Developing community: MagicBox open source community implementation plan

Justin W. Flory

Table of contents

Preface

Open source is a popular development strategy and yields unique benefits from a traditional software project; it also promotes a positive image of transparency. MagicBox is in an early phase of an open source project. The suggested roadmap assists in building a community around the project. At the end of this roadmap, MagicBox and other related projects are primed to bring in and sustain project participants from outside UNICEF. With time, patience, and persistence, a community form around the project.

This plan builds off of the sixth Digital Development principle of the UNICEF Office of Innovation, specifically on supporting developer communities.

The tasks below are suggested as secondary tasks to actual software development and can be implemented over seven to eight months.

About open source

Open source means public source code, but it also means a public bug tracker, design documents, and a developer discussion forum. Open source projects have sets of best practices and unique workplace culture from traditional proprietary projects. Jamie Zawinski, a founding developer of the Firefox browser said, “Open source does work, but it is not a panacea. […] you can't take a dying project, sprinkle it with the magic pixie dust of "open source," and have everything magically work out. Software is hard. The issues aren't that simple.”

Committing to open source culture translates the vision of a project to the community. Adopting an open development practices encourages a community to form. This is seen in other humanitarian projects like Ushahidi, an information-crowdsourcing tool with over 120,000 deployments and contributors across multiple continents. Projects like Ushahidi effectively leverage a group of contributors to benefit humanitarian open source software by making their vision public. This makes the project a medium to support that vision.

Adopting an open source-friendly culture allows MagicBox to better engage with the wider open source community. Building the foundations for community participation allow a small development team to grow and include wider perspectives into the software. Mozilla, the developers of the Firefox browser, enhance their product with contributions from non-Mozilla employees. In addition to code or project contributions, these participants bring diverse knowledge, expertise, and skills to a project. Ultimately, an open source-friendly culture encourages a developer community to grow and participate with the project.

These milestones follow best practices as understood by other successful open source communities with a compatible development workflow with UNICEF. Each milestone in the roadmap commits the project one step further to positive open source engagement.

Roadmap

What we've done so far: Open interaction, communication, culture

Since January 2018, we focused on practicing an open source-compatible workflow with new tools and services. Effective use of these tools as a safety and standard-setter is important for both core developers and new contributors.

In the beginning, we triaged public issues and tickets on important projects, established a transparent project management technique, adopted a pull request development model, and shifted developer communication to a public forum.

Evaluated current issues on important projects

This organized our development focus for the current state of the project. Current project focus is more clear when issues and pull requests are triaged or updated. This helps a new contributor understand the priorities and needs of the project today.

- Created and configured a GitHub project board for business process management
- Closed outdated / irrelevant tickets
- Added comments with brief updates for pre-existing but still relevant issues
- Added new issues for new requirements or known bugs
- Added tags to sort tickets by category and priority

Pull request development model

A pull request development model is common and ubiquitous across open source communities. Pull requests are public and transparent, like a registry. It is easy to see what is going in and out of a project. It also creates referenceable points in the overall project history.

This focused on adapting the team workflow to be compatible with how an outside contributor would participate.

- Do *verbose* code review

    - Using GitHub code reviews for publicly reviewing code
    - Leaving feedback, asking questions, making clear approvals or change requests
    - Requiring peer review for changes, when possible
- Core contributors push new changes via pull requests
    - Leverages CI and code health checks

Open communication: Public development discussion in Gitter

This gives people outside of UNICEF a door into the project. A public development discussion room offers people a place to ask questions, interact with developers, and see archives of past project discussion.

- Started and launched a new public Gitter chat room for developer conversation
- Moved development conversations public as often as possible
    - Use private chat rooms for UNICEF-confidential conversation
- Frequent engagement: Respond to conversation within working hours

First three months: Open developer documentation

This period focuses on the developer story around MagicBox. It addresses the minimum activation energy it takes for an outsider (i.e. open source contributor or future UNICEF developer) to understand and contribute to our project. It helps a contributor understand where their attention and focus is best spent for the most value for them and the project.

This builds off of work done since January 2018. It looks at resources for developer documentation after the Red Hat engagement. There is existing documentation published on a new ReadTheDocs site (magicbox.readthedocs.io), but it may be outdated from work during the Red Hat engagement. The work during this time focuses on strengthening our approach to attracting developers to the project. Ushahidi has a great example of developer documentation on their site.

This period builds off good open source practices and workflows to continue improving the onboarding experience for new contributors.

