Assessing Technical Candidates on the Social Web

Andrea Capiluppi	Alexander Serebrenik Eindhoven	Leif Singer
Brunel University	University of Technology	Leibniz Universität Hannover,
United Kingdom	The Netherlands	Germany
andrea.capiluppi@brunel.ac.uk	a.serebrenik@tue.nl	leif.singer@inf.uni-hannover.de

Abstract

The *Social Web* provides comprehensive and publicly available information about software developers: they can be identified as contributors to open source projects, as experts at maintaining weak ties on social network sites, or as active participants to knowledge sharing sites. These *signals*, when aggregated and summarized, could be used to define individual profiles of potential candidates: job seekers, even if lacking a formal degree or changing their career path, could be qualitatively evaluated by potential employers through their online contributions.

At the same time, developers are aware of the Web's public nature and the possible uses of published information when they determine what to share with the world. Some might even try to manipulate public signals of technical qualifications, soft skills, and reputation in their favor. Assessing candidates on the Web for technical positions presents challenges to recruiters and traditional selection procedures; the most serious being the interpretation of the provided signals.

Through an in-depth discussion, we propose guidelines for software engineers and recruiters to help them interpret the value and trouble with the signals and metrics they use to assess a candidate's characteristics and skills.

Keywords: H.5.3.b Collaborative computing; H.5.3 Group and Organization Interfaces; H.5 Information Interfaces and Representation (HCI); I.2.13.h Knowledge retrieval; I.2.13 Knowledge Management

1 Introduction

In 2009, 48% of the Inc. 500 companies have used social media sites for recruitment and candidate evaluation [1]. In a survey of 115 small and medium size businesses, the online recruiter Jobvite [2] found that 78% of them used social media in their recruiting efforts. The top three sites were LinkedIn, Facebook, and Jobster [1].

The traditional recruitment process is commonly based on the following steps:

1. **Summary of qualifications**: The first stage of the recruitment process is gathering qualifications from candidates. The aim is to assess their qualifications against the benchmarks of the job description. This is done by evaluating the previous experiences that a candidate summarizes in their documentation. Such experiences may include degrees, certificates, courses taken and taught, and previous positions.

2. **In-situ evaluation**: A second stage (optional) is asking the candidate to sit in a workbench exercise. This helps in assessing whether the qualifications mentioned in the CV have actually been mastered by the candidate.

3. **Reference**: The next stage is usually a request of references from third parties in order to qualitatively assess the impressions that others received from the candidates.

4. **Interview**: In the final stage, the candidate is evaluated in person at a formal interview conducted by a panel of people working for the hiring organization.

This assessment procedure has several shortcomings when assessing candidates discovered through social media, especially for those who lack formal degrees, relevant job experience, or coming from non-IT backgrounds [2] (i.e., "non-traditional" candidates). We discuss how the Social Web enables recruiters and developers alike to assess software developers and which issues may arise when doing so. In particular, this paper defines a set of metrics and signals that enable recruiters and job seekers to evaluate both traditional and non-traditional candidates. Qualifications and work experience can be summarized by gathering the online activities of an individual in open, repeatable, i.e., even when repeating the same analysis, one should obtain the

---

same results, explicit, and unambiguous terms. Similarly, the process of reference gathering could be supported by monitoring the status, or "reputation" of an individual among her peers.

2 Social Media Sites

Social media have changed how developers collaborate, communicate, learn, and how they become informed about new technologies [3]. They use a vast array of social media sites targeted at software development professionals and amateurs. a superset of social network sites (as defined by Boyd and Ellison [4]), these social media sites support identity, content, and interaction transparency (as defined by Stuart et al. [5]). See Table I for a classification of social media sites based on their audience and purpose.[3]

Social media sites target a wide spectrum of diverse users, ranging from the general audience (e.g., Facebook), general professionals (e.g., LinkedIn), or IT experts (e.g., Stack Overflow). The purposes of these sites can also be very diverse. Some provide the ability to share content such as source code fragments (e.g., snipplr), entire projects (e.g., Github, bitbucket) or images (e.g., Flickr). Others support knowledge sharing through questions and answers (e.g., StackOverflow) or news postings (e.g., reddit).

Some support multiple activities, for example, LaunchPad[4] supports code sharing, question answering, and communication.

Profile aggregators create a single public user profile from an individual's profiles on many different social media sites. We additionally classify these aggregators as either *user-driven* or *autonomous*: user-driven profile aggregators require a user to provide information, such as the profiles to aggregate. Autonomous profile aggregators actively crawl the Web for any information they find about an individual without his or her participation or consent.

