Empirical Software Engineering (2024) 29:42
https://doi.org/10.1007/s10664-023-10435-3

Software development metrics: to VR or not to VR

David Moreno-Lumbreras[1] · Gregorio Robles[1] · Daniel Izquierdo-Cortázar[2] · Jesus M. Gonzalez-Barahona[1]

Accepted: 10 December 2023 / Published online: 3 February 2024

Abstract

Context Current data visualization interfaces predominantly rely on 2-D screens. However, the emergence of virtual reality (VR) devices capable of immersive data visualization has sparked interest in exploring their suitability for visualizing software development data. Despite this, there is a lack of detailed investigation into the effectiveness of VR devices specifically for interacting with software development data visualizations.

Objective Our objective is to investigate the following question: "How do VR devices compare to traditional screens in visualizing data about software development?" Specifically, we aim to assess the accuracy of conclusions derived from exploring visualizations for understanding the software development process, as well as the time required to reach these conclusions.

Method In our controlled experiment, we recruited N=32 volunteers with diverse backgrounds. Participants interacted with similar data visualizations in both VR and traditional screen environments. For the traditional screen setup, we utilized a commercially available set of interactive dashboards based on Kibana, commonly used by Bitergia customers for data insights. In the VR environment, we designed a set of visualizations, tailored to provide an equivalent dataset within a virtual room. Participants answered questions related to software evolution processes, specifically code review and issue tracking, in both VR and traditional screen environments, for two projects. We conducted statistical analyses to compare the correctness of their answers and the time taken for each question.

Results Our findings indicate that the correctness of answers in both environments is comparable. Regarding time spent, we observed similar durations, except for complex questions that required examining multiple interconnected visualizations. In such cases, participants in the VR environment were able to answer questions more quickly.

Conclusion Based on our results, we conclude that VR immersion can be equally effective as traditional screen setups for understanding software development processes through visualization of relevant metrics in most scenarios. Moreover, VR may offer advantages in comprehending complex tasks that require navigating through multiple interconnected

Communicated by: Maria Teresa Baldassarre, Jeff Carver and Neil Ernst

This article belongs to the Topical Collection: Registered Reports.

✉ David Moreno-Lumbreras
david.morenolu@urjc.es

Extended author information available on the last page of the article

visualizations. However, further experimentation is necessary to validate and reinforce these conclusions.

Keywords Virtual reality · Dashboards · Controlled experiment · Code review · Pull request · Issues

1 Introduction

Data visualizations are graphical representations of data that leverage the human visual system's abilities to perceive and interpret visual cues, enabling the clear communication of complex datasets (Tufte 2001; Heer et al. 2010; Saket et al. 2017). They utilize principles from perception, cognition, and graphical design to encode data values and attributes using visual elements and properties. By mapping data to visual encodings - such as charts, graphs, maps, and infographics - patterns, trends, and relationships in the data can be easily identified and understood. Effective data visualization design incorporates considerations of layout, color, labeling, and interactivity, drawing upon scientific research to optimize accuracy, efficiency, and usability. Ultimately, data visualizations support data analysis, exploration, and communication, empowering informed decision-making across various domains.

Interaction with data visualizations is typically performed using a keyboard and a mouse, while viewing them on 2D screens. However, this mode of interaction does not fully exploit spatial perception and other abilities developed for navigating the 3D world (Bowman and McMahan 2007; Schuemie et al. 2001). Virtual reality (VR) immersion, where users engage in a virtual 3D environment, offers affordances such as spatial perception, immersive visualization, and natural interaction that align better with these capabilities. Recently, affordable VR devices have emerged, accompanied by standards like *WebXR* and *WebGL* (Jones and Goregaokar 2023; Jackson and Gilbert 2023), enabling VR applications to be easily portable across platforms and integrable with other applications and APIs. Consequently, we are now at a stage where it becomes reasonable to interact with data visualizations in VR. However, more evidence is needed to fully understand the benefits and challenges associated with this approach.

In the field of software engineering, researchers have posited that the integration of virtual reality (VR) can potentially facilitate reduced learning curves, enhanced creativity, and increased productivity among practitioners (Elliott et al. 2015). However, the current body of scientific literature lacks substantial evidence to support the notion that VR-based visualizations offer superior or comparable outcomes to traditional on-screen visualizations when applied specifically to software development data analysis. Further empirical investigations are warranted to comprehensively evaluate the efficacy of VR in this context and establish its potential benefits for software engineering practices.

In this paper, we present a controlled experiment comparing two approaches: **a 2D data visualization using *Kibana* dashboards *vs.* a VR immersive visualization using *BabiaXR*** of data about software development processes. Kibana is a popular technology for visualizing data, acting as a front-end for Elasticsearch. It is being used by Bitergia[1], a company offering commercial services in the area of software development analytics. In this experiment, we use dashboards produced by this company as a part of their commercial offer. *BabiaXR*[2] has been developed by Bitergia and Universidad Rey Juan Carlos. It is a toolset for visualizing

[1] https://bitergia.com

[2] https://babiaxr.gitlab.io/

data in 3D, both on-screen and on VR devices. In our experiment, it is used for producing the VR visualizations. The relevance of *BabiaXR* and *Kibana* for the Software Engineering community stems from their ability to facilitate the exploration and comprehension of software development data. By leveraging these tools, researchers and practitioners can gain deeper insights into code review processes, issue tracking, and other software development aspects, leading to improved software quality, efficiency, and decision-making.

In this study, we considered to visualize software development processes, as part of the Bitergia workflow. Specifically, we considered only changes in the repository via pull requests and issues and tasks of the repository, using the issues part provided in the repository. Pull requests, a part of modern code review (Bacchelli and Bird 2013; Thongtanunam et al. 2017), are a software development activity that has been widely researched by academia in the last years (Kononenko et al. 2018; Maddila et al. 2019; Yu et al. 2015). They are of major interest to industry and practitioners as they are effort-intensive, and often the cause of bottlenecks and inefficiencies (Sadowski et al. 2018). Issues are used for reporting bugs, asking for new features, or informing about important aspects of the software development processes. There are many studies on how projects and developers deal with issues. For example, some of them focus on the user experience (Bissyandé et al. 2013; Bettenburg et al. 2008; Hooimeijer and Weimer 2007), including the proposal of techniques for helping the management of issues (Antoniol et al. 2008; Sun et al. 2011, 2010; Tian et al. 2012).

The experiment was defined in a registered report (Moreno-Lumbreras et al. 2021) before it was performed. The primary aim of the experiment was to assess the comparative effectiveness and efficiency of VR immersion versus on-screen 2D visualization for comprehending and analyzing data pertaining to code review and issue handling processes. In collaboration with Bitergia, we devised a series of questions that required participants to leverage information from various visualizations to provide accurate responses. The questions were administered to participants in two different settings: through *Kibana* 2D dashboards displayed on a single screen and within a virtual reality environment using *BabiaXR*, featuring 3D visualizations. Each participant encountered both settings in a randomized order, with distinct project data assigned to each setting to mitigate the potential impact of prior knowledge when transitioning to the second setting. To minimize visualization-specific biases, we aimed to replicate 2D dashboards as faithfully as possible within the 3D VR scene. The analysis encompassed the correctness of participants' responses as well as the time taken to arrive at those answers. In total, 32 participants from both academic and industrial backgrounds took part in the experiment. The contributions of this paper include: (i) The investigation of the comparative effectiveness and efficiency of VR immersion and on-screen 2D visualization for software engineering data comprehension; (ii) The employment of a pre-registered experiment design to ensure transparency and reproducibility; (iii) A collaboration with the industry, in this case Bitergia, to develop tailored questions and utilize real-world data; (iv) Assessing participants' correctness of answers and time taken in both VR and 2D settings; (v) The inclusion of participants from academia and industry to capture diverse perspectives.

The remainder of this paper is structured as follows. We present the usage scenarios for both on-screen and VR dashboards in Section 2. Section 3 reports how the controlled experiment was structured and performed. Section 4 details the changes that we made from the registered report. Section 5 analyzes the results of the experiment. Section 6 discusses the main points of the results and what they entail. Section 7 details the internal and external threats to the validity of our results. Finally, Section 8 details related research, and Section 9 presents some conclusions.

2 Implementation

For our study, we followed the design presented in our registered report (Moreno-Lumbreras et al. 2021). In this paper, we present the experiment as it was executed, which in some cases differs from the original planning, due to unforeseen issues that we found while implementing or test-running the experiment. In Section 4, we present the changes that we made with respect to the original plans.

We conducted the study as a controlled experiment, following as much as possible the *ACM SIGSOFT Empirical Standards* (Ralph 2021) in aspects relevant for quantitative methods for experiments with humans, satisfying some of the essential and desirable attributes described in that recommendation. We followed the design of a "One Factor, Two Levels" experiment, being the factor the independent variable (the environment), and the two levels the two values it may have (VR or on-screen). Since participants will be presented randomly with the order for the settings (first VR or first on-screen), this experiment can be treated and formally divided into two "One Factor, Two Levels" sub-experiments.

The experiment consists in a set of tasks that a number of subjects (32) will perform. The set of tasks is composed by five tasks that are performed twice by each subject, first in one of two different environments (on-screen or VR) with data from one of two projects (CHAOSS and *OpenShift*), and then in the other environment, with data from the other project (with a total of 10 tasks per subject). Tasks will be performed in sequence, starting one right after completing the previous one. The participants in the experiment will repeat the tasks in order to assess the potential impact of learning and fatigue factors. By performing the tasks twice, once in each environment (on-screen or VR) and with different datasets (CHAOSS and *OpenShift*), it allows for a comparison of participants' performance across these conditions. This repetition enables the evaluation of potential learning effects, as participants may become more proficient or experienced in task completion over time. Additionally, by observing participants' performance across sequential tasks, the influence of fatigue and the order of tasks can be examined, determining if task performance is affected by factors such as mental exhaustion or changing task context. Overall, repeating the tasks facilitates a comprehensive analysis of the effects of learning and other factors on participants' performance and experiences in different environments and with different datasets.

Each task will be framed as a question that the subject should answer. The main variables measured about how each subject performs in each task will be (i) time to solve the task, and (ii) correctness of the answer. All questions will be related to the comprehension of some aspect of pull requests or issue handling (or both) of the corresponding project. To complete their tasks, subjects will be presented with two Kibana-based dashboards (in the on-screen environment) or with one *BabiaXR* scene (in the VR environment), with 2D or 3D visualizations with enough information to correctly answer the corresponding question. Both in the on-screen and the VR environment, the data visualized for the same project is exactly the same, and visualizations in both cases have been carefully designed to be as similar as possible. For this, the VR scene mimics the visualization in the on-screen dashboards, which are a part of a commercial product being used by customers of Bitergia, a company collaborating in the study.

Subjects perform the tasks in VR using Oculus Quest 2 headsets, opening scenes in the Oculus browser, and on-screen using a web browser on a computer with a single screen, 13 to 15 inches in size.

In both cases, so that in both environments the responses can be measured in the most reliable way, for the whole duration of the experiment participants are required to talk aloud.

Oculus has limited input devices, so we decided to use the microphone, since it can be used easily both in the Oculus and in a conventional computer. The experiment is followed by a supervisor, who takes notes of the answers of the participant, and of the time to answer. For all subjects, a video was recorded with their view of the dashboards while running the experiment, including their voice while answering, to check for the correctness of the answers and the time to answer noted down by the supervisor. In both cases the supervisor can see the scene *"with the eyes of the participant,"* and provide support if needed. For on-screen participants, the supervisor can see the screen, and for VR participants the headset was configured to cast the scene to a screen. Figure 1 shows a participant during the VR experiment and the screencast for the supervisor.

Before conducting the experiment, all subjects go through a short training, to make sure they understand how the Kibana-based dashboards, and the VR immersive scenes work. This is necessary because subjects had no previous exposure to Kibana, and only a few of them had previous experience with VR immersion. Upon completion of the training, a demographic form is used to control the subjects confounding variables. Once the form is completed, the experiment can begin by letting subjects complete tasks. After this, the subject will answer a feedback form which will be analyzed for possible improvements or problems that the participant has encountered during the experiment.

The rest of this section describes how the data is collected from the repositories of the projects considered in the study, and the tool chain used to produce the Kibana dashboards and the VR scene. The visualizations offered to subjects will also be described.

2.1 Data Retrieval and Processing

To gather the data from the projects, we use *GrimoireLab*[3] (Dueñas et al. 2021), a toolset for software development analytics. *GrimoireLab* can retrieve data from many kinds of software repositories (Dueñas et al. 2018), store it in an *Elasticsearch* database, and then process and analyze it, producing many different metrics (see Fig. 2), which are also stored. It also includes visualization modules that can be used to interact with the data via traditional, on-screen, web browsers. Data produced by *GrimoireLab* can be fed to *BabiaXR* and *Kibana* from its storage in *Elasticsearch*.

In our study, we use *GrimoireLab* to retrieve and process data from the repositories. The data are then stored in *Elasticsearch*, which serves as a storage system for efficient data retrieval. To facilitate our analysis and investigation, we develop a comprehensive set of visualizations and dashboards specifically tailored for both the on-screen and VR environments (*Kibana* and *BabiaXR*). These visualizations and dashboards enable us to effectively explore and present the data in a meaningful and immersive manner within each respective environment (Fig. 3).

2.2 On-screen Dashboards on *Kibana*

For the development of the on-screen experiment, we use *Kibana*. *Kibana* is a frontend application providing search and data visualization capabilities for data stored in *Elasticsearch*. In our experiment we use two *Kibana* dashboards from the Bitergia Analytics platform, whose main goal is to show several aspects of the pull request and issue handling processes, using data about their timing:

[3] *GrimoireLab*: https://chaoss.github.io/grimoirelab/

Fig. 1 An example of a participant during the VR round of the experiment

1. **Issues Timing**: displays information about the time to close issues.
2. **Pull Requests Timing**: displays information about the timing of pull requests.

Both dashboards offer some insight about the time to close pull requests or issues. They are intended mainly to visually find bottlenecks in the development process, and potential reasons for them, specially when they are linked to specific organizations collaborating in the project. Both dashboards include several visualizations, from simple ones showing raw data to others with more fine grained data, classified by organization, or current status of the pull request or issue, for example. Both Kibana dashboards are integral components of Bitergia's Analytics Platform, which is actively utilized by Bitergia's customers to analyze various processes. These dashboards have undergone validation and testing by Bitergia's customers, ensuring their suitability and effectiveness in supporting data analysis tasks. The validation process by customers helps establish the credibility and reliability of the dashboards, further enhancing their value as representative tools for data visualization and analysis. In

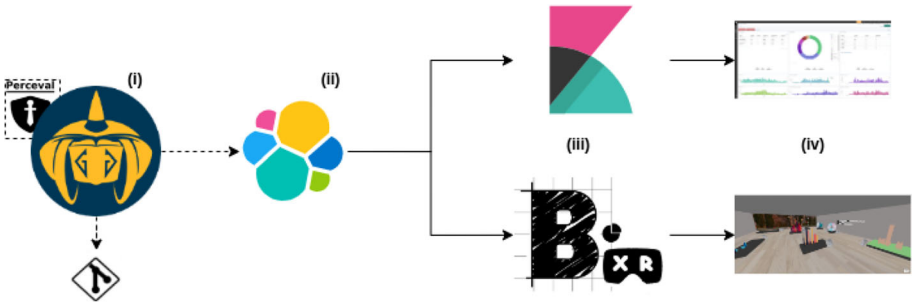

Fig. 2 Data flow diagram: (i) Data is gathered from the data source using *GrimoireLab*, which processes and stores it in *Elasticsearch* (ii). The stored data is then utilized to feed both *Kibana* and *BabiaXR* (iii), which are the tools used for constructing the on-screen and VR data visualizations (iv)

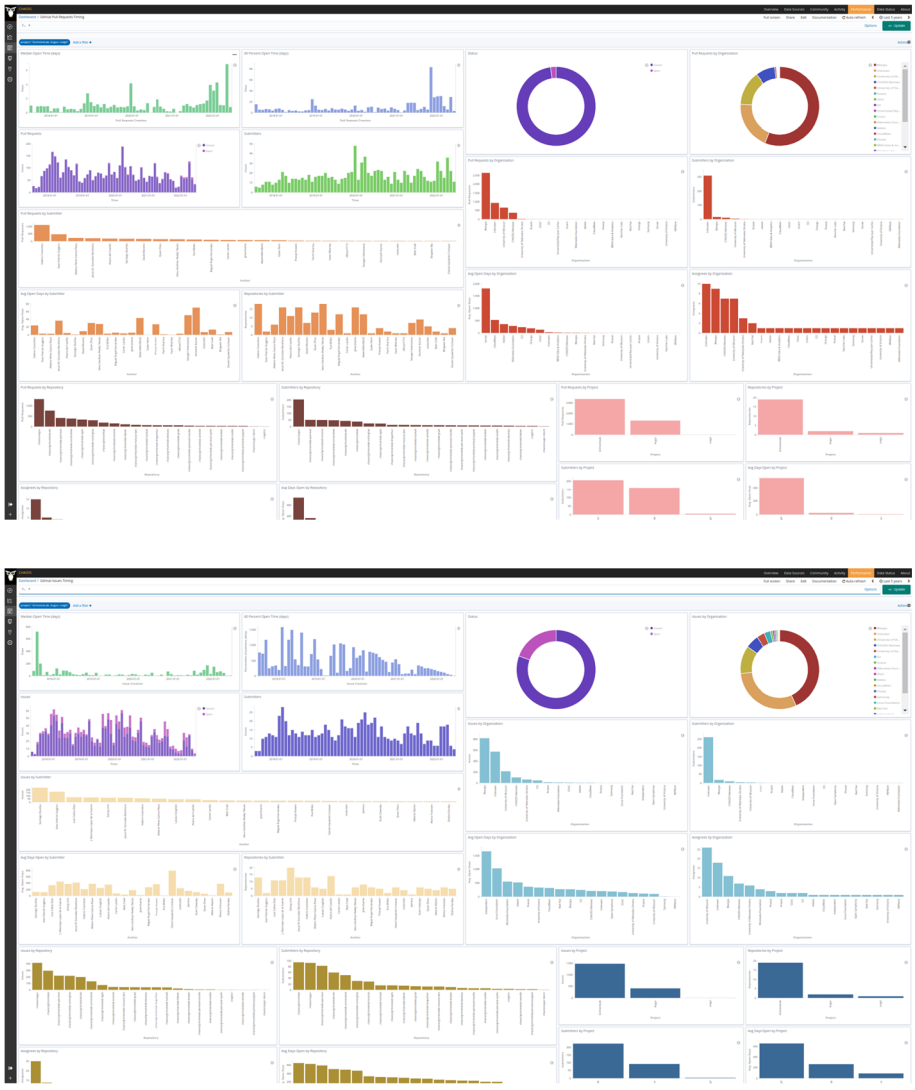

Fig. 3 Layout of the *Kibana* dashboards used for the experiment: Pull Requests Timing (top) and Issues Timing (bottom). Both are for the CHAOSS project as of late June 2022. While the data in the figure is not visible, the layout itself holds significance as it provides an understanding of the visual presentation and organization of the dashboards

particular, the complete set of dashboards for the CHAOSS project is maintained on-line as a demonstrator of the technology[4].

