Hiring in the Global Stage: Profiles of Online Contributions

Anita Sarma
Oregon State University
Corvallis, OR, USA
anita.sarma@oregonstate.edu

Xiaofan Chen
Newnerdy Enterprise Group Limited,
Auckland, New Zealand
xiaofanchen@hotmail.com

Sandeep Kuttal
University of Tulsa,
Tulsa, OK, USA
Sandeep-kuttal@utulsa.edu

Laura Dabbish
Carnegie Mellon University,
Pittsburgh, PA, USA
dabbish@cs.cmu.edu

Zhendong Wang
University of Nebraska,
Lincoln, NE, USA
zrwang@cse.unl.edu

Abstract - **Managers are increasingly using online contributions to make hiring decisions. However, it is nontrivial to find the relevant information of candidates in large online, global communities. We present Visual Resume, a novel tool that aggregates information on contributions across two different types of peer production sites (a code hosting site and a technical Q&A forum). Visual Resume displays summaries of developers' contributions, and allows easy access to contribution details. It also facilitates pairwise comparisons of candidates through a card-based design. Our study, involving participants from global organizations or corporations that draw from the global community, indicates that Visual Resume facilitated hiring decisions; both technical and soft skills were important when making these decisions.**

Index Terms: Developer profiles; online contributions; hiring decisions.

I. Introduction

The growing popularity of online peer production in the past few years is slowly changing the hiring process. Potential employers, as well as recruiters are increasingly using the history of public contributions to locate suitable candidates, filter through applicants for a position, or inform interview interactions [28]. John Resig, the creator of jQuery, tweeted: "*When it comes to hiring, I'll take a Github commit log over a resume any day*."[26]. Likewise, Barney Pell, the founder of Powerset said, "*Online open-source communities like GitHub bring large numbers of developers together and are thus a natural place for recruiting*."[31]

In addition to the technical skills, the "soft" skills of a developer such as motivation and collaboration competency are important when making hiring decisions [16][17]. This is especially true in open source development and global organizations, where contributors and employees come from different countries and cultures [1][2]. In such situations, it is important to identify how well the candidate can fit in the organization culture [7]. Different countries have different social norms, which can affect the interaction and contribution styles of an individual [2][7]. Moreover, because of different emphasis placed on different skill characteristics, resumes and skills highlighted by individuals from different countries may look very different when presented in static resumes or career sites.

The level of transparency afforded by online project hosting sites, such as GitHub or BitBucket, allows access to details about developers' activities: lines of code committed, issues resolved, and interactions and discussions around the code that the developer participated in. Managers and developers have been found to use this information to assess new employees or their colleagues [18][28]. For example, by using activity traces from online sites, one can reconstruct what someone works on, what their code looks like, how they work and interact, and their speed of work [19]. The motivation/ passion of a developer can be assessed based on the projects that she owns, has contributed to, or the diversity in the types of project or languages in which she is involved [18][28]. Similarly, an assessment of whether someone is a team player can be made from discussions about issues as they represent how developers talk about their work or negotiate changes to their project.

However, there are challenges in using online contributions when evaluating potential job candidate. A key challenge is the amount of information that needs to be processed. Developers' contributions may be distributed across multiple projects or even different types of archives (e.g., code hosting sites vs. technical Question and Answer (Q&A) forums). Potential employers, who have limited time and many applicants to review, are unlikely to spend copious amounts of time searching different online archives or even searching a single archive in depth. Marlow and Dabbish [18] found that employers assessed those online activities that require less effort to evaluate. In majority of cases, managers did not investigate the contribution or interaction details, and instead focused only on aggregate amounts of activity. This was despite them mentioning that interaction style and type of contributions were important factors when making hiring decisions.

To address this problem, we have designed a novel tool, Visual Resume that aggregates activity traces of developers across different types of contributions and repositories into a single developer profile. Visual Resume, goes well beyond the current state-of-the-art aggregator sites by: (1) aggregating data across two different types of peer-production sites, (2) creating profiles that not only provide overviews of activity summaries, but also allow deeper exploration of contributions that are contextualized and easy to access, and (3) specifically allowing side-by-side comparison of contributors and different types of contributions.

We evaluated Visual Resume through a scenario-based study with managers and other personnel from global organizations and/or corporations that draw from the global commu-

2329-6313/16 $31.00 © 2016 IEEE
DOI 10.1109/ICGSE.2016.35

nity. Our study found that both technical and soft skills are equally important for making hiring decisions. Our participants used activity summaries and endorsements to judge technical breadth. Finally, our study showed that participants not only used cues that were identified in previous research, but also used additional cues (e.g. code review, commits vs. comments, answers vs. questions) in combination to compare candidates in terms of motivation, coding ability, collaboration competency, and management.

II. Background

The technical and social skills of a developer are crucial in determining their effectiveness in a team. This is especially true in a global setting [2][7][23], where differences in organizational culture can make building a working (trust) relationship difficult [9]. Numerous studies have found that trust among team members allows them to work effectively [1][27]. Past interviews of respondents in GSD settings have found that the characteristics that people look for when evaluating the fit of a new team member, and if they can be trusted are: technical competency, collaboration and communication proficiency, and how passionate they are about the project [1][2].

Similarly, research on recruitment in online communities has found technical and social skills to be important [6][17][18][21][28]). Here, we summarize results on skill set evaluations from these studies (see Table 1).

Technical Skills can be further divided into two areas.

Coding competency: Primary qualifications of any software developer are their programming knowledge and coding ability. The level and amount of past activities (in a project or programming language) indicates the experience level of an individual [2][19]. Managers seek the following cues from an online environment: (1) owned and forked projects, (2) frequent contributions to projects, such as providing commits or answering questions, and (3) the number of languages in which a candidate is proficient.

Quality of work: The amount of contributions by an individual cannot be the sole evaluation criterion. It is important to also understand the quality of the work, which can signal a candidate's competence and skill level [18]. What signals quality is a subjective measure, and can range from code reviews to test coverage metrics. While, the actual lines-of-code produced (or the post) is most frequently used, other criteria exist. For example, contributions that include test cases can indicate a well-thought out contribution [28][33]. Similarly, information about whether the community has accepted a candidate's work (e.g. commits) can also indicate the quality of the work [18, 21].

Soft Skills of a developer can be divided into three categories.