Masterbranch is an example for a user-driven profile aggregator targeted at software developers. Members can import their LinkedIn profiles or StackOverflow reputation, but Masterbranch will not gather information about non-members autonomously. In contrast, Yatedo crawls the Web to collect all publicly available information on any individuals it can find. Autonomous profile aggregators can be expected to contain information about a larger number of individuals than the user-driven ones. However, user-driven profile aggregators are not limited to publicly available information; users may provide them with access to private software repositories to crawl. Therefore, they can provide recruiters with all the information that developers explicitly choose to share.

Purpose // Audience	Software developers	Profes-sionals	General audience
Content sharing		GitHub, bitbucket, SourceForge, Google Code	
Knowledge sharing	General	Mailing Lists, Bug Trackers	Blogging, Wikipedia
	Q&A	Stack Overflow	Cross Validated	Yahoo! Answers
	News	Slashdot, hackernews		reddit, digg
Professional Networking		Careers by Stack Overflow	LinkedIn, Xing
Profile aggregators		MasterBranch, Coderwall, Geekli.st	Zerply
Blogging		Numerous individual blogs spread over multiple sites	

Table 1: Examples and classification of social network sites

---

[3]   URLs of sites mentioned: http://github.com, http://bitbucket.org, http://pinterest.com, http://foursquare.com, http://flickr.com, http://sourceforge.net, http://code.google.com, http://stats.stackexchange.com, http://wikipedia.org, http://stackoverflow.com, http://answers.yahoo.com, http://slashdot.org, http://news.ycombinator.com, http://reddit.com, http://digg.com, http://careers.stackoverflow.com/, http://linkedin.com, http://xing.com, http://twitter.com, http://facebook.com, http://plus.google.com, http://masterbranch.com, http://coderwall.com, http://geekli.st, http://zerply.com, http://yatedo.com, http://wordpress.org, http://livejournal.com

[4]   http://launchpad.net

3 Social Media Activity and Content as Signals for Qualifications

The open source development model, while more technical and requiring specific expertise, has raised the general profile of online user-generated content. At first, it supported a massive collaboration environment, but lately, it has been used as a way to showcase one's skills to peers and recruiters.

Dabbish et al. [6] examined the behavior of software developers in public projects on GitHub. They find that developers consciously manage their online reputations and are aware that other developers judge them by their publicly visible behavior. Developers use those signals to assess personal characteristics such as commitment, but also work quality. Some signals are the type and quality of commits, the number of forks in a project, and the contribution policy of project owners.

Singer et al. [7] investigated the use of developer aggregators by software developers and recruiters. Developers used simplified signals such as badges and achievements to get an overview of another developer's area of expertise, and then used more complicated signals, such as their posts on Twitter or their behavior in online discussions, to derive hypotheses about their technical qualifications and soft skills. As one interviewee in the study said about assessing passion in a developer, "*A 9 to 5 developer is not tweeting about the latest stuff that's coming out of the W3C mailing list. A 9 to 5 developer is tweeting a picture of the hamburgers he's frying at 4:30.*" He used the subjects of a developer's tweets—in this case, technologies and standards vs. leisure activities—as a signal for (lack of) passion about software development. Less passionate developers, labeled as "9 to 5 developers" by the interviewee, would tweet more about leisure activities.

4 Social Media Reputation as a Signal for References

The use of references in job recruitment is widespread, as it can provide insights into the personality traits of candidates [8]. Moreover, references originate from colleagues and co-workers who had the opportunity to observe the candidate's *typical* performance, which is a better predictor for how well individuals will perform on average [9].

Social media sites provide various means for members to express appreciation of their peers' achievements. Some sites support highlighting activities within the site itself, such as StackOverflow's badges and reputation metric. Others allow for the appreciation of activities external to the site. For example, recommendations on LinkedIn are meant to rate past professional collaborations. In both cases, highlighting an achievement contributes to the information about an individual. However, the credibility of such a highlight depends on its meaning among the peers, for example whether it is seen as an actual status symbol (cf. Antin and Churchill [10]).

