



특별 좌담 : 특집 필진의 이신전심 '수다'

정말 빨리, 잘 만들 수 있을까?

이우리 계획을 신중히 세워도 일정은 자꾸 밀린다. 그래서 좋다는 개발도구도 써보고 패턴이나 방법론이나 하는 것들도 익혀도 보고, 빔을 새기며 열심히 만들었지만 여기저기서 '발레'가 득실거리고 돌아오는 반응도 시원찮다. 아마 모든 개발자들에게 기한을 맞추는 것과 품질 높은 소프트웨어를 만드는 이 두 문제는 영원한 '두 마리 토끼' 일 것이다. '두 마리 토끼를 잡기 위해' 준비한 이번 특집에서 필자들은 과연 어떤 생각을 하고 있는지 들여보기로 했다. 2002년 10월 17일 저녁 회사 업무나 개인 사정으로 참석하지 못한 몇몇 필자를 제외하고 가볍고 격식 없는 '수다'를 나눠 보았다.

미소 : 참석해 주셔서 감사합니다. 미리 말씀드린 것처럼 이번 특집은 초보자, 사회 초년생들이 실제 개발 현장에 부딪힐 두 가지 문제, 개발 속도와 품질 향상에 대한 실천 기법을 소개하기 위해 기획했습니다. 오늘 자리는 그동안 글을 쓰면서 느낀 점들을 서로 나눌 수 있는 자리로 마련했습니다. 그런데 원고는 어느 정도 쓰셨나요(은근히 독촉함)?

신승근(이하 신) : 사례 수집이 좀 힘들었습니다. 워밍업을 오래 한 만큼 하루(?)만에 쓸 수 있을 것 같습니다.

박지순(이하 박) : 피상적이지 않게 쓰는 것이 어려웠습니다. 제가 맡은 팀 차원의 실천 기법이란 것은 사실 '직접 경험해야' 아는 것이거든요.

신 : 사실 팀 체계를 제대로 갖추지 못한 곳도 많죠. 그래서 경험을 해도 '잘' 해야 합니다.

프로젝트 수행 기간의 문제

미소 : 최근 들어 프로젝트 수행 기간이 전체적으로 짧아지고

있습니다.

박 : 전체 프로젝트 기간 중 코딩·구현 기간은 확실히 짧아졌는데 오히려 계획, 요구사항 분석, 테스트 기간은 상대적으로 늘어났습니다. CBD의 예를 들면 초기 컴포넌트 구축시 아주 많은 시간과 인력이 필요합니다.

신 : 예전에 데이터베이스에 연결하기 위해 해야만 했던 일들 기억하시죠? 데이터베이스 연결을 하나 하려고 해도 소켓부터 짜야 했습니다. 요즘은 저수준 작업을 하지 않아도 되게 해주는 기술이 많이 나와 정말 편해졌지만 기반 라이브러리나 틀이 부족하던 시대에는 그것 자체를 만드는 것이 힘든 일이었습니다. 그래서 아무나 할 수 있는 일이 아니었죠. 그런데 지금은 아무나 하는 시대가 됐습니다. 소켓에 대해 몰라도 원하는 프로그램을 만드는 데 큰 문제가 없습니다. 하지만 말씀하신 것처럼 비즈니스 업무가 엄청나게 복잡해지면서 개발자가 고민해야 할 일이 더 늘어났다고 볼 수 있습니다.

윤남영(이하 윤) : 시간 맞추는 것은 정말 힘들니다. 일정이 지

켜지는 경우가 거의 없죠. 대부분의 경우 지연됩니다.

박 : 프로젝트 일정을 '정확히' 산정하는 것은 쉽지 않은 일입니다. 거의 불가능하죠. 그보다는 주·월 단위로 프로젝트 진행을 확인하면서 시간이 '새는' 것에 어떻게 대처할 것인지 대안을 세우는 것이 더 중요합니다. 한 곳에서만 병목 현상이 생겨도 프로젝트 전체가 지연됩니다. 일정이 늦어지면 개발자는 좀 솔직해질 필요가 있습니다. 어디에 문제가 있는지 동료나 프로젝트 관리자와 상의해야 합니다. 프로젝트 관리자 역시 통찰력이 필요합니다. 어떤 팀원에게 문제가 있는지, 어느 부분이 진행이 더딘지 등 항상 주시해야 합니다. 문제는 전반적으로 그런 소통이 부족하다는 거죠.

