City on the River: Visualizing Temporal Collaboration

Jessica Perrie
University of Toronto
jssc.perrie@gmail.com

Jing Xie
University of Toronto
jingyi.xie@mail.utoronto.ca

Maleknaz Nayebi
Polytechnique Montréal
maleknaz.nayebi@polymtl.ca

Marios Fokaefs
Polytechnique Montréal
marios.fokaefs@polymtl.ca

Kelly Lyons
University of Toronto
kelly.lyons@utoronto.ca

Eleni Stroulia
University of Alberta
stroulia@ualberta.ca

ABSTRACT

Collaboration is an important component of most work activities. We are interested in understanding how configurations of people come together to create outputs over time. We propose an interactive visualization tool (City on the River) for visualizing collaborations over time. The City on the River (CotR) visualization shows the contributions and artifacts ("products") of a team on a timeline and the individuals on the team who contributed to each product. CotR enables interactive analyses of each of these components for answering questions such as, which people work together on the most products, which products involve the most people, what kinds of products were produced when and by whom, etc. CotR can be used for analyzing diverse domains such as research collaborations, conference participation, email conversations, and software development. In this paper, we present the results of an experiment to assess CotR for analyzing collaboration and outcomes in GitHub projects. We compared the quality of answers, time to answer, and approaches taken to analyze the project collaborations by two groups of people: one group used the GitHub data displayed in a spreadsheet; the other group used the GitHub data displayed using CotR.

CCS CONCEPTS

• **Human-centered computing** → **Open source software**; **Visualization systems and tools**; **Empirical studies in visualization**; • **Software and its engineering** → **Collaboration in software development**;

KEYWORDS

visualization, software engineering, collaboration

ACM Reference format:
Jessica Perrie, Jing Xie, Maleknaz Nayebi, Marios Fokaefs, Kelly Lyons, and Eleni Stroulia. 2019. City on the River: Visualizing Temporal Collaboration. In *Proceedings of CASCON '19, Toronto, Canada, Nov 4-6, 2019,* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CASCON '19, Nov 4-6, 2019, Toronto, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Several visualization tools and techniques have been used to help people understand and analyze collaboration within teams. In software development, most of these focus on visualizations of artifacts such as commits, files, modules, and libraries [5, 11, 16]. It is also important analyze the social and temporal structures in electronic data to help in understanding collaboration [6]. Visualizing the collaboration histories of teams is useful in understanding interactions and outcomes resulting from collaborations. It also enables teams and organizations to understand patterns of collaboration and outcomes over time. Social interaction over time has been visualized to gain insights into dynamic networks over time [6, 8, 24]. Other studies present social and collaboration data along a timeline to allow individuals to reflect on contributions to a community [7, 10, 23]. The visualizations in these past studies use various techniques including *area graphs*, *matrix* and *node-link diagrams* in separate views.

Node-link diagrams in particular have been used to visualize network evolution; however, there are several drawbacks to this method: the graph may become too cluttered and difficult to understand; animation requires additional time and memory while small multiples requires more space; and, with nodes and links displayed so far apart, it may become difficult to follow changes in nodes and edges [16]. Additionally, when collaboration involves more than two people, adding additional edges to the visualization may inflate the perceived amount of productivity.

Presenting productivity using area graphs does not represent the combinations of collaborators nor how such individual relationships change over time [7, 10, 23]. The GitHub Visualizer provides a swarm visualization of files changed between members. In this visualization, animation is used to go between sequences, which is not ideal as it increase the cognitive load of the end user [16].

We propose City on the River (CotR)–a timeline-based visualization approach for understanding collaborative work output over time. CotR represents collaborative activity that results in work outcomes and maps these outcomes to individual contributions within the same timeline. The design leverages how area graphs (used famously in ThemeRiver [9]) allow people to adopt a stream metaphor, view larger continuous parts, and identify silhouettes. CotR displays area graphs where each stream represents an individual's contributions. These streams are displayed underneath a time-based histogram of work artifacts. The contribution streams are mirrored into individual paths through the stacks–hence, showing the combinations of collaborators and bringing to mind the idea of a city with tall buildings and paths. Hence, the name "City on the River" as seen in Figure 2.

In this paper we show how CotR can be used for visualizing the evolution of collaborations over time in software development teams. Software engineering is a collaborative effort and an increasing number of tools (such as Git) are provided to support developers' collaboration. Different collaborative practices or patterns may be observed at different times during the software development process, which may result in different outcomes.

The rest of this paper is organized as follows. In Section 2, we present related work and in Section 3 we describe CotR and its design and implementation. We demonstrate how CotR can be used to analyze collaboration in software development over time in Section 3.3. In Section 4, we present the results of an empirical study to assess the effectiveness of CotR for analyzing collaboration in GitHub projects over time. Finally, in Section 5, we conclude with a discussion of future work.

2 RELATED WORK

2.1 Collaboration Networks Over Time

TimeMatrix represents temporal social networks in a matrix-based approach; it visualizes changes in nodes and edges using a TimeCell bar chart glyph. Similar to CotR, it requires less space than a node-link diagram and represents strengths of relationships over time [25]. Collaborations among researchers have also been visualized as a means to represent researchers' work online (not for analysis)[1]. The kind of data represented visually in these systems is not unlike that in CotR; however, CotR differs from these systems by representing combinations of collaborations over a timeline rather than representing it in a slice-by-slice chord diagram or dynamic bar lists.

The intermediate tool Soylent was built to display connections between email co-recipients and verify that detectable patterns of contact (e.g., a tight core of densely connected people) were visible and relevant to end-users. It includes a node-link diagram of co-recipient relations between people within a start and end time. Because scalability is an issue when drawing cluttered node-link diagrams, the authors developed a minimalistic text interface that summarises the interactions between people on an egocentric point of view [6]. CotR supports a fuller visualization representation of this minimalistic text interface.

2.2 Contributions in Community Platforms

Visually, CotR's contributor streams are similar to approaches that visualize data from communities as a measure of the collaboration. AuthorLines, part of a larger tool that includes views of members' contributions, represents the number of threads initiated and threads not initiated by the author in a timeline in order to let the analyst get an idea of the individual's posting activity in newsgroups [23]. iBlogVis also presents a timeline with the length of the blog post represented by the height above the x-axis, while the height below represents the number of comments [10]. Unlike these approaches, CotR maps the collaboration within the products.

