

Arsonists or Firefighters? Affectiveness in Agile Software Development

Marco Ortu¹, Giuseppe Destefanis², Steve Counsell², Stephen Swift², Roberto Tonelli¹, and Michele Marchesi¹

¹ DIEE, University of Cagliari,

|marco.ortu|michele|roberto.tonelli|@diee.unica.it

² Brunel University, Uxbridge, UK

|giuseppe.destefanis|steve.counsell|stephen.swift|@brunel.ac.uk

Abstract. In this paper, we present an analysis of more than 500K comments from open-source repositories of software systems developed using agile methodologies. Our aim is to empirically determine how developers interact with each other under certain psychological conditions generated by politeness, sentiment and emotion expressed within developers' comments. Developers involved in an open-source projects do not usually know each other; they mainly communicate through mailing lists, chat, and tools such as issue tracking systems. The way in which they communicate affects the development process and the productivity of the people involved in the project. We evaluated politeness, sentiment and emotions of comments posted by agile developers and studied the communication flow to understand how they interacted in the presence of impolite and negative comments (and *vice versa*). Our analysis shows that "firefighters" prevail. When in presence of impolite or negative comments, the probability of the next comment being impolite or negative is 13% and 25%, respectively; *ANGER* however, has a probability of 40% of being followed by a further *ANGER* comment. The result could help managers take control the development phases of a system, since social aspects can seriously affect a developer's productivity. In a distributed agile environment this may have a particular resonance.

Key words: agile, data mining, human aspect

1 Introduction

The study of emotions and psychological status of developers and people involved in the software-building system is gaining the attention of both practitioners and researchers [12]. Feldt et al. [8] focused on personality as one important psychometric factor and presented initial results from an empirical study investigating the correlation between personality and attitudes to software engineering processes and tools.

Software is a complex artefact which requires sharing of knowledge, team building and exchange of opinion between people. While it has been possible to

standardise classical industrial processes (e.g., car production), it is still difficult to standardise software production. Immateriality plays a major role in the complexity of software and despite attempts to standardise the software production process, software engineering is still a challenging and open field. There are too many constraints to take into account. Developers build an artefact that will be executed on a machine; software metrics, design patterns, micro patterns and good practices help to increase the quality of a software [4, 6], but developers are humans and prone to human sensitivities. Coordinating and structuring developer teams is a vital activity for software companies [17] and dynamics within a team have a direct influence on group success; on the other hand, social aspects are intangible elements which, if monitored, can help the team in reaching its goals. Researchers are increasingly focusing their effort on understanding how the human aspects of a technical discipline can affect the final results [3][7][11].

Open-source development usually involves developers that voluntarily participate in a project by contributing with code. The management of such developers could even be more complex than the management of a team within a company, since developers are not in the same place at the same time and coordination becomes more difficult. The absence of face-to-face communication mandates the use of mailing lists, electronic boards, or specific tools such as Issue Tracking Systems. Being rude when writing a comment or replying to a contributor can affect the cohesion of the group and the successfulness of a project; equally a respectful environment is an incentive for new contributors joining the project [13, 20, 24].

In this paper, we empirically analyze more than 500K comments from Ortu et al. [17] to understand how agile developers behave when dealing with polite/impolite or positive/negative (sentiment) issue comments. We empirically built three Markov chain models with states for politeness (polite, neutral, impolite), sentiment (positive, neutral, negative), and emotions (joy, anger, love, sadness). We aim to answer the following questions:

- Do developers change behaviour in the context of impolite/negative comments?
- What is the probability of shifting from comments holding positive emotions to comments holding negative emotion?

The remainder of this paper is structured as follows: In the next section, we provide a summary of related work. Section 3 describes the dataset used for this study and our approach/rationale to evaluate affectiveness of comments posted by developers. In Section 4, we present the results and elaborate on the research questions we address. Section 5 discusses the threats to validity. Finally, we summarize the study findings in Section 6.

