TrustChain

Reputation system for app stores:

A case study on how to build an interface for support and decision making

R&D Team // February 2020

Aptoide

Lisb@20[20]

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

Table of Contents

Introduction

Nowadays, mobile apps are valuable links that connect companies to their customers anytime, anywhere. It is estimated that the average user spends 5 hours using a smartphone, being that 92% of this interaction is using such apps [1]. The App market reveals then an important relevance in establishing social and commercial relations in society. This relevance is also validated by the volume this market has achieved and its fast progress, with the worldwide consumer spending on mobile apps surpassing the 100 billion dollars [2], by the constant increase in downloads, where in 2016 were 140 billion and in 2019 almost surpassing the mark of 200 billion [2], and by the constant increase in revenue, with increases of almost 100 billion dollars, from 2018 to 2019, in a market expected to reach the 935 billion dollars revenue by 2023 [2].

However, the cybersecurity and the quality of these apps still presents considerable challenges, as the malware presence tripled in 2018 [3] and has no obvious solution.

As app stores have not being able to properly respond to this moving threat [4]–[7], and operate in a reactive and slow way, several users are inflicted before the new threat is neutralized. These same users flag the app as malicious and only then does an app store react to this new threat. This is a problem classified as being of moving target [8] due to the very high frequency of manifestation of new threats at an alarming rate.

In this context, the user's feedback regarding the apps they use is critical to allow app stores to know and block new threats, reducing their impact as much as possible (the shorten the time the threats are available on app stores, the lower the impact). However, these feedbacks are, many times, of low accuracy or false. This makes the app store services of cybersecurity and quality inefficient, increases their time for reaction and allows the success of the malicious intents for a considerable period, with impact in millions of users.

Aptoide, despite being considered one of the most secure app stores [9], it has the need for a disruptive approach for facing malware, as it currently contains more than one million apps in its repositories, with more than 200 million consumers worldwide and a rate of submission of 10 000 apps a day. Aptoide receives an average of 5 000 user feedbacks a day, where, around 950 of those feedbacks are reporting virus. Those feedbacks are then analysed by the Quality and Assurance (QA) team to investigate which ones are real threats. However, given the high flow of new apps, there around 20 000 apps pending to be reviewed. From those, it is estimated that between 10%-12% are a threat not detected by the Aptoide security system.

To decide which apps have to be inspected first, each QA professional defines a metric, as number of virus feedbacks, a process that is highly dependent on the QA analyst experience. Also, around 90% of those feedbacks are false, which seriously compromises the work of the QA professionals and the detection of new threats.

The TrustChain project has as strategic goal a considerable decrease of the impact of malicious mobile apps in society. Thus, we aim at creating an environment of security, trust and transparency for the app economy and for all the businesses that use it to get closer to their customers. To achieve this goal, we built a reputation system for consumers and publishers of apps, and an intelligent decision support system for quality assurance and cybersecurity that, based on the reputation of these players, considerably increases the efficiency and efficacy of the app store services. Moreover, the knowledge base related to the reputation will be decentralized in a blockchain, in order to be possible its adoption

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

and contribution from several app stores, thus generating a global app economy with increased security, trust and transparency.

The reputations of a user inside Trustchain is computed accordingly to its actions, where good actions increase the reputation, and bad actions (*e.g.* false flags, upload of malware) decrease it. A consumer is then considered a high reputation user by the system once its reputation reaches a certain value. Whenever a high reputation user provides a virus feedback on an app, the system updates that app's threat level, providing a prioritizable value to the QA analyst to analyze it, through our intelligent decision support system. Similarly, when a developer has low reputation, due to having submitted multiple apps classified as malware, any new submission by him should be considered as potentially malicious.

Building this intelligent decision support system for the QA experts was no easy task, has it had several challenges related with the visualization of cybersecurity threats, system evaluation and adherence of the system by the end users. In this paper, we present how we addressed those problems, and create and intuitive and useful interface for the QA experts.

First, we describe the approach to design and evaluate the interface, as well as the description of the type of users. Then, the description of the interface used by the QA analyst to study the apps. Next, how the managers can evaluate the performance of the system, followed by the interface for the management of reputation models. In the end, we present the evolution of user satisfaction, followed by some conclusions regarding the creation of a novel interface.

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

User-centered Design

To develop the best possible interface for the final users of the TrustChain system, we have used a user-centered design [10] approach for the development of the interface. Using this approach, allowed us to tailor as much as possible the solution to the users need, while providing new and crucial information for the cybersecurity processes. Figure 1 depicts the base process we used to deliver the final interface.

