

# PDA 프로그래밍을 위한 개발 프로세스에 관한 연구

이향숙<sup>0</sup> 김종완 오기욱 박용팔 류성열

승실대학교 컴퓨터학과

[aromas@selab.ssu.ac.kr](mailto:aromas@selab.ssu.ac.kr) [wany69@hitel.net](mailto:wany69@hitel.net)

[kuoh@selab.ssu.ac.kr](mailto:kuoh@selab.ssu.ac.kr) [yppark@alm.co.kr](mailto:yppark@alm.co.kr) [syrhew@computing.ssu.ac.kr](mailto:syrhew@computing.ssu.ac.kr)

## A Study of Development Process for PDA Programming

Hyang-Sook Lee<sup>0</sup> Jong Wan Kim Ki Uk Oh Young Pal Park Sung Yull Rhew

School of Computing Soongsil University

### 요 약

프로젝트를 수행할 때 고객이 원하는 소프트웨어를 고객이 원하는 시간에 인도하는 것은 중요하다. 개발 기간의 엄수가 중요 요소로서 각광을 받는 이유는 개발 기간의 엄수가 프로젝트의 성패를 좌우하기 때문이다. Extreme Programming 은 개발 주기의 단위를 작게 나누어 소프트웨어를 단기간에 개발하기 위한 방법론으로 현재 PDA 개발에 Pair Programming이라는 기법으로 자리 잡아가고 있다. 본 논문에서는 PDA 프로그램을 고객인 원하는 품질로 단 기간에 개발하여 프로젝트의 성패에 중요 요소로 자리 잡고 있는 개발 기간 엄수를 지킴으로써 PDA 개발 환경에 효과적으로 적용할 수 있는 PDA 프로그래밍을 위한 개발 프로세스를 제시하고자 한다.

### 1. 서론

프로젝트를 개발함에 있어서 중요한 쟁점으로 자리 잡고 있는 것은 고객이 원하는 소프트웨어를 고객이 원하는 시간에 인도하는 것이다. 고객의 요구사항이 증가하고 고객이 소프트웨어에 대해 적극성을 띠므로써 고객이 요구하는 소프트웨어의 질은 높아지며, 높은 질 뿐만 아니라 소프트웨어를 빠른 시간 안에 얻기를 원한다.

그러나 소프트웨어를 개발하다보면 예기치 못하거나 이미 예견되어지는 요소들로 인해 프로젝트가 위험에 빠지게 된다. 스케줄의 연기, 프로젝트의 취소, 결함 빈도, 비즈니스 대한 이해력 부족, 비즈니스 변화, 스태프들의 변화 등이 프로젝트의 개발을 방해하는 위험 요소들이다. 이러한 위험 요소를 해결하기 위해서는 비용, 시간, 품질, 범위 4가지의 제어 변수를 두어 프로젝트의 수행함에 있어 유연성을 가질 수 있다[2][3].

많은 돈을 투입한다고 해서 프로젝트가 더 빨리 개발 되는 것이 아니며, 인도 단계에서의 시간을 길게 갖는 것이 품질을 높일 수 있고 범위를 넓힐 수 있지만 개발 단계에서는 문제를 일으킬 수 있다. 또한 품질을 희생하면 비용(사람, 비즈니스, 기술)면에서 얻는 효과는

크고, 범위를 작게 할수록 품질을 높일 수 있다[2][3].

본 논문에서는 소프트웨어를 개발하면서 발생할 수 있는 위험 요소를 제거하여 PDA 개발 환경에서 효과적으로 적용할 수 있는 PDA 소프트웨어 개발 방법론을 제시하고자 한다.

### 2. 관련 연구

#### 2.1 Extreme Programming

Extreme Programming은 위험 요소들을 짧은 기간의 단위 기능 개발과 많은 수의 테스트 작업, 고객을 개발의 중심에 두는(Customer-centered)방법을 통해서 해결하기 위해 제안된 방법론이다[5].

Extreme Programming을 위해서는 고객, 프로그래머, 관리자의 역할이 다음과 같아야 한다[1].

#### ▼ 고객의 역할

- 비즈니스를 위해 필요한 사항들을 선택하고, 그들 간의 우선순위를 정하며, 시스템이 보여줄 사항에 대한 테스트를 정의한다.

#### ▼ 프로그래머의 역할

- 분석, 설계, 테스트, 코딩, 시스템 통합을 하면서, 각 업무에 대한 난이도를 추정하고, 고객에게 시스템 전달을 위한 속도를 조절/관리한다.

▼ 관리자의 역할

- 고객과 프로그래머가 함께 잘 일할 수 있도록 해준다. 즉 프로세스를 직접 진행하기 보다는, 프로세스가 부드럽게 진행되도록 도움을 주는 것이다.

