

A Qualitative Study of Ethnographic Variables Affecting Exchange of Knowledge among Online Help Seekers and Elite Digital Natives with Recommendations for Getting the Best Technology Assistance Possible

Donald Allen Deever¹, Steven Grubaugh², Gabe Gonzalez³ & Greg Levitt⁴

Abstract

In our contemporary world, people of all ages are coding and almost everyone needs general and specific expert technology help from time to time. Software developers as technology experts are a unique ethnographic community that exhibit particular cultural traits, and who work together according to a previously unwritten yet specific set of rules and practices (Wellman & Gulia, 2018). This ethnographic qualitative study examines how experienced members of a small online software development community react and learn through help-seeking questions of newer members who need support in understanding the idiosyncrasies of a particular software development language (Bosch & D'Mello, 2017). This research analyzes the communications between askers and helpers, seeking to discover why the requests for some coding advice goes unanswered, why other new users receive less-than-satisfactory answers, and still yet why certain new members receive beneficial to highly useful assistance and information regarding their requests (Schueller, Tomasino, & Mohr, 2017). Study implications potentially extend to many domestic and world-wide online learning communities and help create a research to practice foundational knowledge of the best practices for asking for online help in order to receive maximum assistance from the most experienced members of various online help forums.

Introduction

Online technology literature indicates that one of the least studied cultures in the modern age is the world of beginning computer software developers and technology help seekers (Wellman & Gulia, 2018). Much of the future of virtually every educator, scholar, office worker, government employee, and member of the private business sector will eventually be dependent upon emerging technologies. Few researchers seem to have considered how beginners can best deal with the current gurus of virtual gizmos and gadgets, which is a vital consideration for those who dream of expanding knowledge for themselves and others. The reason for this dearth of studies is understandable since there seems to be a natural reluctance on the part of scholars to pierce the veil of how such a collective body of creative minds operate and by what codes of honor they bestow or withhold their knowledge to upcoming generations of learners. This study has strong implications for enhancing professional development to improve providing and receiving online or telephone help to the community of online help seekers and providers. Reviewing the literature, it can be seen that much research exists on the related topic of successfully using Collaborative Question Answering (CQA) platforms to effectively pose questions and receive satisfactory answers for technical questions. However, the majority of such studies take a close look at only the biggest and most popular CQA locations for sharing knowledge within such sites as Yahoo! Answers (answers.yahoo.com), Answers.com, Facebook Questions, and Stack Overflow.com.

¹ University of Nevada, Las Vegas, United States

² University of Nevada, Las Vegas, United States

³ University of Nevada, Las Vegas, United States

⁴ University of Nevada, Las Vegas, United States

When it comes to software developer communities, studies concerning Stack Overflow (Anderson, Huttenlocher, Kleinberg, & Leskovec, 2012) were particularly interesting to this research, since Stack Overflow represents the premiere gathering place of software developers for virtually every programming language in the world. By 2014, Stack Overflow boasted more than 4 million registered users and a library of more than 10 million questions that had been asked and answered on the topic of computer programming and software development (Barua, Thomas, & Hassan, 2014). Likewise, GitHub accomplishes a similar purpose as Stack Overflow but with the possibility of doing an overall more efficient job of it (Vasilescu, Filkov, & Serebrenik, 2013). In contrast, this ethnographic study has observed a minor online forum for software developers with the advantage of being more observable due to its far more manageable size.

In any event, the closest study to this research centered on the Web 2.0 resources for learning new technology within Stack Overflow. In that particular study, three researchers took a close look at what kinds of questions get answered well, which ones remain unanswered, and how to phrase questions effectively (Treude, Barzilay, & Storey, 2011). Despite the similarities between that study and this present one, this study is unique in that it explores the practices of how new students of technology successfully ask questions of experts, and do so from the standpoint and frame of mind or attitude in which questions are asked rather than in the manner that questions were phrased. In this sense, this study is truly more ethnographic in nature, since it focuses on how new students of technology act within the software developer culture in order to receive the most beneficial reactions from its members. An overlap that exists between this research and one study accomplished on a larger software developer site (Treude et al., 2011) is in the area of who is most likely to answer questions and why.

