Veteran Developers' Contributions and Motivations: An Open Source Perspective

Patrick Morrison†, Rahul Pandita†, Emerson Murphy-Hill†, and Anne McLaughlin‡
†Department of Computer Science, ‡Department of Psychology
North Carolina State University, Raleigh, NC, USA
Email: {pjmorris,rpandit,ermurph3,acmclaug}@ncsu.edu

***Abstract*—Decades of psychology and sociology research demonstrate that humans change cognitively and socially as they age, so it stands to reason that software developers likewise undergo changes that may affect their work. In this paper, we investigate age-related differences in software developers through the lens of open source software, which is built in communities that tend not to include older software developers. We report on the results of a qualitative panel discussion, then quantitatively analyze such veteran developers' activities on StackOverflow, to understand *why* few veteran software developers' participate in open source, and *how* their contributions to software development differ from their younger peers. Our results suggest that veterans' are less motivated by social interactions than their younger peers. Our results also suggest that veterans could contribute a broader knowledge of software development than their younger peers, as well as knowledge of old technologies that can be applied to newer technologies.**

I. Introduction

Open source software (OSS) is software for which the source code is made publicly available, allowing other developers to build upon that software. OSS is also important because it enables innovation [1]; rather than many organizations implementing the same functionality, they can all contribute to and reuse that functionality as OSS.

Unfortunately, contributions to OSS are not as age-diverse as other kinds of software development. The OSS community is comprised almost entirely of young developers. In Ghosh's survey of 2700 developers, 95% of respondents were under age 42 [2]. David and colleagues' survey of 1588 developers found a median age of 27 [3], whereas a more recent survey found the median age of general open source contributors was 33 [4]. We define "veteran" software developers as people with at least 15 years of software development experience and who are at least 40 years of age, a common age threshold defining "older workers" in the literature [5], [6].

Figure 1 illustrates the skew towards young OSS developers by comparing the age distribution of open source developers [4] against employed US software developers [7]. Although there are some confounding factors (for example, sampling methodology), the trend is clear: veteran developers are not well-represented in the open source community.

The relative dearth of veteran software developers in the OSS community is a problem for several reasons. First, veteran software developers are a large and capable group that could make significant technical contributions to OSS. Second, veterans may bring valuable management and interpersonal perspectives to OSS projects. Last, contributing to OSS and engaging in the community has been shown to have many personal and professional benefits [8] that could be enjoyed by veteran developers, if they were to participate.

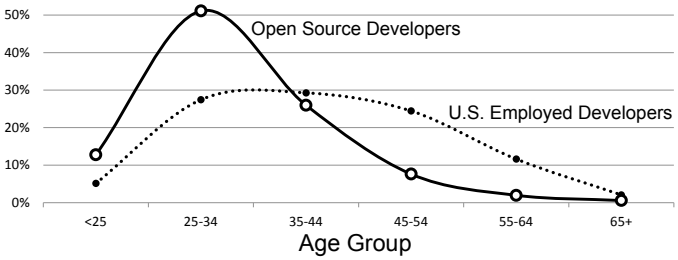

Fig. 1. Age distribution for open source and general developers.

The contribution of this paper is a quantitative study of StackOverflow, based on a qualitative study of a panel of veteran developers, which together provide a better understanding of the unique contributions of veteran developers and reasons for their lack of participation in OSS. From this starting point, in the future we can form, evaluate, and refine hypotheses, and then act on them to foster a long-term increase in the number of veteran software developers in OSS projects.

II. Related Work

The most related work to ours is that of Davidson and colleagues [9], [10], who qualitatively investigated older developers' motivations for contributing to open source and the unique contributions they make. They performed a diary study with 4 first-time older contributors to open source [9] and interviewed 11 older adults who currently contribute to open source [10]. The findings of our panel validate the findings of Davidson and colleagues through triangulation; we point out parallels between our work and Davidson and colleagues' throughout the remainder of this paper. Additionally, our StackOverflow investigation quantifies these findings using a larger sample of developers.

Two previous pieces of research have touched on software development from an aging perspective. The first was Brooke's study of 71 employees of Finnish IT firms, some of whom were engaged in software development, that investigated career paths of IT workers as they aged [11]. According to that study, one of the major difficulties faced by older IT workers was

978-1-5090-0252-8/16/$31.00

an unsupported perception that technical skills decline with age, and that workers over 40 were expected to transition from software development to generic management. The second was Lünstroth's two case studies that confirmed the bias against older developers, noting that as developers aged, they tended to specialize in certain legacy technologies, which reinforced younger developers' biases that older developers were unable to cope with newer technologies [12]. Our work furthers these findings by investigating what motivates and demotivates veteran developers in regards to OSS development.

Our paper explores attitudes and experiences of individuals in the "midcourse" of their lives. The midcourse is a life stage that falls between the career building stage and old age [13]. Changes from the single job career pattern to the increasingly heterogeneous "boundaryless careers" [14] — characterized by work in multiple firms and occupations — have forged a mid-career life stage that is fraught with precariousness and uncertainty [15]. The midcourse also represents a period of life in which individuals must face numerous changes in family relations [13], and typically endure precipitous declines in social interaction and in new social network connections [16]. At the same time, the midcourse offers tremendous promise for personal and social development [13]. It is a time for introspection [17], which serves as the impetus for embarking on "second acts" in their personal and professional lives [15]. By serving as mentors for younger workers, engaging in volunteer work, transitioning to new careers/work projects, and making other life alterations, many midcoursers can reinvigorate their network connections, their self efficacy, and their well-being [18], [17]. Social interaction is related to better health and well-being at all ages [19]. Those with active social lives are better able to deal with stressors and live longer than those without less social activity [20], [21]. One of the attributes of the OSS movement is the social collaboration between developers, on an informational level and on a personal one [22]. Such collaborative problem solving can be an engaging and rewarding portion of the OSS process. At the same time, the midcourse is a tremendously heterogeneous process that follows few patterns and is rarely guided by institutional supports [13]. This adds to the precariousness and uncertainty of this life stage and underscores the importance of studying midcourse experiences and opportunities.

