

A Tailored Participatory Action Research for FOSS Communities

Adam Alami · Peter Axel Nielsen · Andrzej Wąsowski

Received: date / Accepted: date

Abstract Participatory Action Research (PAR) is an established method to implement change in organizations. However, it cannot be applied in the open source (FOSS) communities, without adaptation to their particularities, especially to the specific control mechanisms developed in FOSS. FOSS communities are self-managed, and rely on consensus to reach decisions. This study proposes a PAR framework specifically tailored to FOSS communities. We successfully applied the framework to implement a set of quality assurance interventions in the Robot Operating System community. The framework we proposed is composed of three components, interventions design, democratization, and execution. We believe that this process will work for other FOSS communities too. We have learned that changing a particular aspect of a FOSS community is arduous. To achieve success the change must rally the community around it for support and attract motivated volunteers to implement the interventions.

Keywords Participatory Action Research · FOSS · Change implementation

1 Introduction

Many FOSS (Free and Open Source Software) projects have matured over the years to produce software of considerable size and complexity. Some have seen generational changes. The need to rediscover, refactor, and re-engineer existing code bases will only increase over time [9], as will the need to deal with technological changes, processes, infrastructure, dependencies, and deployment platforms. We can safely assume that handling growth and maturity will require an introduction and intensification of the best practices, processes, methods and tools of software engineering. However, our understanding of how to change a FOSS community is limited.

The pragmatic objective of the work described here is to improve the Robot Operating System (ROS) community quality assurance (QA) practices. Through the mechanisms of a participatory action research process, we intend to introduce, and when necessary, reinvigorate best QA practices in the ROS community. We want to align the ROS QA with practices of other FOSS communities practices and with modern software engineering expectations. This will simultaneously yield a change in the affected community and new method to work with FOSS communities.

How does a FOSS community carry out change implementation? As communities increase in maturity and size, they will adopt more formal decision making processes [15]. An example was the introduction of voting in Apache web server community. Communities also establish meta-processes for creating

A. Alami
Rued Langgaards Vej 7
DK-2300 Copenhagen S Denmark
E-mail: adaa@itu.dk

P. A. Nielsen
Aalborg University Selma Lagerlöfs Vej 300
9220 Aalborg Denmark
E-mail: pan@cs.aau.dk

A. Wąsowski
Rued Langgaards Vej 7
DK-2300 Copenhagen S Denmark
E-mail: wasowski@itu.dk

formal structures such as the steering board in Gnome. However, not all FOSS communities have a governance mechanism in place. Even if they do, it is not understood what role these governance structures play in introducing change to the community. Our subject of study, the ROS community (see Section 5) is self-managed. Recently, the community has experienced a significant growth in its commercial participation including significant interest and investment from Amazon, Intel, Bosch, and Toyota. Many more companies are associated in the ROS-Industrial Consortium.¹ These participants have become outspoken and brought their own requirements for change. We were approached by the ROS-Industrial Consortium to execute a quality assurance enhancement project in the ROS community. This paper shares the experience of two years of collaboration in this endeavor.

To implement this change, we needed a process and a methodological framework. We opted for the out of the box version of participatory action research (PAR). PAR is an appropriate choice given that action research has proven to be a reliable tool to institute change in organizations. However, we quickly realized that the PAR framework is not in tune with the values and norms held in the ROS community. It had to be adapted. Thus, we set off to address the following research question:

RQ: How to adapt the participatory action research method to carry out change in a FOSS community?

We designed PAR4FOSS a participatory action research process that caters for the cultural traits of the FOSS movement:

- We choose to democratize and gather a consensus around our actions via consultation and dissemination in the community.
- The change process mandates transparency. Transparency is reflected in the dissemination of decisions and solicitation of feedback from the community.
- The change process is self-managed. Although the process necessitates a facilitator, the structure is flat and relationships are collaborative.

The contribution of this work is twofold:

- We propose, PAR4FOS a tailored PAR method fit for FOSS communities cultural and social particularities.
- We implemented and executed PAR4FOSS in the ROS community. We share the result of this implementation and the results in this paper.

The intended audience of this work is action researchers, community managers and maintainers who plan to introduce change to a FOSS community or other similar self-managed teams and organizations. We hope that they can leverage the method to introduce change to other communities.

We start the paper by providing a background on participatory action research and FOSS communities (Section 2). In Section 3 we introduce PAR4FOSS, a tailored participatory action research for FOSS communities. In Section 4, we highlight the research design. Section 5 is dedicated to the ROS community and a discussion of its cultural fabric. In Section 6, we describe the process that led to the design of our method. In Section 7, we evaluate the proposed method and we discuss its strengths and weaknesses. We discuss the lessons learnt in Section 8. We conclude in section 9.

2 Background

We begin by introducing the two main concepts driving this paper: The Free and Open Source Software (FOSS) and the Participatory Action Research (PAR).

2.1 Free and Open Source Software (FOSS)

Open source project refers to any software made public and open for others to modify. Bretthauer defines Open Source as “software which is licensed to guarantee free access to the programming behind the pre-compiled binary, otherwise called the source code” [7]. Open Source and source disclosure has always co-existed as a software licensing agreement, until mid-80s. The Internet removed the physical barriers and facilitated virtual collaboration accelerating software distribution. The traditional collocated software development process is no longer the only option. In mid-90s open source has become a recognizable software development and distribution model [2, 8].

¹ <https://rosindustrial.org/ric/current-members/>

In the upcoming subsections, we give an overview of what FOSS is about, and what is the social structure of FOSS communities, who are the contributors of FOSS, and what is the predominant culture of FOSS.

Organizational Structure: The structure of each FOSS community is unique, reflecting its principles, values, beliefs, and norms. The organization of roles in a FOSS community have been described as an onion, with concentric layers, rather than a vertical hierarchy. the core of developers are at the center. The layers consist of the informal community managers, project managers, other developers, all the way to passive users making up the outer layer. Individuals generally move inward through merit, nominated by another member of the community, based on code contributions, facilitating others in their work, mediating conflicts, and solving problems in the community [23].

FOSS Demographics and Motivation: Well-elaborated surveys report that the FOSS population consists of predominately young males, with educational or occupational background in computing, software engineering and programming. The image of the population demographic is consistent across various sampling methods and studies. Around 98% of FOSS contributors are aged between 20 and 30 years [10, 17, 18].

Many members join a FOSS community motivated by the desire to control their work. The “free choice” of assignments is important to members of a FOSS community. The members identify with the work they produce, and it is a source of pride for them. Being able to pursue one’s own passions and interests is a great advantage for FOSS contributors [14]. The underlying motivation, to participate in FOSS are either intrinsic or extrinsic [6, 21, 29]. Intrinsic motivation refers to behavior that is driven by internal rewards [38]. The participation is based on internal satisfaction and self-enjoyment [29]. Ryan and Deci [38] explain that the motivation to engage in an intrinsic behavior arises from within the individual because it is naturally satisfying. This contrasts with extrinsic motivation, which involves engaging in a behavior in order to earn external rewards [29, 39]. This type of motivation arises from outside the individual. It can involve tangible or psychological rewards. Psychological forms of extrinsic motivation can include praise and public acclaim [39].

Meritocracy: FOSS communities are meritocratic. People are valued for the quality and quantity of their contributions [16, 27]. The community members’ self-image includes the desire for “geek fame”, a desire to build trust and reputation within the community, generosity of time, expertise, and the desire to create reliable and quality software [40].

In a social arrangement where meritocratic filtering is a norm, people with higher merit are placed into positions of higher authority [44]. Sociologists argue that meritocratic systems encourage the most talented into the most functionally important positions [44]. Open source communities are almost entirely meritocratic. Reputation in an open source community is gained by valuable and innovative contributions to the community. Meritocracy is a complex mechanism of control and structure.

Scacchi [42] defined this social community conduct as “interlinked layered meritocracies operating as a dynamically organized but loosely coupled virtual enterprise.” Our societies and communities rely on social orders, or the links found among institutions, traditions, values and morals that work cooperatively to keep societies moving forward instead of falling apart [44]. Scacchi’s [42] definition resembles a social order. He explains that “layered meritocracy” is a system to control, structure and oversee the development activities. It implies certain form of authority legitimized by the gained reputation, experience and accomplishment in the community [42, 43]. There is also an implicit hierarchy where core developers are ranked higher. However, this social order is not authoritarian [41]. Consent is sought among core developers and contributors in the decision-making process [42, 43]. Leadership, in FOSS communities, is earned through experience. Core developers are usually the early contributors to the community and some of them are the founders.

2.2 Participatory Action Research

Participatory action research (PAR) [3, 4, 30, 32, 46] is a set of consecutive actions and reflections [25], forming a collaborative approach to research whereby the researcher partners with the subject being studied to achieve a particular outcome. The environment where the research occurs has a great influence on the course of action. In contrast to conventional methodologies, participatory research acknowledges the “local” perspective and knowledge.

Action research becomes participatory when the researchers and participants are engaged collaboratively in the inquiry [25]. Participatory action research is defined by three reiterative phases: inquiry, action, and reflection [25]. The iterative nature of the process provides opportunities for improvements to the knowledge, methods, and better understanding of the subject of the inquiry. Kemmis and McTaggart [24] argue that the stages may amalgamate in practice as the learning experience of the inquiry progresses. PAR differs from action research by its principles of democratic participation and communal reflection. The power is distributed equally between the researchers and the participants. Participants are actively engaged in the research process. Communal reflection provides an avenue to collectively address any bias or pre-established assumptions that the researchers may have [25].

Participatory action research includes analysis of what people do, how they interact with the environment, what they mean in their language, and what they value. PAR elucidates their understanding and interpretation of the world. Understanding material, social, and historical circumstances that people create and then recreate in social settings allows generation of ideas of changes that can address their problem. Participatory action research helps people change their practices, their understanding of these practices, and the situations in which people live and work by self-reflection. Foundations of participatory action are the following [25]:

- PAR is a social process that explores the realms of the individual and the social.
- PAR is participatory, it engages people in examining their knowledge.
- PAR is practical and collaborative.
- PAR is emancipatory, helping to release participants from irrational or unproductive constraints.
- PAR is critical in that it allows to contest and reconstitute unproductive, unjust, or unsatisfying ways of interpreting the world.
- PAR is reflective, it allows people to investigate reality in order to change it.
- PAR attempts to transform both theory and practice.

