

웹의 플랫폼화에 따른 플랫폼 및 소프트웨어개발 전략

Platform Strategies and Software Development Strategies in the era of Web Platform

백영란(Young Nahn Baek)

한국소프트웨어진흥원 정책연구센터

서울특별시 송파구가 락본동 79-2 KIPA 빌딩

Tel: +82-2-2141-5300, Fax +82-2-2141-5349, E-mail: ynbaek@software.or.kr

초록

본 연구는 웹의 플랫폼화에 대응하기 위한 서비스 유통 및 소프트웨어 개발방식의 핵심이슈로 ‘규모의 경제’와 ‘서비스화’이라는 모순적인 요구를 동시에 해결하는 것임을 제시했다. 소프트웨어 관점에서 본 이른바 웹 2.0 시대의 주요한 특징은 웹의 플랫폼화와 롱테일 서비스 시장의 등장이다. 웹이 독립적인 유통채널로 기능하게 되는데, 본 연구는 개발자 및 사용자 커뮤니티를 구축하기 위한 플랫폼전략을 규모의 경제와 상호소통(Interaction)의 관점에서 분석했다. 플랫폼 전략에서 제기된 이슈를 해결하기 위해서는 소프트웨어 개발전략도 오픈소스 커뮤니티를 활용한 레고형 개발이 필요하다. 비용절감과 사용자 채널을 확보하기 위해서는 오픈소스 커뮤니티를 이용한 조직화방식이 필요하고, 고객화된 서비스의 가격경쟁력을 확보하기 위한 레고형 개발 방법이 필요하다.

Abstract

Current strategic trends in service distribution and software development in response to Web as a platform represent a paradoxical concurrent pursuit of ‘economy of scale’ and ‘service’. From the software perspective, the major novel elements of the Web 2.0 era are Web as a platform and the emergence of long-tail service enabling web sites to function as an independent distribution channel. The current study analyzed the platform strategy for establishing a community of interactive developer and customers taking the issue of economy of scale into consideration. Software industry should face the new reality and utilize the open source community for cost minimization and securing customer base. Furthermore, Lego-style software development should be adopted for maintaining cost-competitiveness in customized software development.

Keyword: *Web Platform; Long-tail; SaaS; Open Source Community; Lego-like module*

1. 서론

웹 2.0 으로 표현되는 웹의 진화는 웹이 다양한 콘텐츠 서비스의 플랫폼으로 기능하게 됨을 의미한다. 독자적인 생태계를 구축하고자 하는 기업전략이 성립하기 위해서 웹에 적용한 플랫폼전략이 핵심이다. 단순히 소비자가 원하는 서비스나 솔루션이 아니라 그 위에서 새로운 서비스와 솔루션을 추가할 수 있는 플랫폼을 제공해야 한다. 플랫폼 전략이 성공하기 위해서는 플랫폼을 지원하는 소프트웨어가 핵심경쟁력이 된다. 실제 웹의 진화에 따라 콘텐츠나 서비스를 개발하기 위한 웹 애플리케이션(Rich Internet/interactive applications, 이하 RIAs)과 온라인으로 서비스되는 소프트웨어(SW as a Service, SaaS) 등 웹 기반 소프트웨어가 등장 하고 있다.

본고는 웹 플랫폼 전략의 성공을 이끄는 웹 기반 소프트웨어의 개발전략에 대해 분석하고 있다. 웹 2.0 이란 환경이 소프트웨어시장에 미치는 핵심이슈를 도출하고, 이에 기반 한 소프트웨어 개발전략을 제안한다. 우선 플랫폼의 의미 및 웹 플랫폼 전략에 대해 살펴보고, 그 다음으로 서비스 플랫폼 전략의 핵심요소로 규모의 경제와 상호소통(Interactivity)을 제시하고, 마지막으로 소프트웨어 개발방식의 핵심이슈인 ‘규모의 경제’와 ‘서비스화’이라는 모순적인 요구를 해결하기 위해 방안으로 오픈소스 커뮤니티를 이용한 개발방식과 고객화된 레고형 소프트웨어개발방법을 제안한다.

2. 웹 플랫폼

웹 2.0 의 핵심적 의미는 웹의 플랫폼화(Web as a Platform)이다. 포탈기업은 웹을 기반으로 다양한 서비스를 제공하고 있고, API 를 공개하여 자사의 생태계를 만드는 플랫폼전략을 구사하고 있다. 이러한 플랫폼전략의 핵심요소를 도출하기 위해 우선 플랫폼이란 개념정의와 전통적인 OS 기반 플랫폼 전략과의 동일성과 차별성을 분석해보도록 한다.

