Internews Open Source Software Lightweight Needs Assessment

Table of Contents

Introduction 2

License 3

Instructions

How to run the assessment when working with an evaluator external to the tool team 4

How to conduct a self-assessment as a tool team 6

Capacity Action Planning (CAP) Workbook 8

Lightweight Needs Assessment Handbook for External Evaluators 19

Lightweight Needs Assessment Handbook for Tool Teams 26

Self-Assessment Scoring Sheet ..... attached separately as an editable Excel file

Introduction: Using the Lightweight Needs Assessment Instrument for Open Source Tool Teams

This instrument was originally designed to serve as a lightweight needs assessment process for open source security tools participating in an Internews program called BASICS (Building Analytical and Support Infrastructure for Critical Security tools). The assessment was created by Dr. Gina Helfrich, Program Officer for Global Technology at Internews.

Internews is pleased to make the Open Source Software Lightweight Needs Assessment available to the larger open source community for use in understanding the needs, pain points, opportunities, and strengths of open source tool teams.

This assessment was developed based on a number of existing resources. In particular, it adapts concepts from:

- the Internews Organizational Capacity Assessment (OCA)
- the SAFETAG Capacity Assessment
- the Linux Foundation's CHAOSS (Community Health Analytics Open Source Software) project[1]
- the Apache Project Maturity Model[2]
- the OSS Watch Openness Rating[3]
- and the URSSI maturity model for open source software.[4]

---

[1] Copyright © 2017 CHAOSS a Linux Foundation® project

Licensed under the MIT License (the "License");
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

[2] Copyright © 2020 the Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

See the License for the specific language governing permissions and limitations under the License.

[3] *Openness Rating: How open is your software project?* by **OSS Watch Team** is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

[4] Copyright © 2019 Sebastian P. Benthall/URSSI.

Tool teams who undergo the Open Source Software Lightweight Needs Assessment should walk away with a robust Capacity Action Plan to help guide priorities for the project team over approximately the next year. Outcomes from the assessment can be used to develop grant proposals and a clear rationale supporting the project's need for various resources.

The assessment can either be conducted by an external evaluator or by the tool team themselves. Two versions of the Internews Open Source Software Lightweight Needs Assessment Handbook are available (one "for External Evaluators" and one "for Tools Teams"), depending on which route you choose.

License

Internews Open Source Software Lightweight Needs Assessment by **Internews** is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0).

Instructions for External Evaluators

Prepare for the Assessment Process

First, have the tool team complete the Self-Assessment Scoring Sheet, including writing a short Justification for their self-score.

They will need to reference the self-scoring rubric throughout the self-assessment, which is available here.

Encourage tool teams to include as many core developers/project leaders as possible in the process of self-scoring. Only ONE self-assessment form should be completed per project/tool.

> **HOW TO SELF-SCORE**
>
> For each assessment area, read the descriptions under each number. Choose the number whose description most closely matches the case for your tool team.

Finally, have the tool team rank their top 5 areas to work on (out of the 10 total areas).

Review the Self-Assessment Scoring Sheet

The evaluation team should look at the importance and ranking that the tool team has given for each assessment area on their self-scoring form. Take the top 5 areas ranked of most importance and prepare to further explore those areas with the tool team.

For each assessment area to be discussed:

1. Read through the tool team's justification for their ranking in that area
2. Refer to the associated questions for that assessment area in the Internews Open Source Software Lightweight Needs Assessment Handbook for External Evaluators
    - document as many answers to the questions as you can based on publicly available information about the open source project
3. Note any follow-up questions you'd want to ask the tool team as a result of comparing your research to the scoring rubric, their self-score, and their scoring justification
4. Determine whether you think the self-score is accurate based on your research
    - If you think the tool team warrants a different score, make note of where you would place them and why

Reference Materials

Both the tool team and the evaluation team should have a copy of:

- the tool team's self-scoring answers
- the self-scoring rubric, which is available here
- the Internews Open Source Software Project Capacity Action Planning (CAP) Workbook
    - If possible, transpose the scores, importance, and justification from the Self-Assessment Scoring Sheet to each area in the CAP Workbook
    - At minimum, transpose answers for the Top 5 areas

Ensure the evaluation team has access to:

- the research conducted in (2) above
- the follow-up questions noted in (3) above
- the suggested score & rationale in (4) above
- the Internews Open Source Software Lightweight Needs Assessment Handbook for External Evaluators

How to Facilitate a Conversation Based on the Assessment Questions in the Handbook

Discussion should proceed **one assessment area at a time**. Do not move on to a new assessment area until you have finished discussion of the prior one(s). Consider starting with the most highly prioritized area and move to the next most important area each time.

