www.npfi.org

The Non-Profit FOSS Institute Triad Roadmaps

Project Roadmap for Software Firms

The Non-Profit FOSS Institute (NPFI) assists college-level instructors to develop a prototype software tool with their students for a local non-profit. The resulting free and open source software artifact must then be deployed and maintained by a stable software firm with a local presence to the nonprofit. This roadmap lists the steps a software development firm can take to be involved in a NPFI project.

PHASE 1, 2, AND 3: PLANNING, REQUIREMENTS, AND DEVELOPMENT	
Step 1: **Assess your interests in participating in a NPFI project**	• Often a software firm will enter a NPFI triad after being approached by NPFI, a class instructor, or a non-profit itself. The software firm's role in the triad is to provide deployment and long-term maintenance services to a non-profit client – taking over a software prototype developed by a college-level instructor and class. • Assess your willingness and staff capabilities to engage with the instructor and class as they develop the prototype software for the non-profit. This includes reviewing the project plan and requirements document and the work done by the class so that you know what will likely need to be done to prepare the prototype for deployment, and the platform that will need to be maintained. • Determine your fee structure for this type of project. Your client is a humanitarian non-profit that will be able to pay a small fee for prototype deployment, and then a small annual fee for ongoing maintenance. While no firm will be asked to do this work pro bono, a discounted rate will be necessary – often, the non-profit would not be doing the project at all without the support of NPFI and the college class.
Step 2: **Engage with class and non-profit as they build the software prototype**	• Different firms may enter the NPFI triad at different times depending on when it is recruited and contracted with. If engaged during the project development phase, you may review the requirements document, observe class interactions with the client, and review early code committals to learn about the software being built. • Student-developed software must use a GPL-type open source license for the product; either the instructor and students or the institution (depending on its intellectual property policy) will hold copyright to the software. If your firm further modifies the software, you may add your information to the original copyright statement, but you must agree to use open source licensing (GPL-like) for the product, and must commit to deploying the software and depositing all code and documentation in a publicly-accessible repository.
Step 3: **Receive the prototype**	• At the end of the semester, the class will present the completed prototype software to the non-profit and to your firm. This formal presentation will demonstrate the class' work, and it will provide closure for the class as well.

www.npfi.org

The Non-Profit FOSS Institute
Triad Roadmaps

PHASE 4, 5, AND 6: DEPLOYMENT, TRAINING, AND SUPPORT	
Step 4: Deployment	• Once you receive the functioning prototype, work with the Instructor to understand how it works and where artifacts and documentation are housed. • Prepare the software for deployment. This may include refining code (fixing bugs), extensive testing, adding a security layer, extracting and loading live data provided by the non-profit, and installing the software in the non-profit's live environment. • Work with the nonprofit to teach them how to administer and maintain the software, and support their development of a training plan to roll out the software to its staff and volunteers. • Once you fully understand the software and its architecture, collaborate with NPFI and the non-profit to finalize a hosting and support plan.
Step 5: Long-term support	• Provide long-term support for the software throughout its useful life – including patches and upgrades as needed to the dependent software, fixing bugs as needed, and checking on back-up processes. • While non-profit will be primarily responsible for administration, be prepared to provide some level of user support if problems are encountered. • Additional software development activities (additional development of the software) can be negotiated independently.