Gerosa et al. Journal of Internet Services and Applications (2015) 6:23
DOI 10.1186/s13174-015-0038-0

Journal of Internet Services and Applications
a SpringerOpen Journal

EDITORIAL Open Access

Editorial: Thematic series on software engineering from a social network perspective

Marco Aurelio Gerosa[1*], David Redmiles[2], Pernille Bjørn[3] and Anita Sarma[4]

Abstract

In the Internet age, the concept of social network is emerging as both a useful means for understanding software development activities and a concept for designing software tools to support these activities. Employing Social Networking is useful in part, because it brings a focus to stakeholders in software development including developers, their managers, their support staff, QA analysts, requirement engineers, etc. Social networks represent how people communicate, coordinate, cooperate, and develop working relationships. The concept is also useful as it reflects the network-centric organization for dealing with highly inter-dependent artifacts–including complex source code, systems and subsystems, requirement and specification documents, etc. However, the knowledge on social network in software engineering is spread across the literature and the term social network is used for multiple purposes, such as social networking sites, social capital, interpersonal connections, and social structures. This thematic series puts together papers that employ the concept of social networks either in studying software development from an empirical approach or use the concept as a central basis for developing software tool support. Readers will find in this issue a centralized resource for research in this area.

1 Introduction

This thematic series of the Journal of Internet Services and Applications (JISA) presents a collection of articles around the broad topic of Software Engineering from a Social Network Perspective. Articles on the concept of "social network" are spread out in the software engineering literature. This thematic series brings together different views and applications of the concept. This series comprises papers dealing with social network sites, specialized methods for software engineering activities, social network analysis and visualization, empirical quantitative and qualitative studies, patterns and anti-patterns, tool development, etc.

Software engineering is an inherently sociotechnical endeavor [1–3]. Many studies demonstrate the importance of social interaction in this context [4]. For instance, in a global software development scenario, it was found that the effort involved in collaboration readiness was an essential challenge, and the effectiveness of social interactions among developers impact the work [5]. Social interactions include aspects of commitment, trust, and awareness, and often these aspects are difficult to handle in software engineering, mainly in situations in which developers are geographically and culturally dispersed [6–10].

Communication and coordination is critical in software development, since it is a complex, creative effort, in which tasks are often divided among individuals or groups of individuals, such as developers, managers, product owners, users, systems operators, and other stakeholders. These tasks cannot be carried out independently and the task interdependencies drive the need for communication and coordination [11].

Communication plays a key role in the quality and evolution of a software product [12]. The interactions among the individuals configure implicit social networks. Making these network connections explicit help researchers and tool designers to better understand and support development activities. While previous research clearly has unpacked many of these challenges related to

* Correspondence: gerosa@ime.usp.br
[1]University of São Paulo, São Paulo, Brazil
Full list of author information is available at the end of the article

the social aspects of software engineering, in this special issue, we focus on a social network perspective.

The term "social network" is often used with different meanings. It sometimes describes social media and social networking sites (or platforms), such as the popular Facebook and LinkedIn sites. The term is also used to describe the people who an individual knows. In a narrower and more precise sense, "social network" refers to the graph structure made up of social actors (such as individuals or organizations) and connections between these actors. Thus, a social network reflects the relational information about the actors it comprises [13]. The understanding of complex social networks is an emerging and challenging research field [14, 15]. Social network analysis (SNA) is part of this field, and many constructs that help understand social networks have been proposed so far, such as centrality, density, structural holes, etc. These constructs support the understanding of the individual roles and interaction patterns and anti-patterns in a social network.

From the software engineering perspective, social networks can be extracted from the archives of a software project to be able to better understand the project, the ecosystem, or software engineering as a whole. This social perspective sheds light on the technical aspects of the work, analyzed in a sociotechnical manner. This approach may help to reduce problems that arise because of problematic communication or coordination, a common cause of project failures. Indeed, social features are becoming part of software repositories and coding tools, as can be seen by the recent GitHub phenomenon.