2 Related Work

Several recent studies have demonstrated the importance and relationship of productivity and quality to human aspects associated with the software devel-

opment process. Ortu et al. studied the effect of politeness [16] and emotions [15] on the time required to fix any given issue. The authors demonstrated that emotions did have an effect on the issue fixing time. Research has focused on understanding how the human aspects of a technical discipline can affect final results ([3], [7], [11]), and the effect of politeness ([14], [23], [25]). The Manifesto for Agile Development indicates that people and communications are more essential than procedures and tools [2]. Several recent studies have demonstrated the importance and relationship of productivity and quality to human aspects associated with the software development process. Ortu et al. studied the effect of politeness [16] and emotions [15] on the time required to fix any given issue. The authors demonstrated that emotions did have an effect on the issue fixing time. Steinmacher et al. [22] analyzed social barriers that obstructed first contributions of newcomers (new developers joining an open-source project). The study indicated how impolite answers were considered as a barrier by newcomers. These barriers were identified through a systematic literature review, responses collected from open source project contributors and students contributing to open source projects. Rigby et al. [20] analyzed, using a psychometrically-based linguistic analysis tool, the five big personality traits of software developers in the Apache httpd server mailing list. The authors found that the two developers that were responsible for the major Apache releases had similar personalities and their personalities were different from other developers. Bazzelli et al. [1] analyzed questions and answers on stackoverflow.com to determine the developer personality traits, using the Linguistic Inquiry and Word Count [19]. The authors found that the top reputed authors were more extroverted and expressed less negative emotions than authors of down voted posts. Gomez et al. [9] performed an experiment to evaluate whether the level of extraversion in a team influenced the final quality of the software products obtained and the satisfaction perceived while this work was being carried out. Results indicated that when forming work teams, project managers should carry out a personality test in order to balance the amount of extraverted team members with those who are not extraverted. This would permit the team members to feel satisfied with the work carried out by the team without reducing the quality of the software products developed.

Compared to the existing literature, the goal of this paper is to build Markov chain models which describe how developers interact in a distributed Agile environment evaluating politeness, sentiment and emotions. Such models provide a mathematical view of the behavioural aspects among developers.

3 Experimental Setup

3.1 Dataset

We built our dataset from fifteen open-source, publicly available projects from a dataset proposed by Ortu et al. [18]. We selected the fifteen projects with the highest number of comments (from December 2002 to December 2013), from those projects which had a significant amount of activities in their agile kanban-

boards. The projects were developed following agile practices (mainly continuous delivery and use of kanban-boards). Table 1 shows summary project statistics.

Project	# of comments	# of developers
HBase	91016	951
Hadoop Common	61958	1243
Derby	52668	675
Lucene Core	50152	1107
Hadoop HDFS	42208	757
Cassandra	41966	1177
Solr	41695	1590
Hive	39002	850
Hadoop Map/Reduce	34793	875
Harmony	28619	316
OFBiz	25694	578
Infrastructure	25439	1362
Camel	24109	908
ZooKeeper	16672	495
Wicket	17449	1243

Table 1: Selected Project Statistics

3.2 Affective Metrics

Henceforward, we consider the term “affective metric” as a definition indicating all those measures linked to human aspects and obtained from text written by developers (i.e., comments posted on issue tracking systems). This study is based on the affective metrics (sentiment, politeness and emotions) used by Ortu et al. [15].

Sentiment. We measured sentiment using the SentiStrength¹ tool, which is able to estimate the degree of positive and negative sentiment in short texts, even for informal language. SentiStrength, by default, detects two sentiment polarizations:

- Negative: -1 (slightly negative) to -5 (extremely negative)
- Positive: 1 (slightly positive) to 5 (extremely positive)

The tool uses a lexicon approach based on a list of words to detect sentiment; SentiStrength was originally developed for the English language and was optimized for short social web texts. We used the tool to measure the sentiment of developers in issue comments.

¹ <http://sentistrength.wlv.ac.uk>

Politeness. To evaluate the level of politeness of comments related to a given issue, we used the tool developed by Danescu et al. [5]; the tool uses a machine learning approach and calculates the politeness of sentences providing, as a result, one of two possible labels: polite or impolite. The tool also provides a level of confidence related to the probability of a politeness class being assigned. We considered comments whose level of confidence was less than 0.5 as neutral (the text did not convey either politeness or impoliteness). For each comment we assigned a value according to the following rules:

- Value of +1 for comments marked as polite;
- Value of 0 for comments marked as neutral (confidence level < 0.5);
- Value of -1 for comments marked as impolite.