Collaboration proficiency: Key criteria when deciding to collaborate or trust others include factors such as, whether the person is polite or arrogant, whether they are willing to help others and provide sufficient context to make the solution useful [1]. Interaction histories of an individual can indicate her willingness, and how she is like to work with [12][19][40]. Developer activities serving as cues for (positive) interactions are: (1) comments regarding issues, (2) answers or questions submitted in Q&A forums, and (3) details about the nature of these activities, such as whether developers provide polite, articulate and helpful answers. Endorsements also can be used as a proxy to assess collaboration ability [28].

Project management ability: Managers prefer candidates who have some management skills, as it shows their ability to manage their own work [34]. When someone owns a project, they need to set the project's overall design, manage incoming contributions, and interact with potential contributors [18].

TABLE I. CANDIDATE QUALITIES AND ACTIVITY TRACES

Quality Inferred	Activity Traces that serve as cues for skills
Technical Skills	
Coding competency	- Level and amount of past visible activity - Number of project owned or forked - Number of commits/issues/comments - Frequency of commit/issues/comments - Languages used
Quality of work	- Content of the contribution - Community acceptance of the work - Number of accepted commits/answers - Test case inclusion
Soft skills	
Collaboration proficiency	- Visible communication activity - Number of comments/answers/questions - Types of comments/answers/questions - Endorsements of contributions - Number of followers - Reputation scores
Project management ability	- Number of projects owned
Motivation/passion	- Recency and volume of commits/issues/comments - Number of commits/issues/comments - Recency of commits/issues/comments contributions - Number of non-work-related side projects - Diversity of skills - Number of languages - Number of contributed projects

Motivation/ Passion: An important trait of volunteer contributors is their passion in their projects. Studies [2][32][38] show that motivation and performance are deeply connected; highly motivated individuals are more likely to perform better and influence future engagement in projects. It has also been found that developers are likely to trust others if they are passionate about their work. A developer's motivation can be indicated by: (1) the recency and volume of activities (e.g. commits, issues, comments) across projects, (2) the number of owned or forked projects, which are not directly related to the developer's own work, but are done as a hobby or for fun [18], (3) diversity in languages that a developer is comfortable with, and diversity in projects (e.g., different technologies and programming languages) [28].

Research has found that employers and developers have started using online project hosting sites to evaluate job candidates [18] or their colleagues [28][33]. Since contributions in online peer production sites are archived and maintained by a third party, they are seen as assessment signals that is hard to fake by candidates [32], and are preferred over static resumes or code samples taken out of context [18] [28].

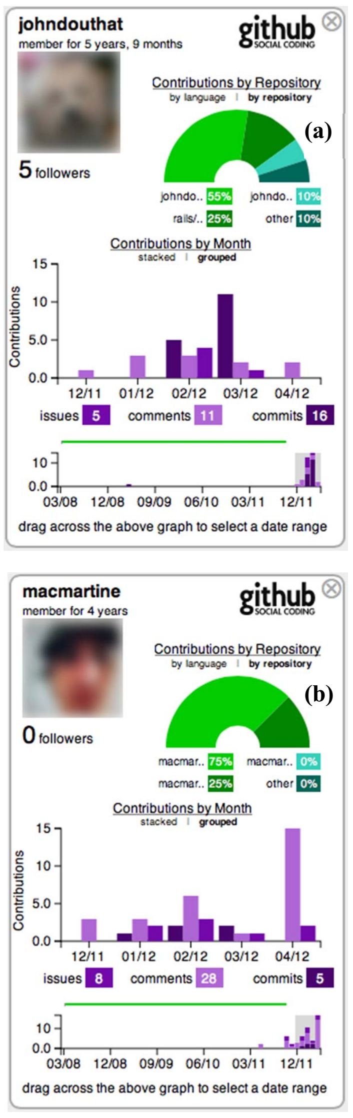
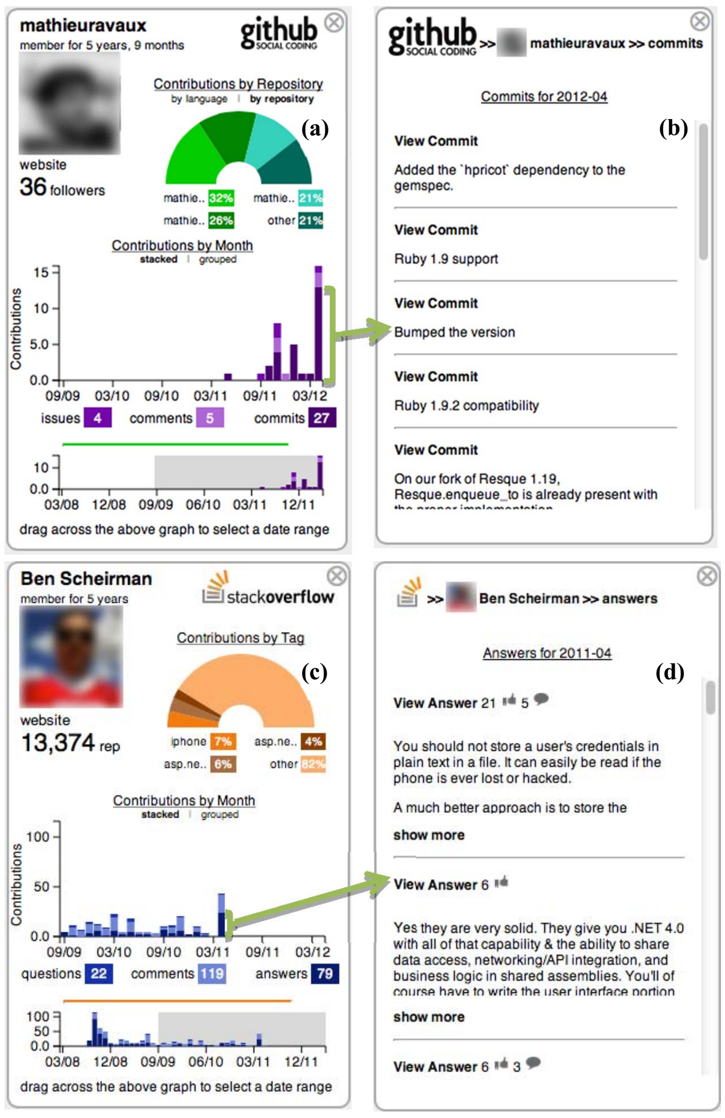


Fig. 1. Tiles showing side-by-side comparison of GitHub tiles for two candidates: (a)johndouthat and (b) macmartine.



View Commit

Added the `hpricot` dependency to the gemspec.

