

오픈소스 커뮤니티 참여를 권하는 글

0224_개발자영어_번역에 대한 공동 피드백

rkJun님이 번역하신 것에 피드백 드리기

<http://www.facebook.com/groups/engfordev/326185810766584/>

[오픈소스 커뮤니티 참여를 권하는 글](#)

[0224 개발자영어 번역 피드백](#)

[번역 피드백 참여방법 \(](#)

[당신을 위한, 오픈 소스](#)

[\[1\] 나솔 째 - 완료](#)

[형편없는 당신의 업무, 프로그래밍 다시 즐겁게 하기](#)

[\[2\] 원강민 째 - 완료](#)

[협력을 통해 훌륭한 개발자 되기](#)

[\[3\] 나솔 째 - 완료](#)

[\[4\] 나솔 - 완료](#)

[현금의 취득](#)

[\[5\] - nassol 째 - 0301-0302 완료](#)

[\[6\] rkJun 째 - 0302 완료](#)

[\[7\] nassol 째 - 완료 0302](#)

[기술외의 경험 넓히기](#)

[\[8\] 나솔 - 완료 - 0225](#)

[\[9\] rkJun 째 완료 0305](#)

[좋은 사람이라고 느끼기](#)

[\[10\] rkJun - 완료 0229](#)

[지금 당장 시작하기](#)

[\[11\] 나솔 째 0302 - 완료](#)

[\[12\] nassol - 완료](#)

[\[13\] nassol - 완료](#)

번역 피드백 참여방법



박준태

이런 부탁을 드려도 될지 모르겠네요. 최근에 영문 블로그 글을 읽고 팬찮은 느낌이 들어 번역을 했는데 발번역이 좀 많이 섞여 있습니다. 염치 불구하고 시간 나실 때 검수해 주실 분 계실까요.

http://rkjun.wordpress.com/2012/02/13/open-source-for-you-korean_translate/



Open Source for You (한글번역) | rkJun's Blog
rkjun.wordpress.com

📖 좋아요 취소 · 댓글 달기 · 게시물 그만받기 · 공유하기 · 9분 전 경기도 과천 근처에서

👍 회원님이 좋아합니다.



김나솔 우와.. 멋지네용.. 저도 살펴볼게요 ^^

3분 전 · 좋아요

대구의 이력차네 ㅇ

번역본 주소: http://rkjun.wordpress.com/2012/02/13/open-source-for-you-korean_translate/

원문 주소 : <http://blog.jupo.org/2011/09/12/open-source-for-you/>

구글문서내에서 적당한 부분에다가 찜 표시 해주세요~

너무 부담스러운 양을 찜하지 마시고 대략 한 문단 정도? 찜해 주세요~

기본적으로는 한국어 부분을 보시고, 이렇게 하면 의미전달이 좀 더 자연스럽겠다..

하는 글을 옆에다가 써주시면 좋겠어요.

파란색으로 글자색을 표시해주시면, 박준태님이 보고 참고하시기 편하시겠죠?

그럼 ~

당신을 위한, 오픈 소스

제안) Open source for you - 오픈소스는 여러분을 위한 것.

[1] 나솔 찜 - 완료

나는 얼마전, 내가 근무중인 회사의 개발팀에서 오픈 소스 활동에 관한 생각을 얘기하는 프리젠테이션 토크를 하게 되었다. 사람들의 공통된 인식은 그리 복잡하지 않았다. 오픈 소스 커뮤니티는 개인적인 보상은 거의 얼마 안되고, 전적으로 애타주의적인 마음이 동기가 되는, 마치 거대한 사회적 히피집단 같다는 것이었다. 나는 이런 근거 없는 믿음을 깨뜨리고, 사람들 대부분이 자기 자신을 위한 동기와, 많은 개인적인 이익을 위해 오픈 소스 활동에 참여한다는 것을 설명하였다. 내가 초점을 맞춘 몇 가지 불편한 진실에도 불구하고, 토크는 잘 마무리되었다.

I recently gave a presentational talk to the development team at my work, on what it's like contributing to open source. A common perception by people not involved is that the open source community seems like one big socialist hippie commune, whose motivations are entirely altruistic with little reward for the individual. The approach I took was to dispel this myth and show that for many who participate, their motivations are mostly self serving, with many personal gains to be made. Despite some of the awkward truths I focused on, the talk was received really well.

(원문에서 해당부분을 붙여놓았어요.)