[1]For example see: (http://www.rjbaxley.com/p/publications.html) and (https://www.cs.umd.edu/users/bederson/papers/index.html)

2.3 Collaboration in Software Development

Storylines is a visualization approach that has been applied to software development to represent collaboration through the layout and proximity of links that represent contributors; if people do not work for some time, their link is disconnected from the main group [14]. It displays a commit histogram below the links representing the number of files in each commit (work representation). Unintentionally, our work closely resembles this representation by also using potentially disconnected links to represent contributors, but in CotR, the grouping of contributors' collaborations is done directly in the corresponding work representation. We also extend this approach by supporting the splitting of links and area graphs that aggregate the contributor's work over time. As a technique, storylines typically follow an optimization model to minimize edge-crossings [14, 20]; however, in CotR, links are tied to the order of the work outputs and optimization is currently outside our scope.

Tesseract is a system that visualizes the technical artifacts and social network analysis within a software project [18]. It uses cross-linked multi-perspectives for a timeline in order to help developers' communication match the code dependencies. Similarly, CodeSaw has separate area graphs above and below the timeline: the top timeline represents code contributions while the bottom represents project communication. By hovering over the stream, analysts can get a brief summary of the developer [7]. Like CotR, the timelines include product output (measured in commits) and there is a collaboration network view (measured in communication); however, CotR ties collaboration to products (rather than basing it on communication) and combines the network with the timeline.

Similar to Tesseract, Ariadne aims to bridge code dependencies and communication to explore the socio-technical relationships, and like CotR, it can identify artifacts in which individuals are connected [4]. However, unlike CotR, it employs a node-link diagram approach and focuses on awareness of development activities–much like a dashboard like FastDash [2]. These dashboards typically represent the current or recent collaboration in order to encourage awareness among individuals, unlike CotR, which was designed to allow an analyst look back on the history of the group or individual.

3 CITY ON THE RIVER

CotR represents a collection of work outputs called *products* on which people contribute at a particular time. The data model for CotR is shown in Figure 1. In software projects, we define a product as a file-commit pair where each pair represents a file in a repository commit. Each product should represent a roughly equal amount of work associated with contributions in a team (for example, the effort needed for each source code object is roughly the same). The other product attributes are similarly mapped: name is *filename*; description is *commit messages*; type is based on the *extension* of the file; timestamp is `when` the commit occurred; the person who contributed to the product is the `developer` who performed the commit.

3.1 Visualization Design

In addition to following visual display techniques from Edward Tufte [21], the final design makes use of feedback from researchers in information visualization and software engineering. CotR has

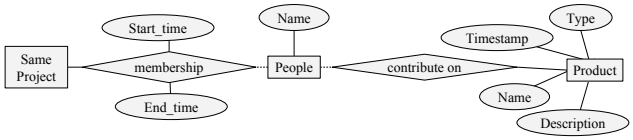

Figure 1: Data Model: attributes of product in product-based collaboration (rectangles are entities, ovals are attributes, and diamonds are relations)

two major components: a chart (visualizing the data) and a navigation panel (interacting with the chart) as shown in Figure 2.

The chart displays a timeline divided into evenly spaced time bins. As shown in Figure 2, the part above the x-axis displays the products completed within each time bin. Each product is represented as a small brown rectangle with a shade to denote the number of contributors and an icon to indicate its type. Individuals who worked on that product are represented as colour-coded links entering and exiting the product horizontally. A link is continuous between time bins if the individual it represents contributed to at least one product in each adjacent bin. Below the timeline, an area graph, much like ThemeRiver [9], contains streams for each individual; each stream expands or shrinks to match the number of products worked on by that individual in the time bins. At a glance, the end user (analyst) can see the amount of collaboration and productivity within each time bin. Tall product stacks with shorter stream heights mean many products were created without much collaboration. Short product stacks with higher stream heights represent collaboration taking place with fewer resulting products.

Interaction: To enable the analyst to drill into and manipulate the data, CotR also supports a number of interactive techniques present in both the chart and navigation panel. From the chart, the end user can manipulate size, find collaborations between people, and drill down for information. Resizing the node dimensions, the number of days in each time bin, and the chart width are all available by dragging. The chart can be panned (while the navigation and y-axis stay fixed) by using browser scrollbars. Other interaction in the chart provides tooltips about the product nodes and streams by hovering the cursor over them and highlighting related streams or products. As shown in Figure 2, the navigation panel contains controls that can be used to reset to the initial configured view; modify the shading increments of products and links; and highlight or filter the chart data. The analyst can also reorder the products within each time bin by differnt attributes.

Highlighting or filtering the chart data is available for each of the attributes – product types ("Products"), number of contributors ("Contributors"), and people who contributed to the products ("People"). The navigation panel displays the different values (e.g. for product type: "Document", "Source code", etc.) of each attribute with their visual encoding and what percentage of these products are currently highlighted or displayed in the chart as shown in Figure 2.

For example, if the analyst wants to understand the collaboration history of two people who worked on the same product, he or she starts by highlighting both of the names together. The products the people worked on together will be highlighted in the chart. The shape of the timeline shows when the collaborations took place.

3.2 Implementation

CotR has been implemented using the JavaScript library *D3.js*[2] and currently runs on Firefox Browser. It can display approximately 300 products with minimal noticeable lag. In designing the visualization approach, scaling for large numbers of products was not an initial consideration.

3.3 Example Collaboration Scenarios

Because CotR implements a general collaboration data model (Figure 1), it can be used to visualize many collaboration scenarios.

Conference participation Figure 3 shows conference outputs from 25 years of the CASCON conference [12]. In this case, people collaborate on products such as papers, demos, posters, or workshops. In the case of a paper, the *name* is paper title, and *description* is the papers' abstract. The timestamp represents the year that the contribution was presented at the conference. The people who contribute to the product are the co-authors. Because the large number of contributions in 25 years of data do not visualize well as is, we added the cability to filter the data by topic area. We experimented with LDA [3] and LSA [19] to define topics of the CASCON contributions. The topics let the user of CotR select papers around a specific theme using keywords.