2.1 플랫폼의 개념

플랫폼의 사전적 의미는 어떤 목적을 위해 놓여진 단상이나 정치적 원칙이다. 이를 IT 의 관점에서 해석해보면, 플랫폼은 수직적 스택(stack)구조에서 수평적 레이어를 우선 의미한다. 또한 마치 대중을 향한 자신의 정치적 원칙을 표명하듯이, 그 수평적 레이어 위에서 솔루션이 개발되기 위한 인터페이스(interface)의 공개를 의미한다. 이를 하나로 정리해보면 플랫폼은 “솔루션이나 콘텐츠가 개발될 수 있도록 제공되는 인터페이스의 집합”(백영란, 2007)이라고 정의될 수 있다.

이에 따르면 서로 다른 소프트웨어의 스택에서 각각의 레이어들은 서로 다른 레이어의 플랫폼이 된다. 예를 들면 CPU는 OS 의, OS 는 애플리케이션 서버의, 애플리케이션 서버는

웹 기반 애플리케이션의 플랫폼이다. 플랫폼을 하부기반(Infrastructure)과 구분해서 사용하는 것은 플랫폼이 IT 산업의 다이내믹스를 나타내는 특별한 의미가 있기 때문이다. 없으면 안 되지만 가치생산에 있어서 핵심적 고리는 아닌 하부기반(Infrastructure)과 달리, 플랫폼은 인터페이스 정책이라는 수단을 가지고 생태계를 구축하는 산업적 영향력을 행사하고 있다.

2.2 PC 시대 OS 플랫폼 전략

플랫폼 전략이 기존전략이 된 PC 시대 플랫폼의 중심은 OS 이다. MS 는 OS 란 플랫폼을 지배함으로써 IT 산업에 대한 영향력을 확보할 수 있었다. MS 의 플랫폼 지배는 자사의 OS 를 둘러싼 하나의 생태계를 구축하였기 때문이다.(백영란, 2005) 인터페이스의 집합으로써 플랫폼은 제품과 서비스간의 상호의존성을 전제로 한다. 그리고 그 상호의존성이 기업 내의 관계가 아니라 외부적인 관계일 때 비로소 의미가 있다. MS 는 자사의 OS 기반 위에 응용프로그램이 만들어 질 수 있도록 API(Application Program Interface)를 공개함으로써, 소프트웨어산업의 수직적 분업을 이끄는 자사의 생태계를 구축하였다.

API 를 공개수준을 정할 수 있는 권한이 실질적 영향력을 가지기 위해서는, 규모의 경제에 기반 한 생산-소비-유통의 네트워크를 만들어야 한다. 우선 자사의 플랫폼인 OS 자체가 시장에서 지배력이 있어야 하고, 공개된 API 에 기반 해 응용프로그램을 만드는 개발자와 파트너 기업들이 다수 존재하여야 한다. 이러한 관계의 네트워크를 통해 바로 제품이 아니라 스탠더드를 만들어내기 때문이며, 역으로 스탠더드는 네트워크를 통해서만 창출될 수 있고 유지될 수 있기 때문이다. 이러한 선순환의 구조를 갖는 생태계를 통해 MS 는 PC 시대 진정으로 유일한 소프트웨어플랫폼기업이 될 수 있었다.

2.3 웹 플랫폼

그렇다면 웹이 플랫폼화 된다는 의미는 무엇인가? 위에서 살펴본 전통적인 플랫폼의 해석에 기초하면, 웹 플랫폼이란 웹에 기반 한 솔루션이나 콘텐츠를 개발하기 위해 공개되는 인터페이스의 집합이다. 또한 API 를 공개하여 소비자, 개발자, 관련 기업들을 네트워크로 묶고자 하는 생태계가 출현한다는 의미가 더해질 수 있다. 웹이란 용어가 다소 부정확한 표현이지만, 실제 웹이 플랫폼화 되는 다양한 사례들이 등장하고 있다.

우선 최근 다양한 웹기반 서비스가 활성화되면서 웹 기반 애플리케이션의 기술적 토대가 되는 콘텐츠 신디케이션(Content Syndication), 매쉬업(Mashup) 같은 다소 일반적 기술이나 Ajax, REST, RSS 등 특정한 기술들이 등장하고 있다. 전통적인 스택의 패키지화된 특정 소프트웨어와는 다르지만, 클라이언트 단 또는 서버 단에 위치한 이들 웹을 둘러싼 기술의 집합자체가 하나의 레이어와도 같은 의미를 가지게 되었다.