All *Kibana* dashboards allow to filter by a time range, as shown in Fig. 4. This is an important aspect of the dashboards, which will be used in our experiment, and which allows for comparison and tracking of the evolution of the metrics over time.

[4] CHAOSS Live Dashboard: http://chaoss.biterg.io/

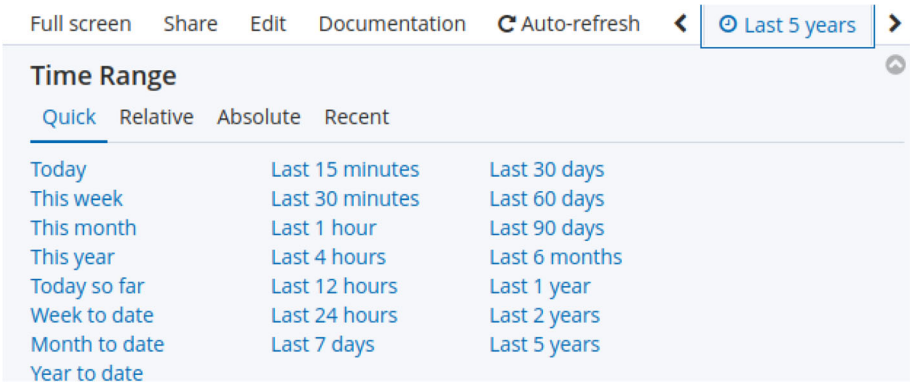

Fig. 4 Kibana time range selector

The two dashboards used in the experiment are an example of the out-of-the-box functionality provided by *GrimoireLab*, and by the Bitergia Analytics Platform. But to some extent, they are also an example of its limitations. One of their main problems (and of the other dashboards provided by the complete platform) is the difficulty in organizing visualizations, due to the lack of screen space. The visualizations are thoughtfully arranged within dashboards, specifically designed to explore specific aspects of a software project. The selection process involves choosing the most relevant visualizations that can be accommodated within a regular screen, allowing them to be conveniently viewed all at once. This deliberate design approach ensures that the key insights and information captured in the visualizations are effectively presented and easily accessible for analysis and interpretation. But there are many cases that require, to gain the needed understanding, moving back and forth between two or more dashboards, maybe applying different filters. In the case of our two dashboards, exploring the same aspect of pull requests and issues handling require moving between the two dashboards. Some examples of these aspects are:

- Organizational performance (number of pull requests or issues still open, time to close pull requests or issues).
- Project performance (number of pull requests or issues still open, time to close pull requests or issues).
- Relationship between time to answer pull requests or issues and the time to close them.
- Comparison of organizations contributing to a project by their pull requests or issues closing time, or existing open backlog (what they have opened).

In our experiment, we include some questions designed to learn about the subject performance in some of these cases.

2.3 VR Dashboard on *BabiaXR*

BabiaXR[5] (Moreno-Lumbreras et al. 2022) is a toolset for 3D data visualization in the browser. *BabiaXR* is based on *A-Frame*,[6] an open web framework written in JavaScript to build 3D scenes, suitable for VR and augmented reality in the browser. *A-Frame* extends `HTML` with

[5] *BabiaXR*: https://babiaxr.gitlab.io

[6] *A-Frame*: https://aframe.io

new entities allowing to build 3D scenes as if they were `HTML` documents, using techniques common to any front-end web developer. *A-Frame* is built on top of *Three.js*,[7] which uses the *WebGL* API available in all modern browsers.

BabiaXR extends *A-Frame* by providing components to create data visualizations, simplify data retrieval, and manage data (e.g., data filtering or mapping of fields to visualization features). Scenes built with *BabiaXR* can be displayed on-screen, or on VR devices, including consumer-grade headsets. Figure 5 shows a sample scene built with *BabiaXR*.

BabiaXR includes a component for retrieving data from *Elasticsearch*, with the most common queries and aggregations, providing a functionality similar to *Kibana* queries. Once the query is done, the data is parsed and formatted in the generic flat format that the *BabiaXR* visualizations use. Data is visualized in 3D by composing a single `HTML` document. It will include one or more of these data retrieval components (`babia-queryes`), and some other *BabiaXR* components that consume the retrieved data by building 3D visualizations, such as `babia-barsmap`, `babia-bars`, `babia-pie`, and `babia-doughnut`.

Once the HTML document is ready, it can be loaded in the browser of a VR device, and it will show the scene in immersed VR. Users will be able of exploring the scene by just moving their head and watching at different elements on it, or moving to approach the desired objects. This way of interaction allows for having much more objects around the user than those fitting a single screen, or even a small array of screens. In our experiment, we leveraged on this fact to produce a single scene including visualizations similar to those in the two Kibana dashboards that we described in the previous subsection. Thus, we display issues and pull request information in a single scene, shown in Fig. 6. For the scene, we have used the museum metaphor, placing elements like objects in a museum, arranged in shelves. We use two different color ranges (blue and red) to make it easier to distinguish between issue and pull request visualizations (similar colors were used in the Kibana dashboard).

For the experiment, we also developed a new component for *BabiaXR*, to mimic Kibana time range filters. This was implemented as a *Kibana*-like menu, linked on demand to one of the VR controllers, as shown in Fig. 7.

Figure 8 shows a screenshot of the VR dashboard presented to participants in the experiment, as seen from a corner of the "museum" room. Note that participants are placed at the beginning of the experiment in the middle of the room, and thus they have shelves around them, which they can view just by directing their gaze at the appropriate point. The complete set of visualizations shown in the scene corresponds to all the the visualizations shown in both Kibana dashboards used for the experiment.

2.4 Visualizations

Both in the Kibana and *BabiaXR* dashboards we included the same visualizations, related to the timing of issues and pull requests in the analyzed project. More specifically, the data is represented at various levels in different visualizations for issues and pull requests, that we refer together as "items" in the next listing:

- **At the level of the entire analyzed project**, showing the total number of items open and closed for the entire project, the total number of items open per organization, the median time of days open and the 80-percentil of days open for all items, the evolution of the total number of open and closed items over time, and the evolution of the number of submitters over time.

[7] *Three.js*: https://threejs.org

Fig. 5 Example of a *BabiaXR* Scene

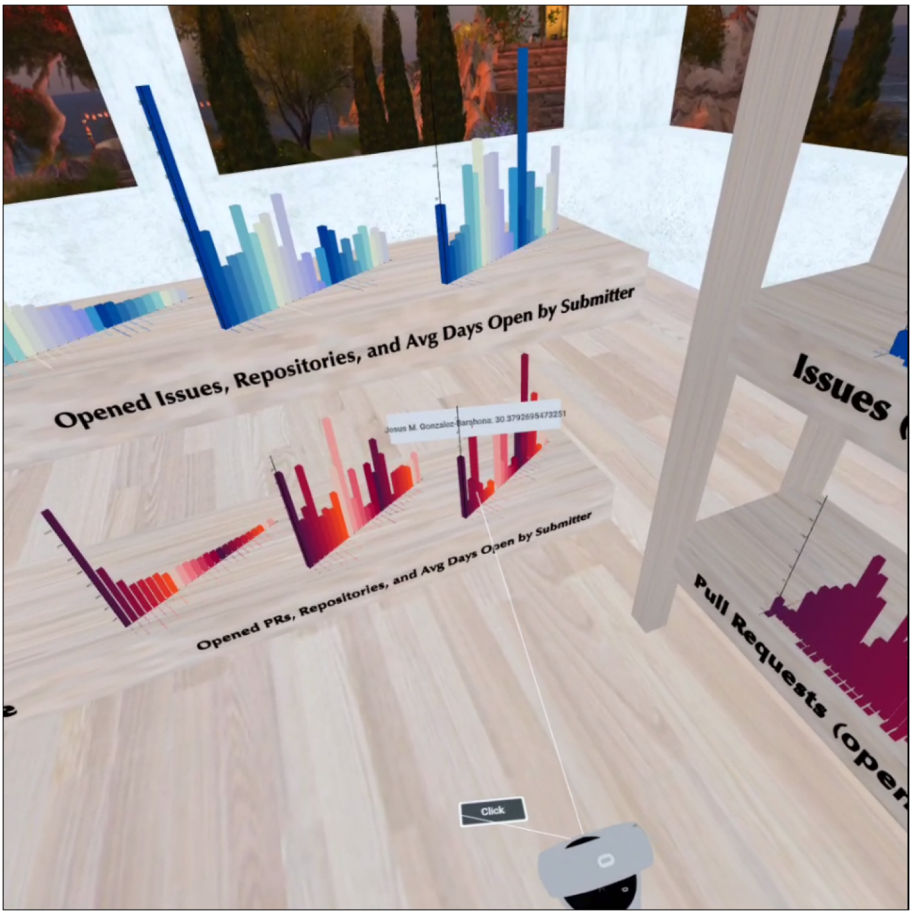

Fig. 6 Screenshot of a part of the VR dashboard scene used in our experiment, as seen with a VR device, showing pull request and issues data in different shelves

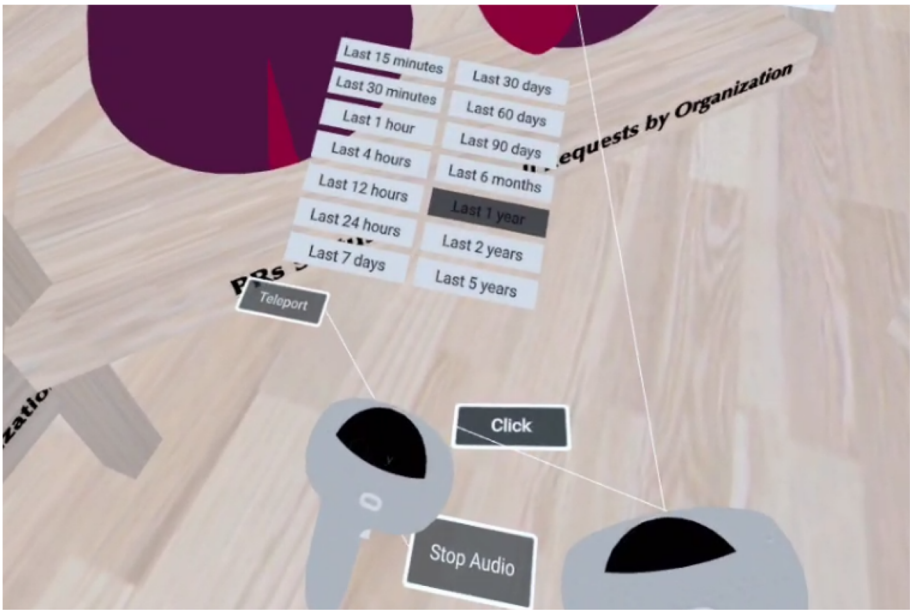

Fig. 7 *BabiaXR* time range selector

- **At the organization level**, showing the number of items per organization, the number of submitters per organization, the number of assignees per organization and the average number of days open for items per organization.
- **At the submitter level**, showing the number of items per submitter, the number of repositories per submitter and the average number of days open for items per submitter.

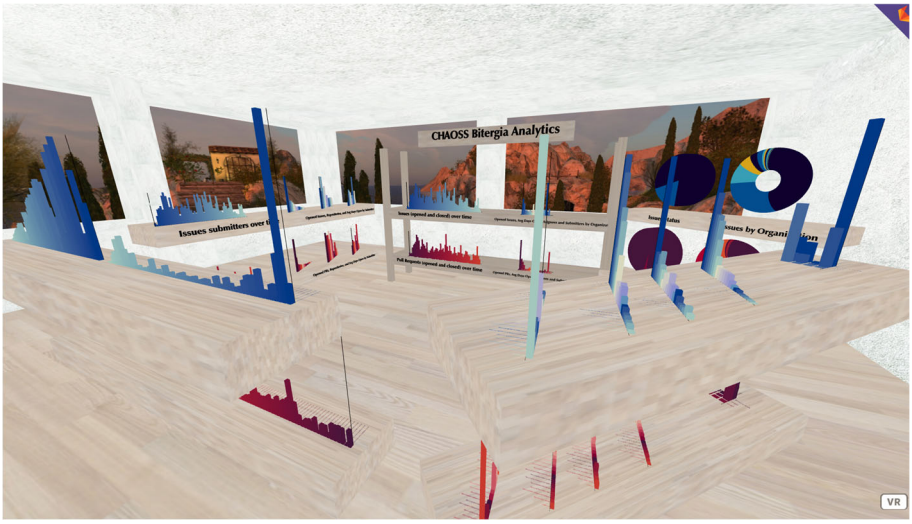

Fig. 8 A *BabiaXR* dashboard used in the experiment. Visualizations are in shelves, placed in a room mimiking a museum room. Data related to pull requests is visualized in bottom shelves, data related to issues is in top shelves

- **At the repository level**, showing the number of items per submitter, the number of submitters per repository, the number of assignees per repository, and the average number of days open for items per repository.
- **At the subproject level** (subprojects within the general project), showing the number of items for each subproject, the number of submitters per subproject, the number of repositories per subproject and the average number of days open for items per project.

Each visualization present in one environment has its namesake in the other environment, with the same title and similar size, color, and layout characteristics. The main difference between environments, as said, is that in *Kibana*, just as the dashboards defined by Bitergia, we have two scenes (one for pull requests and another one for issues), while with *BabiaXR* we have a single scene for all the visualizations.

3 Experiment

We adhered to the reporting guidelines proposed by Jedlitschka and Pfahl (2005) to ensure a comprehensive and transparent presentation of our experimental methodology. These guidelines provide a structured framework for reporting experimental studies, covering key aspects such as research design, participants, data collection, and statistical analysis. By following these guidelines, we aimed to enhance the clarity, replicability, and overall quality of our study.

3.1 Goal

The primary objective of the experiment is to assess whether the visualization of software development processes, specifically metrics related to modern code review and issue handling, is equivalently effective and satisfactory when presented in virtual reality (VR) scenes compared to traditional 2D screens.

3.2 Research Questions and Hypothesis

The main research question of our study is:

> RQ: *"Is comprehension of software development processes, via the visualization of their metrics, at least as good as in 2D screens when presented in VR scenes?"*.

This question tests the hypothesis that presenting visualizations in VR, where available space is much more abundant (you can have visualizations all around you, placing them in different heights), will allow for a better and faster understanding. The hypothesis is disputable because there are factors that work against it, such as the difficulties that perspective and distance may cause to the adequate perception of magnitudes.

For answering the research question, and validating (or not) the hypothesis, we will focus on specific, measurable aspects of the answers given by the subjects: accuracy (as a proxy for correctness), and time to completion (as a proxy for efficiency). Thus, we can refine our main RQ in two:

> RQ1: *"Do the answers obtained in VR provide similar correctness compared to those obtained on-screen?"*

RQ2: *"Do the answers obtained in VR provide similar time to completion compared to those obtained on-screen?"*

The correctness is measured by comparing the difference between the answer provided and the right answer, and the time spent in answering by the period of time needed to produce the answer in time units (i.e., in seconds).

3.3 Participants

Our experiment involved 32 subjects from both academia and industry. We divided participants randomly into four groups of eight people. Participants in each of the groups run first the experiment with data from one of the projects in one of the environments (on-screen or VR), and then with data from the other project in the other environment:

1. Group A was first presented with tasks in VR with data from *CHAOSS*, and then repeated the tasks on screen with data from *OpenShift*.
2. Group B was first presented with tasks on screen with data from *CHAOSS*, and then repeated the tasks in VR with data from *OpenShift*.
3. Group C was first presented with tasks in VR with data from *OpenShift*, and then repeated the tasks on screen with data from *CHAOSS*.
4. Group D was first presented with tasks on screen with data from *OpenShift*, and then repeated the tasks in VR with data from *CHAOSS*.

