



# SaaS

special report

## 소프트웨어의 미래 **SaaS**

끊임없이 새로운 IT 용어를 탄생시켰던 웹 환경이 '웹2.0' 못지않은 또 하나의 히트작을 선보였다. 그것은 바로 SaaS. Software as a Service의 약자인 SaaS는 웹과 어우러진 소프트웨어를 의미하는 용어로, 소프트웨어가 진화해 나갈 하나의 지향점을 가리킨다. 그렇다면 SaaS는 지금의 개발자들에게 어떤 의미가 있을까? 무수히 많은 패러다임 속에서 혼란을 겪었던 개발자들의 고충을 덜고자 이번 스페셜 리포트는 먼저 개발자의 관점에서 SaaS의 의미를 살펴보고, 이어서 SaaS가 소프트웨어산업의 미래에서 어떤 역할을 하게 될지 짚어본다. 아울러 SaaS에 그다지 새로울 게 없다는 지적도 적지 않은 만큼, 조금은 차가운 시선으로도 SaaS의 이모저모를 따져보기로 한다.

기획·정리 | 전도영 기자 mir@imaso.co.kr

1부 | 개발자의 눈으로 바라본 SaaS | 장선진

2부 | SaaS로 가는 길 | 손영수

3부 | 소프트웨어산업의 희망인가, 아닌가? | 류한석

플러스 | SaaS, 이렇게 준비한다! | 전도영 기자



# special report 1

## 패러다임 변화에 주목하라! 개발자의 눈으로 바라본 SaaS

항상 해상처럼 나타나는 IT 패러다임들은 모두 시대의 요구를 반영하는 것이고, 이를 구체화해 애플리케이션 개발 과정에 녹여내야 하는 것은 바로 개발자의 몫이다. 따라서 IT 패러다임이 추구하는 바를 정확하게 인식하고 실현 가능한 기술을 바탕으로 구축해 나가는 안목이 개발자에게 요구된다. 이 글에서는 SaaS(Software as a Service)라는 새 IT 패러다임을 개발자의 관점에서 어떻게 기술로 연결해 나갈지를 함께 고민해 본다.

에덤 스미스의 '국부론'이 1776년에 나왔으므로, 결국 경제학은 200년이 넘는 역사를 가진 학문이라고 할 수 있다. 에덤 스미스가 '보이지 않는 손'이란 아주 애매모호한 용어를 사용했을 당시만 하더라도 많은 사람들은 이 용어를 명확하게 이해할 수 없었다. 하지만 현재는 '보이지 않는 손'이 고등학교 교과서에서도 나오며, 경제 주체간의 모든 이해가 자연적으로 조화를 이뤄 수요와 공급이 유지되고 시장가격이 결정된다는 의미로 이해되고 있다. 이는 에덤 스미스의 '보이지 않는 손'이란 새로운 패러다임이 오랜 역사를 거쳐 일반적인 상식으로 발전한 것이라 할 수 있다.

### IT 패러다임의 문제

하지만 전산의 경우 역사가 짧고 변화의 속도가 매우 빠르므로 새로운 패러다임이 출현한 뒤 철저히 검증되거나 발전할 시간 자체가 부족했다. 따라서 현재 새롭게 출현하고 있는 IT 패러다임과 이를 한마디로 요약할 수 있는 각 용어의 명확한 정의와 구체적인 내용은 빈약한 게 사실이다. 이 때문일까? IT 패러다임을 개발자나 기타 IT 종사자들이 한 번에 이해하기란 무척 어려운 일이다. 사실 2004년 웹2.0이란 용어를 팀 오라일리가 처음

말했을 때에도 그 개념을 정확하게 이해하기 어려웠다. 2년여의 시간이 흐른 현재는 많은 사람들에게 웹2.0이란 용어가 참여와 공유를 중시하는 웹이란 의미로 받아들여지고 있으며, 구체적인 실체로서의 블로그나 위키(Wiki), 그리고 AJAX(Asynchronous JavaScript and XML)나 매쉬업(Mashup)을 통해 서비스를 공유시키고 사용자의 참여를 유도할 수 있는 개발 방안 등이 일반화되고 있다.