나는 얼마전 내가 일하는 회사의 개발팀을 대상으로 오픈 소스 활동에 대해 프리젠테이션을 하였다. 오픈소스에 대한 사람들의 공통된 인식은 대략 이러했다. 오픈 소스 커뮤니티에서 개인은 보상을 거의 받지 않고, 이타주의적인 마음만을 동기로 운영하는 거대한 사회적 히피집단 같다는 것이 오픈 소스 커뮤니티에 대한 공통된 인식이었다. 나는 이런 근거 없는 믿음을 깨뜨렸다. 그리고 오픈 소스 커뮤니티에서는 사람들이 자신을 위해, 그리고 자신의 이익을 위해 오픈소스 활동에 참여한다고 설명하였다. 내가 초점을 맞춘 부분에는 몇 가지 불편한 진실도 있기는 했지만, 개발팀은 발표내용을 잘 받아들였다.

형편없는 당신의 업무, 프로그래밍 다시 즐겁게 하기

[2] 원강민 째 - 완료

만약 당신이 어느 전형적인 회사의 9시~5시 정규직 소프트웨어 엔지니어라면, 여러 이유로 직업에 짜증을 느낄 가능성이 높다. 당신은 매일 똑같은 종류의 프로젝트에서 일을 해야 한다. 매일 똑같은 기술을 가지고 일을 해야 하고, 거의 대부분 시대에 뒤쳐진 기술을 사용한다. 현대 기술의 위에 더욱 세련된 기술이 존재하는데, 비전문적인 다양한 이유로, 업무에서 사용하는 것은 허락되어 있지 않다. 당신에게 마감시간도 정해져 있다. 마감시한은 최대한 빨리 기능을 쳐내도록 앞장 서고, 품질과의 직접적인 충돌에 직면하게 된다. 그래서 당신은 종종 강제로 추잡한 지름길을 택하게 된다, (개발자로서의) 자부심은 약간 버리고.

만약 이런 일이 당신의 유일한 프로그래밍 경험이라면, 금방 싫증나고, 모든 게 전혀 재미도 없을 것이다. 이와는 반대로, 프로그래밍은 진짜 믿을 수 없을 정도로 재미있게 할 수도 있다! 자, 상상해 보라. 당신의 회사가 선택가능한 다른 수백개의 프로젝트를 가지고 있고, 당신이 일로 하고 싶은 프로젝트를 선택할 수 있다고. 그것 말고도, 회사에서는 당신에게 이렇게 말한다. 당신이 원하는 기술을 사용할 수 있고, 당신이 그것을 정확하게 빌드할 때까지 필요한 시간만큼 사용할 수 있다고. – 당신이 작업한 최종 결과물이 매우 잘 설계되었고, 아름답게 작성된 소프트웨어이기만 한다면. 직업적인 프로그래머에게, 꿈같은 소리다. 하지만, 당신이 오픈 소스 프로젝트에서 일할 때는 그게 맞다. 당신의 문제 영역을 선택해라. 당신의 기술 스택을 선택해라. 당신의 페이스를 선택해라. 당신의 품질을 선택해라.

힘들고 짜증나는 업무(는 이제 그만), 프로그래밍을 다시 즐거운 놀이로 만드세요.

만약 당신이 전형적인 회사에서 9시부터 5시까지 일하는 정규직 소프트웨어 엔지니어라면, 여러가지 이유로 인해 업무에 짜증을 느끼고 있을 가능성이 높습니다. 당신은 항상 같은 형식의 프로젝트에 참여하고, 같은 기술을 이용합니다. 심지어는 오래된 기술이죠. 사실 그 기술보다 더 세련된 기술이 존재하지만, 회사에서는 아직 검증되지 않았다는 이유로 실무에 적용하는 것을 허락하지 않을 것입니다. 또한 일정의 마감시한도 지켜야 하구요. 마감시한은 최대한 빨리 기능들을 만들어 내도록 요구하기 때문에(그것은 작업물의 품질에 좋지 않은 영향을 줍니다.) 당신은 개발자로서의 자존심을 버리고 막코딩을 하게 됩니다.

만약 이런 일이 당신의 유일한 프로그래밍 경험이라면, 금방 싫증도 나고, 전혀 재미도 없을 것입니다. 하지만 정반대의 경우라면, 프로그래밍은 정말 굉장히 재밌을 수가 있겠죠! 이렇게 상상해 보세요. 당신의 회사에 수백 가지의 프로젝트가 있습니다. 그리고 당신은 그 중에 당신이 원하는 하나의 프로젝트를 선택할 수 있습니다. 게다가, 회사에서 당신이 원하는 기술을 사용할 수 있도록 허락하고, 최종 결과물이 충분히 멋지게 나올 때 까지 일정을

주겠다고 말합니다. 마치 슈퍼 개발자처럼요. 꿈처럼 들린다구요? 아니요. 오픈 소스 프로젝트에서 일할 때 만큼은 당신은 슈퍼 개발자입니다. 업무를 고르세요. 사용할 기술도 고르세요. 일정도 마음대로 짜세요. 제품의 품질까지도 말이죠.