Research networks Figure 4 represents collaboration of an individual in a research network. From the visualization, we can see the products in which many collaborators worked together. In general, there seems to be fewer products before 2010, reaching high productivity in 2012 before reducing productivity again in 2014.

Email Conversations The visualization presenting different people in the same email conversations (where emails with the same subject are replied to by different recipients) is shown in Figure 5. This data is actual data from the emails received by the main researcher (in blue) in early stages of this study. In these emails, you can see how other another researcher (in pink) was highly involved consistently over time, and near the end additional people made up much of the correspondence (representing when participants contacted the main researcher to be part of the user study).

4 USER STUDY

We designed a controlled user study to evaluate the effectiveness and efficiency of CotR by comparing it with a text-based approach. Our study is similar to the one conducted by Kang et al. [1]. We set out to answer the following research questions:

RQ1: How do people approach data analysis using CotR vs. using a text-based information source?

RQ2: Do analysts obtain better insights faster when using the CotR visualization approach compared to the text-based method?

Our study follows a between-group design by having each participant use one of two interfaces that display the dataset–a collection of products produced by collaborators on a software development team. We used a GitHub repository log of a project completed by students in their third year software engineering course that spanned

[2] http://d3js.org/

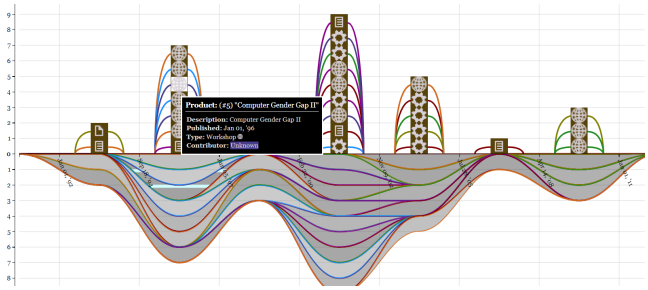

Figure 3: CotR showing the collaborations in a conference through papers, demos, workshops, and keynote presentations.

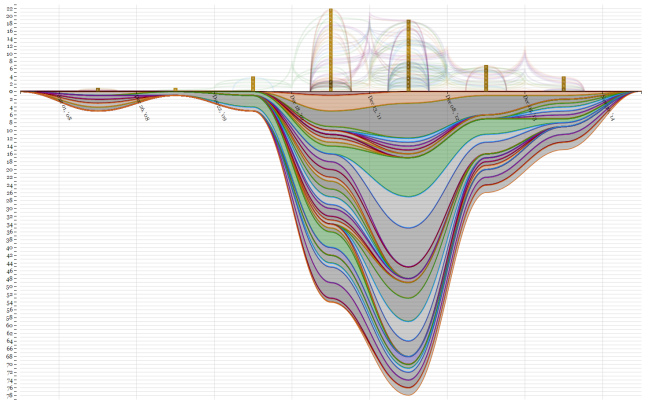

Figure 4: CotR in an egocentric perspective showing the collaborations of an individual in a research network.

four months. We note that student data is acceptable in place of actual industry records, because the two sources have been found to be similar [17]. The two interfaces are CotR and a spreadsheet program acting as a text-based alternative to the visualization. In the controlled experiment, half of the participants used CotR and half used the spreadsheet and log files to answer a set of questions about collaboration and outcomes. There are many other ways we could have represented developers' interaction to compare with CotR [7, 14]; however, we chose this interface because of its ubiquity and similarity to what may be used if the original GitHub repository was not available and because it enabled us to anonymize the data for the study. As a plain text file, the product listings would be too detailed and tedious to browse through; hence, we provide it in a spreadsheet application with access to functions similar to those found in CotR as shown in Table 1.

To answer *RQ1*, we collected notes taken, screenshots and mouse clicks, and asked participants to talk aloud as they answered the questions. To answer *RQ2*, participants were asked questions (Appendix A) about the collaboration in the project and their answers were compared with answers provided by the actual Teaching Assistant (TA) for the course.

Preprocessing: A few steps are required to translate repository log data into the CotR data model. A simple transformation would represent each file in each commit as a product in our data model; however, the number of products could be overwhelming to the analyst using the visualization and several trivial products (e.g., file-commits when the action is to delete the file and move it's content elsewhere in another file-commit) would be unnecessarily included. The data can be preprocessed using the following steps to reduce the amount of potential products: format, clean, filter,