신 : 예전에는 일이 밀린 사람 일까지 받아서 했던 적이 있습니다. 하나둘씩 말다 보던 게 오�히려 힘들어지더군요. (웃음)

프로젝트는 함께 하는 것

신 : 2000년 전에 했던 프로젝트는 지연되는 경우가 많았습니다. 2000년 이후로는 지연되는 경우가 줄었습니다. 2000년 전후의 차이가 있다면 '값'의 책임을 명시했다는 것입니다. 프로젝트는 '함께 하는' 것입니다. 흔히 '90% 정도 진행됐다'고 말하는 것은 프로젝트가 지연되고 있다는 뜻입니다. 고객에게 최대한 투명하게 보이려고 했고 그를 통해 신뢰를 얻을 수 있었습니다.

박 : 역할에 대한 인식이 불분명한 것이 문제입니다. 대부분의 고객이 개발자를 파트너로 생각하지 않죠. 일을 맡은 개발팀에서는 위촉될 수밖에 없고 약속환이 계속됩니다. '계명'은 많지만 '실행 세칙'은 부족하죠. 계명을 잘 지킬 수 있는 상황에 맞는 실행 세칙을 만들어 나가야 합니다.

요구분석과 품질 유지는 어렵다

미소 : 개발자들이 가장 싫어하는 것 중 하나가 디버깅, 테스트, 문서화라고 하던데요. 전체 업무 중 이 세 가지 비율이 높은 것에 대해 어떻게 생각하시나요?



신승근(데브)



윤남영(다음기술)



박지순(컴포넌트비전)

윤 : 전 문서를 먼저 만들고, 테스트도 자주 하는 편인데 주변의 다른 개발자들은 문서 작업이나 테스트를 정말 싫어합니다.

신 : 사실 개발자가 테스트하면 다 잘 됩니다. (웃음) 개발자가 테스트해야 하는 것은 개별 함수 등이 정확하게 동작하는지 확인하는 것으로 한정해야 하고 고객이 요구하고 필요로 하는 기능 테스트는 전담자가 별도로 수행해야 합니다. 사실 '골인을 제대로 시키지' 못하는 개발자가 많습니다.

윤 : 자신을 기준으로 판단하는 것도 위험합니다. 실제 있었던 일인데, 한 개발자가 자바 1.4 기준으로 어떤 프로그램을 만든 적이 있습니다. 지금도 그렇지만 그 때 대부분 자바 1.3 환경을 쓰는 상황이었는데 자바 1.4에 있는 기능을 많이 이용해 프로그램을 작성해서 약간 문제가 된 적이 있습니다.

신 : 역시 요구분석이 중요하죠.

윤 : 때론 싸움이기도 하구요.

신 : 무엇보다 개발자에게는 언제든지 새로운 것을 배우려는 자세가 있어야 한다는 걸 새삼 느낍니다. 이 일을 하는 이유, 보람 중 하나는 많은 것을 배울 수 있다는 것이죠.

이미 복잡해질 대로 복잡해진 기술과 변화무쌍한 환경이란 요소를 끌어안고 때론 적인지 동지인지 헷갈리기까지 하는 사람들과 함께 일한다는 것은 말 그대로 직접 느껴보기 전에는 그 어려움을 모르는 일일 것이다. 뒤에

소개될 그레디 부치 인터뷰에서 정확히 지적되지만 소프트웨어 개발이 근본적으로 복잡할 수밖에 없는 이유는 그것이 인간 활동이기 때문일 것이다. 짧은 시간이었고 지면 관계상 모두 소개하지 못했지만 서로 다른 직급의 사람들(의도했던 것은 아니지만 좌담 참석자들은 CEO, 수석 컨설턴트, 평사원이었다)에게서 나온 공통된 문제의식은 기술보다는 인간의 문제라는 것이다. 그리고 진리는 단순했다. 열린 마음으로 배우고 실천하는 것! 그런 개발자들이 그레디 부치가 말한 '체상이 돌아가는 방식을 바꾸게 해준 소프트웨어를 창조'한 사람들이 될 수 있을 것이다. 송우일 기자