

클라우드소싱 기반 소프트웨어 개발 프로세스 개선

김형락[○], 이욱진

한양대학교 컴퓨터공학과

koliaok@hanyang.ac.kr, sottle@hanyang.ac.kr

Improvement of Crowdsourcing based Software Development Process

Hyunglak Kim[○] Scott Uk-Jin Lee

Department of Computer Science & Engineering, Hanyang University

요 약

현재 해외의 여러 기업들은 클라우드소싱 기반 소프트웨어 개발을 통해 업무를 세분화하여 복잡한 개발 문제를 단기간에 해결하고 생산성을 증가시켜 소프트웨어 경쟁력을 높이고 있다. 그러나 국내에서는 클라우드소싱 기반 소프트웨어 개발이 잘 알려져 있지 않고, 이에 관한 연구를 활용한 사례가 부족한 실정이다. 본 논문에서는 클라우드소싱 기반 소프트웨어 개발의 정의, 장점, 활용되었던 사례 등을 소개하고, 개발 프로세스의 5가지 문제점을 분석해 개선방안을 제시한다. 또한 연구결과를 통해서 향후 클라우드소싱을 활용한 소프트웨어 개발관련 연구에 초석이 될 수 있도록 기술적 발판을 마련한다.

1. 서 론

현재 다양한 산업에서 활용되고 있는 클라우드소싱은 오픈소스(open source) 커뮤니티의 발달로 인해 기존의 내부 자원을 직접 활용하는 인소싱(insourcing)과 외부 자원에 위탁하는 아웃소싱(outsourcing)에서 더욱 진화된 개념이다. 특히 인소싱과 아웃소싱이 한정된 인적 자원으로만 업무를 처리하는 것에 비해 클라우드소싱은 불특정 다수를 선택함으로써 다양하고 많은 정보를 얻을 수 있다. 또한 이를 활용해 성공적으로 사업을 진행할 경우 비용을 낮추고, 문제 해결 소요 시간의 단축 및 솔루션의 폭이 확대되는 장점이 있다[1].

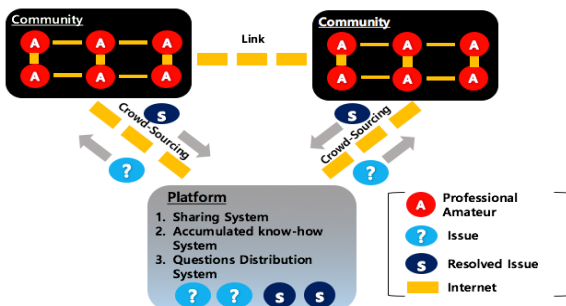


그림 1. 클라우드소싱 구조도

따라서 이런 장점을 바탕으로 다수의 기업에서는 2006년부터 제품 연구와 소프트웨어 개발, 영업과 마케팅 등 기업 활동의 전 영역에 걸쳐 그림 1의 일반적인 클라우드소싱 구조도를 적용시키는 사업을 진행하고 있다. 그 중에서 글로벌 IT기업인 애플(Apple), 구글(Google), 아마존(Amazon) 등이 자체적으로 해결되지 않는 복잡하고, 어려운 소프트웨어 개발 문제를 클라우드소싱을 활용해 해결하고 있다.

이처럼 세계적으로 클라우드소싱 기반 소프트웨어 산업이 빠르게

성장하고 있는 것에 비해 국내에서는 클라우드소싱을 활용해 소프트웨어 개발을 하는 것이 잘 알려져 있지 않는 상황이다. 또한 온라인 상에서 다양한 문화와 성향을 가진 사람들이 모여 하나의 프로젝트를 개발하는 과정에서 다수의 문제점들이 발견되고 있다[2]. 따라서 향후 클라우드소싱을 기반으로 효율적인 소프트웨어 개발을 위해서는 이런 문제점을 개선하는 것은 필수적이다.

본 논문의 구성은 다음과 같다. 2장은 클라우드소싱 기반 소프트웨어 개발의 장점과 활용사례를 소개하고, 3장에서는 TopCoder 사례의 분석으로 밝혀진 총 5가지 클라우드소싱 기반 소프트웨어 개발 프로세스의 문제점을 분석한다. 4장은 이런 문제점을 통해 개발 프로세스를 개선할 수 있는 방안을 제시한다. 마지막으로 5장에서는 개선된 개발 프로세스의 장점과 이를 소프트웨어 산업에 활용해서 향후 클라우드소싱 기반 소프트웨어 개발에 관련된 연구의 방향성을 제시하고자 한다.