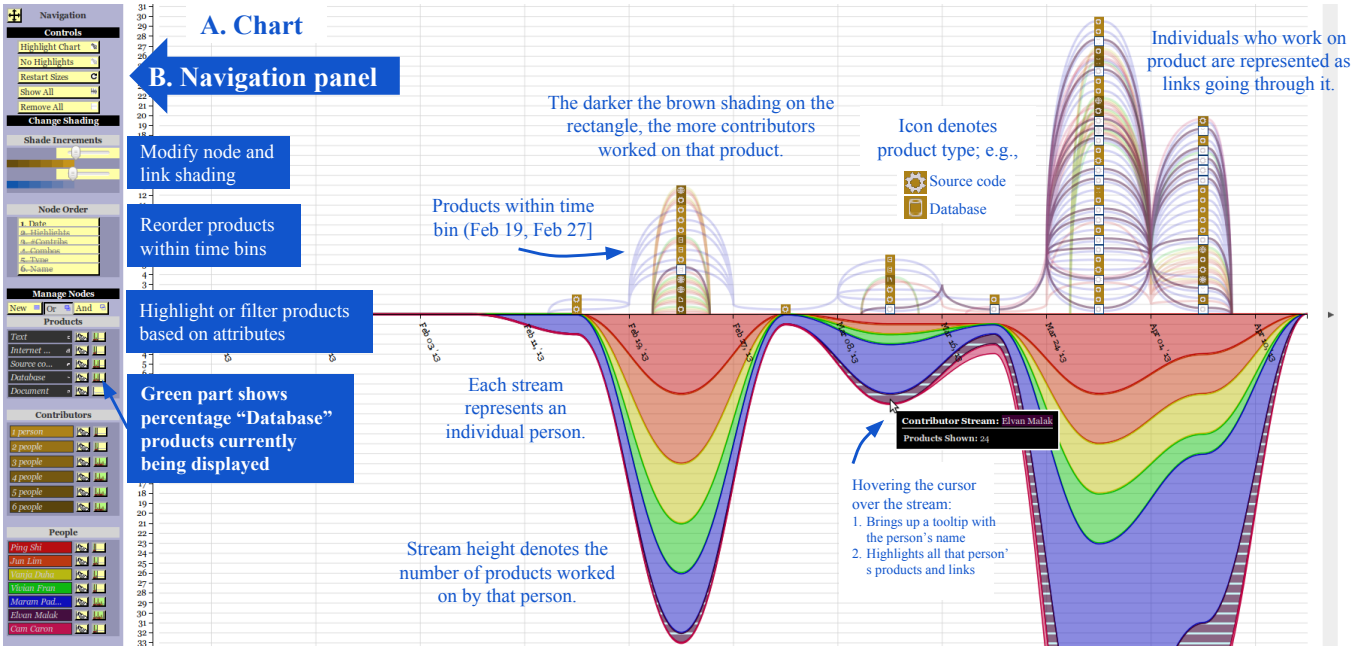

Figure 2: Overview and description of features of CotR panel.

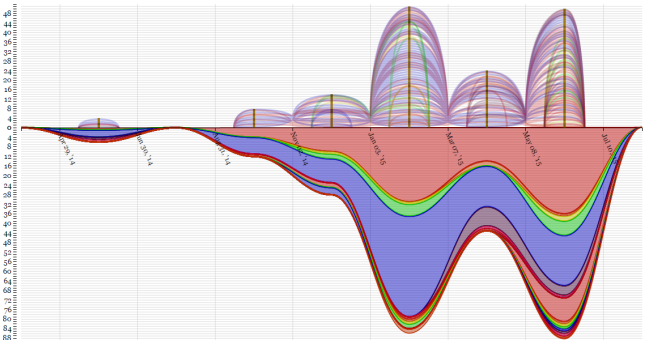

Figure 5: CotR showing the email correspondence of people in email conversations.

Table 1: Comparison of text alternate condition and visualization condition used in user study.

Text-based alternative	CotR
GitHub Contributor visualization, Pivot Table of time bins with contributor columns (*mc1*), TRUE/FALSE columns for each individual	Contributor stream for each individual
Chart based on Pivot Table of time bins vs. product type	Product stacks with icons
Filtering, conditional formatting on the rows	Filtering, highlighting
Dynamic time bin column with access to change number of days in each bin	Resizable time bins
Sorting for each attribute	Sorting within the node stack for some attributes

and merge. Depending on the nature of the development project, different parameters might be chosen for each preprocessing step.

Format: A typical GitHub commit includes the time of the commit, the author and a list of files being added, deleted or modified. In the Format step, no file-commits are removed, but these different attributes are parsed and organized with their file-commit.

Clean: When a file is *moved* in a GitHub repository, the log records a Delete in the old location and an Add in the new location within the commit where the move occurred. The Add and Delete are collapsed into a single Move.

Filter: In the filter step, some of the data is trimmed by removing files that are only committed once and removing all Delete file-commit listings. It is assumed that a file committed once was neither evolved nor was there any collaboration related to it. It is further assumed that if a file was deleted, then its important content has been moved elsewhere within another modified file. Additionally, only listings with certain types might be included. For instance, in our case of student project data, a file of type code, document or similar is saved; these types of files were most important in grading.

Merge: Since each file can be committed multiple times in a short amount of time, in the merge step, products are merged together to form a new product if the commit for the same file occurred within a certain timeframe. This ideal timeframe depends on the data and, specifically, what is known about the project's schedules (e.g., a sprint may take a month).

After the preprocessing, the project had 266 products and 7 individual contributors. It was processed from a GitHub repository with 621 commits and 46323 lines of code. The product attributes for a file-commit are as follows: name is *filename*; description is *commit messages*; type is based on the *extension* of the file; timestamp is *when the commit occurred*; the person who contributed to the product is *the developer who performed the commit*. When two products are merged in the *Merge* step, the name and type stays the same, the merged description concatenates the commit messages, and the timestamp is that of the later commit. The contributors of the product then make up the different developers or collaborators, who committed the file within that time.

Text-based alternative Interface: The text-based alternative interface displays the product listings (with headers) in a spreadsheet application (i.e., LibreOffice 4.2.8 Calc[3]) such that each product is in a row and each attribute of the product is listed in each column. The participant had access to sorting, filtering, search, conditional formatting, pivot tables, and chart creation. They also had access to the GitHub Contributor visualization and the original repository history log.

4.1 Study Procedure

The study consisted of three parts: tutorial (20-30 minutes), session (30-40 minutes), post-session interview (5-10 minutes). The specific questions used in the different parts are found in Appendix A.

1. Tutorial. At the start of the study, each participant was introduced to one of the two interfaces with a series of short tutorial videos and some additional time to get used to the interface and review the tutorial content if needed. This method of learning the interface is similar to the introductory video and training used in [1], and much of the tutorial was designed using guidelines from [15]. After each video, participants were asked to answer some simple questions (e.g., "How many products occurred between March 5 and March 17?") to help them apply, and hence, retain the information they have learned. We created similar tutorial video segments for both interface conditions (e.g., how to adjust time bins), and the tutorial questions were identical in each condition. Each tutorial question had hints to help the participant recall how to use the interface and answer the question, although use of the hints was not required. If participants answered a question incorrectly, they were shown how to answer it correctly with the interface.

2. Session. After completing the tutorial, participants were shown their assigned interface with the software engineering student project dataset. They were led through the study questions (Appendix A) which were more complex (e.g., "Describe the nature of the collaboration of each of the individuals")–these kinds

[3] https://www.libreoffice.org/discover/calc/

of questions are important when measuring the effectiveness of visualizations [13].

3. Post-session interview. After completing the session, each participant was asked about their experience with the interface(s).

4.2 Participants

Sixteen participants were recruited through the researchers' academic network; participants had to have completed a Teaching Assistant (TA) school term position in a computer science course that required student group work at Canadian University. Participants for each interface ranged across different age groups and current level of education. Seven identified as female, while seven identified as male; one identified as other and one preferred that their gender not be recorded. Participants had between 1 to 12 terms (average: 3.62) working as a TA to oversee group projects and all had used spreadsheet applications before. Participants were randomly selected to be in one of the two conditions.

