

# 애자일 기법에 기반한 모바일 응용 SW 개발 단계 및 활동

정양재\*, 이지현\*, 신규상\*  
\*한국전자통신연구원  
e-mail:cornor@etri.re.kr

## Development Phase and Activity of Mobile Application SW Based on Agile Methodology

Yang-Jae Jeong\*, Ji-Hyeon Lee\*, Gyu-Sang Shin\*  
\*Electronics and Telecommunications Research Institute

### 요 약

MASAM 방법론은 모바일 플랫폼 위에서 동작하는 응용 SW 개발을 목적으로 하는 방법론이다. 모바일 응용 SW는 통신 환경과 밀접하게 연결되어 있으며 사용자의 반응이 즉시 나타나는 특징을 가진다. 또한 복잡한 개발 기술보다는 사용자의 요구를 적극적으로 수용하여 제품을 빠르게 출시하는 것이 중요하다. 이러한 특성은 해결하기 위해 MASAM은 애자일 방법론에 기반하여 정의한다.

### 1. 서론

최근 애자일 방법이 많이 사용되고 있다. XP를 사용하고 있는 회사가 늘어나고 있으며, RUP도 애자일 개념을 받아들이면서 Open UP, AUP(Agile Unified Process) 등이 새롭게 등장했다. 애자일 방법은 개발 단계보다 개발팀의 사람에게 초점을 맞춘다. 개발 환경을 공유하여 빠른 결정을 내리고 즉시 개발한다. 모바일 응용 SW는 소비자의 의견을 반영하여 빠르게 제품을 출시하여야만 살아남을 수 있다. 또한 작은 규모의 회사가 많아서 체계적인 방법론을 적용하기 어렵다. 이러한 모바일 응용 SW의 특성은 애자일 방법을 선택하게 한다. 핀란드의 VTT 회사는 모바일 응용 SW 개발 방법론으로 애자일 기법을 적용한 MOBILE-D[1]라는 개발 방법론을 발표했다. 본 논문에서는 애자일 방법론과 아키텍처를 중심으로 하는 MASAM 방법론을 기술한다. MASAM 방법론은 동적 상황을 인식하여 동적 재구성을 수행하는 적응형 SW에 대한 개발 방법을 포함한다.

- 모바일 플랫폼 환경에 따라 다양한 버전 필요
- 모바일 단말기에 영향

MASAM은 위와 같은 모바일 응용 SW의 특징을 만족시키기 위해 개발 초기에 UI 프로토타이핑을 수행하여 사용자의 요구를 적극적으로 수용하고, 빠른 개발을 위해 애자일 방법론을 따른다. 또한 MDA(Model Driven Architecture) 기법을 활용하여 여러 플랫폼에 동작하는 골격코드를 생성하도록 한다.

### 2.2 애자일 방법론

현재까지 모바일 응용 SW는 개인을 대상으로 하는 응용 SW를 주종을 이루고 있으며, 점점 비즈니스 영역으로 확대되고 있다. CPU 및 메모리 등 자원의 제약이 많은 모바일 기기의 특성에 따라 모바일 응용 SW는 비교적 간단한 형태의 SW로 이루어진다. 따라서 비교적 적은 인원의 개발자가 짧은 기간 동안 응용 SW를 개발해야 한다. 이러한 특성에 따라 MASAM은 애자일 방법을 사용한다.

### 2. 개발 방법론 특성

#### 2.1 모바일 응용 SW 개발 방법

모바일 응용 SW는 사용자가 항상 휴대하고 다니는 모바일 단말기에서 동작하는 SW를 말한다. 모바일 응용 SW는 사용자와 쉽게 접할 수 있는 특징과 함께 모바일 단말기의 성능 영향을 받는다는 특징을 갖는다.

- 10명 내외의 개발자
- XP 스타일(스토리 카드, TDD(Test Driven Development), 연속 통합, 동료검토)
- 참여자의 프로세스 적응도를 검사하여 팀의 프로세스 수행 능력 향상
- 단계 수행 여부를 파악하기 위한 체크리스트 사용
- 공동 작업할 수 있는 기법 제시

- 즉각적인 사용자 반응
- 시장 적시성 중요
- 사용자 반응 중요

### 2.3 영역 지식 활용

모바일 응용 SW는 몇 가지 특징적인 형태로 이루어진 경우가 많다. MASAM은 영역 지식을 등록하고 재사용하는 방법을 사용하여 개발 시간을 단축한다. 아키텍처 패턴, 디자인 패턴, UI 패턴을 정의할 수 있는 기법을 제시하여 영역 정보를 저장하고 아키텍처 활동에서 영역 정보를 활용하여 초기 개발 시간을 단축한다. 또한 MDA의 코드 생성 기술을 이용하여 플랫폼 특성을 반영하는 골격코드를 빠르게 생성한다.