1) For each area to be discussed, present the tool team's self-score and summarize their justification. Then, tell them whether you agree with their scoring and present your reasons, based on your research. Ask any follow-up questions you have already identified in the course of your research.

2) Next, talk through the Discussion Questions with the tool team. Try to steer the conversation towards answers that will ultimately prove useful in designing their capacity action plan. (In other words, focus on the practicalities at stake.)

3) Ask the tool team to re-assess their self-scoring for this area. Based on your conversation, do they want to revise their self-score? If so, note the new score. Be sure to emphasize that it is in the tool team's interest to score each area as accurately as possible, so that they have a clear rationale for support in the areas where they most need help.

4) Finish your discussion of the assessment area by asking the tool team to answer the associated open-ended questions in their Capacity Action Planning Workbook. The tool team should discuss their answers out loud with the evaluation team prior to documenting them in the Workbook.

5) Conclude by walking through the questions for the Strategic Planning & SWOT Analysis. Have the tool team discuss their answers out loud with the evaluation team prior to documenting them in the Workbook.

Remember: There are no "right" answers to this assessment! A certain amount of subjectivity is intrinsic to the scoring process. If a tool team feels strongly about scoring themselves a certain way, there is no need to push back even if the evaluation team might disagree with their score.

Instructions for Tool Teams/Project Leaders running their own assessment process

Complete the Initial Self-Scoring Process

Include as many core developers/project leaders as possible in the process of self-scoring. Only ONE self-assessment form should be completed for the project/tool.

Use the Internews Open Source Software Project Capacity Action Planning (CAP) Workbook to record your self-scoring, justification, and importance level for each area to be assessed.

First: Go through the CAP Workbook and fill in your scores, justification, and importance for each area. You will need to reference the self-scoring rubric throughout the self-assessment, which is available here. For now, skip answering the long-form questions – you will come back to these later in the assessment process.

> **HOW TO SELF-SCORE**
>
> For each assessment area, read the descriptions under each number. Choose the number whose description most closely matches the case for your tool team.

Finally, rank your top 5 areas to work on (out of the 10 total areas).

Reference Materials

Ensure you have a copy of:

- The Internews Open Source Software Project Capacity Action Planning (CAP) Workbook
- the tool team's self-scoring answers (marked in the CAP Workbook)
- the self-scoring rubric, which is available here
- the Internews Open Source Software Lightweight Needs Assessment Handbook for Tool Teams

Further Explore Priority Areas

Take the top 5 areas ranked of most importance and prepare to further explore those areas with the tool team. (Of course, you can go through as many of the assessment areas as you like or have time for.)

Discussion should proceed **one assessment area at a time**. Do not move on to a new assessment area until you have finished discussion of the prior one(s). Consider starting with the most highly prioritized area and move to the next most important area each time.

1) Refer to the associated questions for that assessment area in the Internews Open Source Software Lightweight Needs Assessment Handbook for Tool Teams
    - document as many answers to the questions as you can

2) Next, talk through the Discussion Questions in the Handbook. Try to steer the conversation towards answers that will ultimately prove useful in designing your capacity action plan. (In other words, focus on the practicalities at stake.)

3) Re-assess your self-scoring for this area. Based on your conversation, do you want to revise your score? If so, note the new score. It is in your interest to score each area as accurately as possible, so that you have a clear rationale for support in the areas where you most need help.

4) Finish your discussion of the assessment area by answering the associated open-ended questions in the CAP Workbook. The tool team should discuss your answers out loud together prior to documenting them in the Workbook.

5) Conclude by walking through the questions for the Strategic Planning & SWOT Analysis. The tool team should discuss your answers out loud together prior to documenting them in the Workbook.

