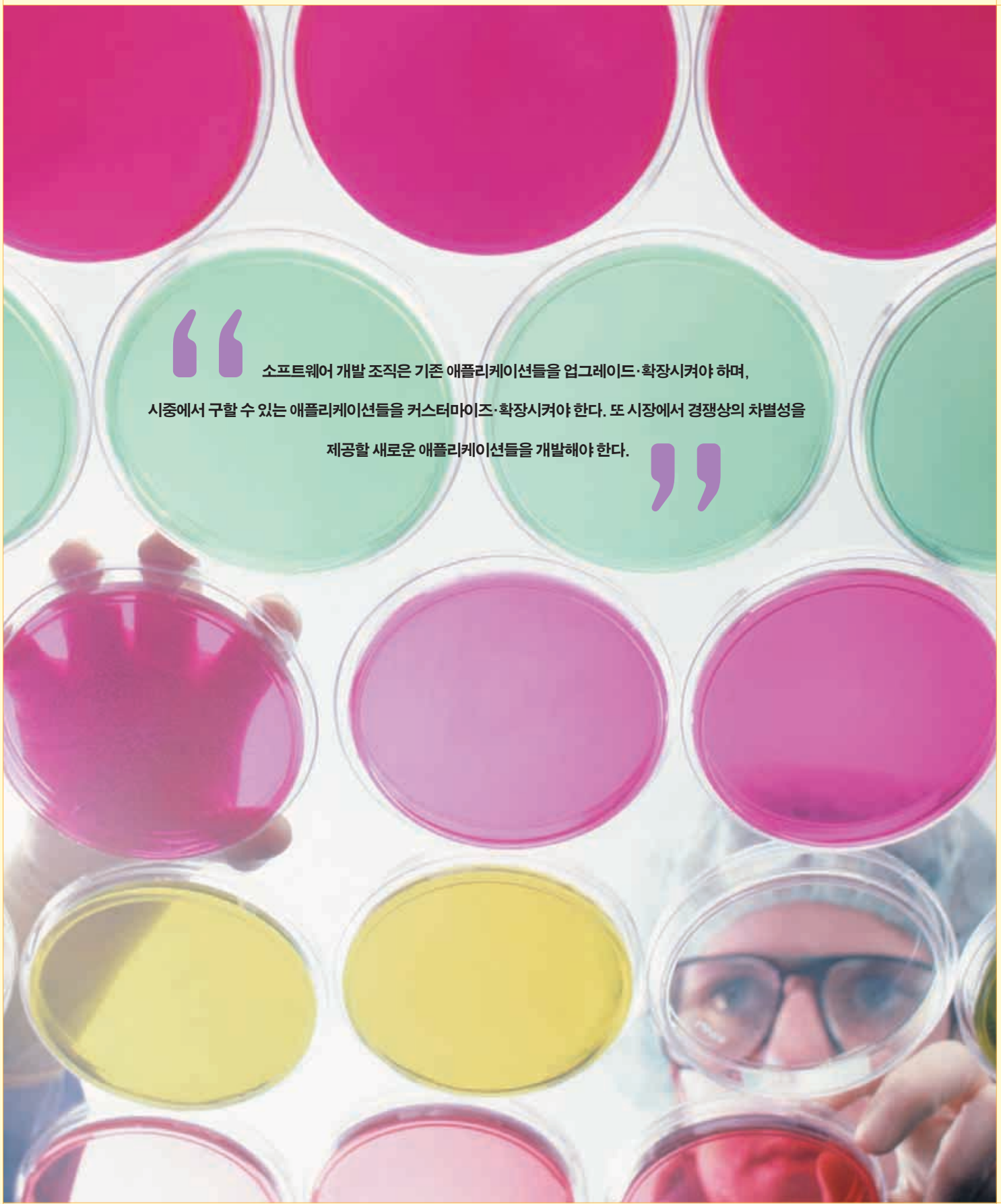

“소프트웨어 개발 조직은 기존 애플리케이션들을 업그레이드·확장시켜야 하며,
시중에서 구할 수 있는 애플리케이션들을 커스터마이즈·확장시켜야 한다. 또 시장에서 경쟁상의 차별성을
제공할 새로운 애플리케이션들을 개발해야 한다.”

온 디맨드 비즈니스의 구현

소프트웨어 개발을 위한 3가지 원칙

온 디맨드 시대에 기업 성공을 결정짓는 주요인 중 하나는 소프트웨어 개발 능력이다. e-business on demand를 규정하는 새로운 비즈니스 요구와 고객 기대 수준을 충족시키기 위해서는 소프트웨어 개발 팀도 높은 수준의 대응 능력과 민첩성을 갖추어야 한다. 이 개발 팀이 개발·유지·관리하는 소프트웨어 애플리케이션은 기업이 지속적으로 경쟁력을 확보, 유지할 수 있도록 그 어느 때보다 혁신적이어야 한다.

온 디맨드 비즈니스와 기업의 경영 환경을 지원하기 위해 비즈니스 소프트웨어 애플리케이션은 반드시 다음과 같은 특징을 구비해야 한다.

- 급변하는 비즈니스 요구에 신속하게 대응
- 전략적 우위를 확보하고 관리
- 점점 확산되고 있는 온 디맨드 비즈니스를 지원 가능하도록 확장 가능하고 안정적

이를 위해서 소프트웨어 개발 조직은 기존 애플리케이션들을 업그레이드·확장시켜야 하며, 시중에서 구할 수 있는 애플리케이션들을 커스터마이즈·확장시켜야 한다. 또 시장에서 경쟁상의 차별성을 제공할 새로운 애플리케이션들을 개발해야 한다. 그리고 이 조직은 커스터마이제이션, 확장, 신규 개발 등을 통해 소프트웨어에서 최대의 가치를 끌어내기 위해서 소프트웨어 개발 능력을 더욱 개선해 온 디맨드 시대에 적합한 기술력을 갖추어야 할 것이다.

특히 소프트웨어 개발 조직은 다음 3가지 소프트웨어 개발 원칙을 견지해야 한다.

- **반복하면서 개발한다** ·· 개발 팀은 결과 위주(result-oriented)의 프로세스를 이용해야 한다. 결과 위주의 프로세스란 소프트웨어 시스템을 배포하기 전까지 소프트웨어 시스템을 반복하면서 개선시킬 수 있는 프로세스를 말한다. 이것은 프로젝트 리스크를 줄여 주며, 예측성을 높여 주고, 적절한 범위를 정해 주며, 설계상의 결함을 줄여 준다.
- **아키텍처에 중점을 둔다** ·· 모든 소프트웨어 디자인은 재활용 가능한 컴포넌트와 서비스 지향 모델에 기반을 두어야 한다. 이렇게 함으로써 전체 기능에 영향을 미치지 않고 관리, 업그레이드, 교체할 수 있다. 아키텍처에 중점을 둔 소프트웨어 디자인은 변화에 맞춰 애플리케이션을 새로 디자인할 수 있으며, 복잡성은 줄이는 대신 품질과 완성도는 더욱 높여 준다.
- **변경사항과 자산을 관리한다** ·· 개발 팀은 개발 중인 소프트웨어의 모든 변경 사항을 추적, 파악하고 있어야 하며, 팀의 활동을 관리하고, 비즈니스의 요구에 따른 전략적 개발 자산들을 보호해야 한다. 이렇게 하면 여러 팀이 동시에 개발 작업에 참여할 수 있으므로 개발 주기를 줄일 수 있으며, 개발 과정의 주요 자산을 보호할 수 있고, 배포된 소프트웨어의 신뢰성을 높일 수 있다.

이전에 별개 업체였다가 지금은 IBM 소프트웨어 브랜드의 하나인 IBM Rational은 위에서 언급한 3가지 원칙에 입각한 포괄적인 소프트웨어 개발 솔루션을 제공한다. IBM Rational 플랫폼은 소프트웨어 엔지니어링의 가장 우수한 실례와 선구적인 툴들 그리고 전문가에 의한 전문 서비스를 결합한 것으로, 온 디맨드 비즈니스에 부응하는 소프트웨어 개발 능력을 신속하게 또 지속적으로 확보할 수 있도록 해준다.

또 IBM Rational은 20년 이상 개방된 통합 소프트웨어 시스템을 제공하면서 상당한 경험과 역량을 축적하고 있다. 개방성과 통합은 온 디맨드 운영 환경의 핵심 요소라고 할 수 있다.

- **통합** ·· IBM Rational은 'Service-Oriented Architecture(SOA)' 와 엔터프라이즈와 소프트웨어 아키텍처, 이기종 플랫폼 지원 등에서의 탁월한 전문성과 선도성을 갖추고 있다.
- **개방성** ·· IBM Rational은 오랫동안 오픈 컴퓨팅의 목표들을 설정, 지원해 왔다. 대표적으로 현재 애플리케이션과 데이터베이스 디자인 및 비즈니스 프로세스의 모델링 표준으로 인정 받고 있는 Unified Modeling Language(UML)의 개발을 들 수 있다. IBM Rational은 이외에도 다양한 오픈 컴퓨팅 표준들의 개발에 적극 참여해 왔다. 또 주요 프로그래밍 언어와 운영 플랫폼에서 이들 개방 표준을 지원할 것을 독려하고 있으며, 써드파티의 툴 통합을 위해 포괄적인 API 세트를 제공하고 있다.

현재 전세계 수천 기업들이 IBM Rational이 주창하는 개방된 통합 방식의 혜택을 인정하고 있다. 이들 기업의 프로세스는 결과 위주이며, 개발 결과들은 잘 설계되어 있으며 재활용 가능하고, 이들 기업은 온 디맨드 시대에 필요한 능력을 바탕으로 개발 작업을 진행하고 있다.

e-business on demand는 모든 고객 요구, 시장 기회, 외부 위협에 신속하고 탄력적으로 대처할 수 있는 능력을 제공할 것이다.

혁신은 지속된다

1990년대 중반 거의 모든 기업들이 인터넷 전략을 채택하느라 열광했던 때를 기억하는가? 그 가능성은 무궁무진해 보였다. 하룻밤 사이에 수백, 수천 개의 웹사이트들이 생겨났고, 거의 모든 상품, 서비스, 정보들이 웹을 통해 쏟아져 나왔다. 은행들은 처음에 개인과 기업이 계좌 정보에 액세스할 수 있도록 했으며, 곧 고객이 직접 계좌 이체를 할 수 있도록 했다. 항공사도 간단한 운항 안내 정보만 제공하다가 곧 온라인 예약 서비스까지 제공했다. 업종을 망라하고 모든 기업들이 안전하고 인터랙션 가능한 인터넷 매장들을 개설, 확장하느라 정신이 없었다.