Others have researched diversity in software development from other angles. For example, researchers have studied gender imbalance in software development [23], [24], [25], [26], [27]. In particular, recent work by Vasilescu and colleagues has shown that in OSS teams, gender and tenure diversity are positive predictors of productivity [28]. In contrast, our work focuses on diversity from an aging perspective.

III. Preliminary Study: A Panel of Veteran Developers

To generate hypotheses to explain veteran developers' lack of participation in open source, we convened a panel of veteran software developers. Because of the formative nature of this first study, we used the group discussion method [29] to elicit development experiences, motivations, and barriers to open source participation. This method helped us gather detailed and contextualized information, much of which was shared with us only after panelists heard others' stories.

Using purposive sampling [30] of people who had experience in software development and were over the age of 40, we distributed advertisements through contacts at local companies. Of the six who volunteered, four were able to meet scheduling constraints and participate in the panel. Although four panelists seem like a small number, recall that the objective is not to draw generalizable conclusions, but instead to generate testable hypotheses.

Three of the participants came from the same large information technology company, and one from another such company. Three participants were male, and one was female. Participants had between 10 and 28 years of software development experience (median of 26.5 years), and were between 50 and 56 years of age.

We asked panelists about their experiences and perceptions about software development and open source, as well as how those experiences and perceptions have changed over time. Two authors of this paper moderated the panel: one made sure that the panelists stayed focused and that all participated, while another author asked technical questions to elicit deeper responses as needed. The panel discussion lasted about 1.5 hours, falling within the optimal 1-2 hour time frame recommended for such groups [29].

We recorded and transcribed video and audio from the panel, then extracted themes related to open source, and differences between older and younger developers. For each theme described below, we compare panelists' claims to relevant prior research findings.

A. Heavy Reliance on OSS, but Few Contributions

Panelists reported that OSS was critical and pervasive in their software toolboxes, each listing several pieces of OSS he or she used regularly. While they thought that 20 years ago they could have relied exclusively on commercial software, they reported that they now "can't get along without" OSS.

Panelists reported that they currently make few or no contributions to OSS, a choice partially forced on them by their employers: their employers allowed them to create OSS, but required that they first complete an approval process that panelists considered too slow, cumbersome, and conservative. This may be because companies are largely motivated to contribute to open source for economic reasons, which is at odds with individual developers' motivations for doing so [31].

Moreover, some panelists noted that after creating software at their day jobs, they had little desire to go home and sit in front of a computer. This did not seem to be because panelists do not enjoy applying their software development skills outside of work — two panelists reported volunteering as coaches for youth robotics teams and reported that those teams benefited from their software development experience. One participant created software to track his long distance bicycling and provide updates on his location to family and

friends. Thus, although participants explicitly indicated little interest in programming at home, nearly all had, at times, been motivated to create software in their free time.

B. A Desire to Contribute to OSS

Panelists expressed a desire to give back to the OSS community, citing several sources of motivation. First, panelists said they would enjoy developing OSS if they knew that many other people were using their work. Second, panelists noted that they enjoyed the problem solving that software development entails. Third, panelists reported an altruistic desire to contribute, one saying "as you get older, you get more altruistic." This finding is consistent with other findings about older adults' contributions to virtual organizations, such as older knowledge workers' altruistic motivations for contributing to online knowledge repositories [32]. Likewise, Davidson and colleagues found altruism a motivator for older developers' participation in open source [9], [10].

There is a conflict, then, between panelists' desire to develop OSS and their perceived difficulty in doing so. One panelist mentioned that he may not be prepared to contribute: What projects would find his skills useful? What projects would he find most appealing? How would he get started?

C. Potential to Uniquely and Significantly Contribute to OSS

Panelists identified three main areas where they could make significant contributions to software projects, especially in comparison to their younger colleagues.

Technology Reinvention. First, panelists felt strongly that old technology is constantly being reinvented. For example, panelists noted a recent resurgence in interest in virtual machine software in industry and research, but that such technology has roots in the 1960s with IBM's System/360 mainframe. Davidson and colleagues' interviewees also hinted at technology reinvention being a strength of veterans [10]. This suggests that veteran software developers can help newer OSS teams learn from the successes and failures of previous development teams.

General Problem Solvers. Panelists stated that they, more so than their younger colleagues, tended to be generalists when it came to problem solving. They characterized the skills of some younger developers as sometimes too focused on a single technology, and their own as applying across technologies. This suggests that veteran developers could serve as flexible problem solvers on OSS teams. This is partially supported by prior work, which suggests that older adults are more problem-focused in solving everyday, instrumental problems [33].

Emotional Intelligence. Panelists suggested that they bring much-needed social and emotional intelligence to software teams. One panelist described some problematic, younger peers as "hotheads" who had difficulty dealing with other people. He reported that he had helped some such hotheads because he used to be one himself. Similarly, Davidson and colleagues' interviewees noted that "maturity" and "life experience as a user, parent, [and] spouse" made veteran developers especially valuable [9]. These findings may suggests that veteran developers' high emotional intelligence helps facilitate the expression of group intelligence [34]. Higher emotional intelligence is a characteristic of older adults, who are better able to problem solve in emotional situations [35].

D. Motivational Mismatch

The motivations of current, mostly young OSS developers did not appear to align with veterans' motivations for contributing to OSS. Let us consider several motivations in turn.

Improving Human Capital. Existing OSS developers cite skill development as a major motivation for contributing to OSS [2], [8]. However, veteran software developers already have extensive experience, and thus are arguably less likely to be motivated by the desire to improve their development skills, at least for the purpose of attaining their next job. This mismatch comports with Davidson and colleagues' interviews [10], which found veterans less likely to be motivated by career-related benefits.