Remember: There are no "right" answers to this assessment! A certain amount of subjectivity is intrinsic to the scoring process.

Internews Open Source Software Project Capacity Action Planning (CAP) Workbook

Code

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project produces Open Source software, for distribution to the public at no charge.	The project's code is easily discoverable and publicly accessible.	The code can be built in a reproducible way using widely available standard tools.	The full history of the project's code is available via a source code control system, in a way that allows any released version to be recreated.	The provenance of each line of code is established via the source code control system, in a reliable way based on strong authentication of the committer. When third-party contributions are committed, commit messages provide reliable information about the code provenance.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Licenses & Copyright

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The code is released under an OSI- or FSF-approved license.	Libraries that are mandatory dependencies of the project's code do not create more restrictions than the project license does.	The libraries that are mandatory dependencies of the project's code are available as Open Source software.	Committers are bound by an Individual Contributor Agreement that defines which code they are allowed to commit and how they need to identify code that is not their own.	The copyright ownership of everything that the project produces is clearly defined and documented.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Releases

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
Releases consist of source code, distributed using standard and open archive formats that are expected to stay readable in the long term.	The project maintains a section of the website/homepage where all releases and release notes are documented and archived.	Releases are signed and/or distributed along with digests that can be reliably used to validate the downloaded archives.	The release process is documented to the extent that it could be handed off to a new release manager.	The release process is documented and repeatable to the extent that someone new to the project is able to independently generate the complete set of artifacts required for a release.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Quality

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project is open and honest about the quality of its code. Various levels of quality and maturity for various modules are natural and acceptable as long as they are clearly communicated.	The project puts a very high priority on producing secure software.	The project provides a well-documented, secure and private channel to report security issues, along with a documented way of responding to them.	The project puts a high priority on backwards compatibility and aims to document any incompatible changes and provide tools and documentation to help users transition to new features.	The project strives to respond to documented bug reports in a timely manner.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Community

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project has a homepage that points to all the information required to participate in the community and contribute to the project.	The community welcomes contributions from anyone who acts in good faith and in a respectful manner and adds value to the project. New contributors receive welcoming messages/comments and thanks for their contributions.	Contributions include not only source code, but also documentation, constructive bug reports, constructive discussions, marketing and generally anything that adds value to the project. Code reviews are undertaken with the goal of accepting as many contributions as possible; comments on code contributions are constructive and show appreciation.	The community strives to be meritocratic and over time aims to give more rights and responsibilities to contributors who add value to the project. There is a defined and consistent process for acknowledging contributions of all kinds. The project attempts to make code reviews efficient and reduce the time between starting review and accepting the contribution.	The way in which contributors can be granted more rights such as commit access or decision power is clearly documented and is the same for all contributors. Most code contributions of reasonable quality are accepted.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Diversity & Inclusion

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project has a published code of conduct that applies to all members of the community. The project states publicly on its website and/or homepage(s) that it welcomes contributions from all people and encourages an environment of friendliness and respect.	The project has clear processes in place for managing code of conduct reports, including a conflict of interest policy and options for reporting to different individuals. The project actively works to welcome new contributors and uphold a friendly environment in its communications channels.	The project has established practices for welcoming new contributors and ensuring friendly and respectful communications amongst the community. The project leadership reflects some diversity of gender, ethnicity, and/or country of origin.	Project leadership works to limit technical jargon and other language that could be off-putting to new and diverse contributors. The project offers alternative modes of communication (e.g. text alternatives to video) and captioning for spoken communication. The community listens to feedback and is responsive to the needs of diverse contributors.	The project actively works to mitigate unconscious bias in communications and decision-making. Leadership actively works to recruit a diverse contributor community. Individuals in leadership represent a variety of gender identities, ethnicities, nationalities, abilities, etc.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Transparency & Consensus Building

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project maintains a public list of its contributors who have decision power -- the project's leadership consists of those contributors.	Decisions are made by consensus among project leaders and are documented on the project's main communications channel. Community opinions are taken into account but the leadership has the final word if needed.	Documented voting rules are used to build consensus when discussion is not sufficient.	The project maintains a public-facing roadmap that is kept up-to-date.	All "important" discussions happen asynchronously in written form on the project's main communications channel. Offline, face-to-face or private discussions that affect the project are also documented on that channel.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Governance

