Supporting Comparison of Developer Profiles across Online Communities

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

Anita Sarma
Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR, USA
anita.sarma@oregonstate.edu

***Abstract*— Current software development practices leave a plethora of activities that are archived in version control systems, issue trackers, mailing lists, or Question and Answer (Q&A) forums. Software managers are increasingly using these online activities to better evaluate job candidates. We introduce our tool, Visual Resume, that displays visual overviews of developers' contributions in code sharing sites (e.g. GitHub) and Q&A forums (e.g. Stack Overflow). The design of the tool is broadly applicable as an approach to summarize and display online contributions.**

I. Introduction

Current software development practices increasingly use code sharing sites and Question and Answer (Q&A) forums, which in turn archive development activities. These activities have been used to triage bugs [1][2], recommend experts for tasks [3][4], or recommend mentors to newcomers [5][6]. However, there is little research that focuses on using activities to establish a profile of developer characteristics to assist project managers in assessing the quality of developers for making hiring decisions. Since the quality of the software produced ultimately depends on the quality of the developer and software managers are busy individuals, we believe that support to help hiring decisions is important to the community.

Traditionally, paper resumes have played a central role in hiring. However, traditional resumes are losing their significance as project managers increasingly refer to a candidate's online contributions. This is likely because paper resumes only present a candidate's static and curated history.

The growing popularity of social media and online peer production in the past few years is changing the hiring process and has provided candidates with a way to publicly demonstrate their capabilities. Developers can contribute to open source software projects as well as in Q&A forums to build and advertise their technical prowess and reputation.

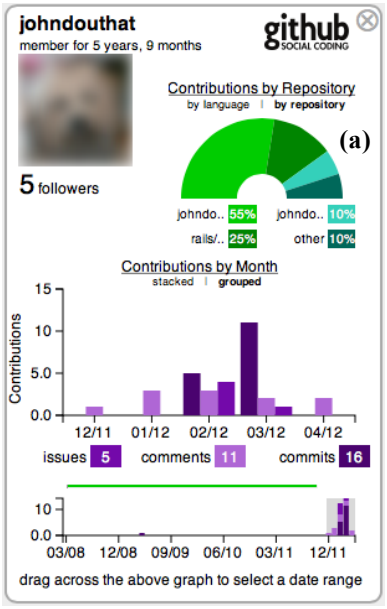
Contributions in online technical communities are archived and persistent, and in most cases constitute an assessment signal that is hard to fake [7, 9]. Therefore, project managers, individuals with limited amount of time, have started to use historical activities as an indicator of technical expertise [8].

Current resume building sites (e.g. Careers 2.0 [10], Gitto [11]) allow users to showcase their efforts from a particular site. For example Gitto aggregates data from GitHub. While useful, these sites have several deficiencies. First, each site incorporates a different design. Navigating across different sites is time-consuming. Second, different sites showcase specific information and developers' contribution history might be missed depending on which site is being showcased. Third, these sites largely present activities that signal developers' technical skills. However, social and communication skills are equally important when evaluating developer characteristics [9]. Other contribution aggregator sites, such as MasterBranch [12], CoderWall [13], and Open Hub [14], have similar problems. For example, these sites also focus on activities (e.g. commits) that depict technical skills, but fail to provide activities (e.g. comments/answers) that signal social and communication skills. Fourth, most of these sites follow a webpage like design, which is inconvenient to compare across candidates. Finally, these sites are centered from the candidate's perspective of creating a profile to showcase their "geek cred" for a particular (aggregator) site.

To alleveiate these problems, we have designed and developed an aggregator tool geared towards easy comparison of developer profiles to help in hiring decisions. We use a set of design principles and resultes from a literature survey to guide the design of our tool – Visual Resume. Visual Resume aggregates online activities to portray a developer's complete activity profile picture. It leverages basic visualization techniques to present summarized historical views of activities across different areas of a site (project, topic, etc.) and over time. The approach supports quick access to activity details through summaries and allows drill down exploration of specific types of activities. Further, the approach allows pairwise comparisons of developer activity across areas or comparisons of activity across developers. The design architecture is general enough that it can be used to display technical and social contributions from a range of different sites (e.g., BitBucket, LinkedIn).