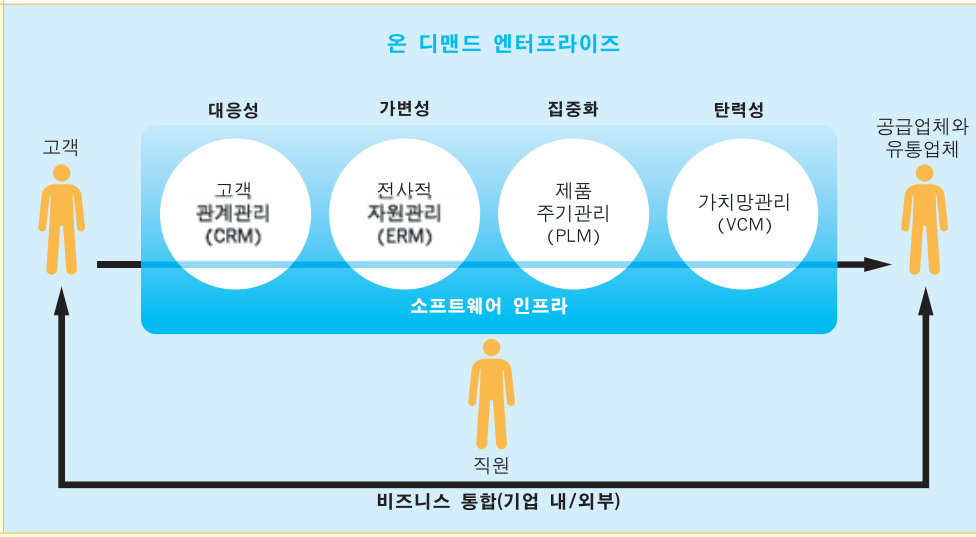
WWW의 출현 초기부터 이를 통해 비즈니스 가치를 높이려는 기업들은 WWW의 발전에 맞춰 많은 작업들을 해야 했다. 급변하는 기술들을 익히고 응용해야 했으며, 차별성을 위해 계속 개혁, 혁신해야 했다. 주로 간단한 인터넷 액세스가 제공되던 초기 단계에서는 세계적으로 휘몰아치던 웹의 거센 물결에 신속하게 대응하기 위해 웹에 정통한 사람들을 고용했다. 그 다음에는 웹에서의 트랜잭션을 지원하기 위해 내부 시스템들을 통합하는 과정에서 문제가 좀 더 복잡해졌다. 가상 광고 게시판의 홍수 속에서 고객들을 끌어들여 실제 매출을 올리기 위해서는 관련된 주요 비즈니스 프로세스들을 자동화해야 했다.

그러면 향후에는 어떤 변화가 일어날 것인가? 새로운 비즈니스 자동화의 시대가 시작되었으며, 점점 더 많은 기업들이 이 흐름에 합류하고 있다. 'e-business'를 선도하는 IBM은 이 새로운 시대에 대해서도 명확한 비전을 제시하고 있다. 바로 'e-business on demand'이다.

기업들이 처음에는 간단한 인터넷 액세스만 제공하다가 이후에 웹으로 비즈니스 프로세스를 통합해 나아갔던 것처럼, 이제 기업들은 세 번째 단계로 나아가야 한다. 전사적으로 또 기업 외부까지 완벽하게 통합된 비즈니스 프로세스에 기반한 기업을 창출함으로써 주요 협력사, 공급업체, 고객들과 연계해야 하는 것이다. 그 결과는? e-business on demand는 모든 고객 요구, 시장 기회, 외부 위협에 신속하고 탄력적으로 대처할 수 있는 능력을 제공할 것이다.

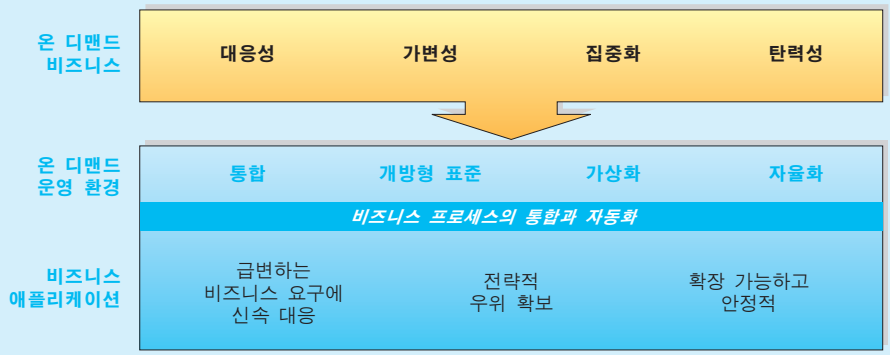
세 번째 단계로 나아가기 위해 기업들은 소프트웨어를 최대한 활용해야 할 것이다. 기존 애플리케이션을 확장해야 하고, 패키지 애플리케이션을 커스터마이즈해 배포하고, 그 어느 때보다 빠르게 새 애플리케이션들을 작성, 배포해야 한다. 이런 이유로 이제 기업의 소프트웨어 개발 능력은 그 기업이 온 디맨드 컴퓨팅으로 성공적으로 도약할 수 있는지를 결정하는 관건이 되고 있다. 기업의 소프트웨어 개발 툴과 기술력은 e-business on demand를 규정짓는 새로운 비즈니스 요구와 고객 기대치를 충족시켜야 한다. 그리고 기업에서 사용하는 애플리케이션들은 기업의 지속적인 경쟁력 우위를 위해 계속 혁신되어야 할 것이다.

이 글에서는 온 디맨드 비즈니스를 지향하는 기업에서 소프트웨어 개발 능력을 향상시킬 방안을 소개할 것이다. 첫째, e-business on demand의 주요 특성과 비즈니스 프로세스를 통합, 자동화하는 애플리케이션 간의 관계에 대해 살펴보고 둘째, 온 디맨드 컴퓨팅의 주요 운영 환경을 지원하기 위한 소프트웨어의 필수요건에 대해 알아보며 셋째, 전략적 우위를 구축하는 데 소프트웨어 개발이 얼마나 중요한지에 대해 알아볼 것이다. 또 새로운 시대의 새로운 경쟁 환경에서 기업이 발전, 번영하는 데 적합한 소프트웨어 아키텍처의 특징들을 살펴볼 것이다. 마지막으로 온 디맨드 시대를 위한 소프트웨어 개발 플랫폼인 IBM Rational 솔루션을 사용함으로써 누릴 수 있는 고유한 장점들도 소개할 것이다.



[그림 1]

온 디맨드 비즈니스에서 일반적으로 공통된 애플리케이션들은 기업 내에서 완벽하게 통합되어야 하며, 공급업체와 유통업체, 또 주요 고객들과도 연계되어야 한다.



[그림 2]

온 디맨드 비즈니스의 과제를 해결할 수 있는 주 요소인 애플리케이션들은 비즈니스 프로세스를 통합하고 자동화한다.

온 디맨드 e-business의 구축

온 디맨드 e-business란 무엇인가? IBM은 기업의 경영진이 자사를 통합된 총체로서 파악·관리할 수 있는 기업이라고 정의하고 있다. 즉 이전에 고립되어 있던 각각의 부서들이 상호 연관을 맺으면서 전사적으로 또 고객까지 포괄해 통합된 비즈니스 프로세스를 구축하는 기업이 온 디맨드 e-business이다.

온 디맨드 비즈니스는 다음과 같은 4가지 특징을 가지고 있다(e-business on demand에 대한 더욱 자세한 내용은 www-3.ibm.com/e-business/index_fl.html 참조)

- **대응성** ·· 수요, 공급, 가격, 노동인력 및 경쟁 측면에서의 예측 불가능한 다양한 변화에 역동적으로 대응
- **가변성** ·· 생산성, 자본, 재무 등과 관련해 변화무쌍한 프로세스나 비용 구조에 민첩하게 적응
- **집중화** ·· 밀접하게 통합된 전략적 협력사들과 함께 차별화된 업무, 자산, 핵심 분야에 주력
- **탄력성** ·· 일관된 가용성과 보안 능력을 통해 변화와 위협 요인들 관리 가능

기업들이 이런 특성들을 확보하려면, 자사 비즈니스 프로세스들을 통합시킨 운영 환경을 구축해야 한다.

온 디맨드 운영 환경의 구축

이러한 광범위한 통합과 연결은 IBM이 '온 디맨드 전산 운영 환경(on demand Operating Environment)'이라고 부르는 것을 통해 달성할 수 있는데, 이것은 기업이 자사 고객에게 더 빨리 더 많은 가치를 제공할 수 있도록 해준다. 이 온 디맨드 운영 환경은 다음과 같은 4가지 특징을 가지고 있다.

- **통합** ·· 온 디맨드 환경은 분산되어 있는 컴퓨팅 자산들을 단순히 연결하는 데 그치지 않고 핵심 프로세스와 시스템을 통합해, 회사 내부뿐만 아니라 다른 여러 기업에서도 이들 프로세스와 시스템을 이용할 수 있도록 한다.
- **개방형 표준** ·· 통합의 전제로 공개 기술 인터페이스와 합의된 표준을 필수적으로 사용한다.
- **가상화** ·· 컴퓨팅 인프라의 통합을 통한 그리드 컴퓨팅을 구현함으로써 분산되어 있는 컴퓨팅 자원들을 하나의 거대한 가상 컴퓨터처럼 공유, 관리할 수 있다.
- **자율성** ·· 인간의 자율 신경체계가 신체의 주요 기능들을 관리하는 것처럼 오늘날의 복잡한 컴퓨팅 시스템도 스스로를 관리할 수 있는 기술을 이용해야 한다.

이 온 디맨드 운영 환경을 구축함으로써 누릴 수 있는 혜택은 방대하다. 비즈니스 애플리케이션들을 개방 표준에 근거해 통합할 수 있으므로 시스템의 상호운용성이 촉진될 것이다. 또 사내 시스템들이 사내의 모든 프로세스들을 지원하게 만듦으로써 컴퓨팅 자원의 가상 그리드를 이용할 수 있게 되어 더 많은 고객에게 더 빨리 다가갈 수 있다. 마지막으로 이런 시스템들의 복잡한 문제들을 자율적으로 관리하게 되어 지금까지 대규모 비즈니스 컴퓨팅에 수반되었던 여러 수작업들을 대폭 줄일 수 있다.

소프트웨어 애플리케이션과 온 디맨드 운영 환경

오늘날 기업에서 가장 많이 사용하고 있는 비즈니스 애플리케이션은 고객관계관리(CRM), 전사적자원관리(ERM), 제품주기관리(PLM), 가치망관리(VCM) 등 다양한 관리 기능을 제공하는 애플리케이션들이다. 이 애플리케이션들은 온 디맨드 비즈니스에서 매우 중요한 자원들을 보유하고 있다. 온 디맨드 시대에 이 애플리케이션들은 전사적으로 완벽하게 통합되어야 할 뿐만 아니라 기업 외부의 주요 공급업체와 유통회사와도 통합되어 고객에게 더 많은 비즈니스 가치를 제공해야 한다 **(그림 1)**.

이 글에서는 온 디맨드 운영 환경의 한 부분, 비즈니스를 통합, 자동화하는 소프트웨어 애플리케이션들을 중점으로 살펴볼 것이다. 온 디맨드 운영 환경이 소프트웨어 애플리케이션으로만 구현되는 것은 아니지만 궁극적으로 비즈니스를 통합하고 자동화하는 부분은 애플리케이션 부분이다. 따라서 기업의 소프트웨어 애플리케이션은 온 디맨드 비즈니스를 구현하는 핵심 요소이다.