Socialization. Younger, existing OSS developers tend to be motivated by the social aspects of OSS development [3]. However, our veteran panelists were not. This difference echoes other findings that suggest that the acquisition of new social network connections tends to decline with age [16], while existing ones become stronger [36]. Still, social aspects of online communities may be beneficial for those who contribute to OSS, such as through improved psychological well-being and life satisfaction [37].

Altruism. Veteran developers on our panel mentioned altruism as a major motivation to contribute to OSS. Psychology research likewise suggests that altruism increases with age [38]. However, when Hars and Ou gave existing OSS developers a choice of eight different motivations for contributing to OSS, altruism was the second-least popular motivator [8].

This mismatch between the motivations of existing, younger OSS developers and veteran developers may partly explain why so few veteran developers participate in OSS. Existing OSS projects and tools may satisfy the motivations of younger software developers, but may not appeal to the motivations of veteran software developers. Thus, projects, processes and tools that better align with veteran developers' motivations may yield higher rates of participation.

IV. A Quantitative Study of StackOverflow

To evaluate the claims that panelists made, we examined data from StackOverflow,[1] a question-and-answer website where community members garner "reputation" by asking good questions and providing good answers, as voted on by the community. StackOverflow users can optionally specify their age; while only about 19% have done so, this still represents more than 300,000 users [39].

While we might want to evaluate the claims of the veterans in our panel by studying open source communities directly,

[1] http://www.stackoverflow.com

we instead study StackOverflow for two reasons. First, we know of no large-scale data that links open source contributors to age. Second, even if we did, we would intrinsically be missing data from non-contributors to open source. Thus, we chose to study StackOverflow as a means to study skills and motivations of veteran developers generally, rather than in open source specifically.

In this study, we examine developer aging in a continuous way for some claims, but in a discrete way for others. When we treat 'veteran' and 'younger' developers as discrete groups, we define veterans on StackOverflow as community members whose self-reported age is 40 or above. Although our earlier definition of 'veteran' requires at least 15 years of experience, we could not reliably measure members' development experience; thus, we relax the experience requirement here.

When possible, we run non-parametric statistical tests to evaluate differences between groups, because much of the data is non-normally distributed. For example, the distribution of answer scores is positively skewed, that is, there are many answers with scores of zero or one, but few with higher scores.

We used StackOverflow data to quantitatively analyze two open questions from the panel. First, what strengths do veteran developers have compared to their younger peers, with respect to technology reinvention, general problem solving, and emotional intelligence? Second, are veteran developers really unmotivated by improving human capital and socialization, but more motivated by altruism? These questions are inherently difficult to answer; for instance, psychologists have raised questions about the construct validity of state-of-the-art questionnaires for measuring emotional intelligence [40]. Like this existing research, and because we do not have direct access to participants, we use proxy measures to evaluate our research questions. To boost the reliability of our findings, we use triangulation where possible by answering the same question in different ways. Finally, while our measures are not validated like some social science measures, they do have one relative advantage; we measure them in the context in which they are practiced, that is, software development itself. We next investigate our two questions in turn.

A. Unique Strengths of Veteran Developers

1) Technology Reinvention: Panelists felt that new technology is often reinvented old technology, and their knowledge of that old technology gave them an advantage compared to their younger colleagues.

To evaluate this claim, we chose a list of old and new technology pairs based on Rigaux's programming language history graph, which shows how programming languages evolved from older programming languages.[2] We then reduced the number of candidate technology pairs to the "top 10" for which there exist the most questions on StackOverflow, based on the questions' tags. The technology pairs we considered are shown in the left two columns of Figure 2. We next found developers who answered questions about the successor technology, then divided this group into those who also answered questions about the predecessor technology, and those that did not. Our hypothesis was that the average answer score for developers who answered questions about the predecessor technology would be higher than for those who did not.

[2]http://rigaux.org/language-study/diagram.html

Technology	Predecessor	Without	With Δ	Sig	# Answers
C#	Java	2.43	+0.30		836686
C#	C++	2.77	+0.37	*	452942
Java	C++	2.75	+0.43	*	374812
C++	C	2.94	+0.12		311581
Python	C++	3.63	+0.46	*	188756
Python	C	3.68	+0.51	*	184872
Ruby	Python	3.62	+0.35		64697
Ruby	Smalltalk	4.53	+1.26		3162
Scheme	with Lisp	3.36	+0.21		2867
Common Lisp	Lisp	3.85	+0.01		2387

Fig. 2. Mean answer scores for developers for predecessor-successor pairs.

To test our hypothesis while controlling for confounding factors, we ran a linear regression for each pair, to predict answer scores based on the presence or absence of the predecessor technology. We wish to control for whether veteran developers have knowledge of both the predecessor and the successor technologies. To do so, we include the age category (younger, veteran) in the regression. Other attributes of the users answering the questions (e.g. overall experience and ability, or frequent use of StackOverflow) may drive high answer scores rather than the presence of a predecessor technology. We include number of answers in the regression, as a proxy for frequent use of StackOverflow. We report where one or more of the presence or absence, age category, and number of answers independent variables is statistically significant with respect to predicting answer scores. Where presence or absence is statistically significant, we can reject a null hypothesis of average answer score will not be different between developers who answered questions about a predecessor technology and developers who did not answer questions about a predecessor technology.

Let us give a short example before explaining the results. Suppose Shirley and Brittany are developers that have each answered 10 questions on StackOverflow about Java. Shirley has answered 5 other questions about Smalltalk, whereas Brittany has not. Our hypothesis predicts that the scores for Shirley's 10 Java answers will be higher than Brittany's scores for her 10 Java answers, because Brittany is assumedly knowledgeable about Smalltalk, the predecessor technology.