이처럼 웹이 하나의 하부기반(Infrastructure)이 되면서 API 공개를 통해 자사가 중심이 된 생태계를 구축하려는 다양한 포탈들이 등장하고 있다. 아마존, 이베이, 구글,

세일즈포스닷컴 등 수많은 다양한 포탈들은 자사의 데이터, 서비스, 솔루션을 공개하여 소비자가 이들을 조합하여 새로운 기능과 서비스들을 만들 수 있도록 하고 있다. 자사의 생태계를 구축하고자 하는 이들 기업의 전략이 성립하기 위해선 우선 플랫폼을 제공해야 한다. 단순히 소비자가 원하는 서비스나 솔루션이 아니라 그 위에서 새로운 서비스와 솔루션을 추가할 수 있는 플랫폼을 제공해야 한다. 구글의 경우 가장 성공적인 서비스인 검색엔진 그 자체만으로는 플랫폼이 될 수 없지만, 다양한 매쉬업 서비스를 가능하게 하는 구글 맵(Maps)은 플랫폼으로 기능하고 있다.

3. 서비스 플랫폼전략: 규모의 경제와 상호소통

웹 2.0 의 가장 중요한 의미는 소비자의 참여에 의한 시장 확대이다. 변화의 핵심은 이른바 IT 의 소비자화라고 말해지는 데, 소비자 시장으로 시장이 확대될 뿐만 아니라 소비자가 시장의 주체로 등장한다는 의미이다. 이에 따라 서비스 플랫폼 전략에서는 시장확대를 이끄는 규모의 경제와 소비자 참여를 보장하는 상호소통이 핵심요소로 등장하게 된다.

3.1 규모의 경제

크리스 앤더슨이 유행시킨 롱테일 경제는 “웹이란 환경에서 기술이 이루어낸 새로운 시장을 의미”한다. 블록버스터를 중심으로 파는 월마트 같은 전통적 소매상이 비용 때문에 다가갈 수 없었던 시장을 아마존 같은 온라인 소매상은 공략할 수 있었다. 새로운 니치(niche)시장으로 진입할 수 있었던 것은 소비자가 수없이 많은 다양한 제품을 손쉽게 찾을 수 있게 한 웹 기술 때문이다. 이에 따라 20%의 제품에서 80%의 수익을 내는 이른바 파레토법칙에 매이지 않은 새로운 수익원을 창출할 수 있었다.

롱테일 경제는 오프라인 경제에서 온라인 경제로의 패러다임 전환이다. 이 전환이 가능했던 것은 소비자가 원하는 제품 또는 서비스를 적절한 가격에 제공할 수 있었기 때문이다. 특히 온라인 소매시장의 경우 비용감소, 곧 거래비용의 감소가 핵심동인이었다. 오프라인 경제에서 전통적인 미디어에 의존한 값비싼 광고나 전시된 제품을 통해서만 소비자는 시장정보를 얻을 수 있었다. 이제 소비자는 구글의 검색엔진을 통해, 소비자에 맞춤형으로 제공되는 애드센스 같은 광고를 통해, 다양한 커뮤니티를 통해 시장정보를 수집하고 처리할 수 있게 되었다. 이에 따라 시장정보를 둘러싼 거래비용(transaction cost)이 인터넷 경제에서는 현격하게 감소하게 되었고, 새로운 시장을 공략할 수가 있게 되었다.