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project maintains a public list of its leadership.	The project's leadership roster clearly states the roles and responsibilities of each leader.	There is documentation of how decisions are made within the project and how conflicts are resolved.	The project has clear and documented opportunities to move into leadership roles which are available to all contributors. There is documentation of how leadership transitions are to be managed. Key information for effective governance of the project is managed such that there is no single point of failure.	The project maintains an active pipeline of potential new leaders for the project, including succession planning and a clear outline of back-up support for all key leadership roles.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

User Friendliness

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project maintains a user-facing website.	The project website includes clear instructions for getting help (e.g. documentation, opening issues).	The project itself contains a "help" function for users to report errors and seek support, native to the user interface of the project/app.	The project has a defined workflow for triaging user support requests and fixing bug reports. Users are notified when their issues are resolved.	The project strives to answer user questions in a timely manner. The issue queue is actively managed to reduce the age of open issues. The project utilizes user personas to help drive the development of features and updates. User support features may be available in multiple languages.

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Open Source Sustainability

Circle or highlight the number that best represents your self-rating for this category.

1	2	3	4	5
The project has published some code and there is at least one user of it.	The project has a small team of developers, often working on a volunteer basis. There is limited insight into who the users are and how they come across the project.	There are multiple developers or development teams spread out at different institutions, working on the same code. Some developers may have paid time to work on the project. The project has developed feedback mechanisms that help to illuminate who the users of the project are.	The project has a self-governing developer community deliberately supporting a broad user community. Funding opportunities to support the project have been identified and pursued.	The project is a self-sustaining organization dedicated to supporting the user and developer community (e.g. through commercial support, events, a software foundation, grants, donations, etc.)

How important is it to the project to work on improving your score in this assessment area?

Not Important | Somewhat Important | Very Important

Justification:

What are the biggest challenges or blockers for the project in this area?

List any quick or easy fixes the project could make in this area:

What steps could the project take in the next 6-12 months to improve in this area? What would you like to see the project accomplish?

Strategic Plan & SWOT Analysis

What is the most <u>important</u> issue that the project needs to address?

What is the most <u>urgent</u> issue that the project needs to address?

<u>With more person-hours devoted to it</u>, which one or two activities of the project would create the most positive impact over the next 6-12 months?

Given the resources you already have at hand, plus any potential resources coming available to you through external support (e.g. – a grant you are about to receive, a Google Summer of Code student coming on) …

What are 3 action-steps the project can take in the next 6-12 months to meaningfully advance its strategic plan and address the key issues identified above?

1)

2)

3)

Internews Open Source Software Lightweight Needs Assessment Handbook For External Evaluators

Assessment Areas and Guiding Questions

Code

Questions for the evaluation team to answer in advance:

- Does the project use a version control system?
- Does the project have a standard template or standardized instructions for commit messages, merge requests, and pull requests?
- Is the full history of the project's code available?
- What tools are required to build the source code, and are these tools widely available?

Licenses & Copyright

Questions for the evaluation team to answer in advance:

- What open source license does the tool use?
- What are the project software licenses?
    - Who has permission to run the software?
    - Who is permitted to examine the human-readable source code of the software?
    - Who is permitted to adapt or modify the source code of the software?
    - Who is permitted to redistribute the modified or unmodified source code of the software?
- Is the license recognized as a common Free and Open Source license?
- How much of the code base has declared licenses?
- Are all project dependencies clearly documented and license compatibilities audited?
- How many different licenses are there?
- Are there license conflicts?
- Is the licensee required to make modified or unmodified source code available if they redistribute the code?
- Are contributors required to sign a document stating they have the necessary permissions to make their contributions?
- Does the project have a standard expression of dependencies, licensing, and security-related issues?
- What methods (tooling) were used to identify project licenses?

Releases

Questions for the evaluation team to answer in advance:

- Is the software release cycle (including snapshots and major releases) consistent and predictable?
- Does the project use release candidates and solicit feedback on them?
- Where are release notes and notifications published?

Discussion Questions:

- Does the project have a formal role or procedure for who manages releases?
- Who is responsible for the creation of release notes, and what is their process for doing so?