Our results are reported in Figure 2. The first column indicates the successor technology. The second column indicates the predecessor technology. The third column indicates the mean answer score for the successor technology for developers without knowledge of the predecessor technology. The fourth column indicates the change in mean answer score for developers with knowledge of the predecessor technology. The fifth column indicates whether the presence of the predecessor technology is statistically significant (* means $p < .05$, R lm() procedure). The final column indicates the total number of

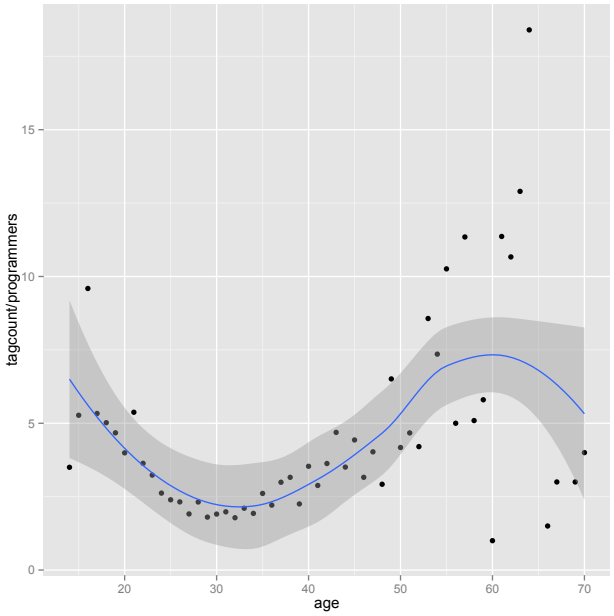

Fig. 3. Unique tags for questions answered by age.

answers to questions about the technology.

For example, the table indicates that the mean answer score for Java answers by developers without knowledge of predecessor C++ is 2.75, while the delta for developers who know C++ is +0.43, and that difference is statistically significant. Four out of ten language pairs substantiate panelists' views that having experience with an old technology gives you knowledge about a reinvented new technology.

Summary of Findings

StackOverflow data suggests that knowledge about a technology sometimes improves developers' knowledge about a successor to that technology, controlling for age and StackOverflow experience.

Consider two main limitations in interpreting this finding. First, C++ was linked with three of four statistically significant results (with C#, Java, and Python), and C was linked with the fourth (Python). However knowledge of C did not have a statistically significant effect on C++. One possibility is that some language pairs may have too much in common (e.g. C# and Java, C++ and C) to show statistical significance. Another possibility is that some other factor associated with developers who know C/C++ is responsible for the significance. Second, the selected successor and predecessor technologies may not be representative of all such technologies, and a predecessor we selected may not have been the most influential predecessor to a technology. We plan to explore these questions further in a future investigation of technology reinvention.

2) General Problem Solvers: Panelists claimed that they were generalists compared to their younger colleagues. We investigated this claim in two ways.

First, we examined the number of tags that developers answer questions about. As we reported in an earlier data challenge paper, veterans do answer a wider variety of questions than their younger peers [39]. Figure 3 shows an updated plot from that paper, which displays the average number of unique tags contained in answers provided by developers at various ages. The X axis is age in years, between 15 and 70. The Y axis shows the average number of unique tags per developer. As the figure suggests, developers tend to answer questions around a few topics at age 30, then steadily answer more questions every year through their 50's and 60's. We interpret the higher values on the left side of the graph as experimentation by young enthusiasts. The data becomes erratic at the right of the graph because there are fewer older developers on StackOverflow than younger ones.

We also evaluated veterans' ability to answer generalist questions by calculating their scores for answering questions tagged only with generalist concepts, those concepts useful in a variety of programming situations. We define generalist tags by starting with 11 concepts mentioned in the table of contents in an algorithms textbook [41]: algorithm, sorting, hashtable, trees, recursion, graph, array, multithreading, string, and set. Then, we add to that set commonly co-occuring technology-agnostic tags. We define commonly co-occuring by each tag's related tags and synonyms on the tag's 'about' page on StackOverflow. For instance, the following tags commonly co-occur with string: regex, compare, replace, integer, and reverse. We define questions as technology-agnostic when they are tagged exclusively with these generalist tags. All other questions we tag as technology-specific. We then compared the answer scores of the different groups using a Wilcoxon-signed ranks test. We randomly sampled up to 50,000 questions per group, capped by the StackOverflow API. We reason that a 50,000 question sample is of sufficient size to detect trends.

Overall, veterans did not garner significantly better scores for technology-agnostic questions. The difference between the score for veterans' answers to technology-agnostic questions versus younger developers' answers is not statistically significant (p=.56). Likewise, the difference between veteran's answer scores to technology-specific questions versus their answers to technology-agnostic scores is not statistically significant (p=.40).

Summary of Findings

The StackOverflow data suggest that veterans have broader knowledge than their younger peers, but their answers for generalist questions are not significantly better.

One limitation to consider is that, when comparing veteran to younger StackOverflow members, it may be that one group tends to be more self-selecting than the other; for example, it may be that younger developers on Stack Overflow are fairly representative of the general developer population, but veterans on Stack Overflow tend to represent only the most knowledgeable veteran developers.

3) Emotional Intelligence: Panelists claimed to have better emotional intelligence than their younger peers. We hypothesized that, if this claim is true, veteran developers on Stack-

Overflow, compared to their younger peers, would behave in a more emotionally regulated manner when interacting peers. In support of this hypothesis, Meeks [42] identified affective wisdom, "positive emotion and behaviors toward others, and absence of indifferent or negative emotions toward others", with multiple subcomponents of wisdom as defined in the neurobiology literature.