Our discussion is based on several studies that have examined the use of reputation signals on the Social Web. Stein and Hess [11] found that reputation signals on Wikipedia influence how favorably an editor's articles are being voted for by the community. Singer et al. [7] found that developers use signals as diverse as a developer's activity on Twitter and GitHub, his or her followers on Twitter and Github, and the achievements he or she displayed by profile aggregators, to determine the reputation of one another. They also actively endorse others in public as an appreciation for good work, e.g., by giving "*free beer*" on Masterbranch or by clicking the endorse button on Coderwall.

5 Social Media and Their Signals

We now explore the various mechanisms available to site members and the implications of being a contributor to various social media sites. The types of sites are ordered by how distinct and reliable their signals can be. Each provides a very distinctive set of data that could be used to assess a candidate. We also show a number of drawbacks that recruiters and candidates should take into consideration due to the design and implementation of the sites. Table II provides an overview of sites with regard to interpreting them for qualifications; Table IIIsummarizes our discussion with regard to reference gathering and reputation.

5.1 Generic Social Network Sites

While generic social network sites are mostly used for personal networking, the available signals for recruiters can be used to infer how the individual might integrate and work in a professional setting.

Qualification signals: Generic social network sites such as Facebook target large, non-specific audiences. Thus, it is more difficult to crawl them for relevant information about an individual's expertise. However, generic social network sites do help users summarize their past and present personal (though not often professional) affiliations.

References signals: Sites such as Facebook can provide a visible indicator of the type of social network that individuals are able to maintain. They also show how peers acknowledge their activities. In general, participation in such sites has been shown to be beneficial because it provides access to non-redundant information (e.g., employment connections). From the candidate's perspective, participation to these sites has been found to significantly increase his or her social skills. This proves beneficial specifically for low-esteem candidates [12].

Recommendations: Using Facebook (or similar sites) to seek information about potential candidates may be problematic for both candidates and employers. Candidates might have published professionally unattractive, private content, making them less interesting for employers. Companies taking this seriously may judge individuals on materials irrelevant to their profession, rejecting otherwise perfectly capable candidates. So, even though there is some value in using generic social network sites for assessing developers, we recommend that they not be used for recruiting.

5.2 Social Code Sharing

Social code sharing sites combine software repositories with networks of developers. Hence, the available signals are related to both the software projects and to the interactions between developers.

Qualification signals: The contributors and core developers of open source projects tend to archive and log all their development process in publicly accessible repositories. Social code sharing sites, such as GitHub, and generic code repositories, such as Google Code, present an opportunity for communicating the skills and experiences of candidates within distributed development teams. Operationalizing an interpretation of these experiences is a matter of summarizing them in a concise, comprehensible, yet still correct way. Metrics and signals about online activity can become open, verifiable, and reproducible, as we addressed in previous work adapting the h-index to developers [13]). For example, an h-index of 10 would point out a developer who worked in at least 10 projects with at least 10 other developers each.

References signals: Sites allowing their members to host source code typically limit the ways in which reputation can be expressed. They mostly focus on the relation between individuals and their code versus the relation between members. SourceForge and Google Code are good examples for this model. On the other hand, GitHub allows its users to *follow* one another. To a degree, this reflects professional reputation, as users who are doing work that is interesting to their peers will tend to have more followers.

Recommendations: Social code sharing sites are only indirectly able to provide information on how individuals achieve their reputation. To assess actual reputation, one would need to compute it from all of the individual's peers. We feel that recruiters who are not familiar with the software development culture should restrict their use of social code sharing sites mainly to assessing an individual's qualifications. Since the content on those sites (source code, tickets, commits, etc.) is fairly technical, we suggest that a recruiters should seek the assistance of software developers to interpret these signals.

5.3 Question & Answer Sites

Q&A sites such as Stack Overflow and Yahoo! Answers allow participants to earn badges and rewards, which are public metrics for what their members have earned through their activities. These metrics reflect the quality and the types of interactions among individuals. For example, contributors on Stack Overflow may earn the Great Answer badge for answers of high quality, while Yahoo! Answers users earn the Best Answer indication for good answers.

Qualification signals: The sites clearly define the areas to which a member has contributed. Metrics such as the number of answers given to a specific topic, the frequency of participation, and the quality of answers as rated by peers make it easier to distinguish a real expert from an occasional contributor.

References signals: Awarding badges and rewards is a very visible approach to assigning reputation to individuals. This practice categorizes people, their skills, and their levels of expertise, as well as their peers' perceptions of these aspects. Most statuses and badges are automatically awarded according to a transparent set of rules and can be easily scrutinized.