View Commit

Ruby 1.9 support

View Commit

Bumped the version

View Commit

Ruby 1.9.2 compatibility

View Commit

On our fork of Resque 1.19, Resque.enqueue_to is already present with

View Answer 21 5

You should not store a user's credentials in plain text in a file. It can easily be read if the phone is ever lost or hacked.

A much better approach is to store the

show more

View Answer 6

Yes they are very solid. They give you .NET 4.0 with all of that capability & the ability to share data access, networking/API integration, and business logic in shared assemblies. You'll of course have to write the user interface portion

show more

View Answer 6 3

Fig. 2. Drill-down functionality for (a) GitHub tile and (b) its commits; and for (c) Stack Overflow tile and (d) its answers. Click on "View Commit"/ "View Answer" opens the post in its parent site.

However, the amount of effort that is required to reliably assess the skills of a developer using online activities is non trivial. For example, to evaluate the lines of code in a commit in GitHub, one has to first identify the projects to which a developer has contributed from their profile, and then navigate to specific projects. Once on the project page, one has to scroll through all the commits in the project to find the commits by the developer by recognizing the user-id. Clicking on the commit-id takes the user to a page where the lines of code changed are listed. If there are multiple commits, one has to keep scrolling through the list to identify the commits from a particular developer.

Marlow and Dabbish [18] found that evaluators do not assess the actual lines of code changed, but instead used their perception of the reputation of the project to which a contribution has been made. This was common when a contribution to a high-status project was not in the top recent activities in the project, as this entailed scrolling through hundreds of commits to identify the commit from the developer. Assessing developer interactions requires even more effort. To evaluate the discussions around a code snippet one has to first identify the pull request associated with the commit (and the lines of code) and then read the discussion around it. In fact, most evaluators did not assess information regarding developer interactions when forming impressions for hiring in GitHub [18].

III. Visual Resume

The design of a tool that allows in depth assessment of a developer's skills from traces of online activities requires us to first answer two key questions:

Question 1: What information should we display? Past studies (Table I) have found that both technical and soft skills are important in a candidate; and developers contribute to multiple projects and different types of content. While a few initial aggregator sites exist [28], they only provide activity overviews: projects and programming languages that a developer has contributed to or own, and overall commits. To assess contribution details, one still needs to perform all the manual effort to navigate to the developer's profile or the project page.

Question 2: How should we present information that is contextualized to the project, easy to access, and allows comparison? Typically, employers first screen developers based on the amounts of contributions, and then perform detailed comparison on a subset of developers [30]. This indicates that profiles not only need to present a high-level overview, but also allow easy access to details about contributions in a manner that facilitates comparison across candidates.

A. *Design Decisions*

Visual Resume aggregates activity traces across different types of peer production sites: code hosting sites (e.g., GitHub) and technical Q&A forums (e.g., Stack Overflow). Visual Resume aggregates developer interactions across projects and languages to portray developers' technical and soft skills (see Fig. 2(a)(c))[1]. More specifically, it extracts histories of commits, issues, comments, programming languages, and projects in GitHub. For Stack Overflow, it groups data on answers, questions, comments, and tags. Aggregating activities across online communities can build a more accurate profile [19][39] as developers can contribute to multiple projects and forums. This also helps in comparison of contributions across sites as different sites can have different presentation styles.

Visual Resume treats technical and soft skills as first order entities. For instance, it provides summaries of, and easy access to technical information (code contributions), as well as cues denoting soft skills (comments on issues in GitHub, questions and answers in Stack Overflow).

It enables evaluation of contributions by allowing users to quickly drill down to a specific item. The quality of code can be assessed from the lines of code changed, as it reflects not only the complexity of the change [32], but also the coding style of a developer. Other metrics such as whether pull requests (or commits) were accepted (or merged) or the centrality or complexity of a contribution can also signal quality. Employers resort to using other activity traces as a proxy for quality when accessing such information is effortful. Visual Resume makes accessing this information straightforward. For Stack Overflow, we present information about the number of "upvotes" for a question or answer, and whether the answer was "accepted" – customary quality cues in Stack Overflow.

Visual Resume presents the social standing of users in a community. Employers often use endorsements in the community as a proxy for quality of work [28]. Some proxies of quality are: the reputation of the project to which a contribution has been made [28], followers in GitHub [32], or reputation points in Stack Overflow [21][37].

[1] web.engr.oregonstate.edu/~sarmaa/visual-resume/ has a demo video.

B. *User Interface*

Visual Resume provides concise summary information about developers' activities to allow quick assessment of skills and facilitate comparison across candidates. We represent a candidate's information through "tiles" – small cards that are 300 x 500 pixels. These tiles evoke the notion of business cards, and allow quick side-by-side comparison of developers – an important recruitment strategy [14][30].

Fig. 1(a) shows a candidate's GitHub (GH) tile, displaying his activity summary. The left top of the tile shows his profile information – GitHub ID, picture, tenure in the site, their personal website(s) or blogs, and number of followers. A user can click and navigate to each item in the profile.

In the rest of the tile, contributions are summarized and visually presented as project managers favor information that take less effort to verify [18]. While details of contributions are the final criteria that shape evaluation outcomes [19], they are harder to access. For example, to assess coding ability, an employer may want to look at the source code and its style, but doing so may require scrolling through numerous pages of commit history in a (highly active) project to get to the specific contribution. Such detailed assessment for all contributions of a developer or for all candidates is infeasible.

To overcome this problem, Visual Resume allows easy access to contribution details. At the right top of the tile, a radial chart provides a breakdown of the repositories in which the person is most involved, based on the number of their commits, comments, and issues in the repository. Hovering over a slice of the chart, presents the repository name, the main programming language of the repository, and the number of watchers. A user can click on a slice and "drill down" to examine the contributions of the developer unique to that specific repository. Clicking on a slice in the radial chart opens a new tile, which is similar to the GH tile (and therefore not shown in Fig. 1); but the difference is that the tile now shows information pertaining to the selected repository. If "contributions by language" is selected, then the radial chart changes to show the breakdown of the programming languages of the contributions by the candidate. In this case, the slices in the chart aggregate activities (commits, comments, and issues) across all projects that use the same programming language.

In the lower half of the GH tile, bar charts summarizes a user's contributions: commits, issues, and comments across projects on a monthly basis. The chart at the bottom of the card presents the entire history of the candidate, from which a user can select a specific date range. Doing so populates the top bar chart with details in the selected time period.