We employed the AB/BA crossover design for our experiment, which offers advantages in addressing the challenge of limited sample sizes and enhancing experimental sensitivity. In accordance with the guidelines outlined by Vegas et al. (2016), we adopted a crossover design framework that incorporated fixed factors such as period, sequence, and carryover effects. Crossover designs have gained recognition in software engineering experiments for their ability to mitigate confounding variables and increase statistical power by utilizing each participant as their own control. This approach allows for within-subject comparisons and effectively reduces the impact of individual differences, leading to more robust and reliable findings. However, it is important to acknowledge the potential challenges and considerations associated with implementing crossover designs in software engineering research. Factors such as washout periods, learning effects, and carryover effects should be carefully considered to ensure valid and meaningful results. By employing the AB/BA crossover design in our study, we aimed to maximize the efficiency of our experimental design and enhance the quality of the insights obtained.

Figure 9 summarizes demographics data for our participants, including age, gender, and job position, from academia and industry.

Figure 10 summarizes the experience level of participants in programming (i.e., `Exp PRG`), in using data visualization tools (i.e., `Exp Dataviz`), with the *CHAOSS* project (i.e., `Exp CHAOSS`), with the *OpenShift* project (i.e., `Exp OPENSHIFT`), in software visualization (i.e., `Exp Softvis`), and in using VR devices (i.e., `Exp VR`), as self-declared in an interview. Figure 11 summarizes the years of experience in programming (i.e., `Exp PRG`) and in using data visualization tools (i.e., `Exp Dataviz`).

None of the participants had ever used a *BabiaXR* visualization before, even if some of them had heard of it. Only one participant was *"a little"* familiar with *OpenShift*, meaning he was aware of the system, but had never used it.

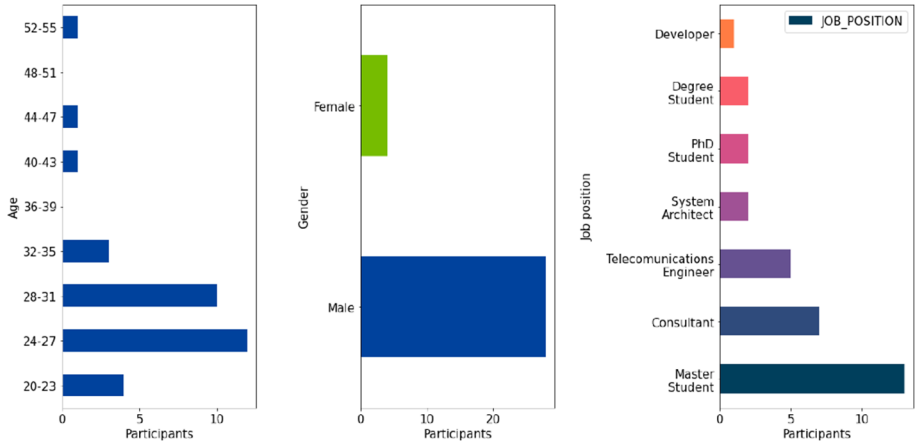

Fig. 9 Demographics of participants

3.4 Datasets and Tools

The datasets used in this experiment consist of data coming from two software projects: *CHAOSS* and *OpenShift*. *CHAOSS*[8], currently under the Linux Foundation umbrella, is a community devoted to produce software and metrics definitions related to software development, producing a mixture of software and documentation, in a relatively small community. *OpenShift*[9] is a large project, mainly devoted to produce software in the area of cloud computing, involving a very large development community. Even when the project is open source, it is also a commercial offering by Red Hat. We selected these projects since Bitergia has the data already analyzed and deployed in some Kibana dashboards already validated by customers.

Time ranges of both environment dashboards represents data of the last 5 years, following *Kibana* feature of selecting time ranges, being 5 years the highest selectable time range. We have developed the same funcionality in *BabiaXR* to be as fair as possible comparing results. Table 1 shows the main characteristics of these two projects. The tools used to retrieve data and produce the dashboards were described in Section 2.

3.5 Variables

The independent variable in our experiment is the group assignment, which is determined based on the order of the environment and project selected in the first round by each participant. The dependent variables in our study are the measurements derived from the performance of subjects in each task. Specifically, we are interested in assessing the correctness of their responses and the time taken to complete each task. These variables serve as indicators of participants' performance and provide valuable insights into the effectiveness and efficiency of their task execution.

- Independent variable:
  Name: Group

[8] *CHAOSS*: https://chaoss.community/

[9] *OpenShift*: https://docs.openshift.com/

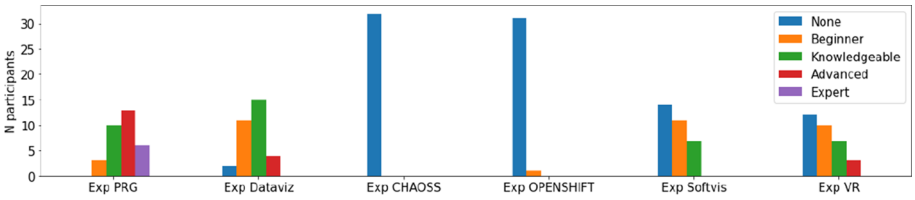

Fig. 10 Demographics: experience level

Description: Group that the subject belongs to.
Scale: Categorical: "First in VR with CHAOSS project", "First in on-screen with CHAOSS project", "First in VR with OpenShift project" or"First in on-screen with OpenShift project".
Operationalization: Subject answers after interacting with visualizations on a 2-D screen, or after interacting with visualizations immerse in virtual reality.

- Dependent variable "Time to complete":
  Name: TimeToComplete
  Description: Time to complete the task
  Scale: Integer (seconds)
  Operationalization: Number of seconds from the moment the subject states that the task starts, to the moment the subject states that the task is done.
- Dependent variable "Correctness":
  Name: Correctness
  Description: Normalized value in the range 0–1, with 1 being "completely correct", and 0 being "completely incorrect".
  Scale: Float, range 0–1
  Operationalization: Operationalized for each task, according to the correct answer and the specifics of the answer.

We also considered several confounding variables, to check for possible causes of bias, or other influencing factors:

- Confounding variable "Project":
  Name: Project
  Description: Project from which the data used in the task comes from.
  Scale: Categorical: "CHAOSS", "OpenShift".
  Operationalization: Determination by inspection of the specific task performed by the subject.
- Confounding variable "Period":
  Name: Period

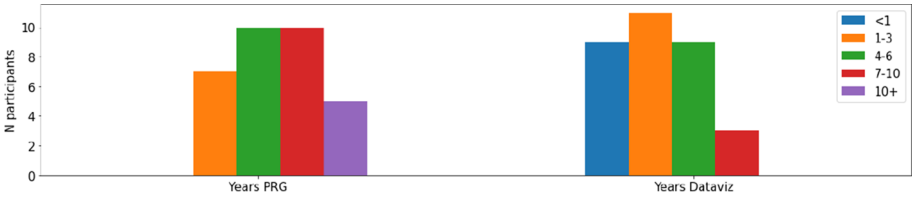

Fig. 11 Demographics: years of experience

Table 1 Main characteristics of *CHAOSS* and *OpenShift*

	CHAOSS	OpenShift
Issues and pull requests	9000+	215000+
Submitters	650+	7000+
Assignees	60+	1500+
Organizations	30+	80+0
Repositories	20+	350+7

Description: Period in which the subject performed the task (all subjects perform tasks in the first round in one environment, with one project, and then in the other environment with the other project).
Scale: Categorical: "First", "Second".
Operationalization: Determination by inspection of the specific task performed by the subject.

- Confounding variable "Environment":
  Name: Environment
  Description: Environment (setting) in which the question was answered by the subject
  Scale: Categorical: "on-screen" or "VR".
  Operationalization: Subject answers after interacting with visualizations on a 2-D screen, or after interacting with visualizations immerse in virtual reality.
- Confounding variable "Experience with *Kibana*":
  Name: ExpKibana
  Description: Overall experience with *Kibana* dashboards
  Scale: Categorical: "None", "Beginner", "Knowledgeable", "Advanced", "Expert".
  Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the categories as possible answers.
- Confounding variable "Experience in data visualization":
  Name: ExpDataviz
  Description: Overall experience in using data visualization tools
  Scale: Categorical: "None", "Beginner", "Knowledgeable", "Advanced", "Expert"
  Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the categories as possible answers.
- Confounding variable "Experience with VR":
  Name: ExpVR
  Description: Overall experience with VR devices
  Scale: Categorical: "'None", "Beginner", "Knowledgeable", "Advanced", "Expert"
  Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the categories as possible answers.
- Confounding variable "Experience in programming":
  Name: ExpPRG
  Description: Overall experience in programming
  Scale: Integer (years)
  Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the number of years of experience as answer.
- Confounding variable "Experience with CHAOSS":
  Name: ExpCHAOSS
  Description: Overall experience with the CHAOSS project

Scale: Categorical: "'None", "Beginner", "Knowledgeable", "Advanced", "Expert"
Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the categories as possible answers.

- Confounding variable "Experience with *OpenShift*":
  Name: ExpOpenShift
  Description: Overall experience with the *OpenShift* project
  Scale: Categorical: "'None", "Beginner", "Knowledgeable", "Advanced", "Expert"
  Operationalization: Self-estimation by the subject, via a question in the demographics survey, with the categories as possible answers.
- Confounding variable "Position":
  Name: JobPosition
  Description: Job position of the subject.
  Scale: Categorical: "Practitioner", "Academic", "Student"
  Operationalization: Self-declaration by the subject, via a question in the demographics survey, with open text as answer, which is later mapped to one of the categories.
- Confounding variable "Gender":
  Name: Gender
  Description: Self-perceived gender of the subject.
  Scale: Categorical: "Male", "Female", "Other"
  Operationalization: Self-declaration by the subject, via a question in the demographics survey, with open text as answer, which is later mapped to one of the categories.
- Confounding variable "Age":
  Name: Age
  Description: Age of the subject.
  Scale: Integer (years)
  Operationalization: Self-declaration by the subject, via a question in the demographics survey, with the number of years of age as answer.

Of the confounding variables, the following were found to be impossible to consider because of their very low diversity: experience with Kibana (none of the subjects had experience with it), experience with CHAOSS (none of the subjects had any), experience with *OpenShift* (only one of the subjects had some experience, and it was self-estimated as "Beginner"), and gender (only 4 subjects self-declared as female, for a total of 28 subjects that self-declared as male).

3.6 Training

To ensure a base level of familiarity with the systems, we added some training for both *BabiaXR* and *Kibana* environments. Before starting tasks in a certain environment, all participants were shown a training dashboard. In the case of the VR environment, the dashboard consists of a sample *BabiaXR* scene with the same visualizations shown in the experiment. In the case the on-screen environment, the training dashboard consisted of a sample *Kibana* dashboard with the same visualizations as the experiment dashboards. To mitigate the potential influence of prior knowledge on participants' responses and ensure unbiased results, the data represented in both training scenarios is sourced from a third project.

The training focuses on the following issues:

- **Interaction.** In both trainings the participant learned how to interact with the visualizations, in order to obtain the maximum information from them (e.g., pointing to the

visualization). In the case of the on-screen training, the limitations of the use of *Kibana* were also explained, avoiding features that the *BabiaXR* dashboard does not provide (e.g., field filtering by clicking).

- **Movement.** In both trainings the participant learns how to move around in the dashboard. In the case of the VR environment, the participant learns how to walk and uses the teleport feature for moving to the visualizations around. In the case of the on-screen environment, the participant learns how to move through the dashboards and how to reach all the visualizations that are not available at first sight.
- **Time range switch.** In both trainings the participant learned how to change the time range for the data, an important action for the experiment. In the case of the VR environment, the participant learns how to change the time range using the controllers, by hiding/showing the corresponding options, and clicking on them. In the case of the on-screen environment, the participant learns where the time range option is located, and is informed about the only ones permitted in the experiment (*Kibana* has more time range options than *BabiaXR*).

3.7 Tasks and Data Collection

Subjects perform five different tasks in each environment (VR and on-screen), totaling 10 tasks per subject. To design our tasks, we leveraged the maintenance task definition framework by Sillito et al. (2006). Table 2 summarizes the five tasks that participants had to address in each environment. We collect data for each task as follows:

- **Answers.** To provide answers to questions in each task, all participants had to speak aloud. For each task, participants were also asked to assess the level of difficulty (five levels, from "Strongly Disagree" to "Strongly Agree").
- **Efficiency.** The supervisor tracked the time that each participant spent on each task. For each task, the participant notified the supervisor both the start and the finish moments.
- **Correctness.** After running the experiment, the supervisor checked the answers of the task, comparing them with the correct values. This check was validated by one of the authors of the paper.

For the whole experiment, we also collected **Feedback** from participants. After finishing the experiment, participants answered a set of feedback and control questions that gave us qualitative results on how the visualizations support users in locating key parts of the development processes. Table 3 summarizes the questions presented in the feedback survey.

To ensure the smooth execution and refinement of the experiment, we conducted two dry-runs prior to the actual data collection. These dry-runs allowed us to simulate the experiment in a controlled setting and identify any potential issues or areas for improvement. We carefully reviewed the experimental procedures, the setup of the virtual reality environment, the data collection instruments, and the instructions provided to participants. During the dry-runs, we invited a small group of individuals who were not part of the participant pool to participate in the experiment. This allowed us to observe their interactions with the experimental setup, identify any ambiguities or difficulties they encountered, and make necessary adjustments to the experimental design, tasks, and instructions. The feedback and observations from the dry-runs were invaluable in refining the experiment. We addressed minor issues, clarified instructions, and made adjustments to the virtual reality environment to ensure a more seamless and user-friendly experience for the participants. Additionally, the dry-runs helped us estimate the time required for each task and allowed us to fine-tune the overall experimental

Table 2 Tasks list of the experiment

Task	Task description & purpose	Category
T_1	**Description.** During the LAST YEAR tell me: The name of the TOP 3 ORGANIZATIONS by number of issues. For each of those 3 organizations, the NUMBER of pull requests for the same period.	Correlation
	Purpose. Identify the most important organizations of the project in terms of the number of pull request and issues and identify if the correlation between issues and pull request is meaningful at the Organization level.	
T_2	**Description.** For issues, during the LAST 90 DAYS, tell me: The number of opened and closed and when is the higher time open? (time open as median in days) For pull request, during the LAST 90 DAYS, tell me: The number of opened and closed and when is the higher time open? (time open as median in days)	Analysis
	Purpose. The quarter is a common measurement system, so the purpose is to know the number of issues and pull request of the entire project in that quarter, identifying the highest point that remained open.	
T_3	**Description.** During the LAST 5 YEARS: For pull requests submitters, who are the top three?. For each of them, for how long their issues stayed open (days on average)	Correlation
	Purpose. Identify the most important submitters of the project in terms of the number of pull request and identify if the correlation between the pull request submissions and the time that the issues remain open is meaningful at the submitter level.	
T_4	**Description.** During the LAST 2 YEARS tell me: The name of the TOP 3 REPOSITORIES by number of pull requests SUBMITTERS. The name of the TOP 3 REPOSITORIES by number of issues.	Analysis
	Purpose. Identify the most important repositories of the project in terms of the number of pull request Submitters and issues and identify which repositories are receiving the most activity in terms of different submitters. Compare the results with the repositories with more issues.	
T_5	**Description.** During the LAST 6 MONTHS, tell me: The name of the TOP 3 SUBMITTERS by the longest time to resolve their issues (average days open). For each of those 3 submitters, the NUMBER of pull requests submitted.	Correlation
	Purpose. Identify the core of the community in the last 6 months, identify who are the users for whom their issues stayed longer opened, and compare this with the number of pull requests that they submit.	

timeline. By conducting these dry-runs, we were able to identify and address potential issues proactively, resulting in a more robust and well-prepared experiment. The insights gained from the dry-runs significantly contributed to the smooth execution of the actual data collection phase and increased the validity and reliability of the findings. We believe that the inclusion of these dry-runs in our experimental process demonstrates our commitment to rigorous methodology and the careful refinement of our study design.