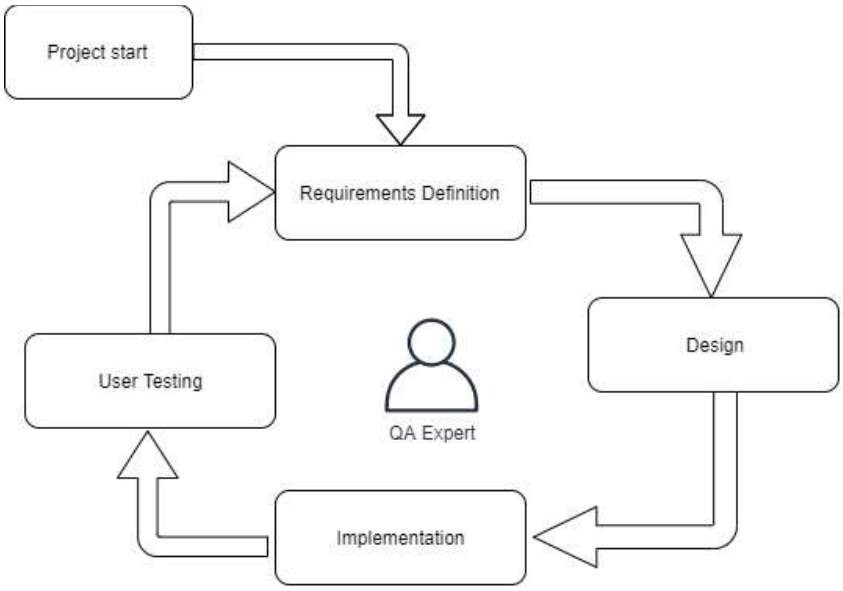

Figure 1 - User-Centered Design approach.

Firstly, we started by understanding the user context by studying the already in place tools, together with the proper research in reputation systems and Aptoide's cybersecurity processes. The study of the previous tools was an important step in our research, as we wanted to minimize difficulty in adherence to the tool by proving an interface close to what was previous known, whilst providing new and powerful information for the cybersecurity processes.

From that research, we identified two primary users:

- QA Analysist
  - General knowledge about the Aptoide's internal flow, i.e., they must be Aptoide employees.
  - Experience in reviewing apps with the Aptoide's tools.
  - Good discerning capabilities: ability to evaluate possible threats (adware, spyware, etc.), or fake apps.

- QA Manager
  - General knowledge about the Aptoide's internal flow, i.e., they must be Aptoide employees.
  - Deep knowledge of the process's involved in the QA analysis flow.
  - Experience in managing QA analysts.
  - Knowledge about Aptoide's cybersecurity processes.

To define the initial requirements, we conduct a focus group with the stakeholders of the project. First, we wanted to officially present the project, where we introduced concepts such as reputation. Then, we promoted a discussion about their old tool, to understand their needs and how it could be improved. Besides the focus group, we also did on-site observation. Here, we wanted to perceive the user environment, their process of analyzing apps and how was their workflow. This on-site observation, as well as individual interviews were constant throughout the duration of the project.

From those requirements, we built the first wireframes, that were, again validated by the stakeholders. Those wireframes were the basis for the first release of the project. The project had a total of 3 releases: 1. a proof-of-concept implementation, through mockups; 2. a working prototype; 3. A real world implementation.
For each of those iterations, the users had an opportunity to test the interface, through user testing. This allowed to further tailor our solution to their needs, as we could identify friction points and eliminate them.

User Testing Definition

To ensure that the interface was adjusted to the users, and the available functionalities were indeed capable of improving the cybersecurity processes, we asked our QA professionals to test the system. To evaluate the results of the user testing, we used usability metrics, defined by the ISO 9241-11 standard as "the extent to which a system, product or service can be used by specified users to achieve specified goals with **effectiveness**, **efficiency** and **satisfaction** in a specified context of use"[11].

To create the user tests, we followed 6 steps:

- Create the user's profiles.
- Define what to test: functionalities (requirements).
- For each functionality to test, create and hypothesis, *e.g.* "User has some friction creating a new account"
- Create the task/scenario to insert on the script (tell a story)
- Write the script: introduction, pre-word, background questions, tasks/scenarios.
- Define the software and equipment to use.

During the tests, the screen was recorded, as well as the audio, with the previous consent from the user, to help us better review the tests. Throughout the test, the users were incentivised to think aloud, and to explain their thoughts. The users were also constantly reassured that they were not under evaluation, but rather the system was. For each test, we had two members of the team, one to guide the tester through the session, and another member to observe and takes notes. Those notes, together with the screen recordings and audio, were a valuable resource, as we could precisely identify where the users were having difficulties, and what functionalities had to improved. We must also highlight that the tests were always performed by Aptoide's QA team, which, despite having one or two new members, it was mostly the same during the project's timeline. Though mandatory, since we needed testers with the necessary knowledge about Aptoide's flow and app security expertise, this brought up some constrains to the evaluation. First, the users learned from one iteration to another, which increases their performance in testing the system.

