

Volunteer Armies Can Deliver Quality Too

Achieving a Successful Result in Open Source, Standards Organizations, and Other Volunteer Projects

Julie Fleischer

Julie.N.Fleischer@Intel.com

Abstract

Delivering quality can be straightforward in an organization that creates multiple products and has a series of time-tested processes and procedures in place to guarantee success. However, at some point in your career, you may be asked to deliver a quality result with a team that has self-defined or loosely-defined processes, includes volunteers whose “day job” is not your project, and who rely on a consensus process with individuals from diverse, and sometimes competing, backgrounds and agendas to make decisions. This paper is written by a Program Manager and QA Lead who has spent the better part of the past decade delivering quality test and certification programs, specifications, and projects in various open source programs and standards development organizations. We'll cover tips and techniques for achieving success in those volunteer-based environments.

Biography

Julie Fleischer has worked at Intel for the past thirteen years and has held a variety of roles in program/project management, QA leadership, and software development. She currently is a Technical Program Manager for Linux driver development and also briefly held a position as the Technical Program Manager for the Yocto Project (www.yoctoproject.org), an open source project for creating embedded Linux distributions. Prior to those roles, she worked as the Chair of the Test and Certification Working Group for Continua Health Alliance where she led the team to define and implement a certification program towards v1.0 and v1.5 of the Continua Design Guidelines. She also chaired the team that created the USB Personal Healthcare Device Class standard and project managed the creation of the v1.0 Continua Design Guidelines. She has spent over a decade in open source and standards organizations and has led many "volunteer armies" to accomplish successful results.

Julie received her BS and MS in Computer Science from Case Western Reserve University and has presented at PNSQC in the past on technical and interpersonal topics.

1. Introduction

What is a volunteer army?

A volunteer army is a team of more than one person where at least some members of the team are not getting financially compensated for the work they do. In other words, their “day job” may not be to work on the project that the volunteer army is working on. These individuals typically come from diverse, and sometimes competing, backgrounds and agendas. In the software world, volunteer armies can be found in open source projects, standards organizations, or other special interest groups. These volunteer armies may not have formal quality assurance teams or plans and may not have official project management roles or practices.

Decision making in a volunteer army is typically consensus-based; however, depending on the parent organization, decision making can be hierarchical or even a mix of types. For example, a standards organization may have strict voting rules on requirements for voting eligibility and participation, but an open source project may use meritocracy and hierarchy in order to make decisions affecting the project.

Not all processes in a volunteer army may be explicit or documented. Standards organizations may have clear membership and voting rules as well as requirements on publication of standards; however, they may not have requirements on how meetings are run, how often members meet, and how day-to-day decisions are made. Depending on the size and complexity of an open source project, it may or may not have a website or Wiki defining the processes and procedures used for submitting and triaging bugs or developing and releasing code. However, processes describing how decisions are made and how the group is managed may be lacking.

Delivering quality products in environments like these can pose unique challenges. It can be difficult to get a group of part-time participants with competing agendas to come to consensus. It can also be challenging to make progress when the team members have unpredictable schedules and availability. Further, maintaining motivation with a team that has competing priorities and interests requires a special finesse. This paper will describe tips for achieving quality results in volunteer armies like these along with anecdotes and advice to assure you that volunteer armies can deliver quality – and have fun while doing it!

2. Respect your volunteer army.

Perhaps the most effective best practice for making progress and delivering quality results in a volunteer army is to respect the volunteer army. Since volunteer armies aren't financially compensated for their work, they require a sense that they are adding value and significant contributions to feel that their time is well spent in a volunteer organization. Eric S. Raymond calls this feeling “egoboo.”ⁱ That is, the sense of enhancing one's reputation among others in the group. As a leader, or active participant, of a volunteer army, it is important to show respect for your volunteer army. This can be anything from simple “Thanks!” to more formal celebrations and parties when warranted.

Some techniques that I have found effective are:

- 1) A “Thank you!” goes a long way. I like to make my “Thank you”s explicit and let the person know exactly what they did that was worthwhile. This helps encourage the same kind of activity again since the individual knows what was valued.
- 2) Listening and letting others talk is the best way to encourage more feedback. When you don't interrupt others, allow conversations to take their time, and actively document and respond to other's input, it tells them their opinions are worth-while and valued.
- 3) Just because a volunteer organization may not have the budget to have the formal milestone parties that a company does doesn't mean you can't celebrate milestones. When a team I chaired finished writing a technical specification, we all knew we wouldn't be meeting again face-to-face to celebrate. So, we held a “virtual” celebration, complete with awards, virtual beer (i.e. a

power point picture of a pint of beer - The team didn't find this as satisfying as the real thing, but appreciated the gesture.), and sharing anecdotes from our work together.