4 Changes from the Registered Report

In our experiment and the subsequent analysis, we have followed with great detail the execution plan, characteristics, and design proposed in the corresponding registered

Table 3 Feedback and control questions

ID	Description
S_1	In what environment did you find it easier to complete the tasks? Why?
S_2	In what environment has it taken you the shortest to complete tasks (what is your feeling)? Why?
S_3	Tell us which parts of each environment are useful to answer the tasks, and tell us which parts make it more difficult to answer the tasks. (Advantages and disadvantages)
C_1	Overall, did you find the experiment difficult? (choose one: strongly agree, agree, don't know, disagree, strongly disagree) Please explain.
C_2	Do you have any suggestions or comments?

report (Moreno-Lumbreras et al. 2021). However, there are some significant changes, which we detail below:

- **Dashboards**. In the registered report we defined a set of five *Kibana* dashboards developed by Bitergia that analyze software development processes, which we intended to use for the experiment. Finally, for this study, we decided to focus only on two dashboards of that set, those designed to explore the timing of pull requests and issues. When designing the details of the execution, we realized that using all five dashboards meant a longer experiment, and the risk of spreading shallow over a large number of aspects of the project. Instead, we decided to reduce the number of dashboards and focus the study on the analysis of two similar and usually related software development processes: timing of pull requests and issues.
- **Tasks**. In the registered report we stated that we would define a set of tasks, and each subject would be presented with half those tasks in one environment (on-screen or VR), and the other half in the other, with data from different projects in each environment. All tasks would be different, and a single subject would not repeat a task in both environments. However, when defining the final version of the experiment, we decided not to divide the set of tasks, but to repeat them for each subject in the second environment, with data from a different project. The main reason is that this way we can analyze in depth if having performed tasks in one system serves as learning when repeating it in the other environment, and thus has some impact on performance. In addition, we decided that having two different sets of tasks could be detrimental when comparing results for a sample of participants that is not very large (in our case, 32 participants).
- **Projects**. Even when we originally planned to use data from two similar projects, finally we decided to have two very different cases. The main reason was that, not interfering with the results, thus would allow us to analyze differences (if any) due to the different characteristics of the projects. All participants are assigned to both projects in a random order, and in random environments, so we can still analyze the overall impact of both environments, which is the main aim of the study. However, we can also control for differences in performance due to the nature of the projects. In particular, we wanted to check if the very different amount of data to visualize for each of the projects produced any measurable difference.
- **Correctness dependent variable**. In the registered report, we defined "Error" as one of the two dependent variables. However, when designing the final version of the study, we decided to study correctness instead of error. In the end, both variables capture the same

characteristic (how good the answer provided by the subject is, when comparing it with the correct answer). But we found that the analysis seemed more natural when mapping answers to a scale in the range 0–1, according to how close they were to the true value than estimating the error.

- **Confounding variables**. For the final version of the analysis, we performed a more detailed analysis of confounding variables, and we decided to make some changes to the list proposed in the registered report. On the one hand, we substituted "Experience in software development" with a more ample variable, "Experience in programming", which should capture the same kind of abilities, but is better suited to our expected demography of participants. On the other hand, we added several new confounding variables that we thought could have an impact:
  - Variables related to the experiment: "Project" and "Round". This way, we could analyze the impact of the characteristics of the project, and the possible "learning effect".
  - Variables related to previous experience: In addition to "Experience with *Kibana*" and "Experience with VR", already present in the registered report, and "Experience in programming" (substituting "Experience in software development"), we include "Experience with data visualization" (to check more broadly for experience with tools related to the experiment), and "Experience with CHAOSS" and "Experience with OpenShift" (to discard the bias of subjects that could already be familiar with the data shown in the experiment, or with the underlying projects).
  - Demography variables: In addition to "Position", already present in the registered report, we added "Gender" (although we could not analyze it, because we lack enough diversity in subjects for this dimension), and "Age", as potential causes of bias.

- **Analysis of confounding variables**. Instead of presenting a separate analysis for all confounding variables, we focused on "Project" and "Round", because we considered that they could be the more determinant for the results of the experiment, due to the way in which the experiment itself was decided. Therefore, in Section 5 (devoted to presenting results), the first sections analyze dependent variables in the context of these two confounding variables, with great detail. The rest of the confounding variables are analyzed together, more briefly, in Section 5.3.
- **Training**. In the original report we designed a process that included some training for subjects before they performed the tasks in the experiment. However, when designing the final version, we decided to make more emphasis on this training, making it more specific and a bit longer than initially intended. The reason to do so was because of our experience in other experiments, where we learned the importance of letting people understand the basic mechanisms needed to perform the tasks, so that we could exclude most of the learning curve from the experiment itself.

5 Results

This section summarizes the results of the experiment with respect to the "Correctness" (see Section 5.1) and "Time to completion" (Section 5.2) dependent variables. These variables also correspond to the two research questions ($\mathbf{RQ_1}$ and $\mathbf{RQ_2}$) of our study. In both cases, the dependent variables are analyzed not only by themselves, but also in the context of the confounding variables related to the experiment itself: "Project" and "Round". The influence

of confounding variables is shown by presenting the results for our four random groups of subjects (see Section 3.3), each corresponding to a combination of a "Project" and "Round" category.

5.1 Correctness (RQ_1)

For answering $\mathbf{RQ_1}$, we analyzed how correct were the answers that participants in the experiment provided for $\mathbf{T_1}$ – $\mathbf{T_5}$. We observed that errors in reporting were only a few, and it was very rare that a subject failed more than, for example, one number when asked for three. So, used a simple mapping of results to the value of correctness, based on the following criteria, which tries to ensure that 0 is mapped for big mistakes (which render all results false), and then any error discounts from the highest correctness value possible (1). For all tasks, we followed these criteria:

- 0: If the time range is wrongly selected (all tasks require the subject to correctly select a time range).
- 1: If everything is correct.

In addition, for each specific task:

- $\mathbf{T_1}$ asks to report the top three organizations by number of issues, and then for those organizations the number of pull requests submitted. Values for correctness:
  - 0.5: If all three organizations are identified correctly, but the number of pull requests is wrong, or if at least one of the organizations was wrong, but the number of pull requests for the identified organizations is correct.
- $\mathbf{T_2}$ asks to report four items: the number of opened and closed issues (1), and when is the highest value in the "time open as median in days" visualization (2), and then the same for pull requests instead of issues (3,4). Values for correctness:
  - 0.25: If only one item is correct.
  - 0.5: If two items are correct.
  - 0.75: if three items are correct.
- $\mathbf{T_3}$ asks first to report the top three submitters by number of pull requests during a certain period, and then for those submitters the number of days on average that their issues stayed open during the period. Values for correctness:
  - 0.5: If all three submitters are correctly reported but the number of days open for their issues was wrong, or if at least one identified submitter is wrong but the number of days open for the issues of the identified submitters is correct.
- $\mathbf{T_4}$ first asks to report the top three repositories by number of pull request submitters during a certain period, and then the top three repositories by number of issues during the same period. Values for correctness:
  - 0.5: If all three repositories by number of pull request submitters were correctly reported, but the top three repositories by number of issues were wrong, or the other way around.
- $\mathbf{T_5}$ asked to report the top three submitters by the longest time to resolve their submitted issues during a certain period, and then for those submitters, the number of pull requests submitted during the same period. Values for correctness:

– 0.5: If the three submitters were correctly reported but their number of pull requests was wrong, or the other way around

Figure 12 summarizes the correctness for all the tasks $\mathbf{T_1}$ – $\mathbf{T_5}$ for all participants, for both round $\mathbf{P_1}$ and round $\mathbf{P_2}$. We can observe that there are several participants that have the maximum correctness for all the tasks, and only three participants have less than 0.5 in one task. Thus, we can infer that the environments were good enough for resolving the set of tasks proposed.

5.1.1 Results by Group of Subjects

Figure 13 summarizes the average of the correctness values of the four groups of participants, for all tasks for both rounds. This analysis helps to understand the effect of the confounding variables "Round" and "Project" (although for "Project" we will present a more detailed subsection below).

Figure 13 also shows how the difference between the different combinations of environments and projects is not high, since the average correctness values are all above 0.8. Comparing correctness for our two rounds, we can see that the second round of experiments has similar or better correctness in general (e.g., in $\mathbf{T_4}$, the second round of the experiment has better correctness). Thus, there is some learning with respect to correctness, but not very acute. These results also show that VR participants have consistently answered with the same accuracy when compared to on-screen participants.

Despite the different results, for all 5 tasks in both rounds, both VR and on-screen participants provided similar answers with respect to the perceived difficulty of the tasks (see Fig. 14). We can also observe in this figure how the perceived difficulty decreases as participants resolve more tasks, being the tasks of the second round found easier by all participants, except for an isolated case that strongly agreed on the difficulty of $\mathbf{P_2}$ $\mathbf{T_4}$ (in this case, in the VR environment), while in $\mathbf{P_1}$ $\mathbf{T_4}$ none did.

Given the specific design of our experiment, which follows a Two-treatment factorial crossover design with the experimental object being a two-level blocking variable , we opted

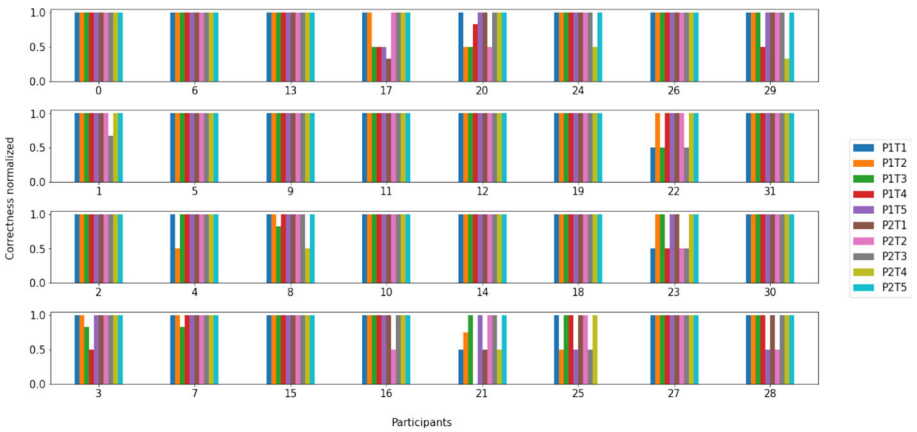

Fig. 12 Correctness of all participants (by participant identifier) for each task. Colors correspond to the different tasks in the experiment, which can be mapped to task identifiers in the legend: $\mathbf{P_x}$ is the identifier of the round and $\mathbf{T_x}$ is the identifier of the task

Fig. 13 Average correctness by group. The color legend shows the group of participants

to utilize a mixed linear model for our analysis, as suggsted in Vegas et al. (2016). This type of model is well-suited for capturing the within-subject dependencies and accounting for the blocking variable, in this case, the group. By employing a mixed linear model, we are able to examine the effects of the different treatments while considering the inherent variability and correlations within the data. This approach allows us to assess the impact of the independent variables on the dependent variable while accounting for the unique characteristics of our experimental design.

We also analyzed Cohen's effect size. Cohen's effect size (Cohen 1988), specifically Cohen's d, was selected as a measure of the magnitude of the differences between the groups of participants in our analysis. Cohen's d is a widely used effect size measure that quantifies the standardized difference between means. By calculating Cohen's d, we obtain a standardized value that allows for meaningful comparisons across different studies and variables. The use of effect size measures like Cohen's d provides valuable information about the practical significance or importance of the observed differences. It helps us move beyond statistical significance alone and provides a clearer understanding of the magnitude of the effects. The choice of Cohen's d as the effect size measure allows us to communicate the practical relevance of the differences between the groups of participants in a standardized and interpretable manner. Table 4 depicts the ranges for the Effect Size (Cohen 1988).

Table 5 depicts the results of the statistical analysis for the average of the First and Second Round (P1 and P2). Based on the mixed linear model regression results and the calculated Cohen's d values, we can gain insights into the differences between the groups of participants with *P1 SC Chaoss - P2 VR Openshift* as the reference category.

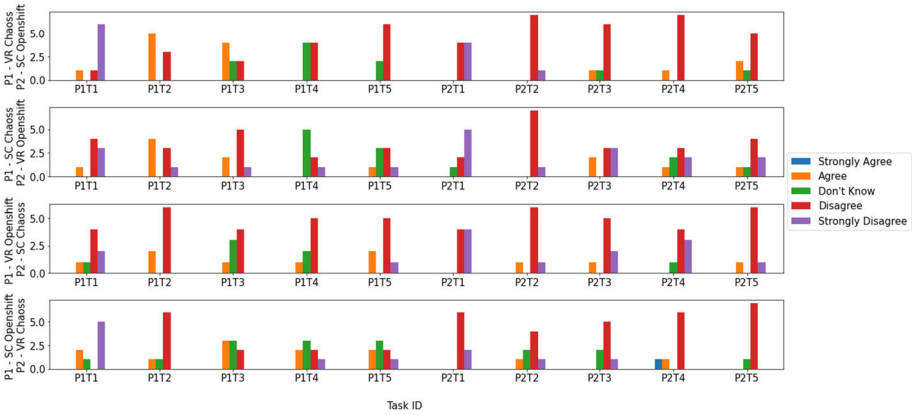

Fig. 14 Answers to "Did You Find the Task Difficult?". Each row of charts corresponds to a group of subjects, and each chart to a task in a round

Table 4 Effect size of different ranges of Cohen's delta

Effect size	Delta
Small	$\|d\| \approx 0.2$
Medium	$\|d\| \approx 0.5$
Large	$\|d\| \approx 0.8$ or higher

- **Average on Round 1.** For *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of -0.076 suggests that, on average, the P1_AVG for this group is 0.076 units lower than the reference group. This difference is statistically significant ($p < 0.05$), as indicated by the corresponding p-value. Similarly, for *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of -0.054 suggests that, on average, the P1_AVG for this group is 0.054 units lower than the reference group. This difference is also statistically significant ($p < 0.05$). On the other hand, for *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of -0.016 suggests that there is no statistically significant difference in the P1_AVG between this group and the reference group ($p > 0.05$).
  The module of the values of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (-1.5946) and "*P1 VR Chaoss - P2 SC Openshift* vs *P1 SC Chaoss - P2 VR Openshift*" (-1.4845) indicate large effect sizes, suggesting substantial differences between these groups in terms of P1_AVG. On the other hand, Cohen's d value for "*P1 VR Openshift - P2 SC Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (-0.3041) suggests a small effect size, indicating a smaller and less substantial difference between these groups.
  Overall, these results suggest that there are statistically significant differences in P1_AVG between the groups of participants compared to the reference group. The effect sizes, as measured by Cohen's d, indicate that the differences are particularly notable for *P1 SC Openshift - P2 VR Chaoss* and *P1 VR Chaoss - P2 SC Openshift* groups, while the difference for *P1 VR Openshift - P2 SC Chaoss* is relatively smaller.
- **Average on Round 2.** For *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of -0.066 suggests that, on average, the P2_AVG for this group is 0.066 units lower than the reference group. This difference is statistically significant ($p < 0.05$). For *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of -0.036 suggests that, on average, the P2_AVG for this group is 0.036 units lower than the reference group. Although the p-value (0.069) is slightly above the significance threshold ($p = 0.05$), it is still worth

Table 5 Results of the Mixed linear method applied to the correctness

Group	P1 AVG			P2 AVG		
	Coefficient	p-value	Eff Size	Coefficient	p-value	Eff size
P1 - SC Openshift	−0.076	0.024	−1.59	−0.066	0.025	−1.58
P2 - VR Chaoss						
P1 - VR Chaoss	−0.054	0.036	−1.48	−0.036	0.069	−1.28
P2 - SC Openshift						
P1 - VR Openshift	−0.016	0.667	−0.30	−0.016	0.551	−0.42
P2 - SC Chaoss						

P1 - SC Chaoss, P2 - VR Openshift as the reference category. For the average of the correctness of the first and second round

noting as it approaches statistical significance. For *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of -0.016 suggests that there is no statistically significant difference in the P2_AVG between this group and the reference group ($p > 0.05$).
The module of the values of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (-1.5858) and "*P1 VR Chaoss - P2 SC Openshift* vs *P1 SC Chaoss - P2 VR Openshift*" (-1.2854) indicate large effect sizes, suggesting substantial differences between these groups in terms of P2_AVG. Cohen's d value for "*P1 VR Openshift - P2 SC Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (-0.4211) suggests a moderate effect size, indicating a relatively smaller but still noticeable difference between these groups.
Overall, these results suggest that there are statistically significant differences in P2_AVG between the groups of participants compared to the reference group. The effect sizes, as measured by Cohen's d, indicate that the differences are particularly notable for *P1 SC Openshift - P2 VR Chaoss* and *P1 VR Chaoss - P2 SC Openshift* groups, while the difference for *P1 VR Openshift - P2 SC Chaoss* is relatively smaller but still observable.

Focusing now in Table 6, we analyze the average of the 2 rounds. For *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of -0.072 suggests that, on average, the TOTAL_AVG for this group is 0.072 units lower than the reference group. This difference is statistically significant ($p < 0.05$). For *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of -0.047 suggests that, on average, the TOTAL_AVG for this group is 0.047 units lower than the reference group. This difference is also statistically significant ($p < 0.05$). For *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of -0.017 and p-value of 0.665 suggest that there is no statistically significant difference in the TOTAL_AVG between this group and the reference group ($p > 0.05$).
The module of the values of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs*P1 VR Chaoss - P2 SC Openshift*" (-1.6842) and "*P1 VR Chaoss - P2 SC Openshift* vs *P1 VR Chaoss - P2 SC Openshift*" (-1.6312) indicate large effect sizes, suggesting substantial differences between these groups in terms of TOTAL_AVG. Cohen's d value for "*P1 VR Openshift - P2 SC Chaoss* vs *P1 VR Chaoss - P2 SC Openshift*" (-0.3062) suggests a small effect size, indicating a relatively smaller difference between these groups.

Overall, these results suggest that there are statistically significant differences in TOTAL_AVG between the groups of participants compared to the reference group. The effect sizes, as measured by Cohen's d, indicate that the differences are particularly notable for *P1 SC Openshift - P2 VR Chaoss* and *P1 VR Chaoss - P2 SC Openshift* groups, while the difference for *P1 VR Openshift - P2 SC Chaoss* is relatively smaller but still observable.