Contributions are displayed either as a stacked bar chart (Fig. 2(a)) or a grouped bar chart (Fig. 1(a)). In the former, all types of contributions are stacked into a single bar, whereas in the latter, each contribution type gets a bar. These charts can help portray contribution patterns or trajectories. For example, if a developer has become more active in a project, the stacked bar chart easily shows this increase, regardless of the type of contribution. However, if one wants to track the activity levels of a specific type of contribution (e.g., commits), then the grouped bar chart is a better option. Hovering over a segment in the chart displays the number of the contribution for the month in a popup. Clicking a bar (segment) in the bar chart opens a new (drill-down) tile that displays detailed information of the type of contribution. For example, if the "commit" segment of the rightmost bar in the (stacked) bar chart is clicked (see Fig. 2(a)), a new tile opens listing commits and an excerpt of the commit comment (see Fig. 2(b)). We opted to display the commit message instead of the commit-id since it can provide some idea about the commit. A user can click the "View Commit" link to further investigate a contribution, for example a commit. Doing so takes the user to the GitHub page where the commit and the lines of code changed is available.

Fig. 2(c) shows a Stack Overflow (SO) tile. This tile is very similar to the GH tile. The left top of the SO tile shows the tenure and reputation score in Stack Overflow. The radial chart gives a breakdown of the various tags (programming languages, concepts, etc.) with which the person is most involved based on their number of questions, comments, and answers in that tag. Bar charts show the individual's contributions (in terms of questions, answers, and comments) on a monthly basis. Clicking on a specific activity bar opens a new tile (Fig. 2(d)), which lists the contributions (here we are seeing a list of answers). The answers include annotations about the number of "upvotes", whether it was accepted ("thumbs up" sign), and the number of comments associated with that answer. Users can drill down to view the full answer, its associated question and comments by clicking on the "View Answer" link, which takes them to the Stack Overflow page.

Each person's activity is shown in a tile, which can be closed or rearranged by simply picking and moving a card across the screen. The cards allow easy <u>pairwise comparison</u>. Viewers can compare contributions between multiple users or for the same user in different contexts. For example, users can compare the GitHub contributions between two developers (Fig. 1) or the contributions across different sites (Fig. 2), or across projects or programming languages.

C. *Implementation Details*

Visual Resume is designed as a web application so that there is no need for installing it at the client site. It follows a 4-step approach: *collect*, *process*, *filter*, and *visualize*. The former two steps are performed at the server side, and require a wrapper for each repository; the latter two are part of a rich web client that uses a model-view-controller architecture.

Collect: Visual Resume can collect data across different repositories. Each data source requires a specific extractor that collects and stores the data in our database. Currently, we have implemented extractors for two different types of peer production sites: GitHub and Stack Overflow [37]. Extractors for other sites can be easily designed. We need site-specific extractors since data from each site are accessible in different ways. For example, the extractor for GitHub uses the GitHub API. The GitHub API only allows 5000 requests per hour when using basic authentication, we need to periodically extract the data and incrementally update the database. Whereas, Stack Overflow provides periodic data dumps of the entire history. The extractor needs to identify the "new" data from the dump to update the database.

Process: This step has three functionalities. First, it transforms data collected in different formats into a uniform format (we use Neo4j – a graph database [22]). Second, it creates a data model designed to generalize across different types of project hosting and Q&A sites. New sources of information (discussion in mailing lists vs. comments on an issue) can be easily added to the schema. The data is linked such that aggregations and queries can be performed per user, per repository, per language, per tag, etc. Other pertinent information (personal page, blogs) available from profiles in GitHub or Stack Overflow are also linked. This model is then encoded as a JSON file.

Visualize: The visualization is created by using the d3.js framework [6]. Currently, our tile template uses a top-down layout. It uses "label", "radial chart", and "bar chart" widgets to display aggregated data. Different templates that use other layout or widgets can be easily implemented and incorporated.

Filter: Different filters can be used to adjust the amount of information presented to the user. A basic filter that we have currently implemented is time period selection. We can easily create other filters that adjust other kinds of information (e.g., the amounts or types of contributions).

IV. User study

We conducted a scenario-based, task analysis study to understand how participants use the information provided by the tool, and what other information they seek when they have to choose from a set of potential job candidates. The goal was not to see if all participants converged on the same candidate, since this is a subjective decision; instead, we observed how participants arrived at their decisions.

Study Design: We first conducted interviews with two industry contacts – team leads with extensive experience in hiring. We asked them about current practices when making hiring decisions and how they evaluated online contributions. We then asked them to select the most relevant cues from Table I. Based on their feedback we designed the assessment task to have two phases – an initial filtering phase (Phase I), followed by an in depth analysis of two candidates (Phase II). The interviews also informed the dimensions in which a candidate was evaluated in phase II[2].

Our study included nine participants, who were asked to evaluate the suitability of a pool of candidates in response to a prototypical job advertisement that we created. We asked participants to think aloud during the study, where they verbalized their actions and the intention behind the actions [10][36]. We screen recorded the participant actions and collected their feedback through an exit interview.

<u>*Task description*</u>: Participants were told that they were the technical lead and had to assess a set of candidates in response to a job advertisement. We created the job description to represent a typical posting for web development positions. To do so, we reviewed postings on popular job posting sites, such as LinkedIn, Careers 2.0 [3], and CareerBuilder [4]. We included

[2] Visit http://web.engr.oregonstate.edu/~sarmaa/visual-resume/ for user study details, including the job posting given to participants.

aspects that typically appeared across many postings, including: position description, job responsibilities, required qualifications, and benefits.

Participants evaluated candidates in two phases: Phase I – a filtering task where they selected two candidates from a pool of five, and Phase-II – a focused comparison of the two selected candidates. The filtering task was designed to solicit higher-level comparisons across a set of candidates. This type of screening is common in the early stage of job candidate evaluation [30]. Next, participants performed focused comparisons of their two "best" candidates across four dimensions: *motivation*, *coding ability*, *collaboration potential*, and *project management abilities*. Participants were then asked to select one of the two candidates along each of these dimensions.

Job candidate selection: We selected five potential job candidates for the study. These candidates were selected to represent typical GitHub and Stack Overflow users with some expertise in the topic areas indicated in the job description (Ruby and Java Script). To identify these candidates, we first extracted a dataset of GitHub participants with at least one commit to the Ruby on Rails project on GitHub. Next we queried Stack Overflow for these users to identify a subset of users with profiles and activity in the site. From this sample of 2300 common users in both communities, we then identified candidates who had reasonable monthly activity on both sites (240 users). From this set, we randomly selected the five candidates for the study.

