

# Agile 개발방법론의 가치 관행이 오픈소스 프로젝트 커뮤니티에 미치는 영향 연구

손효정\* · 이민규\*\* · 성백민\*\*\* · 김종배○

\*충실대학교 SW특성화대학원

E-mail : hyojung.sohn@gmail.com\* marse101@naver.com\*\* feeling127@naver.com\*\*\* kjb123@ssu.ac.kr○

## 요약

소프트웨어 개발 환경의 급격한 변화와 지속적인 요구사항 변경으로 인해 발생하는 전통적 개발 방법론의 문제점을 해결하기 위한 애자일 개발방법론은 관심의 대상이 되었다. 비교적 짧은 주기를 가지고 요구사항 분석, 설계, 코딩, 테스트 과정을 끊임없이 반복하며 점차 소프트웨어의 규모를 늘려가는 애자일 개발방법은 오픈소스 소프트웨어의 개발방식(OSSDP)과 매우 유사하다고 볼 수 있다. 지금까지 애자일 방법론은 Scrum, XP, Lean, Crystal 방식 등 여러 가지 방법으로 시도 및 제안되고 있으며, 점차 다양한 프로젝트에서도 성공적으로 수행할 수 있도록 개선되고 있다. 이러한 애자일 개발방식의 특징을 OSSDP에 적용함으로써 더 많은 오픈소스 프로젝트가 성공할 수 있도록 하는 프로세스를 연구하는 것은 의미가 있다. 애자일 성명서에서 추구하는 4가지 가치인 소통, 단순성, 피드백, 용기를 반영한 관행(practice)을 OSSDP에 채택하여 추가할 경우, 사용자 만족도를 높이고 결과적으로 오픈소스 프로젝트 성숙도를 높이는데 기여할 것으로 기대한다.

## I. 서론

소프트웨어 개발 환경이 점차 불확실해지고 급격한 변화를 요구하는 환경으로 변화하면서 전통적 개발방식은 실패를 거듭하게 되었다. 이와 더불어 문서 및 절차 위주의 방법은 변화하는 사용자의 요구사항을 신속하게 반영하지 못했고 결국 소프트웨어의 품질 하락과 만족도 저하를 가져왔다. 이러한 전통적 개발방법론의 문제점을 해결하고자 개발 환경 등 주위 변화를 빠르게 수용하고, 능동적으로 대응하는 여러 방법론을 통칭하는 애자일 개발방법론은 관심의 대상이 되고 있다.

Gilliam의 OSSPD 모델[1]에 따르면, 프로젝트를 매니저가 인터넷상에 최초로 오픈소스 프로젝트를 시작하면, 개발팀은 버그를 찾고 새로운 기능을 추가하고 수정 파일을 제출하고 매니저는 제출된 가장 좋은 기능과 코드를 통합하고 공식적인 버전을 배포한다. 한편 사용자가 제품을 사용하는 중에 사용자 나름대로 버그를 찾고 기능을 추가하고 수정 파일을 제출하게 되고, 매니저는 역시 제출된 가장 좋은 기능과 코드를 통합하고 공식적인 버전을 배포하는 작업을 반복한다고 설명한다. 비교적 짧은 주기를 가지고 요구사항분석, 설계, 코딩, 테스트 과정을 끊임없이 반복하며 점차 소프트웨어의 규모를 늘려가는 애자일 개발방법은 오픈소스 소프트웨어의 개발방식(OSSDP)과 매우 유사하다고 볼 수 있다.

지금까지 애자일 방법론은 Scrum, XP, Lean, Crystal 방식 등 여러 가지 방법으로 시도 및 제안되고 있으며, 점차 프로젝트의 다양한 조건 속에서도 성공적으로 수행할 수 있도록 개선되고 있다. 이러한 애자일 개발방식의 특징을 OSSDP에 적용함으로써 더 많은 오픈소스 프로젝트가 성공할 수 있도록 하는 프로세스를 연구하는 것은 의미가 있다고 하겠다. 애자일 가치를 반영한 관행이 오픈소스 커뮤니티의 성숙도에 긍정적 영향을 미치는 요인을 분석, 도출하여 기존 Scacchi의 OSSDP의 모델을 기반으로 수정 보완한 프로세스를 제안하고자 한다. 본 연구에서는 요인도출을 위한 모형을 제시하는 것으로 한다.