즉 팀 오라일리가 웹2.0이란 새로운 용어를 만들었을 당시만 해도 구체적이지 않았던 웹2.0 패러다임이 시간이 지날수록 구체화되고 일반화되면서 점점 상식으로 받아들여지고 있는 것이다. 결국 IT 패러다임도 경제학을 비롯한 다른 분야의 패러다임과 마찬가지로 구체화되고 일반화되는 데 시간이 필요하다. 하지만 IT 패러다임은 IT의 변화속도가 워낙 빠른 탓에 구체화되거나 일반화되는 시간 동안 또 다른 패러다임이 출현한다는 문제점을 안고 있다. 이로 인해 많은 IT 개발자들이 패러다임을 인식할 수 있는 시간이 부족하고, 이를 구체적인 기술로 인식하기 전에 또 다른 기술이나 트렌드를 맞이하게 되는 악순환이 이뤄지고 있다. 또한 IT를 주도하는 마이크로소프트나 IBM 등의 대형 벤더들은 각각의 IT 패러다임을 자사에 유리하게 해석해 이를 구체화할 수 있는 방안을 홍보하므로, IT 종사자들의 혼란이 가중되고 있는 상황이다. 그 대표적인 예로 SOA(Service Oriented Architecture)를 꼽을 수 있다. 각 벤더들이 서로 자신에게 유리한 부분을 부각해 SOA를 재해석하고 이를 개발자에게 전달함으

장선진 jangsunjin@gmail.com | Software는 삶을 담는 그릇이라고 생각하며 질 좋은 그릇을 만들기 위하여 Software Architecture 및 Framework 등에 관심이 많다. 많은 사람들 삶에 질을 높이는 서비스를 제공하는 애플리케이션을 개발하고 싶은 꿈 많은 개발자이다.

로써 개발자들이 정확한 SOA의 가치와 의미를 파악하기 어렵게 되었다.

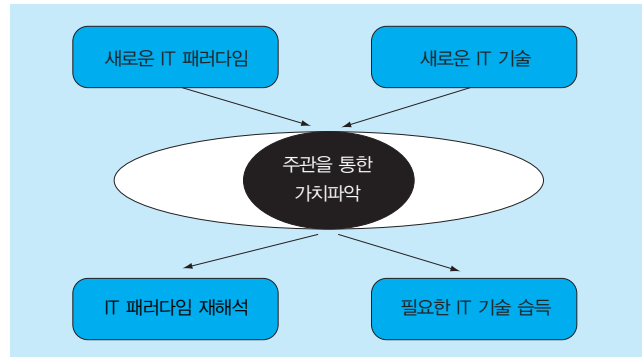
### IT 패러다임을 이해하는 안목

웹2.0이나 SOA, 그리고 SaaS와 같은 새 패러다임에는 근본적으로 사용자들의 요구가 반영되어 있다. 블로그란 이름의 경우 웹2.0이란 용어가 등장한 2004년보다 무려 10년이나 앞선 1994년에 미국의 펜실베이니아 스와스모어 단과대학생인 저스틴 홀에 의해 처음 만들어졌다(물론 블로그의 기원은 보는 시각에 따라 조금씩 다를 수 있지만 웹상에서 참여와 공유를 강조하는 흐름은 훨씬 이전부터 있었던 것이 분명하다). 마찬가지로 AJAX란 용어가 있기 전부터 AJAX와 거의 동일한 기능을 제공하는 웹사이트의 개발이 진행되고 있었으며, 이는 AJAX가 출현하면서 더욱 대중화되었다고 볼 수 있다.

새로운 IT 패러다임이 개발자에게 고통만 던져주는 것은 아니다. AJAX란 용어의 출현으로 많은 웹 개발자들이 사용자에게 편의를 제공하기 위한 UI를 본격적으로 고민하기 시작했으며 프로토타입(Prototype)과 같은 자바스크립트 프레임워크가 등장하고 JavaEE의 DWR(Direct Web Remoting)과 같은 기술이 등장하게 되었다. 이는 패러다임의 변화가 기술 변화를 이끌어낸 것으로 볼 수 있다.

