

Editorial

A Call to Promote Soft Skills in Software Engineering

Luiz Fernando Capretz, PhD, PEng^{1*}; Faheem Ahmed, PhD, PEng²

¹Department of Electrical and Computer Engineering, Western University, London, ON N6A5B9, Canada

²Department of Computing Science, Thompson Rivers University, Kamloops, BC V2C0C8, Canada

*Corresponding author

Luiz Fernando Capretz, PhD, PEng

Professor of Software Engineering, Department of Electrical and Computer Engineering, Western University, London, Ontario N6A5B9, Canada

E-mail: lcapretz@uwo.ca

Article information

Received: December 4th, 2017; **Accepted:** February 19th, 2018; **Published:** February 20th, 2018

Cite this article

Capretz LF, Ahmed F. A call to promote soft skills in software engineering. *Psychol Cogn Sci Open J*. 2018; 4(1): e1-e3. doi: [10.17140/PCSOJ-4-e011](https://doi.org/10.17140/PCSOJ-4-e011)

We have been thinking about other aspects of software engineering for many years; the missing link in engineering software is the soft skills set, essential in the software development process. Although soft skills are among the most important aspects in the creation of software, they are often overlooked by educators and practitioners. One of the main reasons for the oversight is that soft skills are usually related to social and personality factors, i.e., teamwork, motivation, commitment, leadership, multi-culturalism, emotions, interpersonal skills, etc. This editorial is a manifesto declaring the importance of soft skills in software engineering with the intention to draw professionals' attention to these topics. We have approached this issue by mentioning what we know about the field,¹⁻⁶ what we believe to be evident,⁷⁻⁹ and which topics need further investigation.¹⁰⁻¹⁴ Important references to back up our claims are also included.

Software engineers take pride in the depth of their technical expertise, which separates them from the crowd. But, what makes a good software engineer? First, it is the technical knowledge of relevant methodologies and techniques (i.e., hard skills), as well as the skills necessary for applying that knowledge in practice. Second, but nonetheless important, it is a set of soft skills, in particular collaboration, communication, problem-solving and similar interpersonal and critical thinking skills that are expected from software engineering professionals. In other words, software engineers need both hard and soft skills in order to be successful at the workplace.¹⁵

Engineering software involves performing tasks in distinct areas, such as system analysis, software design, programming, software testing, and software evolution/maintenance; other software occupations in a software team include the project manager, troubleshooter, helpdesk personnel, database administrator, and so forth. Thus today, specialties within software engineering are as diverse as in any other profession.¹⁶ Additionally, software engi-

neers need to communicate very effectively with users and team members, which reinforces the idea that the people dimension of software engineering is as important as the technical expertise.¹⁷

However, computer science and software engineering curricula focus mainly on developing hard skills, thus paying lip services to soft skills. Even the latest guideline for teaching and learning software engineering, namely the SWEBOK V3.0 and IEEE/ACM Curriculum Guide, praises technical competence and gives marginal consideration to vaguely characterized non-technical (soft) skills. We strongly believe that computer science and software engineering curricula should put more emphasis on developing and assessing both hard and soft skills and that both types of skills should be acknowledged.

Psychology tells us that not everybody excels at all types of tasks. Successful software development also depends significantly on how software practitioners perform their tasks and how they interact with their peers.

At the present time, very few courses in computer science or software engineering curricula touch upon the subjects of teamwork and soft skills assessment.¹⁸ It is difficult to even find a university that has an entire course on the human aspects of software engineering, with the exception of a couple of computer science departments.¹⁹⁻²⁰ Unfortunately, soft skills are far from being a part of mainstream software engineering education; a course devoted to this subject would be an ideal solution to remedy this situation. It is important to notice here that this is not only our opinion; when we met senior software engineers who were leading teams, they told us that there was a need for a soft skills set among software developers. The productivity and satisfaction of technical people have been seriously undermined by the lack of this kind of knowledge in computer science education, software engineering, and related areas.