For each issue in our dataset, we built a temporal series of comments, and using the two tools we assigned a value of politeness and sentiment for each comment in the series. Next, for each issue, we calculated, starting from the first comment posted, the probability of having a polite/impolite/neutral following comment (for politeness), and a positive/neutral/negative comment (for sentiment). We thus calculated the probability of shifting from “polite” to “neutral” and *vice versa*; from “polite” to “impolite” and *vice versa*; finally, from “neutral” to “impolite” and *vice versa*.

Emotion. The presence of emotion in software engineering artifacts have been analysed by Murgia et al. [13]. Ortu et al. [15] provided a machine learning based approach for emotion detection in developers’ comments. We used the emotion detection tool provided by Ortu et al. [15] to detect the presence of *SADNESS*, *ANGER*, *JOY*, *LOVE* and *NEUTRAL*.

3.3 Affective Markov Chains

Markov Chains (MC) have been used to model behavioural aspects in social sciences [10], [21]. A Markov chain consists of K states and is a discrete-time stochastic process, a process that occurs in a series of time-steps in each of which a random choice is made.

We built a MC for each affective metric: sentiment, politeness and emotion. Figure 1 shows the steps in building the politeness MC as an example for an issue report in which three developers posted five comments. As a first step, we used the politeness tool [5] to label each comment as *POLITE*, *IMPOLITE* or *NEUTRAL*. Next we collected the politeness labels of the issue report, considering the set of labels as a politeness sequences of N-1 pair-wise politeness-transitions ([P,N,I,I,P] in the example), where N is the number of comments in the issue report.

In this example, the issue report has 4 transitions: polite-neutral, neutral-impolite, impolite-impolite and impolite-polite. Finally, we counted the frequency of each politeness-transition obtaining the corresponding MC. In our example, if we consider the *POLITE* state, we have two transition, P-P and

P-N; hence, the transition from *POLITE* to *IMPOLITE* state will have a probability of 0 and the transitions to *POLITE* and *IMPOLITE* state probability 0.5.

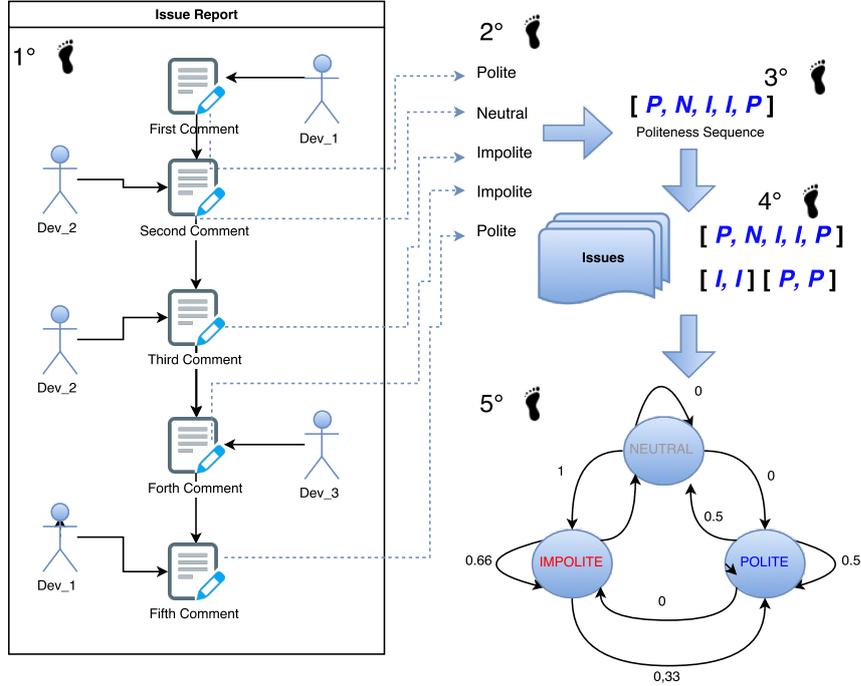


Fig. 1: Politeness' Markov's chain Schema

The MC for sentiment is built in a similar way to the politeness MC. The MC which models emotion transitions is slightly different; however, a comment can be polite, impolite or neutral when considering politeness, but it might contain more than one emotion. We used the emotion classifier proposed by Ortu et al. [15] to analyze each comment and to attribute to it: Anger, Sadness, Joy and/or Love. For example, if a comment is labeled as containing *ANGER* and *SADNESS* and the next labeled as containing no emotion (*NEUTRAL*), then we consider two transitions *ANGER-NEUTRAL* and *SADNESS-NEUTRAL*.