## II. 관련 연구

### 2.1 애자일 얼라이언스의 성명서[2]

2001년 초, 일단의 산업 전문가들이 소프트웨어 팀이 신속하게 개발하고 변화에 대응할 수 있게 할 가치와 원리에 대한 윤곽을 잡고자 모였다. 이들은 스스로를 애자일 얼라이언스(Agile Alliance)라고 불렀다. 이들은 이들에 걸쳐 가치 선언서를 작성하였다. 그 결과가 애자일 얼라이언스의 성명서였다. 문서는 효과적인 소프트웨어 개발에 대한 철학적 토대를 제공한다. 이 성명서에

서 주장하는 4가지 가치는 다음과 같다.

- 절차와 도구를 넘어선 **개성과 화합**
- 문서화를 넘어선 **동작하는 소프트웨어**
- 계약과 협상을 넘어선 **고객과의 협력**
- 계획 준수를 넘어서 **변화에의 대응**

애자일 기법은 Beck 교재[3]에서 4가지 주요가치를 소개하고 있다. 이 가치를 오픈소스 개발방법에 적용할 기본 가치로 본다.

- **소통** : 개발자는 개발자간, 관리자와 고객에게 정직하고 신뢰하며 손쉬운 방식으로 프로젝트에 관한 정보와 아이디어를 교환해야 한다.
- **단순성** : 가능한 단순한 솔루션을 선택해야 한다. 고객에게 현실적이고, 유형의 분명한 가치를 제공하는데 최고의 가치가 있다. Beck은 “(설계에서) 완벽함은 더 이상 더할 것이 없을 때가 아니라, 더 이상 뺄 것이 없을 때이다”고 말한다.
- **피드백** : 올바르게 가고 있는지 확인하기 위해 동료개발자 고객, 그리고 모든 종류의 시험도구를 이용해 피드백 받도록 한다. 이것은 분면 ‘소통’과 매우 연관성 있다. 작업자는 생성하고 있는 코드로부터도 즉각적인 피드백이 있어야 한다. 이것은 빈번한 시험, 통합, 버전 및 배포를 의미한다.
- **용기** : 프로젝트에 관여된 모든 이해당사자는 프로젝트에 관한 자신의 입장을 표현할 용기와 권리가 있어야 한다. 개발자는 자신의 코드를 수정하거나, 자신이 한 작업에 대해 고객과 끊임없이 논의하는 것을 두려워하지 않으며, 더 나은 솔루션을 찾고 필요하고 가능할 때마다 코드를 수정할 용기가 있어야한다. 고객은 개발자가 자신의 돈을 낭비할까 두려워 하지 않아야 한다.

### 2.2 애자일 개발 프로세스[4]

애자일 프로세스는 설계, 코딩, 테스트, 검토 단계를 끊임없이 반복하며 규모를 증가시키는 방식으로 진행된다. 모든 단계에 사용자가 참여해 결정적인 역할을 하게 된다. 즉 사용자로부터 새로운 기능이 추가되어야 하며 이를 위해 애자일 개발에서는 코드베이스의 디버깅(debugging)과 가독성(refactoring)검사를 새로 해야한다.

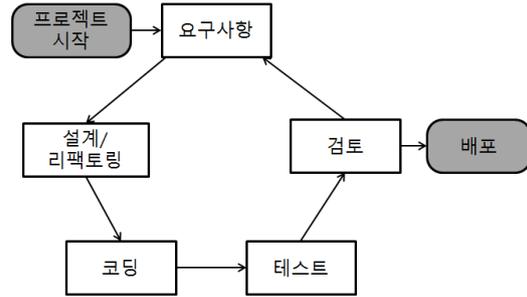


그림 1. 애자일 소프트웨어 개발 프로세스

- **디버깅(debugging)** : 프로그램의 버그를 발견해 수정하는 것이다. 버그는 일반적으로 프로그래밍의 실수에서 나오지만 이를 바로잡는 일은 너무 어렵다. 사용자는 자주, 잘 버그를 잡아 내곤 한다. 그래서 사용자와 개발자간의 의사소통은 중요한 의미를 갖는다.
- **리팩토링(refactoring)** : 소스코드를 재구성해 같은 기능을 더욱 읽기 쉽고 효과적으로 수행할 수 있도록 해준다. 이 과정은 새로운 기능이 추가되지는 않고 오직 코드의 품질만 개선시킨다.