2. 장점과 활용사례

클라우드소싱 기반 소프트웨어 개발의 정의는 공개모집을 통해서 소프트웨어 개발 지식을 가진 사람들을 모으고, 다양한 소프트웨어 지식을 기반으로 조직이 해결하기 복잡하고 어려운 소프트웨어 개발 문제를 해하는 것이다. 시간, 위치, 장소의 제한이 없다는 점에서 클라우드소싱 기반 소프트웨어 개발은 아래의 같은 3가지 대표적인 장점이 있다[3].

- 비용절감: 소프트웨어 개발에 필요한 개발자를 회사측에서 고용하지 않기 때문에 개발자에게 들어가는 비용을 절약할 수 있다.
- 빠른 생산성: 클라우드소싱을 활용한 소프트웨어 개발 프로세스를 통해서 업무를 분산하고, 위치와 시간에 제한되지 않기 때문에 빠르게 소프트웨어 개발이 가능하다.
- 다양한 참여자를 통한 품질향상: 여러 분야의 소프트웨어 개발

기술을 가진 사람들이 참여하고, 고객에 의해 평가되어 선택되어지기 때문에 개발자들의 경쟁을 통한 소프트웨어 품질을 향상시킨다.

이런 장점들을 이용해 클라우드소싱 기반 소프트웨어 개발을 적극적으로 활용한 기업으로 탑코더(TopCoder)와 아마존(Amazon)을 들 수 있다.

아마존의 Amazon Mechanical Turk(AMT)는 고객이 필요로 하는 소프트웨어를 만들어 달라고 의뢰하면 터커(Turker)라는 AMT의 온라인 임시 계약직 소프트웨어 개발자들 중에 관심있는 모든 사람들이 해당 소프트웨어를 개발하는 방식이다. 이후 고객이 평가해서 가장 좋은 소프트웨어를 개발한 프로그램이 입찰 된다. AMT에서 현재 활동하고 있는 터커에 수는 50만명에 이르고, 시간당 평균 2달러를 받고 있으며 아마존은 소프트웨어 개발 중계 수수료로 10%를 받아 다시 AMT에 투자함으로써 클라우드소싱 기반 소프트웨어 개발 산업을 점점 더 확장하고 있다[4].

TopCoder의 경우는 각 기업이 프로젝트를 의뢰하면 프로젝트를 모듈로 나눈 다음 커뮤니티에 상금을 걸고 콘테스트를 열어 어떤 팀이 최고의 모듈을 만들었는지 서로 살펴보면서 겨룰 수 있는 경쟁의 장을 마련한다. 그 결과 최종 소프트웨어가 개발되면 인증 과정을 거친 다음 프로젝트를 의뢰한 기업에게 보낸다. TopCoder는 해당 프로젝트를 의뢰한 기업의 수많은 사내 직원들을 통해 1년 이상에 걸쳐 완성했을 프로젝트를 단지 다섯 달 동안 2명의 아키텍트와 프로그램 관리자, 그리고 콘테스트에 참여한 개발자들만으로 프로젝트를 완성할 수 있었다[5].

이처럼 클라우드소싱 기반 소프트웨어 개발의 장점을 활용한 소프트웨어 산업이 점점 더 빠르게 증가되고 있다. 하지만 소프트웨어 산업이 빠르게 증가되는 만큼 클라우드소싱 기반 소프트웨어 개발 프로세스 상에 해결되지 않는 문제점들이 지속적으로 발견되고 있다. 또한 이에 관한 연구가 진행되고 있지만 세계의 각기 다른 문화와 생각이 다른 집단지성의 관계를 분석하기에는 다수의 어려운 점이 존재해 구체적인 개선방안은 없는 실정이다. 따라서 신뢰성 있는 데이터를 기반으로 문제점을 분석해 개발 프로세스를 개선하는 것이 필요하다.

3. 클라우드소싱 기반 소프트웨어 개발 문제점

클라우드소싱을 소프트웨어 개발에 적용하기 어려운 점은 집단지성을 융합하고, 관리할 수 있는 개발 프로세스를 확립시키기 어렵다는 것이다. 또한 클라우드소싱 기반 소프트웨어 사업이 시작된 기간이 짧기 때문에 검증된 데이터를 확보하기 어려운 문제가 있다[6].