물론 이러한 패러다임의 변화로 인해 개발자들이 알아야 할 것이 늘어났다고 볼 수 있지만 이는 새로운 기회의 등장이라고도 볼 수 있다. 구체적인 사례가 다음의 한메일 Express이다. 다음의 한메일 Express는 AJAX 기반의 웹 메일 서비스를 제공해 주는데 다이내믹 스크롤이나 단축키 지원 등을 통해 사용자의 편의를 높였다. 여기서 흥미로운 점은 한메일 Express의 기획을 개발자가 직접 했다는 것이다. 많은 개발자에게 잘 알려져 있는 페이퍼 프로토타이핑(Paper Prototyping)을 통해 개발자가 직접 기획하고 반복적(iterative) 개발을 통해 성능을 개선하고 있으며 사용자의 적극적인 피드백을 통해 서비스의 질을 높이고 있다. 이는 전통적인 기획자 중심의 서비스 개발 방식에서 개발자 중심으로 서비스 개발의 중심이 옮겨가고 있음을 의미한다. 또 다른 사례는 구글의 지메일(Gmail)인데 개발자가 개발한 웹 메일 서비스를 실제 구글의 주력 서비스로 제공하고 있다. 참고로 지메일은 아직 베타(Beta) 서비스 중인데, 이는 기능이나 기술이 부족해서가 아니라 지메일을 중심으로 부가 서비스를 지속적으로 제공하기 위한 전략으로 보인다.

이처럼 IT 패러다임의 변화는 실제 개발자에게 새로운 기회를 제공하고 있으며 추후 개발자들의 영역이 더욱 넓어지고 권한이나 책임도 늘어날 것이라고 생각된다.



〈그림 1〉 IT 패러다임을 바라보는 안목

〈그림 1〉은 필자가 생각하는 IT 패러다임을 바라보는 안목이다. 필자의 안목은 누구나 가지고 있는 안목이며 누구나 생각할 수 있는 안목이다. 하지만 각 IT 패러다임이나 기술의 가치를 자신의 주관을 바탕으로 파악한다는 것은 말처럼 쉬운 일은 아니다. 웹2.0처럼 새로운 패러다임의 정의 자체가 모호한 것들이 있으며, 구체적으로 사용되는 기술이 애매모호한 경우가 많기 때문이다. 하지만 간략하게라도 각 패러다임에 대해 정리함으로써 자신의 주관을 세우는 것이 가치 있는 패러다임과 기술을 파악하는 기반이 될 것이라 생각한다.

보통 새로운 IT 패러다임을 접할 경우 많은 개발자들이 객관적인 판단을 해야 한다고 생각한다. 물론 객관적인 판단이 수반되어야 하지만, 패러다임을 이해하는 것 자체는 곧 자신의 주관을 바탕으로 패러다임의 가치를 파악하는 것이다. 따라서 남들이 가치 있다는 패러다임이 왜 가치가 있고, 내가 처한 상황에서도 가치 있는 것인지에 대한 주관적인 판단이 필요하다. 특히 IT 대형 벤더들이 객관적인 내용이라고 제시하는 자료나 사례는 비판적인 시각에서 바라볼 필요가 있다.

새로운 IT 기술도 마찬가지다. 요사이에는 오픈소스의 확산으로 정말 좋은 IT 기술들을 쉽게 접할 수 있다. 필자도 이러한 IT 신기술에 열광하며 사용해 보고 장점이 있으면 적극적으로 프로젝트에 도입하려고 노력한다. 하지만 아무리 좋은 기술도 프로젝트에 도입하다 보면 문제가 생기게 마련이다. 이러한 문제가 쉽게 해결될 수 있으면 좋겠지만, 어떤 경우에는 프로젝트 전체에 악영향을 미치는 경우도 존재한다. 따라서 신기술의 장점이 객관적으로 명확하더라도 필자의 경우 프로토타이핑을 통해 검증하려고 한다. 소프트웨어의 프로토타이핑은 매우 중요한 가치 판단의 기준을 제공해 주는데 혼자서 새로운 IT 기술을 이해했을 경우 간과하고 넘어갈 수 있는 문제점들을 프로토타이핑을 통해 파악할 수 있다. 이러한 과정을 통해 소프트웨어의 참 가치를 인식할 수 있는 주관적인 안목이 생기게 되며, 정말 필요한 기술만 습득할 수 있게 된다.



개개인의 안목을 바탕으로 IT 패러다임을 이해한다면 IT 변화를 주도하는 개발자가 될 수 있다. 그럼 이러한 안목을 바탕으로 SaaS에 대한 가치를 파악해 보자.