In Participatory Action Research, the control lies, not with the researcher or the participants, but is shared, in the form of a zig-zag pathway with greater or less participation at various stages [24].

Why Participatory Action Research? To address quality assurance with a change, we need a method and a process that fully engage the FOSS community. The quality problem originates in the community itself. It is defined, analyzed and solved by the community and the researcher together. PAR involves the full and active participation of the community in the entire research process [5, 19].

In the context of FOSS, the strengths of action research are its capacity to provide real world application, to create change and test findings, to research interventions that are specific and targeted, and to create opportunities for stakeholder engagement. The success of the outcomes depends on the degree of negotiated access to the project setting, clear role outlines, commitment of time building relationships, sensitivity toward insiders, and the scope of the project allowing for flexibility [26, 31].

To cement our argument on the suitability of PAR for our case, we discuss several relevant and successful PAR projects executed by other researchers in other contexts:

- Dedding et al. [11] worked on identifying solutions for the problem of digital inequality in the city of Amsterdam. The researchers joined with volunteers, citizens, and education experts to determine solutions for the problem. The study included interviews with parents and focus groups that discussed the problem and sought solutions. The groups met once a week, coming up with a plan for the problem's solution and continuing to meet to update progress. The result of this study was improved digital access for all citizens of Amsterdam by following the plan developed by the group. Like in the case of a FOSS community, there was a clearly identified group of stakeholders that suffered problems with involvement into IT-related processes.
- Donnelly and coauthors [13] established a venture between a university in UK and an analytics company analyzing brand and retailer data, helping small businesses to use customer's data at the basis of business decisions. This partnership provided the basis for three longitudinal participatory action research studies of a mechanism for the small businesses to access shopper insight from Tesco Clubcard. The university and the company examined marketing strategies, their use, and their success. Clients, small business managers and owners, were included in the study through semi-structured interviews and focus group meetings. The finding was that a loyalty card was a successful way to improve marketing efforts for small businesses, and a loyalty card was established by the participants and researchers' efforts. The relevant aspect of this study, pertinent to FOSS, was that many participants in the improvement process partake on a voluntary basis. Like FOSS developers, small businesses, and customers do not have any obligation to join the program.

- In another study, a community center formed a group of elderly persons who wanted to collaborate with the project “Age-friendly Gothenburg” in establishing a design plan for the local environment. The task was to assess and describe the local environment and highlight areas in need of improvement with the use of life filming. The assessment of the local environment was focused on outdoor spaces, buildings, streets, parks, transportation, and information. The findings were compiled in five themes which were later included in the new design for the environment that emerged as a result of this study [20]. Quality assurance in software engineering invariably include some forms of monitoring individuals. A successful project involving an age group naturally critical of monitoring, convinces us that PAR might pave the way for FOSS members to allow themselves to be subject to more stringent QA practices.
- Rajaram [35] reports a participatory action research project that had the goal of collecting data on the awareness of lead poisoning and disseminating information on lead poisoning prevention in Omaha, Nebraska. The purpose was to understand the social and cultural factors influencing health and illness behavior among underserved communities in collaboration with the Lead Safe Omaha Coalition (LSOC), a community-based, nonprofit, grass-roots organization working together to end childhood lead poisoning in Omaha. The researches included members of the community and LSOC to ensure that the results of the study were relevant and beneficial to the community. The research team had to make compromises along the way. One of these was to make researchers welcome in the community, and the way this was done was allowing researchers to be accompanied on their door-to-door visits by LSOC staff or community volunteers. The results of the study were that trust and relationships were built between researchers and community members, including LSOC members. Awareness of the dangers of lead poison was achieved [35]. We know from prior studies [36, 45] that reputation and trust are important factors influencing success of contribution to FOSS. The study suggest that PAR is a framework which may allow to establish the necessary trust to create change in the community.

These examples illustrate the ability of PAR to successfully create impact on communities. They all brought social change to their subject communities. Similarly, we would like to create an impact on the ROS community by elevating its quality practices to higher standards.

2.3 Related Work

Change of FOSS communities has, received little attention from researchers, with the exception of Krafft’s [28] and Özbek’s [34] PhD dissertations. While Krafft looks at influencing factors on innovation adoption in the Debian community, Özbek [34] focuses on concepts related to the introduction of innovations to FOSS communities. Although these studies are the closest to our work, both do not propose a working method, but rather demonstrate the lack of a specific method dedicated to change of FOSS communities.

Krafft followed the Delphi method. In the Debian community, innovations are built on consensus among the community. First, pioneers work on innovation, always ready to improve the tools and products. Pioneering usually begins inside the project or is carried into the project by a member who picked up the concept externally in order to address a need [28]. Krafft recommends as many as 15 factors affecting adoption of technology and tools in the Debian community. The three most relevant to our work are: (1) Sedimentation: awareness of available tools, (2) Marketing: active promotion of tools, and (3) Peercolation: acceptance by peers [28]. These factors can inform a change implementation process, but do not depict a working method to implement change.

Özbek studied how innovation is introduced to FOSS communities [34]. Proprietary software organizations use pain points and monetary incentives to encourage change, but since FOSS are voluntary organizations, finding ways to encourage change is important. An innovation is a tool, a method or a practice which can become part of a software process created by volunteers who are distributed geographically. Using grounded theory, Özbek analyzed 83 episodes of innovation in 13 FOSS projects; 37 failures, 30 successes, and 16 with an unknown outcome. Success is defined as an innovation used on a routine basis that solved a problem and attained its goal. When innovations are introduced, the result can be no adoption, no goal attainment, and failure or postponement in lieu of adoption. Partial migrations occur as well, which is when an existing innovation is replaced by a newly introduced one [34]. Our work is a natural extension of Özbek’s work. We aim not only to study change in FOSS, but we want to systematically introduce change into how communities operate.

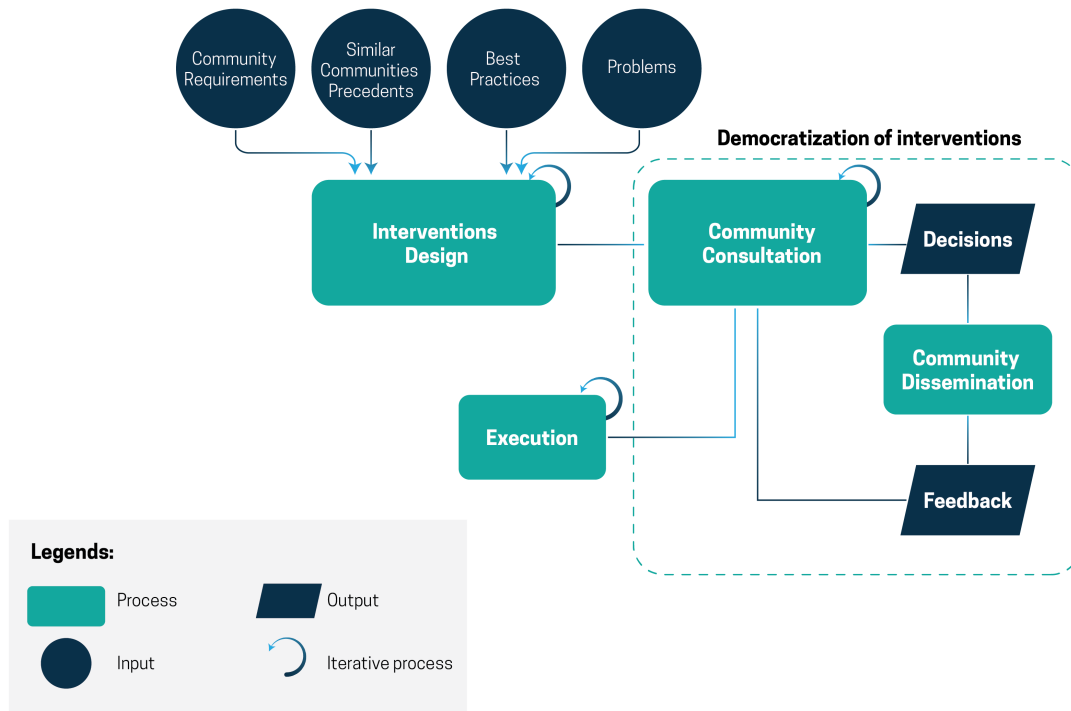


Fig. 1 The PAR4FOSS Method

3 Participatory Action Research for FOSS (PAR4FOSS)

Conscious of the FOSS communities uniqueness, we designed and executed a participatory action research process tailored for FOSS. The method (Fig. 1) we propose is composed of three components: interventions design, democratization and execution. The method (**PAR4FOSS**) is an iterative process with three activities where transparency and open decisions were implemented. The structure is influenced by the self-management and collaborative aspects of the FOSS communities. We believe that this process will be amenable to other FOSS communities and will thus help introducing change in other FOSS projects.

In PAR4FOSS, interventions are designed and enhanced iteratively. They are obtained from four different sources:

- *Community requirements*: legitimate needs to improve a situation or meet certain objectives.
- *Precedents*: something that other FOSS communities succeeded at doing and achieved a positive impact.
- *Best practices*: industry practices turned into community practices to align the community better with industry standards.
- *Problems*: specific community problems related to the scope of the change, the difference between what is happening (as is), and what should be happening (should be).

The democratization of interventions serves to legitimize them. As shown in Fig. 1, the legitimization is gained from community consultation and dissemination. Community consultation is a process by which the community's inputs on the interventions are sought. It aims at gathering local knowledge to test the interventions for cultural sensitivity, social effectiveness and appropriateness. The output of this activity is a set of decisions: tuned interventions and an action plan for each intervention. This consultation involves a consultative body, selected community advocates, enthusiasts for the change and experts on the subject matter. To ensure transparency, a wider community dissemination is recommended to gather feedback, consensus and support for the interventions. When feedback is received from the community, there is an obligation to act upon it. Feedback is communicated to the consultative body for deliberation and an iteration of interventions enhancement.