협력을 통해 훌륭한 개발자 되기

[3] 나슬 찜 - 완료

더 좋은 개발자가 되기 위해서는 두가지 방법이 있다: 코드 쓰기와, 코드 읽기다. 당신의 개인 시스템안에서는, 이런 감각을 뚜렷하게 만들어 주지는 못한다. 오래전에 당신이 짠 코드를 읽는 것, 매년 자신만의 시스템에서 당신이 직접 코드를 작성하는 것, 당신 스스로 교육을 통해 공부하는 것에서 조금은 배우는 게 있다. 진정한 배움은 다른 개발자들과의 교류, 다른 개발자들이 리뷰할 수 있고, 피드백을 주는 코드를 작성하는 것, 그리고 당신보다 더 많은 경험과 다른 접근방식을 가진 다른 개발자들이 작성한 코드를 읽는 것이라 할 수 있다.

Become a Better Developer through Collaboration

There are two ways to become a better developer: writing code and reading code. This clearly **doesn't make sense**(이것은 말이 안된다.) in a silo on your own. There's little to learn from reading your own code beyond reminiscent value, and you could write code on your own for years and the only lessons learned would be those you teach yourself. Naturally the real learning is to be had from interacting with other developers, writing code that they can review and give you feedback on, and reading code written by other developers with more experience and different approaches than you have.

협업을 하면 더 훌륭한 개발자가 된다.

더 좋은 개발자가 되는 두 가지 방법이 있다. 하나는 코드를 작성하는 것이고, 다른 하나는 코드를 읽는 것이다. 혼자서만 개발을 하면 이 두 가지 방법을 쓸 수가 없다. 자기가 쓴 코드를 읽으면 흐뭇할지는 몰라도 별로 배우는 것은 없다. 그리고 혼자서 코드를 작성할 수는 있겠지만, 배우는 교훈이라고는 스스로 배우는 것 말고는 없을 것이다. 너무나 당연하게도 진짜로 배우는 것은 다른 개발자들과 상호작용하는 과정에서 나온다. 다른 개발자들이 검토해보고 피드백을 줄 수 있는 코드를 작성하면서 말이다. 그리고 경험이 더 많은 개발자가 쓴 코드를 읽으면서, 그리고 여러분과는 접근법을 가진 개발자가 쓴 코드를 읽으면서 진정한 뭔가를 배울 수 있다.

[4] 나솔 - 완료

다시 당신의 본업으로 되돌아와서, 당신은 얼마나 많은 개발자들과 직접적으로 협업하고 있는가? 조금 정도? 십여명 정도? 작은 집단에서도 당신은 많은 것을 배울 것이다. 그러나 여전히 상대적으로 제한적인 범위내에 있다. 오픈 소스 커뮤니티는 수천명의 개발자들이 만들어 내고, 오랜 세월을 넘어 수백만 코드를 제공하고 있다. 당신 직장의 외부로 공개된 적 없는, 한번도 노출된 적 없는 코드보다 더욱 더 좋은 품질을 가지고 있다.

Drawing back to your day job, how many developers do you collaborate directly with. A few? A dozen? A small pool that will give you much to learn from, but is still relatively limited in scope. The open source community is made up of thousands of developers, who over many years have produced millions of lines of code. Code of quality much greater than you'd ever be exposed to without stepping outside of your workplace.

당신이 일하는 회사에 대해 생각해보자. 당신이 직접 협업하는 개발자가 몇 명이나 되는가? 두세명? 열댓명? 작은 집단에서도 당신은 많은 것을 배울 수 있을 것이다. 그러나 배우는 범위 면에서는 아무래도 한계가 있을 것이다. 오픈 소스 커뮤니티에는 수천 명의 개발자들이 있다. 그리고 이들은 수년 간 수백만 줄의 코드를 작성했다. 이들은 여러분이 일하는 곳을 벗어나지 않고서는 접할 수 없는 품질의 코드를 작성한다.

