Information and Software Technology xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

ARSENAL-GSD: A framework for trust estimation in virtual teams based on sentiment analysis

Guilherme Augusto Maldonado da Cruz*[a,b], Elisa Hatsue Moriya-Huzita[a], Valéria Delisandra Feltrim[a]

[a] State University of Maringá, Informatics Department, 5.790 Colombo Ave., Maringá-PR, Brazil
[b] 3258 Sophia Rasgulaeff Ave, Maringá-PR, Brazil

ARTICLE INFO

Keywords:
Trust
Versioning system
Sentiment analysis
Virtual teams
Global software development

ABSTRACT

Context: Technology advances has enabled the emergence of virtual teams. In these teams, people are in different places and possibly over different time zones, making use of computer mediated communication to interact. At the same time distribution brings benefits, it poses challenges as the difficulty to develop trust, which is essential for team efficiency.
Objective: In this paper, we present ARSENAL-GSD, an automatic framework for detecting trust among members of global software development teams based on sentiment analysis.
Methods: To design ARSENAL-GSD we made a literature review to identify trust evidences, especially those that could be captured or inferred from the automatic analysis of data generated by members' interactions in a versioning system. We applied a survey to validate the framework and evidences found.
Results: On a scale of 0–9, evidences were evaluated as having importance greater or equal to 5.23, and the extraction techniques used to estimate them were considered as good enough. Regarding differences between subjects profile, no difference was found in responses of participants with theoretical knowledge/none and those with medium/high knowledge in GSD, except for the evidence mimicry, which was considered more important for the group of participants with medium/high knowledge in GSD.
Conclusion: We concluded that our framework is valid and trust information provided by it could be used to allocate members to a new team and/or, to monitor them during project development.

1. Introduction

Software development using virtual teams characterizes distributed software development or global software development (GSD) when the distance between members comprises continents. It aims at providing benefits such as: low costs, proximity to the market, innovation and, access to skilled labor [1]. However, geographic distribution and cultural differences bring some challenges as well, mainly in communication, which depends mostly on computer mediated communication (CMC).

One of the challenges faced by virtual teams and therefore GSD is to generate and sustain trust among team members. There are several studies that show the importance of trust for GSD teams [2–6], the benefits of having high trust teams, and the drawbacks of the lack of trust among members. Trust between developers that are in different and distant locations facilitates collaboration [7], effective knowledge sharing, conflict resolution and teams integration [8]. Thus, trust impacts the teams efficiency, since high trust teams can achieve their goals with less effort than low trust teams. Conversely, the lack of trust brings additional challenges to the team. As reported by Calefato et al.[7], a low level of trust aggravates the feeling of being separated by different objectives, reduce the willingness to share information and cooperate for problem solving, and affects the goodwill with others.

So, in this context, information about trust among people can be used to suggest members to a team and/or to monitor the relationship among members, for example.

Some models have been proposed to estimate trust among people based on trust evidences, such as number of interactions, success of these interactions and similarity among people [9,10]. We consider trust evidence something that indicates the existence of trust or that happens when there is trust among people.

The information used by trust models can be extracted, for example,

* Corresponding author.
E-mail addresses: guilherme.maldonado@accion.com.br (G.A.M.d. Cruz), emhuzita@din.uem.br (E.H. Moriya-Huzita), valeria.feltrim@din.uem.br (V.D. Feltrim).
URLS: http://www.din.uem.br/~emhuzita/, http://www.din.uem.br/~emhuzita/ (E.H. Moriya-Huzita), http://www.din.uem.br/~vfeltrim, http://www.din.uem.br/~vfeltrim (V.D. Feltrim).

http://dx.doi.org/10.1016/j.infsof.2017.10.016
Received 14 December 2016; Received in revised form 24 October 2017; Accepted 27 October 2017
0950-5849/ © 2017 Elsevier B.V. All rights reserved.

Please cite this article as: Maldonado da Cruz, G.A., Information and Software Technology (2017), http://dx.doi.org/10.1016/j.infsof.2017.10.016

from social networks. In general, this information refers to the amount of interactions, evaluation of these interactions and their success. However, in a working environment people may not feel free to provide assessments of co-workers. Besides that, when the number of interactions is high, people may start to provide incorrect ratings, leading to incorrect trust estimation. Skopik et al. [9] developed a set of metrics to analyze the success of an interaction based on the fact that the bigger the amount of successful interactions the greater is the trust among people. These metrics eliminate the need for feedback, however, they are domain dependent and ignore subjectivity, one of the characteristics of trust, since they treat people as everyone thinking in the same way, which is not true.

In this paper we present a framework to estimate trust among members of GSD teams called ARSENAL-GSD. It extracts trust evidences observed in member interactions on versioning systems, without human intervention and using sentiment analysis.

In Sections 2–4 the concepts of GSD, trust and sentiment analysis, used in the development of the proposed framework are presented. Section 5 describes the framework ARSENAL-GSD. In Section 6 we evaluated the trust evidences and formulas used in the framework. Finally, Section 7 presents the conclusion and directions for future works.

2. Global software development

According to Sengupta et al.[11] GSD is a collaborative activity, which can be characterized by having: members from different cultures and organizations, separated by time and space using CMC to collaborate. This team organization aims at providing benefits, such as: reduced development costs, follow-the-sun development, modularization of labor, access to skilled labor, innovation, best practices and knowledge sharing and proximity to the market.

Despite its benefits, GSD also brings challenges that add to those already existing in virtual teams: strategic problems, cultural problems, inadequate communication, knowledge management, processes management and technical problems. Among the challenges in virtual teams, and therefore in GSD, is trust [12,13]. This challenge emerges due, among other things, to the lack of information transmitted by CMC [14]. Among other things, the loss of informal communication channels "can lead to a loss of trust between team members in different locations" [15].

In this context, trust is especially important for GSD teams, due to members' inability to check what other members are doing by just looking [16]. Thus, trust reduces the risk and cost of monitoring [5,17]. It also impacts in information sharing [5,6,17], cohesion [18], cooperation [17], coordination [14] and how people react to situations [19].

According to Treinen and Miller-Frost [15] building personal knowledge about the team and building mutual trust is more important than solving technical problems. This may be due to challenges in GSD that are related to communication (interaction between geographically distant people is more susceptible to misunderstandings), version management and configuration (geographically distributed teams can work continuously, in a follow-the-sun approach).

3. Trust

Trust has been studied in many fields, such as psychology, philosophy and economics. Based on definitions of different areas, Rusman et al. [20] defined trust as:

> a positive psychological state (cognitive and emotional) of a trustor (person who can trust/distrust) towards a trustee (person who can be trusted/distrusted) comprising of trustors positive expectations of the intentions and future behavior of the trustee, leading to a willingness to display trusting behavior in a specific context.

This definition presents one of the trust properties, which is context specificity. Trust is also dynamic, non-transitive, propagative, composable, subjective, asymmetrical, events sensitive and self-reinforcing [21]. Trust can also be defined in terms of two different dimensions, namely a cognitive dimension (cognitive trust) and an affective dimension (affective trust). As stated in many studies [22–26], cognitive trust relates to beliefs about others competence and reliability, while affective trust relates to emotional ties and reflects beliefs about reciprocated care and concerns. According to Al-Ani et al. [24], cognitive trust makes individuals more likely to take risks or, in other words, "trust and collaborate with one another", while affective trust "lead individuals to act in a way they feel is right". The aspects involved in both cognitive and affective dimensions will contribute to a person's decision to trust [23].

The trust we seek is the interpersonal trust, so, in this paper, we are interested specifically in the trust that one person has in another one, and which encompasses aspects of both cognitive and affective dimensions. For the sake of simplicity, from now on we will refer to it as trust. According to Rusman et al. [20] before deciding to trust, a person evaluates the trustworthiness of the person to be trusted and the risk involved, so that if she chooses to trust, she becomes vulnerable positively and negatively to the trusted person, assuming the risk. Thereby providing personal information of members positively influences the team's trust level, since it helps members to assess the trustworthiness of a particular member [20].

Since everyone assess trustworthiness before deciding to trust, the higher the trustworthiness, the higher the chance to be trusted. Trustworthiness antecedents are attributes used to evaluate the trustworthiness of a person. Rusman et al. [20] divided them into five categories: (i) communality, (ii) ability, (iii) benevolence, (iv) internalized norms and (v) accountability.

3.1. Trust evidence

Through a literature review we could not find an exact formula to determine whether there is trust among members of a team. However, some studies indicate behaviors and characteristics that can be used as evidence of trust existence. For example, Jarvenpaa et al. [16] conducted a qualitative study based on observations of three teams with a high level of trust and three teams with low level of trust. In this study they observed common characteristics to high-trust teams that did not appear in low-trust teams. These characteristics were enumerated as: proactivity, task oriented communication, positive tone, rotating leadership, task goal clarity, roles division, time management, feedback and intensive communication.

Besides Jarvenpaa et al.'s [16] work, we found other studies [3,4,12,16,20,27–31] that identified characteristics in teams which serve as evidence of trust among members. The list below sums up all trust evidences found in our literature review:

- Initiations and responses: initiations are defined as questions or statements that lead the receiver to provide a relevant response. Initiations and responses were used in Iacono and Weisband [27] to measure trust. In their study it was observed that high performance teams showed greater amount of initiations and responses than low performance ones.
- Motivation: According to Paul and He [28] motivation and trust are highly correlated, and when one increases the other also increases.
- Knowledge sharing: An experiment in Paul and He [28] showed that the greater the trust among participants, the greater is information sharing among them. Mitchell and Zigurs [32] also point out the impact of trust in knowledge transfer.
- Knowledge acceptance: People tend to accept knowledge of who they trust [4]. In Turek et al. [29] moving text of an author by another in a Wikipedia article was considered a sign of trust, because it expresses the tendency of an author to believe in the credibility of the other.

- Trustworthiness: trust and trustworthiness are highly correlated. When the trust in someone is high, the trustworthiness is high as well [16].
- Proactivity: in high-trust teams members are proactive, volunteering for roles and showing initiative [16].
- Task oriented communication: in high-trust teams most conversations are about tasks that must be performed by the team. Conversations related to social issues are rare [16,30].
- Positive tone: in high-trust teams members tend to show enthusiasm in conversation, praising and encouraging each other. In Jarvenpaa et al.'s [16] experiments, they noticed that discussions among members were so gently resolved that they almost went unnoticed.
- Task goal clarity: high-trust teams tend to discuss more their goals, and when in doubt, seek the coordinators for clarification instead of making assumptions [16].
- Rotating leadership: many members show leadership traits and according to the project needs, they assume the leadership as necessary [16].
- Role division: members assume roles, which does not mean that they are fully independent of each other. Members responsible for each role show the results of their work so others can provide feedback [16].
- Time management: high-trust team members discuss deadlines, establish milestones and care to fulfill them [16].
- Feedback and intense communication: high-trust teams display intense communication and provide feedback about team members' work [3,16,20,32].
- High performance: trust is positively related with cohesion [3], commitment, satisfaction and performance [32].
- Output quality: trust is positively related with quality of outputs [12].
- Common vocabulary: when there is trust between people they tend to share a common vocabulary in CMC [31].

Besides the evidences described above, Khan [12] considered authority delegation, enthusiasm and high quantum of work as signs of trust. According to Rusman et al. [20] resources sharing, task division and delegation also occur when there is trust among members. Table 1 presents a summary of all trust evidences found in our literature review and its supporting reference(s).

Table 1
Trust evidences found in our literature review.