II. Visual Resume

We have designed Visual Resume, a tool that collects activity traces from online repositories (GitHub and Stack Overflow) to create developer profiles that portray their technical and soft skills. Visual Resume follows the following design principles. First, it uses a card-based design to allow quick overview of summaries of contributions. The card-based design is reminiscent of a business card. It can also be easily viewed on mobile devices and embedded into a developers' profile page. Second, it allows a side-by-side comparison of candidates to facilitate evaluation. Third, it enables drill-down investigation of a specific activity or a data source. Fourth, it treats cues of both technical and soft skills as a first order enti-



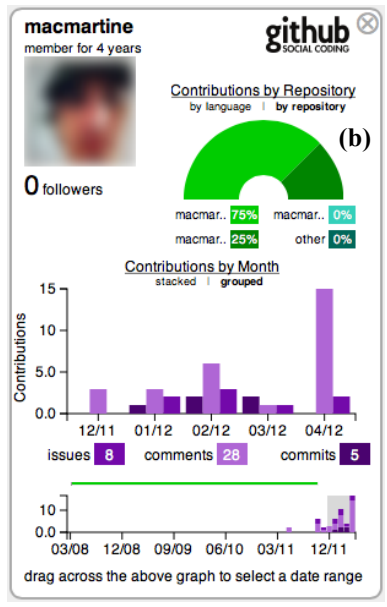
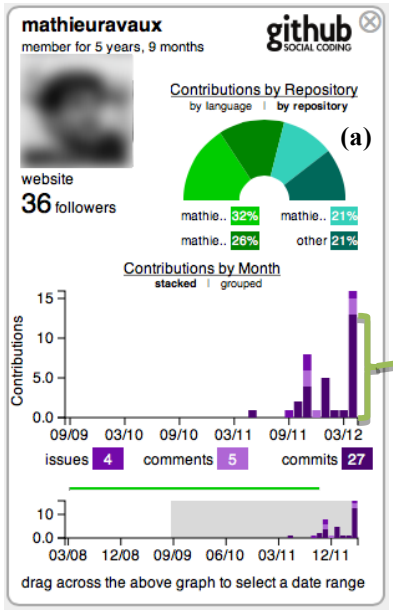
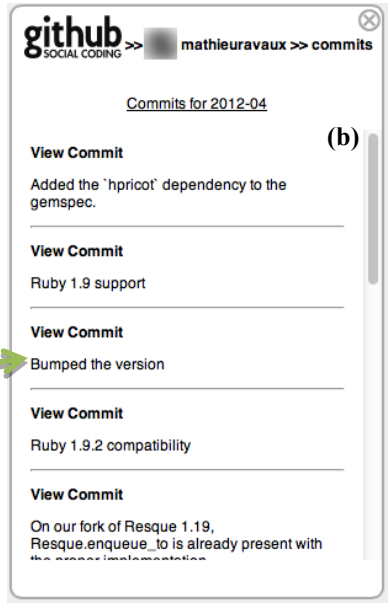
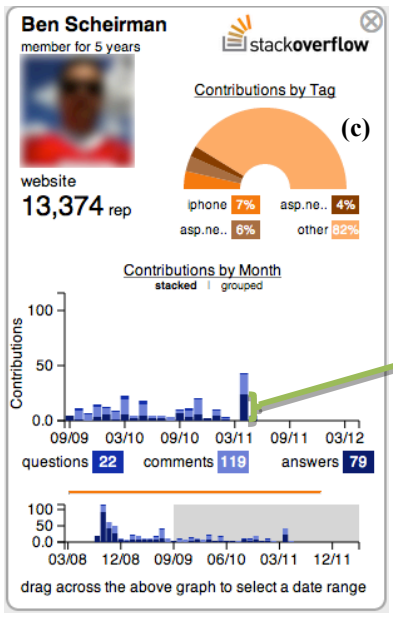


Figure 1. Tiles showing side-by-side comparison of GitHub tiles for two candidates: (a)johndouthat and (b) macmartine.