The first way to tested this hypothesis was by measuring the affect of comments made by a sample of StackOverflow members across the age spectrum. For ages from 15 to 70, we randomly selected 10 developers from each year of age. We then analyzed the sentiment of 10 random comments from each developer. We then automatically analyzed these comments for affect; we chose comments, rather than questions or answers, because comments are more informal and intended to solicit clarifications, which was more congruent with the activities that necessitate emotional intelligence, as described by our panelists. We measured the level of hostility of each comment with De Choudhury and colleagues' affect analysis tool [43]. We then aggregated the data by finding the maximum hostility exhibited by each person, then compared the distribution of hostility for comments from community members that are veterans versus those from younger community members. We use maximum hostility because it is more consistent with panelists' use of the term "hothead," compared to mean hostility; hotheads are not always hostile, but instead have short periods of high hostility. The results show that the mean maximum hostility level for veterans was 0.44 and the mean maximum hostility level for younger developers was 0.43. The difference is not statistically significant ($p = .761$, Wilcoxon rank-sum test).

The second way we tested this hypothesis was by examining what comments on StackOverflow are tagged as 'offensive' by the community. If the hypothesis is true, veteran developers' comments will be less likely to be flagged as offensive. The results show that while comments being flagged as offensive are very rare, the percentage that are flagged as offensive for veterans (0.0018% or 429 offensive comments) and younger developers (0.0016% or 131 offensive comments) are not significantly different ($p = .614$, Wilcoxon rank-sum test).

The third way we tested this hypothesis was by examining the quality of suggested edits to questions and answers. Specifically, if veterans are better able to facilitate group intelligence as suggested by the panel, then we would expect veterans' suggested edits to questions and answers would be of higher quality. Here we define high quality edits as those that receive a high ratio of total upvotes to downvotes from the community. Overall, veterans had a higher ratio (4.6 upvotes per downvote, about 2,680,213 total votes) compared to younger developers (3.7 upvotes per downvote, about 229,350 votes), a difference that was statistically significant (chi-squared test, $p < .001$).

Summary of Findings

While not conclusive, we found some differences in levels of emotional intelligence between veteran and younger developers.

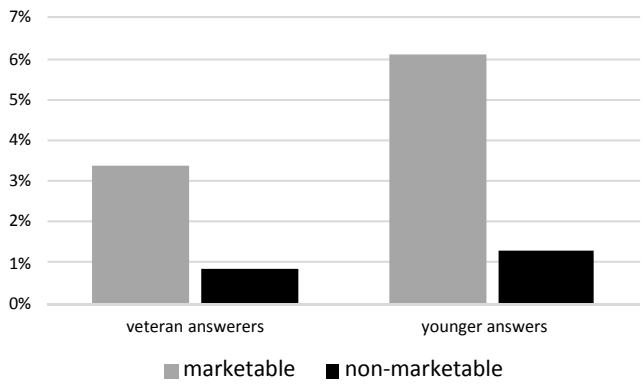

Fig. 4. Percentage of developers in each age group who answer questions with marketable and non-marketable tags.

One of the main limitations of this finding is that De Choudhury and colleagues' affect analysis tool was not trained on StackOverflow data, and thus may provide inaccurate measures of hostility. Another limitation is that we analyzed only a subset of the comments on StackOverflow – it may be that a larger investigation might uncover differences.

B. Veteran Developer Motivations

We next examine 3 hypotheses about veterans' motivations.

1) Improving Human Capital: Veterans may be less interested in improving their job prospects than younger developers. We investigated this claim by looking at whether younger developers will be more likely to answer questions that they believe will improve their job prospects.

To test this hypothesis, we started by analyzing the types of job posts on StackOverflow's Careers website.[3] Each job posting is tagged with a topic, such as javascript, just like any StackOverflow question would be. From this, we created a set of "marketable" tags; these were tags that appeared in at least five different job postings. This list was comprised of 30 tags (e.g. javascript, java, c#, php, python, and ruby-on-rails). We then created a set of "unmarketable" tags; these were a set of the most popular 30 tags used on StackOverflow that were not used as tags for any of the job postings (e.g. regex, xml, facebook, eclipse, and visual-studio-2010). Then, we computed the percentage of veteran developers in the community that answer questions about marketable tags and unmarketable tags, then did the same for younger developers.

Figure 4 displays the results. These results show that, indeed, younger developers are significantly more likely than veterans to answer questions about topics that improve their chances for getting a job ($p < .001$, Wilcoxon signed-rank test). However, younger developers are significantly more likely than veterans to answer questions about topics that do not improve their chances of getting a job ($p < .001$), perhaps because veterans are simply less active than younger developers, in terms of the selected tags. Using a linear

[3]http://careers.stackoverflow.com

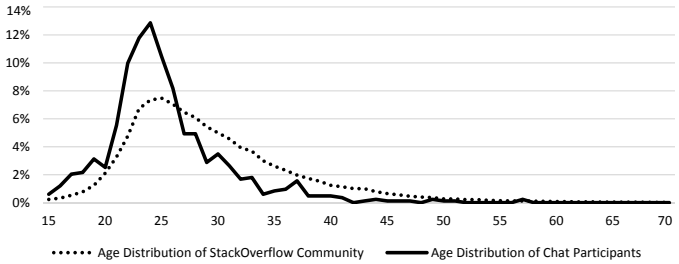

Fig. 5. Distribution of ages of StackOverflow members versus the age of StackOverflow members who used chat rooms in a 30 day period.

model to control for activity as measured by total number of answers given, this effect remains significant ($p < .001$, R lm() procedure).

To express the data in a different way, 4.7 times as many younger StackOverflow developers answer questions about marketable tags than they do about unmarketable tags, but only 4.0 times as many veteran developers do.

Summary of Findings

Veterans appear less interested in marketable skills than their younger peers.