그렇다면 온 디맨드 시대에 소프트웨어 애플리케이션에 요구되는 것은 무엇일까? 비즈니스 소프트웨어 애플리케이션은 다음 3가지 요건을 해결해야 한다.

- 급변하는 비즈니스 요구에 신속하게 대응해야 한다. 온 디맨드 세계에서 비즈니스 요구사항은 전통적인 비즈니스에서보다 훨씬 빨리 변한다. 비즈니스 환경이 변하면, 즉 새로운 기회가 등장하거나 새로운 위협이 나타나면, 기업은 그 어느 때보다 신속하게 비즈니스 애플리케이션을 수정해 새로운 기회를 활용하거나 새로운 위협에 대처해야 한다.
- 전략적 우위를 확보, 유지해야 한다. 기업은 마땅히 시장에서 자사를 차별화시킬 수 있는 고유의 장점들을 활용해야 한다. 그리고 이러한 차별성을 최대한 이용하기 위해서는 소프트웨어 파워를 적극적으로 이용해야 한다.
- 확장 가능하고 안정적이어야 한다. 애플리케이션은 잘 실행되고 적절히 이용될 때만 그 가치를 발휘한다. 주요 애플리케이션들은 지속적으로 또 결함 없이 수행되어야 하며, 특히 기업이 급성장할 때 확장성과 안정성이 밑받침되어야 한다.

"새로운 기회가 등장하거나 새로운 위협이 나타나면, 기업은 그 어느 때보다 신속하게 비즈니스 애플리케이션을 수정해 새로운 기회를 활용하거나 새로운 위협에 대처해야 한다."

온 디맨드 시대에 성공하는 기업은 이 모든 애플리케이션들을 통합함으로써 비즈니스 프로세스들을 통합하고 자동화할 것이다.

소프트웨어 개발의 필요성 : 소프트웨어 애플리케이션에서 비즈니스 가치 끌어내기

온 디맨드 시대의 기업은 모든 종류의 애플리케이션을 활용해 경쟁 우위를 창출해야 한다. 기존 시스템의 가치를 최대로 끌어올려야 하며, 시중에서 구할 수 있는 패키지 애플리케이션을 커스터마이즈해 배포하고, 새로운 커스텀 소프트웨어와 애플리케이션을 개발해야 한다.

- **레거시 애플리케이션** ·· 여전히 고유의 비즈니스 역할을 담당하면서 그 기업 고유의 중요한 로직을 구현하고 있는 기존 시스템을 무조건 폐기, 교체할 수는 없다. IT 부서는 현재 보유하고 있는 시스템들을 확장 또는 업그레이드하거나 그 위에서 다른 애플리케이션들 – 더욱 강력해진 데이터베이스 관리 기술, 미들웨어 및 여러 클라이언트 애플리케이션들 – 을 실행시킴으로써 이들 시스템들을 최대한 이용해야 한다.
- **패키지 애플리케이션** ·· 상용으로 구할 수 있는 소프트웨어를 구입할 생각이라면, 이 애플리케이션이 기업의 중요한 전략적 요구사항을 해결할 수 있는지 반드시 확인해야 한다. 패키지 애플리케이션을 구입하면 시간과 자원을 절약할 수 있지만, 이 애플리케이션을 그대로 적용해 주요 비즈니스 기능을 담당할 수 있는 경우는 극히 드물다. 대개는 이 애플리케이션들을 커스터마이즈하거나 확장시켜야 기업의 핵심 역할을 지원할 수 있게 될 것이다.
- **신규 개발** ·· 시장 경쟁에서 진정한 차별성을 추구하는 기업이라면, 특히 온 디맨드 e-business를 추구하는 기업이라면, 자사 비즈니스에 꼭 들어맞는 새로운 소프트웨어 시스템을 개발해야 할 것이다. 기업은 이런 소프트웨어를 직접 설계, 개발, 테스트, 전개해서 주요 비즈니스를 수행하도록 해야 한다.

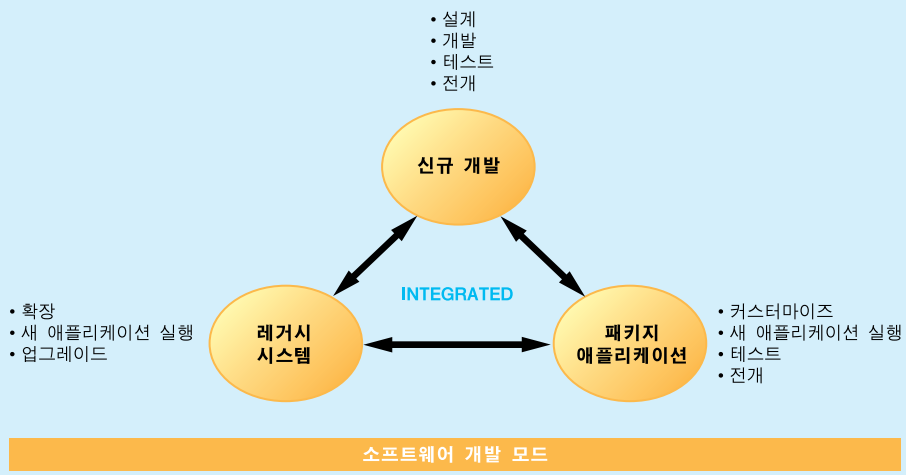
이 모든 활동들, 즉 기존 애플리케이션의 확장, 패키지 제품의 수정, 새 애플리케이션의 개발 등이 모두 소프트웨어 개발의 범주에 든다. 온 디맨드 시대에 성공하는 기업은 이 모든 애플리케이션들을 통합함으로써 비즈니스 프로세스들을 통합하고 자동화할 것이다. 커스터마이제이션과 확장, 신규 개발을 통해 애플리케이션의 가치를 극대화하고, 이 애플리케이션들을 통합할 수 있다면 그 기업의 소프트웨어 개발 능력이 크게 향상될 것이다.

달리 말해 온 디맨드 시대에 기업이 성공하기 위해서는 소프트웨어 개발에 상당한 비중을 두고 추진해야 한다. 통합과 커스터마이제이션, 신규 개발에 필요한 기술을 갖추는 것이 매우 중요하다는 말이다.

소프트웨어 개발 역량을 높이기 위한 3가지 원칙

e-business on demand의 요구 조건을 충족시킬 수 있도록 소프트웨어 개발 능력을 강화하기 위해서 소프트웨어 개발 팀은 반드시 다음의 3가지 원칙을 해결해야 한다.

- **반복하면서 개발한다** ·· 소프트웨어 시스템을 배포하기 전까지 소프트웨어 시스템을 반복하면서 개선시킬 수 있는 결과 위주(result-oriented)의 프로세스를 이용해야 한다.
- **아키텍처에 중점을 둔다** ·· 전체 기능에 영향을 미치지 않고 관리, 업그레이드, 교체할 수 있는 재활용 가능한 컴포넌트와 서비스 지향 모델에 기반을 두고 모든 소프트웨어를 설계한다.
- **변화와 자산을 관리한다** ·· 개발 중인 소프트웨어의



[그림 3]

모든 애플리케이션들의 안정성과 적응력을 높이고 전략적 가치를 발휘하도록 하기 위해서는 이 애플리케이션들을 커스터마이즈, 확장, 신규 개발해야 한다. 이를 위해서는 애플리케이션마다 각각 다른 형태의 개발 작업을 수행해야 한다.

한 번 반복할 때마다 시스템 아키텍처를 검증하고, 애플리케이션 요구 사항들을 해결할 수 있는지를 확인하고 소프트웨어의 품질을 높일 수 있다.

모든 변경 사항을 추적, 파악하며, 팀의 활동을 관리하고, 비즈니스의 요구에 따른 전략적 개발 자산들을 보호해야 한다.

(그림 4)는 세가지 소프트웨어 애플리케이션 개발 원칙과 이 애플리케이션들이 지원하는 운영 환경 간의 관계를 보여 주는 것이다.

온 디맨드 소프트웨어 애플리케이션의 개발 과정

(그림 4)에서 보다시피 e-business on demand는 1)비즈니스 토대와 2)이를 지원하는 운영 환경 및 비즈니스 애플리케이션 그리고 3)소프트웨어 개발 원칙 – 비즈니스 애플리케이션을 관통하는 – 의 논리적 연관성을 바탕으로 하고 있다. 이 연관성을 좀 더 간단하게 표현하면, 다음과 같다.

비즈니스 개선을 위해 소프트웨어 개발 능력을 향상시킴으로써 소프트웨어를 최대한 활용한다.

그렇다면 소프트웨어 개발 능력을 개선하기 위해 앞에서 언급한 세가지 소프트웨어 개발 원칙을 따르려 한다면 어떤 작업들을 해야 할까? 소프트웨어를 개발할 때 개발팀과 개발 프로세스에 특별히 요구되는 것은 무엇일까? 이제 세가지 소프트웨어 개발 원칙에 대해 자세히 살펴보면서, 이 세가지 원칙이 온 디맨드 비즈니스를 지원하는 소프트웨어 개발에 어떤 장점을 제공하는지 알아보자.

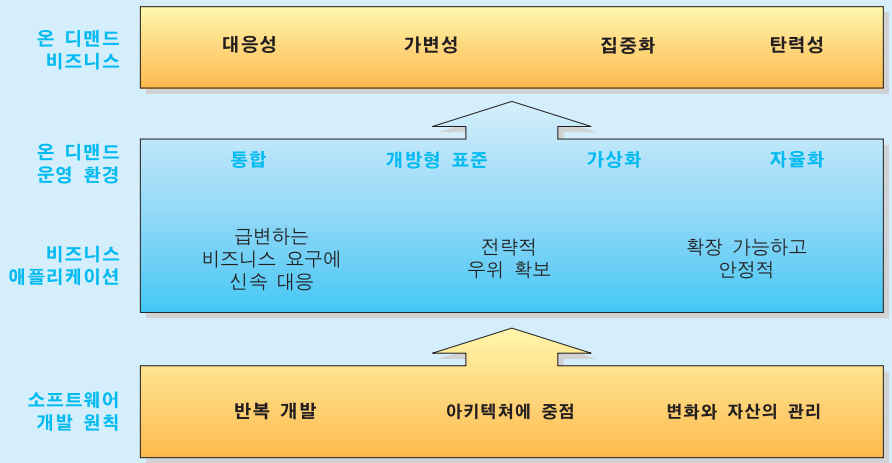
반복하면서 개발한다

반복 개발 프로세스를 거치면 소프트웨어를 직접 전개하기 직전까지 계속 개선된 버전들을 얻을 수 있다. 한 번 반복할 때마다 분석, 설계, 구축, 테스트의 결과가 달라질 것이며, 그 결과 소프트웨어를 시연하면서 어떤 부분은 수용하고 어떤 부분은 수정할 수 있다. 먼저 개발 팀은 그 프로젝트에서 리스크가 높은 기능들부터 해결하면서 소프트웨어를 개발한다. 그 다음 계속 반복하면서 기능들을 추가해 애플리케이션의 워킹 버전을 만들어낸다. 한 번 반복할 때마다 시스템 아키텍처를 검증하고, 애플리케이션 요구 사항들을 해결할 수 있는지를 확인하고 소프트웨어의 품질을 높일 수 있다.

