

시험구동형개발에 기초한 소프트웨어개발공정의 개선에 대한 연구

문일남, 장명민, 김현아

오늘날 품질높은 소프트웨어를 빨리 개발할수 있는 효과적인 개발공정을 확립하는것을 소프트웨어개발단위의 능력을 높이는데서 매우 절실한 문제로 나서고있다.

론문에서는 최근 소프트웨어산업에서 주목되고있는 시험구동형개발방식에 기초한 개발공정을 제안하였다.

1. 문제 설정

시험구동형개발(TDD: Test-Driven Development)은 코드를 구현하기에 앞서 시험경우들을 지정하는 방식으로 소프트웨어를 증분적으로 개발해나가는 개발기법이다.[2]

시험구동형개발의 작업흐름을 그림 1에 보여주었다.

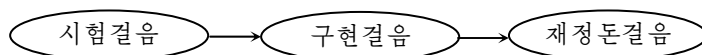


그림 1. 시험구동형개발의 작업흐름

시험결음에서는 검증하려는 대상에 대한 시험경우를 작성하고 구현결음에서는 시험경우가 통과되도록 코드를 구현하며 재정돈(refactoring)결음에서는 이미 시험에서 통과된 기능을 변경시키지 않으면서 코드구성을 세련시킨다.

시험구동형개발은 시험하려는 대상에 따라 단위TDD(UTDD: Unit TDD)와 접수TDD(ATDD: Acceptance TDD)로 나눈다.

소프트웨어개발단위들은 높은 품질의 소프트웨어를 제때에 내놓을수 있는 정연한 개발공정을 확립할 필요가 있다.

시험구동형개발과 관련한 선행연구[3]들에서는 시험구동형개발을 적용하여 소프트웨어개발을 성공시킨 사례자료들을 제안하였다. 그리고 학술적환경과 산업적환경에서 시험구동형개발을 적용할 때 그것이 개발제품의 품질과 개발성원들의 생산성에 어떤 영향을 미치는가에 대한 연구[2]가 많이 진행되었다. 그러나 시험구동형개발방식을 리용하여 개발공정을 어떻게 확립할것인가에 대한 문제는 현재 연구되지 않았다.

2. 단계적시험구동형개발공정

단계적시험구동형개발공정은 프로젝트를 여러개의 단계(반복)들로 나누고 매 단계의 목표를 시험구동형개발기법으로 달성해나가는 반복형개발공정이다.

1) 개발조구성

개발조에는 다음과 같은 역할들이 필요하다.

① 프로젝트관리자

- ② 분석자
- ③ 시험자
- ④ 구조설계자
- ⑤ 프로그램작성자

중요한것은 개발의 전과정에 걸쳐 수요자대표자가 개발에 적극 참여하는것이다. 또한 한 성원이 여러가지 역할을 담당할수도 있고 1개 역할에 여러명이 배치될수도 있다.

2) 개발생명주기

프로젝트를 개시기와 개발기, 결속기로 나눌수 있다.

개발의 생명주기를 그림 2에 보여주었다.

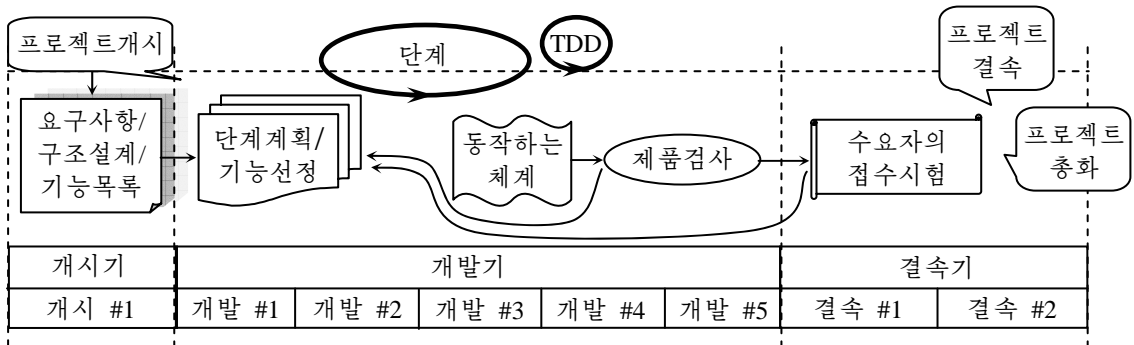


그림 2. 개발의 생명주기

개시기에는 대체로 요구사항들의 대부분을 수집하고 프로젝트의 총적인 범위를 총괄적으로 정의하며 해결이 가능하다는것을 담보할수 있는 충분한 구조설계를 구상하고 공수타산, 전체 일감타산을 비롯한 프로젝트계획을 작성한다.

개발기는 본격적인 개발이 진행되는 시기로서 대부분의 노력이 여기에 들어간다. 개발기는 여러개의 단계로 이루어지며 매 단계에서 TDD를 증분적으로 적용하여 수요자에게 넘겨줄수 있는 체계를 개발한다.

결속기는 개발된 체계를 본격적으로 현장에 전개하는 시기이다. 여기에는 수요자의 접수시험, 체계의 현장전개가 포함된다.

3) TDD에 기초한 단계실행

프로젝트의 매 단계에서는 시험구동형개발방식으로 체계를 증분적으로 개발해나간다. 공정흐름을 그림 3에 보여주었다.