### 2.3 Scacchi의 오픈소스 소프트웨어 개발 프로세스(OSSDP) 모델[5]

오픈소스 프로젝트를 시작하기 전에는 요구사항 수집이 불가능하며, 이는 빠른 제품출시의 근거가 되기도 한다.

Scacchi는 7단계 오픈소스 소프트웨어 개발 프로세스를 제안한다. 곧, 1) 다운로드 및 설치; 2) 최종 사용; 3) 경험 나누기; 4) 분석 및 재설계; 5) 요건 단언 - 디자인; 6) OSS 코드 개발; 7) 구성 관리이다.

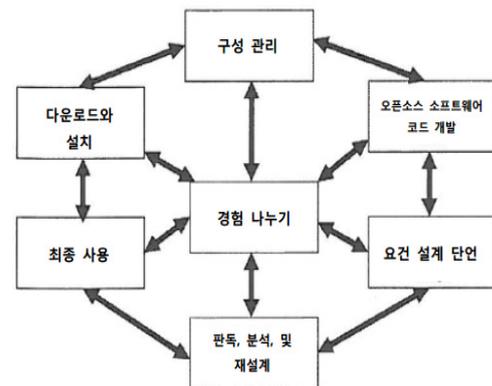


그림 2. Scacchi의 일반적 OSSDP Model

### 2.4 애자일 기법과 OSSDP 비교[6]

Warsta & Abrahamsson의 연구에서 OSS개발

은 애자일과 전통개발방식의 중간 패러다임으로 간주하고 있다.

각 개발방법의 가장 중요한 차이점으로 개발팀의 근접성, 규모, 프로젝트내 고객의 표현, 개발의 주요 목표로 설정하고 있다. 그 결과, OSSD는 전통적 개발방법보다 애자일 방식에 더 근접한 것으로 평가했으며, 오픈소스 개발자들은 지리적으로 분산되어 있다는 점과 대개 고객과 개발자가 일치하는데 반해 애자일 방식은 대부분 일치하지 않는다는 점이 차이점으로 제시하고 있다.

표1은 Warsta & Abrahamsson가 제시한 차이점 중 일부를 제시한 것이다.

표 1. 애자일과 OSSD간 비교

오픈소스 드영역	애자일방법	오픈소스 소프트웨어
개발자	민첩, 박식, 공동, 협력	지리적으로 분산, 협력, 박식 및 민첩한 팀
고객	전용, 박식, 공동, 협력, 대변 및 자율	전용, 박식, 협력, 및 자율
요건	주로 신생; 신속한 변화	주로 신생; 신속한 변화, 공동 소유, 지속적인 발달 - "절대로" 마무리되지 않음
아키텍처	현재 요건에 맞게 설계됨	공개, 현재 요건에 맞게 설계됨
리팩토링	저렴	저렴
규모	소규모 팀과 제품	대규모 분산 팀 및 소규모 제품
주요목표	신속한 가치	문제 해결

### III. 연구 설계

#### 3.1 연구 가설

애자일 방식은 소프트웨어 개발방법 자체를 개선하려는 노력을 바탕으로 진화해 오고 있다. 이러한 이유로 지금까지 다양한 종류의 애자일 방식, 이를테면 Scrum, XP, Lean, Crystal과 같은 방법들이 소개되었고, 다양한 프로젝트에서도 성공적인 개발을 할 수 있도록 시도되고 있다. 이러한 애자일 방식의 특징들을 OSSD에 적용해 오픈소스 프로젝트의 다양한 특성에도 상관없이 성공적으로 성장할 수 있는 프로세스를 제안하고자 한다.

이를 검증하기 위한 연구모형은 애자일 관행(Practices)이 오픈소스 제품의 질을 높이며 이로 인해 '사용자 만족도'에 영향을 미칠 것이며 '사용자 만족도'는 '커뮤니티 성숙도'에 영향을 미치는 것으로 가설을 설정하였다.

높은 커뮤니티 성숙도는 성공적인 오픈소스 프

로젝트로 간주하였으며, 각각의 애자일 가치를 대표하는 애자일 관행은 다음 표2와 같다.