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

Second, since the users were Aptoide's collaborators, therefore, not neutral to the R&D team, their opinion on the system may have some positive bias. We always considered this situation while analyzing the questionnaires. We also countered this situation with the support of our notes and recordings of the sessions. Third, we did not require the users to have previous knowledge of reputation systems, something we also considered during our test analysis.

We defined 6 scenarios, 3 for the QA analyst, and 3 for the QA manager. Each scenario was comprised of a sequential set of tasks, to achieve the goal defined by the scenario story.

Effectiveness

First, we evaluated the system regarding its effectiveness to understand if the users could perform the proposed. This is defined by the completion rate,

$$effectiveness = \frac{Number\ of\ tasks\ completed\ succesfully}{Total\ number\ of\ tasks\ undertaken} \times 100\ .$$

For each task uncompleted, we saw it as default in our system, and not a problem with the user.

Efficiency

Not only is important to complete all the tasks but is also important that those tasks are completed within a reasonable time, otherwise, the system won't improve the cybersecurity processes. Efficiency is then measure in terms of task time, the time (in seconds or minutes) the participant takes to successfully complete a task,

$$task\ time = End\ time - Start\ time.$$

Then, for we computed the time-based efficiency of all tasks,

$$Time\ Based\ Efficiency = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N}\frac{n_{ij}}{t_{ij}}}{NR},$$

and overall relative efficiency,

$$Overall\ Relative\ Efficiency\ = \frac{\sum_{j=1}^{R}\sum_{i=1}^{N}\frac{n_{ij}}{t_{ij}}}{\sum_{j=1}^{R}\sum_{i=1}^{N}t_{ij}} \times 100,$$

where:

N = The total number of tasks (goals)

R = The number of users

n_{ij} = The result of task i by user j; if the user successfully completes the task, then N_{ij} = 1, if not N_{ij} = 0

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

t_{ij} = The time spent by user j to complete task i. If the task is not successfully completed, then time is measured till the moment the user quits the task.

With these measures, we could understand how easily it was for the users to perform the required tests.

Satisfaction

Finally, we wanted to understand how the users felt while using the interface, and if they wanted to make improvements on the interface. This satisfaction is measured through standardized satisfaction questionnaires which can be administrated after each task and/or after the usability test session. Here, two sets of questions were done, first closed questions, then, open questions.

For the closed questions, we used Likert scale [12], from 1 (strongly disagree) to 6 (strongly agree). We purposefully removed the neutral option, to force the user in choosing a side. We defined 9 questions:

- Q1. It was easy was to perform the required tasks
- Q2. The interface is intuitive and user friendly
- Q3. I'm satisfied with the functionalities of the Trustchain
- Q4. The interface delivers what was defined in the focus group
- Q5. The Trustchain, as it is now, is useful for my job
- Q6. I would recommend the Trustchain to my colleagues
- Q7. What was the most difficult task?
- Q8. What do you think we could add to the Trustchain to help you in your job?
- Q9. What do you think we could remove/change in the current implementation?

The last three question were open, so that the user could feel free to write.

By comparing the answers of each iteration, we can not only evaluate the evolution of satisfaction, but also how we were able address the user's concerns and improve the interface. We present those evolutions in the final sections of the paper.

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

Decision support mechanisms for automated quality control of app store cybersecurity's services

The process of reviewing apps, that were flagged by our community is an important step in Aptoide's cybersecurity process's. Therefore, the design of the interface to be used by the QA analysts was of great concerned for the R&D team. We wanted to, not only intuitively represent the reputations of the users that interacted with the apps through the flags, but also to incorporate the information already used by the team in other analysis dashboards. Such approach allowed to minimize the difficulty that the QA analyst may had in using the Trustchain system.

The default dashboard for the QA analyst is the list of apps flagged as virus by high reputation users (Figure 2). This list is based on the system already used by the analysts, where the list of apps flagged are ordered by date, contrasting with our system, where the apps are ordered by threat level.