Recommendations: In general, badges and reputation of users should be used carefully by recruiters because users tend to up-vote one another's questions, answers, and comments (i.e., "leniency bias"). On the other hand, as has been reported before, the distribution of experts and novices is highly skewed; a few quick, knowledgeable individuals earn most of the good ratings, whereas most earn far fewer [14].

Q&A sites present a real advantage when monitoring the status and the expertise of contributors. Scattered evidence suggests that high reputation is immediately visible to recruiters, as questioned and partially answered in Stack Overflow itself.[5] Some commenters believe reputation to affect job recruiters,[6] while others clearly do not.[7]

5.4 Professional Social Networking Sites

Sites that are aimed at professionals, such as LinkedIn, allow individuals to track and publish their career paths, maintain lists of their skills and past experiences, the size and tenure of the teams with whom they have worked, and the roles they played on each team.

Qualification signals: Although knowledge and experience should be assessed by simply reading a set of skills and expertise, the information on Professional Networking Sites represents valuable information for employers. While traditional CVs may only mention the most recent positions, the site can list an individual's entire work history. In addition, recruiters can scrutinize the identity, skills, or trustworthiness of those giving recommendations by contacting them directly. Differently from common references, recruiters can even evaluate the background of the referees and the working connections with the candidate.

Reputation signals: The individual profiles of non-traditional candidates are not easily visible to searches, even though this data would be valuable for recruiters and employers. LinkedIn recommendations are similar to traditional references, stating the relation between the recommender and the applicant and the applicant's position at the time they worked together. Recruiters and employers should find these to be equivalent to direct references, since they do not require additional interpretation steps.

Recommendations: Professional Networking Sites represent an excellent source of value for both qualifications and reputation for recruiters to use. However, candidates with little or no official experience with employers or accredited educational institutions would not be able to credibly list skills or qualifications, which limits their visibility [15.

Recruiters and candidates should be aware of the "leniency bias" of online recommendations: very strong comments against a candidate are unlikely because they are publicly visible, including to the candidate herself. Some software developers have become aware and weary of such courtesy recommendations on LinkedIn [7].

5.5 Profile Aggregators

Profile aggregators create condensed profiles of individuals by collecting their data from other sites. Masterbranch and Coderwall are examples of aggregators for developers. A more general aggregator, Yatedo, automatically creates a profile for any person it can find on the Web. Even though Masterbranch supports adding private source code to a member's profile, developer profile aggregators support mostly open source projects. This might discourage developers predominantly active in closed source projects from joining, potentially creating a significant bias.

Qualification signals: Profile aggregators summarize information from various sites. The most popular developers will have the most visible profiles, interfering with the effectiveness of the profiles of less vocal or active developers. Nonetheless, all developers have the potential to increase their visibility. Aggregated profiles provide insight into their development efforts distributed across multiple open source repositories and relates the knowledge they have to their Q&A profiles or their professional experience.

---

5   Will high reputation in Stack Overflow help to get a good job?
     `http://programmers.stackexchange.com/questions/20407/will-high-reputation-in-sta`
      `ck-overflow-help-to-get-a-good-job`

6   As one participant said: "Look at three or four of the highly voted answers they wrote. If you've ever hired a programmer in your life, it's obvious those people are all some of the best programmers you could ever hire."

7   Another participant said: "It's not a case of direct causation. That is, having a high reputation does not lead to a high-paying job. Being intelligent, articulate, passionate and knowledgeable lead to a high-paying job"

Reputation signals: Many aggregators support their own profile-building mechanisms. For instance, Coderwall awards achievement badges to developers based on the code they commit to open source repositories. In addition, developers can explicitly *endorse* a peer. However, even among developers themselves, there is disagreement and uncertainty with regard to the value of such endorsements. Under some conditions, developers feel flattered by being endorsed. Others discard them as meaningless, as there is an explicit prompt to endorse others. One interviewee summarized his opinion, "I think there's some value to this kind of thing [Coderwall], but I'm not sure where it is yet." [7].

Recommendations: The ability to collect and relate data about candidates makes profile aggregators useful sources for information about a developer's qualifications, especially when used as an entry point for deeper investigations. Similarly to social code sharing sites, non-technical recruiters may find it difficult to comprehend some of the elements used on aggregated profiles and might want to seek the assistance of a technical person.