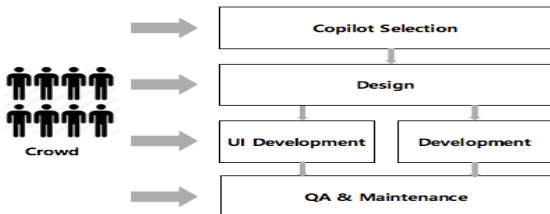


그림 2. TopCoder의 소프트웨어 개발 프로세스

따라서 본 연구의 신뢰성을 높이기 위해 세계적인 클라우드소싱 기반 소프트웨어 개발사인 TopCoder의 그림 2와 같은 개발 프로세스의 문제점을 중점적으로 분석한 결과를 활용해 개선방안을 제시하고자 한다. 아래는 TopCoder에서 분석한 클라우드소싱 기반 소프트웨어 개발 프로세스의 5가지 문제점이다[7].

1. **개발 동기와 합당한 보상문제:** 프로젝트의 짧은 개발 기간, 낮은 급여는 개발자들에게 충분한 동기부여를 주기에 부족하다. 또한 프로젝트 진행 동안 얼마만큼 프로젝트에 참여했는지 수치화 시키기 어렵고, 프로젝트의 복잡성 또한 프로젝트마다 다르기 때문에 합당한 금전적 보상을 하기 위한 문제가 있다.
2. **커뮤니케이션 문제:** 클라우드소싱의 특성상 개개인의 성향과 관계, 소속, 능력, 목표 등이 다르기 때문에 커뮤니케이션을 효율적으로 하기 어려운 문제가 있다. 또한 개발자와 고객 간에 정확한 커뮤니케이션을 온라인을 통해서 모두 반영하기에는 한계가 있다.
3. **개발 계획 문제:** 소프트웨어 개발 업무의 크기와 각 개발자들의 능력이 다르기 때문에 빠른 시간 내에 결과물을 만들어야 하는 특성 상 요구사항 변경에 따른 개발 계획을 유연하게 대처하기 어렵다.
4. **품질 보장 문제:** 클라우드소싱을 활용해 개발된 소프트웨어 품질을 보장하기 위해서 전문적인 개발 지식을 가진 충분한 수의 참가자가 필요하지만, 결과가 좋지 않았던 프로젝트를 분석했을 때 대부분은 참가자 수가 성공한 프로젝트에 비해서 현저히 작았다. 또한 TopCoder에서 프로젝트를 경쟁방식으로 진행하기는 하나 이보다 더욱 확고한 소프트웨어 품질보장 프로세스가 필요하다.
5. **프로젝트 분배 문제:** 클라우드소싱 기반 소프트웨어 개발 프로세스의 특성 상 상호의존적 측면과 복잡한 측면을 모두 고려해 프로젝트를 분배해야 하지만, 그 기준이 분명하지 않아 지나치게 세분화 하거나, 덜 세분화해서 소프트웨어 개발을 어렵게 하는 문제가 있다.

위와 같은 5가지 문제점 모두 다수의 개발자들이 온라인상에서 프로젝트를 진행 했을 때 발생할 수 있는 문제점들이다. 따라서 이를 개선하기 위해 온라인으로 협업과 관리 모두 할 수 있는 개발 프로세스가 필수적이다.

4. 개발 프로세스 개선방안

클라우드소싱 기반 소프트웨어 개발 문제점 5가지를 개선하기 위해 단계적이고, 소규모 팀 프로젝트에 적합한 애자일 개발 방법론인 스크럼의 스프린트에 테스트 주도 개발 방법(TDD)[8]과 스크럼의 큰 틀을 온라인 프로젝트 관리 툴인 Trello로 도구화 및 시각화한 프레임워크를 제시한다[9].