Table 6 Results of the Mixed linear method applied to the correctness

	TOTAL AVG		
Group	Coefficient	p-value	Eff size
P1 - SC Openshift	− 0.072	0.017	− 1.68
P2 - VR Chaoss			
P1 - VR Chaoss	− 0.047	0.021	− 1.63
P2 - SC Openshift			
P1 - VR Openshift	− 0.017	0.665	− 0.30
P2 - SC Chaoss			

P1 - SC Chaoss, P2 - VR Openshift as the reference category. For the average of the total correctness

5.2 Completion Time (RQ_2)

To answer $\mathbf{RQ_2}$ we analyze the dependent variable "Time to completion". We will analyze it by project and environment together (by group), by environment (VR or on-screen), and by project. We consider as "time to completion" the time from the moment the subject states that the task is starting, to the moment the subject states that the task is done (in seconds).

5.2.1 Results by Group of Subjects

We analyzed time to completion for the tasks $\mathbf{T_1}$ – $\mathbf{T_5}$ in each round, for each of the four groups of subjects. Figure 15 summarizes results of this analysis, presented as box plots by group. If we focus on a given project and the subjects who start in the VR environment, we can see that their time to completion is a bit slower than those who start on-screen. However, the difference appears not to be significant. This difference is even smaller when participants face the tasks during the second round, regardless of the environment.

Figure 16 shows the difference between times for each task in both rounds, per group. In it, the times presented by each boxplot are the subtraction of the completion time during the first round minus the completion time during the second round, for each task. In this figure, most of the values are positive, which means that most of the participants spent less time to complete tasks during the second round than during the first round, regardless of the environment. We expected this, due to the learning process (previous knowledge adjured in the first round). Again, participants who started with data from the *OpenShift* project spent more time in the first round than participants who started with data from the *CHAOSS* project. In addition, we can observe that, as participants solve more and more tasks, the time difference becomes smaller, reaching almost the same values in some cases (e.g., $\mathbf{T_5}$ for those who started in VR with *OpenShift* and continued on-screen with *CHAOSS*).

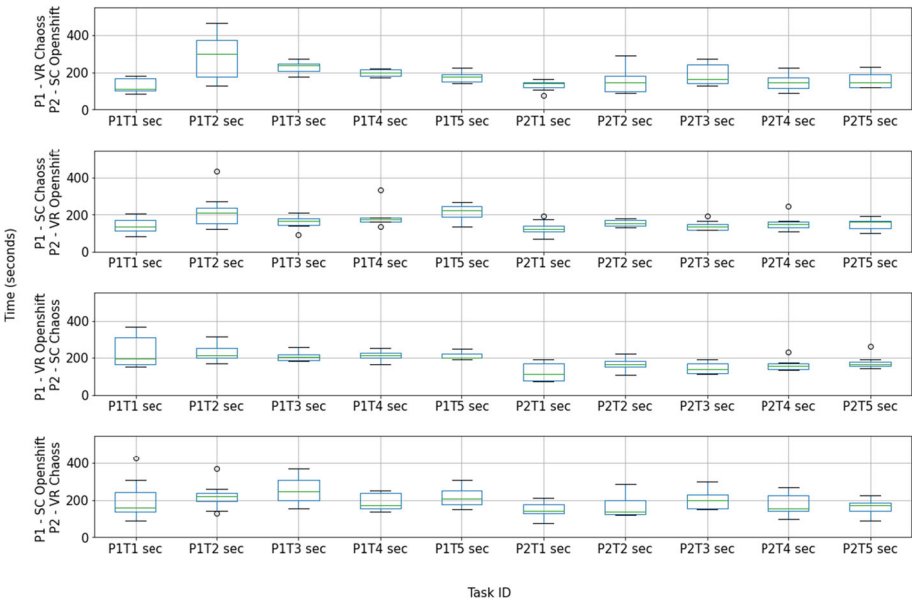

Fig. 15 Distribution of time to completion (in seconds) by group

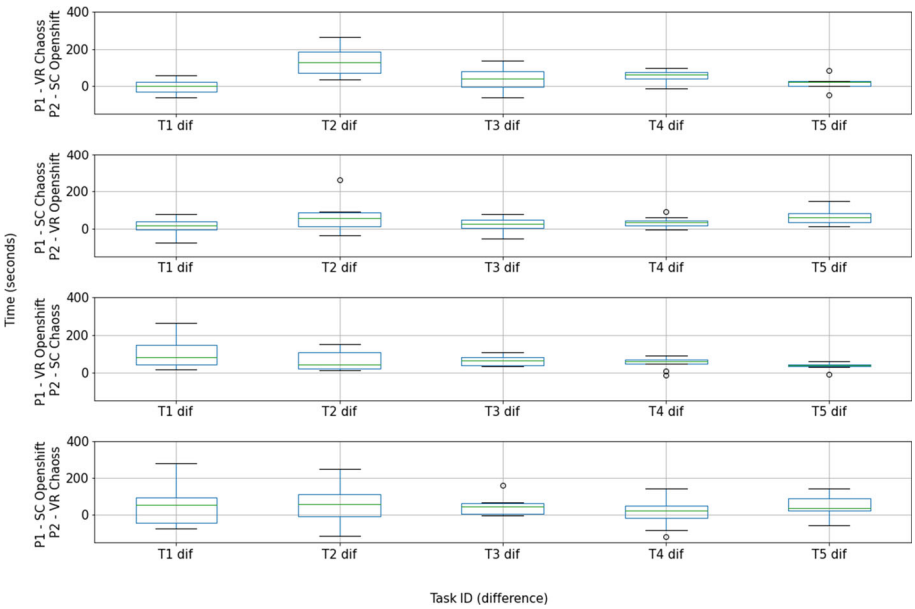

Fig. 16 Distribution of differences in time to completion between rounds (in seconds) by group

In order to further investigate the potential differences in terms of time to completion of the Rounds among different groups, we will employ again a mixed linear model analysis. This approach allows us to account for the nested structure of the data and control for potential confounding factors. By fitting the mixed linear model, we can estimate the effects of the different groups while considering the variability within and between groups. Additionally, we will calculate Cohen's d effect size to quantify the magnitude of the observed differences, allowing us to compare the magnitude of differences in time to completion between groups. By incorporating both the mixed linear model and Cohen's d, we aim to gain a comprehensive understanding of the potential significant differences in time to completion among the different groups.

Table 7 depicts the results of the statistical analysis for the total times of the First and Second Round (P1 and P2). Based on the mixed linear model regression results and the

Table 7 Results of the Mixed linear method applied to the time to completion

	P1 TOTAL			P2 TOTAL		
Group	Coefficient	p-value	Eff size	Coefficient	p-value	Eff size
P1 - SC Openshift P2 - VR Chaoss	152.625	0.091	1.19	137.5006	0.005	1.99
P1 - VR Chaoss P2 - SC Openshift	89.750	0.316	0.71	47.125	0.476	0.50
P1 - VR Openshift P2 - SC Chaoss	163.750	0.061	1.32	43.375	0.518	0.45

P1 - SC Chaoss, P2 - VR Openshift as the reference category. For the total time of the first and second round

calculated Cohen's d values, we can gain insights into the differences between the groups of participants with *P1 SC Chaoss - P2 VR Openshift* as the reference category.

- **Total times on Round 1.** For *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of 152.625 suggests that the P1 TOTAL time for this group is 152.625 units higher than the reference group. However, this difference is not statistically significant ($p > 0.05$). For *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of 89.750 suggests that the P1 TOTAL time for this group is 89.750 units higher than the reference group. Again, this difference is not statistically significant ($p > 0.05$). For *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of 163.750 suggests that the P1 TOTAL time for this group is 163.750 units higher than the reference group. Although this difference shows a trend towards significance ($p = 0.061$), it does not reach the conventional threshold for statistical significance ($p > 0.05$).
  The module of the values of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (1.1969) and "*P1 VR Openshift - P2 SC Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (1.3249) suggest moderate effect sizes, indicating observable differences between these groups in terms of P1 TOTAL times. Cohen's d value for "*P1 VR Chaoss - P2 SC Openshift* vs *P1 SC Chaoss - P2 VR Openshift*" (0.7092) indicates a smaller effect size, suggesting a relatively smaller difference between these groups.
  Overall, these results suggest that there may be some differences in P1 TOTAL times between the groups of participants compared to the reference group. However, the statistical significance of these differences is limited, with only the difference for *P1 VR Openshift - P2 SC Chaoss* showing a trend towards significance. The effect sizes, as measured by Cohen's d, suggest that the differences, if present, are moderate in magnitude for *P1 SC Openshift - P2 VR Chaoss* and *P1 VR Openshift - P2 SC Chaoss*, while the difference for "*P1 VR Chaoss - P2 SC Openshift*" is relatively smaller.
- **Total times on Round 2.** For *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of 137.500 suggests that the P2 TOTAL time for this group is 137.500 units higher than the reference group. This difference is statistically significant ($p < 0.05$), indicating that there is a significant effect of *P1 SC Openshift - P2 VR Chaoss* on the P2 TOTAL time. For *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of 47.125 suggests that the P2 TOTAL time for this group is 47.125 units higher than the reference group. However, this difference is not statistically significant ($p > 0.05$), indicating that there is no significant effect of *P1 VR Chaoss - P2 SC Openshift* on the P2 TOTAL time. For *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of 43.375 suggests that the P2 TOTAL time for this group is 43.375 units higher than the reference group. Again, this difference is not statistically significant ($p > 0.05$), indicating that there is no significant effect of *P1 VR Openshift - P2 SC Chaoss* on the P2 TOTAL time.
  The positive value of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (1.9909) suggests a large effect size, indicating a substantial difference between these groups in terms of P2 TOTAL time. Cohen's d values for "*P1 VR Chaoss - P2 SC Openshift* vs *P1 SC Chaoss - P2 VR Openshift*" (0.5039) and "*P1 VR Openshift - P2 SC Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (0.4570) indicate smaller effect sizes, suggesting relatively smaller differences between these groups.
  In summary, these results suggest that there is a statistically significant difference in P2 TOTAL time between the group "*P1 SC Openshift - P2 VR Chaoss*" and the reference group "*P1 SC Chaoss - P2 VR Openshift*". The effect size, as measured by Cohen's d, indicates a large difference between these two groups. However, there are no significant differences in P2 TOTAL time between the groups *P1 VR Chaoss - P2 SC Openshift* and

"*P1 VR Openshift - P2 SC Chaoss*" compared to the reference group. The effect sizes for these comparisons are relatively smaller.

Table 8 depicts the results of the analysis. Computing the total time of the two rounds, for *P1 SC Openshift - P2 VR Chaoss*, the coefficient value of 290.125 suggests that, on average, the TOTAL time for this group is 290.125 units higher than the reference group. This difference is statistically significant ($p < 0.05$), indicating that there is a significant effect of *P1 SC Openshift - P2 VR Chaoss* on the TOTAL time. For *P1 VR Chaoss - P2 SC Openshift*, the coefficient value of 136.875 suggests that, on average, the TOTAL time for this group is 136.875 units higher than the reference group. However, this difference is not statistically significant ($p > 0.05$), indicating that there is no significant effect of *P1 VR Chaoss - P2 SC Openshift* on the TOTAL time. For *P1 VR Openshift - P2 SC Chaoss*, the coefficient value of 207.125 suggests that, on average, the TOTAL time for this group is 207.125 units higher than the reference group. This difference is marginally non-significant ($p = 0.075$), indicating a weak trend towards significance. However, it does not meet the conventional threshold for statistical significance.
The positive value of Cohen's d for "*P1 SC Openshift - P2 VR Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (1.5683) suggests a large effect size, indicating a substantial difference between these groups in terms of TOTAL time. Cohen's d values for "*P1 VR Chaoss - P2 SC Openshift* vs *P1 SC Chaoss - P2 VR Openshift*" (0.6976) and "*P1 VR Openshift - P2 SC Chaoss* vs *P1 SC Chaoss - P2 VR Openshift*" (1.2584) indicate moderate to large effect sizes, suggesting meaningful differences between these groups.
In summary, these results suggest that there is a statistically significant difference in TOTAL time between the group "*P1 SC Openshift - P2 VR Chaoss*" and the reference group "*P1 SC Chaoss - P2 VR Openshift*". The effect size, as measured by Cohen's d, indicates a large difference between these two groups. However, there are no significant differences in TOTAL time between the groups "*P1 VR Chaoss - P2 SC Openshift*" and "*P1 VR Openshift - P2 SC Chaoss*" compared to the reference group. The effect sizes for these comparisons are moderate to large, indicating meaningful differences, but the statistical significance is not achieved for "*P1 VR Chaoss - P2 SC Openshift*".

5.3 Effect of Confounding Variables

To learn about the possible effect of confounding variables, we depict the main two results (time to complete tasks, and correctness of the answers) for each subject against the value of each relevant confounding variable. In Section 3.5 we already identified some variables for

Table 8 Results of the mixed linear method applied to the time to completion

	TOTAL		
Group	Coefficient	p-value	Eff size
P1 - SC Openshift	290.125	0.027	1.57
P2 - VR Chaoss			
P1 - VR Chaoss	136.875	0.324	0.70
P2 - SC Openshift			
P1 - VR Openshift	207.125	0.075	1.26
P2 - SC Chaoss			

P1 - SC Chaoss, P2 - VR Openshift as the reference category. For the total time of both rounds

which we had very low diversity, and therefore we are not considering them in this analysis. For the rest of the confounding variables, Fig. 17 shows how time to complete tasks for each subject is affected by each of those variables, while Fig. 18 shows how the mean correctness of each subject is affected by them.

Complementing Fig. 17, we conducted a Kendall Tau analysis (Kendall 1938) for analyzing the correlation of the variables. Table 9 depicts the results, where we can observe:

- JOB_POSITION (Job position): The Kendall Tau coefficient for JOB_POSITION is close to zero, indicating a very weak or no monotonic relationship between JOB_POSITION and the total time to finish all the tasks. The p-value is greater than the typical significance level of 0.05, suggesting that this correlation is not statistically significant.
- AGE (Age): The Kendall Tau coefficient for AGE is also close to zero, indicating a very weak or no monotonic relationship between AGE and the total time to finish all the tasks. The p-value is greater than 0.05, suggesting that this correlation is not statistically significant.
- EXP_PRG (Experience in Programming): The Kendall Tau coefficient for EXP_PRG is positive and larger in magnitude, indicating a moderate positive monotonic relationship

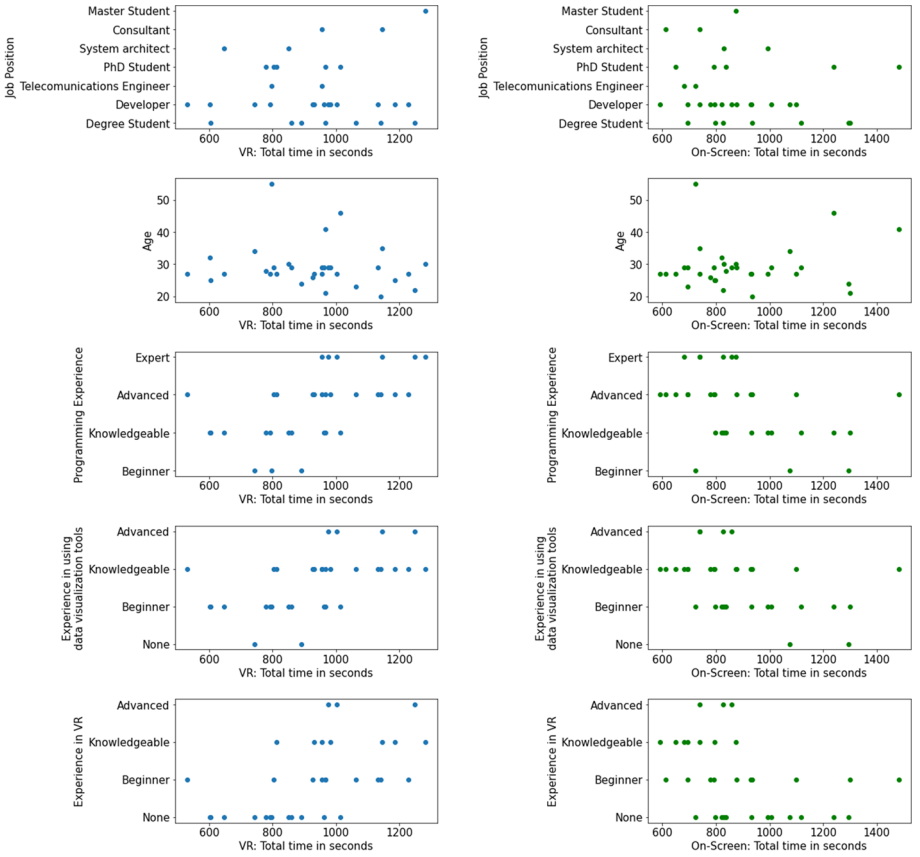

Fig. 17 Confounding variables: effect on time to completion. Each chart represents the effect of a relevant confounding variable (JobPosition, Age, ExpPRG, ExpDataviz, ExpVR) by plotting for each subject its value in the Y axis with respect to the total time to finish all tasks in the X axis

