UNIVERSITY OF CALIFORNIA,
IRVINE

Querying Aggregated Social Media Developer Profiles for New Hires

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Information and Computer Science

by

Shibani Konchady

Thesis Committee:
Professor David F. Redmiles, Chair
Associate Professor Anita Sarma
Associate Professor James A. Jones

2016

DEDICATION

To my family and friends who have been like family.

TABLE OF CONTENTS

Page

LIST OF FIGURES **vi**

LIST OF TABLES **vii**

ACKNOWLEDGMENTS **viii**

ABSTRACT OF THE THESIS **ix**

1 Introduction **1**
- 1.1 Scenarios For Judging New Developer Skills . . . 5
  - 1.1.1 Recruitment for New Employees . . . 5
  - 1.1.2 Onboarding by Team Manager . . . 6
  - 1.1.3 Apprehensions about New Team Member . . . 6

2 Background and Related Work **7**
- 2.1 What are Evaluators looking for? . . . 7
- 2.2 Onboarding New Developers . . . 8
- 2.3 Evaluator- Candidate Communication . . . 10
- 2.4 Trust and Trust Factors . . . 11
  - 2.4.1 Trust . . . 11
  - 2.4.2 Trust Factors . . . 13
  - 2.4.3 Visualizing Trust Factors . . . 14
- 2.5 Technical-Social Platforms that aid with the Recruitment Process . . . 15
  - 2.5.1 GitHub . . . 15
  - 2.5.2 Stack Overflow . . . 17
  - 2.5.3 User Intent in Social Programmer Ecosystems . . . 19
- 2.6 Developer Profile Aggregators . . . 21
  - 2.6.1 Masterbranch . . . 22
  - 2.6.2 Coderwall . . . 23
  - 2.6.3 Visual Resume . . . 24

3 Research Questions and Approach **30**
- 3.1 Research Questions . . . 31
- 3.2 Approach . . . 31

4 TecSo Lens: Filtering Visual Resume **33**
4.1 Lenses: Task-Oriented Design and Filtering . . . 34
4.1.1 Onboarding-Related Questions . . . 35
4.1.2 Trust-Related Questions . . . 36
4.2 Mapping Metrics, Trust Factors and User Questions . . . 36
4.3 TecSo Lens: Design and Implementation . . . 37

5 Evaluation: Pilot Study of TecSo Lens **39**
5.1 Goal . . . 39
5.2 Experiment Design . . . 40
5.2.1 Task . . . 40
5.2.2 Role Requirements . . . 40
5.2.3 Study Structure . . . 41
5.3 Pilot Study Outcomes . . . 42
5.3.1 Participant Profiles . . . 42
5.3.2 Recruiter, Manager and Participant Preferences . . . 44
5.3.3 Useful Information for Selection of Candidates . . . 45
5.3.4 Change in their levels of trust . . . 47
5.3.5 What more information did the participants say that they needed to make their decision . . . 49
5.3.6 Other platforms that could be included . . . 50
5.3.7 Use and Usability of the TecSo Lens . . . 50

6 The Information Gap: Bridging Available and Desired Information **53**
6.1 Location . . . 55
6.2 Communication Media . . . 55
6.3 Educational Qualifications and MOOCS . . . 56
6.4 Work Experience and Recommendations . . . 56
6.5 Communication Skills . . . 57
6.6 Passion and Motivation . . . 58
6.7 Summary . . . 58

7 Discussion **59**

8 Conclusion and Future Work **63**
8.1 Future Work . . . 65

Bibliography **66**

A GitHub Data Fields **73**

B Stack Overflow Data Fields **77**

C Trust Factors- Attributes Matrix **80**

D Questions- Trust Factors- Attributes Matrix **88**
D.1 Onboarding 88
D.2 Trust 96

E Screenshots of the TecSo Lens **103**

F Pre-Session Questionnaire **111**

G Requirements as provided for the Recruiter Role **116**

H Requirements as provided for the Manager Role **119**

I Post-Session Checklist **122**

J Post-Session Interview Questions **124**

LIST OF FIGURES

Page

5.1 Familiarity with GitHub and Stack Overflow 43
5.2 Topics discussed by Recruiters and Managers/Tech Leads from Participants' Experiences 44
5.3 Study Participants' Skill Set Preferences 45
5.4 Match of Requirements with Skills of Selected Candidate from the TecSo Lens 47
5.5 Trust Levels for an Unknown Candidate 48
5.6 Trust Levels for Selected Candidate 48

LIST OF TABLES

Page

2.1 Trust Factors 13
2.2 Correlations of Behaviors and Intents to Users 21
2.3 Technical Skills 25
2.4 Soft Skills 27
2.5 Signals for Quality of Work 28
2.6 Signals for Social Competency 28

5.1 Study Participant Profiles 43
5.2 Additional Useful Information 49
5.3 Platforms and useful information accessible on it 51

ACKNOWLEDGMENTS

I am extremely grateful to Professor Redmiles for his patience, guidance, encouragement and unwavering constant support for the duration of this thesis as well as the initial stages when we were thinking about the possible options that could be used as starting points for the thesis.

I would like to thank Professor Anita Sarma for firstly giving the foundation for the thesis through her work on the Visual Resume, and for providing me with appropriate feedback to help structure the research as well as the writing as and when needed. I would also like to thank Professor James A. Jones for being on my committee and giving me feedback which allowed me to look at the work that I had done from a different perspective, and refine it accordingly.

I would like to express gratitude to Tao, Mengyao, Andrea, Prateek, Charu, Shonali and Kartika, for the discussions and the suggestions that arose from them which helped provide a wider view to this work.

I am eternally grateful to my family for encouraging and supporting me to pursue my aspirations and to my friends for reconnecting me to the real world and for just being there for me.

ABSTRACT OF THE THESIS

Querying Aggregated Social Media Developer Profiles for New Hires

By

Shibani Konchady

Master of Science in Information and Computer Science

University of California, Irvine, 2016

Professor David F. Redmiles, Chair

Software developers often seek to be a part of a new company, team, or even a different project within the same team. The decision to choose a developer who may be an appropriate fit may be made by evaluators such as team leads or recruiters. The developer may have information about his/her technical and social competency available on a resume and/or may have the raw data from technically-relevant activity publicly accessible on various social media platforms such as GitHub and Stack Overflow. An evaluator could go through each of these information sources manually or may choose to utilize Developer Profile Aggregators such as CoderWall or Visual Resume. Such aggregators tend to employ a browsing-style interface, which means the evaluator may need to search extensively for the information (s)he may be looking for.

We researched the kinds of questions that evaluators may have that could be directly answered by the information available from the abovementioned sources. This was then used to develop an interface to more directly filter and view the answers to these questions. This system is called TecSo Lens and it is based on a previous browsing aggregator tool, Visual Resume. The questions and the answers that represent the data are also related to the factors that could influence trust during this evaluation process.

Chapter 1

Introduction

There is a large pool of eligible job applicants in the science and engineering fields today, in spite of the slow growth from about 3.5 million scientists and engineers in the United States in 1999 [78] to 6.2 million in 2012 [69]. In contrast, the vast reach of computers and the advancement of technology across the globe has led to a rapid rise of the tech industry especially in the well-known Silicon Valley with its share of ups and downs with the dot com bubble [60] and occasional bouts of recession. The sustainable growth of this industry depends on a steady influx of highly skilled individuals that balances out the number of employees moving out of these industries [43]. This competitive demand to weed out and acquire the most suitable candidate for the role means that the ones assigned with this duty need to come up with an efficient approach to filter out the applicants effectively.

The organizational structure of a company may determine the identity of the party responsible for carrying out the task of selecting the ideal candidate. Most companies have their first level of the recruitment process as the online application portals, where candidates may submit their resume, along with links to other sites like LinkedIn [35] and GitHub [13] that may contain more information about their activities [45]. As per a talk by Amazon em-

ployees at a recent conference, resumes that contain related or specific keywords that were mentioned on the job profiles on these companies' application portals are more likely to be picked over those that don't [66].

Once the automated filtering of candidates is complete, the human aspect of the evaluation process comes in. Evaluators such as specialized recruiters, team leads or even managers (depending on the company under consideration) are asked to interact with these candidates through various means of communication in order to identify a suitable match to the requirements for the role. These means of interaction could include emails, recruitment stalls at career fairs, or phone or personal interviews.

Apart from thinking of a candidate from the perspective of their individual characteristics, an evaluator may also need to consider them with respect to a team they may have to work with. These could mean the soft skills that they possess for communication and any form of interactions with their peers. In case of teams, which may be local or globally distributed, the onboarding of a new developer into the team becomes an important task for the one in charge which could be a team lead or a manager. Onboarding may then involve mentoring the new hire and introducing him/her to the past and ongoing projects, the work practices of the team and the general culture of the organisation [42]. Bringing them up to speed means that this mentor would need to identify the background and experience of the hire to know how much more information would be needed to bridge the gap, to successfully take up the assigned role on the team [64]. The previous experience of the candidate could thus affect the effort involved in the onboarding process. If a candidate has skills that are at par with the needs of the role, the onboarding process is likely to be less cumbersome and hence, the evaluator may find the candidate to be a better fit for the requirements during the hiring process.

Considering the large amount of information that they may have at hand about each candidate, across a variety of contributions and the limited amount of time available to go through

them, the evaluators/recruiters/mentors may desire to have some direct means of viewing all this information in a consolidated fashion. There are several developer aggregators like MasterBranch and CoderWall [71] that help aggregate contributions made by developers across multiple platforms into a single profile [70]. But these tend to focus on technical contributions rather than on the soft skills that a candidate may possess. Since evaluators are looking for a candidate with a more holistic profile, considering that in the future they envision the hire taking on more managerial responsibilities. In the long run, it may then be more beneficial to have greater information than just the technical capabilities of the candidate.

Visual Resume was proposed as a solution to this; being a developer profile aggregator that aggregates developer activities across different kinds of contributions and repositories in a single developer profile [70]. This application consolidated the raw data as captured on their activity traces from the profiles on a code-hosting site- GitHub [13] and a technical Q&A Forum- Stack Overflow [30] for a candidate into a more visual representation for easy reading and access. But Visual Resume employed more of a browsing- style approach to its interface interaction, that presents a high-level overview of the information of the developer on each of these platforms, and requires clicking on corresponding links to view more detailed information, one layer at a time. So unless the evaluator is aware of what information that they want to view and what category it falls under, they may have to go through a considerable amount of browsing before they arrive at the information they were looking for.

To understand how to go about implementing a more direct approach that would aid evaluators in this process, we first looked into what would be the kind of information that they tend to seek in candidate applications. The attributes that signify a particular characteristic such as the languages they know or the quality of the code they produce or how willing they are to answer questions, could be useful information for this purpose. We looked to identify

the different categories of information that are provided by platforms like GitHub and Stack Overflow, which are used by Visual Resume [70]. Since evaluators may have certain expectations of these candidates, and since trust is the positive confident expectations that one may have about another person's conduct [56], trust also had to be taken into consideration. Thus the attributes or categories of information identified were then mapped to the kinds of trust factors that enable an evaluator to trust the capabilities of a candidate [76] for a post, to find some form of common ground between the two. The specific questions that an evaluator may have were researched and then mapped to these trust factors and to the information provided by platforms like GitHub and Stack Overflow.

The direct approach considered as an alternative to the browsing form of interaction with the interface was to employ a more task-oriented style instead. This meant having a set of common questions that an evaluator may have, and then presenting this user with only the information that is relevant to the question selected. The answers or the information would not be focussed at a high level and would instead include all the relevant information across different levels of detail.

The meta interface created as a proof-of-concept for this model was titled as the TecSoLens, which acted as the filtered view of the existing Visual Resume that filtered the information provided by it through means of direct questions. The intention here was to provide more concentrated information based on what the user was looking for.

A pilot study was conducted with five users interacting with the TecSoLens. The participants were asked to take on roles of either a recruiter or a manager, and were provided with a set of requirements for a role. They were given the task of evaluating five candidates for a given role and were asked to describe their approaches to analysing them through the TecSoLens and choosing one among the five for the job. They were asked to describe why they chose a particular candidate and what other information would have helped them better with their decision.

This gap between the information they have and the information they need, which may be present on other platforms, is the information gap that could be further examined in future work. This knowledge could then be used to improve the evaluation process and experience during new hire recruitment and onboarding.

As an introduction to the interactions involved in the process of recruitment and onboarding, a set of scenarios have been discussed in the following section.

1.1 Scenarios For Judging New Developer Skills

Whenever a newcomer (with respect to the other team members) either joins or is being considered for a team, certain knowledge about the person may be required to set the foundation for the basic level of trust needed to consider the person a viable asset on the team. The following three scenarios illustrate the cases that call for the need to access information about a potential developer to make appropriate judgements to form a basis for trust and future work.

1.1.1 Recruitment for New Employees

ABC, a recruiter for company XYZ for the Software Development Engineer role wants to recruit new employees for a new incoming project. For this, he says that resumes offer a mere overlay of the experience of a developer. To gauge the potential of an employee as per their skills and interactions and trust that they will be a good fit for the company and the role, he wants to know about the technical activities the employee is involved in along with the social interactions involved. He, being from a non-technical background, wants to look at specific statistics which would tell him about their experience, their technical array of skills, and quality of interactions with other colleagues on various platforms.

1.1.2 Onboarding by Team Manager

PQR, a manager for company XYZ has just received a list of new hires for her team. Each of the hires have different experience levels and technical skill sets. She has a new project and she wants to assign them to the different modules based on who would be the best fit for it. Since there is no prior implicit knowledge about the person under consideration, she needs information to determine this. The best fit could be judged on the basis of the frequency/promptness of their interactions and availability if they have to be selected for a role that needs regular communication with the client for the project. For a testing role, perhaps the issue reporting and resolution statistics could be useful. For development roles requiring the quick development of scripts for quick changes in an iterative development model, she would want know about the past history of frequent updates for progress in the project as well as the frequency of commits, and languages worked on. There could be several such roles for the different modules in the project that would require certain knowledge in order to ascertain the suitability of an employee for a specific role.

1.1.3 Apprehensions about New Team Member

DEF, a software developer with four years of work experience on team GHI has just had a new hire JKL join the team, and they have been asked to work directly with each other on all future work for the given module in the project. DEF is unsure whether GHI will be able to keep up to the current pace of DEF's work, and deliver the dependencies on time as needed. If DEF had access to information that shows high availability, self-initiation and past experience of delivering projects on time with accepted commits, then he might be more likely to trust JKL to deliver the required modules on time, and be more acceptable to working with him.

Chapter 2

Background and Related Work

2.1 What are Evaluators looking for?

Social media platforms like GitHub, LinkedIn or Stack Overflow have varying areas of importance based on who is accessing the information on it. Managers and team leads tend to look for specific cues on the profiles provided by applicants, so that they can make insightful inferences about a candidate's technical skills, motivations and values. They trust the activity of a developer on a public platform like GitHub more than a resume that s(he) provides since the actions or activity traces are considered more transparent and harder to manipulate [61]. Thus they are less likely to be biased form of self-promotion [54] which may be the case of resumes and regular interviews. Recruiters on the other hand, look for cues for quality of work and social competency [70]. Apart from technical factors, they look at social media as a means to actively engage with candidates and better assess them with respect to public interaction and engagement. This helps them evaluate a candidate in terms of his/her cultural fit for the organization [71].

An evaluator may have several guidelines at the high-level when faced with the task of hiring an ideal candidate for a role [52]:

- Anticipating the assets that a person must possess as an ideal candidate for a role and their relative importance which could be achieved by ranking them. We will look into more about these assets in the coming sections.
- The level of salary expected by a candidate which is beyond the scope of this thesis. This could be examined further in future iterations of this project if developers are given an option to reveal their current salary and future expectations online.
- Organizing themselves better to find the right candidate which could be extended to organizing the available data of the candidates better.
- Comparison amongst the available candidates, each of whom will have their unique combination of qualifications, strengths and weaknesses. A focused comparison between candidates may involve evaluating them on the basis of their motivation, coding ability, collaboration and project management [70].
- Identifying the best fit, which could mean finding the closest match between their given set of requirements and a candidate's unique combination of abilities.

2.2 Onboarding New Developers

> *"Onboarding is the process of helping new hires adjust to social and performance aspects of their new jobs quickly and smoothly." [39]*

When new developers join an organization, they may already possess some of the technical skills required. But they may still need further guidance on the technical requirements, the tools used and general practices of the team as well as the general work culture of the

organization [42]. They may need to understand the business context of their assignment, which could include the customer base, the market it is intended for, the current and future needs of the project. They would also need a develop a detailed understanding of the different practices followed at different stages of the software development life cycle such as planning, architecture, design and build, as well as the tools and software used for this [64]. Training and onboarding are two common practices employed by organizations to aid with this purpose, where they may choose both or one over the other or a possible amalgamation of the two. Training covers the technicalities of the role in terms of what needs to be done for the job with respect to technology, equipment and procedures for tasks. Onboarding on the other hand, integrates the technical and human aspects to allow the hire to seamlessly blend in with the other employees, management and corporate culture [9]. This could be applicable to local or globally-distributed teams, as well as to collaborators that interact solely through the medium of the internet such as in case of open source development [73].

Developers come from a diverse range of backgrounds where their knowledge may be from formal or informal training, they have varying levels of industry experience or they may coming in from different locations. They may also be self-reliant and may be unaccustomed to collaborative practices. In such cases, they may not even how or whom to approach in case they do need help. To aid with such issues, some organizations have onboarding programs that involve assigning a mentor to the new hire along with the necessary documentation, and having a clear plan of action for the ramp-up. This mentor could be a developer, manager or team lead with a good knowledge of the history of projects, practices and the technology in use as such [42]. The mentor could help the new developer better if (s)he is updated with the skill set and abilities of the hire. This would make it simpler to map the present skill levels to the given requirements and help bridge the gap more easily [64].

Since time and money are two important dimensions in a project, being able to ramp up a new developer to maximum productivity in minimal time and effort could mean that

onboarding may have a very influential impact on a project [64].

2.3 Evaluator- Candidate Communication

> *"Conversation is a medium for decision making. It is through conversation that we create, develop, validate, and share knowledge." [47]*

The recruitment process for organizations generally involves iterative rounds of conversation between potential candidates and the representatives of the organization such as managers, team leads or recruiters, responsible for the selection of a new hire. This conversation could be of the written form such as through applications, resumes or cover letters. It could also be of the spoken form such as career fair elevator pitches, on-phone or in-person interviews, or group discussions.

A more engaging and directed form of conversation across these different means of communication could be via questions. A list of requirements for a role as presented to such an evaluator could be translated in the form of questions, wherein the directed answer could then be the focused set of attributes for the candidate that satisfy these requirements.

In today's world, where the candidate's activity traces may be spread across multiple platforms, one needs some form of media or application capable of consolidating this information and making it more visually direct to comprehend.

Questions could be a useful mode of conversation with the ensemble of platforms that a candidate's data is spread across. It may possibly be a form of one-sided conversation, if the user is just querying the system and obtaining the relevant information as the answers. But perhaps it may not necessarily be one-sided, if the questions are considered one end of the conversation and the other is the information provided by the candidate (directly or

indirectly through activity traces).