In comparison to forums that represent more popular programming languages such as Java or JavaScript or C++, the identity of this programming language is being withheld in this study. It will be said that this observed community has roots that date back to the HyperCard days of Apple Computer, and which has since evolved exponentially into a world-wide following (Europe, North America, and Asia) of developers of multi-platform desktop software and mobile applications for Windows, Apple/Macintosh, and Linux operating systems (Palmer, 2013).

Purpose

Like virtually all major software development languages, various question and answer forums exist for this programming resource to help speed up the learning curves of new users. The particular section of the forum looked at in this study is one that is open to the general public to view. In fact, one of the authors of this study has been actively immersed in this elite digital native culture for nine years and has therefore become accustomed to observing how various new members are treated in response to requests for help (Guo, 2017). Using an inside-out and outside-in look at reactions to beginner's requests for help, the team involved in this qualitative research project found that they were able to offer reasonable interpretations as to the reasons behind the varying degrees of quality of help received by "askers," identified in this qualitative analysis as new learners of coding and technology, who were new to this programming language. In this respect, this study focused more on what the askers did when posing questions to effectively attract help, and considered the actions of the mentors, or expert "helpers," mostly from the standpoint of how they were reacting to the askers (Foong, Dow, Bailey & Gerber, 2017).

To arrive at the conclusions that are presented in this study, a combination of nine years of online experience with this particular community in conjunction with observations of twenty-two subjects, have allowed for interpretations to be made on the suggested practices to be used when asking for highly technical help online, many of which are practices that might have implications in other online situations outside the realm of computer programming/software development forums and learning communities. Such insights might be extended to virtual classrooms, helping create a foundation of knowledge of the best practices for asking for online help in order to promote more beneficial responses in many fields. Such insights might also be applicable to online learning sites where students are becoming increasingly dependent on peer-assistance, such as in flipped learning classroom scenarios.

Overall, it is hoped that this study might offer valuable insights into helping students of all ages to better understand how to approach the process of asking for assistance from expert helpers online through a variety of help forum genres.

Researcher Background

Digital natives are no longer a minority but a dominant culture worldwide, and it would be rare for the ethnographic researcher to find a subculture of people whose lives are not a part of the modern digital revolution. Just as there are many subcultures within any dominant culture, with varying degrees of privilege or status, so too does there exist a hierarchy among digital natives with perhaps the pinnacle of prestige being amongst the subculture of software developers.

For the past nine years, the inside researcher of this study observed the elite of the digital natives, having joined a public forum of computer programmers specializing in one particular programming language whose name has been omitted from this paper in the interest of the privacy of the help seekers and help givers. Facing such a great divide in lifestyle and even modern philosophical outlooks, he was at a decided disadvantage when it came to immersing himself into the lives of software developers; however, to date, he has made approximately 140 postings or contributed to various threads of discussion on the forum that forms the basis of this study. This research team, however, although related to their own experiences of conversing with elite digital natives, focuses more on their observations of other relative newcomers to the software development world. Therefore, what the research team has attempted to accomplish is to interpret the manner in which askers acted positively within this culture to inspire maximum goodwill from the help givers or expert mentors.

The inside researcher's own requests for assistance on the software developer forum are important to this study in order to provide an experienced interpretation of how his own personal successful communications related to what the research team observed in the help-seeking attempts of the subjects of this study. It is through the lens of his personal experience that the team was able to interpret and offer insights, which if heeded, may lead to better reactions and results for new learners of coding and technology when initially seeking help in understanding and utilizing computer language for the purposes of learning to develop software (Stephenson, 2018). In short, the purpose of this study is to determine the unwritten practices that seem to lead most often to successful communications between new software developers seeking help from experienced computer programmers. Moreover, it is the intention of this study that the insights offered herein could lead to new learners confidently venturing into the creative realm of software development and finding quicker success in locating effective mentors to assist them in establishing a proper foundation of knowledge and skills.

Methods

The main research question in this ethnographic study was simply, "What unspoken practices, if any, if followed by new students of software development who were asking for help, seemed to produce the best responses from the mentor programmers who attempted to answer their questions?"