Quality

Questions for the evaluation team to answer in advance:

- Does the project have a documented process for responding to reported security issues?
- Does the software have a security certification?
- How does the project manage reporting of security issues/bugs?
- What is the total number of reported bugs?
- What is the average age of reported bugs?
- What is the approximate ratio of open bugs that are defects vs. feature requests?
- What language is the project written in?
- What are the documented community processes associated with the development of the source code?
- Does the project document when changes are not backwards-compatible? What tools and documentation does the project provide to help users when such changes are made?

Discussion Questions:

- What percentage of the code has test coverage?
    - (for some projects, this might be visible in their code repo)
- Has the project gone through an external review or audit?
- Does the project use continuous integration testing?
- Does the project have an internal standard or goal for responding to bug reports in a timely fashion?
- Is the project able to easily import upstream security patches to their dependencies?
- What level of training does a person need in order to read, understand, and propose modifications to the source code?

Community

Questions for the evaluation team to answer in advance:

- When did development of this tool first begin?
- Approximately how many regular contributors are there?
- Are the maintainers primarily volunteers, or are they employed by an organization or company specifically to work on the tool?
- Is there publicly accessible and easy to find documentation about how to participate in the project?
- Is the process for contributors being granted increasing rights and responsibilities clearly documented and publicly available?
- Is it easy to acquire, build, configure and install the source code from scratch?
- What activity is there within the community? (e.g. mailing list traffic, pull request discussion, chat group activity)
- How are new contributors welcomed as part of the workflow of their contribution(s)? On the mailing list/communications channels?
- How does the project advertise that non-code contributions are welcome?
- How does the project acknowledge and thank people for their contributions?

Discussion Questions:

- What procedures are in place for managing efficient and friendly code reviews?
- What efforts do the project leaders make to recruit and retain new contributors?
- How does the project monitor and measure whether the community is growing, retaining, and upskilling/advancing contributors?

Diversity & Inclusion

Questions for the evaluation team to answer in advance:

- Does the project have a Code of Conduct?
- Does the project maintain conflict-of-interest policies and reporting lines for the Code of Conduct?

Discussion Questions:

- Does the project have established procedures for responding to Code of Conduct reports?
- Who plays the role of community manager for the project?
- Does the project work to ensure that code contributions are not seen as more valuable than other forms of contribution (e.g. - documentation, marketing, bug reports, community management)? How?
- How does the project approach trying to include people with disabilities? People whose first language is not that of the project community?
- How diverse is the project leadership? Is diversity a factor taken into consideration when doing succession planning?

- Does the project leadership communicate the importance of inclusivity, respect, and fairness to the community? How?
- How does the project approach recruiting new contributors?

Transparency & Consensus Building

Questions for the evaluation team to answer in advance:

- Is any of the project knowledge available in more than one language?
- Are there public archives of the knowledge?
- Are there documented data recovery processes in place?
- Are there documentation sources external to the project?
- How good is the developer-specific public documentation?
- Is the knowledge stored in publicly published data formats (with appropriate metadata) that will make it accessible over time?
- Is there any financial, legal, or technical barrier to accessing or acquiring some or all knowledge in the project?

Discussion Questions:

- How are project decisions made?
- How is project knowledge managed in case a key person becomes unavailable? (i.e. – API keys, PGP/signing keys, passwords to the repo and/or user accounts)
- Does the project discourage major project communications outside the approved channels?
- Who is able to access all the (non-private) project knowledge?
- Who is able to contribute to the project knowledge?

Governance

Questions for the evaluation team to answer in advance:

- Approximately how many maintainers are there for this tool, and how much time do they devote to its development?
- Has this tool team ever *applied* for funding before?
- Has this tool team ever *received* funding before?
- Is there clear leadership in the project?
- Are the structure and policies of the project clearly and publicly documented?
- Are there publicly accessible behavioral guidelines for the project?
- Is there an avenue and structure for recourse beyond the project maintainers?

Discussion Questions:

- Is the project leadership elected by the project community?
- Who is able to contribute to the project development?

- Who is able to obtain commit rights to the project development?
- How does the project approach succession planning and developing a pipeline of new potential leaders?
- Is there a single point of failure or control for committing changes to the primary project source?
- Does the actual practice of project governance reflect the documentation describing governance procedures?