Extreme Programming의 주기는 고객이 시스템의 가치들을 정의하고, 개발자가 구현하는 단순 주기가 아니다. 고객이 결정한 부분에 대해 개발자가 비용을 추산하고, 고객이 필요한 범위를 결정하면 개발자는 개발을 한다. 이때 각각의 과정을 통해 얻은 경험을 다음 생명주기에 반영한다[2].

아래 [그림 1]은 가치 정의가 매겨진 Story Card에 대해서 개발자는 Cost와 소요 기간을 측정해서 고객에게 알려준다. 개발자에 의해서 매겨진 비용 산정의 Cost와 소요 기간을 토대로 고객은 다시 Story Card를 선별하게 된다. 고객에 의해 선택된 Story Card에 대해서 개발자는 기능을 구현하게 된다[1][5].

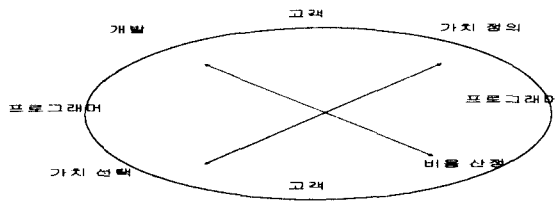


그림 1 Extreme Programming의 개발 주기

2.2 Pair Programming

Pair Programming이란 두 명이 같은 컴퓨터에서 디자인, 알고리즘, 코딩, 테스트를 같이 하는 프로그래밍 기법이다[1].

한명의 프로그래머는 키보드와 마우스를 통하여 타이핑을 하고, 나머지 한 명은 지켜보기를 하면서 잠재적인 품질에 초점을 맞추어 소프트웨어의 품질을 보장하게 된다[1].

Pair Programming은 비용을 증가시킬 수 있는 가능성을 가지지만 품질과 지식 습득 면에서 보게 된다면 충분히 보상이 가능하게 된다.

아래 [그림 2]는 새로운 작업이나 실패한 적용 테스트에 대해서(New Task or Failed Acceptance Test) Pair를 구성하고 새로운 Unit Test를 설치한다. 반복적인 Unit Test를 통해서 새로운 기능이 구현되면(Create a

Unit Test, Pair Programming) 통합 작업을 한다(Continuous Integration). 작업 도중에 구현할 기능에 따라 파트너의 교체가 일어날 수 있으며(Move People Around), 복잡한 코드의 경우에는 분해작업을 통해서 간단한 코드로 고치도록 한다(Refactoring)[1][5].

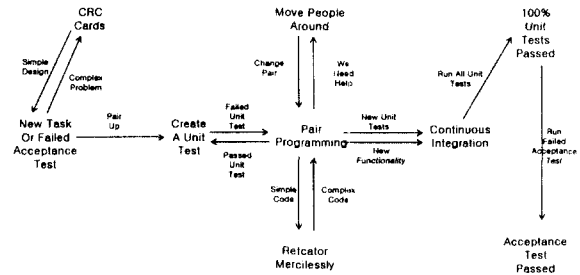


그림 2 Pair Programming 개발 프로세서

3. 본론

소프트웨어를 개발하다 보면 고객의 요구사항이 수시로 변하여 프로젝트 수행에 어려움을 주게 된다. 프로젝트 주기의 시작 단계에서의 고객의 요구사항 변경은 개발 기간 엄수에 크게 영향을 끼치지 않으나 완성 단계에서의 고객의 요구사항 변경이 생긴다면 프로젝트의 성패를 좌우할 정도의 위험 요소로 적용되게 된다. 이러한 프로젝트 위험 요소를 감소시키기 위해서는 개발 범위를 작게 모듈화 시켜야 하며 개발 도중에 지속적으로 고객과의 만남을 갖음으로써 고객의 요구사항 변경을 즉시 반영할 수 있도록 해야 한다.

아래 [그림 3]은 PDA 소프트웨어 개발 프로세스이다. 고객의 요구사항을 기반으로 하여 전체적인 시스템 노드 토폴로지를 구성하고 시스템 노드 토폴로지 단계를 거쳐 나온 시스템 구성도를 갖고서 노드별로 분석하여 전체 시스템에서 얻을 수 있는 노드별 모듈을 산출물로 얻게 된다. 노드별 모듈 산출물은 Pair Programming에 적합한 형태인 Small 모듈로서 기술별로 모듈화 하게 된다. 구현될 모듈을 최대한 작게 하여 피드백 과정을 짧게 하여 고객의 요구사항을 즉각적으로 받아들이며 통합 과정을 통하여 즉각적으로 테스트 되어야 한다. Pair Programming 단계서 Small 모듈이 바로 구현되는 것이 아니라 실로 구현할 모듈에 대한 프로토타입 모듈을 구현한 후 충분한 테스트를 거쳐야만 비로소 모듈별로 구현이 된다. 프로토타입에 근거하여 모듈별 구현이 완성되게 되면 모듈별 테스트를 수행하게 되는데 이때

테스트는 Emulator를 통하여 하게 된다. PDA에 직접 프로그램을 올려서 Test를 하게 될 경우엔 컴파일 시간 뿐만이 아니라 수행 시간이 오래 걸리기 때문에 개발 기간을 짧게 하려고 하는 목적에 위배되게 된다. 위의 과정을 통해 모듈별 테스트가 끝나면 모듈 통합을 하고 모듈 통합 테스트를 하게 된다. 모듈 통합 테스트 역시 모듈별 테스트에서와 마찬가지로 Emulator를 통하여 충분히 테스트를 한 후에 PDA에 올려서 프로그램을 수행한다.

