사용자가 있다는 것의 중요성

개발자는 사용자가 있는 것만으로도 좋은(wonderful) 일이지만, 비단 당신이 누군가의 필요를 충족시켜 주고 있으며 주어진 업무를 잘 하는것 뿐만 아니라 사용자들을 잘 유도한다면 그들은 co-developers가 될 수 있다.

사용자들을 co-developers로 여기는 것은 least-hassle 루트들을 빠르게 향상시키며 효율적으로 debugging을 할 수 있게 한다.

이는 Linus Torvalds가 보여주기 전 까진 저평가되고 있었다. 하지만 그의 성공으로 인해 재평가 되었다.

일찍, 자주 발표하라

일찍, 그리고 자주 발표하는 것은 리눅스 개발 모델의 중요한 부분이다. 초기 버전들은 버그가 많으며 당신 역시 사용자들의 인내심을 시험하고 싶진 않을 것이다. 그래서 cathedral-building 스타일의 개발 방식을 더 선호하게 되었다. 하지만 리눅스는 이와 정 반대의 방식을 지녔다.

리누스는 가장 효과적인 방식으로 그의 사용자들을 co-developers로 여긴 것 이였다.

일찍, 자주 발표하라. 그리고 고객의 소리를 들어라.

리누스의 혁신은 그가 개발하는 것의 복잡성에 비견될 것 만한 레벨로 만들었다는 점에 있다. 이는 리누스 토발즈가 engineering and implementation의 천재인 것에 기반한다. 리눅스의 전반적인 설계는 그의 버그와 개발의 dead-ends를 피하는 육감과 A to point B의 최소 노력 경로를 찾아내는 특성을 바탕으로 본질적이고 보수적이며 단순한 설계 방식이다. 그렇다면 그가 최대화 한 것은 무엇일까? 리누스는 해커/사용자들에게 지속적인 자극과 보상을 주었다. 자극은 그들의 자기만족의 전망이며, 보상은 하는 작업의 지속적인 향상이었다.

충분히 많은 beta-tester와 co-developer 가 주어진다면, 거의 모든 문제가 빨리 파악될 것이며, 누군가 분명히 고칠 것이다. 덜 형식적으로는 “충분히 많은 eyeballs가 주어진다면, 모든 버그들은 사라질 것이다.”로 표현되며 “Linus의 법칙”으로 부른다. 여기서 cathedral-builder와 bazaar의 핵심적인 차이가 있다. 프로그래밍에 대한 cathedral-builder의 관점으로 본다면, 버그는 어렵고 까다로우며 심오한 현상이다. 반면 bazaar의 관점으로는, 버그는 보통 쉽게 해결되는 현상으로 본다.

수 년 전, 사회학자들은 다수의 관찰자들의 의견은 그 중 무작위로 선택된 한 명의 관찰자의 의견보다 더 신뢰가 있다는 결과를 발표했다. 이는 사회학자들이 Delphi-effect라고 부르는 현상이다. 리누스가 보여준 이것은 심지어 운영체제를 디버깅하는 곳에도 사용할 수 있다는 것이다.