User Friendliness

Questions for the evaluation team to answer in advance:

- What information do you have about the community of users of this tool?
- Is the software easy for users to access, install and run so it can be trialled (for those who have access)?
- Is the project and its associated user support features and documentation available in more than one language?
- How long do users generally have to wait before their issue is resolved?
- How good is the user-specific public documentation?
- Who is able to obtain easy access to and use the software?

Discussion Questions:

- What is the project's approach to triaging and managing the issue queue?
- Who are your users? How do you find out about the general populations that are using the project?
- What are the potential challenges you are aware of that your more vulnerable users may be facing? Do these challenges influence development of the project? How?
- Are there user personas used to guide project development? If so, how were the personas developed?

Open Source Sustainability

Questions for the evaluation team to answer in advance:

- What information do you have about the community of users of this tool?
- Approximately how many regular contributors are there?
- Approximately how many maintainers are there for this tool, and how much time do they devote to its development?
- Are the maintainers primarily volunteers, or are they employed by an organization or company specifically to work on the tool?
- Has this tool team ever *applied* for funding before?
- Has this tool team ever *received* funding before?
- How many contributors have some or all of the time they spend on the software paid for?
- How many dependencies does the project have? What are they?

Discussion Questions:

For larger projects:

- What proportion of the core developers are from the same company, institution or department?
- What is the percentage of work in the community that can be attributed to contributors who are all from the same organization/company?

For small projects:

- How did the project come to be? How did the core developers meet?

For ALL projects:

- What opportunities does the project have to bring in needed resources?

Strategic Planning & SWOT Analysis: (strengths, weaknesses, opportunities, threats)

Answers to the following questions will help us to understand the "big picture" for the project/tool team and to develop a capacity action plan that addresses the most urgent needs of the project. As many members of the tool team as possible should participate in answering these questions, in order to highlight areas of agreement and areas where there are differences of perspective.

1) **What is the "strategic plan" for the project?**
    a. What organizational or business model is the project operating under?
        i. Is the goal for the project to be self-sustaining as an openly developed project maintained largely by volunteers? If not:
            1. Which revenue models are available to build a revenue stream around the project?
            2. Are there alternatives (competitors) to the project? If so, are they FLOSS or commercial?
            3. Is the project potentially applicable to more than one industry?
    b. Does the project have a roadmap? Who is responsible for setting the vision and concrete direction of the project?
    c. What are the project's goals with regards to:
        i. Technical capacity
        ii. Product features
        iii. Contributor community
        iv. User experience
        v. Responsiveness to security issues

2) **What obstacles or challenges is the project currently facing? How do they interfere with the project's goals as articulated in question 1?**

3) **Are there obstacles or challenges the project anticipates facing <u>in the next year</u>? Describe.**

4) **What are the project's <u>potential</u> challenges (not an active challenge, but a situation that could easily create problems)?**
    a. E.g. – Low "bus factor" is a vulnerability, since if one or more maintainers are out of commission for any reason, this would pose a huge challenge to the project.

5) **What are the greatest strengths of the project as it currently exists?**

6) **What opportunities could the project capitalize on, given appropriate resources?**

7) **Which of the project's activities currently require the most person-hours (whether or not such time is actually devoted to the activity)?** E.g. – Triaging bug reports and issue management might require a lot of time, but at present the project doesn't have the resources to devote to it, so the queue continues to grow.
    a. Which activities of the project could benefit from more person-hours than are currently available for that activity?
    b. Could any of the project's activities be made more efficient through automation or better tooling, if the time/resources were available?

Internews Open Source Software Lightweight Needs Assessment Handbook For Tool Teams

Assessment Areas and Guiding Questions

Code

- Does the project use a version control system?
- Does the project have a standard template or standardized instructions for commit messages, merge requests, and pull requests?
- Is the full history of the project's code available?
- What tools are required to build the source code, and are these tools widely available?