- ① 이미 선정된 기능모임에서 구현하려는 기능을 선택한다.
- ② 선택된 기능에 대한 접수시험경우를 작성한다.
- ③ 접수시험경우에 대한 단위시험경우를 작성한다.(기능코드는 아직 없음.)
- ④ 단위시험경우를 실행시킨다. 물론 처음에는 실패하며 코드가 콤파일조차 되지 못할수도 있다.
- ⑤ 단위시험경우를 통과시키는데 꼭 충분한 코드를 구현한다.
- ⑥ 이 단위시험경우와 함께 현재의 접수시험경우에 대해 작성된 기타 모든 단위시험경우들을 실행시킨다. 한 시험경우라도 실패하면 결음 ⑤으로 되돌아간다.
- ⑦ 필요하면 코드를 재정돈한다. 재정돈에는 코드의 중복을 없애고 복잡성을 줄이며

3. 효과성평가

제안한 개발공정에 대한 평가를 두가지 측면에서 진행한다.

1) CMMI공정들의 만족범위

CMMI[1]2준위와 3준위공정과 실천들에 대해 제안한 공정은 다음과 같은 만족등급을 가질수 있다.(표 1)

표 1. 제안한 공정의 만족등급

CMMI공정	공정부류	만족등급	만족근거
요구사항관리(REQM)	기술공학	완전만족	수요자대표자에 의한 기능요구 사항을 명시
프로젝트계획수립(PP)	프로젝트관리	완전만족	매 단계에서 기능모임의 선정
프로젝트감독 및 통제(PMC)	프로젝트관리	크게 만족	시험경우들의 통과/실패를 쉽게 식별가능
공급자계약관리(SAM)	프로젝트관리	만족되지 않음.	
측정 및 분석(MA)	지원	부분적으로 만족	시험경우들의 통과 및 실패를 직관적으로 표현가능
공정 및 제품품질보증(PPQA)	프로젝트관리	완전만족	
구성관리(CM)	프로젝트관리	크게 만족	
요구사항개발(RD)	기술공학	완전만족	수요자대표자에 의한 기능모임을 선정, 접수시험경우의 작성
기술적구현(TS)	기술공학	완전만족	시험경우에 따르는 코드구현
제품통합(PI)	기술공학	완전만족	하루에 두번 제품통합 및 자동화된 회귀시험
검증(VER)	기술공학	완전만족	단위시험경우
확인(VAL)	기술공학	완전만족	접수시험경우 및 수요자에 의한 수동적인 접수시험
단위공정중점관리(OPF)	공정관리	등급없음.	
단위공정정의(OPD)	공정관리	등급없음.	
단위양성(OT)	공정관리	등급없음.	
위험관리(RSKM)	프로젝트관리	크게 만족	수요자대표자에 의한 모든 접수시험경우의 실행
의사결정분석 및 해결(DAR)	지원	크게 만족	구조설계자에 의한 결함검토
통합프로젝트관리(IPM)	프로젝트관리	등급없음.	

표 1에서 만족등급은 CMMI의 제3종평가를 진행하여 얻은 결과이다. 표 1에서 보여준것처럼 제안한 개발방식을 적용하는 CMMI공정들중에서 《기술공학》부류의 공정들은 완전히 만족될수 있다. 또한 《공정관리》부류를 제외한 나머지부류들에 대해서도 만족정도를 높일수 있다. 즉 제안한 개발공정 하나만으로도 소프트웨어개발단위들에 필요한 많은 공정들을 만족하게 확립할수 있다.

2) 개발제품의 품질개선

결함밀도(결함수/KLOC)를 척도로 하여 이미 결속된 5개의 소프트웨어개발프로젝트를 대상으로 외부품질에 대한 정량적인 분석을 진행한다. 이때 결함수는 최종적으로 수요자에게 넘겨준 다음 수요자측에서 진행한 접수시험에서 발견된 결함수와 운영기간에 사용자들로부터 제기된 결함수를 합한것을 리용한다.

품질평가를 표 2에 보여주었다.

표 2. 품질평가

방법 \ 지표	프로젝트	KLOC	결함수	결함밀도	결함밀도평균
제안한 방법	웹프로그래밍	12.53	38	3.03	2.13
	웹프로그래밍	2.45	3	1.22	
시험구동형개발을 적용하지 않는 방법	음성인식프로그래밍	28.06	59	2.10	2.44
	웹프로그래밍	5.22	18	3.45	
	인공지능프로그래밍	3.41	6	1.76	

표 2에서 보여준것처럼 제안한 방식이 시험구동형개발을 적용하지 않는 방식에 비해 외부품질의 측면에서 볼 때 12.65% 정도 더 개선되었다는것을 알수 있다.

맺 는 말

매 단계에서 시험구동형개발을 증분적으로 진행하는 새로운 개발공정모형을 제안하고 공정모형의 효과성을 CMMI공정들의 만족등급과 품질에서의 개선평가를 통하여 평가하였다.

참 고 문 헌

- [1] Ammar Mutahar AL-Ashmori et al.; Journal of Software Engineering, 11, 123, 2017.
- [2] Rafique Y, Misic VB; Softw Eng IEEE Trans, 39, 6, 835, 2013.
- [3] Roberto Latorre; Empir Software Eng, 19, 753, 2014.

주체110(2021)년 2월 5일 원고접수

A Study on Improvement of Software Development Processes Based on Test-Driven Development

Mun Il Nam, Jang Myong Min and Kim Hyon A

In this paper, we propose a development process based on the Test-Driven Development(TDD) and demonstrate its effectiveness through some application examples.

Keywords: test-driven development, software development processes, software quality