4 Results and discussion

4.1 Do developers change behaviour in the context of impolite/negative comments?

Motivation. Existing research has already explored links between productivity (as measured by issue fixing time) and discrete emotions, sentiment and politeness [13, 15]. The dynamic of an issue resolution involves complex interactions between different stakeholders such as users, developer and managers. A model able to describe such interactions could inform in the decision making process. The underlying assumption is that a model of social interaction can be used to understand the impact of a certain comment on the whole issue resolution discussion.

Approach. As presented in Sec. 3.3, we built three MCs for politeness, sentiment and emotions to understand how developers reacted to impolite/negative comments when they discuss an issue resolution.

Findings. Developers tended to answer to impolite/negative comments with a positive/negative comment with higher probability than impolite/negative comments.

Figure 2 shows the Politeness' MC describing the probability of changing from a state to another. The "neutral" state is quite stable. If a comment is classified as "neutral", communication flow among the developers involved tends to stay neutral, with a 73% probability. There is an 8% probability of a state-shift from "neutral" to "impolite" and a 19% probability of a state-shift from "neutral" to "polite". Starting from a "polite" state, the probability of shifting to the "impolite" state is quite low, 6%. There is a high probability of moving to the "neutral" state (61%). The probability of staying in the same state is 32%. Starting from an "impolite" state, the probability of moving to a "polite" state is 17%. This is higher than the probability of moving from a "polite" state to "impolite" and is an indication that a positive attitude could be more contagious than a negative attitude. It is interesting to see that the probability of staying in an "impolite" state is only 13% (far lower than the probabilities of staying in both "neutral" and "polite states), and that there is a 70% of probability of a shift from "impolite" to "neutral".

Figure 3 shows the Sentiment MC which describes the probability of changing from one state to another.

The "neutral" state in this case is also quite stable. If a comment is classified as "neutral", communication flow among developers tends to stay neutral, with a 60% probability. There is a 16% probability of a state-shift from "neutral" to "negative" and a 24% probability of a state-shift from "neutral" to "positive". Starting from a "positive" state, the probability of a shift to the "negative" state is 14%. The probability of a move to the "neutral" state is 55%. The probability of staying in the same state is 31%. From a "negative" state, the probability of moving to a "positive" state is 21%. In this case, the value is higher than the probability of moving from a "positive" state to a "negative"

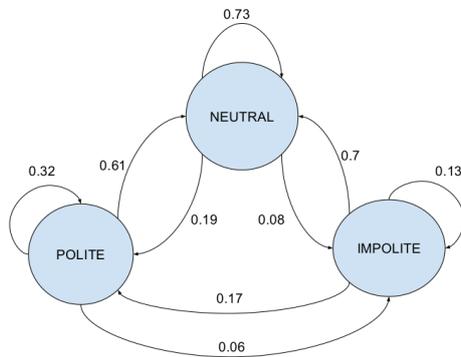


Fig. 2: Politeness MC

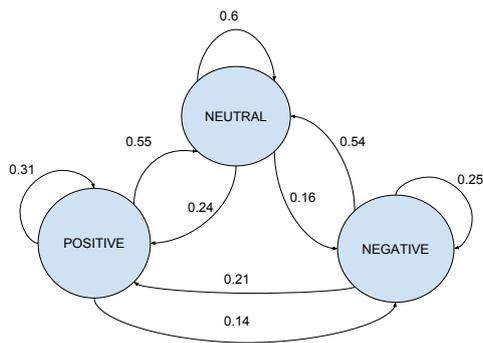


Fig. 3: Sentiment MC

one. The probability of staying in a “negative” state is 25% (also lower than the probabilities of staying in both “neutral” and “positive” states), and that there is a 54% probability to shift from “negative” to “neutral”.

4.2 What is the probability of shifting from comments holding positive emotions to comments holding negative emotion?

Motivation. The first research question showed how agile developers tended to respond more positively than negatively when considering politeness and sentiment. It is interesting to analyze if the same behaviours occur for emotions.

Approach. We built the MCs for emotions as presented in Sec. 3.3 to analyze the probabilities of shifting from an emotion to another when developers communicate.

Findings. Negative emotions such as *SADNESS* and *ANGER* tend to be followed by negative emotions more than positive emotion are followed by positive emotions. Table 2 shows the emotion transitions matrix. As for previous MCs, the numbers represent the probability of a comment containing emotion X being followed by a comment containing emotion Y (e.g., a comment expressing *SADNESS* has a probability of 0.26 of being followed by another *SADNESS* comment).