- H1: SNS기능을 사용한 OSSDP는 '사용자 만족도'를 높일 것이다.
- H2: 리팩토링을 실시한 OSSDP는 '사용자 만족도'를 높일 것이다.
- H3: TDD를 채택한 OSSDP는 '사용자 만족도'를 높일 것이다.
- H4: '사용자 만족도'는 '커뮤니티 성숙도'를 높일 것이다.

표 2. 애자일 가치에 대응하는 관행[7]

가치(Value)		관행(Practice)
소통	↔	SNS 기능 활용
단순화	↔	리팩토링
피드백	↔	테스트 주도 개발 (Test Driven Development)

#### 3.2 연구 모형

앞서 살펴본바와 같이, 본 연구의 애자일 관행과 '커뮤니티 성숙도'의 관계를 개념적 연구모형을 도출하면 그림#과 같다.

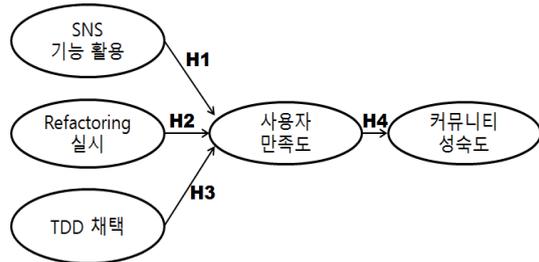


그림 3. 연구모형

### IV. 향후 연구

본 연구에서 도출한 연구 모형을 검증하기 위해 최대 오픈소스 프로젝트 리파지토리인 Sourceforge.net을 대상으로 각 카테고리별로 무작위로 100개의 표본을 추출한다. 실제 운영되는 프로젝트를 대상으로 SNS 기능 활용 여부, 리팩토링 실시 여부, 테스트 주도개발을 위한 툴 사용 여부 등의 실증적 분석을 실시하며, '사용자 만족도'의 기준은 오픈소스 제품의 평가점수로 측정하고 '커뮤니티 성숙도'는 기존 연구의 평가기준을 활용해 등급화하여 측정한다.

애자일 관행을 대표하는 3가지 요인이 사용자 만족도에 영향을 미치는가와 사용자 만족도가 커뮤니티 성숙도에 영향을 미치는가를 실증적 분석을 통해 상관관계를 도출하면, 긍정적 영향을 미치는 요인을 채택하여 새로운 OSSDP 모델을 제

안하는 것을 본 연구의 최종 목적으로 한다.

## V. 결 론

애자일 개발방법과 오픈소스 소프트웨어 개발 프로세스에는 상당한 유사성이 존재한다. 그러나 더욱 다양한 환경과 조건의 프로젝트에서도 성공할 수 있도록 빠르게 진화하는 애자일 개발방법의 특징은 OSSDP에서 참고할만 하다. 본 연구에서 애자일 개발방법의 가치를 반영한 관행이 OSSDP에 긍정적 영향을 미칠 것으로 예상하여 연구모형을 도출하였다.

향후 연구에서는 도출된 연구모형을 실제 오픈소스 리파지토리를 대상으로 실증분석을 실시하고, 긍정적 효과를 보인 요인을 채택하여 애자일의 가치를 반영한 OSSDP 모델을 제안하고자 한다. 이 모델을 통해 다양한 조건의 오픈소스 프로젝트에서도 사용자 만족도를 높이며 결과적으로 오픈소스 프로젝트의 높은 성숙도에 기여할 것을 기대한다.

## 참고문헌

- [1] J. O. Gilliam, "Improving the Open Source Software Model with UML Case Tools", *Linux Gazette*, 67, June 2001
- [2] Agile Alliance Manifesto, <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>
- [3] Beck, K., *Extreme Programming Explained: Embrace Change*, 2nd edition, Addison-Wesley Professional, 2004
- [4] Tucker, A., Morelli, R., & Silva, C. D. *Software Development: An Open Source Approach*. CRC Press, Inc., 2011
- [5] Scacchi, W., "Open source software development process", Volume 2.5, Retrieved on November 11, 2008, from <http://www.ics.uci.edu/~wscacchi/Software-Process/Open-Source-Software-Process-Models/Open-Source-Software-Development-Processes.ppt>
- [6] Warsta, J., & Abrahamsson, P., "Is open source software development essentially an agile method.", In *Proceedings of the 3rd Workshop on Open Source Software Engineering*, May, 2003, pp. 143-147
- [7] Russo, B. (Ed.). *Agile technologies in open source development*. IGI Global., 2009