

This is a **low resolution, black and white** version of the article you downloaded. To download the whole of Free Software Magazine in *high* resolution and color, please subscribe!

Subscriptions are free, and every subscriber receives our fantastic weekly newsletters — which are in fact fully edited articles about free software.

Please click here to subscribe:

<http://www.freesoftwaremagazine.com/subscribe>

# How to get people to work for free

## Attracting volunteers to your free software project

David Horton



As time marches on and our lives become more complicated, it seems we have less and less time to devote to that free software project we started back in our idealistic youth. Rather than abandoning a good project due to lack of time, consider seeking out the assistance of other members of the free software community. With a few simple steps you can make it easy to find volunteers to help you complete your project.

### A roadmap to finding volunteers

You need to start with a solid understanding of your own project before you can expect other people to help you with it. Have you thought about where you want your project to be one year from now? Think about it. Now write it down. Once you know the direction you want your project to go, you can start communicating the big picture to other people. As more people begin to understand your project's ultimate destination it's easier for some of them to become interested in helping you get it there. You may start receiving emails asking, "How can I help?" and when people offer to help you need to come up with a better response than, "I don't know, what can you do?" Be prepared to reply with specific tasks that can be worked on and completed in a reasonable amount of time. You should also make sure you communicate the benefits the volunteers can expect to get from contributing to your project. It's probably not money, but there are things of value that people can gain as free software volunteers.

If this seems like a lot of information to digest, don't worry, this article will cover each of these topics in greater detail. By the time you finish reading you should have some pretty good ideas of how you can make your free software project more attractive to all of that untapped volunteer talent out there.

### Communicating your project's vision

You probably understand your project's vision better than anyone else, after all it's your project and you designed it. But what about everyone else? Can the average free software user look at your project and think, "I know what this project is about and where it wants to be a year from now"? Chances are you've gotten so wrapped up in writing code and releasing patches that you forgot about communicating your project's big picture view. If people don't know where the project is going they won't know how to help it get there. Your project needs a vision.

Now, if you are saying to yourself "I'm a coder, not a management guru", and wondering how to tackle this vision stuff, don't worry. Start by looking at some of the popular free software projects on the internet to get some ideas. Most of them will have an "about" section on their web site that communicates the big picture view in a mission statement. Take OpenOffice.org as an example. Their mission is "To create, as a community, the leading international office suite that will run on all major platforms and provide access to all functionality and data through open-component based

APIs and an XML-based file format”. That one sentence sums up the goals of the entire project.

Your project may not be as monumental as OpenOffice.org, but you can still have a mission statement. Keep it short and to the point and remember that you’re not describing the state of your project as it is today, but rather where it is going to be when all the work is finished.

Say, for example, that you are working on a killer free-software recipe management system. Your project already has a nice looking web browser interface and a really powerful database back-end. But, it would be a lot better if it could read recipe files from other, proprietary recipe management software packages. The vision for this project might be summed up as “To build a powerful, free, web-based recipe management system that is able to import files from the popular proprietary recipe management programs”. Now that wasn’t too difficult was it?

## Identifying goals and tasks

Creating a vision for your project is similar to deciding where to go on vacation. You might know that you want to end up on a sunny beach with a cool drink in your hand, but you still have to figure out how you’re going to get there. Do you fly or drive? If you drive, where will you stop for lunch? Do you need to book a hotel? Free software projects have similar questions that need to be addressed. To answer these questions you need to set some goals.

Start by breaking your project into its major components. For example, if your project’s vision is “to build a free, web-based recipe management system that is able to import files from the popular proprietary recipe management programs”, you could set goals as follows:

- Create an easy to understand user interface with HTML/PHP
- Build an efficient database back-end
- Write code to import recipe files from other programs

If you’ve already put some work into the project there may only be a few items that need attention. These items can be identified and recorded as specific tasks. Suppose that you are happy with the look and feel of the browser interface for your recipe manager, but it’s marked up with HTML 3.2 and really should be updated to XHTML. So the only thing

preventing you from completing the goal of creating an easy to understand user interface is the fact that your HTML is outdated. Congratulations, you have just identified a task. Record this task and continue looking at the other goals to identify more tasks. Another goal you’ve stated is to be able to import recipe files from other programs. This can comprise several tasks such as the following:

- Write code to convert Meal Master file format into native file format
- Write code to convert AccuChef file format into native file format
- Write code to convert RecipeBook-XML into native file format

Continue the process of examining goals and picking out tasks until you have identified all of the significant tasks for the project.