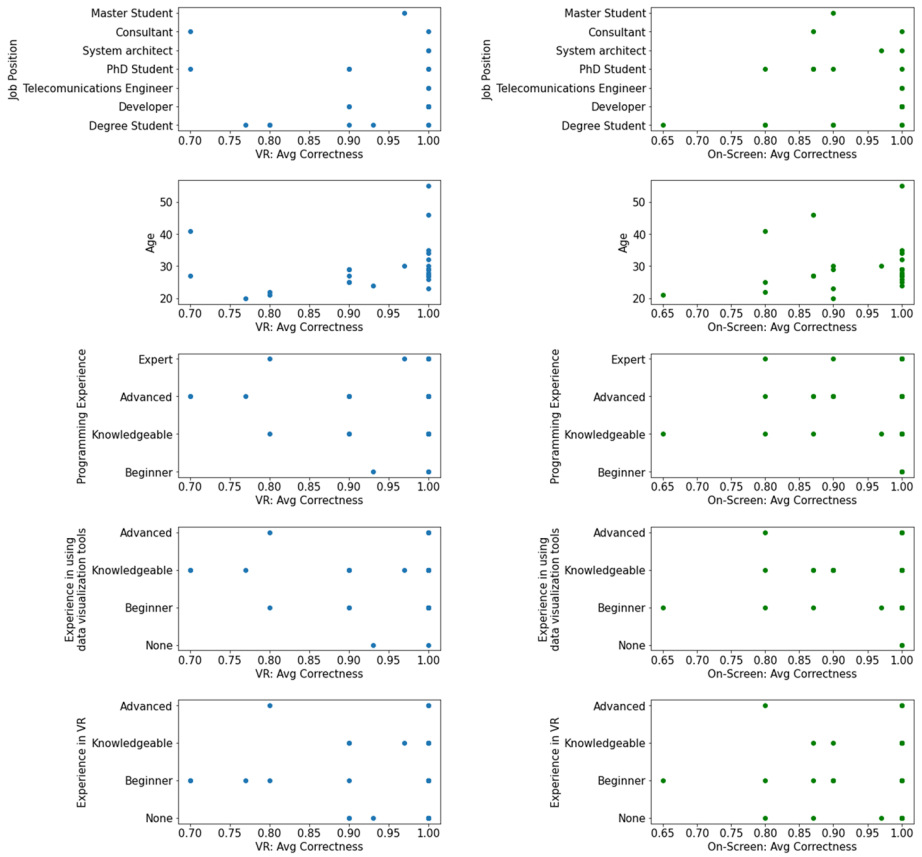

Fig. 18 Confounding variables: effect on the correctness of the answers. Each chart represents the effect of a relevant confounding variable (JobPosition, Age, ExpPRG, ExpDataviz, ExpVR) by plotting for each subject its value in the Y axis with respect to the mean correctness for all answers (scaled in the 0–1 range) in the X axis

between EXP_PRG and the total time to finish all the tasks. The p-value is less than 0.05, indicating that this correlation is statistically significant at the 5% level. This suggests that as the value of EXP_PRG increases, the value of the total time to finish all the tasks tends to increase as well.

- EXP_DATAVIZ (Experience in Data visualization aplications): The Kendall Tau coefficient for EXP_DATAVIZ is close to zero, indicating a very weak or no monotonic relationship between EXP_DATAVIZ and the total time to finish all the tasks. The p-value is greater than 0.05, suggesting that this correlation is not statistically significant.
- EXP_VR (Experience in Virtual Reality): The Kendall Tau coefficient for EXP_VR is close to zero, indicating a very weak or no monotonic relationship between EXP_VR and the total time to finish all the tasks. The p-value is greater than 0.05, suggesting that this correlation is not statistically significant.

In summary, the only statistically significant correlation observed in this analysis is between EXP_PRG and the total time to finish all the tasks, indicating a moderate positive

Table 9 Kendall Tau correlation results of confounding variables and total time to completion

Variable	Kendall Tau	p-value
JOB_POSITION	-0.050410	0.708797
AGE	0.027671	0.830563
EXP_PRG	0.336019	0.015252
EXP_DATAVIZ	-0.086642	0.537350
EXP_VR	0.054354	0.693977

relationship. The other variables (JOB_POSITION, AGE, EXP_DATAVIZ, and EXP_VR) do not exhibit significant correlations with the total time to finish all the tasks.

By observing Fig. 18, we also complete the chart with the Kendall Tau analysis, depicted in Table 10 for correctness:

- JOB_POSITION (Job position): The Kendall Tau coefficient for JOB_POSITION is positive, indicating a weak positive monotonic relationship with the average of correctness. However, the p-value is greater than 0.05, suggesting that this correlation is not statistically significant. Therefore, we cannot conclude that there is a significant association between JOB_POSITION and the average of correctness.
- AGE (Age): The Kendall Tau coefficient for AGE is positive, indicating a weak positive monotonic relationship with the average of correctness. However, the p-value is greater than 0.05, indicating that this correlation is not statistically significant. Therefore, we cannot conclude that there is a significant association between AGE and the average of correctness.
- EXP_PRG (Experience in Programming): The Kendall Tau coefficient for EXP_PRG is close to zero, indicating a very weak or no monotonic relationship with the average of correctness. Additionally, the p-value is greater than 0.05, indicating that this correlation is not statistically significant. Therefore, there is no evidence of a significant association between EXP_PRG and the average of correctness.
- EXP_DATAVIZ (Experience in Data visualization aplications): The Kendall Tau coefficient for EXP_DATAVIZ is negative, indicating a weak negative monotonic relationship with the average of correctness. However, the p-value is greater than 0.05, suggesting that this correlation is not statistically significant. Therefore, we cannot conclude that there is a significant association between EXP_DATAVIZ and the average of correctness.
- EXP_VR (Experience in Virtual Reality): The Kendall Tau coefficient for EXP_VR is negative, indicating a weak negative monotonic relationship with the average of correctness. The p-value is less than 0.05, indicating that this correlation is statistically significant at the 5% level. Therefore, there is evidence of a significant association between EXP_VR

Table 10 Kendall Tau correlation results of confounding variables and correctness average

Variable	Kendall Tau	p-value
JOB_POSITION	0.104167	0.492507
AGE	0.165881	0.253402
EXP_PRG	0.012381	0.936668
EXP_DATAVIZ	-0.154337	0.328973
EXP_VR	-0.322301	0.038110

and the average of correctness, suggesting that as the value of EXP_VR decreases, the value of the average of correctness tends to increase.

In summary, the only statistically significant correlation observed in this analysis is between EXP_VR and the average of correctness, indicating a weak negative relationship. The other variables (JOB_POSITION, AGE, and EXP_DATAVIZ) do not exhibit significant correlations with the average of correctness.

5.4 Feedback

5.4.1 Easier and Faster?

Once the participants had finished all tasks, they were asked to answer a small feedback survey. First, in question $\mathbf{S_1}$ we ask in what environment it was easier for them to answer the questions, and a rationale for the answer. Figure 19 depicts how the answers are distributed.

75% of participants answered that it was easier for them to solve the tasks on-screen. Of these participants, 50% claimed in the justification that it is due to the habit of using the screen on a daily basis (37.5% of the total number of participants). The "habit" factor influences the performance of the experiment. For those who answered VR (6 participants, 16.75%), they claimed better coherence between the data and a better arrangement of the graphs (shelves) in the same environment, instead of having to move between two. Those who did not respond that any specific environment was easier commented that they were not clear about this, and that they did not find much difference. In addition, some participants pointed out that it was easier for them in the second part (regardless of the setting) because they had prior knowledge of the questions.

After answering the first feedback question, $\mathbf{S_2}$ was presented to participants, asking about the feeling of time spent on tasks. In particular, the question was in which environment they had the feeling of having spent less time solving the tasks, along with a justification. Figure 19 depicts how the answers are distributed. In this case, we have almost an equal distribution between those who responded on-screen and in VR. 14 participants (43.75%) claimed that they did it faster due to prior knowledge of the questions or graphs, which is reasonable. 4 participants (12.5%) again claimed that they did it faster due to the on-screen habit. In addition, 2 participants argued that VR was faster due to the layout of the elements; one of

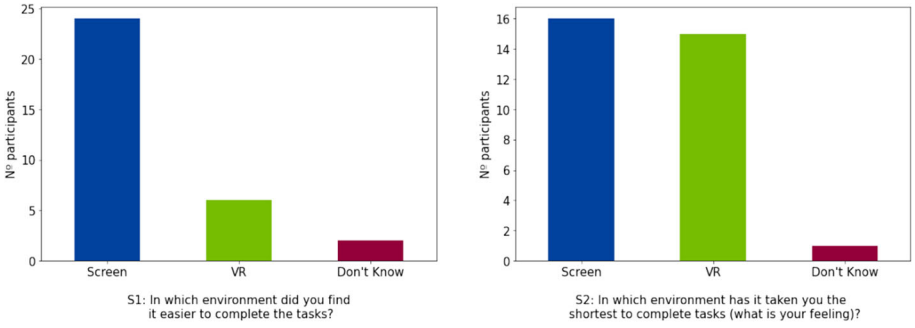

Fig. 19 Answers by participants to questions $\mathbf{S_1}$ (easier environment for answering questions) and $\mathbf{S_2}$ (faster environment for answering questions)

them commented that on-screen the information was overloaded, and another that on VR it was simpler.

5.4.2 Advantages and Disadvantages

S_3 was asked in order to get positive and negative feedback about the *Kibana* (on-screen) and *BabiaXR* (VR) environments. Starting with *Kibana*, this is a summary of the most relevant points:

- *Advantages:* The habit of the use of on-screen application, and the "everyday" interaction with them stand out again. Participants also mentioned that information is in front of you, and in a more compressed manner, that in general it is easier to use and the letters are seen more clearly.
- *Disadvantages:* The most highlighted disadvantage is that information is spread in two dashboards, which makes it more difficult to correlate data from visualizations in different dashboards. In general, participants also found that charts are more difficult to find, because they are all together, and are quite similar to each other. Some participants mentioned that dashboards are too crowded with charts, and on-screen space is very limited.

And regarding the *BabiaXR* environment:

- *Advantages:* One of the greatest advantages highlighted by participants is the use of space, having all visualizations in the same place. Also, the use of colors for the difference between pull requests and issues improves interaction and data correlation. The "museum with shelves" metaphor was also highlighted as a positive aspect, specifically the "shelves" that allow organizing the different graphs in a very intuitive way. It is noteworthy to point out how participants noted that they felt more focused in the VR environment, because there are no distractions around and everything they saw had to do with the experiment.
- *Disadvantages:* The use of the VR headset and the discomfort with it (eye strain, lack of experience in their use, etc.) with it is a point that several participants noted as negative. Another negative aspect is the size of the texts and the legends, which some participants reported to be difficult to read. Finally, the performance and the drop in frames per second in some circumstances is another aspect that they detailed as negative.

5.4.3 Control Questions

To finish the feedback survey, participants were asked if they found the experiment difficult (**C_1**). Figure 20 depicts the distribution of answers by participants and their thoughts.

Only one participant agreed that the experiment was difficult. We can conclude that the design of the experiment is of low difficulty.

To obtain extended comments, we asked a final question (**C_2**) about any other suggestion or comment. We received useful comments like the general comment of several participants in relation to the legends in VR, since sometimes it was difficult for them to look at the data, as well as a better format for these. Others commented on the use of other graphs beyond the bars, since that could improve the general visualization of the data. We will further discuss this in Section 6. The multi-user feature (to be able to have more than one user in the scene and interact between them and with the visualizations), something included in *BabiaXR*, is

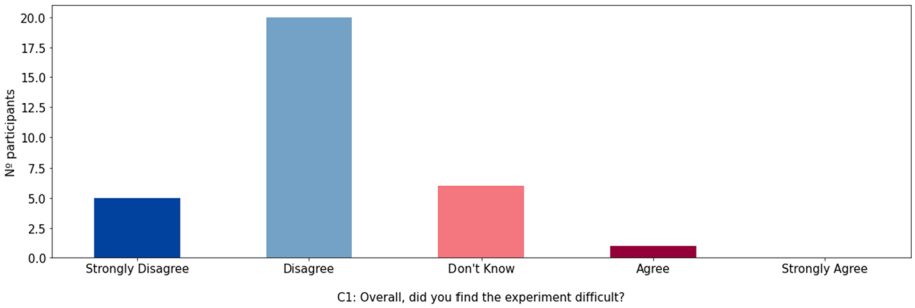

Fig. 20 Participants' answers to $\mathbf{C_1}$ (was the experiment difficult?)

another of the comments that two participants have proposed; it helps us to know that it could help in future experiments. Other issues like performance, Frames Per Second (FPS) drop and graphics size were indicated by 5 participants.

We also received comments and suggestions regarding the *Kibana* environment, such as dividing the graphs better, using a better color range, or even reducing the number of graphs. The merge between issues and Pull Request visualizations is another point that the participants pointed out.

6 Discussion

After analyzing the results we can conclude that *BabiaXR* (VR) is at least as useful in terms of correctness and completion time as *Kibana* (on-screen). However, there are different points and specific cases: to discuss from results, feedback, and other insights.

Habit Factor Most of the feedback in favor of on-screen is related to the habit that the participants have on the computer. Today, the computer is a widely used tool used in most modern jobs, so the use of a screen, mouse, and keyboard is much more widespread than VR headsets and controllers. Even with more habit on screen, the results are not far off. We hypothesize that as VR environments become more used –or developers more exposed to VR technology to perform these tasks–, results may change in favor of VR. Replicating this experiment in the future may be of interest to ascertain if this is true.

Two Rounds of the Same Tasks It is evident that in both correctness and completion time, improvements were observed in the second round, regardless of the environment. This outcome is expected as participants gain familiarity with the visualizations and tasks over time. Even when the environment and project (albeit with varying complexity) change, participants have acquired knowledge of where to look and the objectives of each task. The tasks were carefully designed to reflect common questions addressed by Bitergia in *real* software development environments. It is worth noting that more intensive prior training, specifically explaining the data displayed in each visualization, could potentially lead to a normalization of completion times between the two rounds. This observation indicates that both environments are effective for learning to perform this type of task.

Isolated VR Environment Immersion in VR has the advantage that you control everything that surrounds you in an almost unlimited space. This means that you can place things of

interest in all aspects and control those aspects that can distract the participant when making an experiment. This is something that several participants gave in favor of VR. Even though they also highlighted some negative parts, they pointed out that in VR you have a more controlled focus, you do not feel external distraction because practically everything you look at is visualization in all directions. This is a very important point since in VR we have combined both the views of Issues and Pull Request, but even so, the environment was not 100% exploited. So, this point is something to investigate and without doubt to get more out of it, to be able to show more information in future experiments.

Tasks Favoring VR Specifically, in $\mathbf{T_5}$ we have found a substantial difference in favor of VR. This task is specifically designed to correlate issues and pull requests data, first looking for submitters in the "Average days open for Issues" visualization, and then looking for those same submitters in the corresponding pull requests visualization. As in many software projects, there are user profiles that can create issues and pull requests, but they can also create issues but no pull requests, because, for example, they do not have a developer profile (or vice-versa). Only in $\mathbf{T_5}$, when looking for submitters in the pull requests visualization, they were not found because they had not created pull requests. This can be clearly identified when you have both visualizations available. In the on-screen environment, you have to move between dashboards in different web pages to correlate the information. In VR, instead, you have one visualization directly above the other, being able to correlate the information in a faster way, as can be seen from the results. This gives us an idea of the kind of tasks that are more easily solved in the VR environment. Defining scenarios designed to ease the completion of those kinds of tasks in VR may lead to the creation of more efficient and powerful VR dashboards than those possible on-screen.

Tasks Favoring a Project At first glance, it may appear that certain tasks are designed to favor a particular project based on its size. However, it is important to note that Bitergia, as the designer of the dashboards and a key collaborator in task design, collaborates with customers whose interests span projects of varying magnitudes. Consequently, the tasks and questions are carefully crafted to ensure that project size does not influence their outcomes. In our study, the primary impact of data size is on the performance of the application itself. With current technological limitations, VR dashboards may exhibit slower rendering speeds than on-screen dashboards for larger data sizes. Nonetheless, in our experiment, we observed no significant performance differences between the two projects as the visualizations and questions were carefully selected to ensure consistent and comparable performance. Regarding the discrepancy in $\mathbf{T_1}$ outcomes between the projects, we attribute it to the initial reaction time of participants who were encountering the data for the first time. This variation could be linked to the different data sizes. However, with a larger participant sample size, we anticipate this difference to diminish, as demonstrated by the outcomes of the other tasks.

VR vs On-screen by Participants After the qualitative analysis of the feedback survey, we have realized that participants value being used to the environment above other factors (colors, layout, etc.). In general, everyone felt more comfortable on screen and reported that they perceive tasks to be easier in this environment. It is interesting how for VR several subjects detailed some advantages, such as the use of available space, something that they even highlighted as a disadvantage on-screen. Another very interesting aspect of VR over on-screen is the fact that participants felt more focused in VR because they had no distractions around them, and the information was more spread out in space, something that the metaphor of the museum reflects very well. This is also the case with the placement of the visualizations

on shelves for a better feeling of comfort and familiarity. To this day there is no widespread use of it in VR, and even less for this type of applications. Applications like *BabiaXR* need more development to improve visualization, performance, and interaction problems. So, to some extent, this experiment could be a baseline: once VR is more extended, user interaction and metaphors are improved in VR, and subjects are more used to the specific interactions and metaphors used in the VR dashboard, maybe a similar experiment will show better results for VR. In any case, it is also important to notice that, with respect to correctness, there were no perceived differences, and both environments performed well. Given that Kibana is state of the art with respect to showing visualizations to interact with data, we think this is very good news for data visualization in VR.