2.4 Trust and Trust Factors

2.4.1 Trust

Trust as a term has multiple implications based on the context in which it is used. Literally it is defined as "a bet about the future contingent actions of others" [71]. From a more optimistic perspective, it could mean having positive confident expectations about another person's conduct [56], and thus having the ability to predict their future actions.

Generally in scenarios that involve the coexistence of humans and technology, which is the generic case in today's world, this trust would involve both social and technical aspects. A prominent form of interaction today is through the technological means at our disposal, such as phone calls on the mobiles, online Skype conversations as opposed to face-to-face meetings, emails Vs. letters or even the spoken word for that matter. So the implicit tone of our actions which may have been understood or communicated through in-person conversations, is masked away by the layer of technology through which we choose to interact [76]. In the field of software engineering where one's job itself relates to the use of technology, we have this added dependency apart from the present communication via such means. Having teams where one needs to choose a new person for an incoming task would require trusting this person when not much is known about them before hand. The information that is known could come from a resume [70] that they provide or from recommendations from a mutual contact who may or may not already be in the trusted circles of the decision maker.

Interpersonal trust is what encourages people to trust in each other enough to take risks, which is an important factor for collaboration in any kind of team. This trust in itself could

be classified into two types: cognitive and affective trust [63]. Cognitive trust relates to the beliefs one may have about another person with respect to their competence and reliability. So if one is confident about the capabilities of another, one may be more likely to take risks and engage in less self-protective actions. So if you are more amenable to a programmer with years of experience, with a history of meeting deadlines and who has hardly had an issue raised in code worked on by him/her, then you're more likely to trust that person. Affective trust comes about as a result of emotional ties between group members, and the trust thus becomes a consequence of their observed behavior with each other. So if you are working with a developer who replies quickly to emails or helps resolve other developers' issues, then you may develop affective trust with respect to that person [79]. Since affective trust requires some form of prior connection or emotional tie with the other person, cognitive trust is the only form that would be applicable to candidates or more specifically new hires during recruitment or onboarding.

In software development, this form of cognitive trust could come about by having knowledge about or access to interactions which may set expectations such as the activity traces of candidates in project repositories or documents over time. Collaborative traces like these tend to be useful as they represent the past and current work by developers as they manipulate project artifacts. Visualizing these traces could help increase a developer's sense of awareness about developers and the artifacts that they work with, and this could then act as an indicator of a trust factor. An important point to note would be that not all interaction in software development, albeit forms of collaborative traces are not accessible to the general public. This could be because of work carried out in private sectors or incomplete documentation for any reason [77].

2.4.2 Trust Factors

Trust factors collectively refers to the information that affects people's perceptions of others' trustworthiness [77]. As per the "tug-of-war" model of trust by Al-Ani and Redmiles [37], trust could be influenced by the shift of balance by the contrasting factors. Good leadership and time allocation would be the ones that would have a positive impact while a large team size, high team diversity and a challenging project type could have a negative impact on how trust develops in such an environment.

Trainer identified the following trust factors through an extensive literature review as a part of his dissertation [76]:

Frequent Initiations and Responses	Years of Experience
Frequent Updates of Project Progress	Homophily
Reputation	Shared information
Use of multiple communication media	Shared Photographs
Same location	Team Diversity
Role	Monitoring
Availability	High/ Low Team Size
Leadership	Project Size/ Type
Frequent Meetings	Expertise
Gender	Language

Table 2.1: Trust Factors

These trust factors once identified, needed to relate to real-world software artifacts to better understand their application to the world of software engineering. Thus, the collaborative traces as previously mentioned, were then mapped to trust factors by Trainer and Redmiles [77].

2.4.3 Visualizing Trust Factors

> *"Research approximates that between 80 and 90 percent of the information received by the brain is through the eyes" [51]*

It is often said that seeing is believing [28]. Visual proof, or just being able to see something with your own eyes could help affirm your belief in it. Having any form of knowledge or information thus, in a visible form is useful as it makes it better and easier to access, discuss, value or manage [46]. An image being a form of such media could have the power to clarify a thought or provoke a response as a part of a conversation [72].

Visualizing trust information or more specifically factors as in the case here, could have an influence on user behavior and decisions [81]. As mentioned earlier, visual representations of collaborative traces not only help summarize the information for easier comprehension but also help increase a developer's awareness which could be with respect to other developers or the artifacts (s)he may be working with [77]. This awareness could then in turn affect the interactions and generic communication between the developers, as well as possibly their productivity in the long run. The different kinds of artifacts and the data that encompasses them and their usage could help shed light on various aspects of the abilities of the developers involved. This could be technical as well as social. For example, change-sets and authorship for source code could provide information about a developer's expertise level. On the other hand, the rate of response to e-mails or instant messages could indicate the level of a engagement for a developer, as well willingness to help others.

2.5 Technical-Social Platforms that aid with the Recruitment Process

"Social tools facilitate a participatory development culture in software engineering with support for the social creation and sharing of content, informal mentorship and awareness that contributions matter to one another." [74]

People today are spending almost twice as much time online as they did ten years ago, with the average adult Internet user claiming to spend over 20 hours online per week [31]. As of 2014, 74% of Internet users said that their Internet use involves social networking sites. This includes Facebook, Twitter and LinkedIn, but it is not an exhaustive compilation of all online social activity platforms [29].

If we narrow down the user group on these social platforms to software developers and others working in related fields, and focus on technology-related or more specifically software development-related social activities, we may have our search results list out LinkedIn, GitHub [13] and Stack Overflow [30] among others. When it comes to recruitment, when Ben Horowitz, a partner at Andreessen Horowitz was asked about LinkedIn with respect to GitHub, he replied, "Why would I look at their resumes, when I can look at a body of work?" [12]. Going by this trend and the focus on peer production sites, we take a closer look at GitHub and Stack Overflow.

2.5.1 GitHub

GitHub is a social network site that lets developers connect and collaborate [71]. It is a web-based Git repository hosting service. Like Git, it contains the distributed revision control and source code management (SCM) functionality along with some of its own features. It

provides access control and collaboration features like bug tracking, feature requests, task management and wikis for all of the projects [14].

GitHub addresses the dual issues of complex projects and anti-social developers. In case of the former, it runs one big SCM in the cloud so that each of the development teams does not have to manage it on its own. It handles the latter by organizing projects around people rather than code [15]. The working principle of the social system of GitHub is that if developers contribute to similar code repositories, it could lead to them following each other or vice versa [80].

Developers can host open-source [12] software project repositories using this Git revision control system. Each developer has a profile which is an easily accessible public portfolio of their open source development activity. They can also search for software projects in which they are interested, fork them to add in their own contributions and also follow the activities of others. As the organization structure categorizes the software repositories by software developer rather than project, each developer's repositories are directly accessible through his/her profile. One can also view the developer activity across GitHub directly in a news feed. Raw data is captured for all activities on GitHub. For example, every commit would include details such as the developer involved, the time of change, lines of code changed and any other discussions around it [71]. Features of this kind could then be used to increase awareness of other developers' activities, to form impressions about them and to help them coordinate with each other [80]. Open source in general opens up an arena for long- term evaluation of candidates based on the quality of work, cultural fit, passion through self-initiated work, capability for completion and a degree of humanity [27].

The fields for data captured from activities on GitHub is listed out in Appendix A.

The advantages of GitHub over resumes and LinkedIn for the purposes of recruitment are that [12]:

- The traditional services list the candidates' work histories as per their own submissions, but not the actual work that they might have done. GitHub allows them to actively engage in a public demonstration of their capabilities.

- If developers have taken the time to develop a GitHub profile and have put in the effort to participate actively in the community, they would be in better position to be evaluated as compared to someone who has not.

- This form of selfless participation in open-source projects where developers code a solution to share with the community could be used by companies as an initial indicator of the potential of a candidate and their credibility in this community.

2.5.2 Stack Overflow

> *"All knowledge sharing sites build on the power of human expertise and the motivation of individuals to provide answers and exchange information." [65]*

Stack Overflow is a question and answer platform for enthusiastic and professional programmers. It is a part of the Stack Exchange network of Q&A sites but is specific to topics of programming. This was founded with the intention to evolve into an extensive library of answers for all questions on programming [30]. It is also regarded as a form of documentation by and for developers [68].

Since this site is meant for getting direct answers while not being a discussion forum, it does not encourage casual chit-chat. The questions are expected to be to the point, based on an actual problem with specific details provided. They should not be opinion-based or lead to discussions instead of focused answers. The good answers get upvoted by other users, and the most suitable one as per the one who asked the question is marked as accepted [30].

The organization of questions through association is done with the help of tags. Each tag is a

keyword or label which could describe a category for related questions of a type. Reputation is the term that refers to the score on a user's profile. It increases when others vote up the user's questions, answers and edits. It could also increase if the user provides a satisfactory answer to a question with a special bounty on it [30]. The site uses gamification concepts, which is the use of game design elements in non-game contexts [44]. The reputation points and the bounty encourage and reward community participation [71]. Different levels of reputation unlock different kinds of privileges such as abilities to vote and comment [30]. The experts are incentivized as the ones who answer questions are rewarded more than the ones who post questions [65].

The reputation thus obtained, not only acts as a measure of the perceived quality of the answer [75], but also one of how much the community trusts this user [30]. Stack Overflow also relies on a small number of experts for its contributions, as they are the ones who provide the bulk of the helpful answers. So it needs to identify the users who could become potential strong contributors for better growth [65].

The fields for data captured from activities on Stack Overflow is listed out in Appendix B.

Stack Overflow promotes mindshare among the users who are predominantly software developers. It has replaced web search and forums as a primary starting point for finding solutions to programming problems. The design of this platform is centered around deep community involvement and continuous technical adaptation of the software platform rather than being focused on human-centered design. Since it places a tight focus on technical answers rather than conversational design, there tends to be a lack of opportunity for discussion and debate [59].

2.5.3 User Intent in Social Programmer Ecosystems

When one talks about social media platforms, it does not merely involve the activities of users who engage with it. It also includes the intent of these users. If we are looking at a user set of developers and recruiters as a part of a social programmer ecosystem, we can have the following behaviors and intents that would correlate to these users are in Table 2.2 [71]:

	Developer	Recruiter
Reasons for Participation or taking interest in the social programmer ecosystem	-Enjoy Interactions -Competition -Pride (showing off achievements) -Recognition -Helping others -Pushed into ecosystem by peers -Searching for work	-Potential for better assessment of potential candidates -Engage with candidates by -actively using ecosystem -Speed up finding new candidates
Modes and terms of interaction	-Assess other developers first • Investigate what others have created • Make sense of geographically dispersed coworkers -Use common interests to find interesting people -Recognition by peers	Used relatively traditional means- generic professional networking sites like LinkedIn or Xing • Used personal networks for recruiting -Going where the developers are to recruit them -Using social media to look more authentic

	Developer	Recruiter
	-Acknowledge good work -Avoidance of recruiters viewed as spammers	-Filter developers by skills -Common baseline check-Looking at a developer's activity and engagement in open source -Check if they used best practices -Used other people's endorsements of a developer's code as a proxy to assess their tech skills -Looking for diversity in developers -Fast learners -Passionate Developers -Share company values
Impact of Praticipation	-Gamification is effective -These features lower the barriers for participation -Learning new programming languages -Exploration and experimentation -Motivated software teams to contribute more -Diffuse new ideas	-Enable others to understand complex attributes of developers with less cognitive load

	Developer	Recruiter
Risks and Challenges Participation	-Lack the social connectedness -Struggle to keep up with fast paced environment -Public signals should not receive too much weight	-Irritated by lack of public activity for a developer -Non-tech recruiters struggled with interpretation of signals from developer profile aggregators and code sharing sites

Table 2.2: Correlations of Behaviors and Intents to Users

"Social media.. tailored to software developers, can influence the varied stakeholders in software engineering.." [4]. So social media or rather, the richer data that it provides about an individual (developer) could be used to infer more information for a wider perspective on the person.

2.6 Developer Profile Aggregators

> *"Social media..tailored to software developers, can influence the varied stakeholders in software engineering.." [4]*

Employers as well as developers use information from developers' profiles and activities to assess potential employees or peers [61][71]. Marlow and Dabbish found that employers assessed the online activities of developers which took less effort to evaluate [70][61]. At times, developers choose to participate in social media because they think that this might increase their chances of being spotted by a recruiter, while looking for work [71].

Since developers contribute to different kinds of projects, they could vary in terms of the hosting sites or even on the kind of content on the site [70]. Developer Profile Aggregators help aggregate contributions across multiple platforms into a single profile. This helps in the comparison of contributions across these multiple sites which could have different content as well as presentation styles. The two issues with designing such an aggregator tool are information aggregation and presentation. Information needs to be aggregated in such a way that it is more than a shallow overview, and yet does not result in information overload. The presentation needs to provide context, allow comparisons and be easy to access [70].

Some of the currently used developer profile aggregators are discussed in the following sections.

2.6.1 Masterbranch

Masterbranch which started in 2009, is a community for developers to network with their peers and to grow their skillset. It monitors all the open source projects from Google Code, SourceForge, GitHub, Apache, CodePlex, Wordpress etc [21]. The developer profile itself would consist of [71]:

- Name, location and image of the user
- DevScore which is calculated from developer activities like files, commits, projects and the period of activity [4].
- Endorsement button that symbolically allows developers to give 'free beer' to other developers.
- User Repositories table that shows the distribution of programming languages across projects.

- List of projects that a developer has worked on with the name, duration, description and the programming language used.
- Achievement Badges for the most valuable programmer.

2.6.2 Coderwall

Coderwall launched in 2011 and is also a developer profile aggregator that analyzes the repositories of developers on social code sharing sites [4]. The sites covered under this are GitHub, Twitter, LinkedIn and a link to the Stack Overflow profile page [70][5]. The developer profile here consists of [4]:

- Name, current company, location and image of the user.
- Chronological timeline of the events that the developer was involved such as talks and earning achievements.
- Developer skills
- Number of endorsements
- Achievement badges earned by developers.
- List of people that this developer is connected to on Twitter.

Both Masterbranch and Coderwall only provide high-level summaries of the number of projects that the developer is involved in, and the top contributions of a developer [70].

2.6.3 Visual Resume

Visual Resume is a developer profile aggregator that aggregates activity traces of developers across different types of contributions and repositories into a single developer profile. It takes the data about the contributions from two peer production sites– GitHub (GH), the code-hosting site and Stack Overflow (SO), the technical Q&A forum. It summarizes the developer contributions across these and enables side-by-side pairwise comparisons of candidates through a card-based design [70].

It analyzes both technical and soft skills to evaluate a candidate. The technical skills as listed in Table 2.3 [70][61] relate to the details about the developer activities such as lines of code committed, issues resolved and the interactions or discussions around the code that the developer participated in. Knowing such details could help reconstruct aspects like what someone works on, what their code looks like, working style and speed. They include coding ability and quality of work. The soft skills as listed in Table 2.4 [50][62][70][71][82] identify how well a candidate will fit in in the organization culture, especially when the contributors are globally distributed. These include collaboration competency, project management ability and motivation[70].

Type	Description
Coding Ability	Qualifications of software developer: • Programming languages known • Coding ability Experience lets us know about: • Level of expertise • Level of understanding • Type of work preferred • Overall experience level

Type	Description
	Usable cues: • Owned and forked projects • Frequent contributions to projects (eg. commits and answers) • Number of languages that the developer is proficient in
Quality of work	Categories: • Candidate's competence • Skill Level Usable cues: • Range from code review to test coverage metrics • Acceptance by community -GitHub- Accepted commits -Stack Overflow- Accepted/Upvoted Answers

Table 2.3: Technical Skills

Type	Description
Collaboration Competency	Attributes of a Team Player: • How a developer talks about work • Negotiates changes to projects • Includes discussions on issues, code reviews or mailing lists. Key criteria for collaboration: • Manner • Willingness to help

Type	Description
	• Provision of sufficient context for a useful solution • Interaction styles Cues for positive interaction on developer activities: • Comments on issues • Answers & Questions submitted in Q&A forums • Nature of these activities in terms of whether they are polite or provide helpful answers Cues on endorsements: • Number of followers • Developer's Reputation
Project Management Ability	Management skills imply the ability to manage their own work. For an owner, it would include responsibility for: • Project's overall design • Incoming contributions • Interactions with potential contributors
Motivation	Key assessment criteria: • Projects owned • Projects contributed to • Diversity in types of project or languages involved Usable cues: • Recency and volume of activities (eg. commits,issues)

Type	Description
	• Number of owned or forked projects not directly related to developer's own work • Diversity in languages used • Diversity in projects (using different technologies, modes of thinking, programming languages)

Table 2.4: Soft Skills

Apart from the technical and soft skills that were analyzed for Visual Resume, and the cues that could be used as indicators of these, the signals that could indicate the quality or work and social competency were also examined.

Signals	Cues and Implications
Association with popular projects	Whether candidate owned projects or contributed to other projects (especially popular ones). The perception about the reputation of the project could be an influencing factor.
Contribution Type and Style	-Commits preferred over comments -Answers preferred over questions -Source code evaluated to assess style based on code structure, variable names and comments. -Comments could be either on other developer projects or on commits or issues. This could also be a sign of a good documenter. - Good communication is indicated by polite, articulate and easy to understand answers, with sufficient context or detail provided (eg.code snippets)

Signals	Cues and Implications
Commit-related details	-Amount of code in change set (as compared to candidates with similar number of commits). -Changes central to code base (core files). -Commits that solve complex issues. -Accepted pull requests.
Tinkering vs significant changes	-Commit to add feature or just tinker -Writing big blocks or tweaking code -Building Infrastructure, adding new features or just doing tinkering work

Table 2.5: Signals for Quality of Work

Signals	Cues and Implications
Endorsements	Selection decisions based on community endorsements (followers in GH and reputation in SO)
Passion for learning	-High activity in GH-proxy for interest and motivation -Forked projects- Key cue to evaluate candidate's passion to learn something -Reputation scores in SO
Management skills	-Ownership Whether candidate owned projects -Communication Number of answers/questions and comments (Polite) -Better manager- more answers than questions (articulate and willing to help)

Table 2.6: Signals for Social Competency

The social programmer environment formed here, needs to consider the aspect of social competency along with the quality of work. In the case of the Visual Resume, GitHub shows a candidate's productivity while Stack Overflow shows the candidate's enthusiasm.

The Visual Resume employs the use of tiles with information from GitHub and Stack Overflow that can act as cues of technical or social attributes of a developer. Each developer in the system has a set of tiles that contains all of their data captured from these platforms. The view for these begins at a high-level where the information for GitHub and Stack Overflow can be viewed separately for each developer. At this high level, the tile contains basic profile information along with an overview of the activity statistics. There are links which allow one to "drill-down" into more detail about the selected aspect on a new tile, that opens when you click on a link. The tiles can be moved around on the interface for a more convenient comparison of candidates [70].

3

Research Questions and Approach

This thesis aims to look into the kind of information about a candidate that could be deemed valuable during the recruitment process, as well as the platforms that could provide this insight. This information could then be used by evaluators such as developers and recruiters to answer any questions that they may have about these potential candidates, which could prove helpful in establishing a base level of trust and aid with the eventual onboarding process. But all of the information that could assist with decisions pertaining to new hires may not always necessarily be available on the platforms commonly in use. This information gap could be used to improve upon developer profile aggregation in future work to provide a more holistic picture.