현금의 취득

금전적 수입을 얻기

[5] - nassol 짬 - 0301-0302 완료

오픈 소스 개발에 대한 가장 큰 오해 중 하나는 돈벌이에 도움이 안되는 헛된 노력이라는 오해다. 어느 정도까지는 사실이지만, 실제로 금전적인 수익을 낼 수 있는, 직접적이고 간접적인 몇가지 방법이 있다.

One of the biggest misconceptions about open source development is that it's a financially fruitless labour. While true to a certain extent, there are actually several ways in which it can pay a monetary return, both indirectly and directly.

오픈 소스 개발에 대한 오해가 많이 있는데, 그 중 가장 심한 오해 중 하나는 오픈소스개발에 참여해서 돈을 벌 수 없다는 것이다. 어느 정도까지는 사실이지만, 실제로 금전적인 수익을 낼 수 있는 몇 가지 방법이 있다. 직접적인 방법도 있고, 간접적인 방법도 있다.

당신은 정규직으로 꾸준히 일할 수 있고, 아니면 고정급여를 받는 직원이 될 수도 있다. 또는 반대로 그들의 다음 식사가 언제지 모를 멋대로인 프리랜서와 사업가들도 있다. 사실 이견 틀렸다. 당신은 고용주처럼 당신 자체가 사업이다. 당신에게는 고객이 있고, 고용주가 있다. 때로 당신은 실제로 하나의 고객에게 열중하고 있을 때, 당신이 본업에 빠져있다고 느낄지도 모른다.

You may work a steady, full-time job, or you're looking for one, and think of yourself as an employee with a fixed salary, in total contrast to those wild freelancers and entrepreneurs who don't know where their next meal will come from. Wrong. You are a business, just like your employer is. You have one customer, your employer. Sometimes you might feel as though you're stuck in your day job, when in actual fact you're simply stuck with one client.

여러분은 고정된 직장을 가진 상태일 수도 있다. 그리고 자신이 고정수입을 갖는 월급쟁이라고 생각할 수도 있다. 다음 수입이 어디서 나올지 알 수 없는 프리랜서나 사업가와 는 다르다고 생각할지도 모른다. 미안하지만 틀린 생각이다. 여러분의 사장과 마찬가지로 여러분도 사업을 하는 것이나 다름없다. 여러분에게는 고객이 한 명 있다. 바로 여러분을 고용한 사장이다. 어떤 때에는 이 직장일에 얽매어 있다고 느낄 지도 모른다. 하지만 실은 한 명의 고객에게 얽매어 있는 셈이다.

QQ. stuck in your job..이 부분의 뉘앙스를 잘 모르겠네요. 묶여있다는.. 부정적인 뉘앙스를 넣은 건지...

[6] rkJun 째 - 0302 완료

그래서 어떻게 더 많은 고객을 얻을 수 있을까? 어떻게 더 좋은 고객을 얻을 수 있을까? 당신의 비즈니스 코스를 홍보해라. 오픈 소스는 당신의 개발자로서의 능력과 당신을 돋보이게 하는 대단한 방식을 제공해준다. 당신의 오픈 소스 기여는 다른 개발자로부터 당신의 이력, 즉 당신의 실무 경쟁력을 더욱 두드러지게 해준다.

그래픽 디자이너는 항상 그들의 포트폴리오를 호화롭게 보여줄 수 있고, 개발자들 또한 가능하다. 나는 고용 협상 테이블과 잠재적인 고용가능성 측면에서 모두, 누군가 오픈 소스 기여는 그들이 무엇을 하는지에 대한 뚜렷한 열정을 이미 입증시켜주는데 중요하다. 그러므로 그것을 보여줄 수 있는 어떤 것도 가지고 있지 않아도, 설득력 측면에서 중대한 이점을 가지고 있다.

So how can you gain more clients? How can you gain a better client? By promoting your business of course. Open source provides a great way for you to showcase you and your ability as a developer. Your open source contributions can distinguish your CV from other developers, namely, your business's competition. Graphic designers have always had the luxury of being able to display their portfolios, and developers can too. I've been on both sides of the hiring table, and when choosing between potential hires, someone with significant open source contributions has already demonstrated a clear passion for what they do, and therefore has a major advantage over those who are tasked with convincing people of the same, without having anything to show for it.