Table 2: Transition Matrix for Emotion MC

	SADNESS	ANGER	JOY	LOVE	NEUTRAL
SADNESS	26.11%	4.49%	7.88%	6.45%	55.08%
ANGER	13.79%	40.11%	5.61%	4.10%	36.39%
JOY	17.46%	4.43%	11.89%	12.22%	54.00%
LOVE	15.84%	3.84%	8.29%	15.59%	56.44%
NEUTRAL	16.42%	4.29%	7.64%	7.80%	63.85%

As confirmed by other studies [13], most of the comments expressing emotion are likely to be followed by *NEUTRAL* comments, with the exception of *ANGER*. Figure 4 is a graphical representation of the portion of Table 2 for the *ANGER* emotion showing it has probability of 0.4 of being followed by an *ANGER* comment against probability of 0.36 to be followed by a *NEUTRAL* comment. This represents an interesting finding which seems consistent with the common experience: negative emotions are more contagious than positive emotions.

5 Threats to validity

Several threats to validity need to be considered. Threats to external validity are related to generalisation of our conclusions. With regard to the system studied in this work, we considered only open-source systems and this could affect the generality of the study; our results are not meant to be representative of all environments or programming languages. Commercial software is typically developed using different platforms and technologies, with strict deadlines and cost limitations and by developers with different experience. Politeness, sentiment and emotions measures are approximations given the challenges of natural

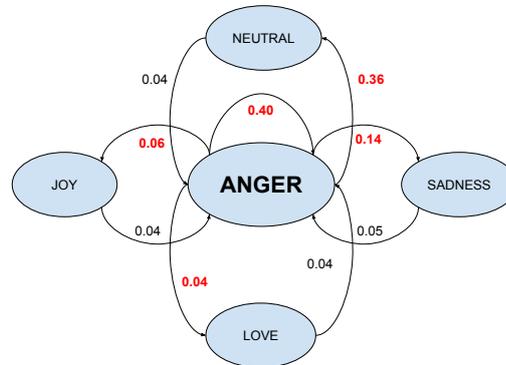


Fig. 4: Anger Markov chain. For simplicity only edges from/to ANGER are displayed

language and subtle phenomena like sarcasm. To deal with these threats, we used SentiStrength form measuring sentiment, Danescu et al.’s politeness tool [5] and Ortu et al. [15] for measuring politeness. This is a threat to construct validity. Threats to internal validity concern confounding factors that could influence the obtained results. Since the comments used in this study were collected over an extended period from developers unaware of being subject to analysis, we are confident that the emotions we mined are genuine. This study is focused on text written by agile developers for developers. To correctly depict the affectiveness embedded in such comments, it is necessary to understand the developers’ dictionary and slang. This assumption is supported by Murgia et al. [13] for measuring emotions. We are confident that the tools used for measuring sentiment and politeness however are equally reliable in the software engineering domain as in other domains.

6 Conclusions and future work

This paper presented an analysis of more than 500K comments from open-source issue tracking system repositories. We empirically determined how agile developers interacted with each other under certain psychological conditions generated by politeness, sentiment and emotions of a comment posted on a issue tracking system. Results showed that when in the presence of impolite or negative comments, there is higher probability for the next comment to be neutral or polite (neutral or positive in case of sentiment) than impolite or negative. This fact demonstrates that developers, in the dataset considered for this study, tended to resolve conflicts instead of increasing negativity within the communication

flow. This is not true when we consider emotions; negative emotions are more likely to be followed by negative emotions than positive. Markov models provide a mathematical description of developer behavioural aspects and the result could help managers take control the development phases of a system (especially in a distributed environment), since social aspects can seriously affect a developer's productivity. As future works we plan to investigate possible links existing between software metrics and emotions, to better understand the impact of affectiveness on software quality.

7 Acknowledgement

The research presented in this paper was partly funded by the Engineering and Physical Sciences Research Council (EPSRC) of the UK under grant ref: EP/M024083/1.