Although listening and letting others express their ideas is one way to show respect for your volunteer army, it is still important to guide discussions. Volunteer organizations often can get new members with many ideas, which may or may not be in the direction of the group. Standards organizations frequently face this issue when a face-to-face meeting may bring out many different attendees than the attendees on regular con-calls. While it is important to solicit and listen to newcomers ideas in order to ensure the newcomers become regulars and feel valued, sometimes you also need to ensure the group is true to their original vision. It is important to remember that listening to others and respecting their opinions does not have to mean changing your original plan. Taking the time to listen to a new idea and investigate it if appropriate is often enough to make a newcomer feel their opinion is valued. Most people don't expect that every idea they provide will be acted on and implemented, but they do appreciate a forum where their ideas are listened to and considered. If a conversation appears to be going on too long, taking a follow up action item to investigate is often an effective means for ensuring opinions are heard and the team stays on track. In addition, section 5 will talk about one technique for maintaining direction in a conversation by using a straw man proposal.

3. Always have a list of tasks needing volunteers

One quick way to ensure you can engage volunteers from the start is to make your "to do" list for your project or team visible to any volunteer that might want to join. This can take many forms: Wiki page, bug database, web site, Kanban, document, or anything else that is effective for the project. It is useful if your task list contains an idea of the scope of the task in terms of time required and technical expertise and experience required to complete. If you have the time, a phone call to a prospective new member to introduce them to your project as well as review the requirements of the tasks that that member may be a fit for can help integrate that member quickly.

4. Don't be afraid to jump in.

In organizations with clearly defined roles and responsibilities, a program manager, QA lead, or other person responsible for making things happen, has a management escalation path to ensure the project is getting the maximum effort and prioritization from the individuals on the team. However, with collections of volunteers, those lines of authority don't exist. If you want to encourage others to participate, you sometimes need to take the initiative yourself. This means being willing to do whatever is required to make sure the team makes progress, even if it's not something that is exactly in your skill set.

Some ways I have used this tip successfully in the past were:

- 1) When I chaired a working group which was writing a specification for transport connectivity of consumer electronic devices and personal health devices, I relied on one expert and other technical and personal healthcare experts on the team to establish our architecture and design. However, these people didn't have time to write the specification. So, I took on the job of translating the architectural decisions that we agreed upon in meetings into specification language. With the additional help of a great technical editor, we were able to create a readable and implementable specification.
- 2) When someone sends out a document or proposal for review in a volunteer group and has disclosed to me that they want the group's feedback, I like to use the "Reply All" technique on my review to encourage others to jump in and perform a review as well. I especially find this effective if I add in questions to others or describe areas where someone else's expertise may be of assistance. This encourages them to look at the area I called out and maybe perform an even larger review.

In those cases when you are not a leader but simply a participant, the only way to become a part of the group may be to simply jump in. This is especially true in groups where there is no clear leader or

defined roles and responsibilities, such as in school or extra-curricular volunteer organizations. This technique is best used when combined with other techniques, such as respecting your volunteer army (see section 2) to ensure you take on a reasonable amount but still empower others to take on roles as well.

5. Begin decision making with a straw man proposal.

There is an old folk story about a group of hungry travelers who would like food from the local villagers, but have no compensation to offer¹¹. As the story goes, the hungry travelers begin by filling a pot with water and a stone and declare that they will make “stone soup.” The amused villagers watch in amazement as the travelers begin heating the pot to cook their “stone soup.” When one traveler acknowledges that a little salt is needed to give the soup taste, a villager agrees to provide just a pinch of salt to the cause. Next, the travelers ask for barley to give it texture, and another villager steps forward. Then, they ask for and receive carrots to give it flavor. Each villager figures they can part with just a small amount of food or seasoning since they don’t have to give everything in their larders. By the end of the story, the pot is filled with a delicious soup that all the travelers and the villagers enjoy.

This folk story about making something from simply a solid (“stone”) base illustrates how people can more easily be motivated to provide the valuable resource of their input and energy once they understand they don’t have to be responsible for the success or completion of the project as a whole. Once the villagers realized that they didn’t have to cook an entire meal for the travelers, just provide some carrots or seasoning, they were much more willing to step forward and offer input. In technical groups, the “stone” is typically called a straw man proposal. This is a proposal may be nearly complete except for corner cases or may be as basic as simply a framework for encouraging team input.