Engineering software revolves around methodologies establishing (more or less) well-defined processes to develop a product.²¹ However, because people execute these processes, no matter how good they are, the ultimate success lies in how the processes are being executed especially when people interact. The successful implementation of any process methodology ultimately depends on how employees perceive that methodology. For example, besides the criticism, iterative and incremental approaches of Agile methodology have paved the way to reduce costs and shorten development time. Arguably, Agile is not a methodology but a mindset or a different way of managing people to develop software products. Although Agile is suited for many practitioners it is not suited for everyone.²² Agile methodologies emphasize on some social aspects of the process such as competency, collaboration, trust, analytics, and devolution of decision-making. However, numerous other social attributes which vary from organization to organization such as work ethics, respectful work environment, reward mechanism, internal politics, relationship management, conflict management, etc., are not addressed and have been given little or no attention by the software industry in general.

Furthermore, software engineers who are promoted to managerial positions, typically experience similar emotions like others do. In addition to earning more money, and possibly a better parking spot, promotions are also a recognition of their expertise and dedication by their superiors and peers. At the same time, promotions also cause headaches for software engineers. In addition to the common challenges, such as becoming the boss of their previous colleagues (or foes), software managers need to manage conflicts, negotiate and compromise, measure performance, etc. Software engineers often struggle to adopt a managerial culture because they are not trained to be managers. Thus, software engineers moving from a technical to a managerial position struggle with the transition because now their focus shifts from product development to managing people, which is relatively more difficult. The success or failure of a software manager resides heavily on the performance and creations of others. The dilemma of having technical knowledge and a managerial position at the same time often forces them to wade in and attempt to fix problems that rightly belong to a subordinate. Software managers may also find it hard to accept that, eventually, someone younger than them and working under them will outstrip their technical knowledge of some domain because they themselves are too busy managing people to keep up with rapid changes in technology. One of the reasons why software managers tend to keep looking at the trees instead of at the forest (that is, the big picture) is that they have been trained under the notion of the predictable behavior of software, whereas the behavior of people is less predictable.

In summary, technical people tend to overlook the importance of soft skills as it is unrelated to their technical area and because their training is in dealing with the technical issue; thus considering the soft skills in the software development process to be foreign to them, since the field deals with human factors and touches social sciences. These are topics that software professionals do not have expertise in. We believe that it is high time for the software development community to realize that the human element is pivotal to success in the engineering of

software. We have to recognize that software engineering is a people-intensive discipline, hence requires appropriate treatment. Therefore, human aspects of software engineering are important subjects to teach, study and research. We urge software engineers to take on this challenge.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Cruz S, Silva FQB, Capretz LF. Forty years of research on personality in software engineering: A mapping study. *Comput Human Behav.* 2015; 46(1): 94-113. doi: [10.1016/j.chb.2014.12.008](https://doi.org/10.1016/j.chb.2014.12.008)
2. Capretz LF. Personality types in software engineering. *Int J Hum Comput Stud.* 2003; 58(2): 207-214. doi: [10.1016/s1071-5819\(02\)00137-4](https://doi.org/10.1016/s1071-5819(02)00137-4)
3. Ahmed F, Capretz LF, Campbell P. Evaluating the demand for soft skills in software development. *IEEE IT Professional.* 2012; 14(1): 44-49. doi: [10.1109/MITP.2012.7](https://doi.org/10.1109/MITP.2012.7)
4. Capretz LF, Ahmed F. Why do we need personality diversity in software engineering. *ACM SIGSOFT Software Engineering Notes.* 2010; 35(2): 1-11. doi: [10.1145/1734103.1734111](https://doi.org/10.1145/1734103.1734111)
5. Capretz LF, Ahmed F. Making sense of software development and personality types. *IEEE IT Professional.* 2010; 12(1): 6-13. doi: [10.1109/MITP.2010.33](https://doi.org/10.1109/MITP.2010.33)
6. Ahmed F, Capretz LF, Bouktif S, Campbell P. Soft skills and software development: A reflection from the software industry. *International Journal of Information Processing and Management.* 2013; 4(3): 171-191. doi: [10.4156/ijipm.vol4.issue3.17](https://doi.org/10.4156/ijipm.vol4.issue3.17)
7. Capretz LF. Bringing the human factor to software engineering. *IEEE Software.* 2014; 31(2): 102-104. doi: [10.1109/MS.2014.30](https://doi.org/10.1109/MS.2014.30)
8. Capretz LF, Varona D, Raza A. Influence of personality types in software tasks choices. *Comput Human Behav.* 2015; 52: 373-378. doi: [10.1016/j.chb.2015.05.050](https://doi.org/10.1016/j.chb.2015.05.050)
9. Waychal P, Capretz LF. Universality of egoless behavior of software engineering students. *International Journal of Technology and Human Interaction.* 2018; 14(1): 99-112. doi: [10.4018/IJTHI.2018010106](https://doi.org/10.4018/IJTHI.2018010106)
10. Jia J, Mo H, Capretz LF, Zupeng C. Grouping environmental factors influencing individual decision-making behavior in software projects: A cluster analysis. *Journal of Software: Evolution and Process.* 2018; 30(1): 15. doi: [10.1002/smr.1913](https://doi.org/10.1002/smr.1913)
11. Capretz LF, Ahmed F, Silva FQB. Soft sides of software. *Information and Software Technology.* 2017; 92: 92-94. doi: [10.1016/j.infsof.2017.07.011](https://doi.org/10.1016/j.infsof.2017.07.011)
12. Silva FQB, Monteiro VFC, Santos IE, Capretz LF. How