Evidence	Supporting reference(s)
Initiations and responses	[27]
Motivation	[28]
Knowledge sharing	[28], [32]
Knowledge acceptance	[4], [29]
Trustworthiness	[16], [23]
Proactivity	[16]
Task oriented communication	[30], [16]
Positive tone	[16]
Task goal clarity	[16]
Rotating leadership	[16]
Role division	[16]
Time management	[16]
Feedback and intense communication	[16], [20], [32], [3]
High performance	[3], [32]
Output quality	[12]
Common vocabulary	[31]
Authority delegation	[12]
Enthusiasm	[12]
High quantum of work	[12]
Resources sharing	[20]
Task division	[20]
Delegation	[20]

3.2. Trust models

In our literature review we found two models to estimate trust among people. The framework proposed by Skopik et al. [9] aims at determining trust automatically, without the need for explicit feedbacks. The framework generates a graph in which nodes represent both services and people, and edges represent the trust value between them. Trust values are derived from the number of successful interactions relative to the total number of interactions. Successful interactions are computed by a set of metrics, such as occurrence of errors in services. Once the graph is generated, one can form teams and determine trust between nodes that have never interacted, among other functions.

The downside of Skopik et al.'s [9] work is that, by relying on metrics, it ignores the subjectivity that is intrinsic to trust by treating all people equally. For instance, if a service takes up to 30 seconds to respond an interaction, one person may consider it a success, while some other person may consider it a failure, even if it spends 10 seconds. Thus, this type of metric fails to capture such subjectivity.

The trust model proposed by Li et al. [10] aims at assisting users of E-commerce in choosing best sellers. In this work, trust is estimated based on interactions between users and, in the absence of interactions, on the similarity between assessments provided by users. It generates a user graph in which edges and weights represent the trust between them. The weight T of edges is calculated using Eq. (1).

$$T = \begin{cases} DT_{a \to b} & \text{if there is a direct edge} \\ SIM(u_a, u_b) & \text{if } SIM(u_a, u_b) > \Theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The formula proposed by Li et al. [10] has three cases: (i) it will use assessment from u_a to u_b weighted by the assessment time in order to calculate direct trust ($DT_{a \to b}$), (ii) in the absence of assessments from u_a to u_b, it will use the similarity between u_a and u_b's assessments ($SIM(u_a, u_b)$) using the Spearman coefficient, and (iii) if similarity is not greater than a threshold θ, trust value will be considered to be 0.

In addition to these models, a number of works that explore how to build trust (based on different trust models) can be found in the literature. Wang and Redmiles [33] discussed how to develop trust within a team using e-cheap-talk, which consists of starting conversations with subjects that have nothing to do with work, thus cultivating trust and establishing cooperation. They propose a model that uses simulation based on game theory and are also developing a tool that suggests strategies for interacting with unfamiliar collaborators. In Calefato et al. [7], the authors seek to quantify the effects that trust or other personal trait of the developer has on the projects that adopt the pull-request based model. It is argued that propensity to trust is an important aspect for building trust between people. Calefato and Lanubile [34] present the Social TFS tool based on the idea that informal information shared on social media can act as a substitute to social awareness earned during informal chats, thus helping to build trust among global team members. Al-Ani et al. [35] examined the potential for using tools to support the development of trust in global teams and facilitate contagion. In addition to communication tools, the authors mention the Trusty tool [36], which was specifically designed to support trust in distributed teams by providing mechanisms to support communication, coordination, exchange of knowledge, and generation of reports and statistical analysis of established social networks. Calefato and Lanubile [22] propose to measure affective trust through sentiment analysis in pull request comments. Based on the assumption that affective trust can be established through social communication, the authors aimed at assessing whether affective trust is a predictor of successful collaboration in distributed projects. For this, they propose the use of SentiStrength to compute the amount of affective trust between a pull-request contributor and the integrator.

ARTICLE IN PRESS

Table 2
Polarities' distribution in manual annotation.

Polarity	Number of comments	Percentage of comments
Positive	75	32.9
Neutral	86	37.7
Negative	67	29.4
Total	228	100.0

4. Sentiment analysis

Sentiment analysis and opinion mining is a broad and interdisciplinary research area that concerns the study of opinions, sentiments, attitudes, and emotions. It has gained a lot of attention by the research community in the last decade, "finding its application in almost every business and social domain" [37].

One of the tasks focused by sentiment analysis systems is to determine the polarity of the sentiment expressed for a given text: if positive, neutral or negative. The unity used in the analysis determines its level, which usually falls into one of the three: (i) document level, (ii) sentence level and (iii) entity-aspect level [37].

Sentiment analysis has also been applied in the context of software development research. In Guzman [38] sentiment analysis was used to capture emotion during software development phases and provides emotional climate awareness. Borbora et al. [39] considered that the sentiment expressed in communication is an indicator of the presence/absence of trust among stakeholders. Zhang et al. [40] suggested the use of sentiment analysis to get a better understanding of trust among users. In fact, one of the trust evidences presented in Section 3.1 is the positive tone in communication, and one way to detect it automatically is by using sentiment analysis tools.

Jongeling et al. [41] evaluated the performance of sentiment analysis tools in the field of software engineering research. The tools SentiStrength [42], NLTK [43], Alchemy API[1] and Standford NLP [44] were evaluated on a set of 392 comments, which had been manually annotated as part of Murgia et al.'s [45] work. The tools did not perform well, but among them NLTK and SentiStrength had the best results. The authors also noted that the choice of the sentiment analysis tool can lead to contradictory results, since besides the low performance in the software engineering domain, these tools do not agree to each other. In a more recent study, Jongeling et al. [46] confirmed these findings and suggested that there is "a need for sentiment analysis tools specially targeting the software engineering domain".

Many software engineering studies used SentiStrength as sentiment analysis tool [47–49], a fact also acknowledged by Jongeling et al. [46]. Nevertheless, before choosing a sentiment analysis tool to use in this work, we analyzed the performance of three publicly available tools: SentiStrength, Standford NLP and Alchemy API. It is worth noting, however, that our goal was not to carry out an extensive evaluation of these tools, but rather to compare their performance in the context of GitHub comments, thus helping in the selection of the sentiment analysis tool for this study.

We run the tools on a set of 228 GitHub pull requests comments that were manually classified according to three polarity values: positive, negative or neutral. Following Murgia et al. [45], we opted to perform classification at the comment level. Each comment was classified by two annotators, both with a software engineering background. They performed the classifications separately and according to their personal interpretation of the polarities. After a first annotation round, inter-annotator agreement measured by Cohen's *kappa* [50] was 0.46.

Although considered as moderate agreement [45,46], this kappa also indicates that there was a fair amount of disagreement. Further analysis of the annotation showed that annotators disagreed more often about positive-neutral or neutral-negative comments than about positive-negative. While one of them was stricter in classifying comments as positive or negative, the other one considered more subtle evidences of positivity or negativity in her classification. Also, there was no previous agreement about how to proceed in the classification of comments containing several sentences (possibly with different polarities) and annotators behaved differently in this case.

After the annotation experiment, disagreements were discussed and settled with the help of a third opinion. In the case of multi-sentence comments, they were classified as positive or negative if they contained at least one positive/negative sentence, even when they contained other neutral sentences. Table 2 shows the distribution of comments per polarity in the final manual annotation.

Table 3
Sentiment analysis tools' accuracy.

Tool	Accuracy
SentiStrength	0.535
Alchemy API	0.531
Standford NLP	0.351

Table 3 presents the general accuracy of each tool. As we can observe, SentiStrength and Alchemy API obtained the best results, with approximately 53.5% and 53.1% correct classifications, respectively. When compared to a simple baseline that generates predictions by respecting the sets category distribution, both outperformed it by approximately 16%. In Table 4, the tools performance are presented in terms of its precision, recall and F1 score per polarity. The lines Positive, Neutral, and Negative present the values for these categories. The line Total shows the averaged values of each measure for each tool. Considering the averaged F1 score, the tool with better results was SentiStrength, with approximately 0.54, followed by Alchemy API, with approximately 0.52. These results are consistent with the work of Jongeling et al. [41] and confirm the authors' claim that there is a need for sentiment analysis tools targeting this particular domain.

Based on these results and on the frequent use of SentiStrength in software engineering studies, we chose to use SentiStrength. However, it is worth pointing out that, conceptually, the framework is not bound to this specific tool and other sentiment analysis tools can be used instead, as described in Section 5.2.

5. ARSENAL-GSD

As previously discussed, virtual teams need trust among members in order to achieve their goals, since it is related with the team's efficiency [6].

Trust models can be used to estimate trust among team members. However, some models require users to provide evaluations of others, or that there are means of informing if interactions were positive or not. This can be a problem in the context of software development teams, since team members can not feel free to evaluate co-workers. Even if it were not an issue, there would be a lot of interactions and members

Table 4
Sentiment analysis tools' performance in terms of precision (P), recall (R) and F1 score.

	SentiStrength			Alchemy API			Standford NLP		
Pol	P	R	F1	P	R	F1	P	R	F1
Pos	0.606	0.533	0.567	0.571	0.640	0.604	0.714	0.067	0.122
Neu	0.580	0.465	0.516	0.535	0.267	0.357	0.500	0.140	0.218
Neg	0.452	0.627	0.525	0.495	0.746	0.595	0.320	0.940	0.477
Avg	0.546	0.542	0.536	0.534	0.551	0.519	0.511	0.382	0.272

[1] http://www.alchemyapi.com/products/alchemylanguage.

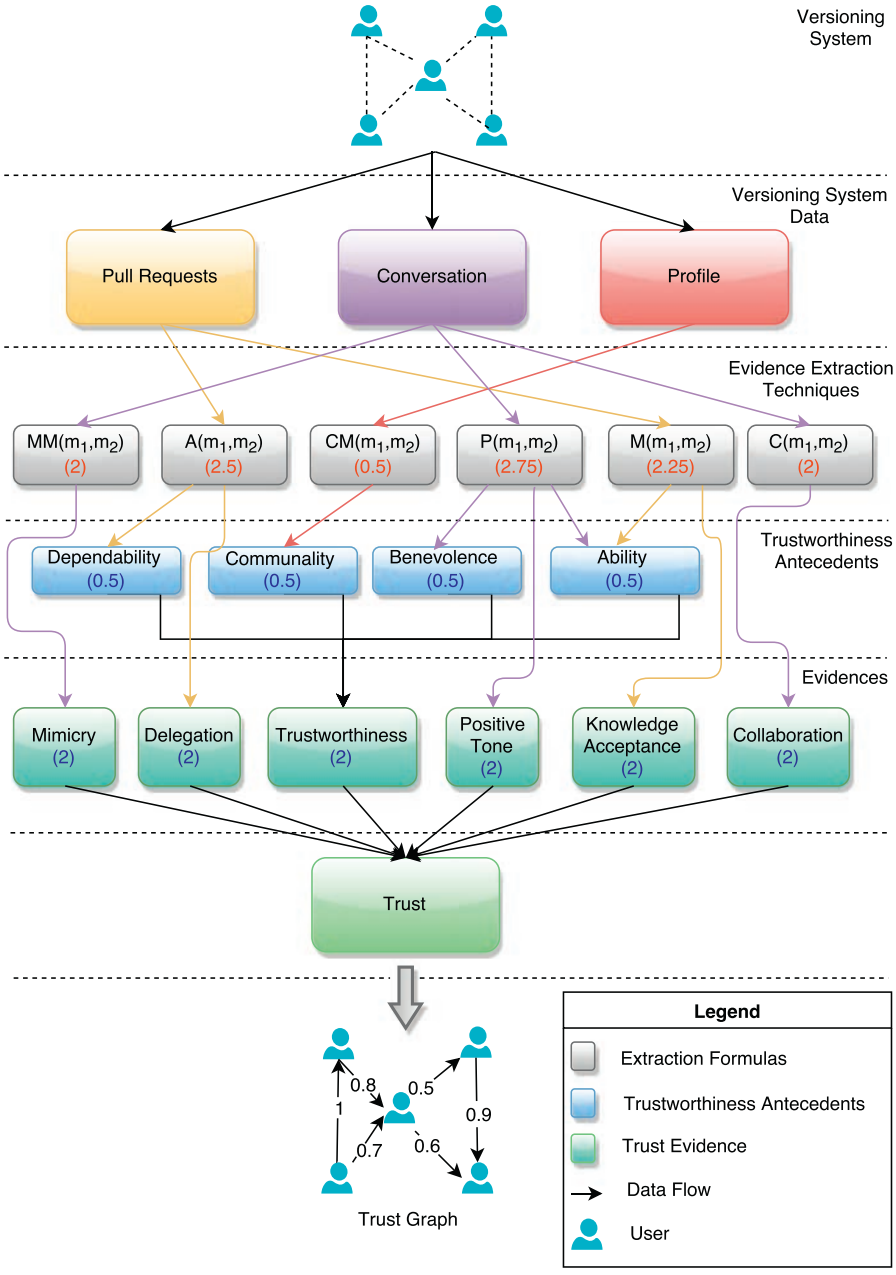


Fig. 1. Framework ARSENAL-GSD to estimate trust existence.

could end up getting tired of evaluating each interaction, providing nonsense evaluations that compromise the outcome of the models [10].