Researchers have started leveraging the power of social network analysis to understand different aspects of the software development process. For example, the structure of the social organization often dictates the technical structure of the product, as stated in Conway's Law [16], and is a topic of ongoing research. Researchers are also investigating the interplay of complex interdependencies among technical artifacts and the people who create and maintain these artifacts [17]. Others have examined the relationships between the stakeholders based on qualitative studies, and have been developing software tools to support awareness and trust in global software development [18]. Finally, researchers are studying the effects when companies switch from hierarchical to network-centric ways of organizing work and how these companies leverage expertise within their organization [19, 20].

In this thematic series, researchers investigated how popular social network sites can be used to support requirement engineering [21]; how software engineering social networks lead to social debt [22]; how interactions are mined from software repositories to understand how companies collaborate and compete [23]; how complex information systems can be analyzed from a sociotechnical perspective [14]; how prestige in a social network correlates to socio-economic data [24]; and how a social networking site can support a professional development community. In the next section, we give a brief overview of each selected paper before we explain the selection process.

2 Papers

The set of papers selected for this thematic series covers different research approaches, such as quantitative and qualitative studies, tool development and archival analysis, open source software and industrial projects, analysis of existing social networking sites and development of a new one, etc. This gives a broad overview of the area, with its tools, platforms, and research methods. Next, we briefly describe each paper.

There is no surprise in the fact that social network sites, such as Facebook, are quite popular. What is still a surprise is how little these sites are used in the software development process. Software engineers can leverage these platforms to support software engineering activities, as illustrated by Seyff et al. [21] in the paper **"Using popular social network sites to support requirements elicitation, prioritization and negotiation."** The authors aim to engage a large number of heterogeneous, globally distributed, and potentially anonymous stakeholders, including end-users, in the process of requirement analysis. Thus, they cope with the current demands for short time-to-market periods and fast, easy, and inexpensive methods. The authors propose an approach comprising a set of activities for requirements elicitation, prioritization, and negotiation, and show how Facebook may support these activities. This is particularly useful when end users are not within immediate reach and it seamlessly integrates mobile devices in the process. The authors investigated the usefulness of the existing features and they conducted three exploratory studies within requirement engineering classes. Their findings show that end users were able to follow the process, expressing their needs, engaging in discussions, and using the Facebook "liking" feature for prioritizing their needs. The idea seems promising and can be further investigated in future studies.

Social networks are all about interactions among people, which are influenced by the cultural and collaboration processes of the underlining organization. Tamburri et al. [22] in the paper **"Social debt in software engineering: insights from industry"** investigated how accumulated sub-optimal decisions about people and their interactions turn into social debt, in a similar fashion in technical debt is accrued [25]. While technical debt has received a lot of attention over the years, social debt has remained relatively unexplored. As software engineering success is

highly dependent on the well-being of the development communities, managing social debt is crucial. It is necessary to pay attention to the socio-technical decisions as some costs are not immediately visible and their resolution is often postponed. Social debt may lead to an additional cost that occurs when strained social interactions get in the way of smooth software development and operation. Tamburri and colleagues conducted a qualitative study in a large software company to better characterize forces that lead to social debt, anti-patterns that cause social debt, and some best practices to pay back the debts. One of their findings is that social debt can be quantified by means of social network analysis.

Social network analysis of software engineering data can in fact reveal many interesting patterns and phenomena, as illustrated by Teixeira et al. [23] in their paper **"Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem."** The authors combined social network analysis (SNA) and mining software repositories (MSR) techniques on an open source software (OSS) repository. As the authors show, the OSS ecosystem is a rich source of information and archival data for analysis, which may reveal interesting relations and support theory building. Teixeira and colleagues visualized collaboration evolution in a sequence of networks in order to understand how rival firms collaborate in the open source ecosystem. Their findings show that competition for the same revenue model does not necessary affect collaboration within the ecosystem, and that developers do not necessarily work with developers from the same firm. They provide a rich description on how heterogeneous actors within a highly-networked ecosystem develop together a complex infrastructure for big-data in the open source arena.