Study Participants: We selected nine participants in our study who had some experience in the hiring process. We also recruited participants to obtain both corporate (P1-P3) and small software company (P4-P9) participants, since practices in established corporations may vary from those used in lean startup operations. Our participants included those who hire for their teams as well as those who interview their peer developers. Table II summarizes our participants' background.

TABLE II. BACKGROUND OF STUDY PARTICIPANTS

Code	Background
P1	Senior architect at a large software development firm. Involved in hiring and interview process. Experience in hiring 15 developers and 5 interns.
P2	Senior manager at a large software development company. Involved in hiring process. Experience in hiring 8-10 developers.
P3	Project leader of a small team in a mid-sized company. Involved in interviewing peer developers to provide hiring recommendation.
P4	Senior member of a student-led organization for incubating new project ideas and connecting with local industry. Involved in hiring developers.
P5	Founder and primary developers of a startup. Involved in interviewing and hiring subcontractors for development and UI design jobs.
P6	Web designer and hiring person for a non-profit organization. Involved in hiring peer developers and sub-contractors.
P7	Front-end web developer at a web startup. Involved in hiring and interview process for recommendations though not primarily responsible.
P8	CTO of a web startup. Experience in hiring 10 engineers at the company.
P9	Front-end developer at a small technology startup. Experience in hiring 6 engineers for the company.

Data Analysis: To analyze the data from our study, we transcribed verbalizations and actions of our participants from observations, notes, and think-aloud data. The transcripts were transcribed using a code set related to technical and soft skills (second column of Table III and Table IV). Two researchers collaborated on the coding, until 80% agreement was reached on about 20% of the data. While analyzing the data, we found additional cues related to quality of work and social competency, which we added to our (cue) code set.

Limitations of Study: Visual Resume currently only collects activities from two sources: GitHub and Stack Overflow, and our results might not generalize to cues found in other sources. Our participants were limited to a small subset of developers who volunteered, and therefore might be biased towards assessing online contributions. Finally, we evaluated Visual Resume by using only five subject candidates. Our results regarding strategies and cues used might not hold true if the candidate pool is larger.

V. RESULTS

We found that participants used the entire set of tool features, and they were accessible and useful. In the rest of the section, we first discuss the technical and soft skills cues that were used by the participants. We then discuss how participants used the tool, followed by a discussion of the cues that were used by participants as signals for the quality of work and for collaboration competency.

A. *Popular Technical and Soft Skill Cues*

Table III summarizes the cues about technical and soft skills that participants used to select two candidates from the set of five. Table IV summarizes the cues that participants used as signals in the focused evaluation. We do not repeat the tool features that were used in this task (in Table IV) since they are the same as in Table III.

TABLE III. CUES AND FEATURES USED IN THE FILTERING TASK

Quality	Cues (# of participants)	Features
Technical skills	1. Number/frequency of commits (8)	Bar charts; Drill down; Pairwise comparison
	2. Owned/forked projects (6)	Radial charts; Drill down; External webpages; Pairwise comparison
	3. Languages/tags used (3)	Radial charts; Drill down; Pairwise comparison
	4. Nature of commits (3)	Bar charts; Drill down; External webpages
	5. Code quality (1)	Drill down; External webpages
	6. Commits vs. comments (1)	Bar charts; Drill down
	7. Centrality of commits (1)	*Unable to display*
Soft skills	1. Endorsements (5)	Description in GH/SO tiles; Pairwise comparison
	2. Nature of answers (4)	Bar charts; Drill down
	3. Number of answers (2)	Bar charts; Pairwise comparison
	4. Owned/forked projects (2)	Radial charts; Drill down; External webpages
	5. Answers vs. questions (1)	Bar charts; Drill down
	6. Nature of comments (1)	Bar charts; Drill down

Filtering Candidates (Task 1)

When evaluating technical skills, participants focused on the amount and breadth of experience in a candidate. When evaluating amount of experience, the most investigated cue was the number and frequency of commits (8 participants). This is not surprising as this was a relatively easy cue to identify. The most significant cue signalling breadth of experience was the number of owned or forked projects (6 participants). The next cue signalling breadth was the diversity in the programming languages of contributions. Participants also used the "view by language" widget in GH tile (3), while one also used the "contribution by tags" in the SO tile. At this stage, few participants tried to assess the quality of the work by reviewing the actual lines of code changed.

In contrast, participants assessed soft skills by investigating both the summaries (5 participants), as well as the quality of contributions (4 participants). This is likely because it was easier for participants to assess the quality of an answer (polite, articulate, has code snippets). Further, Stack Overflow metrics of "upvotes" and "accepts" are easily accessible.

Although all participants used technical breadth (e.g. number of activities) to screen candidates, participants from big companies (in comparison) investigated deeper to evaluate the type of contribution and its quality during assessment. None of the participants from small companies checked the nature of commits and a majority of them did not assess the nature of posts in Stack Overflow.

Focused Evaluation of Candidate Pairs (Task 2)

Participants performed pairwise comparisons of candidates across the four dimensions (Table IV, see task design website for task details). They used cues for both technical and soft skills. While coding ability and collaboration competency were judged based on technical skills, motivation and management skills were evaluated by using both types of skills.

TABLE IV. CUES USED IN THE FOCUSED EVALUATION TASK

Questions	Cues (# of participants)
Q1: Motivation	1. Number/frequency of Commits/issues (7) 2. Contribution to projects/languages/tags (6) 3. Endorsements (5) 4. Number of Answers/questions (3) 5. Nature of answers (2) 6. Nature of commits (1)
Q2: Coding ability	1. Number/frequency of Commits/issues (4) 2. Contribution to projects/languages/tags (4) 3. Endorsements (2) 4. Nature of commits (1) 5. Complexity of issues (1) 6. Accepted pull requests (1)
Q3: Collaboration	1. Contribution to projects/languages/tags (3) 2. Endorsements (3) 3. Number of Answers/questions (3) 4. Nature of answers (3) 5. Nature of comments (2)
Q4: Project management	1. Number of Commits/issues (4) 2. Owned projects (3) 3. Number of Answers/questions (1) 4. Nature of comments (1)

When evaluating motivation, a majority of participants assessed candidates based on the activities in commits and issues (7 participants), diversity in projects, programming languages used or tags (6 participants), and endorsements (4 participants). While for assessing coding ability the three main cues (used by 4 participants) were: amount of activity in commits/issues, number of owned or forked projects or number of languages/tags. Collaboration competency of a candidate was evaluated based on a set of five cues, which were used equally frequently. Project management capability was evaluated largely by observing the number of commits or issues made by the candidate (4 participants), since large amounts of activity signal involvement with the project and an interest in its growth and direction. The second most used cue was the number of projects owned or forked (3 participants). Other cues used were the number of answers/questions and comments.