그럼 더 많은 고객을 만들고, 더 좋은 고객을 만들기 위해선 어떻게 해야 할까? 답은 여러분의 비즈니스 코스를 홍보하는 거다. 오픈 소스는 여러분의 개발자로서의 능력과 여러분을 더욱 돋보이게 해주는 멋진 방식을 제공해준다. 여러분이 오픈 소스에 참여하는 것은 다른 개발자들보다 여러분의 이력과 실무 경쟁력이 훨씬 더 우위에 있음을 나타내 준다. 그래픽 디자이너와 개발자들은 항상 화려한 포트폴리오를 보여줄 수 있다. 고용 가능성이 있고, 고용 협상에 대한 자리에서 오픈 소스에 참여한 경험은 이미 열정으로 가득하다는 걸 입증해 주고, 그러므로 그 자리에서 그러한 능력을 보여줄 필요도 없이, 다른 사람들보다 더욱 유리한 입장에 설 수 있다.

>> 제안 : 오픈소스 프로젝트에 참여하면 여러분은 개발자로서의 능력을 더욱 돋보이게 하는 기회를 얻을 수 있다.

[7] nassol 째 - 완료 0302

품질과 유용성의 측면을 위해 당신의 오픈 소스 프로젝트를 얻어서, 사람들이 실제로 그것을 운영 시스템에 반영했을 때, 그것 그 자체로 큰 성과이다. 돈을 벌면서 일할 수 있는 직접적인 기회가 그로부터 생기게 된다. 당신의 기업 사용자는 자신들의 사용 용도에 맞게 당신의 소프트웨어를 커스터마이징 해달라고 요청할 지도 모른다. 게다가 그들이 더 좋은 아이디어로 프로젝트에 기능을 추가하고 싶어하는 행운이 올지도 모른다. - 당신이 생각해볼 수 없었거나, 프로젝트의 다음 단계에서 하려고 했던 것들을. 그 소프트웨어를 개발한 사람보다 더 좋게 만들 수 있는 사람은 누구이겠는가? 나는 이런 일이 일어나는 여러가지 경우를 경험했고, 회사는 기능 개발을 지원해 주었으며, 어쨌든 내 오픈 소스를 자유롭게 개발할 수 있게 하는 통찰을 얻게 되었다.

Getting your open source projects to the level of quality and usefulness where people are actually building production systems with them is huge achievement in itself. Stemming from that are direct opportunities for paid work. Your corporate users may need your software customised for their use cases. You may even be lucky enough that they'd like features developed into the project that are great ideas - things you might not have thought of, and will take the project to the next level. Who better to develop these than the creator of the software themselves? I've had several instances where this has occurred, and companies have sponsored development on features that, had I had the insight to think of, would have developed freely on my own anyway.

오픈소스 프로젝트를 일정 품질에 준하게 만들고, 사람들이 이 프로젝트를 사용하여 실제로 무언가를 만든다면, 이 자체만으로도 큰 성과이다. 이로부터 돈을 받고 일할 수 있는 기회가 생겨난다. 오픈소스 프로젝트를 사용하는 기업 사용자의 경우에 기업에 맞게 소프트웨어를 커스터마이징해달라고 요청할 수도 있다. 또한 이들이 여러분이 생각지도 못했던 좋은 기능을 프로젝트에 추가해달라고 요청할지도 모른다. 이런 과정을 통해 여러분의 프로젝트는 한 단계 업그레이드 될 것이다. 이런 기능을 제일 잘 개발할 수 있는 사람은, 원저자가 아니고 누구이겠는가? 나는 이런 경험을 직접 몇 번 하였고, 몇몇 회사에서는 내가 생각했던 기능을 추가하는 것에 대해서 금전적인 지원을 해주었다. 지원을 해주지 않았더라도, 나는 그 기능을 어쨌든 개발했을 것이다.

기술외의 경험 넓히기

[8] 나솔 - 완료 - 0225

기술 이외의 경험을 폭넓게 하기

코딩은 소프트웨어 개발의 작은 부분중 하나이다. 프로젝트 관리, 커뮤니티 구축, 서식, 제품 개발 및 마케팅 등, 성공적인 제품의 출하 과정을 구축하는 것에는 많은 학문이 있다. 프리랜서와 1인 기업의 경우는 다른 영역안에서 살짝 발을 담그고 있는 이점을 얻고 있을지도 모르겠다. 전형적인 회사의 개발자는 매일 코딩하는 역할 외에 다른역할은 경험해 보지 못한다. Hands on coding is only one small aspect of software development. There are many other disciplines that make up the process of shipping a successful product. Project management, community building, **formal writing**(요게 의미하는 게 몰까요?), product development and marketing. While freelancers and one-man commercial studios may get the benefit of dipping their toes into these different areas, as a developer at a typical company your role often doesn't take you outside of the day to day coding.

