출물 자체에 문제점이 내재되기 때문이다. 결함관리 OT프레임워크를 설계하기 위해서 도메인 분석과 선정, 사용자의 요구사항 분석, 운영 시나리오, 시스템 영역과 환경을 설정한다. 도메인 분석단계의 요구사항 분석 및 분석작업은 중요도면에서 전체 프로젝트에서 가장 많은 비중을 차지하게 된다. 그 이유는 요구사항의 추출 및 분석의 결함이 코딩 이후의 단계에서까지 결함을 유발시킬 수 있기 때문이다. 분석가는 사용자의 요구사항을 도메인 전문가의 지식을 기반으로 분석을 한 후에 분석한 결과를 문서로 저장한다. 도메인 선정과 사용자 요구사항의 분석이 끝나면 이를 기반으로 OT 프레임워크 운영 시나리오를 추출하게 된다.

4.2 시스템 환경 분석

시스템 환경 설정 단계에서는 시스템 구성인자간의 연관성을 추출하며 전체적인 시스템의 구성인자를 식별하게 된다. 또한 시스템의 연관성 추출을 기반으로 기능적인 요구 사항의 충돌 및 실행가능성을 판별할 수 있다. 그러므로 시스템 구조도에서는 시스템 요소간의 관계를 명시함으로써, 연관성을 쉽게 인식할 수 있다. 시스템 환경을 식별하여 구성인자를 파악하는 과정에서 추가되어야 할 사항으로는 스테이크 홀더를 식별하는 것이다. 스테이크 홀더는 시스템과 이후의 요구사항 추출 및 분석에서 중요한 역할을 하게 된다. 추출된 스테이크 홀더는 도메인 지식을 시스템화 하려는 관점의 기능적 요구사항과 시스템의 비 기능적 요구사항에 연관된 인력자원을 주축적으로 추출한다.

스테이크 홀더 명	설명
설문서 작성자	OT 프레임워크를 개발하기 위하여 설문서 작성하여 초기 요구사항을 조사하는 사람
개발자	OT 프레임워크를 개발하는 사람
결함관리 OT 담당자 및 종료 최종 담당자	OT 프레임워크 내용이 클라이언트에게 전달되기 이전에 발생하는 문제를 담당하는 사람
개발과정 관리 담당자	개발과정의 진척도를 검사하고 관리하는 사람
클라이언트	OT 프레임워크 개발을 요구하는 사람
사용자	원하는 정보를 얻기 위하여 탐색기능을 이용하는 사람
정보주거자	OT 프레임워크에 추가적으로 발생하는 정보를 추가하는 사람
관리자	OT 프레임워크의 계층을 정렬하고 시스템의 접근을 관리하는 사람
도메인 전문가	OT 프레임워크와 연관된 지식을 가진 사람
소프트웨어 유지 보수 담당자	OT 프레임워크 내용이 클라이언트에게 전달되면서 발생하는 문제를 담당하는 사람

그림 2. 스테이크 홀더의 정의

4.3 요구사항 분석 및 추출

요구사항 분석은 도메인 지식을 기반으로 시스템 분석을 수행한 후에 결함관리 OT 프레임워크에서 설계하고자 하는 구체적인 내용을 파악하기 위한 단계이다. 요구사항 분석을 위한 설문서는 GQM방식에 의하여 설문서를 작성하였으며 이를 기반으로 사용자가 요구하는 기능을 파악하고자 하는 것이 목적이다. 우선 목표(Goal)는 결함관리라는 것을 수행하기 위해서 필요로 하는 하위목표가 몇인가를 결정해야만 한다. 따라서 서브목표는 전체 목표를 달성하기 위한 단위로 간주할 수 있다. 분석된 결함관리 OT항목 빈도에 의하면 여러 항목 중에서 마케팅 전략을 제외한 나머지 항목이 결함관리 OT프레임워크의 항목의 대상이 되었다. 우선 프로젝트 수행 시 발생하는 문제점들을 제시하여 빈도 분석도 병행하여 수행하였다. 분석 결과는 여러 문제점들 중에서 결함을 찾고 해결하기 위한 신뢰성 모델의 필요성과 불필요한 결함 자료 작성시 소요되는 일정과 비용의 낭비, 결함 발생 시 작업의 책임과 권한 등을 추출한다.

