

# The Non-Profit FOSS Institute Triad Roadmaps

## Project Roadmap for Non-Profits

The Non-Profit FOSS Institute (NPFI) assists in matching a humanitarian non-profit with a college-level instructor and software firm. The non-profit is the client in an NPFI triad. It brings to the triad: a software need, a staff member willing and able to work with a college-level class over 1-2 semester(s) as it develops prototype software to meet the non-profit's need, a small level of funding to support software deployment and long-term maintenance by a software firm, and the commitment to training its own staff to use the software once deployed to meet its mission needs. This roadmap lists the steps a non-profit takes for project formation, execution, and completion.

### PHASE 1 AND 2: PLANNING AND REQUIREMENTS

<p><b>Step 1:</b> <b>Assess your needs for software and your interest in working with a college class</b></p>	<ul style="list-style-type: none"> <li>• Review your business processes and tools in areas such as donor management, volunteer scheduling, and donation/resource management (e.g., inventory management, resource availability).</li> <li>• Determine if you have a software need that is a good candidate for an NPFI project, and if your staff and volunteers will be receptive to a new electronic tool. Examples: donor database system, volunteer scheduling system, donated goods database system, resource allocation system (e.g., assigning guests to rooms, goods to donation sites). NPFI.org provides links to past and current projects that provide concrete ideas.</li> <li>• Determine whether you have a staff member that would enjoy working with a college-level computer science instructor and class to build prototype software to meet your software gaps or needs. This staff member will be responsible for keeping the non-profit team informed of progress as the software is developed.</li> <li>• Assess your ability to pay a software firm a fee to finalize and deploy the prototype software developed by the class, and an ongoing annual fee for maintenance.</li> </ul>
<p><b>Step 2:</b> <b>Work with college-level instructor to identify your software requirements and develop a project plan</b></p>	<ul style="list-style-type: none"> <li>• Work with NPFI to be matched with a college-level instructor and class that has the capabilities to meet your needs.</li> <li>• Work with college-level instructor to understand what kind of software tool is realistic to expect from the class, given your needs and the class' capabilities.</li> <li>• Describe your current work processes that the identified software tool will address, and provide examples of your current tools (forms, calendars, reports)</li> <li>• Review requirements document and project plan developed by the instructor so that you clearly understand the scope of the prototype that the class will develop. Project requirements must map realistically to student capabilities and time constraints, so that the students are highly likely to succeed, while still providing your organization with a product with value.</li> </ul>

# The Non-Profit FOSS Institute Triad Roadmaps

<p><b>Step 3A: Work with NPFI to support search for software firm to form third part of triad</b></p>	<ul style="list-style-type: none"> <li>• Once a project is identified, NPFI will lead the effort to identify a software firm to form the third part of the triad, unless you and/or the Instructor you are working with has already identified a technically qualified partner.</li> <li>• While this step may occur later in the process, identifying the software firm early allows its earlier involvement in understanding the project and its goals.</li> </ul>
<p><b>PHASE 3: DEVELOPMENT</b></p>	
<p><b>Step 5: Collaborate with the class building the software</b></p>	<ul style="list-style-type: none"> <li>• During the semester, meet with the Instructor on a regular basis (face-to-face or via web conferencing) to review progress and iterate requirements as the class builds a working prototype. Your designated staff member must be willing to actively engage with the class on a weekly basis and provide constructive feedback as the work unfolds.</li> <li>• Accept the software license terms: The student-developed prototype software will use an open source license for the product (NPFI can explain what this means); either the instructor and students or the institution (depending on its intellectual property policy) will hold copyright to the software.</li> </ul>
<p><b>Step 6: Select candidate software firms that can take over and support the software</b></p>	<ul style="list-style-type: none"> <li>• If not already completed in Step 3A above, work with NPFI to identify local information technology/software firms with the capability to provide long-term support for the software being developed. The firm will join you and the instructor/class as the third part of the project's triad. As appropriate, NPFI can introduce you to candidate software firms, invite proposals for software support, and help you select the best firm for your needs.</li> </ul>
<p><b>Step 7: Receive the prototype</b></p>	<ul style="list-style-type: none"> <li>• At the end of the class' work, it will present the completed prototype software to you and the software firm. This formal presentation will demonstrate the work done by the class, and celebrate its success.</li> </ul>
<p><b>PHASE 4: DEPLOYMENT</b></p>	
<p><b>Step 8: Deployment and staff training</b></p>	<ul style="list-style-type: none"> <li>• Once the software firm receives the functioning prototype, the firm finalizes the software and installs it for use in your environment.</li> <li>• Provide "live" data and test the software, work with NPFI and the software firm to develop a roll-out and training plan to introduce the software to its staff and volunteers, and finalize a support plan with the IT firm.</li> </ul>
<p><b>Step 9: Long-term support</b></p>	<ul style="list-style-type: none"> <li>• For a mutually-agreed-upon fee, the IT firm provides long-term support for the software throughout its useful life. This includes doing software upgrades and patches, fixing bugs, and doing back-ups. Additional software development work may be negotiated independently.</li> </ul>