Participants from large companies investigated the actual contributions when evaluating coding ability and collaboration competency. Participants from the small companies largely relied on technical breadth and endorsements.

B. Information Access

Here we discuss how participants accessed information through the feature set in Visual Resume.

Using data summaries: Participants heavily used activity summaries – the number and frequency of commits. It was the top cue followed by 8 participants. This is not surprising as this is an easy cue to investigate. Participants extensively used the radial and bar chart widgets to access contribution amounts. For example, P3 first selected two candidates who had a high reputation on Stack Overflow, but then he saw that they did not have much activity in GitHub, which made him reconsider his choice. P3 commented: *"I truly believe in Stack Overflow rating system. Can you create something like that for GitHub. If that is the case then much easier to compare across people on a given normalized scale*."

Quantifying skills: Participants used the quantification of contributions as the primary metric for the initial evaluation of candidates. More activity in GitHub was associated with better coding ability. Participant P4 commented: "*More activity in GitHub shows that the candidate has more breadth in language he knows.*" P4 also deferred to the Stack Overflow reputation scores: "*the community has a good mechanism of quantifying the contribution by taking into [account] different types of context and context of the information provided.*"

Comparing side-by-side: Five participants used the Visual Resume tiles to do a side-by-side comparison of all candidates to filter out weak candidates. As evident from P7's comment: "*Nice to be able to compare directly side-by-side and then being able to pare down further*." We found equal distribution of participants (4) who evaluated candidates one at a time and then filtered out weak candidates at the end. We found that participants were more likely to conduct pairwise comparisons to compare endorsements and the number of activities, projects, and languages between candidates.

Drilling down: All participants (8) after assessing the summarized commit data drilled down further to get a deeper view of the commits. Three of these participants went further to identify the nature of the commit (commit messages and the change set) by drilling down to the GitHub commit page to determine the nature of the contribution. We found that the drill-down feature of our approach was the most used feature, especially in the focused evaluation task, followed by bar charts, pairwise comparisons, radial charts, and external websites. This indicates that although inferring amounts of activity is easy (via bar charts), our participants were diligent and drilled down to investigate deeper qualities of the candidate, including external websites.

Visiting external sources: We found that a majority of participants (8 out of 9) assessed all aspects of a candidate. That is, they not only investigated activities from GitHub and Stack Overflow, but also consulted information in external webpages (e.g. profiles in GitHub or Stack Overflow, personal websites and blogs). We found that once participants had identified a "good" candidate, they explored the personal websites (and blogs, etc.,) to know more about the candidate. P1 commented: "*I like to see what he writes in [his] page, what parts [of contribution] is he proud of.*" Visual Resume, therefore, acted as a landing site from where different information could be accessed through a single click.

C. Signals for Quality of Work

Participants used different types of information to decipher quality. Assessing the quality of the work was important in the decision making process, as P4 mentioned "*...[while looking at the activity] this is meaningless without quality...how can he have fixed anything if there is no code in the commit [consisting of formatting changes].*"

Association with popular projects: Three participants assessed whether a candidate owned projects or contributed to other projects. Participant P2 selected the candidate who had contributed to popular projects and owned projects, "*this shows that he knows how to write good open source code.*"

Contribution type and style: Total amounts of commits were preferred over total amounts of comments. For example, participant P4 compared the number of commits to the number of comments, and selected those who had more commits. Participants also evaluated the source code to assess its style based on the code structure, variable names, and inline comments.

Comments were treated as a signal of a good documenter, and therefore a team player; P5 assessed the inline code comments to identify whether the candidate was a good documenter. He also assessed whether the candidate made comments on other developers' projects, commits, or issues.

Participants investigated whether candidates provided answers or asked questions; answers were favored over questions. They (4 participants) assessed whether the candidate was a good communicator in terms of their proficiency in the area and whether their answers were polite and articulate, easy to understand, provided enough details and context to make the solution useful, or included code snippets. For example, P4 checked the manner in which a candidate expressed himself, how he helped another person, and how he addressed the topic, and then commented "*...this guy like to teach. Gives details in answer, provides code snippet. This is good...gives nice details...I like the phrasing.*"

Commit details: Several participants identified further details on commits: (1) participant P7 indicated that the amount of code in the change set could differentiate candidates with similar number of commits, (2) participant P2 wanted to know whether the changes were central to the code base. He commented: "*It would be nice if we could see which commits actually had to do [changes] with core files.*", (3) P1 wanted to know whether the commits solved complex issues: "*Assessing coding ability is more complex – would need more information than that of just a single commit in order to see problem solving ability*", and (4) participant P7 wanted to see whether pull requests were accepted.

Tinkering vs. significant changes: Many participants investigated the contributed lines of code to identify coding ability, such as whether the commit is for adding a feature or tinkering, writing big blocks or tweaking code, or fixing configurations or dependencies. As P3 notes, "*...are they writing big blocks of code or only tweaking code, building infrastructure, adding new features or just doing tinkering stuff.*" Later he commented, "*Checking code in the commit is the most useful step when hiring someone, something not [fudged].*"

D. Signals for Social Competency

Participants inferred social competency using these cues.

Endorsements: Participants (5) made selection decisions based on community endorsements (followers in GitHub and reputation scores in Stack Overflow). As P5 commented, "*Based on GitHub followers versus Stack Overflow reputation (points), I will actually choose...*" This was because they trusted the community generated endorsements: P4 comments, "*High reputation in Stack Overflow means that the person knows what they are talking about, they help others usefully, so must know a lot*", while participant (P1) commented: "*The number of followers a user has can possibly tell us whether they are good or not, because people tend to follow those who make tangible contributions.*"

Passion for learning: Different cues were used to infer passion for learning. Most participants considered high activity in GitHub as cues for interest and motivation. Two participants considered forked projects as a key cue to evaluate a candidate's passion to learn something. However, one participant (P3) depended solely on reputations scores in Stack Overflow noting that these scores trump any other activity measures.