In this context, we propose a framework to estimate trust among GSD team members called ARSENAL-GSD (Automatic tRust eStimator based on sENtiment anALysis for GSD). The framework collects trust evidences observed in member interactions on versioning systems, without human intervention and using sentiment analysis. Fig. 1 displays a representation of the proposed framework in terms of its inputs, used techniques, trust evidences considered and output. This section describes ARSENAL-GSD, and it is divided in four subsections: characteristics, design and implementation, how it works, and example of use.

5.1. Characteristics

The main characteristics of ARSENAL-GSD are:

a) It uses versioning systems as data source. Versioning systems are tools heavily used in software development and therefore in GSD [51].

b) It uses trust evidences extracted from versioning system data (comments, commit state and user profile), to estimate trust among members of a project. We could not extract all the trust evidences described in Section 3.1, so the ones we consider in the framework are: mimicry (common vocabulary in Section 3.1), delegation, trustworthiness, positive tone, knowledge acceptance and collaboration. We extract the trustworthiness of a member by estimating four of the five trustworthiness antecedents presented in Rusman et al. [20], namely: dependability, communality, benevolence, and ability. We did not use the fifth antecedent, internalized norms, because there is no way to estimate it automatically from versioning systems data.

c) It is automatic. To set the framework, one has to inform the target project, extraction techniques that will be used, data extraction mechanism, size of temporal window, and update frequency. From that the framework retrieves data without human intervention,

estimates the existence of trust according to the temporal window and updates estimated values according to the update frequency.

d) It preserves the subjectivity inherent to trust by using sentiment analysis and considering mimicry as trust evidence, instead of a set of definite metrics. Sentiment analysis values and mimicry are extracted from how members write their comments, which is personal for each member. Thus, the framework takes into account subjectivity as it infers trust evidences from personal data. With sentiment analysis we can infer positive tone directly. Benevolence can also be inferred, since it was defined in Rusman et al. [20] as the positive attitude and courtesy displayed by the trustee. Ability can also be inferred from sentiment analysis, since the polarity of comments may be interpreted as feedback about the commit, and the commit in turn is the result of a member's ability to solve a problem. The idea is that the content of the pull request is a result of the members abilities, so positive comments indicate that a good job was done and therefore the member has the skills needed for that pull request. Thus, we use the polarity of comments as a feedback about the member's ability.

e) It updates evidences and trust values over time. It considers a time window to perform extractions, and from time to time it moves this window, discarding old data, retrieving new ones, and updating the values according to the data in the current time window.

f) It generates an initial graph of relations as exemplified in Fig. 2. This graph shows in which pull request team members interact. The initial graph has a `PR:*` edge between a pair of members if they interact in a pull request. During execution, we add partial and final edges to the initial graph of relations. Partial edges keep partial values used to calculate final edges. There is a final edge for each evidence extraction technique. For instance, in Fig. 3 the edge `PSentiment` keeps the polarity value for one comment and the edge `Sentiment` keeps the rate of positive comments.

g) It generates a trust graph with its estimative of trust existence for each pair of members that interacted in at least one pull request. In this graph, nodes represent members of a project and edges represent the existence of trust between them. The weight of the edge, ranging from 0 to 1, displays the probability that there is trust between two members. The closer to 1 the trust value is, the higher is the chance of existing trust between the members.

Comparing ARSENAL-GSD with the works presented in Section 3.2, we also use interactions like them both [9,10]. Unlike Li et al. [10], we removed the need for assessments, but kept the time factor by using a temporal window. We wanted it to be automatic like in Skopik et al. [9], but we did not use metrics. Instead, we used mimicry and sentiment analysis in order to preserve subjectivity. We also added more trust evidences found in the literature in order to enrich the information used to estimate trust.

5.2. Design and implementation

We designed and implemented an instance of ARSENAL-GSD to work with GitHub versioning system. This implementation is available under a Creative Commons License at https://github.com/Tulivick/ARSENAL_GSD. Fig. 4 presents the component diagram for ARSENAL-GSD. The framework is composed of four components:

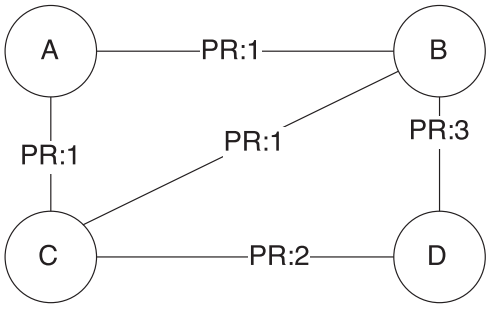


Fig. 2. Initial graph of relations.

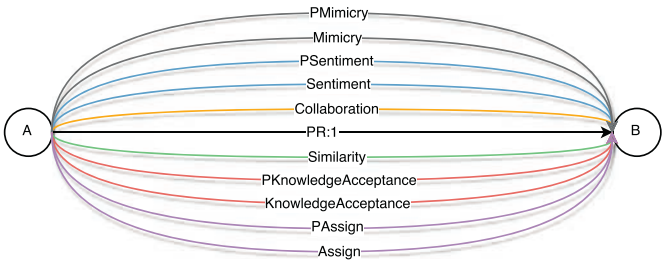


Fig. 3. Edges between nodes in a graph of relations.

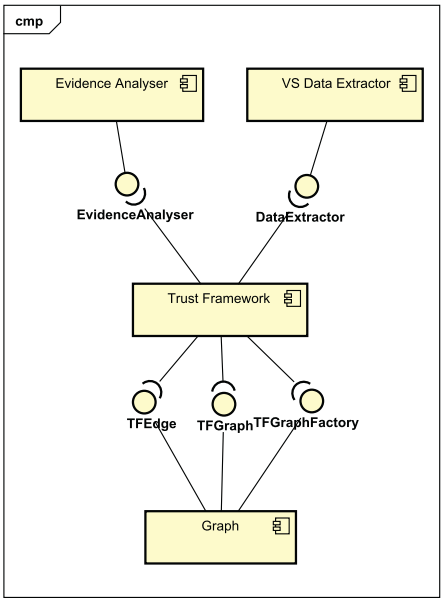


Fig. 4. Component diagram for trust framework.

`Graph` This component provides the framework with graphs that will be used as initial relations graph, relations graph and trust graph. By changing this component, we are able to alter how the framework keeps its data. The default implementation uses graphs provided by the JGraphT API[52].

`VS Data Extractor` This component extracts data from the versioning system. It extracts profile information from members in the project, pull requests information and conversations. In our implementation for GitHub we used the GitHub Java API,[2] a Java interface for GitHub API [53]. Besides extracting data, this component also generates the initial graph of relations.

`Evidence Analyzer` This component provides classes implementing the *EvidenceAnalyser* interface, each one representing an evidence extraction technique. Each of these classes will analyze data extracted from versioning system, and generates a value that is stored in the graph of relations and used to estimate trust existence. We provided six evidence extraction techniques: mimicry, assignments, communality, polarity, merges and collaboration. These evidence extraction techniques are formulas that will be described further in this section. To add more evidences to the framework it needs only a new implementation of *EvidenceAnalyser* interface to a new evidence extraction technique.

[2] https://github.com/eclipse/egit-github/tree/master/org.eclipse.egit.github.core.

`Trust framework` This is the main component. It provides ways to configure and use the framework. This component is responsible to get data from `VS Data Extractor`, and transmit it to `Evidence Analyzer` with the graph of relations, so it can be updated. From the graph of relations this component generates the trust graph using the formula described further in this section.

Note that by providing new implementations for `VS Data Extractor` we are able to extend the framework to other versioning systems. However, as each versioning system may have different data, the `Evidence Analyzer` is bound to the `VS Data Extractor` component. If one wants to provide support to another versioning system, it may be necessary to replace `Evidence Analyzer` by one that supports the new versioning system. It's also possible to extend the set of considered evidences, as well as to use other sentiment analysis tools, as mentioned in Section 4, by implementing other classes that extend the *EvidenceAnalyser* interface.

The Abstract Factory and Facade patterns were also used in this implementation. Abstract Factory was applied to the `Graph` component in order to create graphs. Facade in turn was used on the other components through their interfaces to make it simpler to use them by grouping methods to extract data, run evidence extraction techniques, and generate results in one single method.

As mentioned before, in order to extract evidence values, we used a set of formulas, that we named evidence extraction techniques on Fig. 1. To extract conversations mimicry, we calculate how similar the vocabularies used in a pull request conversations are. We calculated how similar two comments vocabularies are by using cosine similarity on word frequency. In Eq. (2), $SC(c_1, c_2)$ is the cosine similarity between two comments, and FV is the words frequency array for each comment.

$$SC(c_1, c_2) = \frac{FV_1 \cdot FV_2}{\|FV_1\| \|FV_2\|} \tag{2}$$

The member m_1 mimicry value in relation to member m_2, $MM(m_1, m_2)$, is the average of the similarities between m_1's comments and m_2's comments that precede in the same pull request. In Eq. (3), PR_{12} is the set of pull requests where m_1 and m_2 interact, C_1 is the comments set of m_1 in pull request pr_i and C_{2j} is the comments set of m_2 preceding the comment c_j.

$$MM(m_1, m_2) = \frac{\sum_{pr_i=1}^{|PR_{12}|} \sum_{c_j=1}^{|C_1|} \sum_{c_k=1}^{|C_{2j}|} SC(c_j, c_k)}{\sum_{pr_i=1}^{|PR_{12}|} \sum_{c_j=1}^{|C_1|} \sum_{c_k=1}^{|C_{2j}|} 1} \tag{3}$$

Communality is calculated by averaging the similarity of three GitHub user profile attributes: (I) followed users, (II) watched projects, and (III) location. (I) and (II) are calculated using Jaccard similarity (Eq. (4)) where C_1 and C_2 are evaluated sets, in this case, the followed users and watched projects for both members. (III) in turn uses a variant of the Euclidean distance [54, Eq. (5)] on Geert Hofsted index values [55]. Indexes of Geert Hofsted characterize the culture of a region using six indexes: power distance, individualism, masculinity, uncertainty avoidance, long-term guidance and indulgence. In Eq. (5), L_i is a location, K is the amount of indexes, I_{ik} is the value of the index k to a location L_i and V_k is the variance of the index k. If the indexes do not exist for a particular location, we will consider the biggest distance possible between two locations.

$$JS(C_1, C_2) = \frac{C_1 \cap C_2}{C_1 \cup C_2} \tag{4}$$

$$ED(L_1, L_2) = \sqrt{\sum_{k=1}^{K} \frac{(I_{1k} - I_{2k})^2}{V_k}} \tag{5}$$

Therefore, the communality $CM(m_1, m_2)$ between members m_1 and m_2 is given by Eq. (6), where F_i, W_i and L_i are respectively the set of followed users, watched projects and m_i's location.

$$CM(m_1, m_2) = \frac{JS(F_1, F_2) + JS(W_1, W_2) + \frac{1}{1 + ED(L_1, L_2)}}{3} \tag{6}$$