소프트웨어 개발 팀이 반복 개발 프로세스를 따르면, 애플리케이션의 최종 사용자들을 개발 과정에 참여시켜 시스템의 워킹 버전을 직접 검사하도록 할 수 있다. 이것은 소프트웨어 개발 팀이 항상 결과에 초점을 맞출 수 있다는 점에서 매우 중요한 개념이다. 더욱이 Rational Unified Process(RUP) 같이 완성도가 높은 반복 개발 프레임워크를 사용한다면 어떤 규모의 프로젝트에서든 적용할 수 있는 유연한 프로세스를 활용할 수 있다. 개발 팀은 비즈니스 요구가 변하거나 새 애플리케이션의 필요성이 제기될 때 기존의 개발 과정을 변화에 맞춰 신속하게 개정할 수 있다.

이 반복적인 '결과 위주' 의 프로세스는 또한 기존의 '폭포' 방법론 같이 '행동 위주' 의 프로세스에 비해 훨씬 개선된 개발 프로세스이다. 폭포 방법론은 개발 과정에서 다음 단계(예, 코딩)가 시작되기 전에 전 단계(예, 설계)가 완료되어 전 단계의 결과가 다음 단계로 '쏟아져 내리는' 것이다. 이런 프로세스에서는 개발 팀이 시스템의 각 부분들을 별도로 책임지고 진행했다가 마지막의 조립 및 테스트 단계에 이르러서야 실수를 발견하게 되는데, 그제서야 크게 낙담하면서 작성한 코드를 파기하고 다시 작업하게 된다**(그림 5)**.

반대로 반복 개발 프로세스는 온 디맨드 운영 환경을



✓

[그림 4]

3가지 소프트웨어 개발 원칙은 온 디맨드 운영 환경에서 애플리케이션의 적응력과 안정성, 전략적인 가치를 높여 준다.

[그림 5]

기존의 폭포 방법론에 의한 프로젝트에서는 프로젝트의 최종 단계에 이르러서야 소프트웨어를 조립해 테스트할 수 있으므로 품질을 제품 기능 측면에서 평가하는 것이 아니라 애초 계획으로 평가할 수밖에 없다. 마지막 단계에서 문제가 발생하면(늘 그렇듯이) 개발 팀은 코드를 처음부터 다시 작성해야 하는데 이렇게 되면 비용이 많이 들고 목표 기일을 놓치게 되거나 최선이 아닌 차선의 애플리케이션을 제공할 수밖에 없다.

[그림 6]

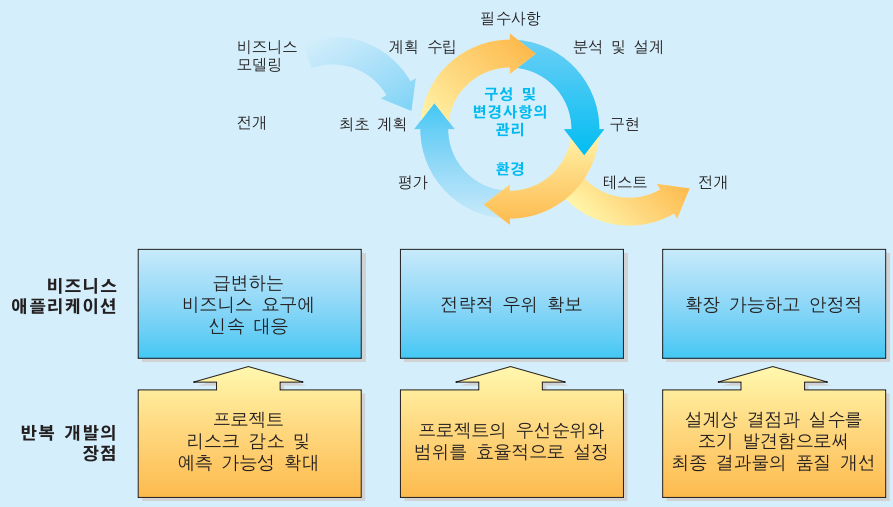
반복 개발 프로세스에서는 한 번 반복될 때마다 개발 중인 소프트웨어의 워킹 버전(짙은 회색 점)이 나오게 된다. 이를 통해 개발 팀은 '현재의 위치'를 파악하고 필요한 수정을 가할 수 있다.

위한 애플리케이션을 개발하는 데 있어 다음과 같은 장점들을 기본적으로 제공한다.

- **프로젝트 리스트 감소** ·· 즉 시스템 성능과 최종 사용자의 요구 사항 충족 등 가장 까다로운 문제들을 처음부터 고려한다. 개발 팀은 처음에 일부 주요 필수 사항들만 가지고 개발에 착수하는데 이렇게 하면 처음부터 어려운 문제의 해결에 초점을 맞출 수 있다. 초기에 주요 리스크를 해결하거나 줄여 놓으면 프로젝트의 후반 단계에서 결정적인 실수가 나타날 가능성이 희박해지며, 최종 목표일 – 온 디맨드 비즈니스에서는 결코 연기할 수 없는 – 도 쉽게 맞출 수 있게 된다.
- **예측 가능성의 확대** ·· 기존의 '폭포' 방법론을 따르는 소프트웨어 개발 팀은 최종 조립 단계에 이르러서야 소프트웨어의 워킹 버전을 얻을 수 있으므로, 그 전까지는 제품 품질을 총체적으로 평가할 수 없다. 이 후반 작업에서 실수가 발견되면 – 이것은 늘 있는 일이다 – 애플리케이션을 재작업하거나(상당한 비용 초과를 유발하고 고객과의 약속 기일을 놓치게 된다) 고객 기대치에 못 미치는 애플리케이션을 제공하게 된다. 그러나 반복 개발 프로세스를 따르면 각각의 반복 과정에서 설계하고 코드 작성하고 테스트하므로 최종 결과를 비교적 정확하게 예측할 수 있고, 그 결과 애초의 목표대로 작업 결과를 내놓게 된다. 프로세스의 한 단계에서 문제가 발생하는 경우 다음 반복 과정에서 그 점을 수정, 적용할 수 있다.
- **적정 범위의 설정** ·· 반복 개발 프로세스에서는 프로젝트 관리자가 첫 단계에서 마지막 단계까지 포괄적으로 달성 목표를 설정하는 것이 아니라 각 반복 과정에서 그때그때 개선시킬 범위를 정할 수 있다. 이렇게 함으로써 개발 팀은 애플리케이션에서 가장 중요한 부분에 중점을 두고서 최종적으로 완성도 높은 시스템을 개발할 수 있다.
- **설계상의 결함 축소** ·· 가장 리스크가 높은 부분을 처음에 해결, 리스크를 줄이고, 반복 과정에서 소프트웨어 품질을 계속 검증함으로써 보다 안정적인 고품질의 애플리케이션을 얻을 수 있다. 이것은 통합 작업에서든 새 애플리케이션 개발 작업에서든 아니면 기존 시스템의 확장 작업에서든 모두 해당된다.

반복 개발 : 비즈니스 애플리케이션에 미치는 혜택

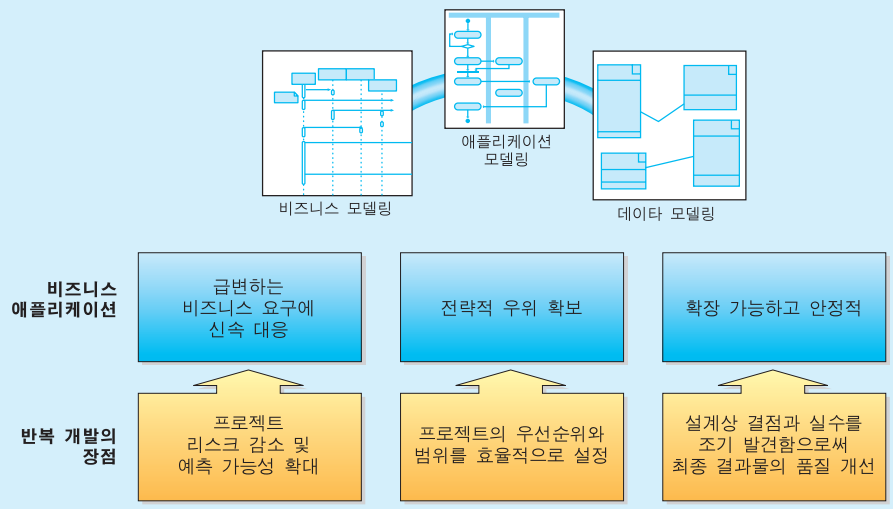
반복 개발 프로세스는 온 디맨드 운영 환경에서 실행되는 애플리케이션의 주요 목표들을 달성하는 데 필수적이다. 왜냐하면 반복 개발은 리스크는 줄이는 반면 예측 가



✓

[그림 7]

반복 개발 프로세스는 온 디맨드 운영 환경에서 실행되는 애플리케이션의 주요 목표들을 지원할 수 있다.



✓

[그림 8]

아키텍처에 중점을 두면, 온 디맨드 운영 환경에서 실행되는 애플리케이션에 많은 혜택들을 제공할 수 있다.

능성은 높여주므로 소프트웨어 개발 팀은 어떤 애플리케이션이든 – 기존에 사용하던 것이든 패키지든 아니면 새로 개발하는 애플리케이션이든 – 급변하는 비즈니스 요구에 맞춰 신속하게 수정할 수 있다. 따라서 애플리케이션에 대한 ROI가 크게 향상될 것이다. 또 반복 개발은 프로젝트 범위를 더욱 효율적으로 설정할 수 있도록 해주므로 기업의 전략적 요구에 한층 잘 부응하는 애플리케이션을 제공할 수 있다. 그리고 반복 개발 과정에서 디자인상의 결점들을 제거함으로써 IT 관리자는 최종 전개할 애플리케이션의 확장성과 안정성을 확신할 수 있을 것이다. **[그림 7]**은 온 디맨드 운영 환경에서 실행되는 애플리케이션의 주요 목표들을 반복 개발 프로세스가 얼마나 잘 지원할 수 있는지 그 연관성을 보여주는 것이다.

아키텍처에 중점을 둔다

애플리케이션의 아키텍처는 애플리케이션의 성공여부를 결정짓는 주요인이다. 아키텍처가 잘 설계된 애플리케이션은 비즈니스 요구를 충족시키고, 만족스럽게 실행되며, 기업의 발전에 맞춰 확장할 수 있고, 시간이 흘러도 수정하면서 활용할 수 있다. 그러나 아키텍처가 잘못 설계된 애플리케이션은 유연성이 부족하고, 쉽게 손상되고 안정적이지 못하며 비용이 많이 들게 된다.