References

1. B. Bazelli, A. Hindle, and E. Stroulia. On the personality traits of stackoverflow users. In *Software Maintenance (ICSM), 2013 29th IEEE International Conference on*, pages 460–463. IEEE, 2013.
2. K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. Manifesto for agile software development. 2001.
3. A. P. Brief and H. M. Weiss. Organizational behavior: Affect in the workplace. *Annual review of psychology*, 53(1):279–307, 2002.
4. G. Concas, G. Destefanis, M. Marchesi, M. Ortu, and R. Tonelli. Micro patterns in agile software. In *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013, Proceedings*, volume 149, page 210. Springer, 2013.
5. C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, and C. Potts. A computational approach to politeness with application to social factors. In *Proceedings of ACL*, 2013.
6. G. Destefanis, R. Tonelli, E. Tempero, G. Concas, and M. Marchesi. Micro pattern fault-proneness. In *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, pages 302–306. IEEE, 2012.
7. A. Erez and A. M. Isen. The influence of positive affect on the components of expectancy motivation. *Journal of Applied Psychology*, 87(6):1055, 2002.
8. R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson. Towards individualized software engineering: empirical studies should collect psychometrics. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 49–52. ACM, 2008.
9. M. N. Gómez, S. T. Acuña, M. Genero, and J. A. Cruz-Lemus. How does the extraversion of software development teams influence team satisfaction and software quality?: A controlled experiment. *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*, 3(4):11–24, 2012.
10. M. I. Jordan. *Learning in Graphical Models:[proceedings of the NATO Advanced Study Institute: Ettore Mairona Center, Erice, Italy, September 27-October 7, 1996]*, volume 89. Springer Science & Business Media, 1998.

11. E. Kaluzniacky. *Managing psychological factors in information systems work: An orientation to emotional intelligence*. IGI Global, 2004.
12. W. Ke and P. Zhang. The effects of extrinsic motivations and satisfaction in open source software development. *Journal of the Association for Information Systems*, 11(12):784–808, 2010.
13. A. Murgia, P. Tourani, B. Adams, and M. Ortu. Do developers feel emotions? An exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th Working Conference on Mining Software Repositories, MSR 2014*, pages 262–271, New York, NY, USA, 2014. ACM.
14. N. Novielli, F. Calefato, and F. Lanubile. Towards discovering the role of emotions in stack overflow. In *Proceedings of the 6th International Workshop on Social Software Engineering*, pages 33–36. ACM, 2014.
15. M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. Are bullies more productive? Empirical study of affectiveness vs. issue fixing time. In *Proceedings of the 12th Working Conference on Mining Software Repositories, MSR 2015*, 2015.
16. M. Ortu, G. Destefanis, M. Kassab, S. Counsell, M. Marchesi, and R. Tonelli. Would you mind fixing this issue? an empirical analysis of politeness and attractiveness in software developed using agile boards. In *Agile Processes, in Software Engineering, and Extreme Programming*, pages 129–140. Springer, 2015.
17. M. Ortu, G. Destefanis, M. Kassab, and M. Marchesi. Measuring and understanding the effectiveness of Jira developers communities. In *Proceedings of the 6th International Workshop on Emerging Trends in Software Metrics, WETSoM 2015*, 2015.
18. M. Ortu, G. Destefanis, A. Murgia, M. Marchesi, R. Tonelli, and B. Adams. The Jira repository dataset: Understanding social aspects of software development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering*, page 1. ACM, 2015.
19. J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001, 2001.
20. P. C. Rigby and A. E. Hassan. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In *Proceedings of the Fourth International Workshop on Mining Software Repositories*, page 23. IEEE Computer Society, 2007.
21. T. A. Snijders. The statistical evaluation of social network dynamics. *Sociological methodology*, 31(1):361–395, 2001.
22. I. Steinmacher, T. U. Conte, M. Gerosa, and D. Redmiles. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*, pages 1–13, 2015.
23. S. Tan and P. Howard-Jones. Rude or polite: do personality and emotion in an artificial pedagogical agent affect task performance? In *2014 Global Conference on teaching and learning with technology (CTLT 2014) conference proceedings*, page 41, 2014.
24. P. Tourani, Y. Jiang, and B. Adams. Monitoring sentiment in open source mailing lists – exploratory study on the apache ecosystem. In *Proceedings of the 2014 Conference of the Center for Advanced Studies on Collaborative Research (CASCON)*, Toronto, ON, Canada, November 2014.
25. J. Tsay, L. Dabbish, and J. Herbsleb. Lets talk about it: Evaluating contributions through discussion in github. *FSE. ACM*, 2014.