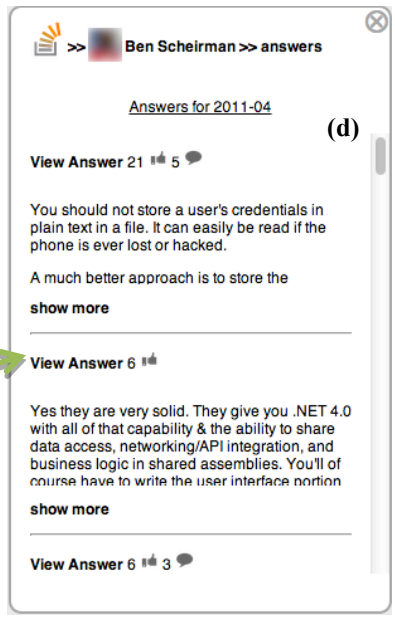


Figure 2. Drill down functionality for (a) GitHub tile and (b) its commits; and for (c) Stack Overflow tile and (d) its answers.

ty. Finally, it aggregates activity history from multiple repositories (communities) to provide a holistic view of a candidate, since a user may have different types of activities in different communities.

A. Usage Scenario

Here we discuss the interface of Visual Resume through a scenario. Let us assume that Sally, a senior tech lead, has been provided with a set of five candidates whom she needs to prioritize to bring in for interview. To review the candidates, she first opens the GitHub (GH) tiles of the first two candidates in her list for a quick *side-by-side comparison* of their *technical skills* (Fig. 1 (a) and (b)). Each tile displays an overview of activities per repository, and can be spatially rearranged.

The left top of the tile displays profile information – user ID (GitHub), picture, tenure in the site, their personal website(s) or blogs, and number of followers. Sally quickly determines that "johndouthat" has a higher *endorsement* (5 followers) and has been more active in GitHub than "macmartine". She decides to forgo visiting the individual's website as this is her initial evaluation pass.

She next views the type of programming *languages* the candidates are familiar with, and the *repositories* that they have *contributed to or forked*, by viewing the radial chart at the top right corner of a tile. Hovering over a slice of the radial chart shows aggregated contributions for a language (number of commits, comments, and issues) across the different projects that the candidate has worked on. It also provides a similar view for repositories: a breakdown of the repositories that the person is most involved in based on the number of their commits, comments and issues. Hovering over a slice of the chart, shows her the repository name, the main programming lan-

guage used in the repository, and the number of watchers that the repository has.

She clicks on a (repository) slice to further explore the activities unique to that repository. Doing so displays a summary of activities as bar charts in terms of commits, issues, and comments broken down on a monthly basis. The lower chart presents the entire history of the candidate, from which Sally selects a date range in 2011. This populates the top bar chart with information for that period. Sally, can now compare side-by-side the amount and type of contributions from both candidates. Hovering over a bar segment in the chart shows the number of commits, comments, and issues per month.

Contributions can be displayed as a grouped bar chart (Fig. 1(a)) or a stacked bar chart (Fig. 2 (a)) to highlight contribution patterns or trajectories (e.g. Sally can see that "john" has increased his commit frequency as compared to "mac", by a quick glance at the grouped bar chart). "Mac" on the other hand has made a lot of comments on issues, which shows that despite the low number of commits, he is still involved heavily in the community.

Sally next decides to more closely evaluate another candidate's ("Mathieu", Fig. 2 (a)) commits. Clicking on a bar (segment) in the bar chart (Fig. 2(a) markup) opens a new (drill down) tile that displays details of the commits (or issues or comments) (see Fig. 2(b)). To review the actual commit, Sally clicks the "View Commit" link, which takes her directly to the committed lines of code in GitHub.

Next, Sally decides to investigate the *soft skills* of "Ben", so she opens his Stack Overflow (SO) tile (Fig. 2(c)). The SO 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.) that Ben is most involved in based on the number of his questions, comments and answers in that tag. Bar charts show his contributions (in terms of questions, answers, and comments) broken down on a monthly basis. When Sally clicks on the "answers" segment in the bar chart it opens a new tile (Fig. 2(d)) that lists Ben's contributions (she sees a list of answers that he posted). 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. Sally can further drill down to view the full answer, its associated question, and comments by clicking on the "view answer" link, which opens the specific Stack Overflow page.

B. Implementation Details

Visual Resume is designed as a web application so that there is no need for installing it at the client site. We chose to do so, to make it feasible for adoption by online communities.