I find a straw-man proposal is quite effective in discussing technical problems. With a basic straw man proposal, the team’s work becomes that of finding corner cases and fleshing out the proposal, opposed to brainstorming a solution on the spot. Sometimes, the technical problem has multiple potential solutions and the team needs to choose between them. In this case, creating multiple straw man proposals and pros/cons lists can help the team better debate the merits of the proposals and possibly even create a new proposal which leverages the pros of the strongest proposals.

I have also found it effective to use a straw man proposal as a mechanism for making progress in a variety of other situations where I know I may not get all the feedback I want during a meeting. For example, if the team needs to brainstorm the agenda for our next face-to-face, I will typically lay the framework, placing in our regular discussion topics and our lunch and other breaks. After seeing the groundwork, the group is always able to fill in the open spaces with important topics I could not have come up with on my own.

6. Work within the system first, then propose change

As a newcomer to a volunteer army, it is essential to your success to understand the existing system and processes before you work to promote changes. It is challenging as a newcomer to come up to speed on the processes and procedures of the group you are joining, and there is a strong temptation (sometimes subconscious) to encourage the team you are joining to abandon their current processes in favor of a process that you are familiar with and that worked well in environments you were in previously. However, it is important to understand that the team you are joining, especially a team of volunteers, will have a high resistance to change unless they fully understand your proposal and agree that the pay offs are significant.

I encountered this situation when I was hired as the program manager for an open source project that had been running successfully before I came on board. I was able to implement some changes to the

process because I waited to implement them until I understood the full environment. For example, the team had solid processes in place for organizing and tracking the progress of their work, and I spent my first few months understanding and adopting these. However, I noticed that something was occurring in those months. A few weeks after I joined, I asked the question, "Did we finish milestone X?" And, a flurry of emails were sent so that the team could officially package up milestone X and announce that milestone X was completed. A month or so later, a few members of the team noticed that milestone X+1 had come and gone without any packaging or announcement. It was at this point that I realized that the processes that had been working well for the team in tracking progress were not working in officially releasing software. Typically, milestones were passed without enough fanfare for the team to even notice the passing. Because this was a need that the team agreed existed and because I had built up a rapport adopting the other processes of the team, I was able to add in the concept of release criteria and hold release readiness meetings before all subsequent milestones to ensure that the milestone criteria were met and the milestone was officially released and announced. Had I come in during my first few weeks and attempted to propose release criteria, I believe the team would have strongly resisted the "extra process and paperwork" I was adding. However, because I waited until I understood the team and the team trusted me, I was able to make this change more easily.

When you attempt to understand the environment you are entering first and then begin proposing changes, you are much more likely to achieve success. This fact is especially true in volunteer organizations, such as the open source project I was part of, because the group is already together because of a common passion and a pride in the work they do. I have been in volunteer organizations where a newcomer will come in with feedback on how to "fix" our "broken processes" and "solve" all of our problems without really understanding our true problems or processes needing support. When this happens, the newcomer typically finds resistance much stronger than they expected and, without an understanding leader who follows the tips above, the newcomer may eventually leave in frustration.

7. Focus on the need, not the name.

Once you understand the system and documented or undocumented processes that your volunteer army is following and are interested in motivating the group to change, it is useful to focus on the need that your idea is solving, opposed to focusing on championing a specific technique or process you knew worked in a previous organization, especially a previous non-volunteer organization. Especially in volunteer armies that pride themselves on being volunteer armies, such as open source projects, attempting to bring in process and procedures, and especially jargon, from non-volunteer armies, such as closed source projects, can backfire.

When I was a newcomer to an open source project, I was actually asked to do exactly what I just advocated not doing. I was asked to create a "Product Requirements Document" by some members of the team, despite the fact that formal "Product Requirements Documents" are typically not created in open source projects.

Initially, I followed this advice and began creating a Product Requirements Document using the template that the team believed would be valuable. However, it soon became clear that I was creating a document that the team had no interest in using or reviewing, even though they had asked me to create the document. I would try to schedule reviews of the document, and the team would ignore my requests, or the team would attend and take action items to help me improve the document and then would not follow up. So, on the third unsuccessful attempt at a review, I probed the team deeper on their true need. I wanted to know what they really wanted me to do because a "Product Requirements Document" didn't seem to be the answer. At that time, I found out that what the team really wanted was to show their maturity as a project to certain managers more typically involved in closed-source projects. In closed-source projects, having a "Product Requirements Document" was essential to success.