3.1 Research Questions

The research questions that this thesis ventures to examine the answers for are as follows:

1. What information about a newcomer(candidate) helps an evaluator in his/her decision to choose this candidate for a job/role in terms of technical competency and social fit in an organization and/or a team?
2. What role does trust play in the information evaluated as per the first question?
3. What information can GitHub and Stack Overflow provide in accordance with the first two questions, and what information is missing that could be included in future applications?

3.2 Approach

Visual Resume– a developer profile aggregator as researched and developed by Sarma et al. [70] was used as a starting point for this research. This platform utilizes raw data from GitHub and Stack Overflow and visualizes it in the form of information on multiple movable tiles. One can browse through the high-level overviews of this information for a candidate on Visual Resume. But for more detailed views, one would have to go to the GitHub and Stack Overflow profiles of the candidate as needed. This data is listed out in Appendix A and Appendix B in the form of fields.

Along with this list of attributes that could be useful to an evaluator, I also looked into the literature on what evaluators tend to look for in potential candidates. This included what would enable them to trust a candidate for a job and make it easier to onboard the candidate as a new hire. These aspects were then translated into potential questions that an evaluator may have about a candidate.

Trust factors that were consolidated together by Trainer [76] and then individually correlated to collaborative traces by Trainer and Redmiles [77] were then mapped to the original list of attributes from Visual Resume, as listed in Appendix C. The questions that were identified, were again mapped to the trust factors and to their correlation to the Visual Resume attributes as listed in Appendix D.

I developed a prototype interface called the TecSo (Technical-Social) Lens, which is based on the Visual Resume but has a task-oriented interface as compared to the latter's browsing-oriented interface. The task questions are sample questions from the set obtained as mentioned above. A question selected in conjunction with a specific candidate displays information about the candidate with respect to this question. A pilot study was conducted with five subjects to evaluate the useful information as available from GitHub and Stack Overflow, which is interfaced with TecSo Lens for making a decision about a candidate. The change in the user responses regarding trust with respect to activities involving a candidate looks to address the second question. The information from the Visual Resume, the literature survey and the pilot study responses answer the first question in terms of what information may be relevant, and the third question about what information may be missing. This evaluation process is described in more detail in the next chapter.

4

TecSo Lens: Filtering Visual Resume

Visual Resume is a Developer Profile Aggregator that utilizes data from the GitHub (GH) and Stack Overflow (SO) profiles of a candidate to summarise the contributions for him/her. One can browse through all the information for an individual by starting with the initial profile information and overall statistics from either GH or SO. This information is displayed on the first tile that is opened for the combined selection of a particular person and either GH or SO as the platform for which we seek the information. Clicking on links within this tile, opens up another tile with more detail related to that link. You can keep browsing through this information until you either need to go to the GH or SO profiles of the person for more information, or you reach the limit of tiles that could be opened on a single page. The amount of effort required to evaluate a candidate could be extensive [70]. A task-oriented interface was considered as a possible alternative to this form of browsing-type interface. This would aid in a more focused filtering of the information available on the Visual Resume.

4.1 Lenses: Task-Oriented Design and Filtering

"It's often said design is a dialogue between designer and user." [36]

Since it is human nature to converse with others as means of sharing knowledge and organizing themselves, it could be useful to talk to the users about what they want and what they need. Design here could be used as a dialogue to get a view of these needs and then provide it to them [36]. Stack Overflow in itself provides an example for this with its design being tightly focused on technical questions and answers rather than long-drawn discussion-oriented software forums. It prioritizes information over usual conversation through its Q&A format [59]. Even though this format considers information to have a higher priority than conversation, the inherent nature of asking questions and then receiving answers is a form of conversation in itself. The difference is that this advocates moving away from the vague, generic answers or the further questions that a single question may generate in the form of a multi-layered tree. Instead, it focuses on having a single layer of conversation.

This concept was then translated to a task-oriented interface keeping the base functionality of the Visual Resume which is providing information about candidates from GH and SO intact. Questions were employed as a starting point, where the selection of one question would yield the standard set of relevant information (for a selected candidate) and this would be the unique answer set for this question. The answer set thus obtained acts as the filtered view of the information, being restricted to the domain of the question that was asked.

Lenses as per their usage in user applications and interfaces are "..detailed structures to represent the information of specific concern to them. ...devote most of their attention to the messages concerning these factors and ignore all the rest...." [58]. The factors here would be the specific candidate as well as the question that is selected (from a collection of questions). The detailed structures would be the tiles that would contain the filtered information as per these two factors.

Lenses could reduce the complexity of a user's view by removing data which is irrelevant to the current task (question in this case). They could also be used to filter a dataset so that only the objects of interest are shown [57]. When it comes to what could be considered an object or view of interest, there were two perspectives to be considered here– Onboarding and Trust.

An Onboarding Lens could focus on the information that would aid an evaluator to help a new hire get adjusted to the social and the performance aspects of their new jobs quickly and smoothly [39]. This could be for a recruiter looking to hire a candidate which would involve minimal cost and effort in onboarding. It could also be for a manager or a team lead who wants to onboard a new hire for his/her team. A Trust Lens could focus on information which could lead to an evaluator having positive confident expectations about the candidate's conduct [56] and be able to predict his/her future actions as a hire.

Since each of these lenses need to perform the task of filtering through queries, a sample set of questions that an evaluator may have (relevant to Onboarding and Trust) were identified.

4.1.1 Onboarding-Related Questions

1. How willing is a potential candidate to actively contribute to software and share projects openly on the site? [61]

2. What is the coping capacity of the candidate in terms of what and how much work can be assigned to him/her? [52]

3. What is the knowledge that the candidate already possesses, and hence what is the knowledge that the candidate needs? [19]

4. How capable is a candidate of independent judgement and actions such as ability to research problems or debug independently? [19]

5. What is the quality of a candidate's contributions? [70]

4.1.2 Trust-Related Questions

1. What is the coping capacity of the candidate in terms of the technical and mental capabilities of being able to handle different situations? [52]
2. How agreeable is a candidate with respect to qualities of cooperation, consideration and trust? [54]
3. How technology, people and management abilities of a candidate help gain trust? [38]
4. How can the quality of deliverables help establish trust? [38]

4.2 Mapping Metrics, Trust Factors and User Questions

Once the questions were identified, the information that was available from GH and SO had to be filtered through to form the relevant answer sets. For this, first the information fields for GH and SO had to be determined. These fields have been discerned from their sites [13][30] and the JSON files used with retrieved data for Visual Resume [70], and they have been listed out in Appendix A and Appendix B respectively.

Since trust may play a role at different stages of recruitment from selecting a candidate to the actual onboarding of the hire, this information needs to be considered with respect to it. So the next step was to establish the factors that influence trust. As discussed in Chapter 2, the trust factors as identified by Trainer were listed in Table 2.1 [76]. We also spoke about the mapping of these trust factors to collaborative traces in and around software artifacts

as tabulated by Trainer and Redmiles [77].

This was used as the rough basis for mapping the trust factors to the information fields (attributes from GH and SO). The Trust Factor-Attribute matrix for this was collated and can be accessed from Appendix C. A point to note here is that not all of the trust factors could be mapped to the corresponding relevant attributes from GH and SO such as in cases where matching attributes could not be found, for example, location.

This Trust Factor-Attribute matrix then needed to be mapped to the questions that we had identified earlier in the form of the answers for them. The trust factors, the collaborative traces that could help explain them, and the attributes from these platforms themselves, were used to determine the possible answers to the questions. The answers for the quality-related questions were as per the quality metrics being used in Visual Resume. The Question-Trust Factor-Attribute matrix is present in Appendix D.

4.3 TecSo Lens: Design and Implementation

TecSo (Technical-Social) Lens is intended as a prototype task-oriented interface that provides a filtered view of the information available on the Visual Resume. Unlike the latter, it does not require the user to separately access the GH and the SO information; it provides information from both platforms together. It allows the user to query the interface for information on any of the listed candidates based on the question selected from the ones provided. Since quite a few of the Onboarding and the Trust Lens questions seem to have a certain extent of overlap in terms of some of the questions as well as the answers displayed, the TecSo Lens is not intended to emulate either of them individually. Rather, as it is a prototype, two questions that could relate to both of the lenses were selected from the question set. They are:

1. What knowledge does the candidate already possess? [19][38]

2. What is the quality of the candidate's contributions? [70] [38]

This interface would then contain the options for selecting these two sample questions and five possible candidates whose GH and SO information was available as it was used in Visual Resume, and the TecSo Lens was based on it. Since the prototype needs to display only the information relevant to these two questions, I opted to use images of tiles containing this information instead of modifying the existing Visual Resume application. As it was already known as to what attributes would provide the answers to these questions, I modified the images of the tiles for each of the candidates, with respect to the answer set, i.e., the relevant information for it. So when a particular combination of one of these two questions and a particular candidate is selected and you click on the 'Search' button, tiles with relevant information from both GH and SO are displayed. This selection can be reset with the use of the 'Reset' button. One can perform side-by-side comparison of the candidates by selecting the same question and searching with different candidates. One can also view the answers for both questions for the same candidate by keeping the candidate the same and searching for different questions.

The answers that are displayed on the TecSo Lens use images of tiles, which may span across the levels of the "drill-down" hierarchy on Visual Resume. So the tiles here do not contain links that would allow you to browse and view more details. Instead, it is intended to provide the high-level answers to these questions with sufficient detail without resulting in much information overload. The screenshots that depict these scenarios and the usage of the TecSo Lens are in Appendix E.

Note that both the browsing interface of the Visual Resume and task-oriented interface of the TecSo Lens provide different ways of accessing the GH and SO information for a candidate. The appeal of either approach would depend on the user's personal preferences.

5

Evaluation: Pilot Study of TecSo Lens

As discussed in the previous chapter, the TecSo Lens is a prototype interface that allows one to filter information for candidates from their GitHub [13] and StackOverflow [30] profiles based on a question that the evaluator selects. A pilot study of this interface was conducted that looks at the different aspects of the research questions discussed in Chapter 3.

5.1 Goal

The pilot study conducted as a proof of concept of the TecSo Lens aims to examine the following areas:

- What information do evaluators look for while assessing candidates for a job role; through their personal experience and through the use of the TecSo Lens?
- How access to this information on the TecSo Lens affects their perception of trust about a candidate for a role?
- What is the other information that could have been useful for evaluating a candidate

on this application?

This study aims to use the task of recruitment for this purpose which will be assigned to the study participants, and will be discussed in further detail in the following sections.

5.2 Experiment Design

5.2.1 Task

Given a set of requirements for the role of a software development engineer, the study participants (as either recruiters or managers) are asked to use the TecSo Lens and utilize the accessible information for selecting a suitable candidate among the five provided.

5.2.2 Role Requirements

The replication of the real-life scenario of recruitment through a pilot study needed a realistic start. For this purpose, the requirements for the task had to be clearly defined. The requirements as listed in Appendix G and H have been taken from an Amazon job portal [2] with minor modifications. The instructions preceding the requirements are different for the recruiter (Appendix G) and the manager (Appendix H). The recruiter's instructions focus on the the recruitment process of selecting a new candidate. The manager on the other hand is encouraged to think in the long term about this recruitment process as well as which candidate would be easier to onboard [39] onto their team once the candidate has been selected.

5.2.3 Study Structure

The pilot study was conducted with five participants who are all graduate students at University of California, Irvine (UCI). They were first asked to fill in a pre-session questionnaire as in Appendix F. Erik Trainer's dissertation and more specifically the procedure followed for his user studies, was used as a guideline that helped structure the questionnaires used in this study.

This form asks the participant about his/her demographic information, educational and work experience. The latter two sections also have questions about their experiences working in projects and the good as well as bad experiences that they may have had with the other people. This was intended to encourage them to think about the qualities that they may or may not prefer in a fellow developer. They were also asked about their level of familiarity with GitHub and Stack Overflow, and experience as well as preferences for selecting a person for a project. This information helped with assigning roles to the participants wherein two of them were assigned the role of a recruiter and the other three were asked to play the role of a manager, with requirements provided in Appendix G and Appendix H respectively.

Along with these requirements, they were also asked to rate their initial levels of trust as per the questions provided, when there is no information available about the candidates. The studies were conducted separately and individually for each participant.

The participant was then asked to use the TecSo Lens with the requirements provided. They were told that they had five participants who had been short-listed, and were encouraged to think out loud [67] and use the application, until they reach a decision. Once they chose a candidate, they were asked to return the requirements document and were given a checklist (Appendix I) with the same requirements to reflect on how many did they check for while using the interface. This was again intended to encourage them to think along the lines of what they might have missed and also to identify what were the aspects that they focused

on during the selection process.

On completion of this checklist, an interview was conducted as per the questions listed in the Post-Session Questionnaire (Appendix J). The participant was asked various questions related to the choices made, expectations and experience while using the TecSo Lens interface. This included the reason for choosing a particular candidate, the rate of match with the requirements, the usefulness of the information provided alongside what more information could have been there with the platforms that it could be found on. They were also asked to rate their levels of trust with respect to the selected candidate for various scenarios. The interface was evaluated in terms of its usability, usefulness for accessing the information with the use of questions to do so and the general improvement suggestions for it were obtained as feedback from the participant.

5.3 Pilot Study Outcomes

5.3.1 Participant Profiles

As a disclaimer, since the participants for the pilot study were recruited from the pool of graduate students in the school of Information and Computer Science (ICS) at UCI, the participant responses cannot be considered as a holistic representation of the responses of the recruiter or manager roles that they have been asked to emulate. But their knowledge and past experience with recruiters and/or managers does give them a certain advantage to be able to play these roles as a preliminary glance into the information that these people may be looking for while searching for a suitable candidate for the job. The participant profiles have been listed as in Table 5.1.

Study Participants		Degree and Department	Age	Highest Degree Attained	Years of work Experience in general and in software-related fields (years)	Last time applied for a job/ internship	Mode of Last Recalled Conversation with Recruiter for a job/internship	Mode of Last Recalled Conversation with Manager/ Tech Lead/ Technical Developer for a job/internship
Recruiter	R1	MS (Informatics)	26	Undergraduate	2-3	Past 6 months	In- Person Interview	In- Person Interview
	R2	PhD (Informatics)	>26	Graduate	2-3	Past 6 months	Email, Phone Call	Email, Phone Call, Google Hangouts
Manager	M1	MS (ICS-CS)	23	Undergraduate	0-1	Past Week	Career Fair	Career Fair
	M2	MS (ICS-Networked Systems)	26	Undergraduate	2-3	Past 3 months	Phone Call	In- Person Interview
	M3	MS (Informatics)	25	Undergraduate	2-3	Past 6 months	Phone Call	In- Person Interview

Table 5.1: Study Participant Profiles

The two participants who were selected as recruiters, namely R1 and R2 have software knowledge but are not actively involved in software development. Also as depicted in the table below, they were not very familiar with GitHub and Stack Overflow. The other three participants, on the other hand, were the ones selected as managers, namely M1, M2 and M3. These three were relatively more familiar with both GitHub and Stack Overflow, and also had an active interest in software development.

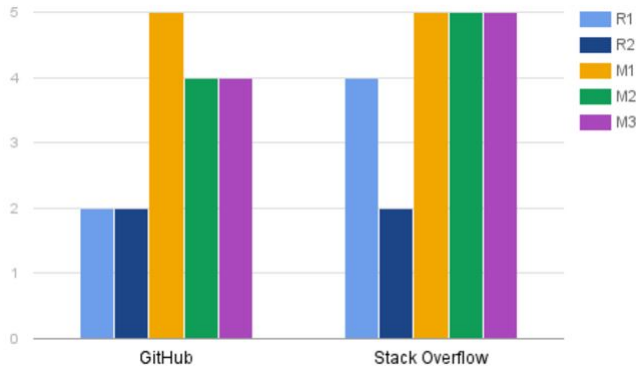


Figure 5.1: Familiarity with GitHub and Stack Overflow

Both set of participants have had applied for a job in the past six months and have had a conversation with a recruiter and a manager or a technical person in that period, that they

can recall. Hence, they are considered to be capable of knowing what each role tends to look for and thinking along the same lines for this task.

5.3.2 Recruiter, Manager and Participant Preferences

On the basis of their recent personal experiences with recruiters and managers/technical people as a part of the job application process, the participants discussed the areas that these focused on during the pre-session questionnaire. The figures below show the summation of the participant responses about what topics recruiters and managers/technical people discussed with them as a part of their most recent job application experience. Recruiters appear to focus more on the introduction and work experience whereas the managers or the tech leads focused more on the technical questions.

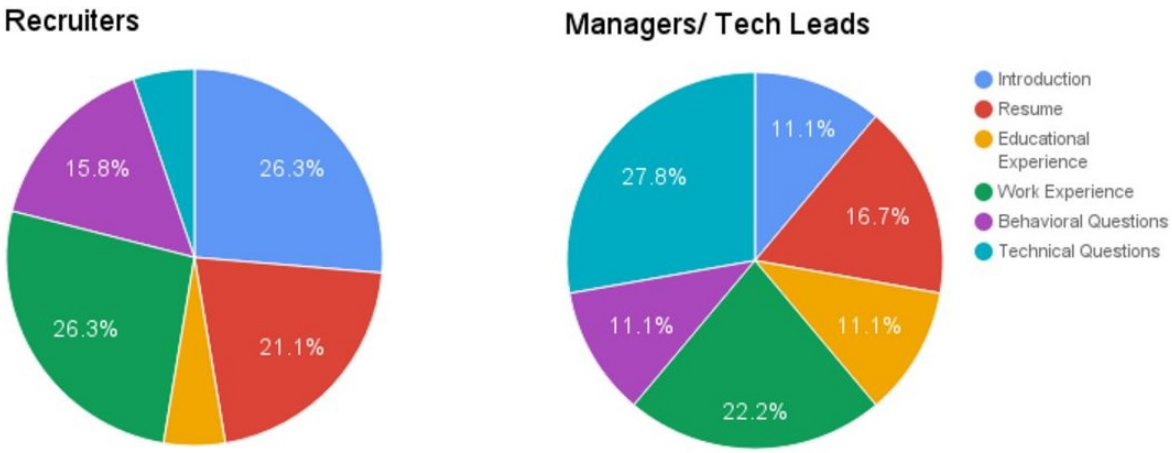


Figure 5.2: Topics discussed by Recruiters and Managers/Tech Leads from Participants' Experiences

As a part of the pre-session questionnaire, the participants were also asked to rate the skills that they would deem important in a potential hire. These have been summarized in the table below. From this we can see that technical, communication and team cohesion skills were given the highest priority across all participants. The ones playing the roles of the recruiters seemed to give a much lower priority to education and work experience, as

compared to those who were acting as managers. But this was a direct contrast to their own personal experiences where the recruiters that they spoke to were interested in their work experiences.