We used comments polarities to estimate benevolence, ability and positive tone. The polarity value between two members is given by Eq. (7), where $P(m_1, m_2)$ is member m_1 polarity value in relation to member m_2, C_{+12} is the amount of comments with positive polarity from m_1 to m_2, and C_{12} is the total amount of comments from m_1 to m_2. Comments from m_1 to m_2 are comments that m_1 wrote in m_2's pull request or comments where m_1 mentioned m_2. Note that there are pull requests created by m_1 where m_2 did not interact, however these are not considered.

$$P(m_1, m_2) = \frac{C_{+12}}{C_{12}} \tag{7}$$

Dependability and delegation can be observed through the assignments of a member by another in a pull request. The assignment value of a member m_1 to a member m_2 is 1 if m_1 assigned at least one pull request to m_2 or 0 otherwise. Eq. (8) calculates the assignment value for m_1 to m_2, $A(m_1, m_2)$, based on the amount of pull requests assigned to m_2 by m_1, PRs_{12}.

$$A(m_1, m_2) = \begin{cases} 1, & \text{if } PRs_{12} \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

We infer values for knowledge acceptance and ability of a member from the amount of pull requests that were merged. In Eq. (9), $M(m_1, m_2)$ is the proportion of pull requests created by m_2 in which m_1 and m_2 interacted, that were merged, PR_{a12} is the amount of pull request created by m_2 in which m_2 and m_1 interacted, that were merged, and PR_{12} is the amount of pull request created by m_2 in which m_1 and m_2 interacted.

$$M(m_1, m_2) = \frac{PR_{a12}}{PR_{12}} \tag{9}$$

We estimate collaboration as the proportion of interactions as given by Eq. (10). In this equation, $C(m_1, m_2)$ is the proportion of interactions between m_1 and m_2 out of the total interactions of m_1. I_{12} is the number of interactions between m_1 and m_2, and I_1 is the amount of m_1 interactions with everyone.

$$C(m_1, m_2) = \frac{I_{12}}{I_1} \tag{10}$$

Finally, we estimate values of trust among members using Eq. (11), which is the weighted average of the formulas listed above. The best way to set the weights α_i would be through the use of history data from previous projects to learn the weights. However, we are not aware of any database with trust information among members in versioning systems. Thus, we defined the weight of each formula (extraction technique) based on the number of trust evidences estimated through its use.

$$T(m_1, m_2) = \frac{\alpha_1 MM(m_1, m_2) + \alpha_2 CM(m_1, m_2) + \alpha_3 P(m_1, m_2)}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6} + \frac{\alpha_4 A(m_1, m_2) + \alpha_5 M(m_1, m_2) + \alpha_6 C(m_1, m_2)}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6} \tag{11}$$

We started by considering that every trust evidence (mimicry, delegation, trustworthiness, positive tone, knowledge acceptance, and collaboration) has the same weight 2. Then we propagate these weights backwards (considering the flow direction presented in Fig. 1) to trustworthiness antecedents and evidence extraction techniques. We chose weight 2 to avoid many decimal places when propagating it to the trustworthiness antecedents and evidence extraction techniques.

Trustworthiness is estimated using the values of the four trustworthiness antecedents (dependability, communality, benevolence, and ability) and we considered that all antecedents has the same weight in the calculation of trustworthiness. Thus, by propagating the weight 2

from trustworthiness to its four antecedents, we obtain a weight of 0.5 for each antecedent.

Finally, each extraction technique received a weight that is the sum of the weights of the trust evidences/trustworthiness antecedents estimated through its use. By propagating the weights from trustworthiness antecedents and other trust evidences (except trustworthiness, which is propagated through its antecedents), we obtained the values $\alpha_1 = 2$, $\alpha_2 = 0.5$,$\alpha_3 = 2.75$, $\alpha_4 = 2.5$,$\alpha_5 = 2.25$ and $\alpha_6 = 2$ presented in front of each formula on Fig. 1.

As an example, we will propagate the weights to $MM(m_1, m_2)$ and $A(m_1, m_2)$. $MM(m_1, m_2)$ infers only one evidence – mimicry, which has weight 2, thus when we propagate its weight to the formula, that gains weight 2 also. $A(m_1, m_2)$ in turn infers one evidence – delegation, which has weight 2, and one trustworthiness antecedent – dependability, which has weight 0.5. Thus, by propagating these two weights to the formula $A(m_1, m_2)$, it gains weight 2.5. By applying the same propagation process to each formula, we obtained the α_i values.

Therefore, in Fig. 1, the numbers in parentheses represent the weights of each evidence and trustworthiness antecedents, except the ones appearing at the evidence extraction techniques, which are the α values obtained by propagating the weights from evidences and trustworthiness antecedents.

As mentioned at the beginning of this subsection, each formula is coded in a class implementing the `EvidenceAnalyser` interface in the `EvidenceAnalyser` component. In order to implement these classes, we used the following APIs:

- Lucene [56]: it was used to generate the word frequency count necessary to calculate mimicry.
- Commons Math [57]: it provides the necessary methods to calculate cosine similarity.
- Sentistrength [58]: it was used as the sentiment analysis API. We used it to retrieve the polarity for every comment and to calculate the polarity value ($P(m_1, m_2)$).
- AlchemyApi [59]: this API provides many linguistic functionalities, including sentiment analysis. We used it to extract comments targets (people to whom the comment is directed) that were used to calculate polarity values. In 2015 AlchemyApi was acquired by IBM and is now called Watson Natural Language Understanding [60]
- Google Maps Geocoding API [61]: GitHub allows the user to provide its location freely, so some users provide more information (full address), others provide less (only country). We used Maps API to discover from which country each address was.

5.3. How it works

To start using the framework we need to configure it. To do it we inform a target project (owner/repository), the evidence extraction techniques to be used and their weights, the size of the temporal window, the update frequency, and a graph factory.

Once it starts running, the framework extracts project data from GitHub. We classify input data into three categories:

- Profile: it includes the projects in which the user participates, users followed, and user location;
- Conversation: it includes comments on pull requests;
- Pull Requests: it includes the state of the pull request, and users assigned to it.

Data from GitHub is in turn processed to extract the trust evidences presented in the Section 5.1. The data retrieved from GitHub is delivered to each instance of *EvidenceAnalyser* interface. This instances will add partial and final edges to the graph of relations.

We use profile data in order to extract a communality value by using Eq. (6).

By using Eqs. (3), (7) and (10) we extract mimicry, polarity and collaboration respectively from conversational data. The mimicry value is extracted calculating similarity between two comments. Polarity is obtained using a sentiment analysis tool on comments. The polarity value in turn is used to estimate positive tone, benevolence and ability. By analyzing member's participation in conversations, we calculate the collaboration value.

Eqs. (8) and (9) extract assignments and merge values respectively from pull requests data. We use the assignments value to infer delegation and dependability. Merge value in turn infer knowledge acceptance and ability, since a merge is the acceptance of someone's pull request, and the pull request is the result of one's ability.

Dependability, ability, communality and benevolence are not trust evidences, however, from them we can estimate trustworthiness, which is a trust evidence. The trustworthiness combined with mimicry, delegations, positive tone, knowledge acceptance and collaboration allow us to estimate trust existence among users through Eq. (11). This estimate is provided by means of a trust graph.

With the trust graph in hands, we can, for example, use trust values to suggest members to a team that has a bigger chance of having a high level of trust. In addition, as the framework process the latest comments and automatically updates the values of trust, it enables us to monitor trust among members, so that the manager can take actions when negative changes in the teams' trust are perceived.

5.4. Using ARSENAL-GSD: an example

Box 1 shows how to instantiate the framework. First, we instantiate a `TrustFramework` class (line 1). Then, we configure this instance informing the graphs factory, project data extractor (in the example we used GitHub), the repository's owner, the repository name and the time

```
tf = new TrustFramework();
...
tf.setFactory(new TFGraphImpFactory());
tf.setdExtractor(new GitHubExtractor(username, password));
tf.setOwner(owner);
tf.setRepository(repo);
tf.setTimeInterval(timeInteval);
tf.addEvidence(new ConversationMimicry(1));
...
TFGraph relGraph = tf.getRelationsGraph();
TFGraph trustGraph = tf.getTrustGraph();
```

Box 1. Instantiating the framework ARSENAL-GSD.

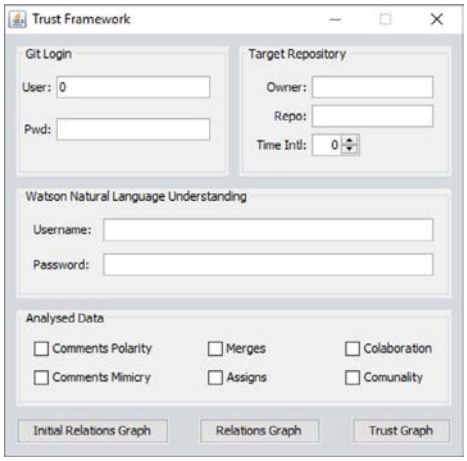


Fig. 5. Main screen.

spam to be considered (lines 3–7). Also as part of the configuration, we inform evidence extraction techniques and their weights. Line 8 adds the mimicry extraction technique with weight 1. After we configure the instance, we call the method *getRelationsGraph* (line 10) in order to obtain a graph of relations or *getTrustGraph* (line 11) for the trust graph.

To illustrate the use of ARSENAL-GSD, we developed a simple application that configures and generates graphs for a GitHub repository. Fig. 5 shows the main screen of this application. In this screen, we inform the name and password of a GitHub user, target project data (owner and repository name), the number of days to be considered for extraction and data analysis, and finally, which evidence extraction techniques that will be used in the analysis.

By clicking on the *Initial Relations Graph* button, the graph of relations in Fig. 6 is generated. It considers a period of 120 days and contains 4 members. Note that the user `loyalt` did not interact with anyone and therefore has no edge with other users. Clicking on the *Trust Graph* button, it generates the trust graph shown in Fig. 7. As `loyalt` did not interact with any user, it does not appear in the trust graph, since the trust estimated value to others would be zero.

5.4.1. Partial and final edges

In order to show how we used the evidence extraction techniques presented in Section 5.2, we will demonstrate how the partial and final edges were created considering the pull requests 135 and 138 (Figs. 8 and 9), in which there is an interaction between **cdzombak** and **dzlobin**. Fig. 10 illustrates the partial and final edges between **cdzombak** and **dzlobin** added to the graph of relations during the calculus of trust estimation. Bellow we explain how the values of each of these edges were calculated:

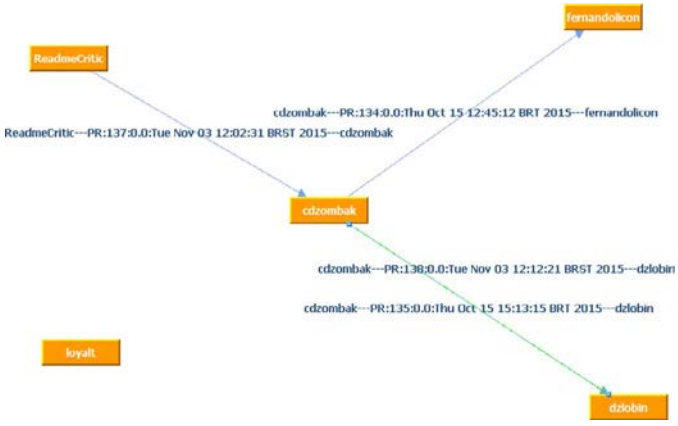


Fig. 6. Initial graph of relations.