Considering the research question, this study sought mainly to look at the interaction between two major categories of subjects, which have been labeled as "askers" and "helpers." Askers represent a single group of subjects who were new at learning to use the particular programming and development language that this research focused upon for this study. The reason askers are considered to be a single category is because of all cases observed in this study, every asker used the portion of the forum titled, "Getting Started with Development Software - Complete Beginners."

When it came to the helpers, however, their group was divided up into three separate categories that included "power helpers," "moderate helpers," and "regular users." The distinction between these three groups of subjects has been based on the amount of postings they have made in the forum. When it comes to the top tier of helpers, they are highly experienced software developers who have made anywhere from 3,000 to 8,000 postings to help others. The more moderate helpers are those who have made from 1,000 to 2,999 postings, and finally, the regular helpers are those people who have made less than 1,000 postings.

In keeping with ethnographic traditions, it was a goal of this research team that a minimum of twenty subjects were to be observed. The actual numbers turned out to include twenty-two separate individuals classified as askers, while the total number of the group labeled as helpers equaled exactly twenty separate individuals. The reason for using the term "individuals" is due to the observation that three of the askers initiated two different requests for help, while at least half of the helpers made dozens of responses each.

The time period chosen for observing these subjects was conducted over a recent holiday period which is not being specified in order to maintain participant anonymity. This represented a total of fifteen days, or precisely half a month. The reason for choosing that time period was due to subjects logically having more days off and therefore more time to experiment with their newfound interest in computer programming and software development.

Results

The raw data for this study consisted of sixty-eight full pages of transcripts that were analyzed using both content analysis and coding by hand. The sixty-eight pages mentioned were collected from exchanges of dialog that took place between people who were new to the world of software development and veteran users with far greater knowledge and experience in working with the programming language involved.

During the fifteen-day calendar period observed, a total of twenty-five questions or pleas for help went out from beginning programmers in need of advice to get a computer function that they were creating to work properly. This seeking of help resulted in a total of 140 messages or “threads” between askers and helpers, which meant that the average number of messages per response was 5.6 threads. In other words, each issue was asked and responded to with approximately six exchanges of dialogue. Altogether, the number of threads per plea for help ranged from a high of eighteen messages to a low of two messages, with seven messages and three messages per request being the most common number of exchanges that occurred on four occasions. The second most common number of responses were represented by five messages, four messages, and two messages, all of which occurred on three separate occasions. All other numbers of exchanges per request (18, 13, 11, 10, 9, and 6) only occurred once. Out of the 140 messages that went back and forth between askers and users, it was found that an equal number of exchanges took place between askers and helpers, which amounted to 70 responses each.

Besides recording the number of exchanges between subjects being observed, the various types of topics being broached were also listed and included how-to questions concerning all of the following subjects in software development: Apps Creation = 2; Databases = 3; Deletion = 1; Digital Photos = 1; Games = 2; Getting Started = 1; Internet = 1; Key Controls = 2; Menus = 1; Multimedia = 2; Naming Objects = 1; Printing = 3; Shells = 1; Software Upgrades = 1; Spreadsheet = 2; Troubleshooting = 1; and Tutorials = 1. As can be seen, the two most popular topics were printing and databases at three discussions each. The second-most popular discussions concerned games, key controls, spreadsheets, and multimedia at two discussions each.

Other important results that were noted concerned how many times an asker received “useful” responses to their question, which is to say how often they were totally satisfied with an answer that was understandable and solved the problem they were experiencing. This category of data was separated into four types of observed outcomes that included “Useful,” “Moderately Useful,” “Self-answered,” and “Not Useful,” in which case it meant that no usable answer was forthcoming or the answers given were inadequate or the asker of the question removed him/herself and failed to interact further with those who wanted to help. According to the useful to not useful outcomes, the data demonstrated that of fourteen times that an asker sought help, they found the response to be Useful. There were six occurrences in which there were “Moderately Useful” responses provided and received. In the category of Self-answered questions (askers who resolved their question), there were two occurrences. Last, three askers found received responses that were Not Useful.