애플리케이션이 기업의 급변하는 요구들을 끊임없이 지원할 수 있어야 하는 온 디맨드 비즈니스에서 소프트웨어 아키텍처의 중요성은 더욱 배가된다. 안정적이며 쉽게 파악할 수 있는 아키텍처는 필수 요건과 컴포넌트 등 주요 작업물 20%의 토대가 됨으로써 새 프로젝트의 성공 여부를 좌우하게 될 것이다. 주요 부분을 담당할 아키텍처에 집중함으로써 온 디맨드 개발 팀은 소프트웨어의 세부 사항들에 신경 쓰기 전에 리스크가 높은 필수 사항부터 해결할 수 있다. 그 결과 전체 프로젝트 진행 과정에서 작업 결과의 파기나 재작성 비율이 그만큼 줄어들어

애플리케이션의 아키텍처는 애플리케이션의 성공여부를 결정짓는 주요인이다. 아키텍처가 잘 설계된 애플리케이션은 비즈니스 요구를 충족시키고, 만족스럽게 실행되며, 기업의 발전에 맞춰 확장할 수 있고, 시간이 흘러도 수정하면서 활용할 수 있다.

온 디맨드 비즈니스의 요구에 신속하게 대응할 수 있다.

아키텍처에 중점을 둔다는 것은 컴포넌트 기반의 설계, 소프트웨어 위주의 아키텍처 그리고 비주얼 모델링의 3가지 측면에서 이뤄진다.

컴포넌트는 기존의 코드 라인들을 결합한 세트로 소스 상태일 수도 있고, 인터페이스와 행동양식(behavior)을 갖춘 실행 가능한 포맷 상태일 수도 있다. 인터페이스가 잘 설계되어 있는 컴포넌트에 기반한 소프트웨어 아키텍처는 애플리케이션의 나머지 부분에 영향을 미치지 않고 컴포넌트만 수정할 수 있으므로 아키텍처를 신속하게 변경시킬 수 있다. 현재 소프트웨어 아키텍처 설계에서 각광 받는 것이 '서비스 지향 아키텍처(SOA : Service-Oriented Architecture)' 모델이다. SOA 모델을 채택한 기업은 컴포넌트 기반 설계 원칙에 따라 내부 시스템들을 통합할 수 있을 뿐만 아니라 기업이 선택한 특정 기능들을 외부 고객에게도 제공할 수 있다. 또 기업은 SOA 모델을 통해 온디맨드 비즈니스로 변신하는 데 도움이 될 소프트웨어들을 구축, 통합할 수도 있다.

효율적인 아키텍처를 설계하기 위해서는 일반적으로 많은 관계자들이 참여하게 된다. 따라서 이 어려운 과제를 성공시키려면 효율적인 커뮤니케이션, 명확한 정의, 설계 디자인을 캡처, 수정하기 위한 안정적인 방법들이 필수적이다. 이런 요구를 해결하기 위해 소프트웨어 업계에서는 현재 UML(Unified Modeling Language)이라는 표준을 개발했다. 개발 팀은 UML을 사용해 애플리케이션의 아키텍처를 명확하게 그래픽으로 표현할 수 있다. 그리고 이 UML의 정확한 정의를 통해 다른 사람들도 이 UML 모델을 쉽게 파악, 이해할 수 있다. UML 모델을 만드는 가장 빠르고 효과적인 방법은 비주얼 모델링 툴을 사용하는 것이다. 오늘날 같이 복잡한 세계에서 비주얼 모델링 툴은 완성도 높은 아키텍처를 구현하는 데 필수적인 요소이다. 비주얼 툴은 단순한 드로잉 툴에 그치지 않고, 비주얼 모델에서 바로 코드 등을 작성해내므로 사람에 의한 실수의 가능성을 없애 주고, 그만큼 생산성을 높여 준다.

온 디맨드 시대에 아키텍처에 초점을 맞추면 다음과 같은 주요 혜택을 누릴 수 있다.

- **변화에 대비한 설계** ·· 웹 서비스와 그리드 컴퓨팅 지원을 목적으로 하는 '서비스 지향 아키텍처(SOA)' 등 컴포넌트 기반 아키텍처는 전체 시스템의 완전성을 손상시키지 않으면서 컴포넌트만 교체 · 수정할 수 있으므로 소프트웨어를 빨리 변경할 수 있다. 소프트웨어의 신속한 변경은 온디맨드 비즈니스에 요구되는 신속한 응답과 탄력성을 지원하기 위한 기본적인 필수요건이다.
- **복잡성의 감소** ·· 비주얼 모델링을 이용하면 프로젝트 관리자들은 소프트웨어 애플리케이션의 핵심 요소에만 신경 쓰면서 세부 기능들은 팀의 코딩 전문가들에게 맡길 수 있다. 컴포넌트 기반 아키텍처에서는 코드 덩어리들이 서로의 메소드나 세부 사항들을 자잘하게 드러내지 않아도 시스템의 여러 부분들 간의 인터랙션이 가능하다. 그리고 비주얼 모델링과 컴포넌트 기반 설계를 같이 이용하면 모든 팀이 '적당한 수준의 추상'을 기반으로 작업할 수 있다. 다시 말해 프로젝트에 참여하는 모든 팀이 모든 부분의 세세한 사항들까지 알 필요 없이 각 팀이 맡은 분야에서 필요한 정도로만 알면 되는 것이다.
- **완전성과 완성도** ·· 많은 소프트웨어 개발자들이 애플리케이션의 아키텍처가 애플리케이션의 품질을 결정짓는 가장 중요한 요소라는 데 동의하고 있다. 아키텍처가 훌륭한 애플리케이션은 오랜 시간이 지나도 수정해서 쓸 수 있다. 그러나 아키텍처가 잘못된 애플리케이션은 융통성도 없고 손상되기 쉬운 디자인 때문에 수정을 가하면 그만큼 기능과 성능이 떨어지게 된다.

아키텍처 포커스 :
비즈니스 애플리케이션에 미치는 장점

아키텍처에 중점을 두면 온디맨드 운영 환경에서 실행되는 애플리케이션들의 주요 목표들을 지원할 수 있다. 잘 설계된 아키텍처는 변화에 맞춰 애플리케이션을 변경할 수 있도록 해주므로 급변하는 비즈니스 요구에 더욱 잘 대처할 수 있게 된다. 확신을 갖고 신속하게 애플리케이션을 변경할 수 있다면 기업도 비즈니스 경쟁에서 전략적 차별성에 초점을 맞출 수 있게 되고, 비즈니스 애플리케이션에서 높은 투자효과(ROI)를 얻을 수 있다. 좋은 아키텍처는 애플리케이션의 완전성과 품질을 높여 주므로 그 결과 애플리케이션의 안정성과 확장성이 높아지게 된다. **(그림 8)**은 아키텍처에 중점을 두었을 때 온디맨드 환경에서 실행되는 애플리케이션의 주요 목표들을 어떻게 지원할 수 있는지를 보여주는 것이다.

변화와 자산을 관리한다

e-business on demand에 내포되어 있는 주요 컨셉 중 하나가 바로 변화에 대한 고려이다. 새로운 기회와 고객 요구 또는 새로운 위협 및 전반적인 비즈니스 안정성에 신속히 대응할 수 있는 능력은 온 디맨드 시대에 매우 중요한 결정적인 요인이라고 할 수 있다. 그러나 온 디맨드 소프트웨어 개발에서 '신속한 대응'은 단지 '즉각적인 반응'을 의미하는 것은 아니다. 변화에 대해 조직적으로 대응한다는 것은 개발 과정에서 만들어진 자산의 변조나 파기를 예방하는 것으로, 이렇게 함으로서 프로젝

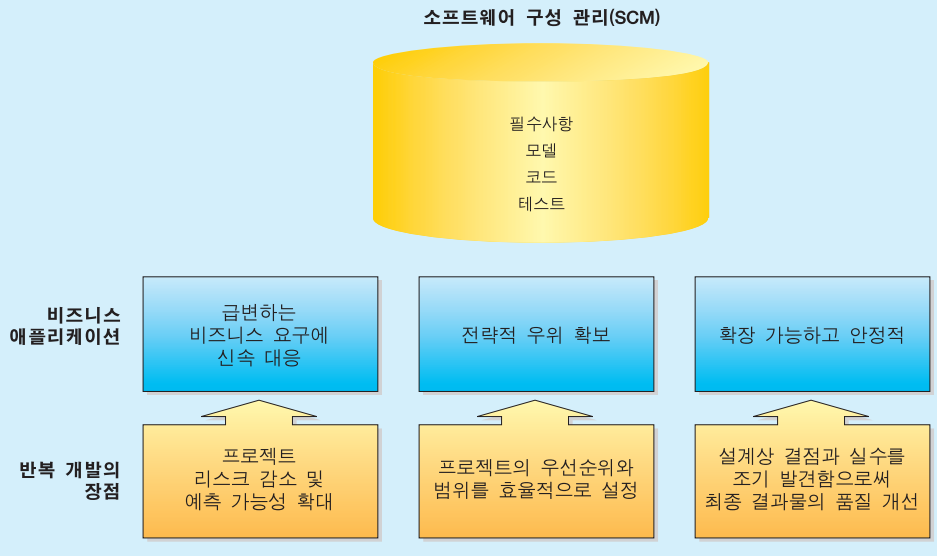
트 관리자는 연속적으로 반복되는 개발 프로세스를 원활하게 진행할 수 있다. 소프트웨어 구성 관리(SCM : Software Configuration Management)는 현 대 소프트웨어 개발에서 매우 중요한 기능이다. SCM은 필수 사항 자체의 변화 등 여러 가지 변화 요인에도 불구하고 전체 프로젝트 라이프사이클에서 필수 사항들을 자세히 추적할 수 있도록 해준다.

변경 사항과 자산을 관리하면, 온 디맨드 소프트웨어 개발에서 다음과 같은 중요한 이점들을 누릴 수 있다.

- **가상 팀의 구축 및 동시 개발** ·· SCM 시스템을 이용하면, 한 프로젝트를 두고 여러 개발 팀이 동시에 프로젝트의 일부분들을 때로는 중복해서 작업할 수 있다. 따라서 완성도의 저하 없이 더 많은 작업을 더 빨리, 요구에 맞춰 바로 수행해 낼 수 있다.
- **주요 자산의 보호** ·· 기업의 소프트웨어 개발 자산들 – 필수 사항들을 정리한 문서, 설계 모델, 소스 코드, 자동화된 테스트 스위트 등 – 은 외부에서 구입하거나 외부 소스를 이용해 만들어낼 수 없는 기업 고유의 전략적 자원이다. 기업의 비즈니스 자산만큼 중요한 이들 소프트웨어 개발 자산은 마땅히 보호하고 관리해야 한다. 효율적인 변경 사항 관리 시스템이라면 현재 개발 중인 코드 유닛이나 컴포넌트를 분실하거나 덧씌워지지 않도록 예방해야 한다. 이렇게 함으로써 보안 침해나 재해 발생의 위협으로부터 안전하게 개발 자산을 보호할 수 있다.
- **확신에 찬 소프트웨어 전개** ·· 변경 사항과 자산을 관리하면, 복잡한 시스템을 구축, 관리하는 팀들이 소프트웨어 코드의 다양한 부분이나 다른 버전들을 결합할 때도 쉽게 동기화할 수 있도록 해준다. 또 변경 사항 관리 시스템은 전체 프로젝트 진행 과정 동안 모든 필수 사항들을 추적해, 하이 레벨의 아키텍처가 사용자의 기대치에 중점을 둔 소프트웨어 시스템으로 전환되도록 보장한다.