결함관리OT항목	내용
결함 자료 수집	결함 관리를 위하여 요구되는 자료를 수집하고 각 개발 단계별 결함을 찾아냄.
결함 연관성 분석	각 단계의 결함이 다른 단계의 결함과 연관성이 있는지를 찾아냄.
진전 결함 수정	중 진전 결함에 대해서 수정 가능한 시간 예측 가능.
결함 제거	찾아낸 결함을 제거함
결함 예방	이전의 결함 자료를 이용하여 새로운 프로젝트에서 결함을 예방함.
마케팅 전략	결함 잔존에도 불구하고 출시해야 하는 경우 마케팅 전략

그림 3. 결함 관리 OT 항목의 내용

5. 개발 공정의 진척도 관리요소

결함 자료 수집 및 식별을 위하여 설문서를 개발 단계별로 조사한다. 결함 자료 수집 및 목적은 회사에서 수행되는 프로젝트를 대상으로 하여 결함을 검출하기 위함이다. 또한, 검출된 결함은 각 결함간의 원인을 분석하는데 활용되고, 그 결함을 예방하고 원인을 제거하는 동시에 제품공정의 진척도를 확인하여 높은 성능의 제품을 빠른 시기에 출하할 수 있는 생명주지(Life Cycle)를 제시해준다. 이를 위해서 MBASE의 LCO와 LCA에 대한 이정표를 정의한 후 각 조건에 맞는 요소의 여부를 확인하여 개발 기간의 단축과 진척도 관리를 수행한다.

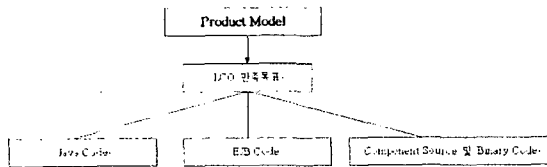


그림 4. Life Cycle Objectivity 항목

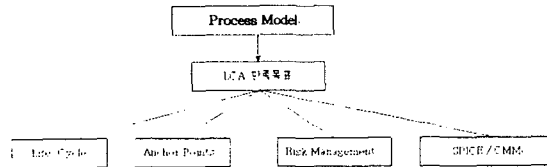


그림 5. Life Cycle Architecture 항목

6. OT 구현 및 평가

6.1 개요

각종 프로젝트를 수행 시 결함의 발생은 다양한 형태로 나타난다. 때로는 필요한 사항의 추가와 삭제 및 수정이 필요하다. 따라서 이를 모두 수용하여 실시간으로 수행할 수 있는 것은 웹 기반의 구현이 가장 적합하였다. 설문서를 통한 여러 결함을 1차적으로 수집하여 데이터를 입력한 후 결함에 대한 측정 및 분석을 수행하고 그에 따른 결함목록을 나타낸다. 그와 함께 결함예측과 예방에 대한 제안서를 보여주며 이를 통한 현재의 개발공정의 진척도를 그래픽과 트리 형태를 통해 보여준다. 결함 목록에 대한 수정과 해결된 결함에 대한 삭제 및 수정이 가능하다.

6.2 구조도

전체적인 구조도는 일반적인 웹으로 제공하는 서버릿/자바 서버 페이지를 응용했다. 전체적인 결함관리를 위한 OT 프레임 워크는 WAS 환경에서 가동되며 WAS환경에서 데이터베이스와 연동되게 된다. OT 프레임워크 엔진 역시 자바 서버 페이지에서 수행되도록 구현하였다. 대부분의 서버릿과 자바 서버페이지에서는 데이터베이스와의 트랜잭션을 자바서버페이지에서 수행하는 경우가 많다. 하지만 본 구현에서는 데이터베이스와의 트랜잭션을 한 단위의 클래스에 맡겨서 처리시켰다. 따라서, 각각의 테이블과의 트랜잭션을 담당하는 반복적인 커리가 많은 부분을 줄일 수 있고, 서버페이지에서 사용하는 선택, 업데이트, 추가, 삭제 등의 SQL코딩의 양을 줄일 수 있었다. 따라서 이를 통한 액세스 제어 목록은 데이터베이스 관리 시스템에 대한 접근제어 모듈을 추가할 지점에 대한 위치를 정확히 확보 할 수 있다.