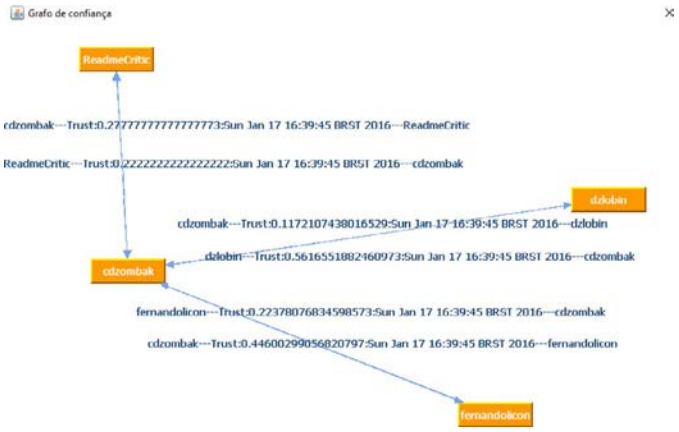


Fig. 7. Trust graph.

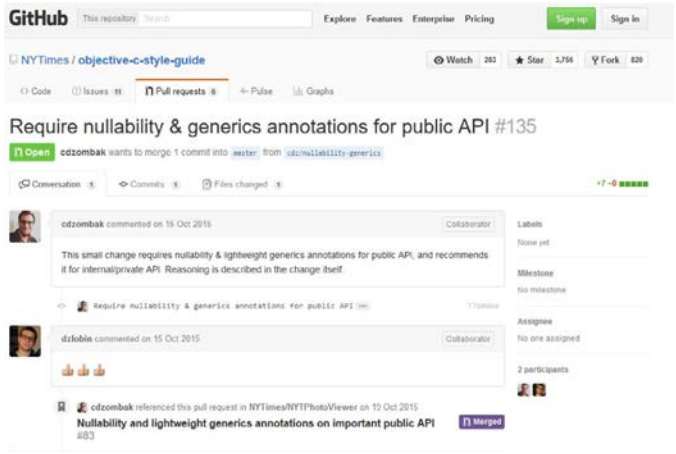


Fig. 8. Pull request 135.

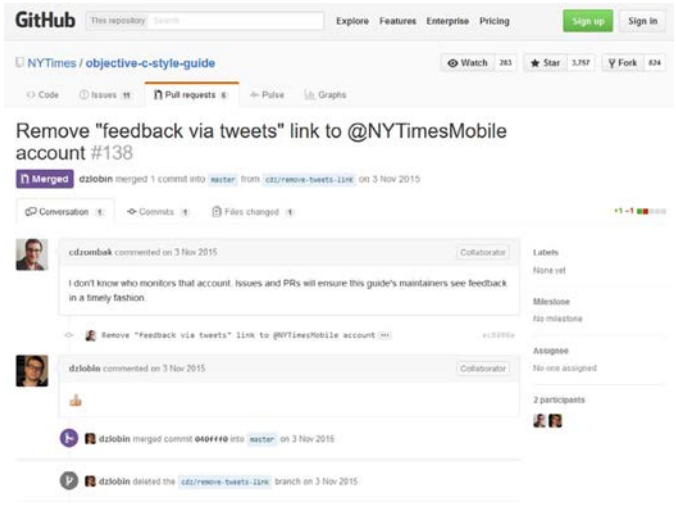


Fig. 9. Pull request 138.

Sentiment For each comment of **dzlobin** in pull request 135 and 138 we add a `PSentiment` edge having the weight 1, since the polarity found in these reviews are positive. The final edge `Sentiment` from **dzlobin** to **cdzombak** has weight 1, because it is the proportion of edges `PSentiment` weighing 1.

Similarity The value of the Similarity edge is the similarity between **cdzombak** and **dzlobin**, which was calculated based on location, watched projects and followed users.

Fig. 10. Edges between **cdzombak** and **dzlobin** in graph of relations.

Collaboration The value of `Collaboration` edges are calculated based on the initial graph of relations. Looking at Fig. 6, we note that **dzlobin** interacts in pull requests 135 and 138, in both cases with **cdzombak**, which takes the value 1 (100% of **dzlobin**'s interactions are with **cdzombak**). Since **cdzombak** interacts in pull requests 134, 135, 137 and 138, and in these four only two are with **dzlobin**, the edge `Collaboration` from **cdzombak** to **dzlobin** has value 0,5.

KnowledgeAcceptance Pull requests 135 and 138 were created by **cdzombak** but only 138 was merged, which creates a `PKnowledgeAcceptance` edge with value 1. The edge `KnowledgeAcceptance` from **cdzombak** to **dzlobin** has value 0,5 because half of the **cdzombak**'s pull requests in which **dzlobin** had interacted were merged.

Mimicry **dzlobin**'s comments in pull requests 135 and 138 have no word equal to the descriptions provided by **cdzombak**, so the two edges `PMimicry` from **cdzombak** to **dzlobin** are 0. As the `Mimicry` edge is the simple average of `PMimicry` edges, its value is also 0.

Assign There is no assignment in this example, so the edges `PAssign` and `Assign` were not added to the graph.

5.4.2. Using outputs

The framework ARSENAL-GSD provides two outputs: the graph of relations and the trust graph. With the graph of relations managers can analyze members who interact most on the team, those who are more involved in a project. Members with these characteristics have a great chance to possess a wider knowledge of the project and are ideal for training new members, and also to coordinate the team. In the case of the graph of relations in Fig. 6, we see that **cdzombak** would be this a member, since he is involved in all interactions and is a very active member compared to others.

The trust graph is the main output of the framework. Consider the following situation: the project manager should allocate two people to perform a new task. Which two members he should choose for the activity to be carried out as quickly as possible, but with quality? ARSENAL-GSD can help the manager to make that decision. During this work, we listed many benefits presented by teams high trust. As a matter of fact, several of these benefits were considered as evidence of trust existence, as the quality of outputs, cohesion, motivation. These characteristics are essential for GSD and ensure greater efficiency for the team, so the best action would be to allocate members who have a higher level of trust between them. Considering the graph in Fig. 7 and taking into account that all members have the necessary skills to perform the task at hand, the best action would be to allocate **cdzombak** and **dzlobin** (0.18 + 0.56 = 0.74) for this team.

The graph of relations and trust graph can also be used together to monitor the level of trust among team members. Suppose that in a new trust graph, generated 30 days after the trust graph in Fig. 7, the value of the edge from **dzlobin** to **cdzombak** fell to 0.42, a down 25%. The manager should find out what caused this drop in trust and assess what measures it should take to restore trust. The relationship graph can be used in conjunction to see which trust evidence estimative was affected.

6. The survey

Until this section we described the framework ARSENAL-GSD and how it can be used. Since we do not know of any versioning system dataset with trust annotations we considered that all trust evidences had the same weight in order to estimate trust, however this is not the best way to set up the weights. It would be ideal to set weights and evaluate framework's efficiency, that we executed some machine learning algorithm over an annotated dataset from which we could learn the best weight for each evidence extraction technique. Once we set up the weights, experiments could evaluate ARSENAL-GSD's efficiency by comparing its estimation with the annotations in the dataset. As it was not possible to carry out these tasks, we chose to perform a survey.

According to Kasunic [62], a survey is "a data-gathering and analysis approach in which respondents answer questions or respond to statements that were developed in advance", whose development comprises seven steps. When conducted properly, it is possible to generalize the results to a lot of people from a subset of them. The steps for our survey are detailed in the following sections.

6.1. Identify research objectives

This survey aimed to answer two questions:

- **Q1:** Considering that the weights of each evidence extraction technique have been defined empirically. Among the evidence presented in Fig. 1, how important each one is to estimate trust?
- **Q2:** Several formulas were created to extract evidences (evidence extraction techniques). Are these formulas suitable for the capture of these evidences?

6.2. Identify and characterize the target audience

Our target audience are people from software engineering area with a degree in a course in the area, preferably those with knowledge in GSD and versioning systems that work in industry.

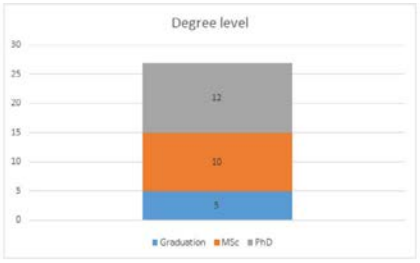
6.3. Design the sampling plan

The subject selection was by convenience. The questionnaire was distributed to known people who matched the target audience description for this survey. We also asked some of them, specifically the ones which were working as part of virtual teams, to forward the questionnaire to their co-workers. We personally distributed it to 53 people, and got 27 responses.

6.4. Design and write the questionnaire

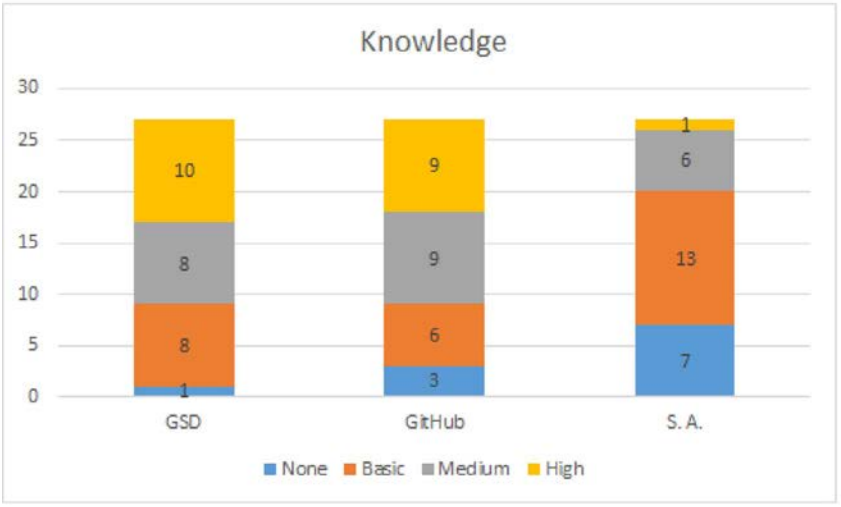
The questionnaire was developed using Google Forms[3] and following the guidelines of [62]. It was divided into four sections: (i) survey description, as well as the term of participation; (ii) the subjects' characterization with six questions about degree level, profession and

[3] https://www.google.com/forms/about/.



(a) Subjects' degree level

(b) Subjects' position



(c) Subjects' Experience

Fig. 11. Response frequencies - Subjects characterization.

knowledge about the issues addressed; (iii) evidences validation, with 10 questions regarding the importance of trustworthiness antecedents and evidences, and (iv) formulas validation, with 11 questions about the formulas used for evidence extraction. In addition to these issues, the sections (ii) and (iii) contained an open question for suggestions/complaints. In order to help subjects in answering the questions, a document describing the framework, the evidences considered, and the formulas was made available.

For future reference, the questionnaire questions were:

1. Subjects characterization
   (a) Degree level
   (b) What is your current company/institution?
   (c) What is your current position?
   (d) What is your knowledge and experience with global software development (GSD)?
   (e) What is your knowledge and experience with GitHub?
   (f) What is your knowledge and experience with Sentiment Analysis/Opinion Mining?
2. Evidences validation
   (a) For each evidence, how important is it for trust estimation (0-unimportant-9 very important)? (Question as table - 6 questions at all)
   (b) For each trustworthiness antecedent, how important is it for trustworthiness estimation (0-unimportant-9 very important)? (Question as table - 4 questions at all)
3. Formulas validation
   (a) For each evidence, is the formula used to calculate it valid and adequate? (Question as table - 6 questions at all)
   (b) For each trustworthiness antecedent, is the formula used to calculate it valid and adequate? (Question as table - 4 questions at all)
   (c) is the formula to estimate trust valid and adequate?

6.5. Pilot test the questionnaire

The pilot test was conducted with two PhDs, one from global software development area and one from natural language processing area. The issues raised were corrected.

6.6. Analyze the results and write a report

Fig. 11 presents charts with response frequencies for questions about subjects' characterization. From the subjects who answered the questionnaire, 5 are graduates, 10 are masters and 12 are PhDs, most of whom (14) are teachers/researchers in higher education institutions. Regarding the knowledge about this work domain, except for sentiment analysis, in which the vast majority answered none/basic, the subjects have a rate of 1 (none/basic):2 (medium/high) with few occurrences of none answers.