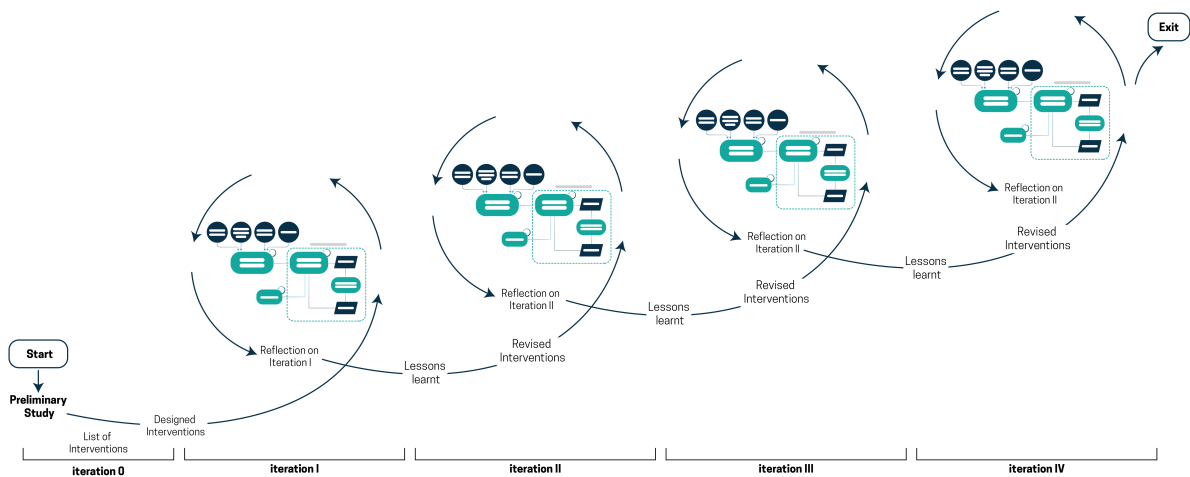


Fig. 2 PAR4FOSS Iterations

Eventually, the interventions stabilize and become candidates for execution. The execution should leverage existing community capabilities to implement the interventions. But, the execution is not the end point. Eventually, new learning emerges from execution of an activity that should be shared with the consultative body for knowledge acquisition and future considerations.

The PAR research is designed to be executed iteratively. The PAR4FOSS iterations consist of a set of aggregated interventions to implement change in the FOSS community. We propose a PAR process (Fig. 2) in two consecutive phases: (1) the pre-interventions phase and (2) the interventions phase. We have ourselves worked with four iterations [3, 4].

Pre-interventions phase (iteration zero in Fig. 2): We begin with an empirical investigation of the research environment [12], the identification of problems and issues related to the proposed change. This is an exploratory phase which aims to discover the scope of interventions, the problem space, and the relationship between these two components. During this phase, the design of interventions takes place.

Interventions implementation phase: The interventions will be executed in iterations (four iterations in Fig. 2). Each iteration is the execution of PAR4FOSS method. Upon the completion of each iteration, we conduct a reflection session to draw learned lessons and revise the list of interventions.

4 Application of PAR4FOSS in ROS community

4.1 The Project Context

ROS managed to attract a wide global community of users and contributors. ROS-Industrial is a branch of ROS with a specific industrial application focus. Incepted in 2012, ROS-Industrial Consortium has secured the collaboration of key players in the robotics industry (ABB, Yaskawa, Siemens, John Deere, BMW, Bosch, etc.) ROS-Industrial ambition is to become the worldwide open-source standard for industrial robots. ROS-Industrial stakeholders commenced raising their concerns regarding the quality assurance practice in ROS not being aligned with FOSS communities and software engineering best practices. Although, we were originally external to the community, the motivation for the project came from within the community. Thanks to the push from the ROS-Industrial Consortium, the H2020 project ROSIN was established to enhance ROS quality assurance practices and to promote ROS as a reliable robotic platform for industrial users. This work is one of the results of this community inspired project.

Access to the research setting. During the inception of the project, we contacted the ROS core team and we entered a dialogue regarding the idea of the project. The core team assigned one of its members to the project and we recruited another contributor with over ten years experience in the community. Both have distinguished reputations in the community. We asked our informants to lead the recruitment of participants for a quality assurance working group. The aim was to leverage our informants' meritocratic status in the community for credibility. This approach worked well.

Our proposed PAR4FOSS (Fig. 1) advocates community consultation for change introduction. We contextualized this guideline in the ROS community by instituting a *community working group*. The group’s mission is to collaboratively promote an extended array of quality assurance best practices in the ROS community, so that it may endeavor to carefully construct a cultural environment where quality is part of the fabric of the community. The scope of the group work is to promote and implement quality assurance initiatives, for example testing, code reviewing, continuous integration, and code scanning. The group became the problem owner and the project conduit in the community.

Defining the problem. The inquiry stage of PAR is the identification of a shared practical problem by collaborators, and methods to address the problem [24]. We conducted a qualitative study [1] during the inquiry phase, to understand the problem and the underlining causes. We present a summary of this study in Section 5.1.

Interventions. During the inquiry phase we identified the problems and proposed solutions. Each solution is an intervention. Community intervention encompasses the creation, implementation, and evaluation of best practices designed to affect the community as a whole. Table 2 lists the interventions. The list of interventions was presented to the working group for review and validation. A final list was agreed upon and published in the QA forum for a wider community consultation.

4.2 Participatory Action Research in ROS

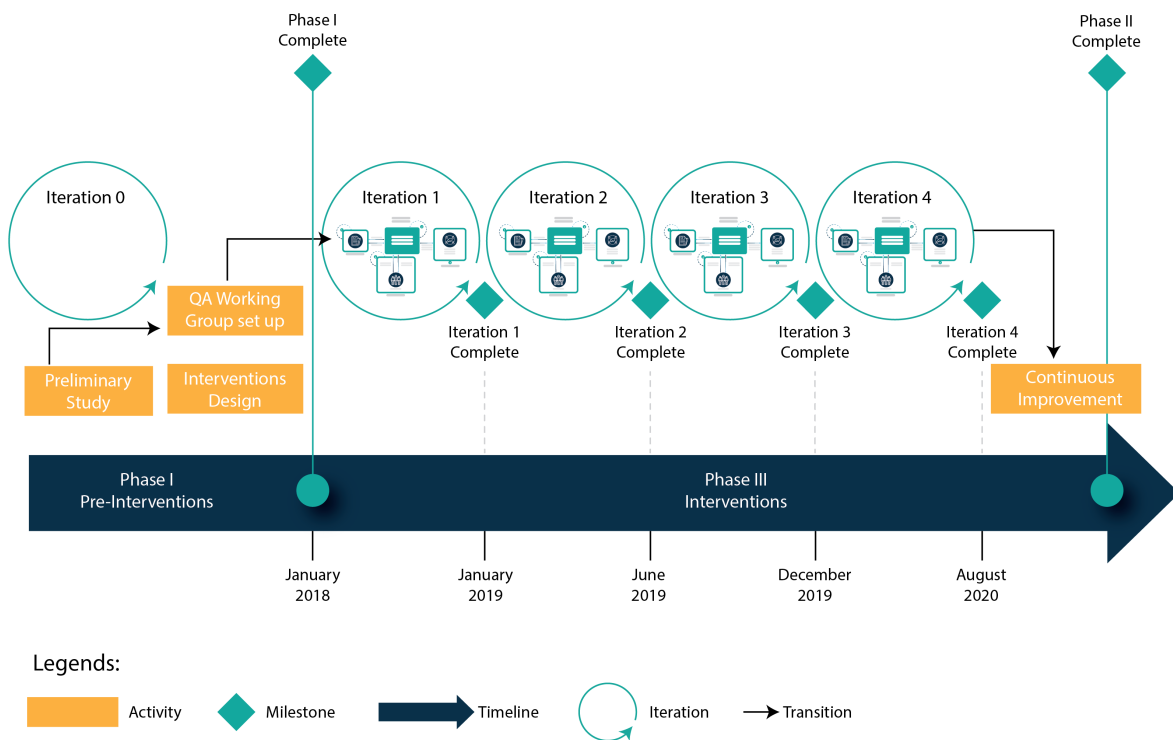


Fig. 3 ROS Community Participatory Action Research Time Line and Iterations

The PAR process (Fig. 3) typically includes two broad steps. First, research implies the participation of the community in the process of elucidating and analyzing the problems (Phase I) [25]. This process resulted in new understanding. The second phase of the research process is cyclic (Phase II). It is implemented through iterations of planning, acting, observing, and reflecting [25, 46]. We were unable to execute this phase as prescribed in the literature. In order to succeed in the community, we had to design a change process tailored for the ROS community that still embraces the principles of PAR. Section 6 presents the process we deployed to execute the implementation of interventions.

Phase I — Pre-Interventions: The process began with the identification of QA problems and issues. The objective was to discover the scope of interventions, the problem space, and the relationship between these. This was an empirical investigation of the current environment [12]. The aim was to understand the cultural characteristics of the community and the quality practices taking place in the community. This phase of the study identified the current challenges in implementing quality processes, methods, and practices in the community (see Section 5).

ROS Quality Assurance Working Group: PAR advocates for active engagement of the community [25, 46]. Participation is a democratic principle to overcome the researchers' dominance in the process and build on the knowledge of the community necessary to implement the interventions.

To this end, we established a quality assurance working group. The aim of the group is to serve as a conduit in the community for the change initiative. Its role is consultative. The first step to gain community consent for the interventions. It has been decided to implement this group as an implementation tool as it is a community cultural norm to reach consensus regarding changes in the community. The group had 23 active members. The group has been meeting monthly and has accumulated a total of 18 meetings. Minutes of each meeting were taken and published in the community forum ².

Interventions Design: Intervention comes from the Latin *intervenire*, meaning “to come between, interrupt.” The intention of an intervention is to make things better or to remediate a situation. In the context of this research, we define an intervention as a combination of action elements or strategies designed to produce behavior changes and improve quality assurance practice in the ROS community. Interventions may include educational programs, new or enhanced practices, improvements to the processes, or a quality promotion campaign. This research interventions create change by:

1. Influencing individuals' knowledge, attitudes, and beliefs.
2. Increasing QA infrastructure and tools.
3. Creating supportive environments for QA practices and resources.