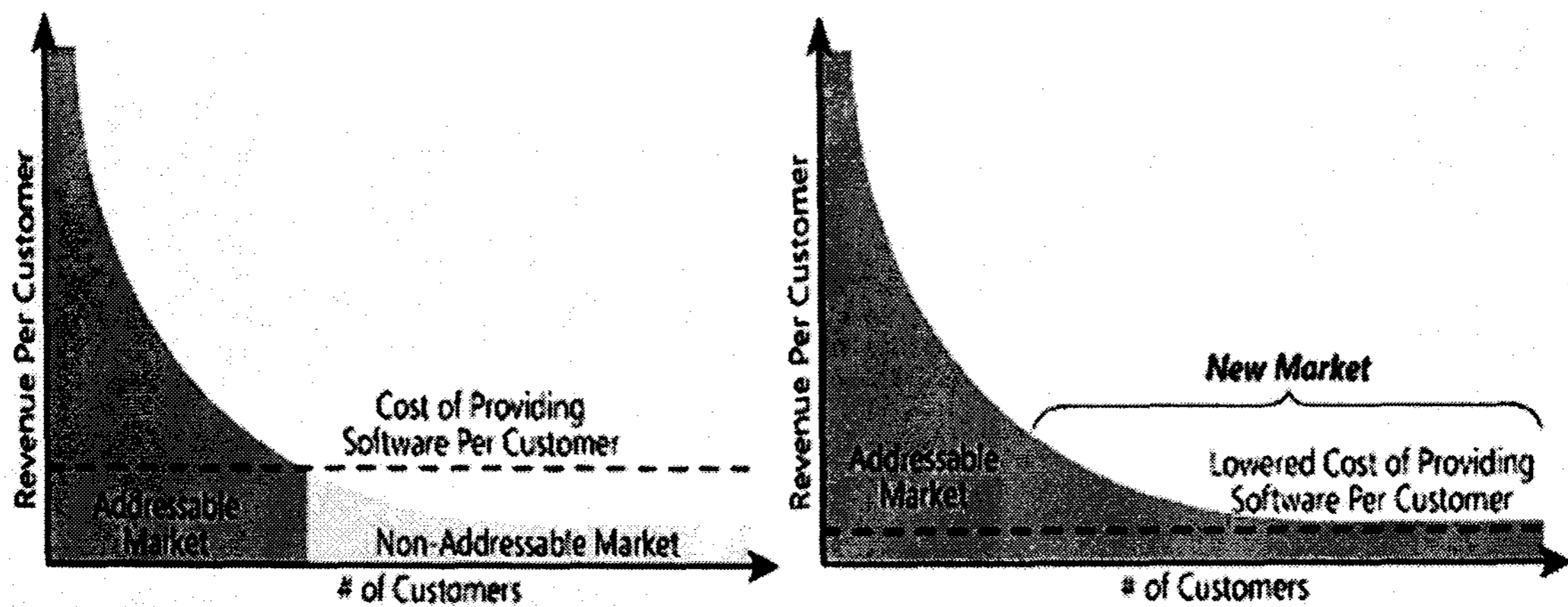


그림 1- 롱테일 시장기회와 비용감소

한편 소프트웨어시장에서는 아직 주목하지 않았던 롱테일 시장이란 이른바 SaaS 가 만드는 새로운 시장이 등장할 것이다. 웹 오피스 같은 일반소비자를 대상으로 하는 시장뿐만 아니라 특히 중소기업시장이 중요해진다. 실제 소프트웨어시장에서 중소기업의 시장잠재력은 엄청나다고 전망된다. 미국에서 대기업의 경우 평균 IT 투자 2 백만 달러인데 비해, 중소기업과 소기업은 각각 14 만 달러와 2 천 달러에 불과하다. 이들 중소기업은 글로벌 GDP 의 약 60%정도를, 고용의 70%를 차지하고 있기 때문에 중소기업의 시장기회는 앞으로도 엄청나다. 특히 웹 2.0 시대 새로운 사업기회를 찾으려는 미디어, 엔터테인먼트, 소매상 기업 등을 공략한다면, 1500 억 달러의 롱테일 시장이 있다고 분석되기도 한다.(Frank Gens, 2006)

롱테일로의 시장확대를 위해서는 규모의 경제에 기반해야 한다. 구글의 검색광고에서 볼 수 있듯이 오늘날 검색포탈의 비즈니스 모델은 대규모의 트래픽에 기반 한 광고모델이다. 이에 따라 대규모생산에 따른 비용감소, 곧 규모의 경제를 반드시 필요로 한다. 이때 특히 소프트웨어자체의 가격경쟁력이 중요해진다. 소프트웨어는 거래비용을 감소시키는 기술적 수단이거나 소비자를 유인하는 부가서비스로 제공된다. 이 경우 소프트웨어는 생산수단적인 의미를 가지게 되는데, 비용절감을 위해 소프트웨어가격은 중요한 의미를 지닌다. 또한 유료로 서비스되는 중소기업을 대상으로 한 SaaS 시장에 진입하기 위해서는 특정 서비스를 적절한 가격에 제공할 수 있어야 한다. 곧 소프트웨어가 생산수단 혹은 부가서비스가 되는 경우나 완제품으로 제공되는 서비스인 경우 모두 소프트웨어자체의 가격하락이 관건이 되어가고 있다.

3.2 상호소통

웹 2.0 의 기본 철학은 인터넷을 통한 상호작용(Interaction)이다. 사람들은 서로 대화하고 협업하기 위해 소프트웨어에 의존한다. 소비자는 소프트웨어 그 자체가 아니라 소프트웨어를 통해 제공되는 서비스에서 사용가치 그리고 교환가치를 찾는다. 웹 기반

서비스를 제공하는 수단으로써 소프트웨어의 새로운 역할을 이해하기 위해 소프트웨어의 사용목적은 보다 자세히 분석해보자.

사람들의 소프트웨어에 대한 필요는 '소프트웨어를 통한 상호작용'(interaction through software)과 '소프트웨어와의 상호작용'(interaction with software)으로 구분된다. 사용자 사이에 상호작용을 목적으로 하는 기술적 수단을 제공하는 소프트웨어로는 협업 도구인 위키(wiki)나 사진 등을 공유하는 수단인 플릭커(flicker) 등이 있다. 한편 사용자가 소프트웨어사용자체에 포커스를 두는 Ajax 나 Flash 등은 RIAs(Rich Internet Applications) 나 전통적인 응용프로그램의 온라인 진화형태인 SaaS 등이 있다.