변화와 자산의 관리 :
비즈니스 애플리케이션에 미치는 장점

소프트웨어 개발 과정에서의 변경 사항과 자산의 관리는 온디맨드 운영 환경에서 실행되는 애플리케이션들의 주요 목표들을 충족시키는 데 꼭 필요한 작업이다. 가상 팀의 구축 능력과 동시 개발 능력을 갖추면 프로젝트에 소요되는 시간을 줄여 급변하는 비즈니스 요구에 신속하게 대응할 수 있다. 소프트웨어 개발 자산들은 최종 결과물인 애플리케이션 못지않게 기업에 중요한 자산으로, 이 개발 자산들을 보호하고 관리하는 것도 전략적 가치 창출에 기여할 수 있다. 그리고 소프트웨어를 전개할 때 개발 팀이 소프트웨어의 완성도에 대해 확신할 수 있다면 그것 자체로 큰 장점이다. **(그림 9)**는 변경사항과 자산을 관리하는 것이 온디맨드 운영 환경에서 실행되는 애플리케이션의 주요 목표들을 지원하는 데 어떻게 기여하는지를 보여 준다.



[그림 9]
소프트웨어 개발 과정에서 변경 사항과 자산을 관리하면, 온 디맨드 운영 환경에서 실행되는 애플리케이션의 주요 목표들을 지원할 수 있다.

소프트웨어 개발 과정에서의 변경 사항과 자산의 관리는 온 디맨드 운영 환경에서 실행되는 애플리케이션들의 주요 목표들을 충족시키는 데 꼭 필요한 작업이다. 가상 팀의 구축 능력과 동시 개발 능력을 갖추면 프로젝트에 소요되는 시간을 줄여 급변하는 비즈니스 요구에 신속하게 대응할 수 있다.

온 디맨드 운영 환경의 지원

앞에서 설명했듯이, IBM은 온 디맨드 운영 환경의 특성을 통합, 개방형 표준, 가상화, 자율성의 네가지로 정의하고 있다. 소프트웨어 개발 팀은 팀의 모든 노력을 이 새로운 환경을 구현하는 데 집중해야 하며, 그렇게 함으로써 기업은 고객에게 더 많은 가치를 더 빨리 제공할 수 있게 된다.

효율적인 소프트웨어 개발 작업은 온 디맨드 운영 환경의 4가지 특징 중 통합과 개방성의 특징을 제공할 수 있다. 가상의(그리드 컴퓨팅 기반의) 자율적인(자가관리) 시스템이 향후 보편적인 비즈니스 컴퓨팅 플랫폼이 되면, 통합되고 개방된 운영 환경을 갖추고 있는 기업이 온 디맨드 환경의 장점들을 가장 잘 활용할 수 있게 될 것이다.

여기서는 온 디맨드 고객과 비즈니스에서 통합 비즈니스 애플리케이션과 개방 표준의 중요성에 대해 자세히 알아본다.

통합

e-business의 발전에서 가장 중요한 단계는 이종의 컴퓨팅 기능들을 인터넷으로 통합해 고객들이 웹을 통해 실제 비즈니스를 수행하기 시작한 때라고 할 수 있다. 현재 많은 기업에서 고객과의 기본적인 트랜잭션들을 가능하게 해주는 상대적으로 단순한 소프트웨어들을 이용하고 있다. 그러나 온 디맨드 컴퓨팅을 지원하기 위한 통합 요구 사항은 이보다는 좀더 복잡하다. 하지만 그 혜택은 상상 밖으로 클 것이다.

현재 총 IT 지출의 40%가 통합 작업 – 단순히 상호 연결되어 작동되도록 하는 – 에 지출되고 있다. 이런 양상은 지난 몇 년 동안 기존 시스템과 커스텀 애플리케이션 등 상당한 자산들이 누적된 결과로 이제 이 자산들을 적절히 안배해 고객과 기업 내부에서 증대되는 요구들을 해결해야 하는 것이다. 온 디맨드 컴퓨팅에서 소프트웨어 개발자들은 보다 경제적인 방법으로 더욱 빠르게 고객에게 다가가기 위해 새로운 형태의 통합 작업을 추진해야 한다. 이 새로운 형태의 통합 작업에는 웹 서비스와 서비스 지향 아키텍처(SOA) 같은 새로운 기법들이 요구되는데, 이것들은 궁극적으로 통합 작업에 따르는 부담을 상당 부분 덜어주어, 개발 에너지를 다른 곳으로 돌려 더 큰 비즈니스 가치를 창출할 수 있도록 해줄 것이다.

소프트웨어 개발 조직이 전문성을 집중적으로 키워야 하는 통합 분야는 세가지가 있다. 서비스 지향 아키텍

개방 표준을 사용한다고 해서 기업 전체에서 동일한 기술을 사용해야 한다는 것은 결코 아니다. 개방 표준을 사용한다는 것은 e-business 네트워크에서의 데이터 교환이나 트랜잭션을 다른 시스템에서 예측할 수 있는 방식으로 처리, 공유한다는 것으로 이것은 고객의 주요 요구 사항이기도 하다.

처, 엔터프라이즈 아키텍처 그리고 이기종 통합이다.

- **서비스 지향 아키텍처(SOA)** ·· SOA는 컴포넌트 기반 개발의 다음 단계이다. 현재의 통합 작업들을 지원하는 웹 서비스를 구현하는 데 핵심 역할을 담당하는 SOA는 기업이 컴포넌트 기반 설계 원칙에 따라 사내 시스템들을 통합할 수 있도록 할 뿐 아니라 기업이 선택한 일부 기능에 대해 외부의 고객들이 액세스할 수 있도록 해준다. 즉 SOA는 기업이 온 디맨드 비즈니스로 전환하는 데 도움이 될 소프트웨어를 구축, 통합할 수 있도록 해준다.
- **엔터프라이즈 아키텍처** ·· 엔터프라이즈 아키텍처(시스템 아키텍처 등)는 기업이 현재 무엇을 갖고 있고, 무엇이 중복되어 있는지를 파악하며, 최적의 아키텍처에서 정보를 공유할 수 있는 최선의 방법을 찾아낼 수 있도록 해준다. 시스템 설계자는 시스템의 여러 부분들이 인터랙트할 방법, 주요 스트레스 포인트, 그리고 시스템이 해결해야 할 비즈니스 과제와 범위를 파악하고 있어야 한다. 이상적으로는 시스템 아키텍처에 따라 시스템을 구축, 관리할 팀을 결정하는 것이 좋다. 그렇게 하면 개발 프로젝트에 참여하는 팀들이 아키텍처의 특정 부분에 대해 더 많은 권한과 책임을 부여 받게 되어 프로젝트 진행에 더욱 적극적으로 참여하게 될 것이다.

  대규모 시스템 통합 프로젝트를 지원하는 시스템 아키텍처는 소프트웨어 디자인에서 비즈니스의 연관 분야들도 통합한다. 이런 경우 소프트웨어 개발 조직은 연관 분야의 특성에 맞춰 시스템 개발에 접근함으로써 개발 노력의 가치를 높일 수 있게 된다.
- **이기종 통합** ·· 앞에서도 언급했듯이 온 디맨드 시대의 기업은 전사적으로, 나아가 주요 협력사와 핵심 고객까지 데이터를 공유할 수 있도록 시스템을 통합해야 한다. 이 통합 작업을 위해서는 소프트웨어 애플리케이션의 통합에 대해서 뿐만 아니라 외부 세계와 기업을 원활하게 통합시켜 주는 엔드투엔드 아키텍처에 대해서도 이해해야 한다. 이를 위해서는 다음과 같이 하는 것이 필요하다.
  - 요구사항들을 잘 이해하고 관리해야 한다 ·· 일반적으로 개개 시스템들은 서로 다른 목적과 서로 다른 툴, 언어로 설계되었기 때문에, 개발 팀이 시스템 통합 프로젝트를 추진하기란 여간 어려운 일이 아니다. 비즈니스 측면에서 통합의 필요성과 중요성에 대해 이해해야 하며, 통합 작업을 수행할 기술적 방법에 대해서도 충분히 숙지해야 한다.
  - 반드시 견실한 아키텍처를 기반으로 통합해야 한다 ·· 애플리케이션을 개발할 때 향후 변화에 관계없이 신속한 대응과 탄력성을 제공할 수 있도록 구현하는 것처럼, 통합 작업도 이후의 변화를 고려해 설계해야 한다.
  - 통합 작업의 결과는 자동 테스트를 통해 검증되어야 하며, 지속적으로 높은 완성도를 달성하는 데 기여해야 한다.
  - 시스템 통합은 반드시 이기종 환경을 지원하는 표준에 기반해 이뤄져야 한다(아래의 '개방' 부분 참조).

개방형 표준

WWW이 크게 확산되면서(모든 데스크톱에서 HTML을 보여주는 브라우저를 실행하고 있는 데서 단적으로 드러나듯) 개방 표준을 채택하는 경우도 크게 늘어나고 있다. 많은 기업들이 온 디맨드 컴퓨팅 시대로 진입하게 되면 개방 컴퓨팅 표준은 모든 통합 작업들(바로 앞에서 살펴보았던)을 추진하는 데 산소와 같은 역할을 담당하게 될 것이다. 온 디맨드 비즈니스의 성공 여부는 사내·외 통합(글로벌 시장에서 신규 고객을 유치, 계속 유지하는 데 목적을 둔)을 위한 방법으로서 개방 소프트웨어 개발 표준을 사용하느냐로 판가름될 것이다.

개방 표준을 사용한다고 해서 기업 전체에서 동일한 기술을 사용해야 한다는 것은 결코 아니다. 개방 표준을 사용한다는 것은 e-business 네트워크에서의 데이터 교환이나 트랜잭션을 다른 시스템에서 예측할 수 있는 방식으로 처리, 공유한다는 것으로 이것은 고객의 주요 요구 사항이기도 하다. 심지어 기업의 전략적 차별성을 목표로 개발되는 기술인 경우에도 향후 다른 시스템과의 통합 및 인터랙션을 반드시 염두에 두어야 한다.

다음 3가지 경향은 개방 컴퓨팅 표준의 중요성을 보여주는 것으로 온 디맨드 운영 환경을 채택하려는 기업에게도 유용한 지침이 될 것이다.