VR Metaphor The VR environment allows the use of different metaphors for the layout of the visualizations. In this case, the metaphor of the museum has been chosen, placing the visualizations in an environment similar to that of a museum, with a limited room and different elements at various heights with their respective titles/posters. We still do not have enough feedback to know if this metaphor is the right one for these types of visualizations and data, so using a different metaphor can change the results drastically. What this experiment has shown is that the museum metaphor for these data and tasks is similar in terms of effectiveness and efficiency to the *Kibana* on-screen environment. The use of natural movements, such as bending over, moving arms, and head gestures is something to explore for the improvement of the metaphor to be used.

On-Screen Environment Interactions The interaction with *Kibana* is an interaction based on keyboard and mouse, a very mature interaction that has been widely used by all participants that have carried out the experiment. It is a natural interaction that has yet been limited for a fair comparison with *BabiaXR*. The best ways to interact with visualizations in VR are still being researched, and as experiments like this are carried out, better ways to interact, more natural and similar to the on-screen environment, will appear. In addition, we have limited the use of on-screen to one monitor, so the use of more than one monitor could influence the results, due to a greater correlation of the data.

***BabiaXR* as a Library** The visualization library, *BabiaXR*, is currently in its developmental phase, and we are actively seeking feedback to drive its evolution and maturity. Conducting experiments plays a crucial role in obtaining valuable insights for improving the library's visualizations and data interaction capabilities. As we gather feedback, we will be able to refine the approach, both in terms of visualization quality and data interaction mechanisms. It is important to note that the current comparison between *BabiaXR* and the 2D visualizations of *Kibana* is limited in fully harnessing the potential of VR. *BabiaXR* offers additional visualizations, such as networks, cylinders, and even a city visualization (Wettel and Lanza 2007), which have not been utilized in this study. Integrating these advanced visualizations into the experiment could potentially bias the results in favor of *BabiaXR*. Therefore, further experimentation is warranted to explore and evaluate the extended capabilities of *BabiaXR*.

Experiment Duration In this experiment, the presence of a supervisor in the same physical location as the participant during the experiment posed limitations on scalability. Each participant's session, including the experiment and subsequent video review by the supervisor, lasted approximately one hour. This significantly restricts the number of participants that can be included. To address this challenge, we are actively exploring options to automate the experiment process for future studies. While conducting the experiment remotely and

in a sequential manner for *BabiaXR* presents difficulties due to the limited availability of VR headsets, *BabiaXR* offers a multi-user feature that allows the supervisor to join the same virtual environment as the participant, facilitating remote supervision without the need for physical proximity. Furthermore, for data collection purposes, *BabiaXR* provides the capability to record and plot user positions, responses, and movements within the library itself. On the other hand, for *Kibana*, which offers greater ease of replication for remote experiments, a simple screen-sharing call can suffice for remote monitoring without the need for in-person presence. These considerations highlight the potential for exploring alternative methods to enhance the scalability and replicability of these experiments in the future.

VR Expertise One of the most unexpected results is the apparent influence of experience with VR in time to completion of the tasks. As we explained, apparently the more VR expertise, the shorter times to complete on-screen, compared with VR. This result seems counter-intuitive, because it seems reasonable to expect that the more expertise and familiarity with VR, the easier it is for the subject to interact with the VR scene, and therefore, the shorter time to completion of the task in VR. Nonetheless, the number of subjects with some experience in VR in our experiment was relatively low, which means that this could just be an artifact, and not a real result. On the other hand, maybe the question about VR expertise was too broad: given the large variety of devices and environments providing VR experiences, it is difficult to know how subjects interpreted the question. Finally, this diversity of devices and interaction methods that currently happens in the VR world maybe makes it difficult to take advantage of past experiences with VR for our experiment: maybe the devices used were very different, or maybe the interaction mechanisms and gestures were very different. In any case, given the unexpected and potential importance of this result, it is worth exploring it in more detail in future research.

7 Threats to Validity

7.1 Internal Validity

Internal validity is related to uncontrolled factors that can influence the causal relationship between independent and dependent variables (Wohlin et al. 2012). In our case, it pertains to:

- **Subjects.** We ensured that all participants had experience in different relevant topics about programming using a questionnaire, focusing to recruit people with job positions related to the software development (including academia and industry), reducing the threat that they were not competent enough. Moreover, we asked for their experience in the relevant topics to mitigate the threat that participants' experience was not distributed fairly. However, their training for the environment of their experiment (i.e.,*Kibana* or *BabiaXR*) was not uniform, with persons participating in the VR experiment being much less experienced in VR environments than on-screen participants in on-screen environments.
- **Tasks.** The choice of tasks may have been biased in favor of *Kibana* or *BabiaXR*. We mitigated this threat by using *Kibana* dashboards validated by Bitergia, replicating the same visualizations in the *BabiaXR* dashboard. Moreover, in the two environments, we

have exactly the same tasks, so the level of difficulty was as similar as possible. We also included tasks that put both modes at a disadvantage: Tasks focused on precision could be easier on-screen, while tasks focused on locality could be easier in VR. Not controlled aspects (e.g., the external environment of the *BabiaXR* scene) could have an influence on the results as well.

- **Training.** In both environments (i.e., *Kibana* and *BabiaXR*) the text to be followed for performing the tasks explains how the tool is used and how the interaction with the elements works. No participant had relevant previous experience with *Kibana* or *BabiaXR*. Moreover, an optional tutorial about the first steps with the VR device was proposed to them (a generic starter tutorial included in the Oculus Quest 2). This could balance a bit the situation for VR participants, but given the extensive on-screen experience, this would hardly make them more efficient. It remains to be investigated whether a practical tutorial on how to interact with a VR headset could reduce the experience gap between VR and on-screen, improving the correctness of VR activities.
- **Fatigue and Learning Factors.** The experiment design introduced a potential threat related to the influence of fatigue and learning factors on participants' performance. Due to the sequential nature of the tasks, participants might experience fatigue as they progress through the experiment, which could impact their cognitive abilities, attention, and task performance. Moreover, the learning effect could influence participants' performance over time, as they become more familiar with the tasks and the specific environments. The order of the tasks and the repetition of the tasks in different environments may interact with the learning and fatigue factors, potentially affecting the validity of the results.
- **Repeated Measures Design.** The use of a repeated measures design, where participants are measured under different conditions, introduces potential threats to validity. One potential threat is order effects, where the order in which conditions are presented may impact participants' performance or responses. To mitigate this threat, counterbalancing was employed, ensuring that participants experienced the conditions in different sequences. Another potential threat is carryover effects, where the experience of one condition may influence participants' performance in subsequent conditions. To address this, appropriate rest periods were provided between conditions to minimize carryover effects. Also, we analyzed the learning factor, mitigating the threat.
- **Influence of Virtual Scene Design**. The design of the virtual scene, including its structure and photorealism, may introduce confounding variables that affect participants' performance and perception. Factors such as layout, color schemes, and object placement could impact participants' cognitive processes, engagement, and sense of presence. The level of realism and visual design elements within the virtual scene may influence participants' interpretation and interaction with the data. To mitigate this threat, we made efforts to create a representative virtual scene, but variations in responses due to individual differences and preferences may still exist. Future experiments will address the influence of virtual scene design as a potential confounding variable by carefully considering design elements and gathering participant feedback to better understand and control for these factors.

7.2 External Validity

External validity relates to the generalizability of the results of the experiment (Wohlin et al. 2012). In our case, it pertains to:

- **Sample Size.** The number of participants in the experiment is somewhat limited, which may affect the generalizability of the findings. Increasing the sample size would enhance the statistical power and reliability of the results. However, it should be noted that the current sample size is in the range commonly observed in similar experiments.
- **Subjects.** We employed a combination of convenience sampling and targeted recruitment strategies to ensure subject representativeness. Convenience sampling allowed us to efficiently gather accessible and willing participants. Additionally, we actively recruited individuals meeting specific criteria related to job position and years of experience in programming topics. This involved reaching out to professional organizations, academic institutions, online communities, and industry networks. Our aim was to achieve a balanced mix of academics and professionals, ensuring diverse perspectives. By implementing these strategies, we sought to mitigate biases and enhance the representativeness of our subject sample.
- **Target System.** Another threat is represented by the choice of the projects: *CHAOSS* and *OpenShift*. Participants did not know them in advance, except for one who knew the *OpenShift* project as a *"Beginner"*. We cannot assess how appropriate or representative *CHAOSS* and *OpenShift* are for the software development processes tasks we designed, but the consistent variations in solutions for the same task in both VR and on-screen environments signal that results could be extensible to other systems. Said this, our experimental approach has been validated with experience and expertise from Bitergia, so that we can be sure that the tasks are commonly performed in real, industry settings.

7.3 Contruct Validity

Construct validity refers to the extent to which the measurements or manipulations used in a study accurately represent the constructs they are intended to measure or manipulate (Wohlin et al. 2012). In our case, it pertains to:

- **Time Measurement.** To ensure accurate time measurement and mitigate potential inaccuracies in task completion times, we implemented specific strategies in our experiments. Firstly, a supervisor was present during each experiment run to record the time taken by participants to complete tasks. This provided a reliable and independent source of time measurement. Additionally, participants were instructed to verbally communicate their task completion to the supervisor, serving as a double-check for the recorded completion time. Moreover, the use of the *Kibana* and *BabiaXR* environments facilitated real-time task completion without the need for manual recording on paper, particularly advantageous in the VR environment where paper-based methods can be cumbersome. These measures helped minimize any potential errors or delays in time measurement, enhancing the internal validity of our study.
- **Experimenter Effect.** One of the experimenters is one of the authors of *BabiaXR*, which may have influenced the experiment. For example, task solutions may not have been graded correctly. To mitigate this threat, this author did not interfere in the experiment, and if he had to interfere, the results were canceled. The experimenter built a model of the responses based on previous experiments in the literature (e.g., (Wettel et al. 2011; Romano et al. 2019)). Even if we tried to mitigate this threat extensively, we cannot exclude all possible influences on the results of the experiment.

8 Related Work

Visualization Visualization has a rich history, dating back to ancient times. It has been used for exploration and communication of knowledge since the first known map in 6200 BC and the first chart of star constellations in 134 BC (Friendly 2008). Throughout history, visualizations have played a crucial role in understanding various phenomena. For instance, visualizations have been employed to comprehend the spread of diseases (Snow 1856) and analyze patterns of economic growth (Playfair 1822). The growing availability of data and advanced analytical tools has further amplified the significance of data visualization (Friendly 2008; Liu et al. 2014).

Data within Visualizations Two-dimensional (2D) visualizations are widely utilized to represent data relationships and patterns. Scatter plots, for example, are commonly employed to display the correlation between two variables, where data points are plotted on a Cartesian plane (Cleveland 1994). Line charts, on the other hand, are effective in depicting trends and temporal variations in data (Few 2009). These 2D visualizations, among others, provide valuable insights into the underlying structures and patterns within datasets.

Within the realm of information visualization, a specific research stream called InfoVis focuses on developing highly effective visualizations for data exploration and communication (Munzner 2014). By employing a range of techniques and design principles, InfoVis aims to enhance the understanding and interpretation of complex data. Through interactive and visually appealing visualizations, InfoVis empowers users to explore data, discover meaningful insights, and effectively communicate their findings.

Virtual Reality Use Cases Virtual Reality has been shown to facilitate discovery in domains in which space plays an important role. For example in the field of brain tumors (Zhang et al. 2001), perception of shapes and forms (Demiralp et al. 2006), paleontology (Laha et al. 2014), caves (Ragan et al. 2013), and magnetic resonance imaging (Chen et al. 2012). Data visualization in virtual reality allows the use of multidimensionality for abstract analysis, and even more so for large data sets.

VR Data Visualizations Regarding the VR and immersive data visualizations, there is extensive prior work on the use of VR (Milgram and Kishino 1994; Skarbez et al. 2021) for visualizing scientific data. Seismic, protein-docking, astronomy, and health care data are examples of it (Kaiser et al. 2005; Anderson and Weng 1999; Djorgovski et al. 2013; Ibrahim and Money 2019). Bryson (1996) highlighted the possibilities offered by VR for interaction with complex phenomena and their data visualizations. The standard way to visualize is 2D space, data visualizations in 3D (and VR) have been included in the literature with caution, Munzner (2014) warned of unjustified 3D, and Few (2004) calls to "avoid 3D displays of quantitative data". At the same time, researchers are arguing for the benefits of 3D and VR for data visualization. Some examples are following: Batch et al. (2019) focused on presence, Jacob et al. (2008) focused on embodiment, Rosenbaum et al. (2011) focused on involvement, and García-Hernández et al. (2016) in the aerospace engineering field.

More VR data visualizations can be found in the research literature. Donalek et al. (2014) presented immersive and collaborative data visualization using VR platforms. Before that, Bayyari and Tudoreanu (2006) presented that situational awareness in the visualization of data benefits from immersive, virtual reality display technology because such displays appear to support a better understanding of the visual information. More recently, Millais et al. (2018)

presented a comparison between 2D and VR visualizations, suggesting that users feel more satisfied and successful when using VR data exploration tools, thus demonstrating the potential of VR as an engaging medium for visual data analytics. Navigation in VR is another field of research, Drogemuller et al. (2018) evaluated three-dimensional VR navigation techniques for data visualizations and test their effectiveness with large graph visualizations. There are other fields of study for data visualization, and Augmented Reality (AR) is one of them. Olshannikova et al. (2015) presented an overview of research challenges and achievements in the field of Big Data visualization, and Natephra and Motamedi (2019) explored data visualization using AR, IoT sensors as data entry points.

Software Engineering and Visualizations The field of software visualization in virtual reality (VR) remains relatively underexplored in research. Some early explorations in this domain include the work by Young and Munro (1998), who investigated the use of VR for software visualization. Another notable approach is *Imsovision* (Maletic et al. 2001), which focused on C++ and introduced metrics that continue to be widely used in the literature. With advancements in technology, VR-based software visualizations have gained significant attention. The concept of using a city metaphor in software visualization originates from Wettel and Lanza (2007), and this idea was further developed in the VR context with projects like *CityVR* (Merino et al. 2017). Kobayashi et al. (2013) proposed a city metaphor-based approach for representing software architectures, employing buildings to depict packages and lines to illustrate dependencies between them. Building upon this work, Yano and Matsuo (2017) expanded the concept by introducing additional indexes to capture the characteristics of software based on its dependencies. Another noteworthy approach is *IslandViz* (Misiak et al. 2018), which visualizes software architectures as islands and showcases package dependencies as relationships between these islands, using arrows to denote the hierarchy of dependencies.

VR vs On-screen Comparing VR and 2D implementations is a state-of-the-art field. Raja et al. (2004) used the CAVE (Cruz-Neira et al. 1993) for performing a pilot study with four conditions and four subjects and found that the most immersed one had the best results in terms of users' ability for performing tasks. Millais et al. (2018) presented a 3D scatter plot and a 3D parallel coordinate plot to 16 subjects, half of them using a 2D screen. We also conducted an experiment (Moreno-Lumbreras et al. 2021) for comparing the CodeCity (Wettel and Lanza 2007) tool developed with *BabiaXR* in 2D screens and VR, finding that VR is more effective (in terms of completion time) than 2D screens for these visualizations. Merino et al. (2017) conducted a controlled experiment using 3D visualizations of the city metaphor using a computer screen, an immersive 3D environment, and a 3D printed model. The authors found that on-screen participants perceived the least difficulty to identify outliers; in terms of completion time, the results show that VR participants were faster in resolving the program comprehension tasks. Our study differs with this study, since in terms of completion time VR participants were faster than on-screen participants. Rüdel et al. (2018) conducted a controlled experiment with 20 participants of the city metaphor, but using a different algorithm for the layout. They compared as well VR to on-screen and found that on-screen participants were faster in resolving the proposed tasks. This result does not follow the line of our study, since both the type of metrics, as well as the data and the visualization are different and therefore cannot be compared. Romano et al. (2019) conducted a controlled experiment where they asked participants to perform program comprehension tasks with the support of the *Eclipse* Integrated Development Environment (i.e., IDE) with a plugin for gathering code metrics and identifying bad smells and a visualization tool of the city metaphor displayed on a standard

computer screen and in immersive virtual reality, showing that VR participants were faster than on-screen participants. Our results again are not in the line with those shown in Romano et al. (2019), since our results do not show a statistically significant difference between the on-screen and the VR environment.

9 Conclusions and Future Work

In this paper, we presented how we can replicate a real scenario that is used by a real company for analyzing software development metrics in VR, using data visualizations in a dashboard. First, we have presented on-screen dashboards, created by the Bitergia company with *Kibana*, where software development process metrics are shown. Specifically, we have focused on the timing of Issues and Pull Requests, where bottlenecks in software development are clearly seen as well as community activity. These data are collected with *GrimoireLab*. Then, we present our approach, *BabiaXR*, to represent the same dashboards but in a VR environment, using the museum metaphor. Each of the *Kibana* graphs has been developed specifically for *BabiaXR*, and can have two similar environments for a fair comparison.

We have conducted an experiment that included 32 participants from academia and industry in which they faced 5 tasks, designed with the help of Bitergia, where details of the software development processes were asked. The experiment has two systems, of different magnitude. Participants faced a randomly chosen environment (*BabiaXR* or *Kibana*) with data from a randomly chosen project and solved the tasks. Once solved, they faced the same tasks in the other environment and system.