Interventions (see Section 6) were designed based on the findings of the phase I analysis. These interventions have been put forward to the working group for the community's deliberation and advice.

Phase II — Implementation of Interventions: The interventions have been implemented iteratively. As shown in Fig. 3, four iterative cycles have been proposed. The scope of each cycle is dictated by the interventions' priorities. The priorities have been assigned by the community working group. Members were asked to assign high, medium, or low priority for each intervention. The interventions with the highest number of high votes were assigned to the early iterations' cycles. Subsequent cycles' scope is interventions that received lesser votes for high than for medium and low priorities.

To carry out the implementation of the interventions we faced a conundrum. We had to reflect and design a process that caters for the community cultural and social specifics without disturbing the community ecosystem (see Section 6 for the proposed method).

4.3 Data Collection and Analysis

This is a qualitative research using one subject community, where data is gathered using a combination of techniques, a community working group, interviews, and participatory activities.

Data Collection. The empirical qualitative data has been captured using several techniques that were selected based on the combination of the methods and the nature of the data collected. Table 1 maps methods to their respective data collection techniques.

Data Analysis. We used thematic coding, grounded in the interpretative philosophy. Following the guidelines of Robson & McCartan [37] and Miles, et al. [33]. The aim is to examine the meaning and symbolic content of qualitative data. The process involves the identification of themes and coding. We used line-by-line coding of the data, identifying meaningful parts of the text and assigning labels to them that indicate they are the thematic idea. This labeling or coding process enabled us to retrieve and aggregate together all the text that has been linked with a particular thematic idea. The collation of themes and association with text unveil patterns of thoughts and ideas that contribute to the understanding and later was used to formulate theories about the ROS community.

² <https://discourse.ros.org/c/quality>

Methods	Data Collection Technique	Data Analysis	Key contributions
Phase I:			
Interviews & Participatory Activities	<ol style="list-style-type: none"> Ten interviews have been recorded and transcribed. One researcher attended 29 community events (i.e. Conferences, meetings, etc.). Notes were taking during these events and field notes were compiled, thereafter. A total of 30 field notes have been accumulated. 	Thematic Coding	<ol style="list-style-type: none"> A cultural understanding of the ROS community. Definition of the problem space (see section 5.1)
Phase II:			
Participatory Activities	<ol style="list-style-type: none"> Field notes have been used to capture observations and interaction with the community. This applies to both real-life and virtual activities. Emails, and other forms of interaction with the community (i.e. Forums discussions) are captured in field notes. Working group meetings and workshops have been recorded and transcribed. At this stage of the project, 18 meetings for the working group have been conducted. In addition, other ongoing meetings with volunteers working on the implementation of interventions have taken place. 	Thematic Coding	<ol style="list-style-type: none"> Successful implementation of two iterations of QA interventions. A change method tailored for the ROS community.

Table 1 Data Collection and Analysis Methods

5 The ROS Community

In 2000, Stanford University began working toward integration of embodied artificial intelligence (AI), such as the Stanford AI Robot and the Personal Robots Program. They created flexible, dynamic software systems for robotic use. In 2007, Willow Garage, a visionary robotics incubator started by Scott Hassan to accelerate the advancement of non-military robotics, gave significant resources to extend and create implementations. Many researchers from various institutions and labs began to contribute time and resources to core Robot Operating System (ROS) ideas and software, all in the open BSD license. Willow Garage became a for-profit company, and ROS is maintained by the Open Source Robotics Foundation (OSRF). Over the years, the developed model has been seen as one of the strengths of ROS. This model allows any group to start their own ROS code repository that they maintain control and ownership of, and if they make their code public, they can receive recognition and credit, and they benefit from the open source software projects.

ROS is an open-source meta-operating system providing a flexible framework for writing robot software. It is a collection of tools and libraries that help to simplify creation of robotic platforms. The collaboration of researchers enables robotic products to be more robust. ROS provides a communications infrastructure as middleware. It offers asynchronous message passing, recording and playback of messages, request and response remote procedures, and a distributed parameter system. In addition to middleware communication, ROS provides robot specific features, such as standard message definitions, a robot geometry library, robot description language, diagnostics, pose estimation, localization, mapping, and navigation. The ROS toolset is one of its strongest features, and this toolset enables debugging, plotting, and visualizing the state of the system with rviz and rqt tools.

All together, over 3,000 ROS packages are available publically. These packages cover everything from proof of concept implementations to new algorithms for industrial quality drivers and capabilities. The ROS community has over 1,500 participants on the mailing list and 3,300 users on the Q&A forum, with 22,000 Wiki pages and over 30 edits per day.

Recently the community has embarked in an overhaul project called ROS 2 to re-engineer and re-architecture the current software. ROS 2 is under heavy development and attracted the participation of some heavy industry players (e.g. Amazon, Intel). The project has become the focal focus of the community. This enthusiasm had some consequences on ROS. The community attention and effort is shifted away from the current software to ROS 2.

5.1 The State of Quality Assurance in ROS

The ROS community is large and diverse, and it is different from other FOSS communities, as most ROS developers are not software engineers. The ROS community has awareness of quality, and has implemented a well-defined development process, defects management processes and tools, code review, continuous integration, unit testing, and knowledge sharing. However, the efforts for quality are constrained by participation motives, community priorities, meritocratic culture, sustainability, complexity, and adaptability [1].

Most robots are complex distributed systems, encompassing many fields of engineering. Thus, it is critical to implement a successful quality assurance management strategy. The scope of our engagement in the ROS community has been focused in creating sustainable QA practices.

Before this project, ROS was not without QA practices. It had a continuous integration infrastructure in place, an issue tracker and process of managing defects. However, they did not live up to the expectations of industrial participants. We do not imply that ROS is low quality. The collective intelligence of the community has produced a remarkable software platform that powers various industrial applications. Even though the popularity of ROS testifies for its quality, industrial stakeholders have been calling for visible and concrete community practices to raise the confidence in ROS even further. Hence the initiative of this project which is to bring ROS QA inline with other FOSS communities and software engineering best practices.

Once we became accustomed with the ROS community, we realized that its cultural fabric has to be accommodated. We needed a method because it was important to us to implement the interventions without disturbance. Executing the intervention blindly would have been naive, likely clash with the community and possibly failing.

5.2 ROS Community Cultural Traits

The ROS community over the years embraced the FOSS values, believes and norms. The ROS community is a meritocratic social system. It values transparency, freedom of choice, collaboration, self-management and innovation. We summarize its main cultural traits based on one year and a half worth of participatory observation in the community [1].

Meritocracy. Meritocracy is a social system in which success and status depend primarily on individual talents, abilities, and effort. The ROS community is a highly meritocratic system. Meritocracy is implemented via two mechanisms: the “Karma” system and reputation. The “Karma” system is a virtual rating system, where points are allocated to contributors by other community members for helping answering and resolving technical questions and challenges. The “Karma” points showcase the contributor’s technical expertise and her engagement in the community. The more “Karma” points, a contributor has, the more important is her status in the community. Contributors also have an additional “reputation” outside the “Karma” system. While the “Karma” system is an indication of the technical expertise and knowledge, reputation is the community’s believes and opinions held about a member. Contributors are conscious of their reputation in the community and they thrive to maintain it.

In an early meeting with the community to set up the project infrastructure, a community member warned us: *“This is a meritocratic community. Anything you do must take in consideration this community aspect.”* In a community meeting, the attendees stopped and congratulated a member of the community with high “Karma” points. One attendee said: *“Congratulations man! 40 thousand karmas. That’s impressive.”*

The challenge that ROS meritocracy poses to participatory action research projects is that of the researcher being new to the community and without “reputation” or “Karma”. Our strategy was to enter into partnerships with community members enthusiastic about quality and with distinguished reputation in the community. The partnerships gave us leverage in initiating the project and establishing a community working group to be the owner of QA problems.

Transparency. The ROS community believes that decisions should be based on what is best for the entire community, not just a small group of developers and members. Transparency is ingrained in the community conducts. Decisions, opinions, concerns and ideas are publicized in the community forums. The aim is to increase participation and inclusiveness in decisions making.

In the first meeting of the quality assurance working group, a member of the group asserted: *“The meeting must be minuted and published in the forum. We want the community to know what we [are]*

doing.” Another member went further to suggest: “*The meeting should be video recorded and put up in the forum for the community.*”

Transparency does not pose a direct challenge to the implementation of PAR. However, it is an important cultural trait that should be embedded in the process to ensure success. As per the community tradition, we publicized every decision, discussions and outcomes during the working group deliberations in the community quality assurance forum.

Freedom of choice. Contributors, in the ROS community, have the freedom to select the type of work they would do in the community. Tasks allocation is informal; there is no pre-defined process to assign tasks and roles.

A participant in the ROSCon 2017 conference commented about our project during a poster session: “*I’m not sure how you gonna do this? The ROS community likes to code and they participate because of the freedom they get. You may struggle to find people to join this project.*”

Although this cultural trait is not a barrier for PAR, it can constrain the execution of actions. Especially, if the actions are not popular amongst contributors. For example, QA is deprioritized and consequently QA tasks are regarded mundane [1] in the ROS community,.

Collaboration. The ROS community is a collaborative social system. Collaboration is the bedrock of creative solutions and innovation in the community. Its members value joining forces to achieve goals. Community collaboration is essential to effective development of the community’s product. Ideas and projects are announced in the community forums. Interested members join the discussion or volunteer to contribute to the project. Since its transition to a community structure, ROS software has been developed and grown from collaborative work. In the ROS community, initiatives are not handed down from a role with a certain power. They are shared through collaborative channels where the initiator has equal power to the rest of the community contributors.

During the planning of our project, in a meeting with members of the community to discuss an approach to execute the project, a project member suggested to form a “task force” and start implementing our interventions. A community member opposed, he stated: “*That would be disastrous and calls for failure. This community doesn’t operate like that! We need to engage the community.*” Another community member concurred: “*I agree. We shouldn’t bypass the community. We risk to alienate ourselves.*”

This cultural trait is in line with PAR principles of democratic participation and communal reflection. However, PAR literature does not specify how to operationalize collaboration in large and distributed communities. We established a working group dedicated to oversee the work. The community working group serves as our main means of collaboration. To ensure inclusiveness of the larger community, we publicize the group meetings notes and decisions in the community QA forum.