One major limitation of this analysis is our operationalization of non-marketability; specifically, some non-marketable tags are still valuable when it comes to gaining employment. Another potential confounding factor is that veterans may be not knowledgeable about marketable technologies because they tend to be new technologies that were not available when veterans started their careers. However, our prior work casts doubt on that explanation, showing that veteran StackOverflow users are equally or more knowledgeable about new technologies [39].

2) Socialization: Veteran panelists claimed to be less motivated by social aspects of software engineering.

To validate this claim, we measured the proportion of veterans that engage in social activities compared to younger developers. In an initial examination, we were only able to link about 10% of participants in StackOverflow to participants on Meetup.com[4], and if we include only participants who indicated their age, that number drops significantly.

Instead, we examined which StackOverflow members participated in virtual socialization activities. We scraped chat participant identifiers from the 10 most active chat rooms on chat.stackoverflow.com over a 30 day period, yielding 1984 chat participants. The mean age of chat participants was 24.9, whereas the mean age of participants on StackOverflow as a whole was 29.2, a statistically significant difference (p=.031, Wilcoxon rank-sum test). Figure 5 shows the distribution.

Summary of Findings

Veteran developers appear less likely to participate in online social activities than their younger colleagues.

[4]www.meetup.com/stackoverflow/

One limitation of this study is that chat session participation may not be representative of all types of socialization a developer may engage in. Another limitation is that we measured level of socialization as a binary option: participation or no participation. A more sophisticated notion of socialization may yield different results.

3) Altruism: Panelists claimed that one reason that they would participate in open source was for altruistic reasons. For the purpose of measuring age's effect on altruism, we hypothesize that younger developers are more likely than veteran developers to exhibit a drop-off in activity after receiving public recognition. The notion is that veterans are more likely to do something for altruistic reasons, while a younger person might be swayed by perceived gain for themselves in the form of recognition. Grant and Betts have shown that, indeed, a drop-off in activity does occur for many StackOverflow users after they receive public recognition in the form of badges [44]. Likewise, Mamykina and colleagues call this the "shooting star" pattern [45]. We build on this by exploring what *kinds* of users exhibit these drop-offs.

In order to operationalize this using the StackOverflow data, we sought badges that were measurable, popular, and participatory. By 'measurable', we mean that the badge was awarded based on criteria that could be measured through the data made available through the StackOverflow data site, rather than though criteria that were subjective or non-visible. By 'popular', we mean that the badge receives a great deal of attention and is awarded frequently, relative to all badges, making it a stronger indicator of average behavior than more specialized or rare badges. By 'participatory', we mean badges that are based on activity rather than on knowledge or insight, making them easier to work for as an end, rather than as a consequence of desirable behavior.

Based on these criteria, we selected the 'Copy Editor' badge, defined as 'Edited 500 posts'.[5] We selected users who had achieved the badge, and who had specified their age, and summed the number of post edits each user had performed in the six months before they received the badge ('before'), and the six months after they had received the badge ('after').

We applied Wilcoxon signed rank tests to the before and after number of edits for each age group. For the 'Copy Editor', badge receipt is not statistically different. Figure 6 shows this result; both groups show a pattern of decreasing edititing activity after receipt of the badge.

Summary of Findings

Younger developers and veterans did not show a statistically significant difference in altruism.

Several limitations apply to this analysis. First, the copy editor badge may not be representative of all StackOverflow badges. Second, something other than altruism may cause veterans to continue to edit others' posts after receiving the badge; for instance, veterans might be more prone to habitual behavior.

[5]http://stackoverflow.com/help/badges

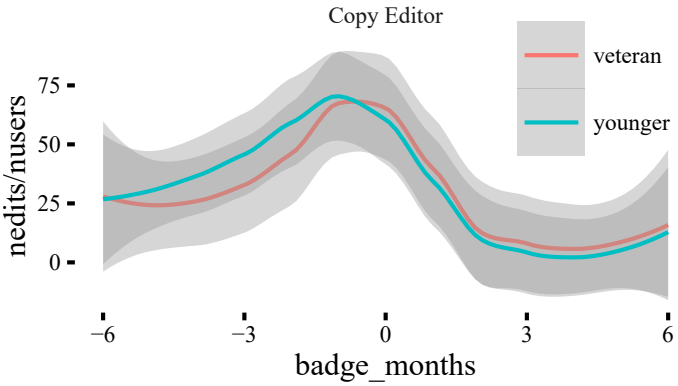

Fig. 6. Number of edits for six months before and after badge receipt.

V. Discussion

Although the results from these studies provide limited generalizability, they provide a starting point from which we can hypothesize about challenges and opportunities faced by veteran software developers when they contribute to OSS.

Development Beyond Retirement. Now is a critical time to study and increase veteran developers' participation in OSS. A large segment of veteran software developers are *baby boomers*, who are just now beginning to retire. These developers are experienced and highly capable of making significant contributions to OSS. We are facing a significant shortfall of qualified software developers [46] and researchers have made significant progress in training and equipping our youth with the skills and tools necessary to become such developers [47]. However, researchers have not, to our knowledge, made an effort to understand how to keep the highly capable veteran developers in the community. In other words, significant resources have been expended to build human capital by increasing the number of people who choose software development as a career, but there has not been a significant effort to maintain this human capital by retaining people as their careers progress.

Integration of Unique Skills. Finally, while veteran software developers appear to have significant contributions to make to open source, such as through their breadth of knowledge, it is unclear how and where they can best contribute. Open source is traditionally viewed as a meritocracy [48], where participants advance by "proving themselves technically in the responsibilities of their position" [49]. These studies suggest that, when veteran software developers have experience in high-level roles in their careers, they cannot simply "jump in" to a high-level role in an OSS project.

VI. Limitations

Although our panel and StackOverflow investigation provide several insights into veteran developers' challenges in contributing to open source and their potential unique contributions, the reader should consider several limitations.