The chart in Fig. 12 present the frequency of responses regarding the importance of each evidence to estimate trust and trustworthiness antecedents to estimate trustworthiness, respectively. For these questions, we got varied answers. Some values have a higher frequency, but no chart presented a response with 48% frequency or more.

Finally, Fig. 13 present the answers about validity of evidence extraction, trustworthiness antecedents and trust formulas, respectively. Note that for all formulas, at least 74% of the subjects chose `Yeah, good enough` or `Yes, perfect`.

Table 5 presents descriptive statistics for the responses of Fig. 12. From this table we can answer the research question **Q1**, using the averages as weight for evidence and antecedents. This is the reason why we used a 10-point Likert scale, to facilitate the mapping of these averages to weights. These weights represent the importance of each

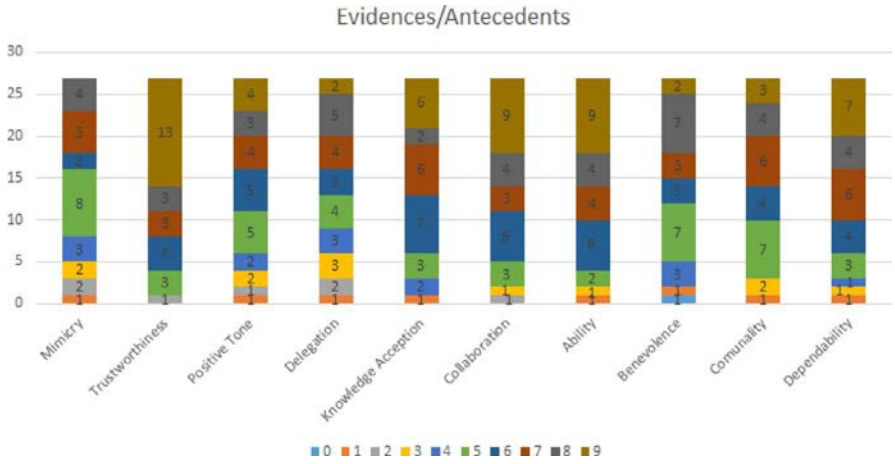


Fig. 12. Response frequencies - Evidences/Antecedents validation.

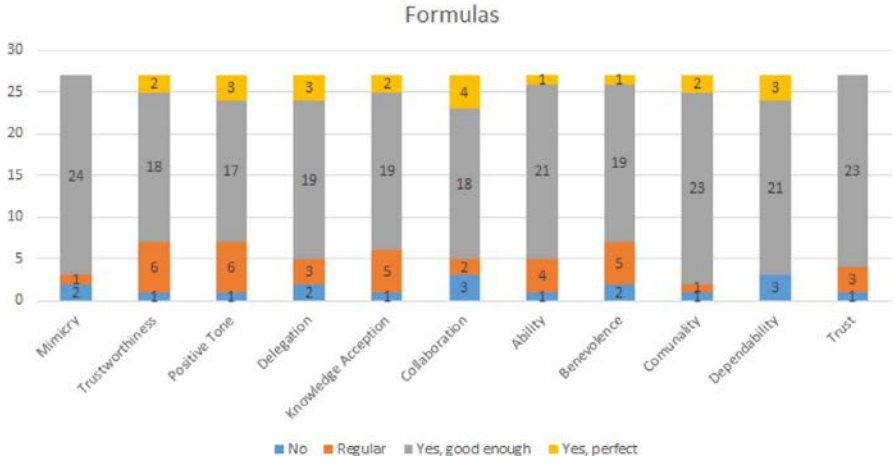


Fig. 13. Response frequencies - Formulas validation.

Table 5
Descriptive statistics - Evidences and trustworthiness antecedents.

Evidences/Antecedents	Mean	Variance	Standard deviation
Knowledge Acceptance	6.59	3.64	1.91
Collaboration	7.04	3.81	1.95
Trustworthiness	7.52	3.41	1.85
Delegation	5.56	5.26	2.29
Mimicry	5.23	3.89	1.97
Positive Tone	5.92	4.76	2.18
Ability	7.00	4.85	2.29
Dependability	6.81	4.16	2.20
Communality	6.18	3.77	1.94
Benevolence	5.96	4.96	2.23

evidence to estimate the trust and each antecedent to estimate someone's trustworthiness. Thus, applying the same procedure to propagate the weight from trust evidences and trustworthiness antecedents to evidence extraction techniques used in Section 5, we get the new values for α: $\alpha_1 = 5.23, \alpha_2 = 1.79, \alpha_3 = 9.68, \alpha_4 = 7.53, \alpha_5 = 8.62$ e $\alpha_6 = 7.04$. Fig. 14 presents ARSENAL-GSD with new α_{1-6} values.

With survey data we decided to examine whether there is a statistically significant difference between the importance in values of trust evidence and trustworthiness antecedents provided by participants with no knowledge or theoretical knowledge about GSD and participants with average or high knowledge. For this we used the statistical test T-test, if the data follows a normal distribution, or the Mann-Whitney test, if the data does not follow a normal distribution. To test whether the data follow a normal distribution or not we used the Shapiro–Wilk normality test that has two hypotheses:

Null hypothesis (H0): Data follows a normal distribution.
Alternative hypothesis (H1): Data does not follow a normal distribution.

In Table 6 we split subjects into two sets according to the knowledge and experience in GSD: None/Theoretical and Medium/High. For both sets and for each evidence, we applied the Shapiro–Wilk test to determine if the data follows a normal distribution. The test result, *p-value*, can be seen in the second and third column. When the value is less than .05, we reject the null hypothesis (**H0**) test, accepting the alternative hypothesis (**H1**). These values are presented in blue and red, meaning that the set follows a normal distribution or not respectively. For each evidence, when the two sets follow a normal distribution, we used a parametric test. This analysis is presented in the last column.

Based on results from Table 6, we chose to use *T-test* or Mann-Whitney test. For *T-test* the hypothesis are:

Null hypothesis (H0): There is no difference between means.
Alternative hypothesis (H1): The difference between means is different from 0.

To Mann-Whitney test hypothesis are:

Null hypothesis (H0): The location shift is equal to 0.
Alternative hypothesis (H1): The location shift is different from 0.

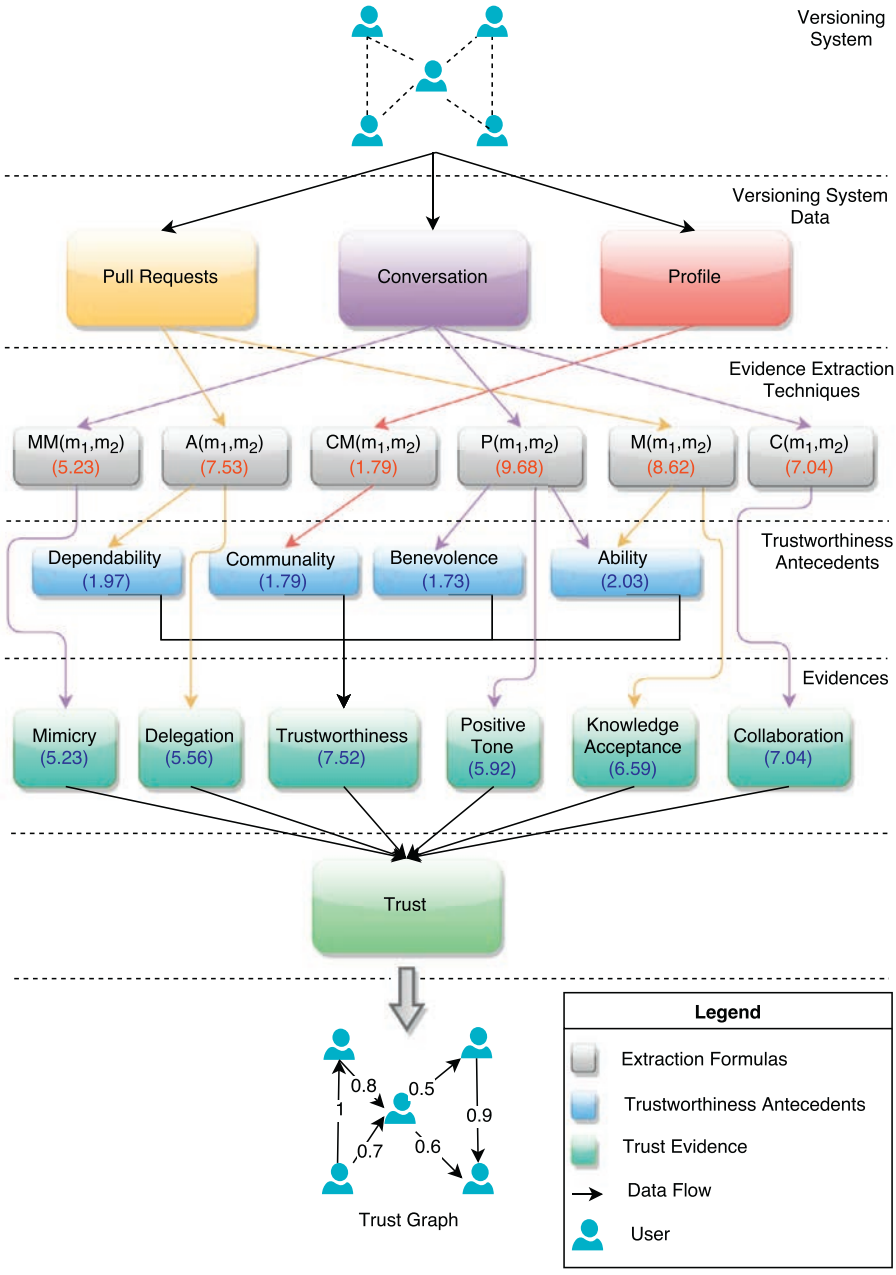


Fig. 14. *Framework* ARSENAL-GSD with adjusted weights.

Table 6
Results for Shapiro-Wilk normality test.

Evidences/Antecedents	None/Theoretical	Medium/High	Parametric?
Knowledge Acceptance	0.05477	0.1638	Yes
Collaboration	0.06571	0.02667	No
Trustworthiness	0.00069	0.00564	No
Delegation	0.32610	0.2009	Yes
Mimicry	0.40500	0.02577	No
Positive Tone	0.67240	0.42720	Yes
Ability	0.00735	0.05128	No
Dependability	0.00968	0.30070	No
Communality	0.66300	0.03576	No
Benevolence	0.5055	0.02114	No

Table 7
Test results.

Evidences/Antecedents	Test	Result
Knowledge Acceptance	T-test	0.90680
Collaboration	MannWhitney	0.9789
Trustworthiness	MannWhitney	0.45970
Delegation	T-test	0.16730
Mimicry	MannWhitney	0.01968
Positive Tone	T-test	0.39260
Ability	MannWhitney	0.57880
Dependability	MannWhitney	0.31980
Communality	MannWhitney	0.1574
Benevolence	MannWhitney	0.238

Table 7 presents the test used and its result for each trust evidence and trustworthiness antecedent. Note that all results, except for mimicry, were higher than 0.05, so we cannot reject the null hypothesis (**H0**). Therefore, we can conclude with 95% of confidence that these trust evidences and trustworthiness antecedents has the same importance independent of knowledge in GSD.

As we obtained a p-value of .01968, pointing that exists a difference based on the GSD knowledge of our subjects, we applied the test again

ARTICLE IN PRESS

Table 8
Descriptive statistics - Formulas.

Formulas	Median	Mode	Range
Knowledge Acceptance	2	2	3
Collaboration	2	2	3
Trustworthiness	2	2	3
Delegation	2	2	3
Mimicry	2	2	2
Positive Tone	2	2	3
Ability	2	2	3
Dependability	2	2	3
Communality	2	2	3
Benevolence	2	2	3
Trust	2	2	3

but with a different alternative hypothesis: *The location shift is less than 0* and obtained a p-value of .00984 telling that mimicry is more important for subjects with higher knowledge in GSD.