software development group leaders influence team members' behavior. *IEEE Software*. 2016; 33(5): 106-109. doi: [10.1109/MS.2016.120](https://doi.org/10.1109/MS.2016.120)

13. Monteiro VFC, Silva FQB, Capretz LF. A pilot case study on innovative behaviour: Lessons learned and directions for future work. Paper presented at: *IEEE 10th International Symposium on Empirical Software Engineering and Measurement (ESEM 2016)*; 2016; Ciudad Real, Spain. doi: [10.1145/2961111.2962618](https://doi.org/10.1145/2961111.2962618)

14. Gilal AR, Jaafar J, Capretz LF, Omar M, Basri S, Aziz IA. Finding an effective classification technique to develop a software team composition model. *Journal of Software: Evolution and Process*. 2018; 30(1): 12. doi: [10.1002/smr.1920](https://doi.org/10.1002/smr.1920)

15. Graziotin D, Wang X, Abrahamsson P. Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ*. 2014; 2: e289; doi: [10.7717/peerj.289](https://doi.org/10.7717/peerj.289)

16. Fritz T, Begel A, Müller SC, Yigit-Elliott S, Züger M. Using psycho-physiological measures to assess task difficulty in software development. Paper presented at: *36th IEEE International Conference on Software Engineering*. May 31 - June 07, 2014; Hyderabad, India. doi: [10.1145/2568225.2568266](https://doi.org/10.1145/2568225.2568266)

17. Fritz T, Muller SC. Leveraging biometric data to boost software development productivity. Paper presented at: *23rd IEEE Conference on Software Analysis, Evolution, and Reengineering*. March 14-18, 2016;

Suita, Japan. doi: [10.1109/SANER.2016.107](https://doi.org/10.1109/SANER.2016.107)

18. Petkovic D, Thompson G, Todtenhoefer R, et al. e-TAT: online tool for teamwork and soft skills assessment in software engineering education. Paper presented at: *40th ASEE/IEEE Frontiers in Education Conference*. October 27-30, 2010; Washington, DC, USA. doi: [10.1109/FIE.2010.5673130](https://doi.org/10.1109/FIE.2010.5673130)

19. Hazzan O, Tomayko JE. Reflection processes in the teaching and learning of human aspects of software engineering. Paper presented at: *17th IEEE Conference on Software Engineering Education and Training (CSEET'04)*. March 1-3, 2004; Norfolk, Virginia, USA.

20. Donald J, Lachapelle S, Sasso T, Augusto K, Gonzales-Morales MG. Why does it matter? Explaining the importance of complimentary studies to first-year engineering students. *Canadian Engineering Education Conference (CEE'16)*. 2016. doi: [10.24908/pceea.v0i0.6474](https://doi.org/10.24908/pceea.v0i0.6474)

21. Ahmed F, Capretz LF, Bouktif S, Campbell P. Soft skills requirements in software development jobs: A cross-cultural empirical study. *Journal of Systems and Information Technology*. 2012; 14(1): 58-81. doi: [10.1108/13287261211221137](https://doi.org/10.1108/13287261211221137)

22. Kuusinen K, Gregory P, Sharp H, Barroca L, Taylor K, Wood L. Knowledge sharing in a large agile organisation: A survey study. In: Baumeister H, Lichter H, Riebisch M, eds. *Agile Processes in Software Engineering and Extreme Programming*. Cham, Switzerland: Springer; 2017.