The group members were recruited democratically. One of our collaborators, a community member with over ten years of experience in the community and high karma points and distinguished reputation, put a post in the community forum asking for volunteers to join the group. We succeeded attracting 23 members. However, the group meetings attendance has on average approximately 16 attendees.

Self-management. Another norm of the ROS community is informal self-management. The ROS community has one horizontal path of mobility, and members move from one role to another fluidly. At the center of the community are the core developers. The core team is managed by OSRF, a non-profit organization that has the responsibility of community stewardship. We entered the community with no prior exposure or engagement with the community. We worked on establishing our own reputation. We attended six community events and we showcased our work through presentations and posters. We adopted self-management in our PAR process by ensuring an equal distribution of power amongst participants.

Innovation. The ROS community has an innovative culture. The community environment cultivates and nurtures unconventional and creative thinking. It subscribes to the belief that innovation is not the province of an elite but the effort of the collective and that anyone in the community can be innovative. The energy of the community is directed toward creating new features and the depth and breadth of the ROS platform. In our earlier study of the ROS community [1], we observe that innovation drives the community and remains the centre of attention. Over time, this led to the deprioritization of QA practices.

A participant in the ROSCon 2018 conference commented, privately during the reception, after a presentation made by one of the authors of this paper. He said: “*This is a tough project to execute. Not*

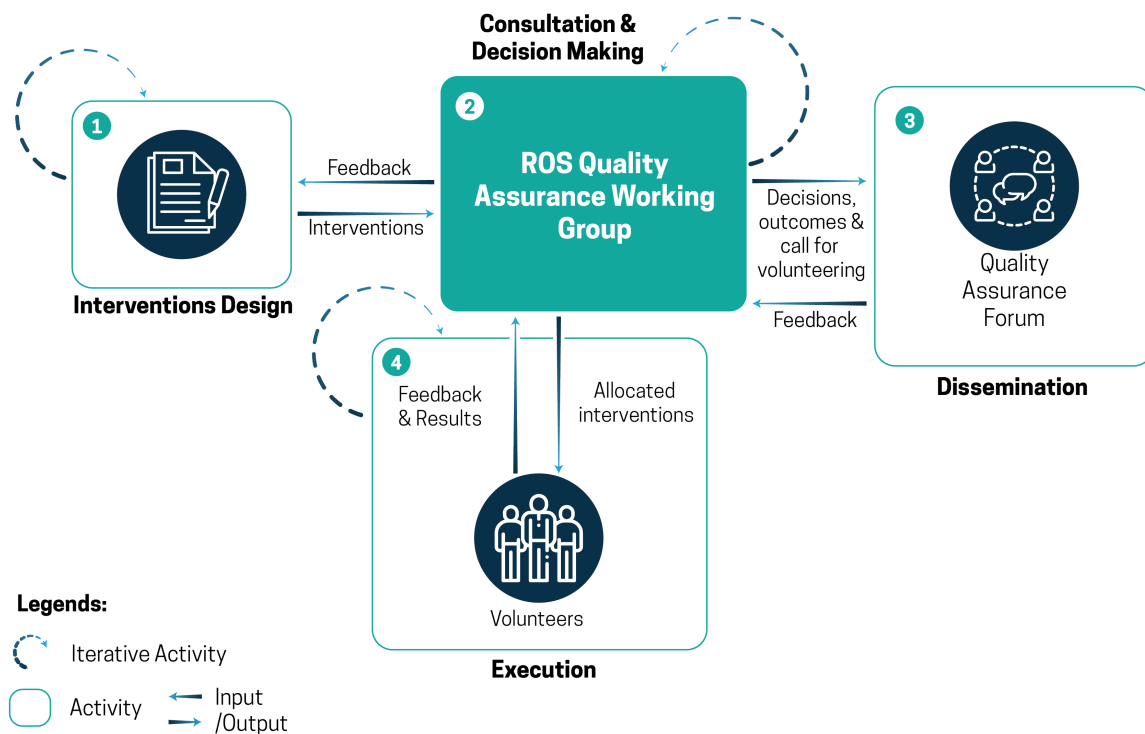


Fig. 4 PAR4FOSS Tailored for the ROS Community

only because people are not under the same roof, but because this community likes to innovate. They prefer working on new features rather than developing their processes.”

Our agenda is to push QA practices in the ROS community. The focal point of the community though is innovation. The QA tasks are seen as mundane. This misalignment of priorities is a challenge for our PAR implementation. PAR assumes that collaboration, authority and control facilitate the execution of actions. However, in an open source community, the implementation of actions relies on volunteers.

6 The Change Process and the Experience

In the following sections, we discuss how we tailored PAR4FOSS (Fig. 1) for the ROS community.

6.1 Tailoring of PAR4FOSS for the ROS Community

Figure 4 depicts PAR4FOSS for ROS and the infrastructure we put in place to carry out interventions. Upon the design of the interventions, we established a working group, ROS Quality Assurance Working Group, to be the problem owner and a platform for initial consultation. We started the process by a prioritization exercise. The outcome was a list of prioritized QA interventions (see Tbl. 2).

The group has been meeting on a monthly basis. The scope of each meeting was to discuss the specifics and details of the implementation of one of the interventions, according to the order of their priorities. The discussions are, then, summarized and publicized in the community quality assurance forum ⁽³⁾. This is an opportunity to engage the wider ROS community and be transparent about the group work. Feedback then an is received from the community and communicated back to the working group for further discussion and deliberations. Once an intervention is stabilized (i.e. unlikely change in the design and the requirements of the intervention), a call for volunteering is made in the QA forum. This is the start of the execution phase (i.e. effectively implementing the interventions). The implementation of interventions is overseen by the group.

³ <https://discourse.ros.org/t/ros-quality-assurance-working-group-october-2019-meeting-notes/10891>

Iterations	Interventions	Description
Iteration 0	Establishing a Quality Assurance Working Group	This is to establish ownership for QA problems. The QA working group has become the owner of QA.
	Quality Assurance Forum	The aim of this intervention is to create a forum for promoting dialog on quality, and the development of quality practices.
Iteration 1	Make ROS packages' quality visible (Part 1)	Create a process and tool where packages quality can be measured, assigned and displayed.
	Appoint ownership	Appoint ownership for quality assurance tools and processes.
	Energize the code review process	Code review is an existing process; unfortunately, it is loosely implemented and practiced. The aim of this intervention is to bring this practice back to ROS QA core quality practices.
	Implement a code scanning method and tool	Identify and implement state of the art code scanning and analysis techniques and tool.
	Maintenance issues	The objectives of this intervention is to attract and recruit new maintainers.
Iteration 2	Make ROS packages' quality visible (Part 2)	This is a continuation of part 1. During this part, we will develop the QA dashboard.
	Quality Hub website	A central "go-to" place for QA knowledge sharing (documentation of QA practices).
	Formalize the code ownership process	Define and implement a code ownership process. Ownership in general establishes the responsibility relationship for a software component, and a responsible developer.
	On-boarding process for core and non-core community members	Description. Define an on-boarding process for core and non-core community members.
Iteration 3	Model-in-the-loop (MIL)testing.	Identify a functional MIL testing setup that is to be used as an integral part of the CI services.
	Automated unit test generation	The goal of this intervention is to automate the creation of the tests.
	QA promotion events	Regular community campaigns to promote QA practices or support a particular community need.
Iteration 4	Model Driven Development (MDD)	Identify and demonstrate opportunities to use the paradigm of MDD for code generation.
	User Rating of packages	Implement a rating of packages feature. The ability of ROS users to rate packages quality.
	Implement a continuous improvement process	A continuous improvement process is ongoing efforts to improve quality assurance practices. These efforts are incremental review and improvement of practices in place. This is a set of community activities and processes to ensure the survival and the sustainability of quality assurance practices.

Table 2 List of iterations and their corresponding interventions

6.2 The Method

The tailored method (Fig. 4) has four dependent activities: (1) Interventions design, (2) Consultation and decision making, (3) Dissemination, and (4) Execution. These activities have inputs and outputs and some of them are iterative sub-processes. They are interlinked to constitute a method that substitutes management and authority and caters for the cultural specifics of the ROS community.

Activity 1: Intervention design. This activity is a pre-requisite to launch the implementation activities (activities 2, 3, and 4). However, the initial list of interventions has been subject of iterative review at the end of each PAR iteration (see Fig. 3), the group conducted a review session for the list of intervention. Table 2 represent the list, after one review at the end of Iteration 1.

The design of the interventions followed a systematic approach. Interventions are derived from four distinct sources: community requirements, similar communities precedents, best practices and problems (i.e. opportunities). During the inquiry phase, we identified opportunities for improvement and solutions for them. Each solution is an intervention. Interventions are composed of a set of actions (small actionable tasks). The implementation of actions will lead to full implementation of interventions. Actions are either an initiative or a corrective measure. Initiatives are new actionable changes with no prior precedent in

the community. Corrective measure is a change to something existing but dysfunctional or unsuccessfully implemented. Each intervention should generate an outcome, i.e. creating a QA capability and behavioral change. Failed intervention will not have an outcome and consequently no impact.

This is an iterative activity. The first version of the interventions was proposed to the working group (part of activity 2) for discussion and validation. There were no objections to the interventions proposed by the researcher. But, the group requested clarifications and definitions of some interventions. Then, the list was put in the QA forum for dissemination. Feedback was received from the wider community (part of activity 3) and discussed in the working group. Consequently, action were taken to amend the list.

Activity 2: Consultation and Decision Making. Community consultation is a reciprocal process and a genuine partnership between the community and the researchers. This community engagement provides guidance to researchers in order to make well informed, acceptable and sustainable decisions.

This is the first activity in the implementation of an intervention (Activity 2 in the process). The first instance of consultation occurs with the working group meetings. Every meeting had a specific agenda proposed by the facilitator (i.e. the researcher) and circulated to the group members for comments, one week prior to the meeting. Then, a final agenda is put forward for the meeting. During the meeting, the facilitator endeavors to balance contributions from the attendees. Each attendee is given the opportunity to express his views.