코딩을 작성하는 것은 소프트웨어를 개발하는 전 과정에서 일부분에 불과하다. 성공적으로 제품을 배송하는 전 과정에 관한 많은 학문 분야들이 있다. 이런 학문에는 프로젝트 관리, 커뮤니티 구축, 격식을 갖춘 글쓰기, 제품 개발 및 마케팅이 있다. 프리랜서나 1인 기업의 경우에는 이런 학문 분야를 접할 기회가 있을지도 모른다. 그러나 일반 회사에서 일하는 개발자의 경우에는 매일 코딩 외에는 다른 분야를 접할 기회가 없다.

formal writing ? 서식.. 이라면 뭐의 서식일까요?

<http://www.facebook.com/groups/engfordev/326632067388625/>

- 

문현호 소프트웨어의 개발에 관련된 각종 사항이라던가(기획,테스트 등) 문제점,보안등의 요소와 그에 대한 방책이라던가 하는 식의 전반적인 것들을 통틀어 말하는게 아닐까 하네요
토요일 오후 6:06 · 좋아요 취소 · 1
- 

Tw Shim 아마 제안서, 개발계획서, 기술 블로그 같은 일을 말하는거 같아요.
저라면 "형식을 갖춘 글쓰기"로 번역하겠어요.
토요일 오후 6:13 · 좋아요 취소 · 1
- 

박범진 Official 한 표현 문구를 사용해서 글을 적어야한다는 의미정도
토요일 오후 6:57 · 좋아요 취소 · 1
- 

Dae-Seon Moon 형식에 맞춘 글입니다. 일반적으로 회사에서 고객들에게 보여주기 위한 공식적인 문서등에 사용되죠. 사내 직원열람용은 Semi-formal로 쓰여지는 경우가 많고요. Formal한 문서는 보통 회사내 카피라이터들이 작성하죠
토요일 오후 10:43 모바일에서 · 좋아요 취소 · 1
- 

김나솔 흠.. 그럼 official writing은 '공식 문서 작성' 정도로 표현하면 되려나요..
15시간 전 · 좋아요

오픈 소스 프로젝트의 경험은, 전문 외의 범위까지 폭넓게 스킬 업하게 해줄 것이다. 나는 나의 가장 인기있는 프로젝트에 내 시간을 많이 소비한다. 그렇지만 요즘 실제로 코딩에 소비하는 시간은 작은 부분에 불과하다. 다른 개발자들과의 기능관련 아이디어 논의, 이 기능들을 포함하느냐 빼느냐에 대한 조정, 신규 사용자를 위한 지원에 가장 많은 시간이 소비된다. 그것은 어려운 일이지만, 멋지고 의미있는 경험이다.

Running an open source project properly will allow you to skill up in broad range of non-technical areas. I spend a lot of my time working on my most popular project, however these days only a small portion of this time is actually spent coding. Most of the time is spent in discussion with other developers on their feature ideas, coordinating the inclusion or exclusion of these, and providing support for new users. It's hard work, but it's a great and relevant experience.

오픈 소스 프로젝트를 운영해보면, 기술(technology) 이외의 분야에 관한 스킬을 익힐 수 있을 것이다. 나는 내 프로젝트 중에서 가장 인기있는 프로젝트 관련 작업을 하는데 시간을 많이 쓴다. 이 시간 중에 코딩 작업에 쓰는 시간은 그 비중이 얼마 안된다. 시간 중 대부분은 다른 개발자들과 기능에 대한 아이디어를 나누고 토론하는데 쓴다. 어떤 기능을 넣을지, 뺄지를 의논하는데 쓴다. 이런 작업은 어렵기는 하지만 멋지고 의미있는 경험이다.

[9] rkJun 째 완료 0305

이러한 결과, 어떤 방식으로든 오픈 소스에 기여하고 싶어하는 많은 비전문적인 사람들은 그들이 가치있는 제안을 하는 것 외에 아무것도 가진게 없다고 믿는다. 모든 것이 연관되어 있어서, 대부분 이익이 되는 기여의 일부는, 비전문화된 특성에 있다.

만약 이것이 당신을 설명한다면, 망설이지 말고 뛰어들라. 당신의 지원은 당신이 상상하는 것보다 더 의미있다.

Corollary to this, many non-technical people who'd like to contribute to open source in some way, believe that they don't have anything worthwhile to offer. With all that's involved, some of the most beneficial contributions are those of a non-technical nature. If this describes you, then please don't hesitate to dive in. Your help means more than you can imagine.