We designed a 4-step approach (collect, process, filter, and visualize), so that the approach can generalize across multiple data sources. Therefore, we separate the collect and process steps from the filter and visualize steps. The former functionalities are performed at the server side, and require a wrapper for each repository; the latter is part of a rich web client, which uses a model-view-controller architecture.

Collect: Visual Resume is designed to present data across repositories. Each data source requires a specific extractor that collects and stores the data in our database. Currently, we have implemented extractors for two popular communities GitHub and Stack Overflow. Since data from each site are accessible in different ways site-specific extractors are needed. For example, the extractor for GitHub invokes the Github API. Since 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 and update the database.

Process: This step has three functionalities. First, it transforms data collected in different formats into a uniform format (we store the data in Neo4j – a graph database [15]).

Second, it creates a data model designed to generalize across different types of project hosting and Q&A sites. It extracts and links the following categories of data: (1) profile information – user details, endorsements, tenure, (2) cues related to technical skills – commits, issues, projects contributed to, and languages used, and (3) cues related to soft skills – comments, questions asked, and answers provided. It also collects and annotates information on votes or badges when available. Note that a new source of information (discussion on mailing lists as opposed to 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. Also note that other pertinent information (personal page, blogs) that is available from profiles in GitHub or Stack Overflow is also included and linked to. This model is then encoded as a JSON file for the client.

Visualize: The visualization is created by using the d3.js framework [16]. 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. We can envision implementing another set of tile templates that use widgets, such as a heat map for amounts of code, line graphs for showing trends in contributions, and scatter plots for correlating contributions and their quality.

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. Other filters can easily be created that adjust information based on the amounts or types of contributions.

III. Conclusions

It is a fast growing trend in software developer hiring that project managers consult candidates' historical activities in online communities to evaluate their expertise. We developed the Visual Resume tool to allow project managers to easily access candidates' activities and judge their performance effectively and efficiently. Our tool aggregates past visible activities from several popular online communities. Providing summarized and detailed activity information gives project managers a better picture of a candidate's characteristics. The tile design allows viewers to perform pairwise comparison. Our tool can assist project managers in locating popular cues to assess candidates' performance.

ACKNOWLEDGMENT

We thank Corey Jergensen and Caleb Larsen for building the Visual Resume tool. We are grateful to NSF grants 1253786 and 1314365.

REFERENCES

[1] J. Anvik, L. Hiew, and G. C. Murphy, "Who Should Fix This Bug ?," in *ICSE'06*, 2006, pp. 361–370.

[2] J. Anvik and G. C. Murphy, "Reducing the effort of bug report triage: Recommenders for Development-Oriented Decisions," *ACM Trans. Softw. Eng. Methodol.*, vol. 20, no. 3, pp. 1–35, Aug. 2011.

[3] A. Mockus and J. D. Herbsleb, "Expertise Browser: a quantitative approach to identifying expertise," in *ICSE 2002*, 2002, pp. 503–512.

[4] T. Fritz, J. Ou, G. C. Murphy, and E. Murphy-Hill, "A degree-of-knowledge model to capture source code familiarity," *ICSE '10*, pp. 385–394, 2010.

[5] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. de Vries, "Moving into a new software project landscape," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10*, 2010, vol. 1, pp. 275–284.

[6] G. Canfora, M. Di Penta, R. Oliveto, and S. Panichella, "Who is going to mentor newcomers in open source projects?," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering - FSE '12*, 2012, pp. 1–11.

[7] J. Tsay, L. Dabbish, and J. D. Herbsleb, "Social media in transparent work environments," in *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013, pp. 65–72.

[8] J. Marlow and L. Dabbish, "Activity traces and signals in software developer recruitment and hiring," *Proc. 2013 Conf. Comput. Support. Coop. Work - CSCW '13*, pp. 145–156, 2013.

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

[10] "Careers 2.0." [Online]. Available: https://careers.stackoverflow.com.

[11] "Gitto." [Online]. Available: https://gitto.io.

[12] "MasterBranch." [Online]. Available: https://masterbranch.com.

[13] "CoderWall." [Online]. Available: https://coderwall.com.

[14] "Open Hub." [Online]. Available: https://www.openhub.net.

[15] "Neo4j." [Online]. Available: http://www.neo4j.org.

[16] "D3.js." [Online]. Available: http://d3js.org.