For their participation in the study, participants were given a $20 gift card to Starbucks. The study was designed to take between 60 to 75 minutes, although participants took between 60 and 120 minutes.

4.3 Observation & Coding Procedures

To answer RQ1 (*How do people approach data analysis using CotR vs. using a text-based information source?*), we analyzed the screen recordings of the participants' interactions with the interfaces. We first coded when certain interactions were used (e.g., *modifies filters to only see Naseem's products in chart*; or *opened window containing the Git visualization*), and then looked for patterns in these observed interactions. Common strategies were grouped together to understand how the different interfaces were used during the study.

To answer RQ2 (*Do analysts obtain better insights faster when using the CotR visualization approach compared to the text-based method?*), we coded the participants' answers and transcripts of their audio recordings, and compared those to the answers given by the actual course TA. All numeric answers were in a range from 1 to 5. We computed the accuracy of the answers of the participant (P) as the difference between the numeric rating given by the TA and that given by the participant as: − |TA-P| so that a score of 0 is the best (same as the TA) and a score of -4 is the worst (TA gave 1 or 5 when the participant gave 5 or 1, respectively). For each question, the TA also gave an explanation for their rating. Hence, we coded the verbal answers given by the participants. If the participant expressed the same explanation as the TA (we refer to this as *articulated common reasoning*), they were given an extra point for accuracy, in answering that question.

We also counted the number of times each participant expressed utterances of difficulty or frustration and we counted the number of times each participant mentioned specific numbers during their analysis. In answers to Q2 (where participants had to assess the collaboration of each of the team members), we coded the different aspects of the data that participants mentioned. These different possible aspects are as follows: the relative amount of collaborative products; how well the team member contributed products over time; types of products; how the individual worked with other team members in terms of the number of fellow collaborators, or with whom they collaborated more frequently; their number of products; their number of collaborative products. Only Q2 was coded in this intensive manner because it was the first complex question in the session.

4.4 Results

In this section we present the results of our study and address the research questions.

RQ1: Data Analysis Strategies

In answering RQ1 *How do people approach data analysis using CotR vs. using a text-based information source?*, we observed general approaches that were used by participants in both conditions, and some approaches that were used in one condition, but not the other.

Data Analysis Approaches used in Both Conditions:

- Participants from both conditions generally looked at an overview of the data before filtering the products by different team members and other attributes (e.g., number of contributors). This unguided procedure follows the common visualization mantra "Overview, filter, details on demand" [1].
- Although the data was in a time series, time was not always used as a factor in participants' analysis; however, in answering Q2, more visualization condition participants than text condition participants mentioned time as a factor in describing the nature of the collaboration for each individual, likely because the visualization prominently displays a timeline.
- As the session went on, participants in both conditions interacted less and less with the interfaces, which may suggest that through the process of answering the previous questions they gathered information in their memory.

Differences in Data Analysis Approaches Between Conditions:

- In answering Q2 (assessment of individuals' collaboration), text condition participants would sometimes filter the amount of products between each of the team members rather than going through each team member and viewing their products with more than two contributors. This was a more work-intensive approach because it required more interactions and the ability to compare the number of products of all team member pairs. This approach was less frequent in the visualization condition, possibly because of how the contributor streams of collaborating team members appear when viewing an individual's products.
- While participants in both conditions interacted less with the interface as the study progressed, this phenomenon was more pronounced among participants in the text condition. This suggests that data was better understood in the text condition, or better conveyed through the Git visualization or *mc1*.
- Compared to the participants who used the text alternate, visualization participants' note-taking had fewer numbers and fewer words. Additionally, they used fewer precise numbers and percentages in their answers suggesting that their reasoning was more qualitative than quantitative.
- In answering Q1 (general assessment of team collaboration), two of the eight participants from the visualization condition did not interact with interface, but just *looked* at interface to answer the question. Perhaps this is because the initial view

Table 2: Results (average, standard deviation): quality of answers, time to complete each question, and efficiency (quality / time). Faster times and greater correctness are bolded. Answers with significance difference are indicated with a *.

	Q1	Q2	Q3	Q4	Q5
Average score for quality of answers					
TEXT	**(-0.125,0.835)**	**(-1.071,1.204)**	**(-0.339,0.959)**	(0.75,0.463)	**(-0.571,0.535)**
VIS	(-0.286,1.113)	(-1.179,1.539)	(-0.554,1.094)	**(1,0)**	(-0.75,0.707)
Average Time (seconds):					
TEXT	(287.383, 151.938)	(870.42, 245.971)	**(466.402, 306.194)**	**(92.684, 40.401)** *	**(78.059, 48.636)**
VIS	**(244.844, 153.693)**	**(869.588, 399.19)**	(629.556, 212.973)	(224.748, 147.389)	(89.033, 33.688)
Average efficiency (quality of insights / time (minutes)):					
TEXT	**(-0.061,0.201)**	**(-0.55,0.22)**	(-0.486,0.558)	**(0.585,0.377)**	**(-0.342,0.419)**
VIS	(-0.161,0.467)	(-0.665,0.596)	**(-0.404,0.457)**	(0.39,0.278)	(-0.574, 0.602)

provides an overview of the team and contained all the products and contributions in one window.

- When answering Q3, some visualization condition participants would rapidly go back and forth between team members—likely to create a view of contributions for each team member in order to gain an overview of all members' work relative to each other. This comparative view was more available in the text alternate condition: small individual commit graphs (Git Visualization) or amounts in the *mc1*.
- Finally, when answering Q5, visualization condition participants filtered products based on the number of contributors—actions that had been done earlier in the session. This suggests that these participants would benefit by being able to save their views or see the stack area graph of contribution streams separated out.

In summary, while some approaches to analysis were similar between conditions, there were several differences. CotR is a useful alternative for generally assessing collaboration of the whole team. The prominence of the timeline leads to characterizations that include the contributions over time. It also seems that users of CotR took fewer notes and that CotR is useful for more qualitative rather than quantitative assessments.