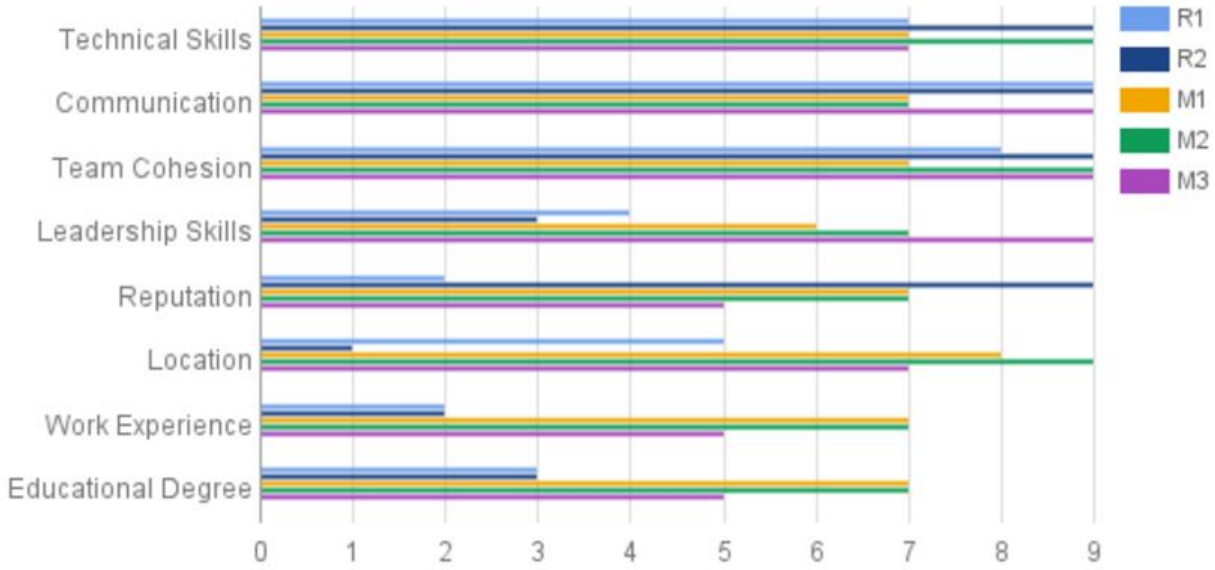


Figure 5.3: Study Participants' Skill Set Preferences

5.3.3 Useful Information for Selection of Candidates

All five participants selected the same candidate C1 amongst the five given through the use of the TecSo Lens with the requirements provided to them. R1 chose this candidate based on the enthusiasm and passion that R1 saw in his profiles on both GitHub and Stack Overflow. Both R1 and R2 said that C1 seemed to have more commits, pull requests, and had worked on more projects as compared to the others, so he seemed to be interested in his field. While using TecSo Lens, R1 also stated that even though C1 had answered less questions on Stack Overflow, it could mean that C1 was not very helpful or was not good with teams, but that was not considered prominent enough to change the decision.

Since all the participants who were asked to essay the role of managers were more familiar with GitHub and Stack Overflow, they focused on more specific aspects of the platforms for

their decisions. M1 was highly impressed by C1's contributions to the Rails repository on GitHub. M2 said that C1 had the most experience shipping code with good code quality, and the highest number of activities that had been closed on GitHub. M3 considered C1's code contributions, tests and the fact that he had opened and closed a number of issues as a sign that he was willing to work as well as multitask. Even though C1 did not appear to have worked on all the languages that were listed in the requirements, M3 took the knowledge that he knew Ruby very well as a sign that he had a potential to work in other languages. Even though he had less contributions on Stack Overflow, M3 observed that since 2 out of the 3 answers that he had given were accepted, it meant that whatever he answered was not wrong. M3 also said that, "I always believe that if you work a lot, you tend to learn a lot." This seemed to be the basis of the decisions of most participants as they selected the candidate with the most work as a representation of capability, irrespective of whether the candidate was a complete match with the given requirements.

With the use of the Post-Session Checklist that had been provided to the participants, they had a chance to reflect on which of the requirements they had focused on. Four out of the five participants answered that they did not know whether the candidate that they had selected, satisfied the educational requirements or not. One assumed that the candidate must have a Bachelors degree at least. All participants believed that the selected candidate possessed the knowledge on professional software engineering practices, had experience with the required software development skills, the ability to work in fast-paced development environment and a prior history of delivering high quality code. Four of the participants said that C1 had the motivation to drive tasks to completion and take ownership of projects, while two considered C1 to have experience collaborating in globally distributed teams from the information that they had seen on the TecSo Lens interface.

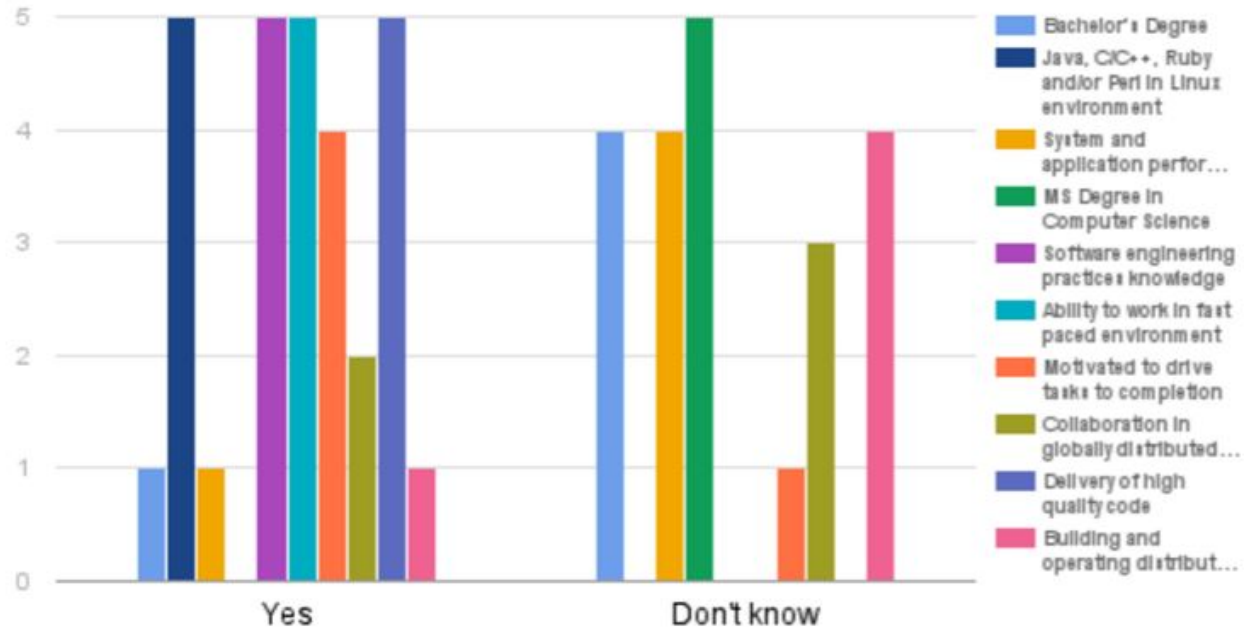


Figure 5.4: Match of Requirements with Skills of Selected Candidate from the TecSo Lens

5.3.4 Change in their levels of trust

At the beginning of the study, the participants had been asked to rate their initial levels of trust for a new candidate that they have no information about, with respect to the different aspects as provided. These questions were structured with the use of the study from Trainer's dissertation for the evaluation of trust [76].

Once the participants had selected a candidate, they were again asked to rate their levels of trust on the same questions with respect to the selected candidate. The trust levels seemed to increase overall from the initial levels that they had for an unknown candidate to the time that they had information about the candidate from the TecSo lens. Since trust issues are said to be easier to mitigate through translucence and social transparency, having access to information about a potential candidate which provides a positive influence could be useful to this end.

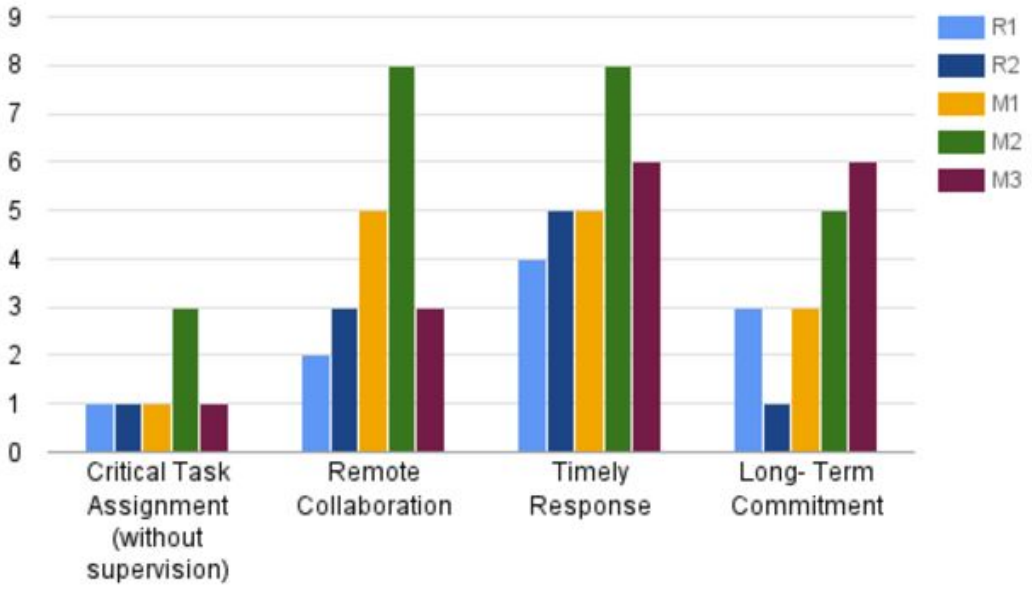


Figure 5.5: Trust Levels for an Unknown Candidate

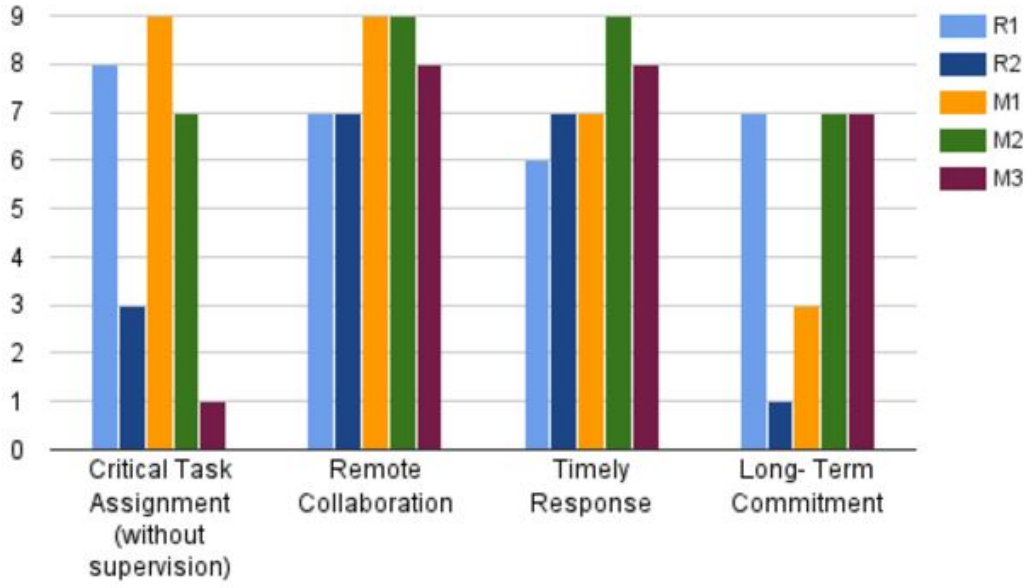


Figure 5.6: Trust Levels for Selected Candidate

5.3.5 What more information did the participants say that they needed to make their decision

All the participants said that they found the information as provided through the two questions on the TecSo Lens from the two platforms useful for making their decision. R1 and M2 felt that having more information would not have helped them with their choice. The others would have liked to have more information although both R1 and M3 felt that there was a lot of noise in the information provided, which could be distracting. M3 said that the commits, comments and code quality on GitHub and maybe the answers on Stack Overflow would be the only attributes that would matter for the selection. M1 considered the information sufficient for screening candidates for an interview but insufficient for a direct job offer. M3 also mentioned that it was important to get to know what a candidate is interested in, because they may be interested in hiring the candidate but the candidate's interests may not align with their own. On the whole, the candidates spoke about the different (and not necessarily) more information that could have been useful on the interface.

Additional Useful Information
Resume Details
Profile-Educational Experience
Years of Working Experience
Leadership
Behavioral Skills
Background Check
Reference Check
Videos of Candidate's Talks at conferences (Communication Skills)
Technical articles posted by Candidate on Social Media

Table 5.2: Additional Useful Information

5.3.6 Other platforms that could be included

Since the Visual Resume, and hence the TecSo Lens use only GitHub and Stack Overflow for their information, these may not contain sufficient information to be applicable to cues which may be found in other sources [70]. So as a candidate may have his/her data on a variety of platforms, the participants of the study were asked what other platforms would they have liked to have seen information from. All the participants said that they would have liked to have had access to information from LinkedIn [35]. As per M2, "LinkedIn shows experiences, not skills". M1 said that "It is easy to lie on a resume but it is harder to lie on LinkedIn because you have all your connections and your ex-colleagues present. So any discrepancies between the resume and LinkedIn would be evident". LinkedIn was also considered useful in terms of the recommendations and endorsements on skills that you get from others. This is useful because one can assess the authenticity of the person who has given the candidate a recommendation or an endorsement. YouTube videos of talks given at conferences were considered a possible way to gauge the candidate's communication skills. M3 spoke about Facebook [11], Google+[16] and Twitter [32] as forms of social media where a candidate's interests could be inferred from what the candidate shares. M3 also mentioned HackerRank [17] and MOOCs [20] as a means of seeing how willing a candidate is to learn. The former would also indicate motivation and time management. The combination of a MOOC certification and a resume was considered as a comprehensive source set. Table 5.3 lists the platforms as well as the useful information that participants mentioned that you could get from them.

5.3.7 Use and Usability of the TecSo Lens

Most participants found it moderately easy to use the TecSo Lens interface, wherein they may have faced confusion about the usage at the start, but it got easy as they got accustomed to it.

Platform	Information Provided
LinkedIn	Recommendations/Testimonials
	Work History
	Endorsements on Skills
Resume	Educational Qualifications
	GPA
	Work Experience
Personal Website	Introduction
YouTube	Videos of candidates talking at technical conferences
Facebook, Google+	Interests (eg. Technical Articles shared)
Twitter	Interests (eg. Technical Articles shared)
MOOCs	Certify candidate's skill set
	Interests
	Willing to learn
HackerRank	Technical skills
	Time Management
	Motivation

Table 5.3: Platforms and useful information accessible on it

They found it helpful for accessing information about a candidate, especially the option for side-by-side comparison of the candidates, where the candidates could be analysed by having the tiles for the same information open one next to the other. But this again considered a problem to view in a single screen when there were more than four candidates. Another issue pointed out was possible information overload, where some found the answer to the "What knowledge does the candidate already possess?".

With respect to the questions provided on the interface to view information about a specific aspect of the candidate, all participants said that it had been helpful for the information that it had provided which helped them in making their decision. They also said that they used the information provided from both the questions, and not just one. R2 said that the question on knowledge, had more information that could be utilized. On the other hand, M1 said that the question on the quality of contributions helped more since it helped to see who had made more commits to the core of the project. Other commits were assumed to be something relatively trivial such as changes to configuration files. M2 found GitHub a more

useful information source as compared to Stack Overflow.

With respect to the user interface of the TecSo Lens, there were several improvements suggested. One was a direct way to match the requirements to the informations wherein the participant wanted direct options for each requirement that could be selected to view just those aspects of the candidate for better comparison. Infographics could be used to indicate their relative skills such as their GPA or Java expertise. The source for this could be platforms where users are asked to rate their own expertise in a particular technical skill.

Some of the features did not contribute to an overall positive experience. The GitHub and Stack Overflow logos were considered distracting on the home screen. One was the extensive process for selecting the same question for multiple candidates by choosing them one by one. An alternative to this could have been having checkboxes that could select all the candidates and then selecting a question for them. There could also be a button to directly compare all candidates at once for a question. It would have been useful if the option selected on the drop box would have been visible even after selection. Filters focused on specific features were also suggested such as GitHub commits or the reputation from Stack Overflow.

Some of the participants suggested changes that were similar to the general functionality of Visual Resume, but which had been removed here in the TecSo Lens for the sake of simplicity of the prototype. Some examples include the ability to shift tiles, close them and the tabs for simultaneously viewing information like commit, issue and comment data. They also said that it would have been useful if key information could be selected from these tiles and copied for better comparison.

6

The Information Gap: Bridging Available and Desired Information

Visual Resume [70] and hence the TecSo Lens interface as discussed in the previous chapters utilize information from GitHub and Stack Overflow as the platforms of preference. But GitHub in itself may not be the ideal tool to provide a holistic overview of a candidate. Some of the issues could be that [53]:

- It is not necessary that every repository be a development project.
- Several projects are for personal use only or they may be inactive or have only a few commits.
- History could be manipulated by squashing multiple commits into a single one. [1].
- Only 10% of projects use pull requests.
- Not all developers and not all of their work have a presence on GitHub.
- The graph that is displayed shows your activity only for the past week.

As pointed out by some of the participants in the pilot study, having worked on a large number of repositories or projects does not necessarily indicate a level of expertise. As M3 elaborated, "He may have worked on projects with databases in them but it does not mean that he knows databases". M1 found the number of commits an inconclusive form evidence on its own for making decisions. But in conjunction with the information about the proximity to the core, it seemed to provide a relatively better context. Participants would have liked either the Resume or the LinkedIn profile for getting an overall summary of the candidate and then using developer aggregator tools like the TecSo Lens to look into specifics. These platforms were also considered useful means for accessing information related to their educational qualifications and work experience which were not available here. The table on other platforms in the previous chapter provides an overview of the other types of information that could contribute to the impression and impact of a candidate.

The developer aggregator profiles that we have today (including the TecSo Lens) do not give us a complete picture of the candidate. This could be from the aspects of the roles that the candidates are being considered for, the expectations set by the previous employees, the abilities of the existing team members for relative comparison of capabilities and collaboration. The latter though could be examined through tools like these by viewing the summarised profiles of each of these team members and then comparing them. So this appears to ignore the organizational aspect and fit, and instead solely focuses on the candidate, more specifically the activity traces of the candidate.

Let us look at this facet of information gap [48], where it is said that the information that we have is insufficient and other additional information is required for better context. As per Golman and Lowenstein, "when missing information is not available to an individual, demand for this information plays a role in decision making under uncertainty" [49]. The latter could also play a role in the level of trust established in the midst of this very uncertainty.

Summarizing the additional information areas that could help bridge the gap due to the

mismatch between the available information and the information that one desires to make a better informed decision as a part of this recruitment process are as follows:

6.1 Location

The impact of location on recruitment and eventual collaboration such as in the case of local or globally distributed developers could be better understood from the perspective of trust. Higher proximity is said to positively influence trust. Burt and Knez showed that trust was found to be the highest among contacts who met face-to-face on a daily basis than those who did not [41]. Lewicki and Bunker said that it is easier to generate and sustain trust when people are spatially clustered around the same location. Co-location and thus, interaction within the same time zone allows one to have more knowledge of others around them [55].