- 패턴을 활용하여 기존 자산 활용
- MDA 기법을 활용하여 플랫폼 구현 코드 생성

2.4 동적 재구성 구축

MASAM은 “동적 아키텍처 런타임 플랫폼” 세부과제를 지원하기 위해 모바일 응용 SW를 동적 재구성하기 위한 상세한 작업 절차와 지침을 제공한다. 동적 재구성 기능을 활용하면 유연한 SW 개발이 가능해지며 부족한 모바일 환경의 자원을 최대한 효율적으로 사용할 수 있다.

- 동적 재구성을 위한 UI 프로토타입 생성 기법
- 동적 재구성을 위한 테스트 기법
- 동적 재구성을 위한 아키텍처 정의 방법

3. 개발 방법론 단계

고적전인 폭포수 개발 방법론의 경우 요구사항, 설계, 구현, 테스트가 순차적으로 이루어지지만 애자일 방법론에 기반을 둔 MASAM은 점진적, 반복적 기법을 사용한다. 고객의 의도를 파악한 이후 개발 단계를 반복하면서 제품을 개발한다. 또한 제품 개발 과정에서도 적극적으로 고객의 요구사항을 반영한다.

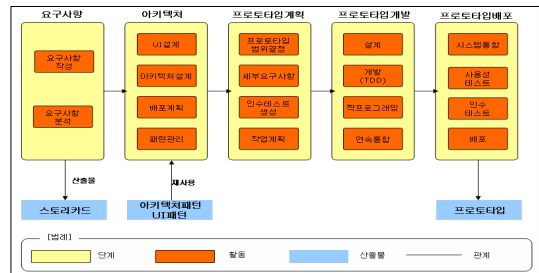
3.1 제품 기획 단계

MASAM의 전 과정에서 고객과의 대화가 필요하지만 제품 기획 단계에서는 특히 더 많은 고객과의 대화가 필요하다. 고객이 필요한 기능을 제시하고, 개발자는 고객이 무엇이 필요한가 알 수 있도록 돕는다. 제품을 사용할 고객의 유형을 정함으로써 제품의 목적을 구체적으로 묘사할 수 있다. 제품 개발 초기에는 제품에 대한 개념이 일치하지 않기 때문에 제품의 개념을 일치하는 과정이 중요하며, 개발팀 내부의 지식, 개발 환경 등을 일치시키기 위한 과정이 중요하다.

- 제품파악 : 고객의 의도를 알기 위한 과정으로 제품 개욕을 파악하고 제품 패턴을 분석한다.
- 사용자파악 : 제품 사용자를 세분화하여 제품의 기능을 명확하기 위한 활동이다. 사용자 후보를 추출하고 그룹화 하여 제품 사용 유형을 파악한다.
- 개념프로토타입 : 고객이 제품을 사용하는 방법을

예상하여 제품의 목적과 기능을 파악한다.

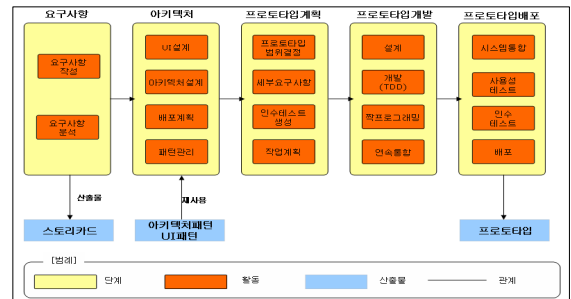
- 프로젝트설정 : 개발팀이 제품을 개발하기 위한 준비 과정으로 제품 개발을 위한 사전학습, 프로젝트에 필요한 자원 준비, 구현 환경을 준비한다.



<그림 1> 제품 기획 단계 개념도

3.2 구체화 단계

구체화 단계는 고객의 의도를 명확히 표현하고, 그것을 제품에 어떻게 반영할 것인가를 구체적으로 결정하는 단계이다. 고객의 의도를 사용자의 언어로 스토리카드를 이용해서 표현하고, 스토리카드를 아키텍처, UI 설계를 통해 구체화한다. UI 중심의 프로토타입 개발 범위를 결정하고 프로토타입을 개발함으로써 제품의 완성된 모습을 미리 알아본다. 고객이 눈으로 볼 수 있는 프로토타입을 이용하여 고객의 의도를 다시 한 번 파악하고 사용자이성 중심으로 테스트한다.