RQ2: Quality of Insights and Time Taken In Table 2, we summarize the results of our analysis of time and quality of results in answering *RQ2 Do analysts obtain better insights faster when using the CotR visualization approach compared to the text-based method?*

Time Performance: Participants in the text condition took longer to answer the first two questions, while for the later three questions the participants in the visualization condition took longer. One interface did not yield significantly faster results over the other except when answering Q4; with the Welch's t-test with unequal variances and two independent groups, there was a significant effect for the conditions ($t(8.05) = -2.44$, $p < 0.05$, Cohen's $d=1.22$) with the text condition being faster than the visualization condition.

The length of time people took to answer questions largely depends on the participants' approaches. The text condition participants answered in less time in the last three questions likely due to the fact that text condition participants had access to multiple overviews of the data and could internalize the data faster; hence, coming up with answers required less interaction and subsequently less time. The visualization condition participants were faster answering the first questions because it presented a clearer overview of the collaboration in one view. However, it is likely that difficulties (learning the interface, having fewer views and remembering how team members performed relative to each other) meant that the data was harder to remember without manipulating the interface again—meaning more time was needed.

Correctness / Quality of Results We consider correctness and quality for each question.

Q1 & Q5: General assessment of team collaboration. We were interested to see if there were changes in answers to the general assessment question after using the interfaces. From the average score of both questions, the text condition yielded slightly more correct results that were closer to the TA answers and based on an articulated similar reasoning; however, these differences were not significant. The majority of participants in both conditions judged the level of collaboration to be less than that specified in the TA answers possibly because they lacked a comparative group with which to measure performance. Viewing multiple student submissions before handing out marks is an important practice as one participant stated: *"Once I have seen multiple groups and seen the others' collaboration, then I will have a better sense."*. The lack of ability to compare with other groups might explain why most participants (Q1: 5 TEXT, 3 VIS; Q5: 5 TEXT, 7 VIS) generally chose the middle ranking (3/5 for "Acceptable Collaboration").

It is interesting to note that three participants (2 VIS, 1 TEXT) lowered their rating of the group in Q5 compared to their answer in Q1, while one (VIS) increased their rating. It's possible that most participants felt tied to their older assessment. Some participants noted that answering the more other session questions did not change their assessment, or that a scale of 1 to 5 was too narrow (that a change in mark would mean a change of 20-25% of grade).

Q2: Assessment of team members' collaboration & Q3: Assessment of team members' performance. Figure 6 shows a breakdown of the average quality answer for Q2 and Q3 for both conditions and for each of the team members being assessed. All of the participants performed well. Text scores are slightly better overall, but the difference in the results is not significant.

Q4: Who would you want to work with OR not want to work within a software development project. In answering this question all participants in the visualization condition included the name mentioned in the TA answers in the *want field* compared with the 6 of 8 text condition participants. One of the two text condition participants who did not mention the TA's selection said, *"I tried to get the most of [the data].I focused on some variables, [but] maybe didn't have the whole picture."*, while the other said, *"I don't think you can judge based on the numbers here. There are many factors: Do they fit the rest of the team? Do they fit the culture of the team?"*.

Because the TA answers did not include any team members in the *not want to work with condition*, we did not access participants for their choices in the latter part of the question.

In summary, in general, the text condition yields results that were closer to the TA answers than the visualization condition and, for the later questions, took less time to complete. Compared to the two participants who had heard of CotR before, all participants had used a spreadsheet application at some point prior, and hence, did not need additional instruction to figure out how products' details were "encoded" in the product listings. As found in other research, it is possible that the existing text condition interface may simply work better or that participants did not have enough time to become accustomed to the visualization interface [22]. Different backgrounds in participants may also help explain the variability in the quality of insights generated from the visualization condition: visualization condition participants made up the top two and bottom three quality rankings out of all the participants. We discuss these results in great detail in Section 4.5.

Participant Experience

When asked about their experience, all participants in the visualization condition and most participants in the text condition stated that the interface was okay. Participants from both conditions identified some issues with their experience—especially when learning and using the interface in terms of its form and data.

When asked what features they liked, many text condition participants mentioned being able to filter the data—especially with the TRUE/FALSE columns for each team member—and using charts from the Git visualization and spreadsheet functions. Participants from the visualization condition mentioned the colours of team members, filters, button percentages, product icons and other hovering functions. Participants in both conditions mentioned liking the filters, adjustable time bins, and individual contribution area graphs (i.e., contribution streams or Git visualization)—and indicated features that should be included in interfaces for this type of assessment.

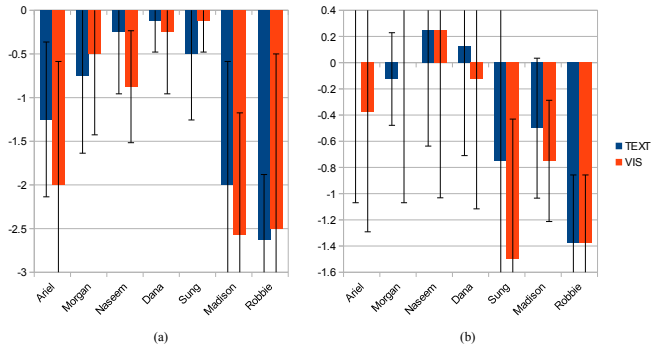

Figure 6: Average (and standard deviations) of coded quality scores of participants from both conditions for questions (a) Q2 and (b) Q3.

Table 3: Suggestions to improve the text condition or visualization condition interfaces from participants from each respective group.

TEXT Condition	VIS Condition
Macro operations to product stats instead of interacting with filters. More visual views, differences between team members, contributions of individuals along timeline. Drill-down into commit details from products. Summary of commit messages available.	Improve learnability (instructions, recognition). Display what elements are filtered in or out. Improvements to resizing functions: remove buffers, show resize height in legend Considerations for scaling: text box to select people, keep colours alternating Improvements for reading data: highlight axis lines for ease of counting, orient both part of charts upwards

When asked about what they did not like, three text condition participants said "Nothing"; however, others mentioned that they felt that they were missing details, that the interface was poorly setup and not intuitive, and the data was at times overwhelming. Similar issues were mentioned by participants of the visualization interface; four participants found that the filter interface behaved unexpectedly and was confusing to use, especially at the beginning of the session. Other comments were about reading the stacked area graphs; this issue is supported by how people often chose to filter through team members instead of looking at the overall stacked graph.