### 개발자의 눈으로 바라보는 SaaS

인터넷의 발달로 기존의 소프트웨어들이 웹 기반으로 제공되고 있으며, 이러한 추세는 더욱 빠르게 진행되고 있다. IT 산업 전체가 기존의 PC 플랫폼에서 웹 플랫폼으로 자연스럽게 진화하고 있는 것이다. 단적인 예가 이메일인데 PC 플랫폼에서 이메일 기능을 사용하려면 MS 아웃룩과 같은 소프트웨어를 PC 플랫폼에 설치해 사용해야 했지만, 지금은 웹 브라우저를 바탕으로 다음의 한메일 Express와 같은 웹 메일을 써서 MS 아웃룩보다 편리하고 쉽게 메일 기능을 이용할 수 있다.

여기서 한 가지 중요한 차이점은 많은 사람들이 기존의 MS 아웃룩을 소프트웨어로 인식하고 있는 것과 달리, 한메일 Express는 소프트웨어로 인식하지 않는다는 점이다. 필자 역시 소프트웨어 중 패키지 소프트웨어인 MS 오피스나 스타크래프트는 하나의 소프트웨어로 인식하고 있는데 반해, 웹으로 제공되는 소프트웨어는 소프트웨어로 인식하지 않는다. 소프트웨어는 자신의 PC에 설치함으로써 제공받는 것이 당연하다는 인식이 사용자들에게 존재하기 때문이다. 하지만 웹을 기반으로 서비스를 제공하는 소프트웨어는 설치 과정이 존재하지 않으므로 많은 사용자들이 소프트웨어로 인식하지 않고 서비스로 인식하고 있다. 즉 SaaS라는 이름의 의미 그대로 소프트웨어를 서비스로 인식하기 시작했다는 것이다.

이러한 변화에 대한 가능성은 이미 오래 전부터 시작되었다. 2001년 9월 21일자 전자신문은 '포스트SW-SW 사지 말고 웹서비스 이용하세요' 라는 기사를 통해 지금의 SaaS와 동일한 개념을 포스트 소프트웨어 모델로 제시하고 있다. 다음은 그 주요 내용을 발췌한 것이다.

'웹서비스가 소프트웨어 산업의 지형을 바꾸는 포스트 모델로 떠오르고 있다. 웹서비스는 단말기 종류에 구애받지 않고 언제 어디서나 웹에 접속해 최적화, 개인화된 서비스를 구현한다는 개념으로 기존 SW산업 질서를 재편하는 혁신적인 내용을 담고 있다. 웹서비스가 일반화되면 각 단말기별로 필요한 SW를 구매, 설치할 필요 없이 언제 어디서나 어떤 장비에서도 일관된 내용의 웹서비스를 받을 수 있게 된다. 따라서 '구매-구매' 라는 단선적인 사이클을 가진 상품으로서의 SW는 사라지고 '상시 제공-수시 이용'의 순환 체인을 갖는 서비스로서의 SW 모델만이 남게 돼 기존 SW산업 시스템에 일대 변화가 일어나는 것이다.'

'웹서비스 모델은 기업의 인프라스트럭처를 표준기술에 기반한 플랫폼으로 바꾸는 작업에서부터 SW 구매 및 판매방식에 대한 새로운 정책을 수립하는 문제에 이르기까지 광범위한 변화를 요구하고 있다. 이에 따라 기존 시장의 질서나 업체간 판도가 웹서비스에서 재편될 가능성을 내포하고 있다. 우선 SW유통 모델이 바뀌는 만큼 SW를 판매하고 설치하는 기존 오프라인 공급업체들의 입지가 줄어들다. 따라서 마케팅이나 프로모션 방식에도 큰 변화가 일어날 것이다. 뿐만 아니라 어떤 개발언어, 어떤 틀을 쓰느냐의 문제는 점점 중요하지 않게 되는 반면에 플랫폼이나 아키텍처, 프레임워크 기술의 중요성은 더욱 부각될 것이다.'