To verify the results, both accuracy (correctness) and efficiency (completion time) were measured. In addition to these quantitative data, participants previously answered a demographic survey and subsequently a feedback survey to obtain qualitative results. The isolated tasks in which the VR environment show better results have been analyzed in order to identify what sections of these make the results better. It is of great importance to isolate what benefits the VR environment brings to solve certain tasks, study the characteristics and be able to create a dashboard where everything is favorable and to be more efficient and effective in VR. A future experiment would be of interest to verify the impact of the changes.

The objective of the experiment is to compare which environment (VR or on-screen) allows for a more accurate and faster solution to tasks. After analyzing the results, we have seen that the correctness results are very good (above 80% on average in all tasks) regardless of the sequence. For time data (completion time), the results show that as participants solve tasks, responding times improve, especially in the second round. So we can conclude that the *BabiaXR* environment is at least as good in terms of correctness and completion time as in *Kibana*, the on-screen environment.

The qualitative results show that participants feel more comfortable and find it easier to solve tasks in the on-screen environment, mostly due to the habit of using it. Instead, they have a similar perception of speed in solving tasks in both environments. This suggests that, as the use of VR becomes more common, these results may improve and even exceed those obtained for the on-screen environment. In general, the experiment has been perceived as easy for all participants. In addition, the qualitative feedback gives us a good basis for future work to improve the 3D approach by improving the interface, the environment in which the

graphs are displayed, performance, and other visual details to make the interaction more comfortable for the user.

Replication package

The data and the analysis obtained for our experiment, and the materials needed to reproduce the experiment are available in the *Replication Package*.[10]

Acknowledgements We acknowledge the financial support of the Community of Madrid for the project IND2018/TIC-9669, and the Spanish Government for the project RTI-2018-101963-B-100, and of the Madrid Regional Government (e-Madrid-CM - P2018/TCS-4307), co-financed by EU Structural Funds (FSE and FEDER). We also thank all the participants of our experiments.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflicts of Interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

Anderson A, Weng Z (1999) VRDD: applying virtual reality visualization to protein docking and design. J Mol Graph Model 17(3–4):180–186

Antoniol G, Ayari K, Di Penta M, Khomh F, Guéhéneuc YG (2008) Is It a Bug or an Enhancement? A Text-Based Approach to Classify Change Requests. In: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds. CASCON '08. New York, USA: Association for Computing Machinery. Available from: https://doi.org/10.1145/1463788.1463819

Bacchelli A, Bird C (2013) Expectations, outcomes, and challenges of modern code review. In: 2013 35th ICSE. IEEE pp. 712–721

Batch A, Cunningham A, Cordeil M, Elmqvist N, Dwyer T, Thomas BH et al (2019) There is no spoon: evaluating performance, space use, and presence with expert domain users in immersive analytics. IEEE Trans Vis Comput Graph 26(1):536–546

Bayyari A, Tudoreanu ME (2006) The Impact of Immersive Virtual Reality Displays on the Understanding of Data Visualization. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology. VRST '06. New York, USA: Association for Computing Machinery, pp 368–371. Available from: https://doi.org/10.1145/1180495.1180570

Bettenburg N, Just S, Schröter A, Weiss C, Premraj R, Zimmermann T (2008) What Makes a Good Bug Report? In: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering. SIGSOFT '08/FSE-16. New York, USA: Association for Computing Machinery, pp 308–318. Available from: https://doi.org/10.1145/1453101.1453146

Bissyandé TF, Lo D, Jiang L, Réveillère L, Klein J, Traon YL (2013) Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub. In: 2013 IEEE 24th International symposium on software reliability engineering (ISSRE), pp 188–197

[10] Replication Package of the experiment: https://doi.org/10.5281/zenodo.8011220

Bowman D, McMahan R (2007) Virtual reality: how much immersion is enough? Computer 08(40):36–43. https://doi.org/10.1109/MC.2007.257
Bryson S (1996) Virtual reality in scientific visualization. Communications of the ACM 39(5):62–71
Chen JJ, Cai H, Auchus AP, Laidlaw DH (2012) Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization. IEEE Trans Vis Comput Graph 18:2130–2139
Cleveland WS (1994) The Elements of Graphing Data. Hobart Press
Cohen J (1988) Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates
Cruz-Neira C, Sandin DJ, DeFanti TA (1993) Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp 135–142
Demiralp C, Jackson C, Karelitz D, Zhang S, Laidlaw D (2006) CAVE and fishtank virtual-reality displays: a qualitative and quantitative comparison. IEEE Trans Vis Comput Graph 12:323–30. https://doi.org/10.1109/TVCG.2006.42
Djorgovski S, Hut P, Knop R, Longo G, McMillan S, Vesperini E et al (2013) The MICA experiment: astrophysics in virtual worlds. arXiv:1301.6808
Donalek C, Djorgovski SG, Cioc A, Wang A, Zhang J, Lawler E et al (2014) Immersive and collaborative data visualization using virtual reality platforms. In: 2014 IEEE International conference on big data (big data), pp 609–614
Drogemuller A, Cunningham A, Walsh J, Cordeil M, Ross W, Thomas B (2018) Evaluating navigation techniques for 3D graph visualizations in virtual reality. In: 2018 International symposium on big data visual and immersive analytics (BDVA), pp 1–10
Dueñas S, Cosentino V, Gonzalez-Barahona JM, del Castillo San Felix A, Izquierdo-Cortazar D, Cañas-Díaz L et al (2021) GrimoireLab: a toolset for software development analytics. Accepted, publication pending, PeerJ Computer Science
Dueñas S, Cosentino V, Robles G, Gonzalez-Barahona JM (2018) Perceval: software project data at your will. In: Proceedings of the 40th international conference on software engineering: companion proceeedings, pp 1–4
Elliott A, Peiris B, Parnin C (2015) Virtual reality in software engineering: affordances, applications, and challenges. In: 2015 IEEE/ACM 37th IEEE International conference on software engineering vol 2. IEEE, pp 547–550
Few S (2004) Show me the numbers. Analytics Pres
Few S (2009) Now You See It: Simple Visualization Techniques for Quantitative Analysis, 1st edn. Analytics Press, Oakland, CA, USA
Friendly M (2008) In: A Brief History of Data Visualization. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 15–56. Available from: https://doi.org/10.1007/978-3-540-33037-0_2
Friendly M (2008) Milestones in the history of thematic cartography, statistical graphics, and data visualization
García-Hernández RJ, Anthes C, Wiedemann M, Kranzlmüller D (2016) Perspectives for using virtual reality to extend visual data mining in information visualization. In: 2016 IEEE Aerospace Conference, pp 1–11
Heer J, Bostock M, Ogievetsky V (2010) A tour through the visualization zoo. Commun ACM 53(6):59–67. https://doi.org/10.1145/1743546.1743567
Hooimeijer P, Weimer W (2007) Modeling Bug Report Quality. In: Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering. ASE '07. New York, USA: Association for Computing Machinery, pp 34–43. Available from: https://doi.org/10.1145/1321631.1321639
Ibrahim Z, Money AG (2019) Computer mediated reality technologies: a conceptual framework and survey of the state of the art in healthcare intervention systems. J Biomed Inform 90:103102
Jackson D, Gilbert J (2023) WebGL 2.0 Specification. Khronos Group Specification
Jacob RJ, Girouard A, Hirshfield LM, Horn MS, Shaer O, Solovey ET et al (2008) Reality-based interaction: a framework for post-WIMP interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp 201–210
Jedlitschka A, Pfahl D (2005) Reporting guidelines for controlled experiments in software engineering. In: 2005 International symposium on empirical software engineering, 2005, p 10-pp
Jones B, Goregaokar M (2023) WebXR Device API. W3C Working Draft
Kaiser P, Vasak P, Suorineni F, Thibodeau D (2005) New Dimensions in Seismic Data Interpretation with 3-D Virtual Reality Visualisation for Burst-Prone Mines, pp 33–45
Kendall M (1938) A New Measure of Rank Correlation. Biometrika
Kobayashi K, Kamimura M, Yano K, Kato K, Matsuo A (2013) SArF map: visualizing software architecture from feature and layer viewpoints. In: 2013 21st International conference on program comprehension (ICPC), pp 43–52

Kononenko O, Rose T, Baysal O, Godfrey M, Theisen D, De Water B (2018) Studying pull request merges: a case study of shopify's active merchant. In: Proceedings of the 40th ICSE SEIP, pp 124–133
Laha B, Bowman D, Socha J (2014) Effects of VR system fidelity on analyzing isosurface visualization of volume datasets. IEEE Trans Vis Comput Graph 20:513–22. https://doi.org/10.1109/TVCG.2014.20
Liu S, Cui W, Wu Y, Liu M (2014) A survey on information visualization: recent advances and challenges. The Visual Computer 30(12):1373–1393
Maddila C, Bansal C, Nagappan N (2019) Predicting pull request completion time: a case study on large scale cloud services. In: Proceedings of the 2019 27th ESEC/FSE, pp 874–882
Maletic JI, Leigh J, Marcus A, Dunlap G (2001) Visualizing object-oriented software in virtual reality. In: Proceedings 9th international workshop on program comprehension. IWPC 2001, pp 26–35
Merino L, Fuchs J, Blumenschein M, Anslow C, Ghafari M, Nierstrasz O et al (2017) On the impact of the medium in the effectiveness of 3D software visualizations. In: 2017 IEEE Working conference on software visualization (VISSOFT), pp 11–21
Merino L, Ghafari M, Anslow C, Nierstrasz O (2017) CityVR: gameful software visualization. In: 2017 IEEE International conference on software maintenance and evolution (ICSME), pp 633–637
Milgram P, Kishino F (1994) A taxonomy of mixed reality visual displays. IEICE TRANSACTIONS on Information and Systems 77(12):1321–1329
Millais P, Jones SL, Kelly R (2018) Exploring data in virtual reality: comparisons with 2d data visualizations. In: Extended abstracts of the 2018 CHI conference on human factors in computing systems, pp 1–6
Millais P, Jones SL, Kelly R (2018) Exploring Data in Virtual Reality: Comparisons with 2D Data visualizations. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems. CHI EA '18. New York, USA: Association for Computing Machinery, pp 1–6. Available from: https://doi.org/10.1145/3170427.3188537
Misiak M, Schreiber A, Fuhrmann A, Zur S, Seider D, Nafeie L (2018) IslandViz: a Tool for Visualizing Modular Software Systems in Virtual Reality. In: 2018 IEEE Working Conference on Software Visualization (VISSOFT), pp 112–116
Moreno-Lumbreras D, Gonzalez-Barahona JM, Villaverde A (2022) BabiaXR: virtual reality software data visualizations for the web. In: 2022 IEEE Conference on virtual reality and 3D user interfaces abstracts and workshops (VRW). IEEE, pp 71–74
Moreno-Lumbreras D, Minelli R, Villaverde A, Gonzalez-Barahona JM, Lanza M (2021) CodeCity: On-Screen or in virtual reality? In: Working Conference on Software Visualization, VISSOFT 2021, Luxembourg, September 27-28, 2021. IEEE, pp 12–22. Available from: https://doi.org/10.1109/VISSOFT52517.2021.00011
Moreno-Lumbreras D, Robles G, Izquierdo-Cortazar D, González-Barahona JM (2021) To VR or not to VR: Is virtual reality suitable to understand software development metrics? CoRR. arXiv:2109.13768
Munzner T (2014) Visualization analysis and design. CRC Press
Natephra W, Motamedi A (2019) Live data visualization of IoT sensors using augmented reality (AR) and BIM. In: 36th International symposium on automation and robotics in construction (ISARC 2019)
Olshannikova E, Ometov A, Koucheryavy Y, Olsson T (2015) Visualizing big data with augmented and virtual reality: challenges and research agenda. Journal of Big Data 2. https://doi.org/10.1186/s40537-015-0031-2
Playfair W (1822) A Letter on Our Agricultural Distresses, Their Causes and Remedies: Accompanied with Tables and Copper-plate Charts, Shewing and Comparing the Prices of Wheat, Bread and Labour from 1565 to 1821. 23431. W. Sams
Ragan ED, Kopper R, Schuchardt P, Bowman DA (2013) Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task. IEEE Trans Vis Comput Graph 19(5):886–896. https://doi.org/10.1109/TVCG.2012.163
Raja D, Bowman D, Lucas J, North C (2004) Exploring the benefits of immersion in abstract information visualization. In: Proc. Immersive Projection Technology Workshop vol 61, pp 69
Ralph P (2021) ACM SIGSOFT empirical standards released. SIGSOFT Softw Eng Notes 46(1):19. https://doi.org/10.1145/3437479.3437483
Romano S, Capece N, Erra U, Scanniello G, Lanza M (2019) On the use of virtual reality in software visualization: the case of the city metaphor. Inf Software Technol 114:92–106. https://doi.org/10.1016/j.infsof.2019.06.007
Rosenbaum R, Bottleson J, Liu Z, Hamann B (2011) Involve me and i will understand!–abstract data visualization in immersive environments. In: International symposium on visual computing. Springer, pp 530–540
Rüdel MO, Ganser J, Koschke R (2018) A controlled experiment on spatial orientation in VR-based software cities. In: 2018 IEEE Working conference on software visualization (VISSOFT), pp 21–31

Sadowski C, Söderberg E, Church L, Sipko M, Bacchelli A (2018) Modern code review: a case study at google. In: Proceedings of the 40th international conference on software engineering: software engineering in practice, pp 181–190

Saket B, Endert A, Çagatay Demiralp (2017) Data and task based effectiveness of basic visualizations. arXiv:1709.08546

Schuemie M, Straaten P, Krijn M, Mast C (2001) Research on presence in virtual reality: a survey. cyberpsychology and behavior: the impact of the Internet, multimedia and virtual reality on behavior and society 05(4):183–201. https://doi.org/10.1089/109493101300117884

Sillito J, Murphy GC, De Volder K (2006) Questions Programmers Ask during Software Evolution Tasks. In: Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering. SIGSOFT '06/FSE-14. New York, NY, USA: Association for Computing Machinery, pp 23–34

Skarbez R, Smith M, Whitton MC (2021) Revisiting milgram and kishino's reality-virtuality continuum. Frontiers in Virtual Reality 2:647997

Ometov A, Olshannikova E, Olsson T, Koucheryavy Y (2015) Visualizing Big Data with augmented and virtual reality: challenges and research agenda. Journal of Big Data 2. https://doi.org/10.1186/s40537-015-0031-2

Sun C, Lo D, Khoo SC, Jiang J (2011) Towards more accurate retrieval of duplicate bug reports. In: 2011 26th IEEE/ACM international conference on automated software engineering (ASE 2011), pp 253–262

Sun C, Lo D, Wang X, Jiang J, Khoo SC (2010) A discriminative model approach for accurate duplicate bug report retrieval. In: 2010 ACM/IEEE 32nd International conference on software engineering vol 1, pp 45–54

Thongtanunam P, McIntosh S, Hassan AE, Iida H (2017) Review participation in modern code review. Empirical Software Engineering 22(2):768–817

Tian Y, Sun C, Lo D (2012) Improved duplicate bug report identification. In: 2012 16th European conference on software maintenance and reengineering, pp 385–390

Tufte ER (2001) The Visual Display of Quantitative Information, 2nd edn. Graphics Press, Cheshire, CT

Vegas S, Apa C, Juristo N (2016) Crossover designs in software engineering experiments: benefits and perils. IEEE Trans Software Eng 42(2):120–135. https://doi.org/10.1109/TSE.2015.2467378

Wettel R, Lanza M (2007) Visualizing software systems as cities. In: 2007 4th IEEE International workshop on visualizing software for understanding and analysis, pp 92–99

Wettel R, Lanza M, Robbes R (2011) Software systems as cities: a controlled experiment. In: 2011 33rd International conference on software engineering (ICSE), pp 551–560

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science & Business Media

Yano K, Matsuo A (2017) Data access visualization for legacy application maintenance. In: 2017 IEEE 24th International conference on software analysis, evolution and reengineering (SANER), pp 546–550

Young P, Munro M (1998) Visualising software in virtual reality. In: Proceedings. 6th international workshop on program comprehension. IWPC'98 (Cat. No.98TB100242), pp 19–26

Yu Y, Wang H, Filkov V, Devanbu P, Vasilescu B (2015) Wait for it: determinants of pull request evaluation latency on github. In: 2015 IEEE/ACM 12th working conference on mining software repositories. IEEE, pp 367–371

Zhang S, Demiralp C, Keefe DF, DaSilva M, Laidlaw DH, Greenberg BD et al (2001) An immersive virtual environment for DT-MRI volume visualization applications: a case study. In: Proceedings visualization, 2001. VIS '01, pp 437–584

Authors and Affiliations

David Moreno-Lumbreras[1] · Gregorio Robles[1] · Daniel Izquierdo-Cortázar[2] · Jesus M. Gonzalez-Barahona[1]

Gregorio Robles
gregorio.robles@urjc.es

Daniel Izquierdo-Cortázar
dizquierdo@bitergia.com

Jesus M. Gonzalez-Barahona
jesus.gonzalez.barahona@urjc.es

[1] Universidad Rey Juan Carlos, Fuenlabrada, Spain

[2] Bitergia, Leganés, Spain