6.2 Communication Media

Communication media collectively refers to the different means of communication that developers tend to employ while interacting with each other. Most interactions take place over the computer nowadays and such computer-mediated communication tends to make the personal identities of those involved less visible [40]. As compared to direct communication where "something once said cannot be unsaid" [26], media of this kind varies not only in terms of reviewability and revisability where messages can be edited before sending or can even be undone after sending such as GMail Undo Send [34]. It also varies in terms of the audibility and visibility of an individual as well as sequentiality in terms of turn-taking in a group setting [40]. So knowing what form of media was used for communication could provide context to the statistical information one may already have such as the number of commits or issues on GitHub [13]. This on its own just gives us the quantity of their contributions,

and not the quality of the contributions as well as communication.

6.3 Educational Qualifications and MOOCS

Several organizations tend to lean towards job seekers with college degrees, more prominently, a Bachelor's degree [22]. Even the sample requirements (Appendix G and Appendix H) used for the pilot study involved a Bachelor's degree as a basic qualification and a Master's degree as a preferred qualification. Resumes and LinkedIn do contain this information but with this so-called 'degree inflation' [6], having educational qualifications listed out on a developer profile aggregator could be an optional but useful addition.

But "education is more than qualifications" [10]. It does not have to be comprised of only education obtained through a degree. It could also include what one may have learnt of their own self- interest such as through learning on MOOCs [20] like Coursera [8] and Udacity [33]. As pointed out by one of the participants in the pilot study, MOOCs not only provide certifications but show a willingness to learn. This could be a tipping factor if you have a candidate who does not have all the required technical skills but displays a strong potential to learn which could compensate for the lack of qualified knowledge.

6.4 Work Experience and Recommendations

Resumes are not the most reliable sources of information, although may be very comprehensive ones. Since there are no so-called references to verify their claims, candidates are free to be more deceptive about their experience. As per a study by Cornell University, 92% of the participants have lied at least once on their resume. LinkedIn on the other hand, could have potential for greater honesty. On a public profile, it is harder to lie as your claims

are easier to verify within your social network [25]. Akin to a panopticon [24], having your colleagues and acquaintances in your network, and not knowing which of them may spot a discrepancy in their working experience and your claims about the same may lead you to be more cautious before making tall claims.

One of the participants in the pilot study said that experience on GitHub or Stack Overflow does not indicate the actual years of work experience that a candidate may have. Work experience "equips you with certain soft skills such as teamwork, communication skills and commercial awareness" [3], which provides an additional advantage over the technical knowledge and experience one may have.

The recommendations and endorsements on skills available on LinkedIn help provides a layer of verification to the claims of a candidate. Having the ability to verify the identity and authenticity of the recommender as well provides another layer of verification to the candidate's profiles. The endorsements provided on the skill set of a candidate help quantify the skills which could aid with possible comparison of candidates on an aggregator tool.

6.5 Communication Skills

"Communication is key to Software Engineering." [7]

Since software engineering does not solely comprise of coding, all the other activities in its software development life cycle need to be taken into consideration. Software engineers spend a lot of time collaborating across different forms of media, reading and writing requirements and design, reviewing code and engaging in discussions in face-to-face meetings [7].

Comments on platforms like GitHub and Stack Overflow may be in informal language and may consist of a few lines at most. This may not be a sufficient basis to judge a candidate's

written communication skills. Personal Websites and Technical Blogs could be more relevant sources that could be useful in evaluating written communication skills.

When it comes to in-person spoken communication skills, it gets harder to evaluate this on aggregator tools, since they appear to focus on platforms with textual content. As per a suggestion by a participant in the pilot study, YouTube videos of a candidate giving talks at conferences could be useful means for gauging his/her spoken communication skills.

6.6 Passion and Motivation

Candidates may list out or describe their passion and commitment through objective statements or cover letters that they provide [23]. But since this does not really demonstrate these qualities, it calls for a need to look at alternative platforms for the same purpose. Evaluators may want to look at self-motivated and passionate candidates over highly-qualified ones [18][23]. Platforms like HackerRank [17], MOOCs and even GitHub and Stack Overflow do indicate self-motivation by sheer participation. But the former two have an added advantage over the latter that indicate a willingness to learn and go beyond means of formal education for learning and testing themselves.

6.7 Summary

These are some of the areas that could help shed light on different facets of a candidate if the information is made available in an aggregated fashion. This may help bridge the gap wherein evaluators are asked to make judgements in a state of uncertainty, not having all the information that they need to effectively make that decision.

7

Discussion

This research was initially intended as an alternative means of accessing information which was already available through the Visual Resume through different visualizations. I was to also look at what more information could be included that could aid recruiters and/or managers or any other person in the technical domain (collectively referred to as evaluators) to trust a candidate through the information provided about him/her. For this I started with the information that was already available from GitHub (GH) and Stack Overflow (SO) on the Visual Resume as in Appendix A and Appendix B.

I wanted to look at what information do these evaluators look for in a candidate for a potential role. Apart from the straight-forward answer of requirements, I also wanted to consider this from two different perspectives of onboarding and trust. The former dealt with the information that would enable an evaluator to know how easy it would be train and onboard a candidate onto a team or an organization. The latter was about the expectations that an evaluator may have about a candidate in terms of fitting in with the work culture and the general requirements of the role. Since usage of the Visual Resume interface seemed to have a browsing-oriented interaction, an alternative means of having a more focused,

task-oriented form of interaction was examined. So the information that evaluators may need to know from the above mentioned two perspectives could be proposed in the form of questions where the answers would be relevant to it. The Onboarding and Trust Lens were thus planned to be two forms of filtering that along with the original view from the Visual Resume would allow one to view the information from GH and SO with respect to the questions for either onboarding or trust.

The two lenses were to form a prototype interface where the user would have the option to select a lens and then the questions as per the lens for one among the five candidates already accessible through the Visual Resume application. The information then displayed through tiles as in Visual Resume would be relevant to the lens, question and candidate selected. Also, the GH and SO information would be visible simultaneously for these criteria rather than the separate views as accessible on the previous application.

A sample set of four to five questions were proposed for each lens on the prototype interface. Once I arrived at a final list of questions for both onboarding and trust as per the information that evaluators need to assess a candidate, I mapped them to the trust factors from literature and the information available on GH and SO as in Appendix C and Appendix D. The trust factors were included as any form of interaction between the evaluators and the candidate would involve the expectations the evaluator would have of the candidate.The trust here is about the trust an evaluator has with respect to the suitability of a candidate for a role and the ease in onboarding, rather than how much an evaluator may trust the information provided through this interface. As I could see some overlap among the questions and in the interest of time, rather than implementing separate lens interfaces for both onboarding and trust, I opted to implement a single lens that would filter the information as per the sample questions selected on the interface for a particular candidate. This lens was named the TecSo Lens and two sample questions were selected from the overall sample set of questions for this prototype interface.

Since the TecSo Lens was to be a prototype interface with two sample questions and the GH and SO information already mapped to it as per Appendix D, I chose to modify the tile images for the information through Photoshop rather than altering the original Visual Resume application. Also, the TecSo Lens was not intended as the focus of the research rather it was proposed as the means through which one could identify the information that was missing on this application and thus aid in improving future aggregator applications. Some of this information gap was identified from literature in terms of what evaluators look for as compared to what was available on GH and SO.

To look into the usefulness of the questions for accessing this information, examine the effect of it on trust with respect to a candidate and identify the information gap if any, a pilot study was conducted with five graduate students at the University of California, Irvine. This study has been discussed in detail in the previous two chapters.

The study shed some light on the information that the participants found useful in making their decision, and also informed us about the other information as well as platforms they may have liked information from for a more holistic view. It also showed the general increase in trust for a selected candidate as compared to when a candidate was unknown with no available information. The participants' previous experiences with recruiters and managers provided us with the information that the two respectively look for in a candidate. This may have also shaped a participant's behavior when asked to select a candidate as either a recruiter or a manager with the use of the TecSo Lens.

This study along with other literature showed that GH and SO were not sufficient on their own as information sources for evaluators to trust a candidate and more information as per the previous chapter could prove useful in improving developer profile aggregator applications in the future. But since the study was a pilot one conducted with only five graduate students asked to pose as recruiters and managers, the results cannot be considered a holistic representation of the behavior and expectations of real-world recruiters and managers.

Extensive studies would need to be conducted with real-world representatives of these roles with a larger pool of candidates. This could utilize a wider set of platforms for information without resulting in information overload, for more accurate representative behavior and expectations of evaluators with respect to candidates.

8

Conclusion and Future Work

Developer Profile Aggregators may be used by evaluators for analyzing the suitability of a candidate for a vacant role in their organization. They may need certain information about this candidate for effective evaluation of the candidate's skills in comparison with the given requirements. The Visual Resume was taken as the developer profile aggregator tool that formed the foundation of this research, and the TecSo Lens prototype interface was based on the information that it uses as well as the basic layout of tiles that it used for presenting this information.

This thesis seeks to answer the three research questions as listed in Chapter 3. Since Visual Resume utilizes only GitHub and Stack Overflow data, the first two questions involve the use of these as the base set. The first of the questions deals with what information helps an evaluator determine the technical competency and social fit of a candidate for a role. This information is partially dependent on the requirements of a role, and partially on the generic expectations of the evaluator or the organization. Focusing on the latter, what an evaluator may be looking for was examined through the guidelines in Chapter 2 and the possible questions that they may have as identified in Chapter 4. The pilot study of TecSo

Lens as discussed in Chapter 5 also contributed to the possible aspects that evaluators may be looking for. This includes the personal experiences of the participants with recruiters, managers and tech leads with respect to the latter's expectations of them. It also involved the features of GitHub and Stack Overflow for which the participants displayed an affinity through the pre-questionnaire, as well as while using the TecSo Lens playing the role of either a recruiter or a manager tasked with selecting a suitable candidate as per requirements.

The second research question examines how trust plays a role in context of the information as identified above. Trust Factors being aspects about a candidate that could positively or negatively impact the expectations of an evaluator have been listed in Chapter 2 as per Trainer's dissertation. The mapping of these trust factors to collaborative traces in real-life artifacts helped map each of the trust factors to the existing data fields in use from GitHub and Stack Overflow as listed in Appendix C. This was then again mapped to the questions identified in Chapter 4 and listed in Appendix D to correlate the second research question and the first, i.e., trust and information about a candidate. The pilot study provided a preview of how trust could increase when the initial levels of trust with respect to an unknown candidate were compared to their levels of trust with respect to the candidate they selected after using the TecSo Lens as illustrated in Chapter 5.

The third question aimed to identify what information from GitHub and Stack Overflow could be used with respect to the first two questions, and what is the missing information that could be included in future applications. The first part of this question has been addressed in Appendix D; the second part has been discussed in Chapter 6 along with some of the platforms that could provide this information.

8.1 Future Work

From this research I have come to understand that GitHub and Stack Overflow as platforms may not contain all the information that is required for evaluators to make informed decisions.

We may need to look at alternative platforms that could help satisfy this requirement instead, and I have done so to a certain extent in Chapter 5 and Chapter 6, which discuss platforms that could potentially provide relevant information. Since this involved a pilot study with graduate students asked to play the roles of either recruiters or managers, the results cannot be conclusive proof of the kinds of information that recruiters and managers look for in a candidate, or the only platforms where such information can be found.

But this information gap as identified in Chapter 6 could be used as a starting point for a study with real-world recruiters and managers to better gauge what information they may be looking for in a potential candidate, and the platforms that would be considered as reliable sources for the same. This information gap once holistically identified could be used to develop a developer profile aggregator that could help evaluators make better-informed decisions, while taking care not to indulge in information overload.

The TecSo Lens may have had too much information presented at once which was perceived as noise by some of the participants. Accordingly an evaluation of the information within this information gap to prioritize it by importance could also help with consolidating it across multiple platforms. One would also need to ensure that it remains concise enough for quick comprehension by evaluators to make concrete and well-informed decisions during the recruitment process.

Bibliography

[1] About pull request merge squashing - user documentation. https://help.github.com/articles/about-pull-request-merge-squashing/. (Accessed on 05/16/2016).

[2] Amazon.jobs. https://www.amazon.jobs/en-gb/jobs/407199. (Accessed on 05/19/2016).

[3] The benefits of work experience - the iet. http://www.theiet.org/students/work-careers/work-experience/benefits.cfm. (Accessed on 05/21/2016).

[4] Calculate your github hacker score — hacker news. https://news.ycombinator.com/item?id=2468830. (Accessed on 05/16/2016).

[5] Coderwall — a community of great programmers and their programming. https://coderwall.com/. (Accessed on 05/16/2016).

[6] College degree required by increasing number of companies - the new york times. http://www.nytimes.com/2013/02/20/business/college-degree-required-by-increasing-number-of-companies.html?pagewanted=2&_r=3&hp&adxnnlx=1361365497-aWz0hS0WEabQsBzifK/6ZQ. (Accessed on 05/21/2016).

[7] Communication is key to software engineering — donnie flood — linkedin. https://www.linkedin.com/pulse/communication-key-software-engineering-donnie-flood. (Accessed on 05/21/2016).

[8] Coursera - free online courses from top universities. https://www.coursera.org/. (Accessed on 05/21/2016).

[9] The difference between onboarding and training — hr resolutions. http://www.hrresolutions.com/blog/2015/11/03/the-difference-between-onboarding-and-training.html. (Accessed on 05/12/2016).

[10] Education is more than qualifications - rsa. https://www.thersa.org/discover/publications-and-articles/rsa-comment/2014/04/education-is-more-than-qualifications. (Accessed on 05/21/2016).

[11] Facebook - log in or sign up. https://www.facebook.com/. (Accessed on 05/21/2016).

[12] Forget linkedin: Companies turn to github to find tech talent - cnet. http://www.cnet.com/news/forget-linkedin-companies-turn-to-github-to-find-tech-talent/. (Accessed on 05/15/2016).

[13] Github. https://github.com/. (Accessed on 05/15/2016).

[14] Github - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/GitHub. (Accessed on 05/15/2016).

[15] Github pours energies into enterprise – raises $100 million from power vc andreessen horowitz — techcrunch. http://techcrunch.com/2012/07/09/github-pours-energies-into-enterprise-raises-100-million-from-power-vc-andreesen-horowitz/. (Accessed on 05/15/2016).

[16] Google+. https://plus.google.com/. (Accessed on 05/21/2016).

[17] Hackerrank — technical recruiting — hiring the best engineers. https://www.hackerrank.com/. (Accessed on 05/21/2016).

[18] Hire self-motivated people — the single smartest thing a hiring manager can do — ere. http://www.eremedia.com/ere/hire-self-motivated-people-the-single-smartest-thing-a-hiring-manager-can-do/. (Accessed on 05/21/2016).

[19] How to onboard software engineers – interview with kate heddleston - fog creek blog. http://blog.fogcreek.com/how-to-onboard-software-engineers-interview-with-kate-heddleston/. (Accessed on 05/18/2016).

[20] Massive open online course - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Massive_open_online_course. (Accessed on 05/21/2016).

[21] Masterbranch. https://masterbranch.com/. (Accessed on 05/16/2016).

[22] More employers saying, 'no bachelor's degree, no job'. http://www.businessnewsdaily.com/7103-no-degree-no-job.html. (Accessed on 05/21/2016).

[23] Motivation or qualifications? how passion is often more important than credentials - christopher frederick. http://www.chrisfred.com/motivation-qualifications-way-candidates-employers-approach-job-search-can-even-important-credentials/. (Accessed on 05/21/2016).

[24] Panopticon - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Panopticon. (Accessed on 05/21/2016).

[25] People lie less on linkedin resumes – usatoday.com. http://usatoday30.usatoday.com/tech/news/story/2012-03-14/linkedin-resume/53530838/1. (Accessed on 05/21/2016).

[26] Quote by philippa gregory: "words have weight, something once said cannot b...". http://www.goodreads.com/quotes/361715-words-have-weight-something-once-said-cannot-be-unsaid-meaning. (Accessed on 05/21/2016).

[27] Reduce the risk, hire from open source (loud thinking). http://david.heinemeierhansson.com/arc/000505.html. (Accessed on 05/15/2016).

[28] Seeing is believing - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Seeing_Is_Believing. (Accessed on 05/15/2016).

[29] Social networking fact sheet — pew research center. http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/. (Accessed on 05/15/2016).

[30] Stack overflow. http://stackoverflow.com/. (Accessed on 05/15/2016).

[31] Time spent online doubles in a decade — ofcom. http://media.ofcom.org.uk/news/2015/time-spent-online-doubles-in-a-decade/. (Accessed on 05/15/2016).

[32] Twitter. https://twitter.com/?lang=en. (Accessed on 05/21/2016).

[33] Udacity - free online classes & nanodegrees. https://www.udacity.com/. (Accessed on 05/21/2016).

[34] Undo sending your mail - gmail help. https://support.google.com/mail/answer/1284885?hl=en. (Accessed on 05/21/2016).

[35] Welcome! — linkedin. https://www.linkedin.com/. (Accessed on 05/21/2016).

[36] Why conversational design is the future — inside intercom — medium. https://medium.com/intercom-inside/why-conversational-design-is-the-future-2c05f65aa68e#.33uw0x285. (Accessed on 05/18/2016).

[37] B. Al-Ani and D. Redmiles. Trust in distributed teams: Support through continuous coordination. *Software, IEEE*, 26(6):35–40, 2009.

[38] M. A. Babar, J. M. Verner, and P. T. Nguyen. Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *Journal of Systems and Software*, 80(9):1438–1449, 2007.

[39] T. N. Bauer. Onboarding new employees: Maximizing success. *SHRM Foundation's Effective Practice Guideline Series*, 2010.

[40] M. J. Bietz. Effects of communication media on the interpretation of critical feedback. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 467–476. ACM, 2008.

[41] R. S. Burt and M. Knez. Trust and third-party gossip. *Trust in organizations: Frontiers of theory and research*, 68:89, 1996.

[42] J. Chan. The importance of onboarding developers. http://business.stackoverflow.com/blog/2014/04/16/the-importance-of-onboarding-developers, April 2014.

[43] L. Csorny. Careers in the growing field of information technology services : Beyond the numbers: U.s. bureau of labor statistics. http://www.bls.gov/opub/btn/volume-2/careers-in-growing-field-of-information-technology-services.htm, April 2013.

[44] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15. ACM, 2011.

[45] A. Doyle. Tips for job searching where employers are recruiting. http://jobsearch.about.com/od/findajob/a/jobsearchcomp.htm, April 2016.

[46] M. J. Eppler and R. A. Burkhard. Visual representations in knowledge management: framework and cases. *Journal of Knowledge Management*, 11(4):112–122, 2007.

[47] T. Erickson and W. A. Kellogg. Social translucence: an approach to designing systems that support social processes. *ACM transactions on computer-human interaction (TOCHI)*, 7(1):59–83, 2000.

[48] R. Golman and G. Loewenstein. Curiosity, information gaps, and the utility of knowledge. *Information Gaps, and the Utility of Knowledge (April 16, 2015)*, 2015.

[49] R. Golman and G. Loewenstein. An information-gap framework for capturing preferences about uncertainty. In *Proceedings of the fifteenth conference on theoretical aspects of rationality and knowledge*, 2015.

[50] C. Gutwin, R. Penner, and K. Schneider. Group awareness in distributed software development. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 72–81. ACM, 2004.

[51] D. Hyerle. Thinking maps: Visual tools for activating habits of mind. *Activating and engaging habits of mind*, pages 46–58, 2000.

[52] K. J. Jenne and M. Henderson. Hiring a director for a nonprofit agency: A step-by-step guide. *PoPULAR GovERNMENT*, page 25, 2000.