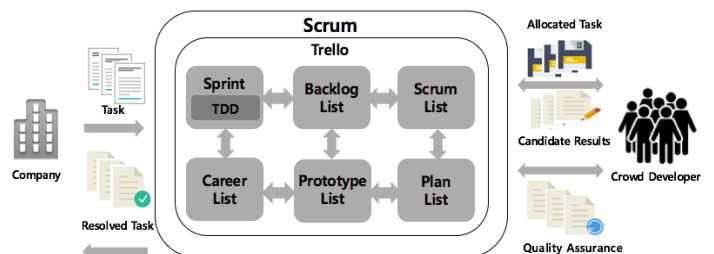


그림 3. 개발 프로세스 개선을 위한 프레임워크

그림 3의 제시된 프레임워크는 Trello에 스크럼의 제품 백로그, 스프린트 백로그, 스프린트 계획 회의, 일일 스크럼 회의, 스프린트 회고 등을 리스트로 만들어서 프로젝트 요구사항 변경에 대처하고, 프로젝트의 계획, 관리, 커뮤니케이션, 개발 동기, 보상 등의 문제를 해결해 개발 프로세스를 개선한다.

1. **개발 동기와 합당한 보상 문제:** Trello의 체크 리스트 기능과 라벨을 각 스크럼에 리스트에 적용해 개발 동기와 합당한 보상 문제를 개선한다. 체크리스트는 작업의 목표에 대한 할당량을 체크함으로써 매주, 매일 개발자 한 일의 양을 수치화 시키고, 라벨을 통해 업무의 복잡도를 나타내서 일의 어려움 정도를 측정할 수 있다. 또한 이를 통해 보상에 대한 기준을 확실히 세울 수 있다. Trello에서 추후 자신의 개발 커리어를 증명할 수 있는 증거가 되므로 적절한 동기부여를 통해 개발 동기와 합당한 보상 문제를 개선할 수 있다.
2. **커뮤니케이션 문제:** Trello에서 스크럼 리스트를 활용해 개발 프로세스 동안 커뮤니케이션 문제를 개선한다. 고객이 Trello의 사용하기 쉽고, 편리한 인터페이스를 이용해 제품 백로그, 실시간 개발 진행 상황, 결과 등을 보면서 피드백이 가능하기 때문에 고객과 개발자 사이에 정확한 커뮤니케이션을 제공할 수 있다. 또한 개발자들은 스크럼의 도구화 및 시각화된 Trello에서 의견교환과 프로젝트 진행사항, 각자 맡은 일, 해야할 일, 했던 일 등을 스크럼 리스트에서 직접 보면서 커뮤니케이션이 가능하기 때문에 보다 정확하게 서로의 의견을 전달 할 수 있다.
3. **개발 계획 문제:** Trello에서 제품 백로그 리스트를 활용해 프로젝트 개발 계획의 문제점을 개선한다. Trello에서 고객과 개발팀이 제품 백로그 피드백을 반영해 변화된 요구사항에 적절히 대처할 수 있다. 또한 번다운 차트를 활용해 프로젝트의 진척도를 확인하고, 이를 이용해 개발 계획을 한눈에 볼 수 있도록 시각화 하여 개발 계획의 차질이 없는지 확인이 가능하다.
4. **품질 보장 문제:** 스프린트에 테스트 주도 개발 방법을 도입하고 Trello의 일일 스크럼 회의 리스트를 활용해 품질 보장 문제를 개선한다. 스프린트에 테스트 주도 개발을 도입 함으로써 유닛 테스트를 작성하고 이를 통과할 수 있는 코드를 계속 작성하는 방법으로 디버깅에 소요되는 시간과 비용을 단축시켜 오류를 점진적으로 감소시킨다. 또한 지속적으로 프로토타입을 제공해 프로젝트의 방향을 잃지 않도록 하고, Trello의 일일 스크럼 회의 리스트와 스크럼 회고 리스트를 활용해 스프린터 주기 동안 어제 한일, 오늘 할일, 문제점 분석 등을 통해서 지속적으로 소프트웨어의 품질을 개선해 나간다.
5. **프로젝트 분배 문제 개선방안:** Trello에 참여하는 각 개발자의 커리어 리스트를 만들어서 프로젝트 분배 문제점을 개선한다. 개발자의 커리어 리스트를 통해 개발언어, 참여 프로젝트, 전공분야, 관심분야 등을 리스트에 더해서 개발자 간의 개발 수행 능력을 비교하고 등급을 나누어 등급과 개발 복잡도에 따라 프로젝트를 분배해서 문제점을 개선한다.

클라우드 소싱 기반 소프트웨어 개발 프로세스는 다양한 지성이 모여서 공동으로 작업을 하기 때문에 효과적으로 협업과 관리가 필요하다.