Management Skills: Several different cues were used to infer management skills. A cue was whether candidates owned projects (2 participants). Other cues were the number of answers/questions and comments. These cues showed whether candidates were polite and would be able to appropriately manage communication with others. One participant (P2) considered a person to be a better manager if they had more answers than questions. P1 commented: *"[candidate] was more articulate and provided more information/background/detail... suggests he is willing to help others.*"

VII. Related work

We have identified seven aggregator sites that create developer profiles. Gitto [11] and Statocat [29] generate profiles by summarizing the amounts of contributions in GitHub. Statocat also provides information on the kinds of programming languages that a developer is associated with. Careers2.0 creates developer profiles by largely extracting activities in Stack Overflow. Additionally, it presents a list of GitHub projects to which a developer has contributed. MasterBranch [20] and CoderWall [5] collect activities across code sharing sites (e.g. GitHub, BitBucket). CoderWall also awards achievement badges to developers when developers have a certain number of repositories in a specific programming language. Open Hub [24] generates profiles based on activities collected by their own code search engine (Open Hub Code search), and awards achievement badges based on the amounts of activities. Data on Talentbin [41] is sourced from profiles on Twitter, Facebook, Github, LinkedIn, Quora, and other platforms.

The majority of these aggregator tools provide overviews of contributions that typically represent technical skills. Gitto, CoderWall, MasterBranch, Talentbin and Statocat create developers' profiles based on activities in GitHub. MasterBranch, Statocat, and Open Hub provide information about the numbers of commits, contributed projects, and programming languages, but they do not display interaction histories (e.g. comments or answers or questions). CoderWall lists programming languages and project names; but it has no information about commits or comments. Gitto only represents the "contribution diagram" from GitHub.

Only few aggregator tools provide indirect mechanisms to represent social skills. CoderWall and Open Hub award achievement badges to developers who meet (site specific) criteria based on the number of contributions. Other tools link to the GitHub profile page (Gitto, Statocat) or the Stack Overflow profile page (CoderWall).

Careers 2.0 creates (high-level) resumes, listing developers' top answers and the projects in GitHub. However, it does not provide summaries of project activities in GitHub, such as commits, forks, or merges.

Moreover, these tools focus on a specific type of content management systems (e.g., code hosting site or a Q&A forum). Gitto and Statocat focus only on contributions in GitHub, whereas MasterBranch, CoderWall, and Open Hub create developer profiles that are generated by aggregating activities across multiple code sharing sites. Careers 2.0 aggregates activities across multiple Q&A sites.

Few of these tools provide mechanisms to assess the quality of contributions. Most tools (CoderWall and Gitto) link to the project page (in GitHub) via the profiles. A user can then manually investigate activities in the project (page) to identify specific commits and the code associated with the commits. Open Hub follows a similar approach. The only difference is that the project page is also hosted by Open Hub (instead of an external site such as GitHub). Statocat simply links to the GitHub developer profile, from where a user can manually investigate project contributions. Careers 2.0 presents a list of the most recent answers in Stack Overflow and displays if the answer was accepted and the number of votes for each answer.

In contrast to these tools, Visual Resume, aggregates activity across two different types of peer production sites and presents the profile in such a manner that it is easy to drill down to specific contributions and compare across candidates.

VI. Conclusions

Recruiters and team members are increasingly turning to traces of online activities to evaluate team members, since such activity are recorded in third-party peer production sites making them more trustworthy [8][14]. However, activity history of an individual can be fragmented across different projects, code repositories, and other types of content management system. Further, evaluators end up only assessing traces that are easy to evaluate, because of which evaluators do not assess contribution details in a large project, especially interaction details [18]. Therefore, we have developed a card-based, aggregator tool that improves on current aggregator sites by creating developer profiles that not only summarize activity histories from different types of peer production sites, but also allow quick drill down to specific contributions.

In global software development, virtual teams are created to leverage different skill sets internationally and team members may not have a working history. In such cases, Visual Resume can help in the creation of well-rounded, diverse teams by allowing managers to investigate different aspects of a developer's working styles and competency [13][25]. Further, different individuals often have different strengths – some have better collaboration competency, while others are technical gurus. Project risks are reduced when there is both technical and social diversity in a team [15], and is especially important in the context of virtual teams [13]. Visual Resume can be of use in such a context, since it has been designed to highlight both social and technical skills.

Finally, Visual Resume also has the potential to build rapport and trust in virtual teams as it can enable (virtual) team members to create an awareness of an individual's past working styles, technical and social competency, and passion in a project – factors that have been found to be critical in developing trust in global teams [2].

To evaluate how Visual Resume can help in recruiting/evaluating team members, we performed a user study with nine industry participants. All our participants had experience recruiting from a global community. We found that participants largely screened candidates based on amounts of contributions and then performed focused investigation by comparing contribution type and quality; P6 said that "*it is nice to be able to compare directly side by side and then being able to pare down further.*"

Participants from large companies were more likely to perform deeper investigations of contribution quality, whereas participants from smaller companies tended to accept endorsements and reputation of projects as proxies for quality. This difference could arise from the fact that the former group may have had more experience in evaluating candidates from a larger pool and had to be more selective.

Some participants felt that more explicit quality metrics could be useful. Their feedback suggests three additional cues: (1) centrality of commits, which will allow viewers to identify if commits dealt with core files, (2) complexity of commits or issues, which will indicate whether candidates' contributions were simple tweaks or dealt with complex issues or solutions, and (3) information on whether commits or pull requests were accepted, which can signal the quality of the work based on the acceptance by the community. Participants also suggested it would be helpful allow sorting candidates based on different criteria: the amounts of activities (e.g. commits, issues, comments, posts), reputation scores, and the number of followers.

We plan to extend our tool to incorporate these suggestions. When doing so, we note that information especially in the context of a global organization needs to carefully designed, such that metrics are not misused and we are cognizant of the privacy and confidentiality of participant data.

In summary the contributions of this paper are:

- First tool that aggregates activity data from project hosting site and Q&A forums. Past studies have shown that both types of information (technical and social skills) are important when assessing developers, especially when they are geographically separated. The traces afforded by Visual Resume can be used to comprehend the working styles of candidates or to build trust among virtual team members.
- First study to empirically investigate what cues managers use to assess candidates when actually performing an evaluation task. Other studies in hiring as well as GSD have been post hoc and use retrospective interviews.
- Empirical evidence of the ease of use of drill-down feature and side-by-side comparison. The drill-down feature was widely used and follows from results that report [18] that employers desire to minimize assessment effort, but consider detailed information important. Similarly, side-by-side comparison is considered useful when rapidly evaluating alternatives.