[53] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian. The promises and perils of mining github. In *Proceedings of the 11th working conference on mining software repositories*, pages 92–101. ACM, 2014.

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

[55] R. Lewick and B. Bunker. Developing and maintaining trust in work relationships. *Trust in organizations: Frontiers of theory and research*, page 114, 1996.

[56] R. J. Lewicki, D. J. McAllister, and R. J. Bies. Trust and distrust: New relationships and realities. *Academy of management Review*, 23(3):438–458, 1998.

[57] J. Looser, M. Billinghurst, and A. Cockburn. Through the looking glass: the use of lenses as an interface tool for augmented reality interfaces. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 204–211. ACM, 2004.

[58] T. W. Malone, K. R. Grant, and F. A. Turbak. *The information lens: an intelligent system for information sharing in organizations*, volume 17. ACM, 1986.

[59] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2857–2866. ACM, 2011.

[60] A. Mann and T. Nunes. After the dot-com bubble: Silicon valley high-tech employment and wages in 2001 and 2008. `http://www.bls.gov/opub/btn/archive/after-the-dot-com-bubble-silicon-valley-high-tech-employment-and-wages-in-2001-and-2008-pdf.pdf`, August 2009.

[61] J. Marlow and L. Dabbish. Activity traces and signals in software developer recruitment and hiring. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 145–156. ACM, 2013.

[62] J. Marlow, L. Dabbish, and J. Herbsleb. Impression formation in online peer production: activity traces and personal profiles in github. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 117–128. ACM, 2013.

[63] D. J. McAllister. Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations. *Academy of management journal*, 38(1):24–59, 1995.

[64] P. McCuller. *How to Recruit and Hire Great Software Engineers: Building a Crack Development Team*. Apress, 2012.

[65] D. Movshovitz-Attias, Y. Movshovitz-Attias, P. Steenkiste, and C. Faloutsos. Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 886–893. IEEE, 2013.

[66] K. Nico and B. Anjum. Resumes: Your ticket to the interview, April 2016. Talk by Kim Nico and Bushra Anjum, Amazon at CWIC SoCal 2016.

[67] J. Nielsen. *Usability Engineering*. AP Professional, 1993.

[68] C. Parnin, C. Treude, L. Grammel, and M.-A. Storey. Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. *Georgia Institute of Technology, Tech. Rep*, 2012.

[69] J. F. Sargent Jr. The u.s. science and engineering workforce: Recent, current, and projected employment, wages, and unemployment. `https://www.fas.org/sgp/crs/misc/R43061.pdf`, February 2014.

[70] A. Sarma, X. Chen, S. K. Kuttal, L. Dabbish, and Z. Wang. Hiring in the global stage: Profiles of online contributions. In *International Conference of Global Software Engineering 2016 (accepted)*. ICGSE, 2016.

[71] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider. Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 103–116. ACM, 2013.

[72] J. Snyder. Visual representation of information as communicative practice. *Journal of the Association for Information Science and Technology*, 65(11):2233–2247, 2014.

[73] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*, pages 1379–1392. ACM, 2015.

[74] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, pages 100–116. ACM, 2014.

[75] Y. R. Tausczik and J. W. Pennebaker. Predicting the perceived quality of online mathematics contributions from users' reputations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1885–1888. ACM, 2011.

[76] E. H. Trainer. *Supporting trust in globally distributed software teams : the impact of visualized collaborative traces on perceived trustworthiness*. PhD thesis, University of California, Irvine, 2012.

[77] E. H. Trainer and D. F. Redmiles. Foundations for the design of visualizations that support trust in distributed teams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 34–41. ACM, 2012.

[78] R. K. Wilkinson. How large is the u.s. s&e workforce? `http://www.nsf.gov/statistics/infbrief/nsf02325/`, July 2002.

[79] J. M. Wilson, S. G. Straus, and B. McEvily. All in due time: The development of trust in computer-mediated and face-to-face teams. *Organizational behavior and human decision processes*, 99(1):16–33, 2006.

[80] Y. Wu, J. Kropczynski, P. C. Shih, and J. M. Carroll. Exploring the ecosystem of software developers on github and other platforms. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 265–268. ACM, 2014.

[81] Z. Yan, C. Liu, V. Niemi, and G. Yu. Exploring the impact of trust information visualization on mobile application usage. *Personal and Ubiquitous Computing*, 17(6):1295–1313, 2013.

[82] M. Zhou and A. Mockus. What make long term contributors: Willingness and opportunity in oss community. In *Proceedings of the 34th International Conference on Software Engineering*, pages 518–528. IEEE Press, 2012.

Appendix A

GitHub Data Fields

The raw data that is captured for a developer on GitHub can be translated to the following fields:

- Name
- Login
- Website
- ID
- GitHub page
- Avatar
- Creation Date
- Followers
- Issue Count
- Commit Count

- Comment Count
- Activity (for a particular time period)
  - Commit Count
  - Comment Count
  - Issue Count
  - Closed Issues
  - Non-merged Pull Requests
  - Merging Pull Requests
  - Failed Commits
  - Passed Commits
  - Centrality of Commits to core:
    * Low Centrality
    * Median Centrality
    * High Centrality
- Repository
  - Name
  - Language
  - Forks
  - Watchers
  - Website
  - Creation Date
  - isFork (Whether it has been forked)

  - Description
  - Contributors
  - Collaborators
  - Activity (for a particular time period)
    * Commit Count
    * Comment Count
    * Issue Count
    * Closed Issues
    * Non-merged Pull Requests
    * Merging Pull Requests
    * Failed Commits
    * Passed Commits
    * Centrality of Commits to core:
      · Low Centrality
      · Median Centrality
      · High Centrality
  - Commit Count
  - Comment Count
  - Issue Count
- Commits
  - URL
  - Message
  - Date
  - Hash

  - State
  - Centrality
  - Repository
- Comments
  - Date
  - Parent URL
  - Body
  - Issue Number
  - Repository
- Issues
  - URL
  - Body of the issue
  - Date
  - Title
  - Ismerged (whether it has been merged)
  - Repository
  - Open/Close

Appendix B

Stack Overflow Data Fields

The raw data that is captured for a developer on Stack Overflow can be translated to the following fields:

- Display Name
- ID
- Avatar
- Creation Date
- Website
- Reputation
- Question Count
- Answer Count
- Comment Count
- Accepted Count

- Non-Accepted Count
- Activity (for a particular time period)
  - Answer Count
  - Accepted Count
  - Non- Accepted Count
- Tags
  - Activity (for a particular time period)
    - Answer Count
    - Accepted Count
    - Non- Accepted Count
  - Answer Count
  - Accepted Count
  - Non- Accepted Count
  - Related Tags (tag list)
- Questions
  - ID
  - Date
  - Score
  - Body
  - Title
  - Favorite
  - Tags

  - Comment Count
  - View Count
  - Answer Count
- Answers
  - ID
  - Date
  - Score
  - Body
  - Title
  - Favorite
  - Tags
  - Comment Count
  - Accepted
- Comments
  - ID
  - Date
  - PostID
  - Score
  - Body
  - Tags

Appendix C

Trust Factors- Attributes Matrix

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Frequent Initiations & Responses	Comment Count Commit Count Activity-specific • Comment Count • Commit Count Repo-specific • Activity-specific • Comment Count • Commit Count Commits • Date • Repository • URL	Question Count Answer Count Comment Count Activity-specific • Answer Count • Commit Count Questions • ID • Date • Score • Title • Favorite • Tags

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Frequent Updates of Project Progress	Comments • Date • Issue Number • Repository • Parent URL Issue Count Comment Count Commit Count Activity-Specific • Commit Count	• Comment Count • View Count • Answer Count Answers • ID • Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID • Post ID • Body • Score • Tags Answer Count Comment Count Activity-Specific • Answer Count • Commit Count

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
	• Comment Count • Issue Count • Close Issue • Non-Merged PR • Merged PR Failed Commit • Passed Commits • Low Centrality • Median Centrality • High Centrality Repository-specific • Activity-Specific • Comment Count • Commit Count • Issue Count Issues • URL • Body • Date • Title • Is Merged • Repository • Open/Close Commits	Answers • ID Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID • Post ID • Body • Score • Tags

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Reputation	• Date • URL • Repository Comments • Date • Issue Number • Repository • Parent URL Followers Each repository • Forks • Watchers • Contributors • Collaborators	Reputation Accepted Count Non-Accepted Count Activity-Specific • Accepted Count • Non-Accepted Count Tag-Specific • Activity-Specific • Accepted Count • Non-Accepted Count Question • Favorite • View Count Answers • Score • Favorite

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
		• Accepted Comments • Score
Use of Multiple Communication Media	NA (Different means of communication on this platform itself could be comments on activities, repos,issue descriptions and commits)	NA (Different means of communication on this platform itself could be questions, answers, tags or comments)
Same location	Location and Time Zone information does not appear to be present although the name, ghpage and website could be used to look these up if publically available.	Location and Time Zone information does not appear to be present although the display name and website could be used to look these up if publically available.
Role	Name Login Avatar Followers Following Repo-Specific • Description	ID Display Name Avatar Reputation Questions • Body Answers

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Availability	• Name • Website • Contributors • Collaborators Name Login Commit Count Comment Count Issue Count Activity-Specific • Commit Count • Comment Count • Issue Count Repo-Specific • Description • Name • Website Issues • Date • Open/Close (No time/location specific information)	• Body Comments • Body Display Name Reputation Question Count Answer Count Comment Count Questions • Date • Body • View Count • Answer Count • Comment Count Answers • Date • Body • Comment Count Comments • Date • Body
Leadership	Followers Following	Reputation

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Frequent Meetings	NA	NA
Years of Experience	Creation Date	Creation Date
Homophily	Name	Tags
	Login	
	Avatar	
	Followers	
	Following	
	Repo-Specific • Description • Name • Language • Website • Contributors • Collaborators	
Shared Information	Comments	Comments
Shared Photographs	Avatar	Avatar
Team Diversity	Repo-Specific • Contributors • Collaborators	-
Monitoring	-	-
High/low team size	Contributors	-
Project size/type	Repo-Specific • Description • Name	

Trust Factors	Developer Attributes in Visual Resume (GitHub and Stack Overflow)	
	GitHub	Stack Overflow
Expertise	• Language • Website • Contributors • Collaborators Followers	Reputation
Gender	Name Avatar Website	Display Name Avatar Website
Language	Repo-Specific Language	Tags

Appendix D

Questions- Trust Factors- Attributes Matrix

D.1 Onboarding

Onboarding questions focus on the information that would aid an evaluator to help a new hire get adjusted to the social and the performance aspects of their new jobs quickly and smoothly. This could be for a recruiter looking to hire a candidate which would involve minimal cost and effort in onboarding. It could also be for a manager or a team lead who wants to onboard a new hire for his/her team.

The Trust Factors-Attributes Matrix from Appendix C has been used for mapping the GitHub and StackOverflow attributes to these questions.

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
How willing is a potential candidate to actively contribute to software and share projects openly on the site?	Frequent Initiations & Responses	Comment Count Commit Count Activity-Specific • Comment Count • Commit Count Repo-Specific • Activity-Specific • Comment Count • Commit Count Commits • Date • Repository • URL Comments • Date • Issue Number • Repository • Parent URL	Question Count Answer Count Comment Count Activity-Specific • Answer Count • Commit Count Questions • ID • Date • Score • Title • Favorite • Tags • Comment Count • View Count • Answer Count Answers • ID • Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
What is the coping capacity of the candidate in terms of what and how much work can be assigned to him/her?	Frequent Updates of Project Progress	Website Issue Count Comment Count Commit Count Activity-Specific • Commit Count • Comment Count • Issue Count • Close Issue • Non-Merged PR • Merged PR • Fail Commit • Pass Commit • Low Centrality • Median Centrality • High Centrality Repo-Specific • Activity-Specific • Comment Count • Commit Count	• Date • ID • Post ID • Body • Score • Tags Website Answer Count Comment Count Activity-Specific • Answer Count • Commit Count Answers • ID • Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID • Post ID • Body

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		• Issue Count	• Score
		Issues	• Tags
		• URL	
		• Body	
		• Date	
		• Title	
		• Is Merged	
		• Repository	
		• Open/Close	
		Commits	
		• Date	
		• URL	
		• Repository	
		Comments	
		• Date	
		• Issue Number	
		• Repository	
		• Parent URL	
	Availability	Name	Display Name
		Login	Reputation
		Commit Count	Question Count
		Comment Count	Answer Count
		Issue Count	Comment Count
		Activity-Specific	Questions
		• Commit Count	• Date
		• Comment Count	• Body

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		• Issue Count Repo-Specific • Description • Name • Website Issues • Date • Open/Close	• View Count • Answer Count • Comment Count Answers • Date • Body • Comment Count Comments • Date • Body
What is the knowledge that the candidate already possesses?	Language Frequent Updates of Project Progress	Repo-Specific • Language Issue Count Comment Count Commit Count Activity-Specific • Commit Count • Comment Count • Issue Count • Close Issue • Non-Merged PR • Merged PR • Failed Commits • Passed Commits • Low Centrality • Median Centrality	Tags Answer Count Comment Count Activity-Specific • Answer Count • Commit Count Answers • ID • Date • Score • Title • Favorite • Tags • Comment Count • Accepted

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		• High Centrality Repo-Specific • Activity-Specific • Comment Count • Commit Count • Issue Count Issues • URL • Body • Date • Title • Is Merged • Repository • Open/Close Commits • Date • URL • Repository Comments • Date • Issue Number • Repository • Parent URL	Comments • Date • ID • Post ID • Body • Score • Tags
How capable is a candidate of independent judgement	Availability	Name Login Commit Count	Display Name Reputation Question Count

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
and actions such as ability to research problems or debug independently?		Comment Count Issue Count Activity-Specific • Commit Count • Comment Count • Issue Count Repo-Specific • Description • Name • Website Issues • Date • Open/Close	Answer Count Comment Count Questions • Date • Body • View Count • Answer Count • Comment Count Answers • Date • Body • Comment Count Comments • Date • Body
	Leadership	Followers Following	Reputation
	Frequent Updates of Project Progress	Issue Count Comment Count Commit Count Activity-specific • Commit Count • Comment Count • Issue Count • Close Issue • Nonmerged PR	Answer Count Comment Count Activity-Specific • Answer Count • Commit Count Answers • ID • Date • Score

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		• Merged PR • Failed Commits • Passed Commits • Low Centrality • Median Centrality • High Centrality Repo-Specific • Activity-Specific • Comment Count • Commit Count • Issue Count Issues • URL • Body • Date • Title • Is Merged • Repository • Open/Close Commits • Date • URL • Repository Comments • Date • Issue Number	• Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID • Post ID • Body • Score • Tags

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
What is the quality of a candidate's contributions?		• Repository • Parent URL Commit Quality: • Fail Commit • Pass Commit • Low Centrality • Median Centrality • High Centrality Issue Quality: • Issue Count • Close Issue • Non-Merged PR • Merged PR	Answer Quality: Answers • ID • Date • Accepted

D.2 Trust

Trust-related questions focus on information which could lead to an evaluator having positive confident expectations about the candidate's conduct and allow them to predict his/her future actions as a hire.The Trust Factors-Attributes Matrix from Appendix C has been used for mapping the GitHub and StackOverflow attributes to these questions.

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
What is the coping capacity of the	Frequent Updates of Project Progress	Issue Count Comment Count	Answer Count Comment Count

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
candidate in terms of the technical and mental capability of being able to handle different situations?		Commit Count Activity-Specific • Commit Count • Comment Count • Issue Count • Close Issue • Non-Merged PR • Merged PR • Fail Commit • Pass Commit • Low Centrality • Median Centrality • High Centrality Repo-Specific • Activity-Specific • Comment Count • Commit Count • Issue Count Issues • URL • Body • Date • Title • Is Merged • Repository • Open/Close	Activity-Specific • Answer Count • Commit Count Answers • ID • Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID • Post ID • Body • Score • Tags

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
	Reputation	Commits • Date • URL • Repository Comments • Date • Issue Number • Repository • Parent URL Followers Each Repository • Forks • Watchers • Contributors • Collaborators	Reputation Accepted Count Non-Accepted Count Activity-Specific • Accepted Count • Non-Accepted Count Tag-Specific • Activity-Specific • Accepted Count • Non-Accepted Count Question • Favorite • View Count Answers • Score • Favorite • Accepted

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
			Comments • Score
	Leadership	Followers Following	Reputation
How agreeable is a candidate with respect to qualities of cooperation, consideration and trust?		Name Login Commit Count Comment Count Issue Count Activity-Specific • Commit Count • Comment Count • Issue Count Repo-Specific • Description • Name • Website Issues • Date • Open/Close	Display Name Reputation Question Count Answer Count Comment Count Questions • Date • Body • View Count • Answer Count • Comment Count Answers • Date • Body • Comment Count Comments • Date • Body
How technology, people and management capabilities of	Language Leadership	Repo-Specific • Language Followers Following	Tags Reputation

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
candidates help gain trust?	Frequent Meetings Same location	NA Location and Time Zone information does not appear to be present although the name, ghpage and website could be used to look these up if publically available.	NA Location and Time Zone information does not appear to be present although the display name and website could be used to look these up if publically available.
	Frequent Updates of Project Progress	Issue Count Comment Count Commit Count Activity-Specific • Commit Count • Comment Count • Issue Count • Close Issue • Non-Merged PR • Merged PR Failed Commit • Passed Commits • Low Centrality • Median Centrality • High Centrality Repository-specific • Activity-Specific	Answer Count Comment Count Activity-Specific • Answer Count • Commit Count Answers • ID Date • Score • Title • Favorite • Tags • Comment Count • Accepted Comments • Date • ID

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		• Comment Count • Commit Count • Issue Count Issues • URL • Body • Date • Title • Is Merged • Repository • Open/Close Commits • Date • URL • Repository Comments • Date • Issue Number • Repository • Parent URL	• Post ID • Body • Score • Tags
How can the quality of deliverables help establish trust?		Commit Quality: • Fail Commit • Pass Commit • Low Centrality • Median Centrality • High Centrality	Answer Quality: Answers ID • Date • Accepted

Question	Trust Factors	Attribute (GitHub)	Attribute (Stack Overflow)
		Issue Quality: • Issue Count • Close Issue • Non-Merged PR • Merged PR	