따라서 애자일 개발 방법론 중 스크럼을 변형해서 테스트 주도 개발 방법을 스프린트에 도입하고, 스크럼의 큰 틀을 온라인 프로젝트 관리 툴인 Trello로 도구화 및 시각화해서 2장의 클라우드소싱 기반 소프트웨어 개발의 문제점 5가지를 개선하는 개발 프로세스를 제안했다. 개선방안에서 가장 중요한 부분은 클라우드소싱의 특성과 개발 프로세스를 모두 고려해 개발 프로세스를 개선하는 것이다. 특히 클라우드소싱을 개발 프로세스에 도입하기 위해 Trello를 사용함으로써 개발자 사이의 커뮤니케이션이 쉬워지고, 다양한 개발 프로세스를 적용시킬 수 있도록 하여 고려사항을 만족시킬 수 있었다.

5. 결론 및 향후 연구방향

클라우드소싱은 불특정 다수를 선택함으로써 다양하고 많은 정보를 얻어 프로젝트의 비용을 낮추고, 문제 해결 소요 시간을 단축시킬 수 있는 장점이 있다. 현재는 이를 활용하는 기업이 점차 늘어나고 있는 추세이다. 따라서 본 논문에서는 클라우드소싱을 활용한 소프트웨어 개발의 장점과 활용 사례를 소개하고, TopCoder의 사례를 통해 문제를 분석하여 이를 해결하는 개발 프로세스의 개선방안을 제시했다.

개선된 개발 프로세스는 오픈소스인 Trello에 적용하기 때문에 확장성과 접근성이 뛰어나다는 장점이 있다. 이런 장점을 이용해 기업의 클라우드소싱 기반 소프트웨어 개발 환경에 맞도록 변형시킬 수 있어 소프트웨어를 더욱 효율적으로 개발할 수 있다. 또한 온라인으로 개발 프로세스를 진행할 수 있기 때문에 시간과 위치에 제한 없이 다양한 클라우드소싱 기반 소프트웨어 산업에 활용될 것이다.

향후 본 논문의 클라우드소싱 기반 소프트웨어 개발 프로세스 개선 방안을 적용한 온라인 및 모바일 협업 관리 어플리케이션을 개발해 그 효율성을 입증하여 다양한 소프트웨어 산업에 적용시키고자 한다.

참고문헌

- [1] Eric Bonabeau, "Decisions 2.0: The Power of Collective Intelligence", MIT Sloan Manage Rev, Vol. 50, No. 2, pp. 45-52, 2009
- [2] Daren C. Brabham, "The Myth of Amateur Crowds: A critical discourse analysis of crowdsourcing coverage", Information, Communication & Society, Vol.15, No.3, pp.394-410, 2012
- [3] Klaas-Jan Stol, Brian Fitzgerald, "Researching Crowdsourcing Software Development: perspective and concern", In Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering, pp.7-10, 2014
- [4] Panagiostis G. Ipeirotis, "Analyzing the Amazon Mechanical Turk marketplace", XRDS: Crossroads, The ACM Magazine for Students, Vol.17, No.2, pp.16-21, 2010
- [5] Karim R. Lakhani, David A. Garvin, and Eric Lonstein, "TopCoder (A): Developing Software through Crowdsourcing", Harvard Business School 610-032, 2010
- [6] Ke Mao, Ye Yang, Mingshu Li and Mark Harman, "Pricing Crowdsourcing-Based Software Development Tasks" In Proceeding of the 2013 International Conference on Software Engineering, pp.5-8, 2013
- [7] Klaas-Jan Stol, Brian Fitzgerald, "Two's company, three's a crowd: a case study of crowdsourcing software development", In Proceedings of the 36th International Conference on Software Engineering, pp.187-198, 2014
- [8] Sok Kee, Neunghoe Kim, Donghyun Lee, Hoh Peter In, "A Scrum Development Process of an Agile Methodology using Concurrent Testing", Journal of KIISE: Software and Application, Vol.38, No.2, pp.84-90, 2012
- [9] Aaron C Ault, James V Krogmeier, Dennis Buckmaster, "Mobile, Cloud-Based Farm Management: A Case Study with Trello on My Farm", Paper-American Society of Agricultural, pp.51-64, 2013