Based on Table 5 where the lowest average was 5.23, we observe that all these evidences are important for trust estimation. In an open question about possible trust evidences, subjects suggested:

- Number of defects in commits, and rework. The intuition behind the first suggestion is that the greater the number of defects generated by a person's commit, the less is this person's trustworthiness. In a similar way, the intuition on the second suggestion is that the more members have to redo the work done by a person, the less reliable she is. Both suggestions of evidence are not relevant since we use pull requests, they tend to be accepted only when they are correct (would not need someone to redo it) and not cause problems to the project (no defects). Thus, these suggestions are in some way covered by the knowledge acceptance evidence extracted analyzing merges.
- Clear and detailed feedback: the subject said that it supports the feeling of team and collaboration. It's a good suggestion however there is no way to determine if a feedback is clear and detailed automatically.
- How supportive is the person: the subject suggested to use the frequency of PR accepted in other projects. We believe that it could be a good evidence, and will analyse it in a future work.

To tabulate answers about formulas validity we considered 0 = `not`, 1 = `Regular`, 2 = `Yes, good enough`, and 3 = `Yes, perfect`. Table 8 presents descriptive statistics for the answers about formulas validity. We used a Likert scale in these answers, so there is no point in presenting the mean, variance and standard deviation as it will not have meaning. Therefore, the table shows the median (value that divides the data in half), mode (value that repeats the most) and range (difference between the highest and lowest value). As we can see all the formulas obtained median and mode equal to 2, then we can consider that the formulas were considered valid, thus answering the question `Q2`. All formulas got an amplitude of 3 because one subject did not fell capable of evaluating the formulas and answered no for all of them.

In a similar manner to what we did with the importance of the evidence and antecedents, we analyze whether there is a dependency between the level of knowledge/experience in GSD and acceptance and rejection of the formulas. For this we consider the answers `not` and `Regular` as rejection, and the answers `Yeah, good enough` and `Yes, perfect` as acceptance. In order to analyze this dependence, we used Fisher's exact test that has the following assumptions:

Null hypothesis (H0): Both variables (knowledge/experience and acception) are independent.
Alternative hypothesis (H1): Both variables are dependent.

Table 9 presents the distribution of responses grouped by accept/reject according to the knowledge/experience in GSD as well as the *p-value* obtained in Fisher's exact test. Since none of the formulas has *p-value* $< .05$, the null hypothesis (**H0**) cannot be rejected, and therefore, we consider that there is no dependence between knowledge/experience in GSD and accept/reject the formulas.

Table 9
Responses distribution grouped by knowledge in GSD as Accept/Reject together with Fisher's test result.

Formulas	None/Theoretical		Medium/High		Fisher
	Accept	Reject	Accept	Reject	
Knowledge Acceptance	8	1	13	5	0.6279
Collaboration	8	1	14	4	0.6361
Trustworthiness	7	2	13	5	1
Delegation	7	2	15	3	1
Mimicry	9	0	15	3	0.5292
Positive Tone	7	2	13	5	1
Ability	8	1	14	4	0.6361
Dependability	8	1	16	2	1
Communality	8	1	17	1	1
Benevolence	8	1	12	6	0.3632
Trust	8	1	15	3	1

At the end of the survey there was an open question where subjects could provide suggestions or complaints about the framework. Below we list the suggestions/complaints provided:

- *Knowledge acceptance has a relationship with dependability*: Some trust evidences and trustworthiness antecedents are related to each other, but these relations were not addressed in our work. This analysis is a suggestion for future work.
- *It is strange delegation to be binary*: In delegation we chose to analyze whether there is trust to delegate a task, a pull request. If someone delegate more pull requests for one person than for another means he gave more work to one than to another, this would be the evidence quantum of work. However, it is not possible to estimate this evidence since we do not know how much is a lot of work on the project context with the information we have.
- *Mimicry is not very significant for trust since as members spend more time working together they end up "speaking the same language"*: from our point of view these members come to "speak the same language" precisely because trust develops among members over time.
- *Using statistical TF-IDF to calculate mimicry*: one subject suggested that TF = IDF would be a more robust metric to calculate vocabulary similarity. This analysis is a suggestion for future work.
- *Study interference of more than two members in the formulas*: As our work focuses on trust between two members, there is no need to analyze the interference of other members. We are just interested in what a member think of the other.
- *Simplifying trust formula to take into account computations already performed*: ARSENAL-GSD takes into account the computations already performed, but does it through the relationships graph. The data that can be reused are kept in the graph of relation for the next trust estimation.
- *Mimicry would consider discourse*: however we did not find a way to consider the discourse when estimating mimicry.

6.7. Threats to validity

Below we list the threats to the validity of this work:

Threats to conclusion validity: the biggest threat is regarded with the number of subjects who answered our survey. We got only 27 subjects. However, the vast majority of subjects have knowledge in GSD, and 67% of them have practical knowledge. As we focus on trust, a concept that everyone knows, subjects' contributions were significant.

Threats to construct validity: the pilot test was conducted to ensure that the questionnaire and the document needed to answer it were appropriate. The suggestions obtained in the pilot test were analyzed and the problems found were corrected.

Threats to internal validity:

Difference between subjects: 33% of the subjects had no knowledge/experience in GSD or it was only theoretical, and the others had medium/high (practice) knowledge/experience in GSD. In order to analyze the effect of this difference in knowledge/experience, we applied tests to analyze whether there were differences in their responses.

Subjects response accuracy: subjects received a document explaining the framework, the evidences considered and formulas used on which were based the survey questions. However one of them said that it was difficult to follow the questionnaire with the document provided, the other subjects did not mention it.

Fatigue effect: we consider that a subject would take on average 30 min to read and answer the questionnaire, but to ensure that there was no problem with the fatigue, questionnaire was available online for a period of 30 days.

Knowledge about mathematical formulas: one subject did not feel capable of evaluating the formulas and another mentioned that he is not an expert and the formulas should be reviewed by some specialist in the subject.

Other important factors: it is likely that there was no influence between the participants since the questionnaire was available online and each participant responded it at a different time, probably at a different location, since various participating work in different locations, some in different states.

- **Threats to external validity:** preferably we tried to get participants that work or have worked with GSD, but as few participants responded to the invitation, we extended it to people who have knowledge even if only theoretical on GSD. As the main topic covered in the questionnaire is interpersonal trust, we believe that this difficulty has been toned down. Another threat was the subjects selection method that was by convenience.

7. Conclusions

The efficiency of a GSD team is directly tied to trust among team members. The higher is the trust, the lower is the project costs. Trust also increases communication and facilitate cooperation, coordination, knowledge and information sharing, which improve the quality of generated products.

Motivated by the importance of trust for these teams, we presented an automatic framework to estimate trust existence among members of a GSD team. It uses versioning systems, a collaborative tool used in software development, as a data source. To design the framework, we used some of the trust evidences presented in the literature that can be extracted from versioning systems data. One of the main features of the proposed framework is the use of sentiment analysis to extract some of these evidences, for example, the positive tone of the conversations.

The main contribution of this paper is in the mapping of trust evidences and elements of trust models that can be captured using sentiment analysis. We expect ARSENAL-GSD to provide a better estimative of trust existence than general automatic models in the literature since it uses sentiment analysis and a rich set of evidences. Employing sentiment analysis enables us to extract something unique for each person, thus adding subjectivity to our estimative, which is an important characteristic of trust. This subjectivity cannot be captured with the use of metrics, which are generally used in automatic models. GSD managers can benefit from ARSENAL-GSD to create teams with higher trust levels and to monitor trust level variations, so actions can be taken when the trust level decreases.

We also contributed with an implemented instance of the framework that works with GitHub and SentiStrength. The choice for SentiStrength was based on the performance analysis of three publicly available tools applied to a set of pull requests comments, which were manually annotated for this purpose. The annotation process and the observed disagreements between annotators evidenced the subjective nature of the polarity classification and the need for further research aiming at constructing a more consistent gold standard for this domain. We point out, however, that the evaluation of these tools was not the focus of this work, since the framework is not conceptually tied to a specific sentiment analysis tool.

We lament the lack of a trust annotated GitHub database, allowing us to validate ARSENAL-GSD against real data. As we do not have this, we considered initially all evidence weights the same. However, after we performed a survey to evaluate the importance of each trust evidence and trustworthiness antecedent, we observed that each of them have a different importance. With the results of the survey we suggested new weights for the evidences.

All evidence extraction techniques were considered by the subjects of the survey as good enough, but not perfect, and therefore it can be further improved. The suggestions and complaints provided by the participants were analyzed and those that can be applied will be considered in new versions and future work. As all evidences were considered relevant and their extraction techniques were considered adequate, although we did not evaluate the framework in practice, such evidence lead us to believe that the framework is adequate to estimate trust existence among members of GSD teams.

As future work, we foresee: (i) the addition of other trust evidences to the framework, (ii) to improve our evidence extraction techniques that are not perfect as we could see on the survey and to validate them with experts in mathematical formulas, (iii) to replicate the survey in order to corroborate our results, (iv) to investigate sentiment analysis methods and tools as well as aspects related to the construction of gold standards focusing on the software engineering domain, (v) to monitor a real project, allowing us to collect trust information about team members in order to compare it with the results given by ARSENAL-GSD, and (vi) to implement applications that use ARSENAL-GSD.

Acknowledgments

Funding: This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES).

References

[1] E. O'Conchuir, H. Holmstrom, P. Agerfalk, B. Fitzgerald, Exploring the assumed benefits of global software development, International Conference on Global Software Engineering, 2006. ICGSE '06, (2006), pp. 159–168, http://dx.doi.org/10.1109/ICGSE.2006.261229.

[2] K.V. Siakas, E. Siakas, The need for trust relationships to enable successful virtual team collaboration in software outsourcing, Int. J. Technol. Policy Manage. 8 (1) (2008) 59–75.

[3] F.-y. Kuo, C.-p. Yu, An exploratory study of trust dynamics in work-oriented virtual teams, J. Comput.-Mediated Commun. 14 (4) (2009) 823–854, http://dx.doi.org/10.1111/j.1083-6101.2009.01472.x.

[4] B. Al-Ani, H. Wilensky, D. Redmiles, E. Simmons, An understanding of the role of trust in knowledge seeking and acceptance practices in distributed development teams, 6th IEEE International Conference on Global Software Engineering (ICGSE), (2011), pp. 25–34, http://dx.doi.org/10.1109/ICGSE.2011.25.

[5] B. Al-Ani, M.J. Bietz, Y. Wang, E. Trainer, B. Koehne, S. Marczak, D. Redmiles, R. Prikladnicki, Globally distributed system developers: their trust expectations and processes, Proceedings of the conference on Computer supported cooperative work, ACM, 2013, pp. 563–574.

[6] F. Pangil, J. Chan, The mediating effect of knowledge sharing on the relationship between trust and virtual team effectiveness, J. Knowl. Manage. 18 (1) (2014) 92–106, http://dx.doi.org/10.1108/JKM-09-2013-0341.

[7] F. Calefato, F. Lanubile, N. Novielli, A preliminary analysis on the effects of propensity to trust in distributed software development, Proceedings of the 12th International Conference on Global Software Engineering, ICGSE '17, IEEE Press, Piscataway, NJ, USA, 2017, pp. 56–60, http://dx.doi.org/10.1109/ICGSE.2017.1.

[8] A.E. Akgn, H. Keskin, A.Y. Cebecioglu, D. Dogan, Antecedents and consequences of collective empathy in software development project teams, Inf. Manage. 52 (2) (2015) 247–259 https://doi.org/10.1016/j.im.2014.11.004.