이러한 전 과정은 한번으로 끝나는 것이 아니라 고객의 요구사항이 발생 할 때 마다 반복적으로 수행되어야 하며, 그러기 위해선 고객과의 지속적인 만남이 요구된다.

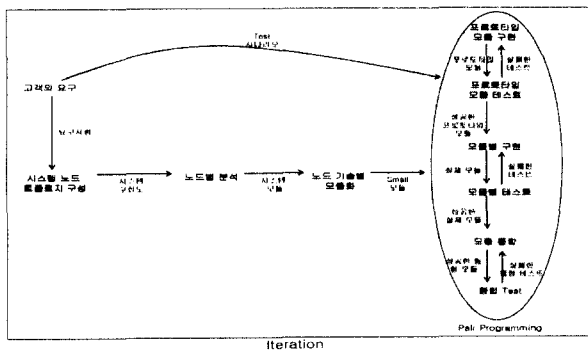


그림 3 PDA 프로그램 개발 프로세스

4. 사례 연구

PDA를 이용한 환자 관리 시스템을 구현한다[4]. Mobile Device인 PDA는 Adams-H 서버와 무선으로 통신을 하여 환자를 관리하게 되고, Adams-H 데이터 베이스는 무선을 통하여 Adams-H 서버와 통신을 하여 동기화를 하게 된다. 시스템 토폴로지 구성은 아래 [그림 4]와 같다.

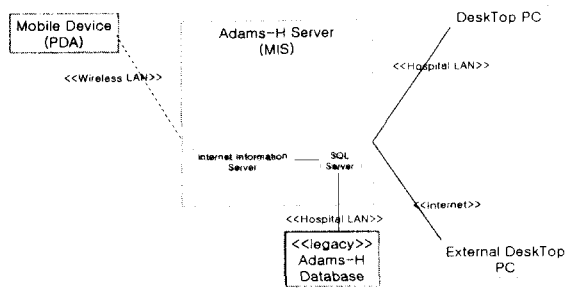


그림 4 시스템 노드 토폴로지 구성

환자 관리 시스템이 노드 기술별 모듈화를 거치게 되면 아래 [그림 5]와 같이 Small 모듈화 된다.

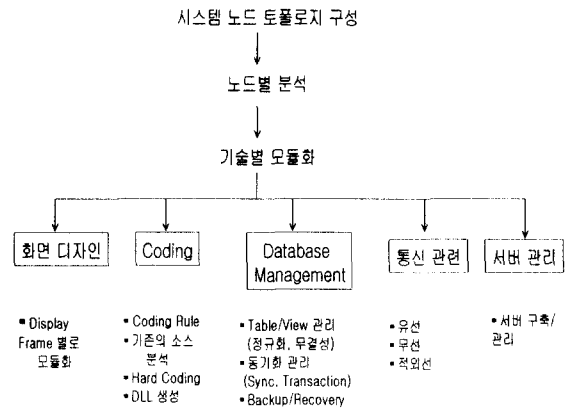


그림 5 노드 기술별 모듈화

5. 결론 및 향후 연구과제

본 논문에서는 PDA 소프트웨어 개발 시 발생할 수 있는 위험 요소를 최소화 하기 위한 PDA 소프트웨어 개발 프로세스를 제시하였다. 고객과의 지속적인 의견 교환을 통하여 고객의 요구사항 변경에 따른 위험 요소를 최소화 하였고, 모듈을 최소화 함으로써 프로젝트 개발 기간을 단축하였다.

향후 연구과제로는 모듈의 크기에 따른 개발 기간에 대한 연구를 통하여 PDA 소프트웨어의 개발에 가장 효율적인 모듈의 크기에 대한 연구를 해보려고 한다.

6. 참고 문헌

- [1] Kent Beck, Extreme Programming Installed, Addison-Wesley, 2000.
- [2] Kent Beck, Extreme Programming explained: Embrace Change, Addison-Wesley, 2000.
- [3] Kent Beck, Planning Extreme Programming, Addison-Wesley, 2000.
- [4] Nick Grattan, Marshall Brain, Windows CE 3.0 Application Programming, Prentice Hall PTR, 2001.
- [5] 공재원, 심우곤, 백인섭, 국내 소프트웨어 개발 환경에서의 Extreme Programming 개발 방식의 적용성 연구, 한국정보과학회 2001.