Review existing content

- Review existing documentation, assess accuracy
- Update existing documents where possible, purge irrelevant pages
- Create easy resources to get high-level overview of project
    - Public version of Mike's project diagram to include in project documentation

Identify new topics, write new content

- Identify topics missing on current state of project
    - Big picture: don't get bogged down in details
- Write new pages to fill gaps currently missing in documentation
    - Provide easy examples where new contributors can start

Next three months: Open user documentation

This period looks at user documentation and improving the user story with MagicBox. It centers the project development around a clear understanding of the end users and what their primary needs are. Effective user documentation helps users understand the project and helps **new developers understand the project vision** and who it is built to serve.

Part of this stage is an audience analysis and identifying their needs and wants. From the outcome of those needs and wants, identify new topics to write about or expand. It may be a small number of strong candidates.

This part of the timeline helps communicate vision and project purpose to reduce misunderstanding and wasted development time later.

- Identify MagicBox users and audience for user documentation
    - Think about who uses MagicBox… who cares, who wants to use it?
    - Country offices? State governments? Other UN agencies? External stakeholders?
- Write new documentation for users of MagicBox that explains how to use it and understand how it works
    - What do they need to know? What do they want to know?
- Include in existing documentation on ReadTheDocs

Final three months: Open identity and brand

This time period focuses more externally than internally. This stage looks at extending the identity and image of UNICEF to the open source development community. By establishing identity through a short mission statement and asserting the project meets its mission with real use cases, it builds presence in the open source space for MagicBox.

Establishing presence and identity helps the wider community understand the compelling nature of challenges and problems faced by UNICEF Innovation staff in our software projects. This supports an outsider becoming a contributor and guides them to participate in a humanitarian project from UNICEF.

Ultimately, this period is styled like a marketing approach to establish credibility and grow an interest in contributing development time to MagicBox projects.

- Write project mission statement
    - 2-3 paragraphs on project purpose and intentions
    - What does MagicBox do? Who is it designed for? What challenges and problems does it solve, generally?
- Create personas / use cases of MagicBox users to better explain challenges and problems
    - Detailed, narrow-focused views of how project meets mission statement
        - Positive impact on actual users
        - Bigger positive impact on potential contributors
    - Supports the mission statement (i.e. supporting evidence to show how we meet our mission)
        - Written in style that a technologist understands to get them on-board

Beyond: Extending the contribution pathway

Now, foundational groundwork is done with project management techniques. This foundation can sustain a community of contributors if nurtured and maintained. During this time period, begin creating contribution gateways through “starter” tasks or improvements.

“Starter” tasks are simple tasks or tickets to help the project. This stage begins with identification of “starter” task candidates, then writing a short how-to on resolving the task candidates. After, the issues are **clearly labeled** as “beginner” or “starter” tasks. These are guided contribution pathways into larger development contributions. Written material or social media promotion can highlight these tasks as contribution opportunities.

This cycle begins the early phases of building and developing a community of people around the project. Its success is integral to the success of open source workflows and methodologies in the project’s daily management.

- Evaluate current state of project, identify 4-5 “easy” tasks that are helpful, but are not critical to project success

    - Think “bite-sized” contributions to get a new contributor familiar with code
- File new issues / update existing issues with steps of how to work on or resolve the “easy” tasks
    - You may spend more time detailing how to solve the problem than it may take you to actually solve the problem – but *that’s the point*
    - Triage the issues: clearly label them as “easyfix” or “good first issue”
- **Write about them**: for blog post challenges, point people towards these tasks and you may have someone jump on a simple task (which over time can lead to something bigger and more involved)
    - Use social media to boost visibility on project content

Scope

Developing an open source community takes time, effort, and patience. Building a vibrant community is made easier when adopting an open source mindset. This document focuses on concrete activities and milestones to encourage that mindset. By practicing and improving in these areas, it speeds up the time needed to see outcomes with growing and developing a community. Adopting an open source mindset and culture is a key part to success.

Ultimately, these steps encourage the development of an open source mindset, and the mindset is built and maintained through some of the actions described in this document.

Outcome

The beginning of an open source community with non-UNICEF developers is attainable, and these milestones track a series of steps to begin building a community. This supports UNICEF leveraging the benefits of open source development practices and methodologies with the MagicBox project. The end objective is a small community of developers and other engineers with the project.

Appendix

This appendix features resources to better understand open source culture, practices, and methodology.

GitHub: Sustaining open source

https://sustainoss.org/

Google: Open source practices

https://opensource.google.com/docs/

Linux Foundation: Starting an open source project

https://www.linuxfoundation.org/resources/open-source-guides/starting-open-source-project/

Opensource.com: Open Organization Maturity Model

https://opensource.com/open-organization/resources/open-org-maturity-model