The group initially agreed on seeking widespread or full agreement via consensus decision-making. In the process all participants' opinions are respected and their contributions are encouraged. The process in which the decision is made is as important as the resulting decision. The facilitator ensures that the power is distributed equally amongst the participants. We observed that no individual or group of individuals attempted to leverage the group to gain power. To the contrary, we noticed that the participants are highly respecting of each other opinions. No frictions in the group were observed either. Personal preferences are less important than a broader understanding of how to work together to help the community.

Decisions are put forward for voting. We use voting to reach decisions. After deliberation, decisions are put forward for voting. A simple majority is sought. We observed, irrespective of the direction a vote takes, the group accommodate the outcome and moves forward. This support for the democratic process has helped the overall process.

Activity 3: Dissemination. All decisions and the working group deliberations are publicized in the community QA forum. We ask the community to provide feedback. In some instances, we received feedback and in other instances, we did not receive any inputs beside "likes."

The aim of the dissemination is to gain a community-wide acceptance of each intervention and the decisions made by the working group. Community dissemination is communication with the wider community through the provision of relevant information and material to ensure effective consultation and transparency. The consultation and the dissemination are an opportunity to democratize the interventions and gain a broader consensus around them. This inclusiveness ensures a democratic process where community members get an opportunity to be heard and provide feedback.

The purpose of the dissemination is to democratize the interventions and gather a consensus around them prior to implementation. Interventions democratization, making interventions information available for everybody, empowers people in the community members to be part of the decision making process.

When feedback is received, it is communicated back to the working group for discussion and consideration. Although, we received some feedback throughout the process, the overall volume has been low (from no comments to ten comments). However, the views have been within the average of other ROS community forums (i.e. the highest 809). This is not a surprise. Our preliminary study [1] shows that QA does not trigger the enthusiasm of the community.

Activity 4: Execution. The execution is the actual implementation of a particular intervention. The execution commences when the intervention requirements and implementation specifications have been stabilized and there is a general positive sentiment for the intervention. This is achieved by consultation and dissemination.

The first forum, we use, to recruit volunteers to implement the actions is the working group. However, in some instances, we were not able to attract enough volunteers or to find volunteers with the right set of skills. Then, we seek volunteers in the larger community by announcing it in the QA forum.

Sub-groups have been formed in some instances to work on particular interventions (i.e. Making ROS packages quality visible). The facilitator attends all the meeting and reports the progress back to the working group.

6.3 The Infrastructure

This process necessitates a simple infrastructure to operate: an (1) online forum, (2) conferencing tool, and a (3) facilitation resource.

Online Forum. This is the platform needed to reach out to the wider community. Although, the community has already multiple forums, we opted for a dedicated forum for QA ⁴. This has been seen before in other communities with well-developed QA capabilities (e.g. Mozilla and Debian). In addition of being a channel to disseminate the QA working group work and decisions, we wanted this tool to attract a sub-community interested in our agenda (enhancing the community QA practices).

Conferencing tool. The ROS community is globally distributed. Organizing meetings with attendees from three different continents has been challenging. However, the intrinsic motivation of some attendees has been a main driver. One attendee stated in a meeting “*I stay awake until 1AM to attend this meeting.*”.

Facilitation Resource. The process requires a facilitator. Facilitation is the activity of ensuring the process is operational at all times. This includes organizing the QA working group meetings, facilitating the meetings, taking and publishing minutes, posting the meetings decisions and outcomes in the QA forum, etc.

7 Evaluation of the Process

In this section, we evaluate the effectiveness of the method (Fig. 4). Effectiveness addresses the degree to which the method has been successful in producing the desired result.

The tailored method (Fig. 4) is deemed successful when its effectiveness can be demonstrated. Effectiveness is determined by (1) the method evaluation: the method activities have produced the desired results, (2) evaluation of interventions: the output (i.e. interventions implementation) is accomplished, and (3) stakeholders evaluation: are the stakeholders satisfied with the method? This can be conducted throughout the life of the project.

Collaborative action research is deemed successful when both parties (Researchers and participants) give and gain benefits, such as new knowledge or improved practical solutions [22]. Meyer explains that action research arises from a different epistemological background than other research methods. This imply, it cannot be evaluated using traditional evaluation criteria. Evaluation of action research is not necessarily based on whether change can be positively demonstrated, but instead on what was learnt from the experience of trying to change practice [32]. In the upcoming sections, we will demonstrate that our participants judged the method being useful and the method has created a change.

7.1 The Method Evaluation

The method is a set of four interlinked activities. These activities exchange inputs and produce outputs (i.e. results). In this section, we evaluate the degree to which the desired results have been produced.

Interventions Design. This activity commenced in the pre-intervention phase. The aim was to produce a preliminary list of interventions for the QA working group to assess, enhance and approve. We designed an initial list of interventions which we presented to the group for discussion and deliberation. We successfully gained a community consensus for the interventions, including by the QA working group and the wider community.

Consultation & Decision Making. This activity was implemented in the form of a consultative body, the QA working group. The QA working group has been a constructive instrument to push the QA agenda forward. The local expertise has been crucial in the review and adaptation of the QA interventions. We observed that the democratic process fits the ROS community, and the process promotes a positive organizational climate. The greater degrees of democratic openness, and transparency of governance, allowed a better engagement with the community.

⁴ <https://discourse.ros.org/c/quality>

The QA working group had initially attracted 23 members and attendance has been in average of 16 attendees in every meeting. However, after a year, this number dropped to seven attendees in every meeting. This did not shake the foundation of the group. The remaining attendees are the highly motivated and dedicated ones. We asked some participants to explain their absence from the group. Most participants justified their absence due to other commitments being prioritized over attending the QA working group. For example, a software engineer stopped attending the QA working group meetings. He emailed the facilitator to convey that his company management reviewed their community engagement and decided to reduce it due to other internal priorities.

Dissemination. We leveraged the QA Forum to disseminate the outcomes of consultations. We, also, leveraged community conferences to promote our work. The facilitator gave three presentations in three different community events (ROSCon 2017, ROSCon 2018, and ROS Industrial conference 2018). This has been productive. The method and the facilitator have become known in the community. Being open about the work with the method and decisions is appreciated in the community. In addition, it fits within the community transparency tenet.

Execution. ROS is a volunteer-based community. Implementing interventions requires volunteers. We attracted a modest number of volunteers to collaborate in implementing the interventions. We learned that volunteers, for this type of contributions, can be unreliable. This, because, they all have a full time employment and their commitment, to contribute to implement interventions, is during their spare time.

7.2 Evaluation of Interventions

The interventions implementation is the output of the method. Every successful implementation of an intervention is a testimony to the usefulness of the method. In this section we evaluate the nine implemented interventions and the change created with the tailored PAR4FOSS. Using the method, we executed two iterations (iteration 0 and 1) and we are half way through iteration 2. Below, we discuss the results of the interventions:

Iteration 0. Establishing a Quality Assurance Working Group. The aim of this intervention was to establish a community body to assume the ownership of QA. This was successfully implemented and the body has become a consultative structure for implementing QA interventions and a community reference for QA. In two separate instances community members reached out to the facilitator to attend the group and pitch their ideas. In another instance, the core team contacted the facilitator and requested to add a new intervention to the list, which is the migration of the current community Wiki to a new upgraded instance.

Quality Assurance Forum: The aim of this intervention was to create a focal place for QA collaboration and a channel to disseminate the QA working group's decisions. This was successfully established and became an active forum for QA-related discussions. The forum receives a steady flow of contributions within the topic of QA. It also created an awareness and visibility for the QA agenda in ROS. The official statistics reveal that in September 2017 the forum had 365 active users and 441 posts.

Iteration 1. Make ROS packages' quality visible (Part 1). The analyses conducted in the pre-intervention phase of PAR and the QA working group identified the lack of visibility into the existing test results as one of the main issues to address. The outcome of the first few meetings of the QA group underlined the importance of reporting Continuous Integration (CI) infrastructure test results. Prior to this intervention, the results of various tests and checks were hidden behind interfaces that require numerous clicks to navigate through. This intervention consolidated the CI test results into a badge (Fig. 5 where the results can be accessed by a simple click. A drop-down menu provides access to a configurable number of historical results, which allow developers to assess a package's health based on trends rather than rely on just the most recent results. This enhancement has produced the following results:

1. We gave ROS wiki visitors insights into the status of tests run by the CI system for a particular ROS package.
2. We provided ROS wiki visitors with a direct way to access package status by consulting test results via UI elements that do not require additional clicks.
3. We enabled ROS wiki visitors to access, by a single-click, the test results stored in the CI system.

4. We provided ROS wiki visitors with the ability to view historical trends in testing results for a particular ROS package.
5. We introduced an element of gamification (i.e. reward developers by showing green badges, checkmarks and other achievement in the form of UI elements). Packages for which all tests are succeeding are rewarded with a green badge with a checkmark. This is the desired situation for all packages.

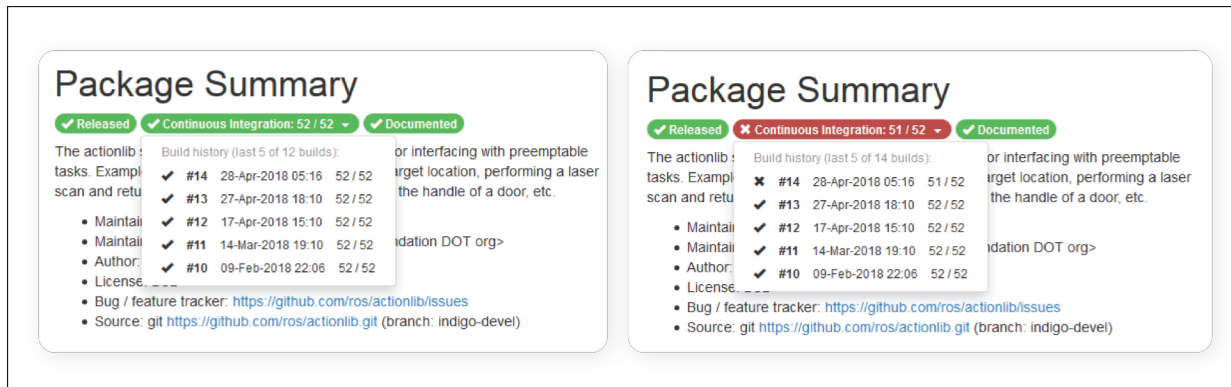


Fig. 5 Green (all tests passed) and Red badge (not all tests passed) and drop-down displaying build history

Appoint ownership. The aim of this intervention is to appoint ownership for ROS quality assurance processes, tools and infrastructure. The QA working group agreed to appoint a default owner to each QA initiative. A default owner is the individual, group of individuals or institution who worked to implement the initiative. If the default owner objects to the ownership, then the QA working group will look in the community to recruit an owner. Ownership was originally suggested to safeguard the sustainability and to ensure the continuous improvement of these interventions. We successfully appointed owners to the implemented interventions.