상호소통을 위한 그 어떤 경우든 소프트웨어자체가 사용자의 참여를 목적으로 하기 때문에, 상호소통의 원리가 소프트웨어에 내재되어 있어야 한다. 곧 사용자가 대상이 아니라 주체가 되어야 한다. 이는 사용자가 반드시 소프트웨어 개발자체에 참여해야 한다는 의미는 아니지만 지속적으로 사용자와의 의사소통이 위해서는 사용자 커뮤니티가 조직화될 필요가 있다는 의미이다.

한편 이러한 소프트웨어의 수요변화는 서비스와 소프트웨어의 융합을 이끈다. 소프트웨어와 서비스의 융합은 다양한 웹 기반 서비스와 소프트웨어의 융합, 패키지 소프트웨어와 IT 서비스의 융합으로 구별될 수 있다. 이제 소프트웨어는 검색, 전자상거래, 온라인 음악서비스, 사회네트워크(Social Network) 등 서비스 자체와 분리될 수 없다. 또한 다수의 소비자를 위한 제품으로써 패키지 소프트웨어와 하나의 소비자를 위해 최적화된 서비스를 제공하는 IT 서비스라는 경계가 융합되고 있다. 이에 따라 대량생산을 위해 표준화해야 하는 제품의 특성과 사용자의 특수한 요구를 반영하는 서비스라는 전혀 다른 요구를 조화시켜야 하는 이슈가 등장하게 된다.

4. 소프트웨어개발방식의 변화

그 동안 패키지 소프트웨어 시장에서 전형적인 소프트웨어 개발방식은 폐쇄적인 기업 내 개발조직에 의해 대량판매를 위해 소프트웨어를 제품화하는 과정이었다. 그러나 규모의 경제에 기반한 가격경쟁력을 가지면서도 동시에 상호소통을 위한 서비스로써 소프트웨어에 대한 새로운 수요는 소프트웨어 개발방식의 변화를 이끌 수 밖에 없다. 본고는 그 변화를 오픈소스 커뮤니티를 활용한 개발의 조직화, 레고형 개발방법으로 설명하고자 한다.

4.1 개발의 조직화: 오픈소스 커뮤니티 활용

오픈소스가 강조하는 지식공유와 교환의 모델은 소프트웨어 개발방식의 새로운 모델로 받아들여지고 있었고, 최근 웹 2.0 조류에 따라 더욱 더 주목을 받고 있다. 오픈소스는 말 그대로 소스를 공개하였기 때문에 누구라도 소스코드를 가지고 버그를 수정하고, 자신의 목적에 맞게 코드를 수정 사용할 수 있는 SW 개발과정의 공유 곧 협업을 할 수 있는 기본조건을 제공한다.

그러나 정보공유로써 코드공유는 완전히 새로운 것은 아니다. 그것은 이미 1960년대 초반 MIT와 버클리의 컴퓨터 랩에서 볼 수 있듯이 SW 개발에서 오래된 전통이었다. 70년대 중반 이후 독립적인 패키지 SW 산업이 독자적인 영역으로써 자리를 굳히면서 소프트웨어 개발방식도 변화하였다. 1980년을 전후하여 SW가 독자적인 지적 재산권으로 인정 받게 되고, 기업 내에 R&D 투자에 기반한 폐쇄적인 소프트웨어 개발방식이 중심이 된다.(백영란, 2004) 그런데 기업 내 방대한 조직에서 장기간으로 수행되는 대규모 R&D 투자에 대한 한계가 지적되면서, 오픈소스 프로젝트가 새로운 대안으로 부상하고 있다.

특히 최근 지식의 공유와 교환을 통해 이룬 오픈소스 프로젝트의 성공과 확산은 조직화와 동기부여에 있어 새로운 모델을 제공하고 있다. 레이몬드(E. Raymond)는 소프트웨어 개발방식을 성당과 시장이라는 두 가지 방식, 곧 폐쇄적인 기업 내 위계질서화 된 개발과 오픈소스 프로젝트 방식으로 분류했다. 오픈소스는 열린 시장모델의 이점, 개방성과 협업적 분업에 기초한 개발방식의 유용성을 보여주고 있다. 오픈소스 개발방법은 다수의 개발자가 병행적으로 프로젝트에 참여하고 있으며, 강력한 리더의 존재나 조직화된 동료집단의 커뮤니케이션에 의해 프로젝트를 위계질서화하고 중앙으로 집중화하는 메커니즘이 있다.

또한 오픈소스 프로젝트는 경제적 보상 이외에 비경제적 동기부여의 중요성을 환기시킨다. 오픈소스에 개발에 참여하고 있는 상당수의 개발자들은 이미 자신의 직업을 가진 전문적인 개발자들이다. 이들이 오픈소스 프로젝트에 참여하는 이유는 새로운 기술을 학습하고 적용하기 위해서, 기존 직장에서 채울 수 없는 자기성취욕구 때문에, 동료들의 인정과 프로젝트의 성공으로 개인적 명성을 얻을 수 있기 때문에 등 다양하다. 이러한 오픈소스 참여자의 성격과 행태는 경제적 보상 이외에 다양한 동기부여 메커니즘의 중요성을 말해준다.