- **통합 모델링 언어(UML ; Unified Modeling Language)** ·· 널리 사용되는 표준인 UML은 전략적/기술적 의사결정들을 사람과 소프트웨어 개발 툴들이 해석할 수 있는 방식으로 캡처하기 위한 비주얼 표기 방법이다. 사람과 기계가 다 이해할 수 있는 표준 언어인 UML은 문화와 배경, 언어가 다른 팀 구성원들 간에 필연적으로 발생할 수 있는 혼란을 제거해 준다. 그 결과 소프트웨어 팀들은 더욱 효율적으로 작업할 수 있으며, 디자인을 결정할 때 고객 요구사항도 오해의 여지 없이 더욱 명확하게 파악해 구현할 수 있게 된다.
- **서비스 지향 아키텍처(SOA)** ·· 웹 서비스 분야에서 그 역할이 점점 커지고 있는 SOA는 XML, Java, SOAP, .NET, WSDL 등 여러 표준들을 포괄하는 구조로서 매우 중요하다. 기업은 이 모든 표준들을 이용해 사내에, 또 궁극적으로 사외의 벤더, 협력사, 고객과의 상호운용 및 통합을 위한 웹 서비스를 구축할 수 있다.
- **컴포넌트 기반 디자인(CBD)** ·· 객체 지향 소프트웨어 디자인의 주요 특징으로 인식되어왔던 컴포넌트와 컴포넌트 재활용은 온 디맨드 소프트웨어 아키텍처에서 또 한 번 그 가치를 인정 받고 있다. 표준 컴포넌트 모델들은 효과적인 애플리케이션 디자인과 상호운용을 가능하게 해줄 것이다.

MEMO

고객 리포트 : IBM Rational의 툴과 최상의 실례, 서비스를 사용하는 고객들은 투자효과(ROI)를 대폭 개선할 수 있었다고 응답했다.

- 개발 비용 33% 절약
- 950만 달러의 NQB (Net Quantifiable Benefit)
- 1440%의 ROI
- 409,000달러의 NQB
- 222%의 ROI

IBM Rational의 소프트웨어 개발 솔루션

앞에서 설명한 소프트웨어 개발의 3가지 원칙 – 반복 개발, 아키텍처에 중점, 변화와 자산의 관리 – 은 Rational 소프트웨어의 다면적 통합 솔루션의 토대이다. IBM Rational은 지난 20여 년간 일관되게 '소프트웨어 개발과 전개에 많은 비중을 두고 있는 비즈니스 고객들의 성공을 위해' 기여한다는 자세로 일해왔다. IBM은 온 디맨드 비즈니스에서는 주요 핵심 소프트웨어 개발이 상당한 비중을 가진다고 판단하고, 올해 2월 전격적으로 Rational을 인수했다.

IBM Rational 소프트웨어는 아키텍처 모델링 언어부터 날로 발전하고 있는 웹 서비스 표준까지 미래의 소프트웨어 개발 기술과 실례를 만드는 데 중요한 역할을 담당하고 있다. IBM Rational은 여러 가지 표준 제정 단체에 참여하고 있으며, 직원들이 발행한 책만도 50여 종이 넘고, 매년 이 직원들은 다양한 업계 행사에 주요 연사로 참여해 리더십을 발휘하고 있다.

여기에서는 Rational이 IBM 소프트웨어 포트폴리오에 제공하는 소프트웨어 엔지니어링의 좋은 실례들과 개발 툴 및 전문 서비스들을 소개할 것이다. 또 Rational의 제품들이 온 디맨드 운영 환경의 개방성과 통합에 어떻게 기여하는지 그 특징들도 소개한다.

온 디맨드 세계를 위한 소프트웨어 개발 환경

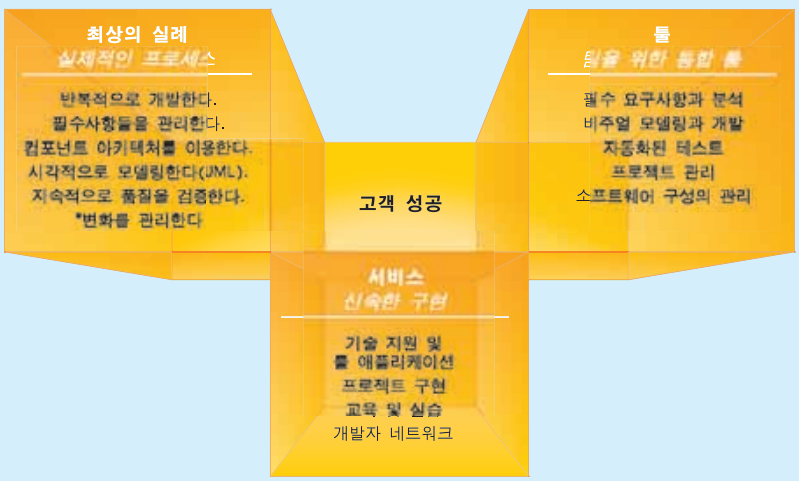
Rational의 소프트웨어 개발 플랫폼은 소프트웨어를 설계, 개발, 테스트, 전개할 수 있는 통합 솔루션이다. Rational 플랫폼은 최상의 소프트웨어 엔지니어링 실례와 우수한 툴, 그리고 전문가의 전문 서비스를 포함하고 있다. 소프트웨어 개발 팀은 이 세가지를 함께 이용함으로써 소프트웨어 개발 능력을 지속적으로 빠르게 향상시킬 수 있다.

최상의 실례와 Rational Unified Process : 실제적인 프로세스

IBM Rational 소프트웨어의 모든 전개 툴의 핵심에는 동사가 지난 몇 년간 소프트웨어 개발업체들과 협력을 통해 획득한 최상의 실례들이 들어 있다. 소프트웨어 개발 팀들은 이 모범 실례를 응용함으로써 목표를 정확하게 겨냥한 최신의 적절한 개발 프로세스를 구축할 수 있다. 여기서 제공되는 최고의 실례들은 :

- **반복적으로 개발한다** ·· 위험 요인이 프로젝트에 악영향을 미치기 전에 이들 요인을 파악, 제거하기 위해
- **필수 사항들을 관리한다** ·· 불가피한 변화에 대비해 탄력성을 확보하기 위해
- **컴포넌트 아키텍처를 이용한다** – 모든 프로젝트 참여자들이 쉽게 아키텍처를 이해할 수 있도록
- **시각적으로 설계, 개발한다** ·· 고품질 아키텍처를 생성, 유지하기 위해
- **지속적으로 품질을 검증한다** ·· 전체 개발 과정에서 품질을 보장하기 위해
- **변화를 관리한다** ·· 팀 내에서 또 회사 전체에서 효율적인 동시 개발이 가능하도록

이 최상의 실례들은 Rational Unified Process에 캡처되어 있는데, 현재 Rational Unified Process는 업계 프로세스 프레임워크의 사실상 표준으로 자리잡고 있다.



[그림 10]

Rational 플랫폼은 소프트웨어 엔지니어링의 최상의 실례와 우수한 툴 및 전문가에 의한 전문 서비스를 결합한 것이다.


“ IBM Rational 소프트웨어의 모든 전개 툴의 핵심에는 동사가 지난 몇 년간
소프트웨어 개발업체들과 협력을 통해 획득한 최상의 실례들이 들어있다.
소프트웨어 개발 팀들은 이 모범 실례를 응용함으로써 목표를 정확하게 겨냥한 최신의
적절한 개발 프로세스를 구축할 수 있다. ”

지식과 기술 전수에 역점을 두고 있는 IBM Rational의 서비스는 1,000여 명의 기술 전문가들과 20여 년에 걸친 경험 그리고 풍부한 온라인 교육 및 지식 자원을 포괄하고 있다.

고객 리포트 : Rational 툴의 시간 단축 효과

- 개발 사이클 시간 66% 단축 (9개월에서 3개월로)
- 2개월 걸리던 수동 테스트 작업이 2일로 단축
- 6~8명의 테스터가 며칠에 걸쳐 진행하던 테스트 작업을 1명의 테스터가 몇 시간만에 수행
- 예정일보다 66% 앞서 결과물 공개

고객 리포트 : 프로젝트 관리 능력의 향상

- 1년에 125개 프로젝트를 효과적으로 관리
- 2년만에 80명에서 280명으로 개발자 증가했으나 효율적으로 수용, 지원 가능
- 동시에 3개 이상의 제품 버전을 개발
- 3개국, 4개 내부 사이트, 300여 명의 개발자와 테스터를 포괄하는 경우에도 효율적으로 프로젝트 관리 가능

개발 라이프사이클의 자동화 :
프로젝트 팀을 위한 통일된 툴

Rational 툴들은 개발 전 과정을 통해 개인 개발자와 큰 규모의 개발 팀 모두를 지원한다. 수상 이력이 있는 이 혁신적인 툴들은 탁월한 통합 기능을 발휘해 지루한 작업들을 자동화시켜 주고, 워크플로우의 속도를 높여 주어 개발 주기를 단축시켜 준다. Rational은 소프트웨어 개발 라이프사이클을 다음과 같이 다섯가지 주요 단계로 구분하고 있다.

- **요구 분석(Requirements and Analysis)** ·· 개발 중인 시스템을 규정하고 전체 프로젝트 진행 과정에서 이 요구 사항들을 유지 관리하기 위한 작업이다
- **소프트웨어 개발(Visual Modeling and Development)** ·· 개발 중인 애플리케이션의 토대가 될 아키텍처와 컴포넌트들을 이해하고 설계하기 위한 작업이다
- **시스템 테스트(Automated Testing)** ·· 개발의 모든 단계에서 품질을 보증하기 위한 작업이다
- **프로젝트 관리(Project Management)** ·· 프로세스 가이드라인과 프로젝트 보고 및 진행 사항 점검 등 예산과 예정일에 맞춰 고품질 결과물을 완성하기 위한 작업이다
- **형상 관리/변경 관리(Software Configuration Management)** – 파일, 디렉토리, 컴포넌트 및 시스템의 작성 또는 구축 현황과 수정 사항을 관리하면서 소프트웨어의 결함 발견과 개정 작업을 추적하기 위한 작업이다.

서비스 : 수행 작업의 가속화

이 글에서 이미 여러 번 언급했듯이, 온 디맨드 컴퓨팅에서는 소프트웨어 개발 전문성이 매우 중요하다. 개발 팀이 확실한 경쟁력을 갖추기 위해서는 기술 지원, 컨설팅, 교육 등을 필요로 할 것이며, 전문가 조언 네트워크에 액세스할 수 있어야 한다. 지식과 기술 전수에 역점을 두고 있는 IBM Rational의 서비스는 1,000여 명의 기술 전문가들과 20여 년에 걸친 경험 그리고 풍부한 온라인 교육 및 지식 자원을 포괄하고 있다.

