Proposal: Guide to rewarding OSS project contributors

# Scope of this guide

For the sake of simplicity and focusing on rewarding contributors, this guide focuses only on maintaining good relationships with contributors by acknowledging and rewarding them.

# Goal of this guide to reward contributors

The main goal for this doc is providing basic guidance to reward contributors, especially those that interacts with the repo directly with activities/actions from submitting PRs to having feedback submissions, providing answers in the issues and discussions

# Background

Any open source software (OSS) projects are often said gaining more popularities and supporters, but in reality, the facts are not that. The popularities of an OSS projects are not common in most of OSS and what so called many supporters are not *really* supporters. This means, the notion or saying that (every) OSS project is gaining popularities and therefore has many supporters are not the common/general facts.

For example: the terms supporters in this guide can mean the backers of the OSS financially or in work contributions from a certain companies and common contributors that have direct interactions with the repo in terms of PRs and feedbacks.

## Known common (too general) misconceptions

A popular and active OSS project is often seen by these traits:

1. Number of stars and forks of the repo. But since there are many OSS repos that have quite large number of stars and forks but when they have issues and PRs that only active for long time ago (e.g., last active/merged PR is a year ago) then the repo can should not be considered popular. There are also OSS repos that have large number of stars and forks, but there are almost no issue/PR at all although the maintainers are actively engaged with any PR/issues when they are submitted.
2. Number of active contributors. This measure of number of active contributors depends on what are those quantifications to be considered to be active contributors, based on how the OSS maintainers judge them or we could have basic common quantification as starters. For example: usually PRs by contributors are the only metrics.
3. Those that submits feedbacks are often not counted as contributors. Most of the time, people that submits feedback are treated with lower trust and often ignored.

The point #3 is crucial, because without any feedback in terms of issues, we couldn’t really measure that the OSS is healthy. Those feedback should be treated as good metric that the OSS projects are getting true interaction and also support in terms of feedback.

Therefore, we should stop to believe/acknowledge common notion of being popular and supporters, because they are too general and too broad. The notion of popular and supporters should be within one combined measurement, i.e., a good/healthy OSS should have active interactions of contributions of both PRs and issues (especially issues of feedback such as bugs and enhancements. The reason is, saying popular is not a full measure that the OSS is actually thriving or healthy.

We need to have good treatments of OSS supporters in terms of those that interact directly with the OSS projects in terms of contributions, mainly in PRs and feedback/suggestions as issues. For the next discussion, the feedback/suggestions can be considered just as feedback whether they are feedback as bug reports, suggestions, and feature proposals.

Before we could give rewards to contributors, we need to have simple and basic starting point of a contributor, now that we have identified the background condition and misconceptions.

# The definition of contributors, expanded

Based on the previous facts, we should define contributors easily as those that do submit PRs and feedback. We should have inclusion defined explicitly to honour those that have submitted PRs and feedback.

To be as much as inclusive as possible, even those that provide answers on feedback must be considered as contributors. These people often help the maintainers to keep the feedback discussion answered and quite often, they also can provide additional feedback to the maintainers.

On some OSS projects, maintainers are considered as part of contributors.

For example: issue #222 discuss a bug in an OSS library, then it’s answered by the maintainers and other people, although these people may not be those that often submit feedback.

This attention of inclusion maybe hard at first, but those valuable answers to feedback, especially when the maintainers may not answer immediately must also be considered as good contributions.

Put it in simplest forms that inclusive/part of contributors, contributors are:

1. Not the original maintainers.
2. Those that submit legit and good PRs
3. Those that submit good feedback
4. Those that provide answers to questions and feedback,
5. Those that provide helpful clarification to help other people that ask questions on submitted feedback.

Point #4 is important, because I have seen many people that don’t submit PRs and feedback at all, but provide solutions as answers and helpful additional tips on using the OSS.

## Good feedback definition

The definitions of good feedback should be left as open as possible to the OSS maintainers, but good feedback should at least have these characteristics: (from easy to a bit require more attention to check)

1. Not spam. See also below notes.
2. Related to the usage (such as bugs report that has reproducible steps, existing features, and additional feedbacks) and part of the OSS including technical governance
3. More test coverages
4. New Feature proposals
5. Almost not related to the OSS directly but provide valuable additional value in medium or long terms, such as adding or enhancing CI/CD support as part of the repo itself.

NOTE on note #4: If the CI/CD support isn’t available at all. we as part of .NET Foundation project committee could jump in to offer help by providing assistance on having CI/CD support as part of project onboarding benefits as long as it’s in accordance by .NET Foundation offering.

These are samples of not spams but they aren’t good feedback and can be very bad:

1. Saying kudos/praising the OSS project submitted as issues. If there’s feedback as part of having good feedback in terms of telling the experience of the OSS usage, we could recommend to minimize noise by including these kind of issue submissions. If we need to accommodate inclusion of praise as a way to promote the repo itself, it can be done by moving to discussion feature in GitHub, as usually discussion is more relaxed.
2. Unconstructive feedback as defined by the maintainers.

NOTES on spam of PRs feedback:

1. Good PRs and feedback are not spam PRs and spam feedback. Those that submit these spam contributions must not be considered as contribution at all. This is important, especially we must reject and delete those spam PRs. We have experienced a flood of spam PRs before like those in Hacktoberfest 2020, by engaging in by gamifications of contributions and rewarded with swags such as tshirts and other goodies. We should reward those legit contributions with useful rewards, but the reward must come with predefined and written rules before.
2. Submitters of spam PRs and feedback is recommended to be banned. But maintainers should be free to have their own governance/rules when dealing with these spams, as long as it still meet/comply with .NET Foundation Code of Conduct.