Aram and Nemann [14] also employed social network analysis in their paper **"Multilayered analysis of co-development of business information systems."** They provided a fine-grained understanding of the participation of stakeholders in the co-development of a complex information system. They proposed a multilayered approach for the analysis of co-development from a sociotechnical perspective, analyzing at the same time the business and the technical structures. As previously discussed, mixing both social and technical aspects in the analysis of a software project can provide a rich perspective of the collaborative work. Besides, in both this and the previous paper, the reader can find interesting social network representations.

Source code hosting services have adopted the "social coding" idea as they are aware of the importance of social aspects to software development. GitHub is currently the most representative example of this phenomenon and it has been attracting many software projects. Figueira Filho et al. [24], in their paper **"A study on the geographical distribution of Brazil's prestigious software developers,"** used data from GitHub projects to analyze social network data from over 4,000 active users. GitHub integrates a source code hosting service with social networking features. Figueira Filho and colleagues analyzed how developers' prestige correlates to Brazilian socio-economic and demographic data per state. They hypothesized that socio-economic characteristics are essential in determining the success of a country's IT industry and they found some evidence that prestigious developers tend to be located in the most economically developed regions. The inference of software developers' prestige from the underlining social network is another good example of the use of social network analysis from a software engineering perspective.

Finally, in the last article of this thematic series, Magoutis et al. [26] report their experience in designing, developing, and evaluating a social networking site in the paper **"Design and implementation of a social networking platform for cloud deployment specialists."** They focus on a new kind of professional that is becoming increasingly important to the software industry, namely DevOps, whose expertise span the fields of development and operation of software systems. The social networking site enables the sharing of automated deployment scripts and reasoning about good deployment strategies by means of interaction with experts. The experimental development of this kind of platforms helps to understand the interaction needs of software engineers.

3 Papers selection

There were two independent cycles of submissions and the papers were published as soon as they became ready. Each manuscript went through several revisions before the final acceptance. We invited a number of leading experts in the area to form an initial editorial committee, which was increased on demand in order to provide valuable feedback for the authors and review different aspects of the papers. All manuscripts were reviewed by at least three members of the editorial committee. For some papers, more reviewers were invited to analyze the manuscripts from different perspectives. We received a total of 12 submissions, from which we accepted 6 for publication. The papers were reviewed by a total of 41 reviewers. The names of the editorial committee members are listed on the acknowledgements of this editorial.

4 Related special issues

We also refer the reader to other related special issues, which can be a source for additional understanding of the theme. Some related special issues are: from IEEE Software on "Bridging Software Communities through

Social Networking" [27] and on "Cooperative and Human Aspects of Software Engineering" [28], the special issue from IEEE TSE on "Socio-Technical Environment of Software Development Projects" [29], and the special issue from the International Journal of Human-Computer Studies on "Collaborative and Social Aspects of Software Development" [30].

5 Conclusion

In summary, social networks model people's communication, coordination, and cooperation, and they can be used as a foundation for tools supporting these activities. Bringing social networks to the foreground of software development practice brings a focus to its stakeholders. The concept also helps explore the notion of network-centric organizations that are required to deal with socio-technical dependencies.

We learned through the work on this thematic issue that using social networks as an analytical lens helps to unpack and learn about software engineering in new ways. In particular, we found that the notion of network-centric organizations makes it possible for us to highlight important relations in the cooperative practices that we might otherwise neglect. While many open-source communities might not be considered traditional 'organizations' since each member in the community is distributed, they still have a governance structure. In such open distributed governance, 'social network' analytics can help to not only understand the coordination challenges, but also to harness the potential for collaboration.

From the improved understanding of social network analysis applied to software engineering, features of social networking sites that can be useful for software engineers, and the sociotechnical perspective for software engineering activities, we hope that new research lines and tools will appear. In this thematic issue we have only begun to scratch the surface of the potential of using social networking in software engineering; the area still has many open challenges and unexplored topics. We hope this issue spurs an interest for academics and professionals to dig deeper into this area in future work. Good luck delving into this exciting world of software engineering from a social network perspective.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors read and approved the final manuscript.