Licenses & Copyright

- What open source license does the tool use?
- What are the project software licenses?
    - Who has permission to run the software?
    - Who is permitted to examine the human-readable source code of the software?
    - Who is permitted to adapt or modify the source code of the software?
    - Who is permitted to redistribute the modified or unmodified source code of the software?
- Is the license recognized as a common Free and Open Source license?
- How much of the code base has declared licenses?
- Are all project dependencies clearly documented and license compatibilities audited?
- How many different licenses are there?
- Are there license conflicts?
- Is the licensee required to make modified or unmodified source code available if they redistribute the code?
- Are contributors required to sign a document stating they have the necessary permissions to make their contributions?
- Does the project have a standard expression of dependencies, licensing, and security-related issues?
- What methods (tooling) were used to identify project licenses?

Releases

- Is the software release cycle (including snapshots and major releases) consistent and predictable?
- Does the project use release candidates and solicit feedback on them?
- Where are release notes and notifications published?

Discussion Questions:

- Does the project have a formal role or procedure for who manages releases?
- Who is responsible for the creation of release notes, and what is their process for doing so?

Quality

- Does the project have a documented process for responding to reported security issues?
- Does the software have a security certification?
- How does the project manage reporting of security issues/bugs?
- What is the total number of reported bugs?
- What is the average age of reported bugs?
- What is the approximate ratio of open bugs that are defects vs. feature requests?
- What language is the project written in?
- What are the documented community processes associated with the development of the source code?
- Does the project document when changes are not backwards-compatible? What tools and documentation does the project provide to help users when such changes are made?

Discussion Questions:

- What percentage of the code has test coverage?
    - (for some projects, this might be visible in their code repo)
- Has the project gone through an external review or audit?
- Does the project use continuous integration testing?
- Does the project have an internal standard or goal for responding to bug reports in a timely fashion?
- Is the project able to easily import upstream security patches to their dependencies?
- What level of training does a person need in order to read, understand, and propose modifications to the source code?

Community

- When did development of this tool first begin?
- Approximately how many regular contributors are there?
- Are the maintainers primarily volunteers, or are they employed by an organization or company specifically to work on the tool?
- Is there publicly accessible and easy to find documentation about how to participate in the project?
- Is the process for contributors being granted increasing rights and responsibilities clearly documented and publicly available?
- Is it easy to acquire, build, configure and install the source code from scratch?
- What activity is there within the community? (e.g. mailing list traffic, pull request discussion, chat group activity)
- How are new contributors welcomed as part of the workflow of their contribution(s)? On the mailing list/communications channels?
- How does the project advertise that non-code contributions are welcome?
- How does the project acknowledge and thank people for their contributions?

Discussion Questions:

- What procedures are in place for managing efficient and friendly code reviews?
- What efforts do the project leaders make to recruit and retain new contributors?
- How does the project monitor and measure whether the community is growing, retaining, and upskilling/advancing contributors?

Diversity & Inclusion

- Does the project have a Code of Conduct?
- Does the project maintain conflict-of-interest policies and reporting lines for the Code of Conduct?

Discussion Questions:

- Does the project have established procedures for responding to Code of Conduct reports?
- Who plays the role of community manager for the project?
- Does the project work to ensure that code contributions are not seen as more valuable than other forms of contribution (e.g. - documentation, marketing, bug reports, community management)? How?
- How does the project approach trying to include people with disabilities? People whose first language is not that of the project community?
- How diverse is the project leadership? Is diversity a factor taken into consideration when doing succession planning?
- Does the project leadership communicate the importance of inclusivity, respect, and fairness to the community? How?
- How does the project approach recruiting new contributors?

Transparency & Consensus Building

- Is any of the project knowledge available in more than one language?
- Are there public archives of the knowledge?
- Are there documented data recovery processes in place?
- Are there documentation sources external to the project?
- How good is the developer-specific public documentation?
- Is the knowledge stored in publicly published data formats (with appropriate metadata) that will make it accessible over time?
- Is there any financial, legal, or technical barrier to accessing or acquiring some or all knowledge in the project?

Discussion Questions:

- How are project decisions made?
- How is project knowledge managed in case a key person becomes unavailable? (i.e. – API keys, PGP/signing keys, passwords to the repo and/or user accounts)
- Does the project discourage major project communications outside the approved channels?
- Who is able to access all the (non-private) project knowledge?
- Who is able to contribute to the project knowledge?