Participants were also asked if they could suggest anything to improve the interface; these results are grouped together in Table 3. Participants from the text condition suggested more high-level improvements and additional functions (including suggesting visualization views similar to those in CotR), while participants from the visualization condition suggested more low-level improvements and slight changes to the interface.

When asked about emotions they experienced, many participants in both conditions said that they experienced no emotions (6; 2 TEXT; 4 VIS). Others mentioned frustration (3; 2 TEXT, 1 VIS), uncertainty (2 TEXT) and confusion (2 VIS), suggesting that some participants found this study challenging.

From coding the utterances of difficulty, the participants in the visualization condition had slightly fewer exclamations (relative to text condition); most utterances occurred when answering Q2, where the participants had to first look at the data in some detail to assess each team member.

In summary, participants in both conditions seemed to be challenged by the study. The visualization participants expressed slightly

fewer utterances of difficulty. Suggestions for changes included visualization features in the text condition and detailed design suggestions for the CotR visualization.

4.5 Discussion

Learning the interface. In the visualization condition, both participants with the highest and lowest quality of answers found difficulty with the tutorials: "*I think the [tutorial] videos were not very helpful for me*" and "*I didn't like the tutorial. [the interface] was very easy, but the tutorial was not*". Hence, while more text condition participants were at relative ease to filter products, many visualization participants may have been still trying to learn the interface. This means that more work is needed in order to design an interface that follows similar conventions to the alternative condition. Additionally, it would be useful to detect when participants are showing signs of difficulty with the interface automatically and then enable them to walk through different aspects of the tutorial at a more leisurely pace.

Access to different data: In addition to product listings, text condition participants also had access to the absolute number of commits of each team member from the Git visualization. Hence these participants had more sources of information to consider in their assessment, which may have changed their opinion and provided them with more details that the actual TA used. For example, from Q4, selecting Dana as a team member with which to work was more likely from participants in the text condition (5 TEXT; 1 VIS). The Git visualization showed that Dana had the fourth highest number of commits with an area graph of commits over time that was similar to the other top committing team members; in the CotR visualization, Dana's commits were not available, and consistently in each bin, Dana's products were less than other team members.

Access to different views. Compared to the text condition, the CotR visualization had a somewhat un-intuitive filtering interface that confused participants (as evidenced by suggestions to improve it). Taking extra concentration to learn this system may have hindered participants' analysis of the data in the chart. Furthermore, the text condition was made up of three (or more) sheets that could be revisited with a single click, whereas, to revisit previous views in the visualization, participants had to select all the filter parameters again. Hence, this additional action cost more time and took more concentration away from understanding the data. Finally, participants seemed to appreciate unstacked area graphs of products of each team members' contributions, as they frequently filtered by single members. Unstacked area graphs of commits were provided in the text condition's Git visualization for easy comparison between the team members. In the CotR visualization, to see this unstacked view, participants had to interact more with the interface.

4.6 Limitations

The small number of participants (8 in each condition) meant that, for the most part, it was not possible to find if one interface yielded results that were significantly greater than results from the other.

By comparing the coded results to the TA answers, we compared the participants with someone who had much greater knowledge and background about the data and access to multiple teams' performance for context.

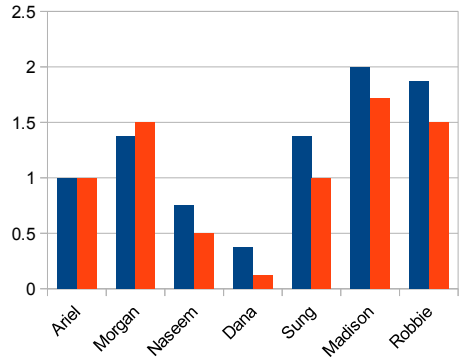

Figure 7: Average difference in ratings for Q2 between adjusted participant and TA answers ratings (e.g., highest rating of team member is 1; lowest rating is -1; otherwise 0)

Additionally, although the data was extracted from the GitHub repository that is used to assess students' work; the main elements of the text office (Libreoffice) were not used in real student assessments and only chosen because it offered an interface with much of the same functions as CotR. The actual GitHub interface and project documents would be better for comparison to CotR; however, they could not be easily anonymized for the study.

5 CONCLUSION

We presented CotR, a visualization approach for understanding collaborative work over time. Within each time bin, CotR displays products in stacks with links running through them indicating individuals who worked on them; an aggregation of these contributions is represented by a stream below. We presented the results of an empirical study that compared CotR to a text-based approach for analyzing collaboration and outcomes in GitHub projects. We compared the quality of answers, time to answer, and approaches taken to analyze the project collaborations by two groups: one that used the GitHub data displayed in a spreadsheet; the other group used the GitHub data displayed using CotR.

We identified different approaches that people used to analyze collaboration in CotR versus the text-based alternative. We found that the text alternate condition yields slightly better answers to the questions and, after the first two questions, resulted in faster times to answer. We found that CotR may be more applicable when considering qualitative assessments. Compared to the text interface, CotR resulted in more variation among answers and less use of precise numbers in the answers. Participants made several detailed design suggestions for CotR. It is interesting to note that when asked what an ideal interface would be, one text condition participant suggested a visual interface similar to the design of City on the River:

Let's say like we have the time bins; we could see how many different commit messages happened in a time bin. We could see what are the people that each person collaborated with the most. In which periods of time there were more people collaborating together. How many

different contributions one person did in each time span to which product. Breakdown by time bin, breakdown by product, maybe filenames. That's what comes to my mind.

There are several areas of future work. CotR's current design and prototype implementation does not scale well. Large numbers of products result in tall product stacks and smaller product boxes. If products' heights are reduced such that their shading is not visible, the analyst loses the ability to spot clusters in the product stacks. More effective techniques in the implementation (e.g., more aggregated or partial rendering of data) may overcome this issue, but such large datasets are outside the scope of our initial studies.

CotR currently represents two timelines: the product stacks and the contributor streams; however, it is possible to add additional timelines by overlaying, replacing, or positioning within the analyst's vision. This opens the possibility to display timelines containing total group size, annotations, events, comparative performance, and more.

Finally, CotR can be compared with other similar visualization approaches (e.g., Storylines [7, 14]) to see how CotR performs in strictly visual environments. It would also be interesting to conduct studies of CotR in other application domains.