Acknowledgements

We thank all the authors, reviewers, editors-in-chief, and staff for the great work, which supported this thematic series on this very important topic both for the research community and for industry. In particular, we thank all the editorial committee members: Akinori Ihara (NAIST, Japan); Alexander Boden (University of Siegen, Germany); Andrea Magdaleno (UFRJ, Brazil); Andrew Begel (Microsoft Research, USA); Arfon Smith (GitHub, USA); Christian Bird (Microsoft Research, USA); Christoph Treude (USP, Brazil); Claudia Werner (UFRJ, Brazil); Damian Andrew Tamburri (VU University Amsterdam, Netherlands); Darja Smite (Blekinge Institute of Technology, Sweden); Erik Trainer (Carnegie Mellon University, USA); Francisco Servant (University of California Irvine, USA); Georgios Gousios (TU Delft, Netherlands); Gregorio Robles (Universidad Rey Juan Carlos, Spain); Igor Wiese (UTFPR, Brazil); Ingo Scholtes (ETH Zürich, Switzerland); Jason Tsay (Carnegie Mellon University, USA); Kari Kuutti (University of Oulu, Finland); Kate Ehrlich (IBM, USA); Kim Herzig (Microsoft Research, UK); Klaas-Jan Stol (Lero, Ireland); Leif Singer (University of Victoria, Canada); Leonardo A. F. Leite (SERPRO, Brazil); Marcelo Cataldo (ConnReperio, LLC, USA); Marcelo Maia (UFU, Brazil); Marcelo Zanetti (UFMA, Brazil); Marco Autili (Univaq, Italy); Mauricio Aniche (USP, Brazil); Milene Silveira (PUC-RS, Brazil); Mohamed Almorsy (Swinburne University of Technology, Australia); Myriam Lewkowicz (University of Technology of Troyes, France); Nicole Novielli (University of Bari, Italy); Panagiotis Karampelas (Hellenic Air Force Academy, Greece); Paolo Tell (ITU, Denmark); Patrick Wagstrom (IBM Watson Group, USA); Peter Axel A Nielsen (Aalborg University, Denmark); Peter Rigby (Concordia University, Canada); Sabrina Marczak (PUC-RS, Brazil); Yu Wu (Penn State University, USA); Yvonne Dittrich (IT University of Copenhagen, Denmark); Zadia Codabux (Mississippi State University, USA). The voluntary work of these researchers and practitioners were crucial for this thematic series.

Author details

[1]University of São Paulo, São Paulo, Brazil. [2]University of California, Irvine, Irvine, USA. [3]University of Copenhagen, Copenhagen, Denmark. [4]Oregon State University, Corvallis, USA.

Received: 23 October 2015 Accepted: 24 October 2015
Published online: 10 November 2015

References

1. Bird C, Nagappan N, Gall H, Murphy B, Devanbu P. Putting it all together: using socio-technical networks to predict failures, Proceedings of the 2009 20th International Symposium on Software Reliability Engineering (2009). 2009. p. 109–19.
2. Souza CR, de Quirk S, Trainer E, Redmiles DF. Supporting collaborative software development through the visualization of socio-technical dependencies, Proceedings of the 2007 GROUP (2007). 2007. p. 147–56.
3. Wiese IS, Côgo FR, Ré R, Steinmacher I, Gerosa MA. Social metrics included in prediction models on software engineering: a mapping study. In: Proceedings of the 10th International Conference on Predictive Models in Software Engineering. 2014. p. 72–81. ACM.
4. Bjørn P, Bardram J, Avram G, Bannon L, Boden A, Redmiles D, et al. Global Software Development in a CSCW perspective, Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing, Baltimore. 2014. doi:10.1145/2556420.2558863.
5. Bjørn P, Esbensen M, Jensen RE, Matthiesen S. Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration. ACM Trans Comput Hum Interact (ToChi). 2014;21(5):1–27.
6. Boden A, Nett B, Wulf V. Trust and social capital: revisiting an offshoring failure story of a small German software company, European Conference Computer Supported Cooperative Work (ECSCW'09), Vienna, Austria. 2009.
7. Jensen RE, Nardi B. The rhetoric of culture as an act of closure in cross-national software development department, Paper presented at the European Conference of Information System (ECIS), Tel Aviv. 2014.
8. Matthiesen S, Bjørn P, Petersen LM. "Figure out how to code with the hands of others": recognizing cultural blind spots in global software development. Baltimore, USA: Computer Supported Cooperative Work (CSCW); 2014.
9. Søderberg AM, Krishna S, Bjørn P. Global software development: commitment, trust and cultural sensitivity in strategic partnerships. J Int Manag. 2013;19(4):347–61.
10. Steinmacher I, Chaves AP, Gerosa MA. "Awareness support in distributed software development: a systematic review and mapping of the literature". J Comput Supported Coop Work. 2013;22(Issue 2):113–58. Springer Netherlands, ISSN: 0925–9724. doi:10.1007/s10606-012-9164-4.
11. Cataldo M, Wagstrom PA, Herbsleb JD, Carley KM. Identification of coordination requirements: implications for the design of collaboration