실제 소프트웨어 기업들이 오픈소스 프로젝트의 참여와 활용이 늘어나고 있고, 특히 다양한 웹 관련 소프트웨어는 오픈소스 프로젝트에 의해 개발된 것이 많다. 기업들이 오픈소스 생태계에 참여하는 인센티브는 '커뮤니티 소스'를 통해서 비즈니스 기회가 확보될 수 있기 때문이다. 웹 2.0의 조류 속에서 특히 강조되는 인센티브는 소프트웨어개발에 있어 비용절감 효과와 사용자와의 채널확보이다.

오늘 날 오픈소스 커뮤니티는 하나의 생태계를 이루고 있는데, 오픈소스 생태계는 다양한 새로운 비즈니스 모델을 추구하는 다양한 참여자를 통하여 서비스를 제공하는 총체로써 설명될 수 있다. 오픈소스 생태계에는 커뮤니티(개발자 커뮤니티, 사용자 커뮤니티), 재단(Foundation), 컨소시엄, 기업, 정부 등으로 구성되어 있고, 이들 개별 주체들은 상호작용을 하면서 오픈소스 소프트웨어의 발전을 견인한다.

우선 이 생태계에서 소프트웨어 기업들은 외부 커뮤니티를 활용하여 궁극적으로 소프트웨어 개발의 비용절감 효과를 이룰 수 있다. 소프트웨어 기업들은 오픈소스프로젝트를 지원하고 기술개발을 흡수하면서 상업적인 제품화와 서비스를 개발하고 있다. 소프트웨어 기업들은 오픈소스 프로젝트를 지원하여, 제품라인에 쓸 수 있는 기반기술을 개발하거나 부분기술로 활용하고 있다. 또한 기업들은 산재해 있는 오픈소스 어플리케이션 관련 원천기술들을

표준화하여 제품화하는 하는 것도 가능하다. 물론 기업들의 차별화를 위해 기업전략에 의해 정해진 핵심 부분은 내부화하고 있다. 다양한 동기에 의해 작동하는 보다 많은 개발자가 참여하여 네트워크화된 외부 커뮤니티를 이용함으로써, 기업들의 기존에 기업내부에서 이루어지던 소프트웨어 개발방식보다 비용절감을 이룰 수 있다.

또한 오픈소스 생태계를 통해서 기업이 사용자와의 상호작용 할 수 있다는 점도 중요하다. 오픈소스 커뮤니티에서는 '사용자=개발자'의 공식이 통용될 수 있고, 최근 오픈소스 프로젝트는 전문적인 사용자라고 할 수 있는 개발자 커뮤니티 외에 보다 아마추어적인 말 그대로 사용자 커뮤니티가 또한 조직화되고 있다.(문장원, 2006) 커뮤니티를 통해 사용자의 의견과 경험이 바로 개발자에게 바로 피드백 될 수 있으므로, 사용자가 소프트웨어개발과정에 있어 참여하고 있다고 해도 무방하다. 곧 오픈소스 프로젝트에서는 커뮤니티자체가 사용자와의 채널역할을 구조적으로 하고 있다.