With respect to the panel, one of the main limitations is that panelists are unlikely to be representative of all veteran developers. Another limitation is that the panel was only 1.5 hours long; although we did this out of respect for panelists' time, other findings may have emerged had we continued the panel for a longer amount of time. Moreover, while the panel format allowed developers to support or contest each others' experiences, the format allows the more vocal participants to sometimes dominate the discussion, despite our attempts to elicit data from all participants. Another limitation is that the panel did not include veterans who are regular open source contributors, who likely have additional perspectives.

With respect to the StackOverflow data, beyond the limitations of our individual analyses, there are two main limitations. First, the participants on StackOverflow may not be representative of the developer population as a whole. Many developers do not participate in StackOverflow, and young developers especially make up the bulk of participants [39]. Second, as we mentioned previously, construct validity is an inherent threat. For instance, StackOverflow's reputation score likely does not capture all dimensions of developer expertise and is also subject to the whims of the community. For example, an expert answer an esoteric programming question may garner more reputation than an inexpert answer to a popular question. Our measures of emotional intelligence, "hostility" and "offensiveness" are negative, and give only partial measures of emotional inelligence. Further, they are confounded with gender, as there is evidence of gender differences in online communications [50].

VII. Conclusion

The research described in this paper just begins to outline an understanding of the challenges and opportunities that face veteran developers in open source communities. We envision two areas of future research. The first is replication studies to generalize and refine our results. The second is to use our understanding of the problem to enact change. As one example, identifying the motivations and skill sets of veteran developers who participate in OSS may indicate steps to take to draw in present non-participants.

Through one lens, this paper is about open source, but through another, it represents a wider inquiry into the challenges and opportunities that developers face as they get older. For instance, a broadening range of skills may also have implications for other areas of software engineering, such as in startup companies and for recruitment and retention. Overall, a better understanding the how developer's motivations and skills change will enable the community to make software that is designed by and for people of all kinds.

VIII. Acknowledgments

Thanks to Jennifer Davidson, Jim Matlock, Steve McDonald, Patrick Wagstrom, James Witschey, and anonymous reviewers for their suggestions. This material is based upon work supported by the National Science Foundation under Grant No. 1252995.

REFERENCES

[1] P. A. Wagstrom, "Vertical communication in open software engineering communities," Ph.D. dissertation, Carnegie Mellon University, 2009.

[2] R. A. Ghosh, "Understanding free software developers: Findings from the FLOSS study," in *Perspectives on Free and Open Source Software*. The MIT Press, Jun. 2005.

[3] P. David, A. Waterman, and S. Arora, "FLOSS-US: The free/libre/open source software survey for 2003," http://www.stanford.edu/group/floss-us/.

[4] G. Robles, L. A. Reina, A. Serebrenik, B. Vasilescu, and J. M. González-Barahona, "Floss 2013: a survey dataset about free software contributors: challenges for curating, sharing, and combining." in *MSR*, 2014, pp. 396–399.

[5] G. Convertino, U. Farooq, M. B. Rosson, J. M. Carroll, and B. J. Meyer, "Supporting intergenerational groups in computer-supported cooperative work $(cscw)$," *Behaviour & Information Technology*, vol. 26, no. 4, pp. 275–285, 2007.

[6] D. Stein and T. Rocco, "The older worker. myths and realities." 2001.

[7] *Occupational Outlook Handbook*. US Dept of Labor, 2013, ch. Computer Software Engineers and Computer Programmers.

[8] A. Hars and S. Ou, "Working for free? Motivations for participating in open-source projects," *International Journal of Electronic Commerce*, vol. 6, pp. 25–39, 2002.

[9] J. L. Davidson, U. A. Mannan, R. Naik, I. Dua, and C. Jensen, "Older adults and free/open source software: A diary study of first-time contributors," in *Proceedings of The International Symposium on Open Collaboration*, ser. OpenSym '14, 2014, pp. 5:1–5:10.

[10] J. L. Davidson, R. Naik, U. A. Mannan, A. Azarbakht, and C. Jensen, "On older adults in free/open source software: reflections of contributors and community leaders," in *Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2014, pp. 93–100.

[11] L. Brooke, "Prolonging the careers of older information technology workers: continuity, exit or retirement transitions?" *Ageing & Society*, vol. 29, no. 02, pp. 237–256, 2009.

[12] U. Lünstroth, *Shaping Better Technologies*. Lit Verlag, 2007, ch. Demography, Aging, and High-Tech - The Case of Software Developers.

[13] P. Moen, "Midcourse: Navigating retirement and a new life stage," in *Handbook of the Life Course*. Springer, 2006, pp. 269–291.

[14] M. Arthur and D. Rousseau, *The Boundaryless Career: A New Employment Principle for a New Organizational Era*, Sep. 1996.

[15] P. Moen, "Beyond the career mystique: "time in," "time out," and "second acts"," *Sociological Forum*, vol. 20, no. 2, pp. 189–208, 2005.

[16] S. McDonald and C. A. Mair, "Social capital across the life course: Age and gendered patterns of network resources," *Sociological Forum*, vol. 25, pp. 335–359(25), June 2010.

[17] A. J. Stewart and E. A. Vandewater, ""If I had it to do over again...": Midlife review, midcourse corrections, and women's well-being in midlife." *Journal of Personality and Social Psychology*, vol. 76, no. 2, pp. 270–283, 1999.

[18] P. Moen and V. Fields, "Midcourse in the United States: Does unpaid community participation replace paid work?" *Ageing International*, vol. 27, pp. 21–48(28), 1 July 2002.

[19] M. Lovden, P. Ghisletta, and U. Lindenberger, "Cognition in the Berlin Aging Study (BASE): The first 10 years," *Aging, Neuropsychology, and Cognition*, vol. 11, no. 2-3, pp. 104–133, June 2004.