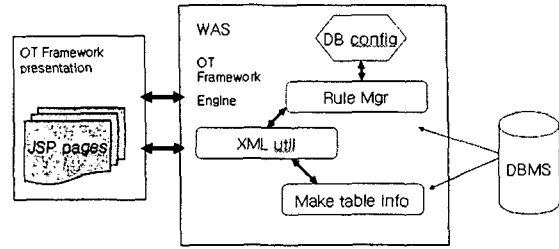


그림 6. OT 프레임워크 구조도

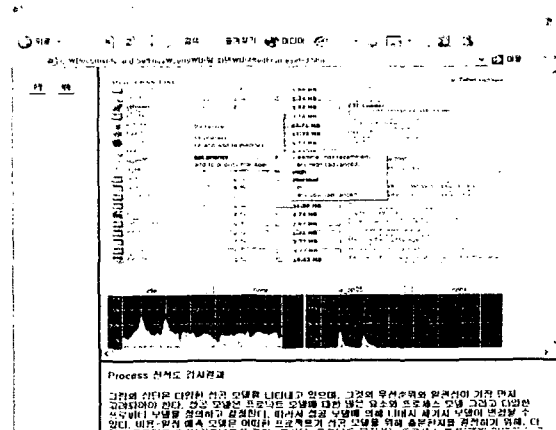


그림 7. 결함 항목 추가 및 원인 분석 화면

7. 결론 및 향후 연구

기존에 존재하는 결함 관리 방법은 결함 데이터를 수집한 후에 결함을 분석하는 단계의 것이다. 따라서, 결함 자료를 활용하여 결함을 제거하고 예방하는 단계의 분석 작업은 수행하지 않고 있다. 또한 개발공정에 대한 진척도에 대한 막연한 계획은 오히려 결함을 증가시키는 원인이 되기도 했다. 결함 관리와 진척도 관리를 하기 위해서는 개발초기에 미리 결함을 예방하고 발생된 결함을 제거함으로 보다 빠른 시간에 높은 수준의 제품을 완성할 수 있다. 이를 위해 프로젝트의 진척도에 대한 지속적인 관리가 필요하다. 결함관리를 위한 OT 프레임워크는 웹 서비스 기반으로 구현되었으며, 결함관리를 위한 활동과 해결책을 연구하여 지속적인 업데이트가 가능하도록 설계되었다. 또한 제품개발의 진척도를 쉽게 파악하여 현재의 개발성과와 방향을 제시함으로 제품출하시간을 줄이고 보다 경쟁력 있는 Speed 경영을 달성할 수 있는 측정 도구로 활용할 수 있다. 향후 연구로는 아직 네트워크 시스템의 안정화가 미비한 부분에 대한 노력과 함께 결과에 대한 좀더 가시적인 3차원 그래픽 활용을 필요로 하며, 국제적인 활용도를 높여서 여러 나라에서도 적용시킬 수 있는 다국적 언어지원에 대한 자동화를 지속적으로 연구할 계획이다.

8. 참고문헌

[1] 이은서, 이경환, "소프트웨어 결함 분석을 위한 트리거 설계", 정보처리학회논문지 D, 제10-D권 제4호 pp709, 2003. 8
 [2] Berry Boehm, Dan Port, Alexander Egyed and Marwan Abi-Antoun, "The MBASE Life Cycle Architecture Milestone Package" 1st Working International Conference on Software Architecture, 1999
 [3] Gaffney, John, "Some Models for Software Defect Analysis", Lockheed Martin, Nov 1996
 [4] B. Boehm, Software Engineering Economics, Prentice Hall, Englewood Cliffs, N.J., pp50-57, 1981