이런 일들에서 알 수 있듯이, 오픈 소스에 참여하고 싶어하는 많은 사람들이 기술적인 분야와는 무관하므로 자신은 기여할 무언가가 없다고 생각한다. 하지만, 모든 것은 연관되어 있어서 기술과는 관련없는 부분이 가장 유익한 기여중 하나라 볼 수 있다. 만약 여러분도 그렇다면, 시작을 주저하지 말아야 한다. 여러분의 참여는 상상이상으로 큰 의미가 있다.

좋은 사람이라고 느끼기

[10] rkJun - 완료 0229

오픈 소스에 기여하는 가장 뚜렷하고, 여전히 적어도 가장 확실한 이점은, 애타주의와 다른사람들에게 기여하는 것에 둘러쌓여 있는 것이라고 먼저 언급해 둔다. 내가 언급했 듯이, 작지 않은 오픈 소스 프로젝트를 관리하는 것은 힘든 일일 수 있다. 높고 낮음이 있고, 때로 끝이 없는 것처럼 느껴진다. 그러나 가끔, 나는 칭찬과 감사의 개인적 메일을 받을 것이다. 비누박스로 관을 만들 수 있는 숨은 동기가 있는 어떤 공개포럼도 아닌 (역주:번역이 매끄럽지 못해 원문을 첨부: **Not in any public forum where the soapbox can be a conduit for ulterior motives**), 개인적이고 진심어린 감사의 표현을 담은.

내가 직접 그것을 경험해 볼 때까지, 나는 그것이 특별히 보람있다고 생각한 적이 없었다. 그것은 실제로 진정한 긍지와 만족감을 채워주고, 궁극에는 모든 것을 가치있게 만들어 준다.

The most obvious yet least tangible benefit in contributing to open source, is the idea I first alluded to around altruism and giving to others. As I mentioned, coordinating a non-trivial open source project can be hard work. It has its highs and lows, and sometimes it feels like there's more of the latter. However every now and then, I'll get a personal email of praise and gratitude. Not in any public forum where the soapbox can be a conduit for ulterior motives, just a private and genuine expression of thanks. I would

never have thought of this as being particularly rewarding, until I actually experienced it. It really fills you with a true sense of pride and satisfaction, which makes it all worth while in the end.

오픈 소스 참여의 가장 확실한 장점은, 애타주의로 타인에게 도움을 주는 거라고 생각한다. 전에도 얘기했지만, 작지 않은 오픈 소스 프로젝트를 관리하는 일은 고생을 동반하게 된다. 오픈소스에 참여하다보면 좋을 때도 있고 나쁠 때도 있다. (김나솔님 - 요기에 보면, 좋을 때와 나쁠 때... 라는 의미가 있다고 나오는데요 :) 오픈 소스에 참여하다보면, 좋을 때도 있고 힘들 때도 있다.. 정도면 어떨까요?

<http://dictionary.reverso.net/english-cobuild/highs%20and%20lows>)

때로는 끝이 없는 일처럼 느껴지기도 한다. 하지만 가끔, 나는 어떤 동기를 가진 공개 모임에서가 아닌, 순수한 진심을 담은 개인적인 감사 메일을 받기도 한다. (심민규님 - soapbox가 숨겨진 동기를 위한 채널(통로)이 될 수 있는"으로 해석될 수 있을 것 같습니다. soapbox 해석이 좀 어려운데요;;; 원래는 비누박스라는 뜻인데 연설자가 올라가는 단상을 의미하기도 하네요. 누군가 비누박스에 올라가서 연설이라도 한건가요;; 암튼 제 생각에, 아무나 어떤 동기를 가지고 글을 올릴 수 있는 퍼블릭 포럼에서가 아니라 순수하게 개인적인 의미의 감사를 받는다는 의미인 것 같습니다.)

직접 해보기 전까지는 그 일이 딱히 의미가 있다고 생각해본 적은 없었다.

내가 실제로 직접 경험해 보고 나니, 진정한 금지와 만족감을 채워주고 궁극적으로 모든 것을 가치있게 만들어준다는 것을 알게 되었다.

지금 당장 시작하기

[11] 나솔 짬 0302 - 완료

중국의 철학자 노자는 말했다:

천리의 여행도 한 걸음에서 시작된다.

(千里之行 始於足下)

시작은 힘겨운 업무처럼 느껴질 수도 있다. 작은 것부터 시작해라. 당신의 오픈 소스 아이디어는 아무도 사용하지 않을거라고 당신의 귀로 속삭이는 내면의 목소리를 무시해라. 설사 그게 사실이라 해도, 그것은 중요한 일이 아니다. 어쨌든 그것을 해라. 내 프로젝트 중 일부는 오직 몇십줄의 코드로 작성되어 있다. 사람들은 내가 상상해 본적 없는 방식으로 그것들을 유용하게 하는 방법을 찾아냈다.