Energize the code review process. ROS has introduced code review to its quality assurance practices, but this could not be sustained due to various influencers (e.g. prioritizing innovation over quality). The aim of this intervention was to this bring back to ROS. The QA working group selected three ROS repositories, ros-comm, rviz and MoveIt, to pilot code review, before an incremental roll out to the rest of ROS repositories and projects. Disappointingly, we learned that the core team has pulled out resources from ros-comm and rviz maintenance due to shifting focus to work on the ROS2 project. For MoveIt, we observed that only maintainers review code which strains the maintenance resources in the project. We approached eight contributors to investigate why they did not review code. We learned that the most common reason was lack of expertise. One contributor stated: “It took me a long time to feel like I understood MoveIt enough to contribute, and it definitely takes longer to be able to understand someone else’s contribution, especially if it’s in a part of the code that you aren’t familiar with. MoveIt is a very big project, so this happens pretty frequently: personally, I don’t feel like I have enough knowledge to review anything outside of MoveIt Core and the OMPL interface, and don’t do so often.”

Consequently, the QA working group decided to pilot code review in the ROS2 project where there is enthusiasm and resources. We currently negotiate with the ROS2 team the implementation of the pilot. This intervention will be cycles of trial, errors and learning before it will be implemented successfully. ROS is a complex ecosystem and its technology is not beginner friendly either. This constrains the participation in code review, only experienced maintainers and contributor can review code. This will have some implications at the community level. ROS may need to consider lowering its entry barriers to attract a steady flow of newcomers. During the discussion of this intervention in the QA working group a participant stated: “we [ROS community] have to rethink our participation model, if we want to sustain these practices [QA practices]. Currently, newcomers face a complex ecosystem to understand, before they can contribute. This chases away newcomers and impacts our sustainability.”

Even though some members start thinking that some of the interventions are beyond what the group is able to deliver, the majority of the group members are determined to persist. At this stage, this intervention is a work in progress. But, it already demonstrates signs of success. The method has shown the ability to adapt to its environment constraints and obstacles.

Implement a code scanning method and tool: This intervention is executed by researchers. Two angles of attack were considered. First, we have studies quality challenges faced by ROS users and contributors

as observed in the issue tracking system of several packages from the distribution. Based on this analysis, we agreed to develop several project specific linters that can detect these ROS-specific issues (mostly build-time, load-time, and run-time dependencies of various kinds). While the checkers are fairly easy to build, the community still needs to work on how to embed them in the code-review and approval process. Second, we have used seed funding available to sponsor a one-time dedicated security scan of ROS and ROS2 packages performed by Alias Robotics. This work resulted in proposing a method of scanning ROS and ROS2 packages in an effort to create a secure codebase.

Maintenance. ROS is suffering from orphaned packages and lack of volunteers to maintain code. Researchers proposed to design and implement a funding model for unfunded maintenance activities. However, this proposal was strongly rejected by the group. A participant commented, “*we are not bringing money in. That’s not what we are here for.*” This view was strongly supported by other group members. This issue was discussed in-depth during two consecutive sessions. The QA working group concluded that this is a community problem that cannot be addressed by the effort of the QA working group. There was a consensus to de-scope the intervention. Although this can be classified as a failed intervention, the experience has some merits. The group has demonstrated its ability to critically evaluate the interventions and make practical judgments about their faiths. The group concluded that addressing this issue is beyond its mandate. The problem is deeply rooted in the community make-up. ROS is a complex ecosystem. The learning curve for newcomers is long and difficult. Consequently, the community does not attract a steady flow of newcomers, which strains its participation model. This cascades to all aspect of participation in the community, including maintenance.

Iteration 2. Making ROS packages quality visible (Part 2). This is the second phase of making ROS packages quality visible. The intent of this intervention is to roll out a “Quality Dashboard” for every ROS and ROS-Industrial package. The QA working group spent multiple session defining and designing the quality dashboard. The group defined a set of software engineering quality metrics ⁵ ⁶ (e.g. Unit test coverage, code style violations, number of closed issues, etc.). Haros, a framework for static analysis of ROS-based code, was extended by volunteers to generate these metrics. This will be integrated into the ROS BuildFarm to generate an output file, which will be presented in a Wiki UI for ROS developers to consult, similarly to the CI Badge (Part 1 of the intervention). This will enhance further the visibility of packages’ quality. This work is in progress.

Quality Hub website. ROS Quality Hub is an online community knowledge base. It is a go-to-place for QA knowledge (e.g. articles, tutorials, etc.). The content will be volunteered contributions from community members. The website has been setup. The QA working group compiled the initial content. The group is yet to define a strategy to encourage the wider community to contribute.

7.3 Stakeholders Evaluation

This evaluation sought stakeholders assessment of the method and its facilitation. To this end we organized a reflection session. After one year, the QA working group dedicated one of its meetings to a reflection on the group work and the method.

The attendees were asked to provide feedback on the method and its facilitation. We observed that the community portrays the method as the facilitator. Over time, the facilitator personifies the method. This association was reflected in the feedback received from the QA working group participants.

The feedback was overwhelmingly positive. In regards to the method, attendees expressed their satisfaction with its implementation and fitness for the ROS community. One participant said: “*The process works well for the community. But we cannot fix every problem! They [the community] can’t except us to fix everything. There are unfixable things and we need to admit it.*” Another participant echoed the previous opinion. He stated: “*I think the process works fine for the community. You’ve done a great job ... but we need to divide the work into actionable items that we can achieve in a month. If we commit ourselves every month to small achievable items, we will move faster and achieve more.*”

We have not noticed any resentments or objections to our work or the method. To the contrary, the community is appreciative of the method and our achievements. The facilitator occasionally receives emails from the community members praising the work being done. In one email from an engineer, he stated: “*I appreciate the work you do for the community. I wish, I could help. Unfortunately, our management*

⁵ <https://discourse.ros.org/t/ros-quality-assurance-working-group-meeting-minutes-april-2018-meeting/4473>

⁶ <https://docs.google.com/spreadsheets/d/1Ujwc2rjmyWpIamCGNRAdD3USNurXLqunIvgSrSbwvM/edit#gid=0>

restricted our involvement in the community to certain tasks. I wish you luck in these initiatives you taken.”

8 Discussion

8.1 Reflection on the Experience

Action research is designed for conventional organizations. We proved that its adaptability for a FOSS community and applicability to software engineering. The primary novelty of this work (i.e. the PAR4FOSS method) is the ability to introduce change to a FOSS community. The method has proven to be successful in the ROS community. To the best of our knowledge, an adaptation of PAR for FOSS communities has not been attempted previously. The difficulties experienced currently are a natural manifestation of a method in practice. Once a method is deployed in the real world, it would face obstacles and suffer from the influence of various forces.

Given this is a process improvement endeavor, we considered software process improvement (SPI) as an alternative. However, this proposition was dismissed. SPI mandates change in a less participatory style. We wanted a method that engages the community and leverages local knowledge, which we did not have at the beginning.

We now reflect on the main challenges and difficulties observed when working with the proposed method.

The QA working group is at the core of the method, leveraged for consultation and decision making, it is crucial to sustain the participation. Attendance declined due to other priorities, conditions and motivation. This issue was put forward for discussion in a meeting. The group decided to work on the motivation to participate. It has been suggested to change the meeting structure. Instead of dedicating the whole hour to discussing the interventions, the group suggested to dedicate the first 30 minutes for a presentation by guests, experts and contributors on a relevant topic to the group, followed by a ten minutes discussion. Then, the remaining 20 minutes is dedicated to interventions implementation. We recently implemented this new format; we are yet to learn how it enhances the participation in the QA working group.

Lesson 1: Competing priorities hinder the motivation to participate in the process (i.e. participation in the QA working group) by the community contributors. Motivation to participate should be nurtured throughout the process.

We learned that volunteers can be unreliable. The volunteers we recruited have full time jobs. Their involvement in the project takes place in their limited spare time. This has decelerated the execution of interventions and may have an impact on the time line of interventions' implementation. We recognize that these are the means available in the community. The method operates with the resources available in the community and it inherits the community constraints and conditions. The interventions need to be regulated and tailored to the available resources.

Lesson 2: Most volunteers in the ROS community are affiliated. They have full time employment and their commitment to the project is during their spare time. This constrains the implementation of interventions.

We learned that managing resources for an open source project is a challenge. Some of our interventions require advanced skills in the community software tools and infrastructure. Finding available volunteers with such skill is difficult. Even though we managed to secure the participation of a contributor with in-depth knowledge of the community tools and infrastructure, his commitment to the project has been fluctuating.

Lesson 3: Securing the participation of resources with the right skills is a challenge. FOSS projects should ensure a steady participation from contributors with skills matching the needs of interventions.