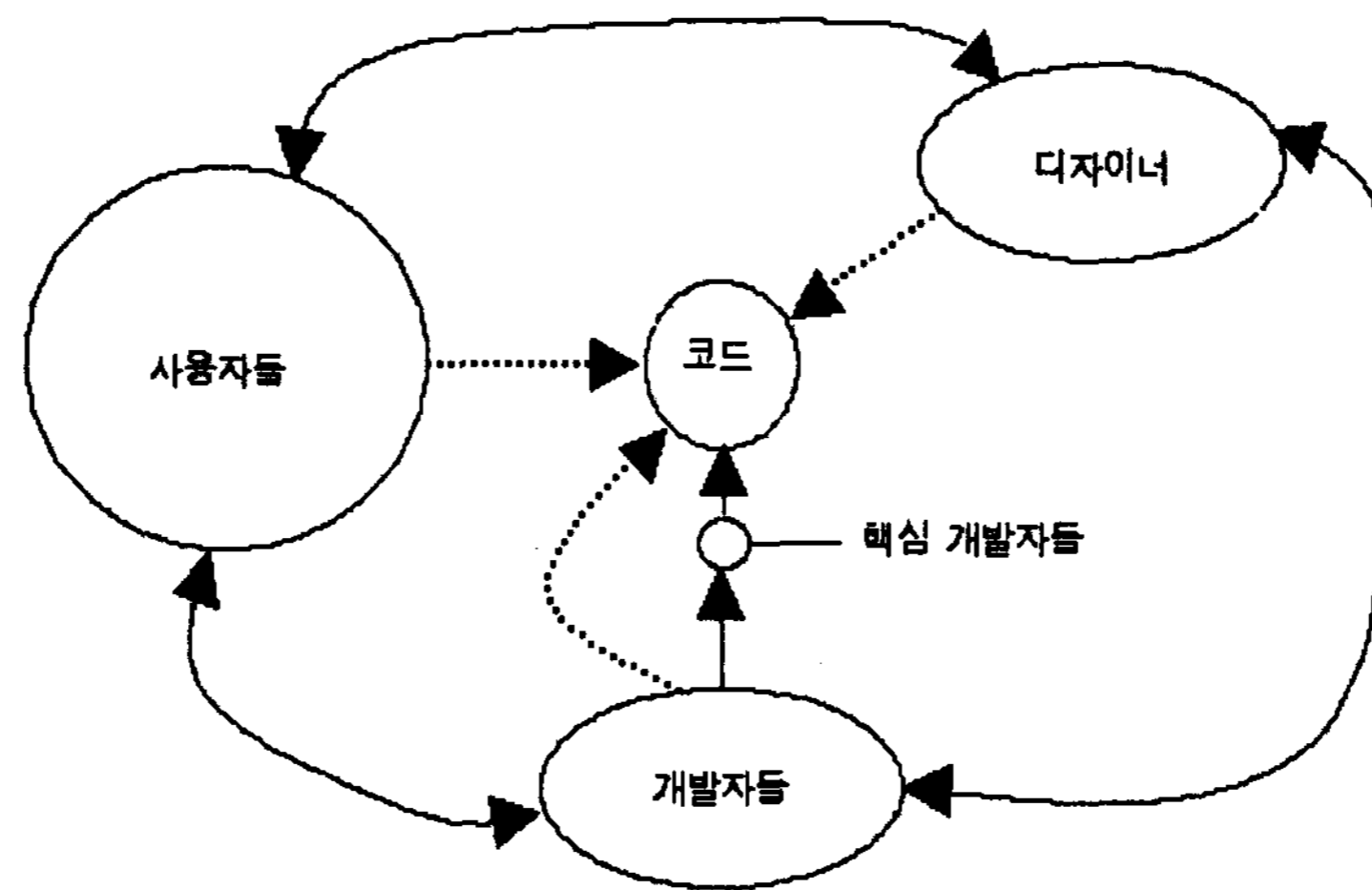


그림 2: 오픈소스 생태계에서 사용자들의 역할(Ron Goldman, 2005)

4.2 개발방법론: 레고형 개발방법

오픈소스 프로젝트가 소프트웨어 개발의 조직화방식을 바꾼 것이라면, 소프트웨어 개발방법론 또한 변화할 필요가 있다. 그 동안 패키지 소프트웨어의 발전은 하나의 제조업으로써 소프트웨어를 산업화하는 과정이었다. 사용자의 참여라는 비즈니스적 요구에 부응하기 위해 소프트웨어 개발방법이란 관점에서 도전은 제품과 서비스를 융합해야 한다는, 곧 제품과 서비스의 모순을 해결해야 한다는 점이다. 제품으로써 패키지 소프트웨어 개발시 달성한 규모의 경제를 유지하면서도, 사용자의 특수한 요구를 반영하는 서비스로써 또한 개발되어야 한다는 뜻이기도 하다.

소프트웨어 개발이 수공업처럼 하나씩 하나씩 만들어지는 과정이 아니라, 마치 제조업 공정처럼 재사용될 수 있는 부품화되어 조립 가능한 표준화된 개발방식을 갖는 것은 오랜 꿈이었을 뿐만 아니라 현실적 목표이었다. 분명 그것은 규모의 경제를 달성하기 위한, 곧 대량생산을 통한 가격경쟁력의 확보하는 비즈니스적 목표였다. 실제 소프트웨어

개발방식에서 규모와 범위의 경제는 중요한 이슈였다. 규모의 경제(Economies of Scale)는 동일한 모델이 다량으로 생산될 수 있다는 뜻이며, 한편 범위의 경제(Economies of Scope)는 유사하나 구별되는(distinct) 모델이 다량으로 생산될 수 있다는 뜻이다. 그 동안 일반소비자를 위한 데스크 탑 시장에서는 규모의 경제에 기반해 개발되었고, 기업을 위한 응용프로그램 시장에서는 범위의 경제가 제한적으로 존재하였다. (Jack Greenfield, 2004) 그러나 물리적인 생산과정이라기 보다는 논리적인 개발과정이라는 차별성 때문에 언제나 혼돈의 과정이기도 하였다.