Type	Advantages	Disadvantages	Metrics
Generic Social Network Sites	summary of past affiliations and experiences	social and professional profiles blended	list of past experiences, interests and passions in posted content
Social Code Sharing	open, verifiable metrics and signals	risk of free riding, wrong attribution, niche market, technical expertise needed, open source bias	number of popular (cfr. [13]) repositories authored/contr

Type	Advantages	Disadvantages	Metrics
Q&A sites	easy to distinguish the real experts	high skewness between experts and novices, minorities under-represented	ibuted number of high quality answers in specific topics; list of topics contributed to
Professional Social Networking	experience in the area with dates, duration, position, and team sizes	duplicated information, non-verifiability of mentioned skills, representativeness of non-traditional candidates	list of past jobs, skills, experience in the area
(Developer) Profile Aggregators	signals based on actual code written	possibly oversimplification, technical expertise may be required for interpretation	combination of the metrics above

Table 2: Social Networks as a source of qualifications: summary of advantages, disadvantages and available metrics

Type	Advantages	Disadvantages	Metrics

Generic Social Network Sites	general psychological well-being; reciprocity and volunteerism; peer recognition	non-reproducibility; partially related to work performance	network size; number of followers
Social Code Sharing	indirectly measured by users downloading and using projects	not clear how "peers" are defined; technical proficiency needed for interpretation	number of downloads of code, size of developer's community
Q&A sites	objectivity; automatically awarded	leniency; long-tailed distributions; non-representativeness	Number of badges associated with self-interest; number of badges associated with

			prosociality.
Professional Social Networking	closest to traditional references	origins of recommendations; leniency; visibility of recommendations	Number and nature of recommendations, size of network
(Developer) Profile Aggregators	professional peer recognition	open source bias	endorsements received & given, diversity in technologies, activity level, achievements earned through actual code

Table 3: Social Networks as a source of references: summary of advantages, disadvantages and available metrics

6 Summary of Recommendations

The activity and experience documented in social media is easily verifiable, skills can be more readily assessed, and qualifications can in some cases be gathered with more confidence than asking a candidate to provide his or her set of references and the list of qualifications.

6.1 Recruiters

We strongly recommend non-technical recruiters to become familiar with social media, especially with the sites used by developers. Those that we discussed above should provide a good place to start. When they think critically about the positive and negative signals found in social media, recruiters will be able to more easily assess and access developers. Moreover, this initiative should improve the image of recruiters among developers because a lack of familiarity with the developer communities has been shown in at least one case to create a negative perception of recruiters among developers [7].

While we believe that profile aggregators are especially useful for recruiters, they are not yet comprehensive. Consequently, they should be used as an entry point into a developer's profile, with more thorough investigations happening at the social media sites that provided the data.

6.2 Candidates

The main issue for candidates without official degrees or experience in their field of work is to be recognized by recruiters. Social media have much to offer in this regard. Good work visible to public has and will be considered by at least some recruiters [1]. Developers with official qualifications will still profit from participating in social media in general and public software development. In particular, this provides recruiters with insights about soft skills [7].

Developers should publish their work, even personal projects that were built for learning. This will document their progress as developers and demonstrate that they are able to learn new technologies. In the process, developers may learn best practices and try out diverse technologies and approaches to problems. At the same time, developers need to be aware that published content will likely be forever public. This is not to say that developers should remove any private topic from their online presence. On the contrary, recruiters appreciate authenticity in developers, just as developer communities prefer authentic recruiters [7].

7 Conclusions and Outlook

During traditional selection procedures, technical proficiency is usually assessed through qualifications, and at times, is augmented by in-situ evaluations. Social skills are charted based on the references provided by the candidate, and validated by means of an interview. In this paper, we advocate the analysis of social media activity as an alternative for assessing qualifications and checking references. We show that a diverse set of signals accessible from social media sites provide insights into the technical and social skills of a candidate. The most promising set of signals are provided by profile aggregators; these combine and triangulate the status of an individual from various social media sites. One should be wary, however, of the inherent flaws associated with the use of social media. Reputation levels could be distorted by long-tail effects and skills could be misrepresented. For instance when candidates overestimate their activities in large, well-known projects. In addition, the system of badges and rewards could exclude a relevant part of interesting candidates. Developers and recruiters alike should keep these issues in mind when assessing peers and candidates on the Web.

References

[1] Thomas J. Hayes, Doug Ruschman, and Mary M. Walker. Social networking as an admission tool: A case study in success. Journal of Marketing for Higher Education, 19(2):109–124, 2009.

[2] Damien Joseph, Wai Fong Boh, Soon Ang, and Sandra Slaughter. The career paths less (or more) traveled: A sequence analysis of IT career histories, mobility patterns, and career success. MIS Quarterly, 36(2):427–452, 2012.