8.2 ROS 2 and the Need for Flexibility

We learned that the whole method should remain flexible and adaptive. The community is fluid and its priorities keep changing. Few months after we started our project, the community launched the ROS2 project, a dramatic re-architecting of the current ROS platform. ROS2 is meant to address some of the fundamental technical design issues found over time in ROS. Upon the completion of ROS2, it will be up to each ROS user to decide whether to migrate their code to ROS2. ROS2 has attracted the attention and the enthusiasm of the community. The core team has shifted most of its resources to the development and the maintenance of ROS2. The project also attracted the participation of large and influential companies (e.g. Intel, Amazon, and Bosch). This shift of priority has consequences on the portfolio of interventions. The QA working group is facing a challenging question, whether to re-design the interventions to reflect the community focus on ROS2 or remain as is. The group does not seem to have the urgency to shift the focus toward ROS2, because interventions are universal, and can apply to both ROS and ROS2. However, the enthusiasm for ROS2 has impacted the project. For example, one of the attendees of the QA working group, a member of the core team, has stopped attending the group meetings. When asked about the reasons behind his absenteeism, he replied *“we were instructed to not spend time and effort outside ROS2.”* This does not have a direct impact on the design of the method. The portfolio of interventions can be refreshed to reflect the focus on ROS2 if needed. But, this shows that the method in use has to be adaptable to its changing environment.

Lesson 4: The method should be flexible and willing to adapt to its changing environment. This quality ensures the method will continue to deliver.

The QA working group occasionally reflects upon these lessons. While this endeavor is a delivery process, it is at the same time a learning journey.

Limitations We identified some limitations of this study. First, the method relies heavily on the facilitator to ensure the operations of the method. This role can be undertaken by a change enthusiast in the community. Second, the interventions design requires a skilled analyst or researcher. Not all FOSS communities can warrant the presence of a researcher to conduct the analysis required to compile a portfolio of interventions. However, this is an inherent feature of PAR. Third, the commitment to the working group can fluctuate. Members of the group have left the group, due to their other working obligations and new members joined half way through. This fluctuation disturbs the process. Finally, the group priorities may change in a long four year project. This will compromise the execution of the designed interventions.

This method is evaluated in one FOSS community. Ideally, for a method to mature, it needs to go through multiple evaluations and ongoing improvements. We hope this work would inspire other action researchers to implement, evaluate and improve the method in other FOSS communities. We also intend to commence similar projects in others communities.

We have only used the method in a FOSS community. FOSS communities are a type of self-managed teams. Thus, we believe this method can be abstracted to cover other self-managed organizations and teams. The aim of the abstraction is to offer a method that can be instantiated in different circumstances and communities that share similar cultural traits to ROS.

9 Conclusion

We initiated a project in the ROS community with the objective to enhance the current quality assurance practices in the community. We opted for the native form of PAR but we quickly stumbled. We reflected upon the situation and we draw conclusions. These conclusions shaped the process (PAR4FOSS) we put in place. According to Mathiassen, uncertainty, instability, uniqueness, and contradiction exist in action research [30]. Reflection and learning are key elements in action research, and researchers must open their mind to engage in reflections, dialogues, and research efforts [30].

We proposed a participatory action research method tailored specifically for the ROS community. We have found that the ROS community shares its core values, believes and culture with other FOSS communities. Hence, we suggest that the proposed method can be leveraged to introduce change in other FOSS communities.

The strengths and the value of PAR4FOSS is that it provides a powerful means of improving and enhancing practice. It also untackles complexities of the situation. What we proposed is a first building

block in a long journey to build a method for introducing change to FOSS communities. Methods building requires experimentation and errors. Learning from the errors strengthen the method and push toward maturity. We hope that others will be able to build on this experience.

Acknowledgments

This work is supported by the EU's H2020 research and innovation programme, grant No 732287 ROSIN. We thank the volunteers for their work in the implementation of interventions. We thank the ROS quality assurance working group members for their ongoing inputs, advice and participation in the process.

References

1. Adam Alami, Yvonne Dittrich, and Andrzej Wąsowski. Influencers of quality assurance in an open source community. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '18, New York, NY, USA, 2018. ACM.
2. Gary Anthes. Open source software no longer optional. *Communications of the ACM*, 59(8), 2016.
3. Richard Baskerville and A Trevor Wood-Harper. Diversity in information systems action research methods. *European Journal of information systems*, 7(2), 1998.
4. Richard L Baskerville. Investigating information systems with action research. *Communications of the AIS*, 2(3es):4, 1999.
5. Fran Baum, Colin MacDougall, and Danielle Smith. Participatory action research. *Journal of Epidemiology & Community Health*, 60(10):854–857, 2006.
6. Jürgen Bitzer, Wolfram Schrettl, and Philipp J. H. Schröder. Intrinsic motivation in open source software development. *Journal of Comparative Economics*, 35(1), 2007.
7. David Bretthauer. Open source software: A history. *Information Technology and Libraries*, 21(1), 2002.
8. Martin Campbell-Kelly. Historical reflections Will the future of software be open source? *Communications of the ACM*, 51(10), 2008.
9. Andrea Capiluppi and Karl Beecher. Structural complexity and decay in floss systems: An inter-repository study. In *Software Maintenance and Reengineering, 2009. CSMR'09. 13th European Conference on*. IEEE, 2009.
10. Paul A. David, Andrew Waterman, and Seema Arora. FLOSS-US the free/libre/open source software survey for 2003. *Stanford Institute for Economic Policy Research, Stanford University, Stanford, CA (http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf)*, 2003.
11. Christine Dedding, Nicole S Goedhart, Jacqueline EW Broerse, and Tineke A Abma. Exploring the boundaries of 'good' participatory action research in times of increasing popularity: dealing with constraints in local policy for digital inclusion. *Educational Action Research*, pages 1–17, 2020.
12. Yvonne Dittrich, Kari Rönkkö, Jeanette Eriksson, Christina Hansson, and Olle Lindeberg. Cooperative method development. *Empirical Software Engineering*, 13(3), 2008.
13. Christina Donnelly, Geoff Simmons, Gillian Armstrong, and Andrew Fearne. The role of action research in the study of small business marketing and retailer loyalty card data. 2012.
14. Margaret S Elliott and Walt Scacchi. Mobilization of software developers: the free software movement. *Information Technology & People*, 21(1), 2008.
15. Brian Fitzgerald. The transformation of open source software. *Mis Quarterly*, 2006.
16. Daniel M German. The gnome project: a case study of open source, global software development. *Software Process: Improvement and Practice*, 8(4):201–215, 2003.
17. Rishab A. Ghosh. Understanding free software developers: Findings from the FLOSS study. *Perspectives on free and open source software*, 2005.
18. Rishab A. Ghosh, Ruediger Glott, Bernhard Krieger, and Gregorio Robles. Free/libre and open source software: Survey and study, 2002.
19. Davydd J Greenwood, William Foote Whyte, and Ira Harkavy. Participatory action research as a process and as a goal. *Human relations*, 46(2):175–192, 1993.
20. Susanne Gustafsson, Christian Falk, Sofia Tillman, Lisa Holtz, and Lisbeth Lindahl. Life filming as a means of participatory approach together with older community-dwelling persons regarding their local environment. *Scandinavian journal of occupational therapy*, 25(5):347–357, 2018.
21. Alexander Hars and Shaosong Ou. Working for free? Motivations of participating in open source projects. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. IEEE, 2001.
22. Robert Home and Niels Rump. Evaluation of a multi-case participatory action research project: The case of solinsa. *The Journal of Agricultural Education and Extension*, 21(1):73–89, 2015.
23. Chris Jensen and Walt Scacchi. Role migration and advancement processes in ossd projects: A comparative case study. In *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 2007.
24. Stephen Kemmis and Robin McTaggart. *Participatory Action Research: Communicative Action and the Public Sphere*. Sage Publications Ltd, 2005.
25. Stephen Kemmis, Robin McTaggart, and Rhonda Nixon. *The action research planner: Doing critical participatory action research*. Springer Science & Business Media, 2013.
26. Sara Kindon, Rachel Pain, and Mike Kesby. *Participatory action research approaches and methods: Connecting people, participation and place*, volume 22. Routledge, 2007.
27. Bruce Kogut and Anca Metiu. Open-source software development and distributed innovation. *Oxford review of economic policy*, 17(2):248–264, 2001.
28. Martin F Krafft. *A Delphi study of the influences on innovation adoption and process evolution in a large open source project: the case of Debian*. PhD thesis, University of Limerick, 2010.

29. Karim R. Lakhani, Robert G. Wolf, and Others. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *Perspectives on free and open source software*, 1, 2005.
30. Lars Mathiassen. Collaborative practice research. *Information Technology & People*, 15(4):321–345, 2002.
31. Robin McTaggart. Guiding principles for participatory action research. *Participatory action research: International contexts and consequences*, pages 25–43, 1997.
32. Julianne Meyer. Evaluating action research. *Age and ageing*, 29(suppl_2):8–10, 2000.
33. Matthew B Miles, A Michael Huberman, Michael A Huberman, and Michael Huberman. *Qualitative data analysis: An expanded sourcebook*. sage, 1994.
34. Sinan C. Özbek. *Introducing innovations into Open Source projects*. PhD thesis, Freie Universität Berlin, 2011.
35. Shireen S Rajaram. An action-research project: Community lead poisoning prevention. *Teaching Sociology*, 35(2):138–150, 2007.
36. Omer F Rana and Annika Hinze. Trust and reputation in dynamic scientific communities. *IEEE Distributed Systems Online*, 5(1), 2004.
37. Colin Robson and Kieran McCartan. *Real world research*. John Wiley & Sons, 2016.
38. Richard M. Ryan and Edward L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1), 2000.
39. Richard M. Ryan and Edward L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1), 2000.
40. Walt Scacchi. Understanding the requirements for developing open source software systems. *IEE Proceedings-Software*, 149(1), 2002.
41. Walt Scacchi. Free and open source development practices in the game community. *IEEE software*, 21(1), 2004.
42. Walt Scacchi. Socio-technical interaction networks in free/open source software development processes. In *Software Process Modeling*. Springer, 2005.
43. Walt Scacchi. Free/open source software development: Recent research results and methods. *Advances in Computers*, 69, 2007.
44. Maureen A Scully. Meritocracy. *Wiley Encyclopedia of Management*, pages 1–2, 2015.
45. Florian Skopik, Hong-Linh Truong, and Schahram Dustdar. Trust and reputation mining in professional virtual communities. In *International Conference on Web Engineering*, pages 76–90. Springer, 2009.
46. Ernest T Stringer. *Action research*. Sage publications, 2013.