지금은 서비스 시대이다. 따라서 똑같은 제품이 아니라 소비자에 따라 달라지는 서비스에서 규모의 경제를 추구해야 한다. 그러나 제품생산에서 말하는 대량생산을 위한 표준화와 하나의 고객을 위한 커스토마이제이션(customization)은 서로 개념적으로 모순적이라고 볼 수 있다. 따라서 이 모순을 해결하기 위해선 고객의 수요를 맞추는 방식이 변화해야 한다. 규모의 경제를 위해서 필연적인 모듈화에는 두가지 방식이 있다. 곧 조각 그림 맞추기(jigsaw puzzle)처럼 한 가지 그림만을 맞출 수 있는 모듈화와 레고처럼 개인이 자유롭게 모양을 만드는 모듈화가 되어야 한다. 서비스시대에 맞는 규모의 경제를 달성하기 위해선 소프트웨어개발이 레고형의 모듈화가 필요하다.

예를 들면 SaaS 를 위한 소프트웨어개발방식은 패키지 소프트웨어를 개발하는 방식과 달라져야 한다. 우선 동시에 요구되는 서비스를 지원할 수 있도록 확장가능(scalable)해야 뿐만 아니라 서로 다른 고객이 서로 다른 데이터를 이용해 서비스 받을 수 있도록 효율적으로 자원을 공유(multi-tenant-efficient)할 수 있어야 한다. 한편 서비스의 개념과 모듈화 자체가 변화하는 것이 본질적으로 중요한 점이다. 커스토마이제이션(customization)이라는 서비스 개념이 바뀌어져야 한다. 기존 서비스처럼 하나의 고객을 위한 커스토마이제이션(customization)을 하게 된다면 다른 고객을 위해서는 또 다른 프로그램이 필요로 하기 때문이다. 오히려 사용자들이 스스로 프로그램의 인터페이스와 작동방법을 설정(configuration)할 수 있도록 서비스를 설계해야 한다. 또한 동시에 이러한 서비스 모델에 의거 해 비용을 수반하지 않는 규모의 경제에 입각한 모듈화를 필요로 한다. 소프트웨어 개발방식은 한 사람의 고객을 위한 'customization'에서 사용자가 참여할 수 있는 'configuration' 방식으로 바뀌어갈 수 밖에 없다.

5. 결론

본 연구의 특징은 소프트웨어 개발전략을 단순히 개발방식의 관점에서가 아니라 웹 비즈니스의 성공을 좌우하는 플랫폼전략과의 연관관계에서 도출했다는 점이다. 플랫폼전략의 성공은 보다 많은 사람들이 참여할 수 있는 생태계를 만들기 위해 필요한 소프트웨어의 경쟁력에 좌우된다. 웹의 진화에 따라 요구되는 소프트웨어 개발전략의 성공은 가격경쟁력을 담보하는 '규모의 경제'와 상호소통을 가능하게 하는 '서비스화'이라는 모순적인 요구를 해결하느냐에 달려있다. 조직화 방식으로 오픈소스 커뮤니티와

개발방법론으로써 레고형 개발방식은 그러한 모순적 요구를 해결하기 위한 대안들이다. 플랫폼전략에서 도출되는 또 다른 소프트웨어 개발전략 이슈는 진화적인 소프트웨어 개발방식의 의미이다. 곧 앞으로 소프트웨어는 제품의 버전 업그레이드로 출시되지 않고 지속적으로 업데이트로 서비스될 것이고, 이에 따라 요구되는 소프트웨어 개발방식의 변화가 또한 앞으로 연구되어야 할 과제이다.

참조

- [1]문장원(2006), “오픈소스 시장 활성화 를 위한 지원 방안에 관한 연 구”, 한국소프트웨어진흥원.
- [2]백영란(2004), “공개 SW 활성화정책의 생산 적 토론 을 위하여”, *KIPA 리포트*, 2004 년 겨울호
- [3]백영란(2005), “소프트웨어산업의 반 독점 이슈”, 한국소프트웨어진흥원.
- [4]백영란(2007), “웹 2.0 의 유행을 넘어, SW 산업의 키워드를 찾아”, *SW Insight 정책리포트*, 2007 년 5 월호.
- [5] Frederick Chong and Gianpaolo Carraro (2006), “Architecture Strategies for Catching the Long Tail”, Microsoft <http://msdn2.microsoft.com/en-us/library/aa479069.aspx>
- [6] Frank Gens(2006), The IT Market’s \$150B SMB Long Tail”, IDC eXchange, <http://blogs.idc.com/ie/?p=53IDC>
- [7] Ron Goldman(2005), Innovation Happens Elsewhere, Morgan Kaufmann Publisher
- [8]Jack Greenfield(2004), The Case for Software Factories, *Microsoft Architect Journal* . July, 2004. <http://msdn2.microsoft.com/en-us/library/aa480032.aspx>