# Rewards for contributors

The actual kind of rewards can be many, but base on the nature of open source that provide collaborative effort of maintainers and contributors, we could require basic traits of the reward in terms of the reward processes and the acknowledgement as reward as starting points.

Forms of rewards can be in any forms, we recommend to meet minimum rewards in the form of:

1. Fair Acknowledgments of contributions by mention them, see below notes
2. Additional privilege on the repo given to those that earned trusts high enough

Other rewards such as swags or other gimmick should be optional, because we should give rewards that comply with the basic spirit of the opensource: acknowledge those contributors.

## Reward prerequisite requirements

Now that we have identified the definitions of what contributors are, the reward model should be simple as possible but it should meet these basic requirements:

1. Should be generally inclusive as governed by .NET Foundation Code of Conduct.
2. Should be inclusive and fair for contributors as we have defined in previous section. Therefore, we should not only consider those that submit PRs only as sole contributors.
3. Should only include active contributors within predefined reasonable time, especially when measure active contributions and consider last contribution date. See the discussion on “encouraging basic sustainability of contributions” below.
4. Clearly acknowledge contributors by having a list of contributors, and this should be updated frequently.
5. Avoid meritocracy to avoid bias to always listen to frequent contributors with high quality bar.
6. Carefully giving additional rights/privilege on some contributors. Giving additional privilege when maintainers start to trust more on some contributors are ok, but this should be done carefully. Of course, trusts can be earned by contributors, but the way to get the privilege of trusts should be clearly communicated in the beginning and it should not be subjective. If trusts are given without clear definitions of how to earn trusts, then the OSS governance may be trapped into favoritism, and this condition may be worse than meritocracy.

It can be optional to have detail contribution type in the contributor lists, therefore we aren’t just giving acknowledge for contributions but also provide acknowledge on each contributor’s contribution kind.

## Define list of contributions

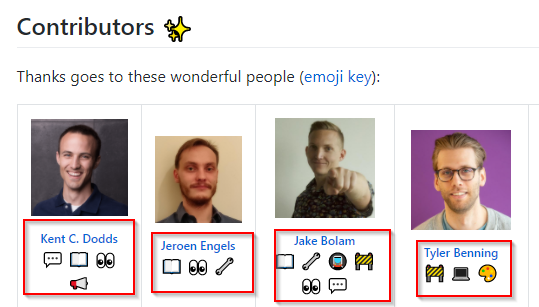
To have a list of contributors in GitHub, we could do manually or leverage tools that are available on GitHub as bot. But if we do manually, it will be quite hard for maintainers to maintain an objective list of contributors, because it is quite common that not all PRs and feedback issues are good, especially in a very active and highly popular OSS projects that has large users base.

We can do automatically adding contributors to ease this. For example, “all contributors” tool (repo link: <https://github.com/all-contributors/all-contributors> )

This “all contributors” bot has basic definitions on how the maintainers can acknowledge contributions from contributors, but it has feature to acknowledge contribution kind by having basic type of contributions predefined, such as PRs, feedback as issue. But it also has the ability to provide additional contribution type as defined by the repo maintainers.

For example, the list of contributors of all contributors bot, like in the bot README.

Under the Github profile picture in the contributor list, the icons defined the contributions that have been verified:



After we have acknowledged them with proper rewards, the reward model should be consistently maintained and it should have continuous and sustainability.

## Encouraging basic sustainability of contributor rewards

The sustainability is important, especially if there are special event happened that may change the governance directly/indirectly. Example of these events/conditions with the recommended suggestions are:

1. The original maintainer transfers project ownership. The transfer should be given to existing contributors, especially if one or a small group of contributors have earned enough objective trusts ***and*** must be still active within common period of last active, for example last active is 1 month to 3 months before. If we allow transfers to other contributors that has been inactive for very long period, it would be definitely unfair.
2. The repo is converted into archive only repo. This is quite common, and it may bring up bad side effect such as losing trusts from contributors. And if the archived repo is part of organizations that has many archived repos, then the trust lose scope will be organization wide. The archive only repo should have replacement repo, or the original repo owner and maintainers could transfer the project ownership to some contributors.
3. Some active contributors are no longer active. Always be careful when maintaining list of contributors, especially when some of the previous active contributors are no longer active. The option/decision to maintain list of contributors that still include contributors that are no longer active could be optional, but it is recommended to reconsider the inclusion because having list of active contributors should be dynamic to maintain objective and fairness, not just a list that keep added. See also maintain the dynamics of the contributors below.

NOTE: on point of transfer of project ownership in #1 and #2, I suggest to require the project owner to inform .NET Foundation before the transfer.

## Bonus: maintain the ongoing dynamics of the contributor rewards

After some time, the contributors might be in active, and even the repo may change owners. This section is the continuation of sustainability, emphasize on having optional guide to maintain the dynamic changes of contributor activities.

Basic management of dynamics of contributors are:

1. Give some contributors that are no longer active but they were highly active and helpful for long time by defining a list of its own. An example of this is Eclipse Project Emeritus: <https://www.eclipse.org/projects/committers-emeritus.php>
2. The period of inactivity should be defined freely by the repo maintainers, but it should give enough time. For example: 6 months of inactivity.
3. Revoke additional privilege given to contributors that are no longer active as fair as possible, to match with predefined inactivity.