[20] N. Eisenberger, S. Taylor, S. Gable, C. Hilmert, and M. Lieberman, "Neural pathways link social support to attenuated neuroendocrine stress responses," *NeuroImage*, vol. 35, no. 4, pp. 1601 – 1612, 2007.

[21] H. Maier and P. L. Klumb, "Social participation and survival at older ages: is the effect driven by activity content or context?" *European Journal of Ageing*, 2005.

[22] N. Ducheneaut, "Socialization in an open source software community: A socio-technical analysis," *Computer Supported Cooperative Work*, vol. 14, pp. 323–368, 2005.

[23] S. Berenson, K. Slaten, L. Williams, and C.-W. Ho, "Voices of women in a software engineering course: reflections on collaboration," *Journal of Educational Resource Computing*, vol. 4, March 2004.

[24] S. Arun and T. Arun, "ICTs, gender and development: women in software production in kerala," *Journal of International Development*, 2002.

[25] V. Grigoreanu, J. Cao, T. Kulesza, C. Bogart, K. Rector, M. Burnett, and S. Wiedenbeck, "Can feature design reduce the gender gap in end-user software development environments?" in *Visual Languages and Human-Centric Computing*. IEEE, 2008, pp. 149–156.

[26] C. Huff, "Gender, software design, and occupational equity," *SIGCSE Bulletin*, vol. 34, pp. 112–115, June 2002.

[27] E. Ruiz Ben, "Defining expertise in software development while doing gender," *Gender, Work and Organization*, vol. 14, pp. 312–332(21), July 2007.

[28] V. Bogdan, D. Posnett, B. Ray, M. v. d. Brand, Filkov, A. Serebrenik, D. Premkumar, and V. Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the Conference on Human Factors in Computing Systems*. ACM, 2015.

[29] D. Stewart and P. Shamdasani, "Focus group research: Exploration and discovery," in *Handbook of applied social research methods*. Sage Publications, 1998, pp. 505–526.

[30] C. Teddlie and F. Yu, "Mixed methods sampling a typology with examples," *Journal of Mixed Methods Research*, vol. 1, no. 1, pp. 77–100, 2007.

[31] A. Bonaccorsi and C. Rossi Lamastra, "Altruistic individuals, selfish firms? the structure of motivation in open source software," *First Monday*, vol. 9, no. 1, 2004.

[32] D. Huffaker and J. Lai, "Motivating online expertise – sharing for informal learning: The influence of age and tenure in knowledge organizations," *International Conference on Advanced Learning Technologies*, pp. 595–599, 2007.

[33] F. Blanchard-Fields, A. Mienaltowski, and R. B. Seay, "Age differences in everyday problem-solving effectiveness: Older adults select more effective strategies for interpersonal problems," *The Journals of Gerontology. Series B: Psychological Sciences and Social Sciences*, vol. 62, no. 1, pp. 61–64, 2007.

[34] A. W. Woolley, C. Chabris, A. Pentland, N. Hashmi, and T. Malone, "Evidence for a collective intelligence factor in the performance of human groups," *Science*, vol. 330, no. 6004, pp. 686–688, 2010.

[35] G. Labouvie-Vief, "Cognitive-emotional integration in adulthood," *Annual review of gerontology and geriatrics: Focus on emotion and adult development*, vol. 17, pp. 206–237, 1997.

[36] L. Carstensen, D. Isaacowitz, and S. Charles, "Taking time seriously. a theory of socioemotional selectivity." *The American psychologist*, vol. 54, no. 3, pp. 165–181, Mar. 1999.

[37] N. Ellison, C. Steinfield, and C. Lampe, "The benefits of Facebook "friends:" social capital and college students' use of online social network sites," *Journal of Computer-Mediated Communication*, vol. 12, no. 4, pp. 1143–1168, 2007.

[38] R. Lowe and G. Ritchey, "Relation of altruism to age, social class, and ethnic identity," *Psychological Reports*, vol. 33, no. 2, pp. 567–572, 1973.

[39] P. Morrison and E. Murphy-Hill, "Is programming knowledge related to age?" in *Companion to the Working Conference on Mining Software Repositories*, 2013, pp. 1–4.

[40] R. D. Roberts, "Measuring emotional intelligence," *Oxford Handbook of Methods in Positive Psychology*, p. 189, 2006.

[41] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2001, vol. 2.

[42] T. Meeks and D. Jeste, "Neurobiology of wisdom: A literature overview," *Archives of general psychiatry*, vol. 66, no. 4, pp. 355–365, 2009.

[43] M. de Choudhury, M. Gamon, and S. Counts, "Happy, nervous or surprised? classification of human affective states in social media," in *AAAI Weblogs and Social Media*, 2012, pp. 435–438.

[44] S. Grant and B. Betts, "Encouraging user behaviour with achievements: an empirical study," in *Proceedings of the Tenth International Workshop on Mining Software Repositories*. IEEE Press, 2013, pp. 65–68.

[45] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2857–2866.

[46] D. Geer, "Software developer profession expanding," *IEEE Software*, vol. 23, pp. 112–115, 2006.

[47] J. Cohoon, "Toward improving female retention in the computer science major," *Communications of the ACM*, vol. 44, pp. 108–114, May 2001.

[48] V. Sinha, S. Mani, and S. Sinha, "Entering the circle of trust: developer initiation as committers in open-source projects," in *Proceedings of the Working Conf. on Mining Software Repositories*, 2011, pp. 133–142.

[49] C. Jensen and W. Scacchi, "Role migration and advancement processes in OSSD projects: A comparative case study," in *Proceedings of the International Conference on Software Engineering*, 2007, pp. 364–374.

[50] D. Nafus, J. Leach, and B. Krieger, "Free/libre and open source software: Policy support. gender: Integrated report of findings," University of Cambridge, Tech. Rep., 2006, deliverable D 16.