and awareness tools. In: Proceedings of the 20th Anniversary ACM Conference on Computer Supported Cooperative Work, CSCW 2006, November 4, 2006 - November 8, 2006. Banff, AB, Canada: Association for Computing Machinery; 2006. p. 353–62.
12. Bettenburg N. Studying the impact of developer communication on the quality and evolution of a software system (Doctoral dissertation, Queen's University). 2014.
13. Wasserman S, Faust K. Social network analysis: methods and applications. 1st ed. Cambridge, New York: Cambridge University Press; 1994.
14. Aram M, Neumann G. Multilayered analysis of co-development of business information systems. J Internet Serv Appl. 2015;6(13):1–30.
15. Strogatz SH. Exploring complex networks. Nature. 2001;410(6825):268–76. doi:10.1038/35065725.
16. Conway ME. How do committees invent?". Datamation Thompson. 1968;14(4):28–31.
17. Cataldo M, Herbsleb J, Carley K. Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity, Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM'08, Kaiserslautern, Germany). 2008. p. 2–11.
18. Sarma A, Van der Hoek A. Palantir: coordinating distributed workspaces. In: Computer software and applications conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International. 2002. p. 1093–7. IEEE.
19. Abrams S, Mark G. Network-centricity: hindered by hierarchical anchors. In: Proceedings of the 2007 Symposium on computer human interaction for the management of information technology. 2007. p. 7. ACM.
20. Ehrlich K, Lin C-Y, Griffiths-Fisher V. Searching for experts in the enterprise: combining text and social network analysis, The 2007 International ACM Conference on Supporting Group Work (GROUP 2007, Sanibel Island, Florida). 2007. p. 117–26.
21. Seyff N, Todoran I, Caluser K, Singer L, Glinz M. Using popular social network sites to support requirements elicitation, prioritization and negotiation. J Internet Serv Appl. 2015;6(7):1–16.
22. Tamburri DA, Kruchten P, Lago P, van Vliet H. Social debt in software engineering: insights from industry. J Internet Serv Applications. 2015;6(10):1–17.
23. Teixeira J, Robles G, González-Barahona JM. Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem. J Internet Serv Appl. 2015;6(14):1–27.
24. Figueira Filho F, Perin MG, Treude C, Marczak S, Melo L, da Silva IM, et al. A study on the geographical distribution of Brazil's prestigious software developers. J Internet Serv Appl. 2015;6(17):1–12.
25. Cunningham W. The WyCash portfolio management system. OOPS Messenger. 1993;4(2):29–30.
26. Magoutis K, Papoulas C, Papaioannou A, Karniavoura F, Akestoridis DG, Parotsidis N, et al. Design and implementation of a social networking platform for cloud deployment specialists. J Internet Serv Appl. 2015;6(19):1–27.
27. Begel A, Bosch J, Storey MA. Bridging software communities through social networking. IEEE Software. 2013;1:26–8.
28. de Souza CR, Sharp H, Singer J, Cheng LT, Venolia G. Guest editors' introduction: cooperative and human aspects of software engineering. Software IEEE. 2009;26(6):17–9.
29. Cataldo M, Ehrlich K, Mockus A. Guest editors' introduction: special section on the socio-technical environment of software development projects. IEEE Trans Softw Eng. 2011;3:305–6.
30. Good J, Romero P. Collaborative and social aspects of software development. Int J Hum Comput Stud. 2008;66(7):481–3.