However, fortunately for me, the managers making this request also understood the difference between a name and a need. On further investigation, I found that a "Product Requirements Document" was not what they required. They wanted to *ensure that the open source project had a solid process in place for*

collecting, prioritizing, implementing, and testing requirements. The further good news was that the team had quite a bit of this process in place already. My job became a matter of finding a way to ensure this process was clearly communicated to all members of the team and their stakeholders. We held a meeting to discuss the true needs, and, from that point on, my job became one of publicizing and clarifying the existing web-based process in place for collecting, prioritizing, implementing, and testing requirements. Once the managers that were part of the project understood the well-defined method for requirements management, the requests for a “Product Requirements Document” went away.

Focusing on the true need allowed us to make progress in this instance. I have found this technique effective in many situations in volunteer armies. For example:

- 1) Instead of asking the team where their schedule and Gantt chart is, I like to ask “How do we know what we will deliver and by when?” A milestone list on a PPT or web page may be all that is needed to answer that question.
- 2) Instead of asking the team where our formal “Release Criteria” are documented, I like to ask “How do we know when we will be done?” This is especially effective in volunteer armies. It’s empowering to change the answer from “When we are all burnt out.” to “When we have accomplished these specific items.” Knowing what you are responsible for up front can actually reduce burn out. It also can help change the other answer seen in volunteer armies, especially standards organizations, of “When every single concern with our work product is completely gone.” to something more realistic and satisfying to all members.
- 3) Instead of asking the team who will do QA, I like to ask “How do we know that what we are delivering is good enough?” A volunteer army frequently can’t afford an official QA team, but they typically can find the time to ensure they have enough volunteers to provide a reasonable degree of confidence that their end work product is of the level of quality the group desires.

8. Expect respect from the team

In addition to respecting the volunteer army (see section 2), it is also important as a volunteer to expect (and receive) respect from the volunteer army you are working with. Because volunteer groups don’t have as many formal rules governing behavior and roles as corporations, it is frequently easier for one person or group of people to dominate the conversation or the agenda. In some instances, this may be welcome. For example, in a standards organization or open source project, the volunteers may all agree on the goal but may not all have the time to dedicate towards accomplishing the goal. They may welcome a few individuals or companies that step up to assist the group in making progress. However, for the situations where the dominance of a few is not welcome, it is important to remember that you as a volunteer should be getting the same respect you give your peers. In these situations, the following techniques may be effective:

- 1) Jump in. As noted in section 4, sometimes jumping in is also a way to share the workload with others that may be taking on more of the work or expressing the majority of the opinions. Occasionally, this is all that is needed for the person or persons that are taking more of the limelight to step aside and let others in.
- 2) Confront the issue. Unlike in business situations where there are clear escalation paths and structures for reporting and confronting issues, in volunteer armies you may need to confront the issue with the offending person(s) yourself or together with a few like-minded allies.
- 3) Tolerate the situation. As noted above, in some cases, a volunteer army may be delivering a quality product even though a few individuals are running the show. If the goals you came to the volunteer army to achieve are being met in general, you may find it acceptable to tolerate the situation because you are achieving the results you wanted to achieve.
- 4) Leave the volunteer army. People come and go from volunteer armies all the time, and it is easier to leave a position or group where your livelihood and families’ financial security does not depend on it. If you find that the situation is such that you cannot simply tolerate it and you are unable to change things by addressing the situation or asserting yourself more, you may find that it is time for you to move on. There are many volunteer armies out there, quite a few of which have similar missions and visions, so it may be time to find a new avenue to express your values and talents.

9. In Summary

Achieving quality in volunteer armies requires patience, persistence, and dedication. However, it is possible to motivate and influence volunteer armies to deliver high quality results. Perhaps the most important lesson to remember is to respect your volunteer army and expect respect yourself and then all other approaches will fall into place. When you expect respect, you are willing to jump in to accomplish goals, and when you respect the team you are excited and eager to solicit their opinions via straw man proposals and lists of tasks needing volunteers. You are also willing to work within the system before suggesting change, and, when you do suggest change, you focus on the true need, not just the terminology or processes you have used before in other positions or organizations. When you do all of these things, you can begin to feel the satisfaction and accomplishment that comes from delivering quality in a volunteer army.

References

ⁱ Eric S. Raymond. "The Cathedral and the Bazaar."

<http://www.catb.org/~esr/writings/homesteading/cathedral-bazaar/ar01s11.html>

ⁱⁱ Wikipedia. "Stone Soup" entry. http://en.wikipedia.org/wiki/Stone_soup.