Acknowledgment

We thank Corey Jergensen and Caleb Larsen for help with the tool development. We thank the participants in our study, and NSF for the grants: IIS-1314365 and CCF-1253786.

References

[1] Ban Al-Ani and David Redmiles. 2009. In Strangers We Trust? Findings of an Empirical Study of Distributed Teams. ICGSE, 121-130.

[2] Ban Al-Ani et al., 2013. Globally distributed system developers: their trust expectations and processes. CSCW, 563-574.

[3] Careers 2.0. Retrieved May 2015 from https://careers.stackoverflow.com.

[4] CareerBuilder. Retrieved May 2015 from http://www.careerbuilder.com.

[5] CoderWall. Retrieved May 2015 from https://coderwall.com.

[6] D3.js. Retrieved May 2015 from http://d3js.org.

[7] da Silva, F.Q., C. Costa, A.C.C. França, and R. Prikladinicki. 2010. Challenges and solutions in distributed software development project management: a systematic literature review, ICGSE, 87-96

[8] Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. 2012. Social Coding in GitHub : Transparency and Collaboration in an Open Software Repository, CSCW' 12, 1277–1286.

[9] Damian, D., Izquierdo, L., Singer, J. and Kwan, I., Awareness in the wild: why communication breakdowns occur. 2007. ICGSE, 81-90.

[10] Ericsson, K. A. and Simon, H. A. 1980. Verbal Reports as Data, Psychol. Rev., vol. 87, no. 3, pp. 215–251.

[11] Gitto. Retrieved May 2015 from https://gitto.io.

[12] Gutwin, C., Penner, R. and Schneider, K. 2004. Group awareness in distributed software development. CSCW, 72-81

[13] Hofner, G. and Mani, V.S., "TAPER: A generic framework for establishing an offshore development center". 2007. ICGSE. 162-172.

[14] Jenne, K. J. and Henderson, M., 2000. Hiring a Director for a Nonprofit Agency : A Step-by-Step Guide, Pop. Gov., vol. Summer, pp. 25–36.

[15] Klein, G., Jiang, J. and Tesch, B. Wanted: Project teams with a blend of IS professional orientations. 2002. Comm. of the ACM, 45, 6, 81-87.

[16] Kristof-Brown, A., Barrick, M.R., and Franke, M. 2002. Applicant impression management: Dispositional influences and consequences for recruiter perceptions of fit and similarity. Journal of Management, 28, 1, 27–46

[17] Long, J. 2009. Open Source Software Development Experiences on the Students' Resumes: Do They Count? Insights from the Employers' Perspectives. Journal of Information Technology Education, 8, 229–242

[18] Marlow, J. and Dabbish, L. 2013. Activity traces and signals in software developer recruitment and hiring, CSCW '13, pp. 145–156.

[19] Marlow, J., Dabbish, L., and Herbsleb, J. 2013. Impression Formation in Online Peer Production : Activity Traces and Personal Profiles in GitHub, CSCW'13, 117–128.

[20] MasterBranch. Retrieved May 2015 from https://masterbranch.com.

[21] Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., and Faloutsos, C. 2013. Analysis of the reputation system and user contributions on a question answering website, ASONAM '13, 886–893.

[22] Neo4j. Retrieved May 2015 from http://www.neo4j.org.

[23] Noll, J., S. Beecham, and I. Richardson, Global software development and collaboration: barriers and solutions. 2010. ACM Inroads, Vol. 3, No. 2, 66-78.

[24] Open Hub. Retrieved May 2015 from https://www.openhub.net.25

[25] Paul, S., Samarah, I. M., Seetharaman, P., and Mykytyn, P. P. An empirical investigation of collaborative conflict management style in group support system based global virtual teams, 2004. Journal of Management Information Systems, Vol. 21, No. 3, 185-222.

[26] Resig, J. February 5, 2011. Retrieved from https://twitter.com/jeresig/status/33968704983138304

[27] Sabherwal, R., The role of trust in outsourced IS development projects. 1999. Communications of the ACM, Vol. 42, No. 2, 80-86.

[28] Singer, L. et al., 2013. Mutual Assessment in the Social Programmer Ecosystem : An Empirical Investigation of Developer Profile Aggregators, CSCW'13, 103–116.

[29] Statocat. Retrived May 2015 from http://www.upsingapore.com/ideas/statocat/.

[30] Stephan, G. and Pahnke, A. 2008. A pairwise comparison on the effectiveness of selected active labour market programmes in Germany, IAB-Discussion Pap., vol. 29/2008, pp. 1–32.

[31] Terdiman, D. August 17, 2012. Forget LinkedIn: Companies turn to GitHub to find tech talent. Retrieved from http://www.cnet.com/8301-10797_3-57495099-235.html.

[32] Tsay, J., Dabbish, L., and Herbsleb, J. D., 2013. Social media in transparent work environments, 6th CHASE, 65–72.

[33] Tsay, J., Dabbish, L., and Herbsleb, J. 2014. Influence of social and technical factors for evaluating contribution in GitHub, ICSE 2014, pp. 356–366.

[34] Upper Rio Grande. What Skills do Employers Want - Workforce Solutions. Retrieved May 2015 from http://www.urgjobs.com/pdf/Skills.pdf.

[35] Vallerand, R. J., et al.,. 2007. On the role of passion in performance., J. Pers., vol. 75, no. 3, pp. 505–33, Jun. 2007.

[36] van Someren, M. W., Barnard, Y. F., and Sandberg, J. A. C. 1994. The Think Aloud Method: A practical guide to modelling cognitive processes. London, UK: Academic Press, 1994.

[37] Vasilescu, B., Serebrenik, A., Devanbu, P. and Filkov, V. 2014. How social Q&A sites are changing knowledge sharing in open source software communities. CSCS'14, 342-354.

[38] Wu, C.-G., Gerlach, J. H., and Young, C. E. 2007. An empirical analysis of open source software developers' motivations and continuance intentions, Inf. Manag., vol. 44, no. 3, pp. 253–262, Apr. 2007.

[39] Wu, Y., Kropczynski, J., Shih, P. C. and Carroll, J. M. 2014. Exploring the ecosystem of software developers on GitHub and other platforms. CSCS Companion'14, 265-268.

[40] Zhou, M. & Mockus, A. 2012. What make long term contributors: Willingness and opportunity in OSS community, ICSE 2012, pp. 518–528.

TalentBin: Retrived May 2015 from https://www.talentbin.com/