The Chinese philosopher [Lao Tzu](#) said:

A journey of a thousand miles begins with a single step

Getting started might seem like a daunting task. Start small. Ignore the inner voice that whispers in your ear, arguing that your open source idea wouldn't be of any use to anyone. That might even be true, but it doesn't matter. Do it anyway. A lot of my projects are only a few dozen lines of code. People have found them useful in ways I couldn't have imagined.

중국철학자 노자는 이렇게 말했다:

천릿 길도 한 걸음에서 시작된다.

(千里之行 始於足下)

시작하려고 하는데 겁이 날 수도 있다. 작은 것부터 시작하자. 여러분의 오픈소스는 아무에게도 쓸모 없을 것이라고 내면에서 속삭이는 목소리를 무시하자. 그럴 수도 있다. 하지만 그래도 상관없다. 그래도 하자. 내 프로젝트 중 많은 것은 몇 십줄 짜리 코드였다. 그런데 사람들은 내가 상상하지도 않았던 곳에 그 소스를 유용하게 사용하였다.

People have found them useful.. 사람들은 내 프로젝트가 유용하다고 생각하였다. .. 내가 상상하지도 못했던 방식으로... => 약간 표현을 바꿔봤네요.

[12] nassol - 완료

아이디어는 구하기 힘들다. 대부분 나의 아이디어는 내 가려운 곳을 긁어주었다. 내가 흥미를 가지고 있는 기술 영역과, 정기적으로 해야 하는 일상적인 프로세스의 결과물의 격차는 그것들이 자동화하고 개발할 수 있었다는 것을 깨닫게 해준다.

Ideas can be hard to come by. Most of mine were itches I wanted to scratch. **Product gaps in the technology stacks I'm interested in**, or mundane processes I went through regularly, upon realising they could be automated and productised.

아이디어는 잘 오지 않는다. 내 아이디어중에 대부분은 내가 아쉬워했던 것들이었다. 예를 들어 내가 관심있는 기술 영역에서 제품이 개발되어 있지 않는 경우에 내가 개발할 수 있겠다는 생각에서 개발 아이디어를 얻었다. 또는 내가 일상적으로 하는 루틴한 것들을 자동화하거나 개발할 수 있겠다는 지점에서 아이디어를 얻었다.

<http://www.facebook.com/groups/engfordev/328414493877049/>

[13] nassol - 완료

아마 당신은 오픈 소스 소프트웨어의 사용자이고, 그 소프트웨어는 완벽하지 않을 수도 있다. 당신이 생각한 것과는 다르게 동작하는 기능도 있고, 어떤 경우는 문서가 없어서 그걸 찾기 위해 고생을 해야 한다. 요즘에는 무료로 사용중인 소프트웨어에 대해서는 그냥 불평만 하고 두 손 들고 포기해 버리는데, 그것은 정확히 당신이 원하는 일이 아니다. 그렇게 하지 않고, 그것에 관해 할 수 있는 어떤 일을 하겠다고 선택할 수 있다. 소스코드 안으로 뛰어 들어가, 무언가 기여하고 돌아오라. 스스로 향상시키고, 무언가의 일부가 되게 하라.

Perhaps you're an end user of open source software, and it isn't perfect. A feature works differently than how you think it should, or something is undocumented and you struggled to find it. These days it seems that the **modus operandi** is to throw up your arms, and complain as loudly as possible about how the software you're using for free, isn't exactly the way you want it to be. Don't. You have the choice to do something about it. Dive into the source code. Contribute something back. Improve yourself, and be a part of something.

여러분이 어떤 오픈소스 소프트웨어를 사용하는데, 소프트웨어가 완벽하지 않을 수도 있다. 여러분이 생각하는 것과는 다른 방식으로 소프트웨어가 작동할 수도 있고, 또 어떤 부분은 도움말에 나와 있지가 없어서 찾기 위해 고생해야 할지도 모른다. 요즘은 이런 일이 있을 때, 무려(!) 공짜로 사용하고 있는 소프트웨어에 대해서 원하는 대로 작동하지 않는다고 불평만 하고 두 손 들고 포기해 버리는게 기본 반응인 듯 하다. 그러지 말자. 그 문제에 대해 여러분이 할 수 있는 게 있다. 소스 코드를 살펴보자. 뭔가 손을 빌려주자. 자신의 능력도 향상시키고, 뭔가를 만들어가는 과정의 일부가 되자.