

현업 · 기술 공동 대처로 개발 '실패요소' 길목 차단

과연 성공적인 애플리케이션 구현을 위해서는 어떤 요구사항이 존재하는 것일까? 전문가들은 총책임자가 관리해야 하는 그룹을 제품, 프로그램, 개발, 품질보증, 사용자 교육 등 5개의 카테고리로 구분하고 있다. 이들 5가지 카테고리를 얼마나 유기적으로 통합, 관리하느냐에 따라 기업의 애플리케이션 환경이 성공과 실패의 기로에 놓이게 된다는 것이다. <편집자>



트라이크와 불의 판정 기준에 대해 3명의 야구심판이 인터뷰를 했다. 그들은 포수 뒤의 위치, 기술적인 세부사항, 주의사항 등에 대해 설명했다. 기자는 최종 판정을 어떻게 내리는지 물었다.

첫번째 심판은 "본대로 판정한다. 100% 정확한 판결은 어렵지만 그래도 본대로 판정한다"고 말했다. 두번째 심판은 "있는 그대로 판정한다"고 말했다. 세번째 심판은 웃으면서 "판정을 내릴 때까지 불인지 스트라이크인지는 아무도 알 수 없다"고 말했다.

S/W 개발의 조건

그렇다면 심판의 판정과 소프트웨어 개발에는 어떤 연관성이 있을까? 개발자들은 지난 수개월 동안 사용자의 요구에 따라 개발을 효율적으로 진행하는 방법에 대해 논의하고 생각했으며, 그것을 체계적으로 정리하기 위해 노력했다. 그러나 기업들이 개발 요구조건을 수집하는 과정을 정확히 정의하지 못했으며, 심판들은 이 문제를 정확히 알고 있어야 한다.

마이크로소프트의 오피스 팀은 최신 소프트웨어 개발에 필요

한 사항들을 문서로 정리했다.

이들 문서에서는 총책임자가 관리해야 하는 그룹을 제품 관리, 프로그램 관리, 개발, 품질보증(QA), 사용자교육 등 5개의 하위 그룹으로 구분해 설명하고 있다.

이 방식에서는 개발 요구조건을 몇 가지로 분류하게 된다. 제품 관리팀은 고객들이 소프트웨어에서 필요로 하는 기능이 무엇 인지를 먼저 파악해야 한다. 프로그램 관리 과정에서는 필요한 기능을 구현해 스펙이나 일정과 같은 소프트웨어 프로젝트 전반에 관련된 사항을 결정한다.

개발 팀은 기능 구현 방법과 소프트웨어 개발에 필요한 사항들을 정리해야 한다. 품질보증에서는 기능과 사용자 교육 계획을 테스트하고, 사용자 인터페이스와 고객 관련문서를 정리한다. 제품 출시 계획도 수립해야 한다.

평가 과정의 타당성 분석

고객의 '요구사항 파악'은 특별한 의미를 갖는다. 각 기능의 지원 여부에 따라 추가적인 판매가 이뤄지거나 제품이 사장될 수

있기 때문이다.

이와 같은 접근방식으로 모든 문제를 해결할 수는 없다. 제품 관리자는 핵심 기능의 효과를 과대포장하고, 개발 소요시간에 있어서 개발 팀의 노력을 빌미로 그럴듯하게 얼버무려 특정 기능을 지원하지 않는 경우가 있다.

평가 과정의 타당성을 살펴봄으로써 문제점을 해결할 수 있다. 제품 관리 평가는 판매에 비하면 다소 느슨하고 시장조사와 밀접한 관련을 갖고 있다. 개발 평가는 실제 개발비용을 기준으로 산정한다.

특정 개발 요구조건과 일반적인 소프트웨어 개발에 관련된 접근방식에서 최종 우선순위 설정에 관심이 집중되고 있다. 제품 출시일을 맞추는 과정에서 해당 제품의 기능이 제대로 지원되지 않거나 제외되는 경우도 있다.

이 같은 과정을 수행하기 위해서는 세번째 기준을 적용해야 한다.