놀랍게도 현재의 SaaS 정의와 동일한 내용이 단지 포스트 소프트웨어 모델이란 이름으로 2001년도에 언급된 것이다. 이를 곰곰이 생각해 보면 현재의 SaaS 관련 논의는 이미 오래전부터 존재한 것으로, SaaS와 같은 소프트웨어 유통방식이 일반 사용자에게도 충분히 받아들여질 만한 여건이 마련되자, SaaS가 큰 이슈로 떠오른 것으로 이해할 수 있다.

### 서비스에 대한 이해

SaaS와 SOA의 공통적인 핵심은 서비스로, SaaS와 SOA는 모두 지향점이 서비스임을 알 수 있다. 그렇다면 서비스란 대체 무엇이며 왜 IT 패러다임의 중심으로 부각되고 있을까?

원래 서비스라는 개념은 생산자보다는 소비자를 위한 것이다. 서비스는 생산자가 소비자에게 제공하는 행위나 물건을 총칭하는 용어이다. 예를 들어 우리는 식당에서 종업원들에게 식사에 관련된 서비스들을 제공받을 수 있다. 특색 있고 맛있는 메뉴를 제공하는 식당의 경우 언제나 손님들이 많다. 즉 식당이라는 생산자가 제공하는 서비스가 고객의 욕구를 충분히 만족시키기 때문이며, 이를 통해 식당은 더 많은 이윤을 추구할 수 있다.

웬지 경제학에서 나올 것 같은 서비스의 의미인데, 이는 IT 세계에서도 동일하게 적용된다. 누구나 자주 방문하는 웹사이트가 하나쯤 있게 마련인데 해당 웹사이트에서 제공하는 서비스가 사용자의 욕구를 충분히 만족시키기 때문이다. 하지만 항상 최고의 서비스를 제공하는 웹사이트도 시간이 지남에 따라 사용자의 새로운 욕구에 맞는 서비스를 제공하지 않는다면 결국 웹사이트 방문자들이 줄어들고 결국 수익이 감소하게 될 것이다. 이에 따라 지속적으로 사용자들에게 만족도 높은 서비스를 제공할 수 있는 노력이 필요한데, 그 방안 가운데 하나가 SaaS인 셈이다.

웹기반의 소프트웨어를 통해 소비자에게 만족도 높은 서비스를 제공하기 위해서는 고려해야 할 점들이 많다. 변화하는 고객의 요구에 맞춰 지속적으로 서비스를 개선해야 하고, 지속적으로

중단 없이 서비스를 제공해야 한다. 또한 웹의 특성상 누구에게나 열려 있는 환경 속에서 고객의 데이터 및 개인정보 등을 보호해야 하는 등 다양한 요구사항들이 뒤따른다. 어쩌면 이러한 요건들을 구체적으로 정리한 것이 바로 SaaS가 아닐까 생각해 본다.

### SaaS 기반의 애플리케이션 구축

생소한 SaaS 개념의 애플리케이션을 구축하기에 앞서 현재 새로 개발하려는 애플리케이션이 SaaS라는 패러다임을 바탕으로 구축될 필요가 있는지 확인해 볼 필요가 있다. 단지 몇 명의 고객을 위한 애플리케이션을 개발하면서 새로운 패러다임이냐 이유로 SaaS 기반의 애플리케이션을 구축하기보다는 기존의 ASP(Application Service Provider) 방식의 애플리케이션 형태로 개발의 용이성을 추구하는 것도 나쁘지 않다. 그러나 수많은 고객에게 유연하고 확장성이 뛰어난 애플리케이션을 제공하려면 SaaS 기반의 애플리케이션이 좋은 대안이 될 것이다.

각 패러다임을 결정했다면 애플리케이션 개발에 필요한 아키텍처를 디자인하는 것이 중요하다. 알다시피 우리가 만들어내는 소프트웨어는 절대 부드럽지 않다. 각 코드가 파일로 만들어지는 순간부터 소프트웨어는 딱딱하게 굳기 시작해 코드가 늘어날수록 서로 붙어 떨어지지 않는다. 이러한 문제를 해결하기 위한 방안으로 Spring Framework를 이용한 AOP(Aspect-Oriented Programming) 기반의 애플리케이션 개발이 각광받고 있다. 하지만 이 방법은 개발 단계에서의 유연성은 지원하지만, 비즈니스 단의 유연성을 부여하지는 못한다. 비즈니스에서의 유연성은 얼마나 유연한 방식으로 메타데이터를 관리하는가에 달려 있다고 생각한다.