- **Rational 기술 지원** ·· 언제든지 문의할 수 있는 전화와 기술 지원을 통해 당면한 문제에 대해 신속한 해답을 구하려는 팀을 위한 지원
- **Rational 전문가 서비스** ·· 지식과 기술 전수를 통해 소프트웨어 개발 능력을 향상시키고 효율성을 높이는 데 중점을 둔 Rational 컨설턴트들이 제공하는 맞춤 프로젝트를 통해 개발 프로젝트의 어떤 단계에서든 작업 속도를 높일 수 있다.
- **Rational University** ·· 전세계 수십 개의 캠퍼스와 고객의 작업 현장에서 직접 교육하는 것으로, 소프트웨어 개발 라이프사이클을 포괄하는 60여 과정으로 구성되어 있다.
- **Rational Developer Network** ·· 온라인 기술 정보 네트워크로, 온라인 교육, 라이센스를 통해 제공되는 컨텐츠, 그리고 개발자들의 온라인 커뮤니티가 포함되어 있다.

온 디맨드 운영 환경의 지원

IBM Rational은 통합 운영 환경으로 온 디맨드 비즈니스를 구현하는 데 어떤 도움을 제공할 수 있는가?

여기서 기술한 3가지 소프트웨어 개발 원칙 – 반복 개발, 아키텍처에 중점, 변경사항과 자산의 관리 – 은 전사적 비즈니스 통합을 촉진하는 중요한 가이드가 될 것이다.

- **서비스 지향 아키텍처(SOA)** ·· IBM Rational은 '서비스 지향 아키텍처' 분야에서, 웹 서비스 구현과 연관된 다른 분야들에서 상당한 리더십을 발휘하고 있다. IBM Rational이 중점적으로 참여하고 있는 표준 제정 기관으로는, WSI (Web Services Interoperability) 조직, UDDI (Universal Description, Discovery and Integration) 프로젝트 그리고 RAPC (Reusable Asset Specification Consortium)를 들 수 있다.

IBM Rational은 엔터프라이즈 아키텍처와 관련해 탁월한 능력과 리더십을 갖추고 있으므로, 모든 고객들이 자사 시스템을 설계하거나 더 낫게 개선하는 데 큰 도움이 될 것이다.

고객 리포트 : 괄목할 만한 생산성 증대 효과

- 해결해야 할 버그 90% 감소
- 300% 생산성 증대
- 100% 생산성 증대
- 버그 90% 감소
- 400% 생산성 증대
- 97% 생산성 증대
- 개발자 생산성 1200% 증대

- **엔터프라이즈와 소프트웨어 아키텍처** ·· IBM Rational은 엔터프라이즈 아키텍처와 관련해 탁월한 능력과 리더십을 갖추고 있으므로, 모든 고객들이 자사 시스템을 설계하거나 더 낫게 개선하는 데 큰 도움이 될 것이다. Rational은 또 현재 업계 표준인 UML의 개발에 크게 기여했다. 오늘날 IBM Rational의 시스템 아키텍처 접근 방식은 전체론적인 것으로 통합 툴과 함께 통합된 프로세스를 제공하며, 앞에서 논의한 여러 분야, 필수 요구 사항의 관리, 비주얼 모델링 및 UML, Rational Unified Process, 자동화된 테스트 및 구성 사항 및 변경 사항의 관리 등을 포괄하고 있다. 각각의 단계에서 IBM Rational은 앞선 툴과 컨설팅 서비스를 제공한다.
- **이기종 세계의 지원** ·· IBM Rational은 주요 플랫폼 벤더들과 전략적 협력 관계를 유지하고 있으며, 다양한 기술과 표준들을 지원하므로 고객들은 전체 시스템을 통합하고 정보를 공유할 수 있다. IBM Rational은 Java와 J2EE, .NET, Linux, SOAP, XML, WSDL, 임베디드 운영체제 및 C, C++ 등의 언어를 지원한다.

IBM Rational은 온 디맨드 비즈니스가 개방 운영 환경을 달성하는 데 어떤 도움을 제공할 수 있는가?

오래 전부터 오픈 컴퓨팅을 개발하고 지원해 온 IBM Rational은 소프트웨어 개발 분야에서 현재 전세계를 선도하고 있다. IBM Rational은 몇 년 전부터 표준 사양을 제정, 관리하는 다양한 컨소시엄 뿐만 아니라 컴퓨팅 플랫폼과 컴퓨팅 표준 제정에 참여하고 있는 주요 벤더, 기관과 협력하고 있다. 또 IBM Rational은 수많은 표준 제정 기관에 적극적으로 참여하고 있다.

- **UML** ·· IBM Rational이 개발한 UML은 객체관리그룹(OMG ; Object Management Group)에서 관리하는 표준 언어이다. OMG는 엔터프라이즈 애플리케이션의 상호운용을 위해 컴퓨터 사양들을 제정, 관리하는 비영리 컨소시엄이다.
- **개방 표준의 장려와 개발** ·· IBM Rational은 자사 제품에서 개방 표준을 적극 사용할 뿐만 아니라 여러 표준제정 기관에도 적극적으로 참여하고 있다. IBM Rational이 현재 참여하고 있는 표준 작업으로는, Business Process Management Initiative, Eclipse.org Consortium, Institute of Electrical and Electronics Engineers, Inc., Internet Engineering Task Force, Java Community Process, Organization for the Advancement of Structured Information Standards, Object Management Group, Reusable Asset Specification Consortium, Universal Description, Discovery, and Integration project, World Wide Web Consortium 및 Web Services Interoperability Organization이 있다.
- **서비스 지향 아키텍처 지원** ·· 오래 전부터 컴포넌트 기반 설계에 역점을 두어 온 IBM Rational이 서비스 지향 아키텍처를 지원하는 것은 자연스러운 결과이다. 점점 더 많은 기업들이 기업 방화벽을 넘어 외부의 고객과 협력사에게도 자사 시스템에 대한 액세스 권리를 제공함에 따라, 서비스 지향 아키텍처는 다른 어떤 컴포넌트 아키텍처보다 경제적으로 크로스플랫폼 통합 기능들을 제공할 수 있는 탁월한 대안이 될 것이다. 이 아키텍처를 채택하면, 지금까지 한두 가지 컴포넌트 모델에 치중했던 소프트웨어 개발 팀도 기존 소프트웨어를 계속 활용할 수 있으며, 기존 모델 포맷을 사용해 컴포넌트를 개발할 수 있고, 또 앞으로 SOA가 점점 보편화됨에 따라 그 혜택을 더 많이 누릴 수 있게 될 것이다.
- **20년 역사** ·· IBM Rational은 자사 툴에서 Ada부터 Visual Basic까지 대부분의 주요 표준 언어들을 지원하고 있다. 또 포괄적인 API를 제공, 써드파티의 툴이나 고객이 직접 개발한 툴도 IBM Rational 툴과 함께 사용할 수 있다. 그리고 여러 운영 플랫폼을 지원하므로 이기종 소프트웨어 개발 팀도 개방 표준을 사용해 전사 애플리케이션들을 상호 연결시킬 수 있다.

고객 리포트 :
소프트웨어 개발 라이프사이클 전체에서 개선된 점들

- 75만 라인의 코드 자동 생성
- 버그 80% 감소
- 가시성, 추적 가능성 및 예측 가능성의 향상
- 전체 테스트 시간 50~75% 감소
- 4백만 라인의 코드를 효과적으로 관리
- 필요한 문서 30% 감소
- 70개 국가 7,000명 컨설턴트의 커뮤니케이션과 협업 능력 개선

더욱 상세한 IBM Rational 고객 리포트는 www.rational.com/success 참조

비즈니스 시스템을 성공적으로 빠르게 바꾸기 위해서 기술 관리자들은 먼저 애플리케이션과 운영 환경이 지원해야 할 기업 목표를 분명하게 이해해야 한다. IBM의 온 디맨드 비전은 분명 주목할 만한 것으로, 비즈니스 시스템뿐만 아니라 전세계의 고객 및 협력사와 인터랙트할 방법까지 바꾸어 준다. 모든 업종의 모든 기업이 IBM 온 디맨드 비전을 채택함으로써 더 높은 완성도, 제품과 서비스의 더 저렴한 세계 공급 등의 혜택을 누릴 수 있다.

e-business on demand를 최대한 이용하기 위해서 소프트웨어 개발 책임자는 온 디맨드 시대가 요구하는 성공 원칙을 숙지하고 그에 따라야 한다. 즉 성공적인 소프트웨어 개발을 위한 3대 원칙에 중점을 두어야 하는 것이다.

- 반복해서 개발한다.
- 아키텍처에 중점을 둔다.
- 변화와 자산을 관리한다.

전세계 수천 기업들이 이 3대 원칙의 혜택을 확인하고 있다. 이 기업들에서 프로세스는 최종 결과를 중심으로 진행되며, 작업 결과물들은 잘 설계된 것으로 재활용 가능하고, 그 어느 때보다 향상된 능력을 바탕으로 작업하고 있다. 결과적으로 이 기업들은 온 디맨드 시대의 목표를 성공적으로 제공하고 있는 것이다.

IBM 소프트웨어 통합 솔루션

IBM Rational은 IBM 소프트웨어의 다른 제품들도 광범위하게 지원한다. IBM 소프트웨어 솔루션은 기업이 역점을 두고 있는 비즈니스와 IT 목표를 달성할 수 있도록 그에 필요한 파워를 제공할 것이다.

- DB2 소프트웨어는 데이터 활성화, 데이터 관리 및 배포를 위한 솔루션으로 정보 활용률을 높여 준다
- Lotus 소프트웨어는 지식의 저작, 관리, 커뮤니케이션 및 공유를 위한 솔루션으로 직원들의 생산성을 높여 준다
- Tivoli 소프트웨어는 e-business 인프라를 운영하는 데 필요한 기술을 관리할 수 있도록 있다
- WebSphere 소프트웨어는 기존의 비즈니스 크리티컬 프로세스들을 웹으로 확장할 수 있도록 해준다
- Rational 소프트웨어는 툴과 서비스, 최상의 실례를 통해 소프트웨어 개발 능력을 향상시킬 수 있도록 해준다.

IBM의 Rational 소프트웨어

IBM의 Rational 소프트웨어는 기업이 소프트웨어 개발 능력을 향상시킴으로써 새로운 비즈니스 가치를 창출할 수 있도록 해준다. Rational 소프트웨어 개발 플랫폼은 소프트웨어 엔지니어링의 최상의 실례와 툴, 서비스들을 통합하고 있다. 이를 이용해 기업은 더욱 신속하게 대응하고, 더욱 탄력적이며 중점이 명확한 기업을 구현함으로써 온디맨드 시대의 승자가 될 수 있다. Rational의 표준에 기반한 크로스플랫폼 솔루션은 소프트웨어 개발 팀이 비즈니스 애플리케이션과 임베디드 시스템 및 소프트웨어 제품들을 만들고 확장하는 데 도움이 될 것이다. 현재 포춘지 선정 100대 기업 중 98개 업체가 Rational 툴을 이용해 더 나은 소프트웨어를 더 빨리 만들고 있다. 더욱 자세한 정보는 www.rational.com과 www.therationaledge.com을 참조하기 바란다.