[9] F. Skopik, H.L. Truong, S. Dustdar, Viete - enabling trust emergence in service-

oriented collaborative environments, Proceedings of the Fifth International Conference on Web Information Systems and Technologies WEBIST, (2009), pp. 471–478.
[10] J. Li, X. Zheng, Y. Wu, D. Chen, A computational trust model in c2c e-commerce environment, Proceedings - IEEE International Conference on E-Business Engineering, ICEBE, Shanghai, China, (2010), pp. 244–249.
[11] B. Sengupta, S. Chandra, V. Sinha, A research agenda for distributed software development, Proceedings of the 28th international conference on Software engineering, ACM, 2006, pp. 731–740.
[12] M.S. Khan, Role of trust and relationships in geographically distributed teams: exploratory study on development sector, Int. J. Netw. Virtual Organ. 10 (1) (2012) 40–58.
[13] V. Gomes, S. Marczak, Problems? we all know we have them. do we have solutions too? a literature review on problems and their solutions in global software development, IEEE Seventh International Conference on Global Software Engineering (ICGSE), (2012), pp. 154–158, http://dx.doi.org/10.1109/ICGSE.2012.43.
[14] V. Pearroja, V. Orengo, A. Zornoza, A. HernÃ!'ndez, The effects of virtuality level on task-related collaborative behaviors: the mediating role of team trust, Comput. Human Behav. 29 (3) (2013) 967–974, http://dx.doi.org/10.1016/j.chb.2012.12.020.
[15] J.J. Treinen, S.L. Miller-Frost, Following the sun: case studies in global software development, IBM Syst. J. 45 (4) (2006) 773–783, http://dx.doi.org/10.1147/sj.454.0773.
[16] S.L. Jarvenpaa, K. Knoll, D.E. Leidner, Is anybody out there? antecedents of trust in global virtual teams, J. Manage. Inf. Syst. 14 (4) (1998) pp.29–64.
[17] L. Striukova, T. Rayna, The role of social capital in virtual teams and organisations: corporate value creation, Int. J. Netw. Virtual Organ. 5 (1) (2008) 103–119.
[18] S. Yusof, N. Zakaria, Exploring the state of discipline on the formation of swift trust within global virtual teams, 45th Hawaii International Conference on System Science (HICSS), (2012), pp. 475–482, http://dx.doi.org/10.1109/HICSS.2012.272.
[19] S.L. Jarvenpaa, T.R. Shaw, D.S. Staples, Toward contextualized theories of trust: the role of trust in global virtual teams, Inf. Syst. Res. 15 (3) (2004) 250–267, http://dx.doi.org/10.1287/isre.1040.0028.
[20] E. Rusman, J. van Bruggen, P. Sloep, R. Koper, Fostering trust in virtual project teams: Towards a design framework grounded in a TrustWorthiness ANtecedents (TWAN) schema, Int. J. Hum. Comput. Stud. 68 (11) (2010) 834–850, http://dx.doi.org/10.1016/j.ijhcs.2010.07.003.
[21] W. Sherchan, S. Nepal, C. Paris, A survey of trust in social networks, ACM Comput. Surv. (CSUR) 45 (4) (2013) 47:1–47:33, http://dx.doi.org/10.1145/2501654.2501661.
[22] F. Calefato, F. Lanubile, Affective trust as a predictor of successful collaboration in distributed software projects, 2016 IEEE/ACM 1st International Workshop on Emotional Awareness in Software Engineering (SEmotion), (2016), pp. 3–5, http://dx.doi.org/10.1109/SEmotion.2016.010.
[23] F. Calefato, F. Lanubile, N. Novielli, The role of social media in affective trust building in customer–supplier relationships, Electron. Commerce Res. 15 (4) (2015) 453–482, http://dx.doi.org/10.1007/s10660-015-9194-3.
[24] B. Al-Ani, S. Marczak, D. Redmiles, R. Prikladnicki, Facilitating contagion trust through tools in global systems engineering teams, Inf. Softw. Technol. 56 (3) (2014) 309–320, http://dx.doi.org/10.1016/j.infsof.2013.11.001.
[25] S. Marczak, B. Al-Ani, D. Redmiles, R. Prikladnicki, The interplay among trust, risk, and reliance in global systems engineering teams, 2014 IEEE 9th International Conference on Global Software Engineering, (2014), pp. 46–55, http://dx.doi.org/10.1109/ICGSE.2014.29.
[26] J.M. Wilson, S.G. Straus, B. McEvily, All in due time: The development of trust in computer-mediated and face-to-face teams, Organ. Behav. Hum. Decis. Process. 99 (1) (2006) 16–33 https://doi.org/10.1016/j.obhdp.2005.08.001.
[27] C. Iacono, S. Weisband, Developing trust in virtual teams, Proceedings of the Thirtieth Hawaii International Conference on System Sciences, 1997, 2 (1997), pp. 412–420 vol.2, http://dx.doi.org/10.1109/HICSS.1997.665615.
[28] S. Paul, F. He, Time pressure, cultural diversity, psychological factors, and information sharing in short duration virtual teams, 45th Hawaii International Conference on System Science (HICSS), (2012), pp. 149–158, http://dx.doi.org/10.1109/HICSS.2012.593.
[29] P. Turek, A. Wierzbicki, R. Nielek, A. Hupa, A. Datta, Learning about the quality of teamwork from wikiteams, IEEE Second International Conference on Social Computing (SocialCom), (2010), pp. 17–24, http://dx.doi.org/10.1109/SocialCom.2010.13.
[30] R. Rico, C.-M. Alcover, M. Sanchez-Manzanares, F. Gil, The joint relationships of communication behaviors and task interdependence on trust building and change in virtual project teams, Social Sci. Inf. 48 (2) (2009) 229–255, http://dx.doi.org/10.1177/0539018409102410.
[31] L.E. Scissors, A.J. Gill, D. Gergle, Linguistic mimicry and trust in text-based cmc, Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW '08, ACM, New York, NY, USA, 2008, pp. 277–280, http://dx.doi.org/10.1145/1460563.1460608.
[32] A. Mitchell, I. Zigurs, Trust in virtual teams: solved or still a mystery? ACM SIGMIS Database 40 (3) (2009) 61–83, http://dx.doi.org/10.1145/1592401.1592407.
[33] Y. Wang, D. Redmiles, Cheap talk, cooperation, and trust in global software engineering, Empirical Softw. Eng. 21 (6) (2016) 2233–2267, http://dx.doi.org/10.1007/s10664-015-9407-3.
[34] F. Calefato, F. Lanubile, Augmenting social awareness in a collaborative development environment, 2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE), (2012), pp. 12–14, http://dx.doi.org/10.1109/CHASE.2012.6223009.
[35] B. Al-Ani, S. Marczak, D. Redmiles, R. Prikladnicki, Facilitating contagion trust through tools in global systems engineering teams, Inf. Softw. Technol. 56 (3) (2014) 309–320 https://doi.org/10.1016/j.infsof.2013.11.001.
[36] G.N. Aranda, A. Vizcaíno, J.L. Hernández, R.R. Palacio, A.L. Morán, Trusty: A Tool to Improve Communication and Collaboration in DSD, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 224–231. 10.1007/978-3-642-23801-7_18.
[37] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Human Lang. Technol. 5 (1) (2012) 1–167, http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016.
[38] E. Guzman, Visualizing emotions in software development projects, First IEEE Working Conference on Software Visualization (VISSOFT), (2013), pp. 1–4, http://dx.doi.org/10.1109/VISSOFT.2013.6650529.
[39] Z.H. Borbora, M.A. Ahmad, J. Oh, K.Z. Haigh, J. Srivastava, Z. Wen, Robust features of trust in social networks, Soc. Netw. Anal. Min. 3 (4) (2013) 981–999, http://dx.doi.org/10.1007/s13278-013-0136-6.
[40] Y. Zhang, H.-J. Chen, X.-H. Jiang, H. Sheng, Z.-H. Wu, RCCtrust: A combined trust model for electronic community, J. Comput. Sci. Technol. 24 (5) (2009) 883–892, http://dx.doi.org/10.1007/s11390-009-9279-3.
[41] R. Jongeling, S. Datta, A. Serebrenik, Choosing your weapons: On sentiment analysis tools for software engineering research, Proc. ICSME, IEEE, 2015, pp. 531–535.
[42] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment strength detection in short informal text, J. Am. Soc. Inf. Sci. Technol. 61 (12) (2010) 2544–2558, http://dx.doi.org/10.1002/asi.21416.
[43] S. Bird, E. Klein, E. Loper, Natural language processing with python, "O'Reilly Media, Inc.", 2009.
[44] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the conference on empirical methods in natural language processing (EMNLP), 1631 Citeseer, 2013, p. 1642.
[45] A. Murgia, P. Tourani, B. Adams, M. Ortu, Do developers feel emotions? an exploratory analysis of emotions in software artifacts, Proceedings of the 11th Working Conference on Mining Software Repositories, MSR '14, ACM, New York, NY, USA, 2014, pp. 262–271.
[46] R. Jongeling, P. Sarkar, S. Datta, A. Serebrenik, On negative results when using sentiment analysis tools for software engineering research, Empirical Softw. Eng. (2017) 1–42, http://dx.doi.org/10.1007/s10664-016-9493-x.
[47] E. Guzman, D. Azócar, Y. Li, Sentiment analysis of commit comments in github: An empirical study, Proceedings of the 11th Working Conference on Mining Software Repositories, MSR '14, ACM, New York, NY, USA, 2014, pp. 352–355.
[48] P. Tourani, Y. Jiang, B. Adams, Monitoring sentiment in open source mailing lists: Exploratory study on the apache ecosystem, Proceedings of 24th Annual International Conference on Computer Science and Software ngineering, CASCON '14, IBM Corp., Riverton, NJ, USA, 2014, pp. 34–44.
[49] V. Sinha, A. Lazar, B. Sharif, Analyzing developer sentiment in commit logs, Proceedings of the 13th International Conference on Mining Software Repositories, MSR '16, ACM, New York, NY, USA, 2016, pp. 520–523, http://dx.doi.org/10.1145/2901739.2903501.
[50] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46, http://dx.doi.org/10.1177/001316446002000104.
[51] R. Robbes, M. Lanza, Versioning systems for evolution research, Eighth International Workshop on Principles of Software Evolution, (2005), pp. 155–164, http://dx.doi.org/10.1109/IWPSE.2005.32.
[52] B. Naveh, Contributors, Welcome to jgrapht - a free java graph library, 2015, (http://jgrapht.org/). Accessed: 2015-11-12.
[53] GitHub, Github api v3, 2015, (https://developer.github.com/v3/). Accessed: 2015-11-12.
[54] O. Dodd, B. Frijns, A. Gilbert, On the role of cultural distance in the decision to cross-list, Eur. Financial Manage. (2013) n/a–n/a, http://dx.doi.org/10.1111/j.1468-036X.2013.12038.x.
[55] G. Hofstede, G.J. Hofstede, M. Minkov, Cultures and Organizations: Software of the Mind, 3 ed., McGraw-Hill USA, New York, 2010. An optional note
[56] Apache, Apache lucene - welcome to apache lucene, 2015a, (https://lucene.apache.org/a). Accessed: 2015-11-05.
[57] Apache, Math commons math: The apache commons mathematics library, 2015b, (http://commons.apache.org/proper/commons-math/index.htmlb). Accessed: 2015-11-05.
[58] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment in short strength detection informal text, J. Am. Soc. Inf. Sci. Technol. 61 (12) (2010) 2544–2558, http://dx.doi.org/10.1002/asi.v61:12.
[59] IBM, Alchemy api | powering the new ai economy, 2015, (http://www.alchemyapi.com/). Accessed: 2015-11-05.
[60] IBM, Watson natural language understanding, 2017, (https://www.ibm.com/watson/services/natural-language-understanding/). Accessed: 2017-10-15.
[61] Google, The google maps geocoding api | google maps geocoding api | google developers, 2015, (https://developers.google.com/maps/documentation/geocoding/intro). Accessed: 2015-11-05.
[62] M. Kasunic, Designing an Effective Survey, Technical Report, CMU/SEI-2005-HB-004, Software Engineering Institute, Carnegie Mellon University, 2005.