Appendix E

Screenshots of the TecSo Lens

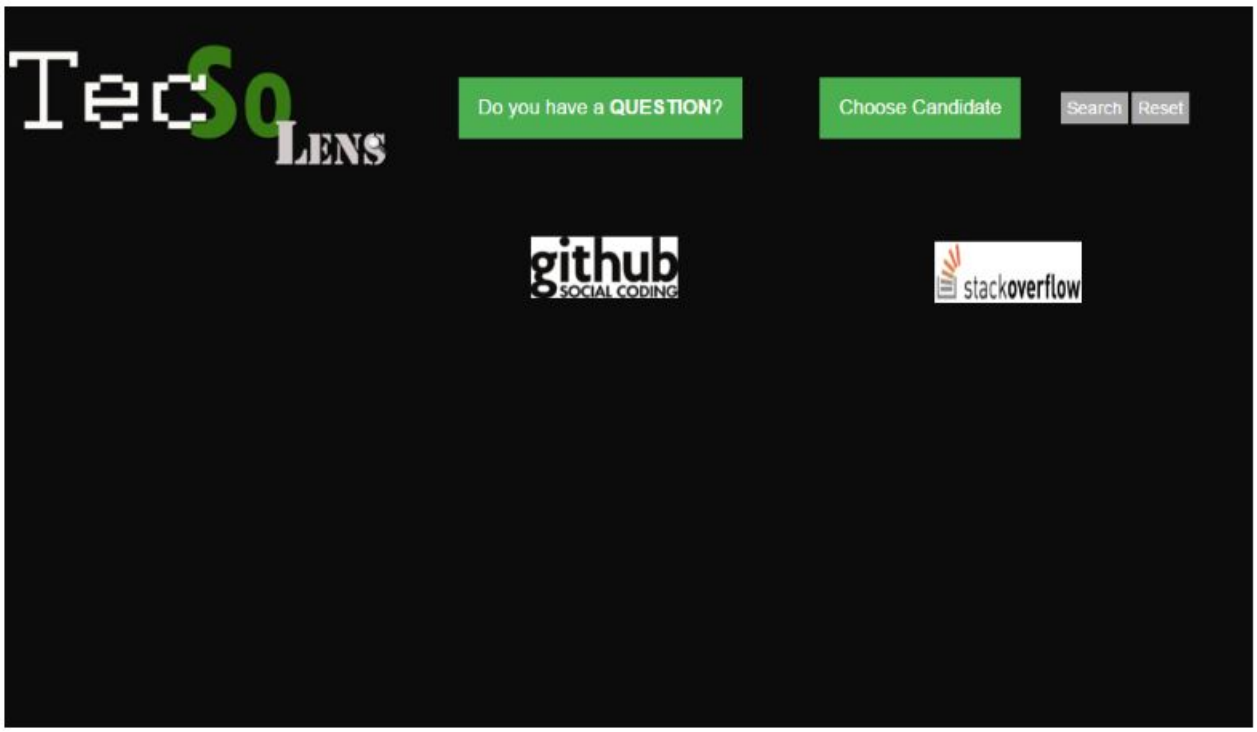


Figure E.1: TecSo Lens Home Screen

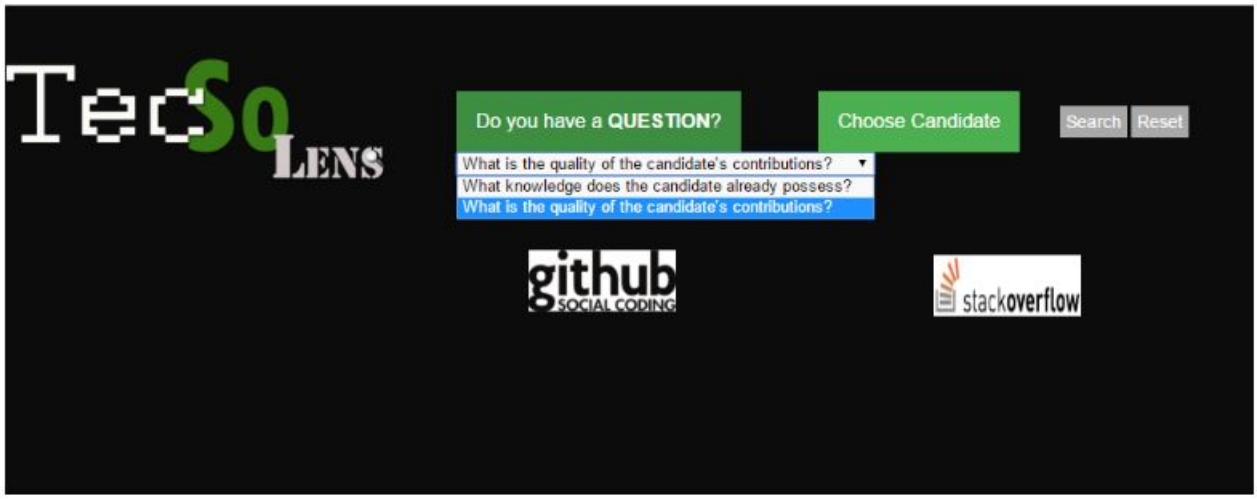


Figure E.2: The Questions that can be selected

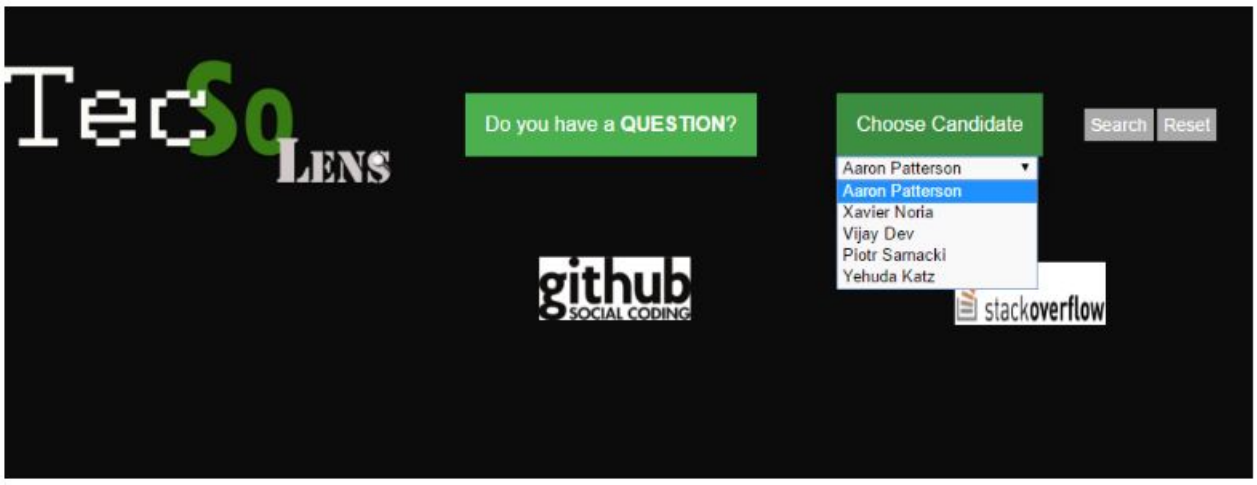


Figure E.3: The five candidates that can be selected

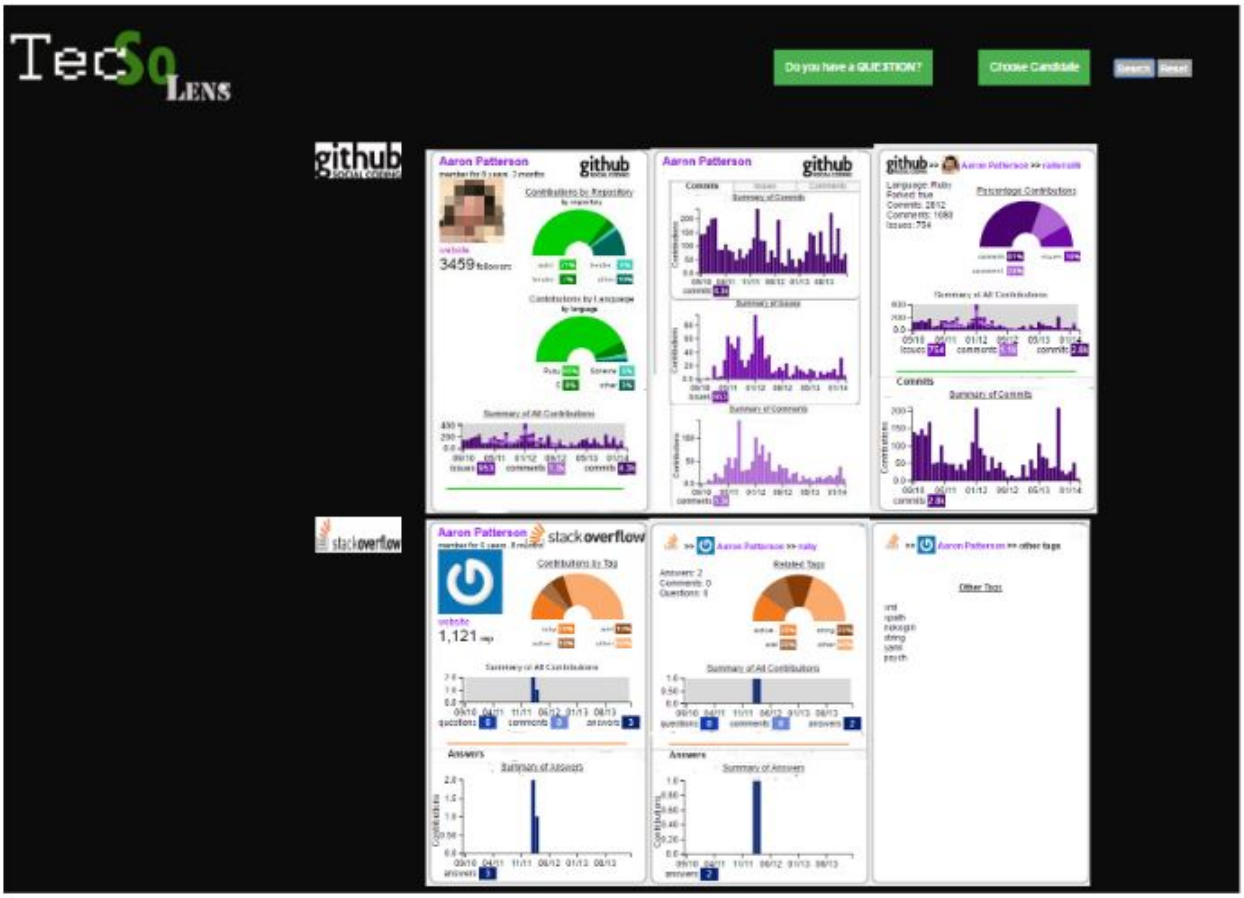


Figure E.4:
Selected question for search: What knowledge does the candidate already possess?
Selected candidate for search: Aaron Patterson

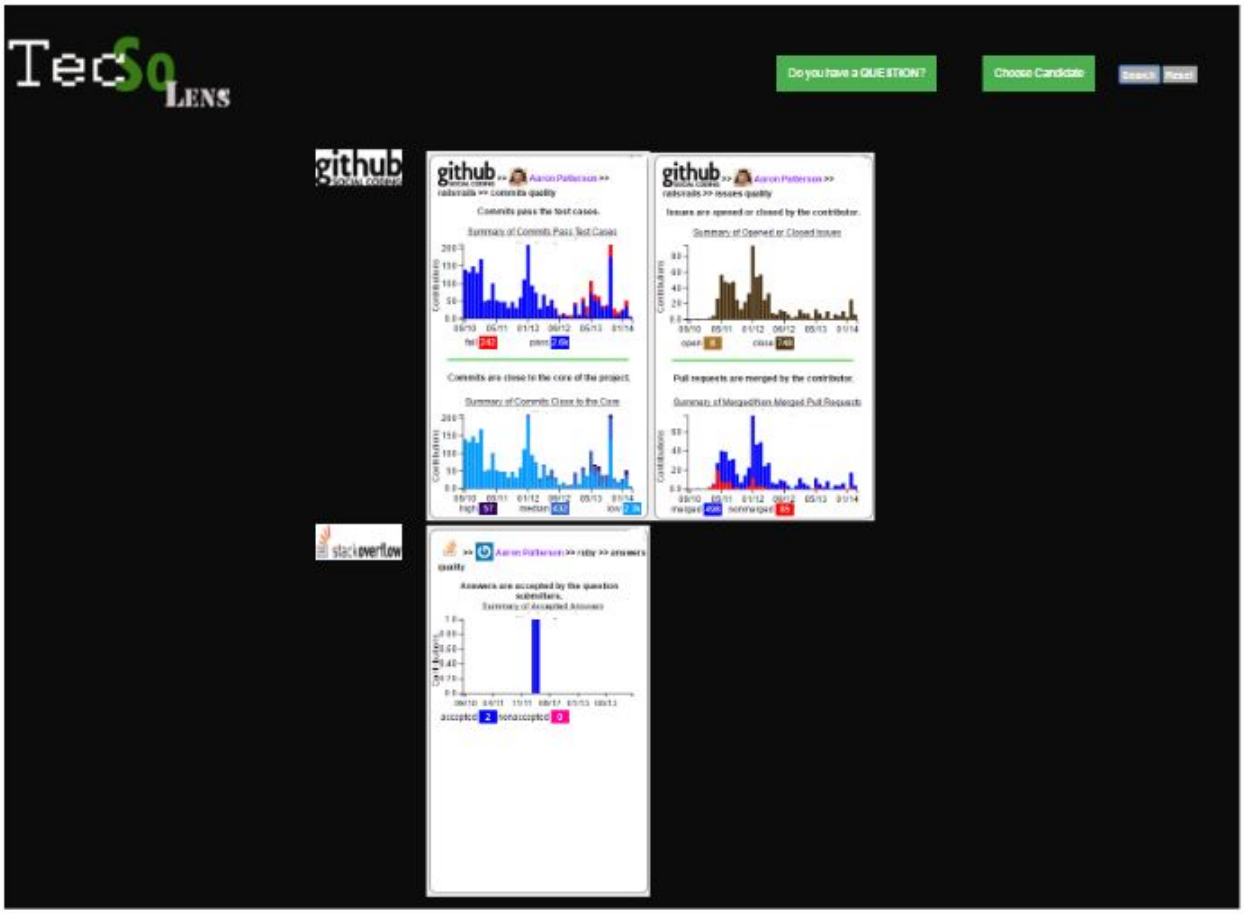
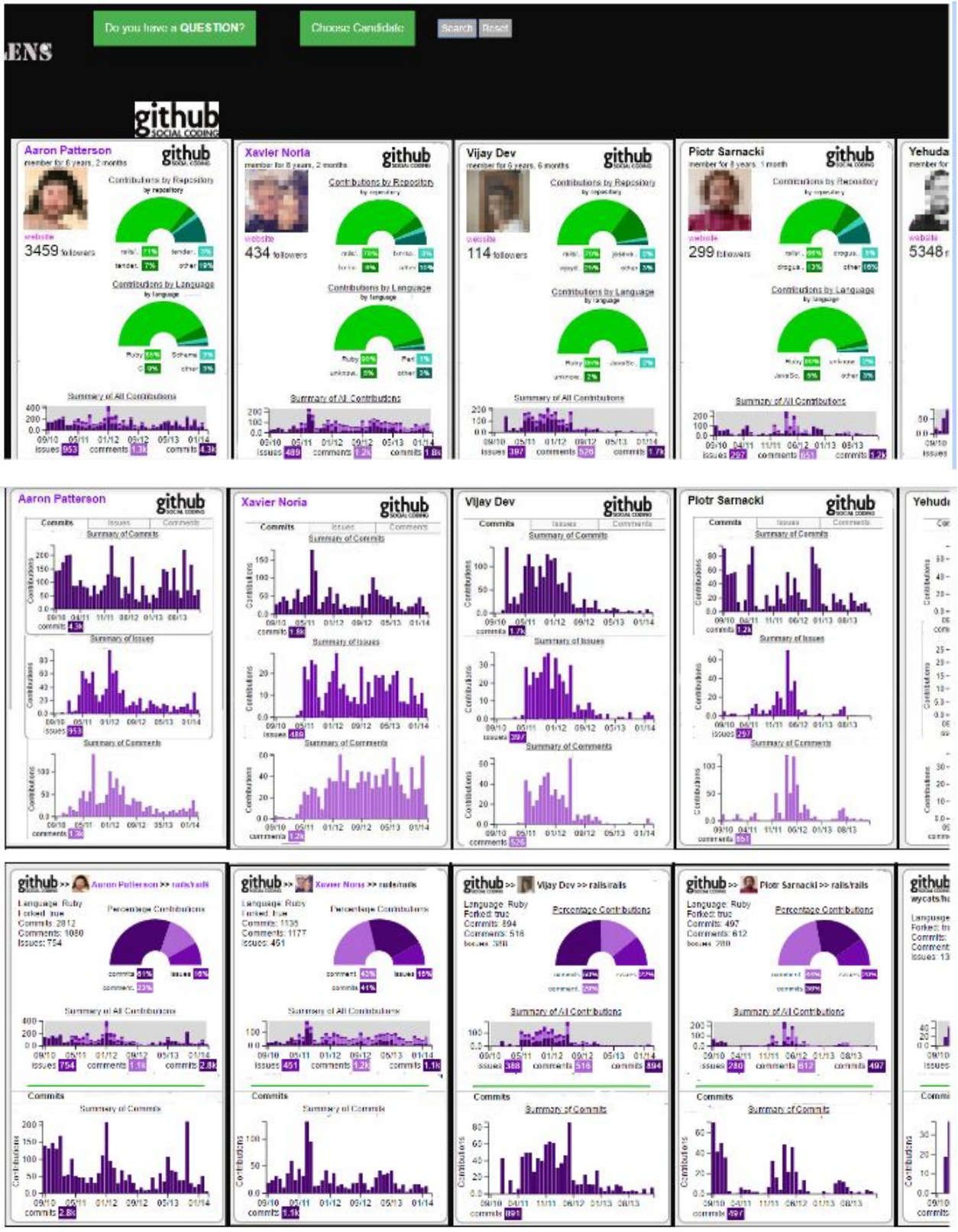


Figure E.5:
Selected question for search: What is the quality of the candidate's contributions?
Selected candidate for search: Aaron Patterson


Do you have a QUESTION?
Choose Candidate
Search
Reset
github
Aaron Patterson
Xavier Noria
Vijay Dev
Piotr Sarnacki
Yehuda
Contributions by Repository
Contributions by Language
Summary of All Contributions
Summary of Commits
Summary of Issues
Summary of Comments
Percentage Contributions
Language: Ruby
Forked: true
Commits: 2812
Comments: 1080
Issues: 754
Commits: 1135
Comments: 1177
Issues: 451
Commits: 894
Comments: 516
Issues: 388
Commits: 497
Comments: 612
Issues: 280

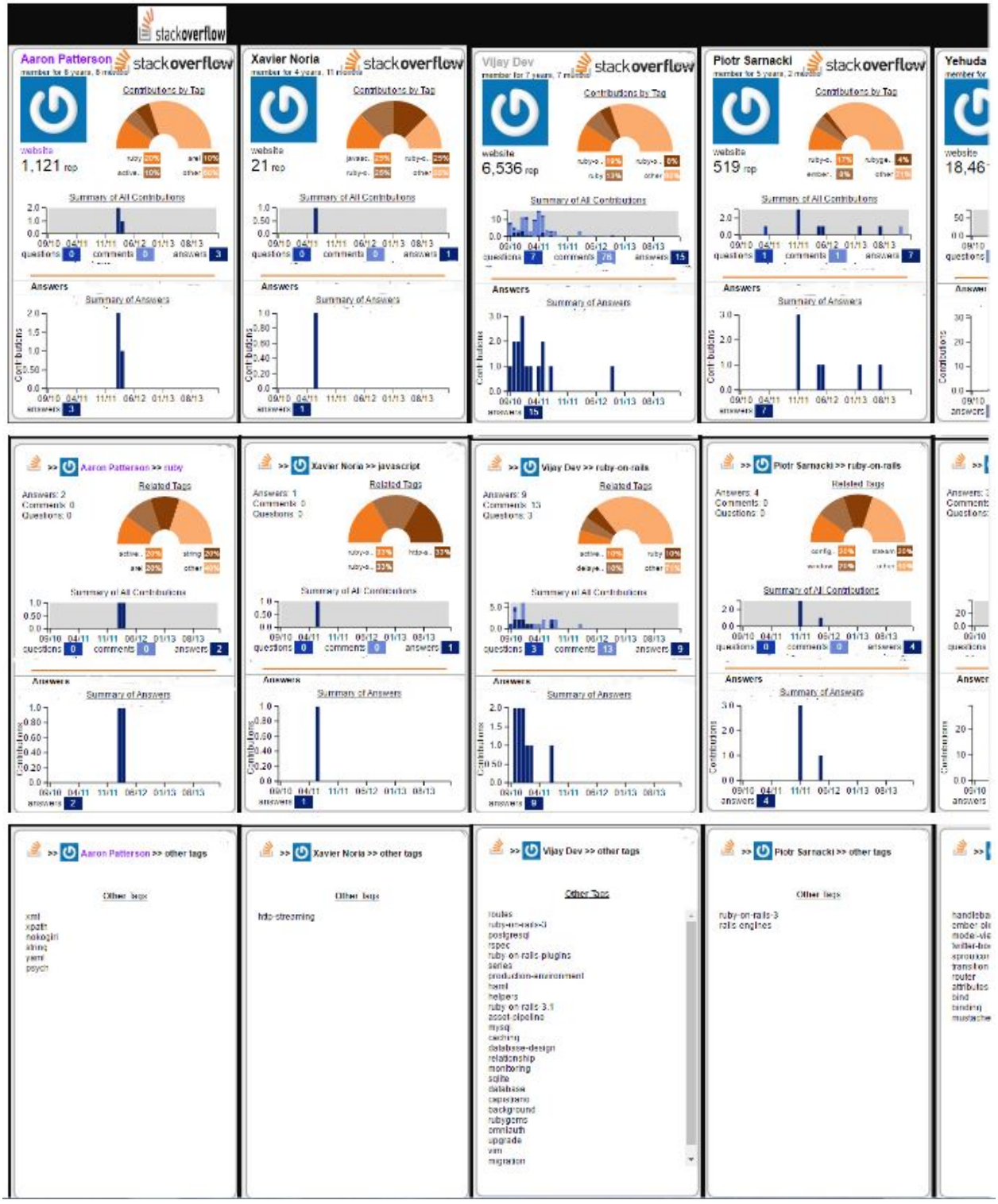


Figure E.6: Side-by-side comparison of all candidates for selected question: What knowledge does the candidate already possess?