Governance

- Approximately how many maintainers are there for this tool, and how much time do they devote to its development?
- Has this tool team ever *applied* for funding before?
- Has this tool team ever *received* funding before?
- Is there clear leadership in the project?
- Are the structure and policies of the project clearly and publicly documented?
- Are there publicly accessible behavioral guidelines for the project?
- Is there an avenue and structure for recourse beyond the project maintainers?

Discussion Questions:

- Is the project leadership elected by the project community?
- Who is able to contribute to the project development?
- Who is able to obtain commit rights to the project development?
- How does the project approach succession planning and developing a pipeline of new potential leaders?
- Is there a single point of failure or control for committing changes to the primary project source?
- Does the actual practice of project governance reflect the documentation describing governance procedures?

User Friendliness

- What information do you have about the community of users of this tool?
- Is the software easy for users to access, install and run so it can be trialled (for those who have access)?
- Is the project and its associated user support features and documentation available in more than one language?
- How long do users generally have to wait before their issue is resolved?
- How good is the user-specific public documentation?
- Who is able to obtain easy access to and use the software?

Discussion Questions:

- What is the project's approach to triaging and managing the issue queue?
- Who are your users? How do you find out about the general populations that are using the project?
- What are the potential challenges you are aware of that your more vulnerable users may be facing? Do these challenges influence development of the project? How?
- Are there user personas used to guide project development? If so, how were the personas developed?

Open Source Sustainability

- What information do you have about the community of users of this tool?
- Approximately how many regular contributors are there?
- Approximately how many maintainers are there for this tool, and how much time do they devote to its development?
- Are the maintainers primarily volunteers, or are they employed by an organization or company specifically to work on the tool?
- Has this tool team ever *applied* for funding before?
- Has this tool team ever *received* funding before?
- How many contributors have some or all of the time they spend on the software paid for?
- How many dependencies does the project have? What are they?

Discussion Questions:

For larger projects:

- What proportion of the core developers are from the same company, institution or department?
- What is the percentage of work in the community that can be attributed to contributors who are all from the same organization/company?

For small projects:

- How did the project come to be? How did the core developers meet?

For ALL projects:

- What opportunities does the project have to bring in needed resources?

Strategic Planning & SWOT Analysis: (strengths, weaknesses, opportunities, threats)

Answers to the following questions will help us to understand the "big picture" for the project/tool team and to develop a capacity action plan that addresses the most urgent needs of the project. As many members of the tool team as possible should participate in answering these questions, in order to highlight areas of agreement and areas where there are differences of perspective.

8) **What is the "strategic plan" for the project?**
    a. What organizational or business model is the project operating under?
        i. Is the goal for the project to be self-sustaining as an openly developed project maintained largely by volunteers? If not:
            1. Which revenue models are available to build a revenue stream around the project?
            2. Are there alternatives (competitors) to the project? If so, are they FLOSS or commercial?
            3. Is the project potentially applicable to more than one industry?
    b. Does the project have a roadmap? Who is responsible for setting the vision and concrete direction of the project?
    c. What are the project's goals with regards to:
        i. Technical capacity
        ii. Product features
        iii. Contributor community
        iv. User experience
        v. Responsiveness to security issues

9) **What obstacles or challenges is the project currently facing? How do they interfere with the project's goals as articulated in question 1?**

10) **Are there obstacles or challenges the project anticipates facing in the next year? Describe.**

11) **What are the project's potential challenges (not an active challenge, but a situation that could easily create problems)?**
    a. E.g. – Low "bus factor" is a vulnerability, since if one or more maintainers are out of commission for any reason, this would pose a huge challenge to the project.

12) **What are the greatest strengths of the project as it currently exists?**

13) **What opportunities could the project capitalize on, given appropriate resources?**

14) **Which of the project's activities currently require the most person-hours (whether or not such time is actually devoted to the activity)?** E.g. – Triaging bug reports and issue management might require a lot of time, but at present the project doesn't have the resources to devote to it, so the queue continues to grow.

a. Which activities of the project could benefit from more person-hours than are currently available for that activity?
b. Could any of the project's activities be made more efficient through automation or better tooling, if the time/resources were available?