[3] Margaret-Anne Storey, Christoph Treude, Arie van Deursen, and Li-Te Cheng. The impact of social media on software engineering practices and tools. In Future of software engineering research, pages 359–364, New YorkUSA, 2010. ACM.

[4] Danah M. Boyd and Nicole Ellison. Social network sites: Definition, history, and scholarship. J. Computer-Mediated Communication, 13(1):210–230, 2007.

[5] H. Colleen Stuart, Laura Dabbish, Sara Kiesler, Peter Kinnaird, and Ruogu Kang. Social transparency in networked information exchange: a theoretical framework. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12, pages 451–460, New York, NY, USA, 2012. ACM.

[6] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in github: transparency and collaboration in an open software repository. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, pages 1277–1286. ACM, 2012.

[7] Leif Singer, Fernando Figueira Filho, Brendan Cleary, Christoph Treude, Margaret-Anne Storey, and Kurt Schneider. Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators. Technical Report DCS-347-IR, University of Victoria, Canada, June 2012.

[8] Toni Schmader, Jessica Whitehead, and Vicki H. Wysocki. A linguistic comparison of letters of recommendation for male and female chemistry and biochemistry job applicants. Sex Roles, 57(7-8):509–514, 2007.

[9] Julie M. McCarthy and Richard D. Goffin. Improving the validity of letters of recommendation: An investigation of three standardized reference forms. Military Psychology, 13(4):199–222, 2001.

[10] Judd Antin and Elizabeth Churchill. Badges in social media: A social psychological perspective. In Human Factors in computing systems. ACM, 2011.

[11] Klaus Stein and Claudia Hess. Does it matter who contributes: a study on featured articles in the German Wikipedia. In Proceedings of the eighteenth conference on Hypertext and hypermedia, HT '07, pages 171–174, New York, NY, USA, 2007. ACM.

[12] C. Steinfield, N. Ellison, and C. Lampe. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. Journal of Applied Developmental Psychology, 29(6):434–445, November 2008.

[13] Andrea Capiluppi, Alexander Serebrenik, and Ahmmad Youssef. Developing an h-index for OSS developers. In Mining Software Repositories, pages 251–254, 2012.

[14] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design lessons from the fastest q&a site in the west. In Human factors in computing systems, pages 2857–2866. ACM, 2011.

[15] Susanne C. Doty-Navarro and Brian H. Kleiner. How to effectively check references and perform background investigations of job applicants. Management Research News, 23(7-8):56–62, 2000.

Andrea Capiluppi is a Lecturer at Brunel University (UK) in the School of Information Systems, Computing and Mathematics. He has been a Senior Lecturer in Software Engineering at University of East London for three years, and a Senior Lecturer at the University of Lincoln, UK, for three years. He gained a PhD from Politecnico di Torino, Italy, in May 2005, and has held a Researcher position and a Consultant position at the Open University in UK. His research interests are Software Maintenance and Evolution, Software Architectures, Component Reuse and Open Source Software. Contact him at andrea.capiluppi@brunel.ac.uk *or on Twitter at @acapiluppi.*
CONTACT: Andrea Capiluppi, Information Systems and Computing, St John's 045, Brunel University Uxbridge UB8 3PH, United Kingdom

Dr. Alexander Serebrenik is an assistant professor of Model-Driven Software Engineering at Eindhoven University of Technology, the Netherlands. He has obtained his Ph.D. in Computer Science from Katholieke Universiteit Leuven, Belgium (2003) and M.Sc. in Computer Science from the Hebrew University, Jerusalem, Israel. Dr. Serebrenik's research interests include software evolution and maintenance, social media, program analysis and transformation. He is member of IEEE and ERCIM Working Group Software Evolution. Contact him at a.serebrenik@tue.nl *or on Twitter at @aserebrenik.*
CONTACT: Alexander Serebrenik, MF 7.143, Math & CS, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Leif Singer is a PhD student in software engineering at Leibniz Universität Hannover, Germany. His main research interest is the systematic use of social software to support software engineers -- e.g. using newsfeeds, game mechanics, or reputation systems. Singer has an MSc in computer science from Leibniz Universität Hannover. Contact him at leif.singer@inf.uni-hannover.de *or on Twitter at @lsinger.*
CONTACT: Leif Singer, Leibniz Universität Hannover, Welfengarten 1, 30167 Hannover, Germany