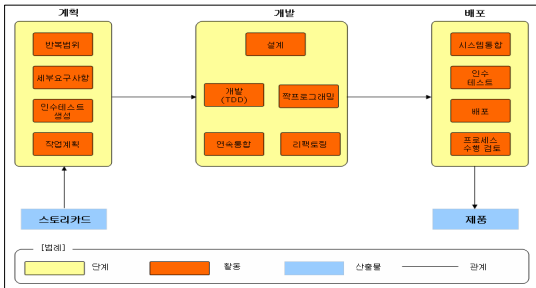
<그림 2> 구체화 단계 개념도

- 요구사항 : 고객의 의도를 글로 명확하게 표현한다. 스토리카드를 사용하여 적절한 크기로 요구사항을 재정의하고 관련 기능별로 배치한다.
- 아키텍처 : 스토리카드의 내용을 분석하여 UI를 포함하는 제품 아키텍처를 결정한다. 영역 정보를 포함하는 아키텍처를 재사용하여 개발 시간을 단축한다.
- 프로토타입계획 : UI 중심의 프로토타입 계획을 수립한다. 개발할 스토리카드의 범위를 정하고 작업계획을 세운다.
- 프로토타입개발 : 설계, 개발, 작프로그래밍, 연속통합 기법을 이용하여 스토리카드를 개발한다.
- 프로토타입배포 : 프로토타입을 고객에게 배포한다.

기능 테스트와 함께 사용성 테스트가 중요 검사 항목이다.

### 3.3 제품 개발 단계

제품 개발 단계는 배포, 반복 계획에 따라 제품을 개발하는 단계이다. 제품 개발 단계는 배포, 반복 횟수에 따라 여러 번 수행하기도 한다. 스토리카드[2]를 이용하여 고객과 세부적인 요구사항을 논의하여 스토리카드의 작업 시간을 추정한다. 반복에 할당된 시간에 맞춰 스토리카드 구현 계획을 세우고 설계, 개발, 연속통합의 기법을 사용하여 점진적으로 개발을 진행한다. 작업 결과를 수시로 통합하고 테스트를 자동화하여 통합 시마다 테스트를 수행한다. 개발 환경에 따라 짝 프로그래밍과 리팩토링을 수행한다. 개발 결과는 배포 계획에 따라 고객에게 전달한다. 작업 과정을 논의하여 다음 반복에 반영할 수 있도록 한다.



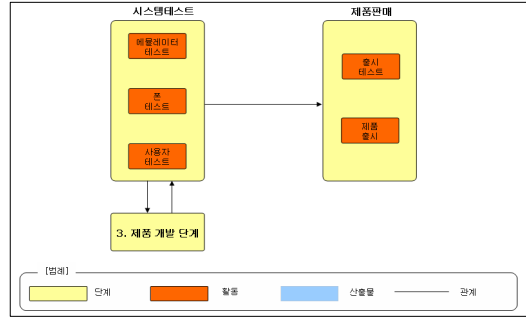
<그림 3> 제품 개발 단계 개요

- 계획 : 반복 과정의 계획을 결정하는 활동으로 반복에서 처리할 스토리카드를 결정하고 세부사항을 논의하여 작업 시간을 추정한다.
- 개발 : 스토리카드 단위로 제품을 점진적으로 개발한다. 테스트 주도 개발 방법으로 설계부터, 구현, 테스트까지 수행한다.
- 배포 : 배포 계획에 따라 제품을 배포한다. 점진적으로 제품을 개발하기 때문에 마지막 배포 전까지 제품이 완료되지는 않지만 개발 완료된 스토리카드의 범위에 대해서는 사용 가능하여야 한다.

### 3.4 제품화 단계

모바일 응용 SW는 에뮬레이터 종류, 모바일 단말기 종류, 통신 환경에 따라 환경이 다르기 때문에 예측하지 못했던 오류가 발생하기도 한다. 모바일 응용 SW를 에뮬레이터와 폰에서 실행하여 테스트를 수행한다. 테스트 과정에서 발생한 오류는 점진적 반복 개발 과정을 다시 수행하여 오류를 수정한다. 모바일 응용 SW는 통신 환경에 따른 절차를 수행하여 상품화한다.

- 시스템테스트 : 제품 출시를 위해 다양한 형태의 테스트를 수행한다. 테스트 과정에서 발견한 오류는 제품 개발 단계를 다시 수행하여 오류를 수정한다.



<그림 4> 제품화 단계 개념도  
모바일환경에 따라 에뮬레이터, 폰 테스트를 수행한다.

- 제품판매 : 모바일 응용 SW는 통신 환경에 따라 제품을 출시해야 한다. 제품판매 활동에서는 통신 환경에서 요구하는 절차를 만족시키기 위한 테스트를 수행하여 제품을 출시한다.

## 4. 향후 연구

본 논문은 MASAM의 개발 과정 중에서 단계 및 활동까지 기술하였다. 단계 및 활동에서는 구체적인 활동과 산출물을 기술하지 않아서 방법론의 특성이 제대로 표현되지 못했다. 향후 과제 진행에 따라 MASAM의 활동을 정의할 계획이며, 방법론을 Eclipse의 EPF[3]에 기술하여 체계화시킬 예정이다.

### 참고문헌

[1] Pekka Abrahamsson, "Mobile-D: an agile approach for mobile application development' Approach" Conference on Object Oriented Programming Systems Languages and Applications, pp174 - 175, 2004.  
 [2] Mike Cohn, "사용자 스토리 : 고객 중심의 요구 사항 기법" 서울 : 인사이트, 2006.  
 [3] <http://www.eclipse.org/epf/>