Conclusions

Interpreting the results, it could be seen that more than half of the time (fourteen out of twenty-five times), askers received information and responses that they deemed Useful. This observation should be encouraging to anyone providing or conversely attempting to acquire knowledge from veterans of technology software development (Perrow, 2017). The outcomes of the data show that askers or, new students of technology are twice as likely to receive substantive (Useful) responses. In the case of those finding Moderately Useful responses, this research team observed that there were several common mistakes made by askers, such as not providing enough information to receive adequate help, not responding to requests from helpers for more information, not being more up-front when an answer was not working for them, not fully admitting when they did not understand the advice received, and not specifically asking for examples of code to help them see the solution to their problem (Kim & Ko, 2017). It also needs to be realized that not every request for help made can be answered. Every software development language has its own limitations or bugs that occasionally prevent a successful conclusion to a particular function.

Implications for Best Practices of Successful Asker Communications

After careful analysis of the resulting data surrounding useful responses and the exchange of knowledge among new learners and veteran users, and in light of the analysis of the transcripts garnered from this study, this research team identified the major variables that appear to be important factors of success when seeking help from elite digital natives with recommendations for getting the best technology assistance possible. As such, askers, whether developing software or simply using an online or telephonic help line for general technology help requests, should avoid the common mistakes identified in this study and ensure that they 1) provide enough information to receive adequate help, 2) respond to requests from helpers for more information, 3) remain more up-front when an answer is not working for them, 4) fully admit when they do not understand the advice received, and 5) specifically asking for examples of code or other technical or digital tangibles to help them see the solution to their problem.

Moreover, these communication practices can be fairly easily emulated by beginners wanting to successfully immerse themselves for the first time into the high-tech world of learning to become software developers as well as effective communicators with more general technology help requests. It was observed by this research team that paying strict adherence to these forthcoming communication practices can significantly improve one's chances of successfully receiving valuable answers to the how-to questions that they seek.

References

- Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2012). Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, 850-858.
- Asaduzzaman, M., Mashiyat, A.S., Roy, C.K., & Schneider, K.A. (2013, May). Answering questions about unanswered questions of stack overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, 97-100. IEEE Press.
- Barua, A., Thomas, S. W., & Hassan, A. E. (2014). What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3), 619-654.
- Bosch, N., & D'Mello, S. (2017). The affective experience of novice computer programmers. *International Journal of Artificial Intelligence in Education*, 27(1), 181-206.
- Dittrich, Y., Floyd, C., & Klischewski, R. (2002). *Social Thinking--Software Practice*. MIT Press.
- Foong, E., Dow, S. P., Bailey, B. P., & Gerber, E. M. (2017, May). Online feedback exchange: A framework for understanding the socio-psychological factors. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 4454-4467). ACM
- Guo, P. J. (2017, May). Older adults learning computer programming: motivations, frustrations, and design opportunities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 7070-7083). ACM
- Kim, A. S., & Ko, A. J. (2017, March). A pedagogical analysis of online coding tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 321-326). ACM
- Morrison, P., & Murphy-Hill, E. (2013, May). Is programming knowledge related to age? an exploration of stack overflow. In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, 69-72. IEEE.
- Palmer, E. K. (2013). Use of LiveCode in CS0. *Journal of Computing Sciences in Colleges*, 29(1), 94-94.
- Perrow, M. (2017). Strengthening the Conversation in Blended and Face-to Face Courses: Connecting Online and In-Person Learning with Crossover Protocols. *College Teaching*, 65(3), 97-105.
- Schueller, S. M., Tomasino, K. N., & Mohr, D. C. (2017). Integrating human support into behavioral intervention technologies: the efficiency model of support. *Clinical Psychology: Science and Practice*, 24(1), 27-45.
- Stephenson, J. (Ed.). (2018). *Teaching & learning online: new pedagogies for new technologies*. Routledge
- Treude, Barzilay, & Storey (2011). How do programmers ask and answer questions on the web?: NIER track. In *Proceedings of the ACM/IEEE International Conference on Software Engineering, ICSE '11*, 804-807.
- Vasilescu, B., Filkov, V., & Serebrenik, A. (2013, September). StackOverflow and GitHub: Associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, 188-195. IEEE.
- Wellman, B., & Gulia, M. (2018). Net-surfers don't ride alone: Virtual communities as communities. In *Networks in the global village* (pp. 331-366). Routledge.