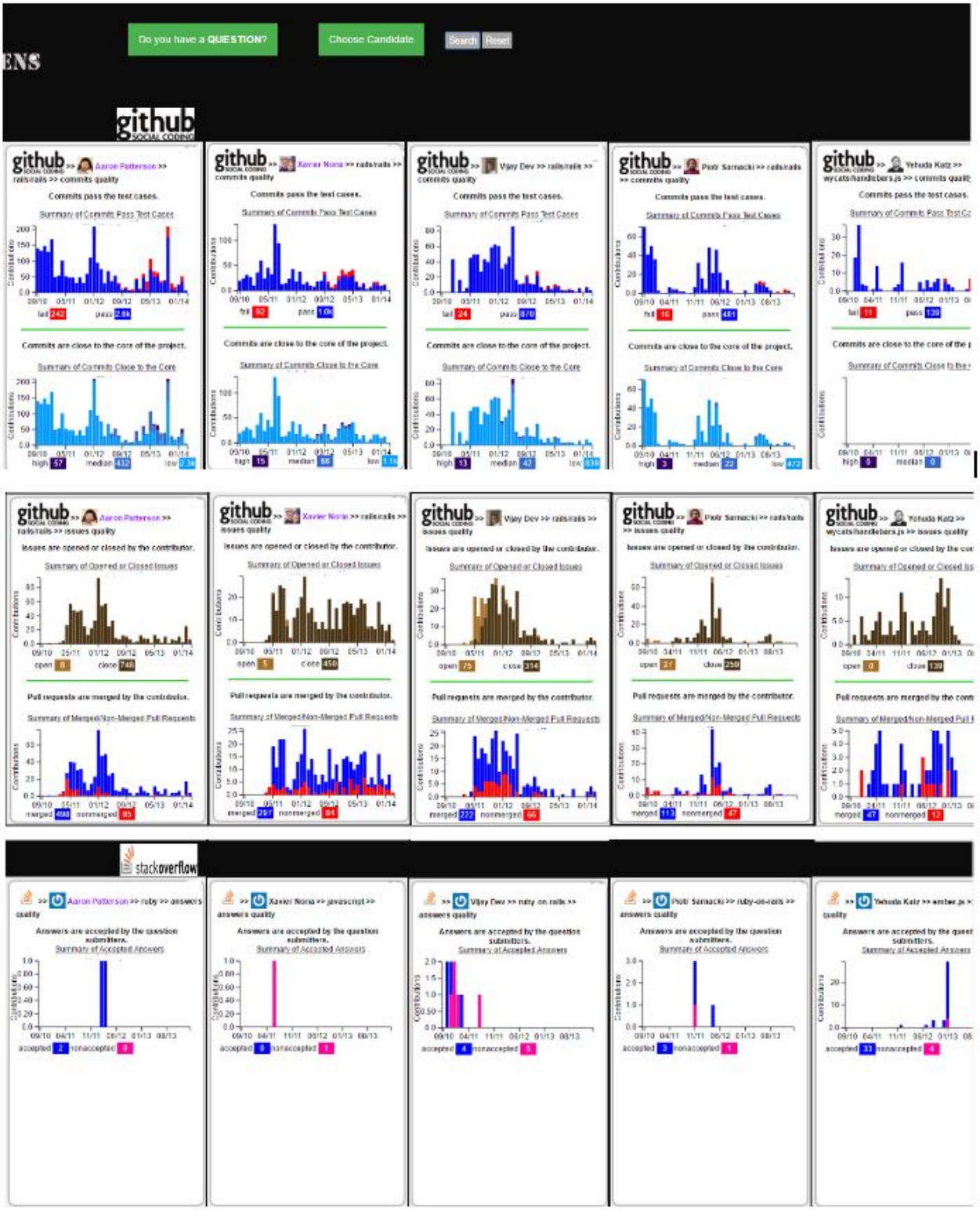


Figure E.7: Side-by-side comparison of all candidates for selected question: What is the quality of the candidate's contributions?

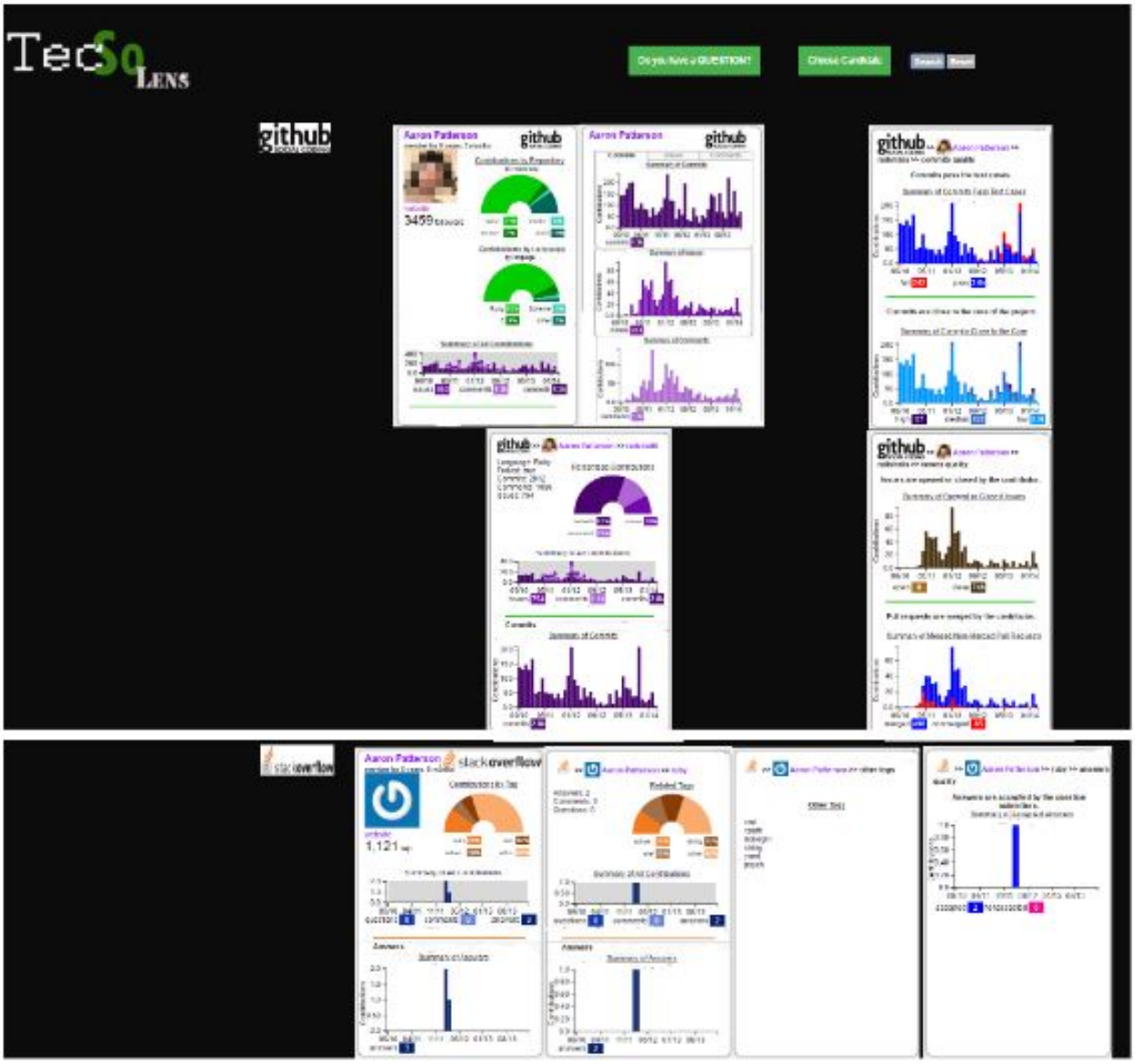


Figure E.8: Side-by-side comparison for the same candidate but with the answers for both questions

Appendix F

Pre-Session Questionnaire

Pre- Session Questionnaire

Name	
Current Occupation	
If Student, please specify: Degree and Department	
Age	

Education

What is the highest academic degree you have received?

☐ Less than high school diploma

☐ High School Diploma or Equivalent

☐ Undergraduate Degree (eg. BA, BS etc.)

☐ Graduate Degree (Masters or Doctorate)

How many class projects have you worked on that involved teams? (approximately) ___

How would you rate your experience working on such projects as compared to working on your own?

O Very Good O Good O Fair O Poor O Very Poor

Did you complete all the projects successfully?

O Yes O No O Unsure

Describe a person that you had a good experience working with:

Describe a person that you had a bad experience working with:

Work Experience

When was the last time that you applied for a job/internship?

O Past Week O Past Month O Past 3 Months O Past 6 months O 1 Year or more

What means have you used for applying to jobs/internships? (check all that apply)

☐ Company Websites ☐ Job Portals ☐ Career Fairs ☐ Email
☐ Referrals ☐ LinkedIn ☐ Walk-Ins ☐ Other. Please Specify_____

What media have you used to provide information while applying to jobs/internships? (check all that apply)

☐ Resumes ☐ LinkedIn ☐ GitHub ☐ Personal Website ☐ Other. Please Specify_____

Do you recall the last time that you had a conversation with a recruiter for a job?

O Yes O No

If yes, what mediated the conversation? (check all that apply)

☐ Career Fair ☐ Email ☐ Phone Call ☐ In- Person Interview
☐ Other. Please Specify _____

What kind of topics were discussed as a part of this conversation? (check all that apply)

☐ Introduction
☐ Resume
☐ Educational Experience
☐ Work Experience
☐ Behavioral questions
☐ Technical Questions
☐ Others Please Specify _____

Do you recall the last time that you had a conversation with a manager/ tech lead/ technical developer for a job?

O Yes O No

If yes, what mediated the conversation? (check all that apply)

☐ Career Fair ☐ Email ☐ Phone Call ☐ In- Person Interview
☐ Other. Please Specify _____

What kind of topics were discussed as a part of this conversation? (check all that apply)
☐ Introduction
☐ Resume
☐ Educational Experience
☐ Work Experience
☐ Behavioral questions
☐ Technical Questions
☐ Others Please Specify _____

How many years of work experience do you have?
O 0-1 O 2-3 O 4-6 O More than 6

How many years of work experience do you have working in software-related fields?
O 0-1 O 2-3 O 4-6 O More than 6

How many software development projects have you worked on that involved teams? (approximately) ____

How would you rate your experience working on such projects as compared to working on your own?

O Very Good O Good O Fair O Poor O Very Poor

Did you complete all the projects successfully?

O Yes O No O Unsure

Describe a person that you had a good experience working with:

Describe a person that you had a bad experience working with:

How familiar are you with GitHub?
(5-Very Familiar, 1-I don't know what is GitHub)
O 5 O 4 O 3 O 2 O 1

How familiar are you with Stack Overflow?
(5-Very Familiar, 1-I don't know what is Stack Overflow)
O 5 O 4 O 3 O 2 O 1

Selection

Have you had experience picking a new person or more specifically, a new developer for a project? O Yes O No

If yes, how many times have you had the opportunity to select a person? ___

Rate the following on a scale of 1-9 on how important you think they are:

(9-Necessary, 1- Optional)

Technical Skills	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Communication	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Team Cohesion	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Leadership Skills	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Reputation	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Location	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Work Experience	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1
Educational Degree	O 9	O 8	O7	O 6	O 5	O 4	O 3	O 2	O 1

Appendix G

Requirements as provided for the Recruiter Role

You are a recruiter for a product-based company XYZ.
You have been asked to recruit a software developer with the following qualifications to work on a new project that will start next month and is to be completed in the next six months :

BASIC QUALIFICATIONS

- Bachelor's degree in computer science, computer engineering or related technical discipline
- Experience with software development skills, preferably in Java, C/C++, Ruby and/or Perl in Linux environment.
- Knowledge of system and application performance characteristics.

PREFERRED QUALIFICATIONS

- An MS degree in Computer Science
- Knowledge of professional software engineering practices & best practices for the full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations
- Ability to work in a fast paced and agile development environment
- Motivation to drive tasks to completion and take ownership of projects
- Experience collaborating in globally distributed teams.
- Prior history of delivering high quality code
- Experience building and operating online services and fault-tolerant distributed systems

You have five candidates that have been short-listed through a preliminary screening process. You need to choose one among the five for the role.

For an unknown Candidate that you may have to work with, rate your agreement for the following questions:
Scale 1-9 (9-Strongly agree, 1-Strongly Disagree)

- I would be comfortable giving the person a task or a problem which was critical even if I could not monitor the person.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I would not have any reservations about working with the person even if we were to collaborate from different locations
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can rely on the person to respond to my messages in a reasonable amount of time.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can say for sure that the person will be agreeable to working in this project for the entire duration (6 months)
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1

Appendix H

Requirements as provided for the Manager Role

You are a manager for a product-based company XYZ.
You have been asked to hire a software developer with the following qualifications to work on a new project that you will be in charge of. The project will start next month and is to be completed in the next six months. You will have one week to onboard the new hire.

BASIC QUALIFICATIONS

- Bachelor's degree in computer science, computer engineering or related technical discipline
- Experience with software development skills, preferably in Java, C/C++, Ruby and/or Perl in Linux environment.
- Knowledge of system and application performance characteristics.

PREFERRED QUALIFICATIONS

- An MS degree in Computer Science
- Knowledge of professional software engineering practices & best practices for the full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations
- Ability to work in a fast paced and agile development environment
- Motivation to drive tasks to completion and take ownership of projects
- Experience collaborating in globally distributed teams.
- Prior history of delivering high quality code
- Experience building and operating online services and fault-tolerant distributed systems

You have five candidates that have been short-listed through a preliminary screening process. You need to choose one among the five for the role.

For an unknown Candidate that you may have to work with, rate your agreement for the following questions:
Scale 1-9 (9-Strongly agree, 1-Strongly Disagree)

- I would be comfortable giving the person a task or a problem which was critical even if I could not monitor the person.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I would not have any reservations about working with the person even if we were to collaborate from different locations
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can rely on the person to respond to my messages in a reasonable amount of time.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can say for sure that the person will be agreeable to working in this project for the entire duration (6 months)
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1

Appendix I

Post-Session Checklist

Check all the requirements that matched with your chosen candidate

BASIC QUALIFICATIONS

- Bachelor's degree in computer science, computer engineering or related technical discipline
  O Yes O No O Don't know

- Experience with software development skills, preferably in Java, C/C++, Ruby and/or Perl in Linux environment.
  O Yes O No O Don't know

- Knowledge of system and application performance characteristics.
  O Yes O No O Don't know

PREFERRED QUALIFICATIONS

- An MS degree in Computer Science
  O Yes O No O Don't know

- Knowledge of professional software engineering practices & best practices for the full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations
  O Yes O No O Don't know

- Ability to work in a fast paced and agile development environment
  O Yes O No O Don't know

- Motivation to drive tasks to completion and take ownership of projects
  O Yes O No O Don't know

- Experience collaborating in globally distributed teams.
  O Yes O No O Don't know

- Prior history of delivering high quality code
  O Yes O No O Don't know

- Experience building and operating online services and fault-tolerant distributed systems
  O Yes O No O Don't know

Appendix J

Post-Session Interview Questions

Post- Session Interview Questions

Selecting a new hire

Which candidate did you pick?

_____

Why did you choose this candidate?

On a scale of 1-9, how would you rate this candidate as a match with the requirements provided to you?

O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1

Rate whether you found the information provided to you satisfactory for making your choice
(*5*-Very useful, *1*-Not at all useful)

O 5 O 4 O 3 O 2 O 1

Do you think it would have helped if you had more information?

O Yes O No O Maybe

What more information do you think could have helped?

Apart from GitHub and StackOverflow, what other platforms do you think would have been useful in providing more information?

What other information do you think you would get from these platforms?

For the candidate that you just picked, rate your agreement for the following questions:
Scale 1-9 (9-Strongly agree, 1-Strongly Disagree)

- I would be comfortable giving this person a task or a problem which was critical even if I could not monitor the person.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I would not have any reservations about working with this person even if we were to collaborate from different locations
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can rely on this person to respond to my messages in a reasonable amount of time.
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1
- I can say for sure that this person will be agreeable to working in this project for the entire duration (6 months)
  O 9 O 8 O7 O 6 O 5 O 4 O 3 O 2 O 1

Application

How easy was it to use the tool?
(*5*-Very Easy, *1*-Very Hard)
O 5 O 4 O 3 O 2 O 1

How helpful was it for accessing information about each candidate?
(*5*-Extremely, *1*-Not At All)
O 5 O 4 O 3 O 2 O 1

How relevant do you think the information was to the question displayed?
(*5*-Extremely, *1*-Not At All)
O 5 O 4 O 3 O 2 O 1

Did a specific question help you make your decision? If so, which? If not, why?

What would you suggest for improving the interface?