업무 분석가 의견 수렴

이 방식은 엔터프라이즈 개발에는 적합하지 않다. 본대로 판정한다는 첫번째 심판의 기준이 적용되는 일반적인 엔터프라이즈 개발 방식으로는 만족할만한 결과를 얻을 수 없기 때문이다.

좀더 친숙한 엔터프라이즈 개발 환경을 구현한 기업에서는 프로젝트 관리자를 중심으로 작업을 진행하는 업무 분석가들을 중심으로 요구조건을 정의한다. 프로젝트 관리자는 프로젝트의 'IT 부문'을 담당하고 IT 팀은 프로젝트의 '업무 측면'을 담당하는 팀과 공동작업을 진행한다.

업무 담당자들은 IT 담당자들에게 시스템에서 수행해야 하는 작업을 설명한다. IT 담당자들은 논리적으로 완전하고 일관성 있는 정보를 기록한다. 정보 사이의 격차를 찾아내 업무 담당자에게 요구조건을 조정하도록 함으로써 완벽함과 일관성을 유지한다.

IT 팀은 첫번째 심판처럼 행동하고 모든 상황을 있는 그대로 받아들인다. 불행스럽게도 업무 사용자들은 기술 문서에 익숙하지 않고 엔지니어링 수준의 설명을 이해하지 못한다.

있는 그대로 판정한다는 기준이 필요하다. 기업 소프트웨어 환경에서 있는 그대로의 상황을 파악하려면 2가지 문제를 해결해야 한다. ABP(Abstract Business Process: 추상적인 업무 절차)라는 기본적인 업무상의 문제와 업무처리 모델이라는 기업 특

유의 구현방식이 바로 그 2가지 문제이다.

ABP는 명백하게 정의한 기본 업무과정으로 '어떻게'가 아닌 '무엇'에 관한 문제이다. ABP는 업계에서 일반화돼 있으며 조정이나 논의가 불가능하다. 이 모델에서는 문제를 수학적 또는 엔지니어링 측면에서 표시해야 하며 사용자와의 대화를 통해 파생되는 것이 아니고 텍스트북과 같이 문서로 정리된 것이다.

프로세스 리엔지니어링

기본적으로 이 모델은 기본 문제에 초점을 맞추고 있으며, 일정한 방식에 따라 확장된다. 요구조건이라는 측면에서 APB와 소프트웨어는 전문가가 개발 팀에게 난해하고 어려운 개념을 설명해야 하는 경우를 제외하면 사용자의 별다른 작업 없이도 구현할 수 있다.

이러한 방식으로 업무 프로세스에 대해 심도있는 이해를 가져올 수 있다. 나아가 시스템을 구축하고 주요 업무를 수행하기 위해 엔지니어들을 교육시킬 수도 있다.

ABP는 요구하는 시스템 기능을 객관적으로 정의할 수 있기 때문에 마이크로소프트 모델과 유사한 방식으로 구축될 수 있다.

고객들과 제품 판매에 대해 논의하기 이전에 제품 관리 팀은 현재의 제품 포트폴리오를 리버스 엔지니어링(Rreverse-engineering)해 현재의 작업 지원에 필요한 기능을 이해해야 한다. 또 최소 기능을 정의한다.

추상 모델위에 구현될 업무과정 모델은 기업의 운영방식을 설명한다. 업무과정 모델 개발은 기술과 관련이 없는 업무 사용자에 의해 이뤄진다. 업무과정 모델과 관련된 선택사항은 엔지니어링 측면에서 볼 때 다소 독단적인 부분이 있다.

리엔지니어링한 제품과 신규 소프트웨어의 작업 흐름을 정의하는 과정에서 신규 시스템 구현의 가치를 찾을 수 있다. 이 같은 정의를 작성하기 위해서는 사용자 팀이 업무과정 리엔지니어링에 정통해야 하며 BPR 모델로 이해될 수 있다.

이와 같은 방식으로 문제에 접근하면 엔지니어링 프로세스와 업무설계 프로세스를 분리할 수 있기 때문에 성공 확률이 높아지고 초기 지원분야를 줄이며 기술 팀이 기본 업무를 파악할 수 있다는 강점을 가지게 된다. 