ACKNOWLEDGEMENTS

We would like to thank Jian Zhao and Ravin Balakrishnan for feedback on the design of CotR, and Bahar Ghadiri Bashardoost for providing input to the interface and study plan. This work is funded by NSERC and GRAND NCE.

REFERENCES

[1] Youn ah Kang, C. Gorg, and J. Stasko. 2009. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. 139–146. https://doi.org/10.1109/VAST.2009.5333878

[2] Jacob T. Biehl, Mary Czerwinski, Greg Smith, and George G. Robertson. 2007. FASTDash: A Visual Dashboard for Fostering Awareness in Software Teams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1313–1322. https://doi.org/10.1145/1240624.1240823

[3] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 113–120.

[4] Cleidson R. de Souza, Stephen Quirk, Erik Trainer, and David F. Redmiles. 2007. Supporting Collaborative Software Development Through the Visualization of Socio-technical Dependencies. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work (GROUP '07)*. ACM, New York, NY, USA, 147–156. https://doi.org/10.1145/1316624.1316646

[5] B. Ens, D. Rea, R. Shpaner, H. Hemmati, J.E. Young, and P. Irani. 2014. ChronoTwigger: A Visual Analytics Tool for Understanding Source and Test Co-evolution. In *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. 117–126. https://doi.org/10.1109/VISSOFT.2014.28

[6] Danyel Fisher and Paul Dourish. 2004. Social and Temporal Structures in Everyday Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 551–558. https://doi.org/10.1145/985692.985762

[7] Eric Gilbert and Karrie Karahalios. 2007. CodeSaw: A Social Visualization of Distributed Software Development. In *Human-Computer Interaction âĂŞ INTERACT 2007*, CÃĺcilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone-DinizJunqueira Barbosa (Eds.). Lecture Notes in Computer Science, Vol. 4663. Springer Berlin Heidelberg, 303–316. https://doi.org/10.1007/978-3-540-74800-7_25

[8] Peter A Gloor and Yan Zhao. 2004. Tecflow-a temporal communication flow visualizer for social networks analysis. In *Proceedings of the ACM CSCW workshop on social networks. ACM CSCW Conference, Chicago.*

[9] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. 2002. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (Jan. 2002), 9–20. https://doi.org/10.1109/2945.981848

[10] Indratmo, Julita Vassileva, and Carl Gutwin. 2008. Exploring Blog Archives with Interactive Visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 39–46. https://doi.org/10.1145/1385569.1385578

[11] R.G. Kula, C. De Roover, D. German, T. Ishio, and K. Inoue. 2014. Visualizing the Evolution of Systems and Their Library Dependencies. In *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. 127–136. https://doi.org/10.1109/VISSOFT.2014.29

[12] Dixin Luo and Kelly Lyons. 2017. CASCONet: A Conference dataset. *arXiv preprint arXiv:1706.09485* (2017).

[13] C. North. 2006. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE* 26, 3 (May 2006), 6–9. https://doi.org/10.1109/MCG.2006.70

[14] Michael Ogawa and Kwan-Liu Ma. 2010. Software Evolution Storylines. In *Proceedings of the 5th International Symposium on Software Visualization (SOFTVIS '10)*. ACM, New York, NY, USA, 35–42. https://doi.org/10.1145/1879211.1879219

[15] C. Plaisant and B. Shneiderman. 2005. Show Me! Guidelines for producing recorded demonstrations. In *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on*. 171–178. https://doi.org/10.1109/VLHCC.2005.57

[16] S. Rufiange and G. Melancon. 2014. AniMatrix: A Matrix-Based Visualization of Software Evolution. In *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. 137–146. https://doi.org/10.1109/VISSOFT.2014.30

[17] Iflaah Salman, Ayse Tosun Misirli, and Natalia Juristo. 2015. Are Students Representatives of Professionals in Software Engineering Experiments?. In *Proceedings of the 37th International Conference on Software Engineering*. ACM, 666–675.

[18] A. Sarma, L. Maccherone, P. Wagstrom, and J. Herbsleb. 2009. Tesseract: Interactive visual exploration of socio-technical relationships in software development. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*. 23–33. https://doi.org/10.1109/ICSE.2009.5070505

[19] Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* 427, 7 (2007), 424–440.

[20] Y. Tanahashi and Kwan-Liu Ma. 2012. Design Considerations for Optimizing Storyline Visualizations. *Visualization and Computer Graphics, IEEE Transactions on* 18, 12 (Dec 2012), 2679–2688. https://doi.org/10.1109/TVCG.2012.212

[21] Edward R Tufte and PR Graves-Morris. 1983. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT.

[22] J.J. van Wijk. 2005. The value of visualization. In *Visualization, 2005. VIS 05. IEEE*. 79–86. https://doi.org/10.1109/VISUAL.2005.1532781

[23] Fernanda B. Viégas and Marc Smith. 2004. Newsgroup Crowds and AuthorLines: Visualizing the Activity of Individuals in Conversational Cyberspaces. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4 (HICSS '04)*. IEEE Computer Society, Washington, DC, USA, 40109.2–. http://dl.acm.org/citation.cfm?id=962752.962972

[24] Ji Soo Yi, Niklas Elmqvist, and Seungyoon Lee. 2010. TimeMatrix: Analyzing temporal social networks using interactive matrix-based visualizations. *Intl. Journal of Human–Computer Interaction* 26, 11-12 (2010), 1031–1051.

[25] Ji Soo Yi, Niklas Elmqvist, and Seungyoon Lee. 2010. TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations. *International Journal of Human-Computer Interaction* 26, 11-12 (2010), 1031–1051. https://doi.org/10.1080/10447318.2010.516722

Appendix A: Questions used in Study

Q1: General assessment of the collaboration of this team: (scale of 1 – no collaboration to 5 excellent collaboration; explain reasoning)

Q2: Describe the nature of the collaboration of each of the individuals. (scale of 1 to 5 for each individual; explain reasoning)

Q3: What is the overall performance of each of the individuals? (scale of 1 – very poor performance to 5 – excellent performance for each individual; explain reasoning)

Q4. Based on the data, which individual(s) (if any) would you WANT (NOT WANT) to work with on a software development team project? (explain reasoning)

Q5: Having completed the questionnaire, please reconsider answers to Q1