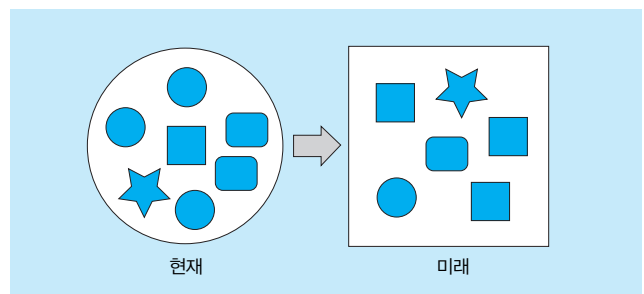
특히 웹2.0의 출현으로 모든 웹서비스에는 개방성과 참여 및 공유가 중요한 이슈가 되었다. SaaS 기반의 애플리케이션에서 제공하는 서비스 역시 사용자의 참여 및 공유가 매우 중요한 요소이며, 특히 사용자의 필요에 따라 제공하는 서비스의 UI나 브랜딩 변경, 그리고 워크플로우 및 룰의 변경 등을 지원해야 한다. SaaS 기반의 애플리케이션은 기업이나 단체를 대상으로 서비스를 제공하는 경우가 많다. 각기 다양한 서비스를 수많은 기업 및 단체에 제공하기 위해 서비스의 특성에 맞는 메타데이터를 관리하고 있어야 하고, 필요에 따라 메타데이터의 내역을 변경함으로써 애플리케이션의 영향을 최소화하면서 사용자에게는 유연성과 확장성을 보장해 줄 수 있다.

메타데이터를 관리하는 방법은 무수히 많지만 결국 구축하고자 하는 애플리케이션의 서비스 범위에 따라 달라진다고 생각한다. 사용자의 확장성이 중요하지 않은 애플리케이션의 메타데이터 관리 모듈이 복잡할 필요는 없지만, 세일즈포스닷컴(Salesfo

re.com)의 Apex처럼 사용자가 직접 세일즈포스닷컴 내에서 동작하는 애드온(Add-on) 애플리케이션까지 제작할 수 있도록 지원한다면 메타데이터는 무척이나 중요한 위치를 차지할 것이다.

### 패러다임의 변화

〈그림 2〉와 같이 현재는 동그라미라는 패러다임이 세상을 지배한다고 가정하자. 동그라미 패러다임이 지배하는 세상에서는 동그라미들이 가장 세상에 잘 적응한 사람이다. 하지만 미래에는 네모의 세상이 올 것이라고 외치는 한 명의 네모가 동그라미 세상에 있었다. 동그라미 세상에서는 인정받지 못하지만 다가올 네모의 세상에 대한 준비를 마친 사람이다. 이러한 네모의 외침에 귀를 기울여 조금씩 네모처럼 변해가는 동그런 네모가 점점 늘어나기 시작한다. 이들은 패러다임의 변화에 대해 적절한 시기에 준비를 시작한 사람이라고 볼 수 있다. 마지막으로 별 모양은 먼 미래의 패러다임 변화를 준비한다. 이런 동그라미 세상에서 변화에 대한 확실한 주관을 가지고 있는 것은 네모와 별이다. 다만 별은 조금 엉뚱해 미래에도 크게 인정받지 못할 수 있지만 나름대로 세상을 주도하고 있다고 생각할 것이다. 네모 세상에 존재하는 동그라미도 나름대로 주관을 가지고 있다고 생각한다. 다만 패러다임의 변화를 받아들이지 못해 변화하지 못했다면 문제가 있다.



〈그림 2〉 패러다임의 변화

국내에서 SaaS의 모범사례로 꼽는 세일즈포스닷컴과 같은 애플리케이션이 개발된다면 미래를 준비하는 네모나 별과 같은 사람이 많아야 한다고 생각한다. 별과 네모가 미래를 향해 자기 주관을 가지고 노력할 때 창의적인 패러다임을 바탕으로 가치 있는 소프트웨어가 개발될 것이다. 보이지 않는 손으로 컴퓨터 앞에서 키보드를 연주하고 있을 네모와 별에게 필자는 박수를 보낸다. +