## Creating job descriptions

Now that you have identified a number of tasks that need to be completed it’s time to find someone to help you work on them. Think about each task for a moment. Is it a one-time thing or is it on-going? How long will it take to complete the task? Can it be accomplished by one person or will it take several? Answering these questions will help form the basis of the job description for a person who would perform the task. Posting these job descriptions on your project’s web site will help you recruit the right people to get the task finished.

Several major free software projects have a “tasks” or “to-do” section on their web site that can be used to get an idea of how to structure a job description. The format is largely a matter of preference, but be sure to include basic who, what, where, when, why and how information in the job description. Who is the type of person to successfully complete this task? What exactly are they working on? Where do they submit the finished work? When does it need to be finished? How should they go about working on this? Including this information ensures that the volunteer working on your project knows what is expected to get the task finished.

Take the example of updating the HTML 3.2 mark-up to XHTML. You could do this yourself, because you consider

yourself to be an HTML guru, but unfortunately you just don't have the time. So you need to find another HTML guru to help you. Congratulations you've just identified the "who" portion of your job description. You need "an XHTML Guru". Continuing on with the what, where, and how you might end up with the following simple job description:

XHTML Guru needed to update HTML 3.2 mark-up for a web-based recipe management system. Approximately 300 lines of mark-up need updating. Use of vi editor is a must. Please contact [admin@free-recipe-project.com](mailto:admin@free-recipe-project.com) if interested.

Now if you are paying attention you may have noticed that the question of "why?" was skipped. That's because "why?" is often the hardest question to answer. There are only twenty-four hours in a day and most people seem to need about twenty-five. If time is so scarce why should people give it to you for free? Think about that when you answer the "why?" question.

## Enticing others to join your vision

There are many reasons that people volunteer to work on free software projects. Some people like the challenge or want to build their skills, others feel obligated to give something back to the community and some simply want to see their name associated with a great free software project. When asking people to volunteer their time to your project, make sure they know that they will get something back in return for their efforts. It is easier to describe concrete benefits so concentrate on them rather than the abstract ideas, like giving something back to the community. Make the benefits of volunteering part of the job descriptions.

Take another look at the job description that was created for XHTML Guru and think about the "why?" question. Why would someone want to volunteer to update HTML 3.2 to XHTML? It doesn't sound particularly glamorous. But, what about someone who is a student enrolled in a web development class? This type of volunteer job might sound like a good opportunity for a class project or as a resume builder. Thinking about things from the volunteer's perspective can help you make your job descriptions more attractive. Take a look at the XHTML Guru job description with this new information added.

Looking to gain some experience as a web designer? The Free Recipe Project is looking for an XHTML Guru to update HTML 3.2 mark-up for a web-based recipe management system. Approximately 300 lines of mark-up need updating. Use of vi editor is a must. Please contact [admin@free-recipe-project.com](mailto:admin@free-recipe-project.com) if interested.

## Putting it into practice

Now that you've had a crash course in recruiting volunteers take some time to apply this information to your own free software project. Take a break from coding and think about the big picture for a while. Where do you want your project to be in a year from now? What are the major pieces of the project. What specific tasks need to be addressed so these major pieces can be completed. Who has the skills that are required to get these tasks done? How will you show appreciation to the people who help you? Now record this information where everyone can see it. If your project has a web site use it as a volunteer recruiting and recognition tool. Add your project's vision to the top of the main web page. Create a "help wanted" page to list job descriptions for tasks that need to be finished. And make sure you create a "thank you" page recognizing all of the people who have volunteered their time to help you advance the project.

## Copyright information

© 2005 David Horton

This article is made available under the "Attribution-Share-alike" Creative Commons License 2.0 available from <http://creativecommons.org/licenses/by-sa/2.0/>.

### About the author

David Horton got started with GNU/Linux in 1996 when he needed a way to share a single dial-up internet connection with his college room-mates. He found the solution he needed with an early version of Slackware and a copy of the PPP-HOWTO from The Linux Documentation Project. Almost ten years later he is older and wiser and still hooked on GNU/Linux. He is also the author of The Pocket Linux Guide available from The Linux Documentation Project and his own website ([www.happy-monkey.net](http://www.happy-monkey.net)).