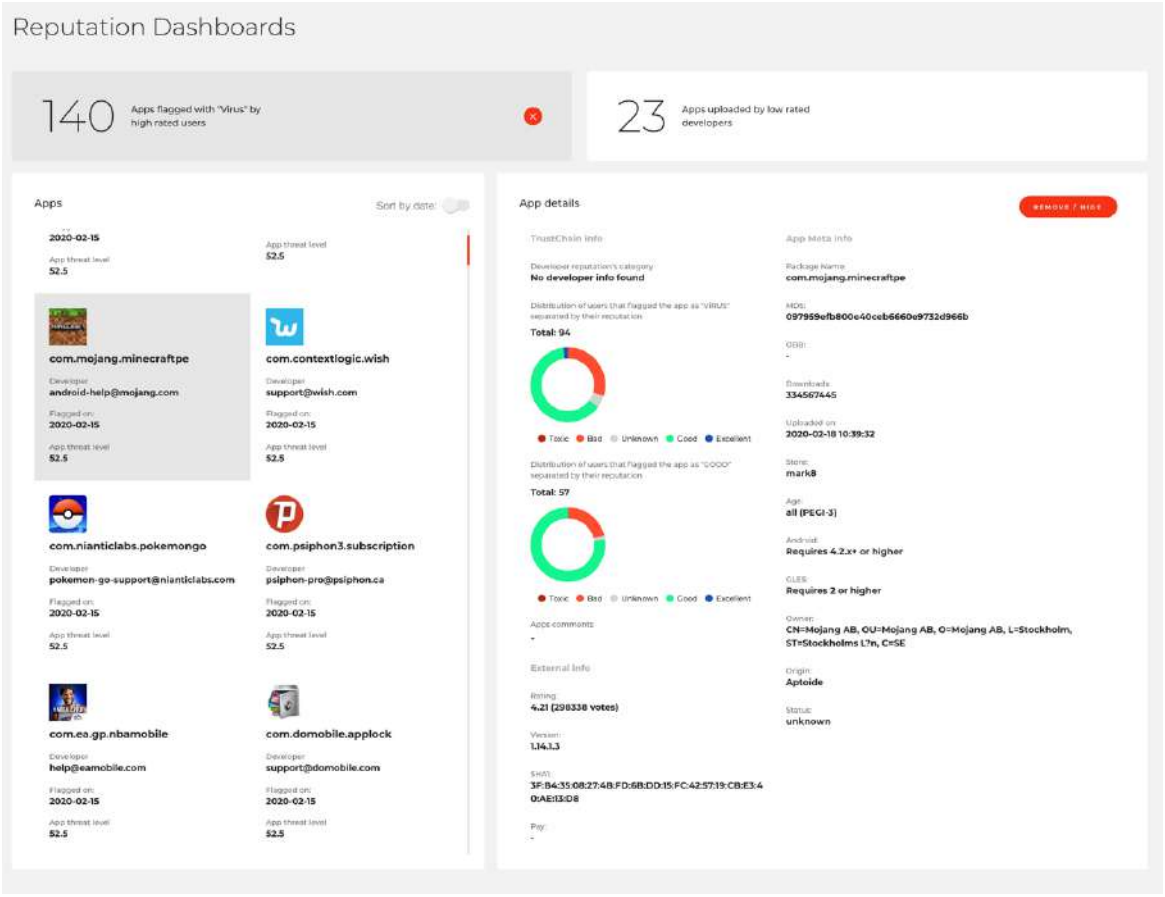

Figure 2 - Dashboard for the QA analyst, with the list of apps flagged as virus by high reputation users.

This interface is divided in two components, on the left, the list of apps that need to be investigated, and on the right, the app details.

The list of apps was carefully designed to give the analyst a quick understanding of each app, before the analysis itself. To achieve this, the apps are displayed in cards, containing the app icon (an important requirement defined by the QA analysts), the name of the developer, the date of the first flag (showing how long the app has been in the system) and the threat level of the app.

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

This list of apps can also be ordered by date of the first flag, allowing the analyst to investigate the most recent apps that were flagged as virus.

Regarding the app details, shown for each selected app, we start by displaying information about the reputation of the developer. Here, beside information already presented in the previous systems, we added two charts, one with the distribution of reputations of users that flagged the app as “Virus” and other with the distribution of reputations of users that flagged the app as “Good”. These pie charts allow the analyst to understand the type of users that interacted with the app, and how reliable is feedback given by the users.

The evaluation of a new imminent threat, given by a submission of a new app by a low rated developer and the consequent real time notification of the QA professional was a sensitive issue, as it was not clear how to generate that notification. Originally, the goal of the notification was to capture the attention of the professional and guide them to threat, but without overtly disrupting him. As so, we proposed a notification through email. However, when discussing this with the QA team, it was settled that an emailing system was not the preferred method, both because it could easily be ignored, and could also lead to unnecessary accumulation of emails. An alternative was conceptualized, through a list that only contains apps submitted by low rated developers, where the professionals would still be aware of new threats, and their level of urgency (Figure 3).

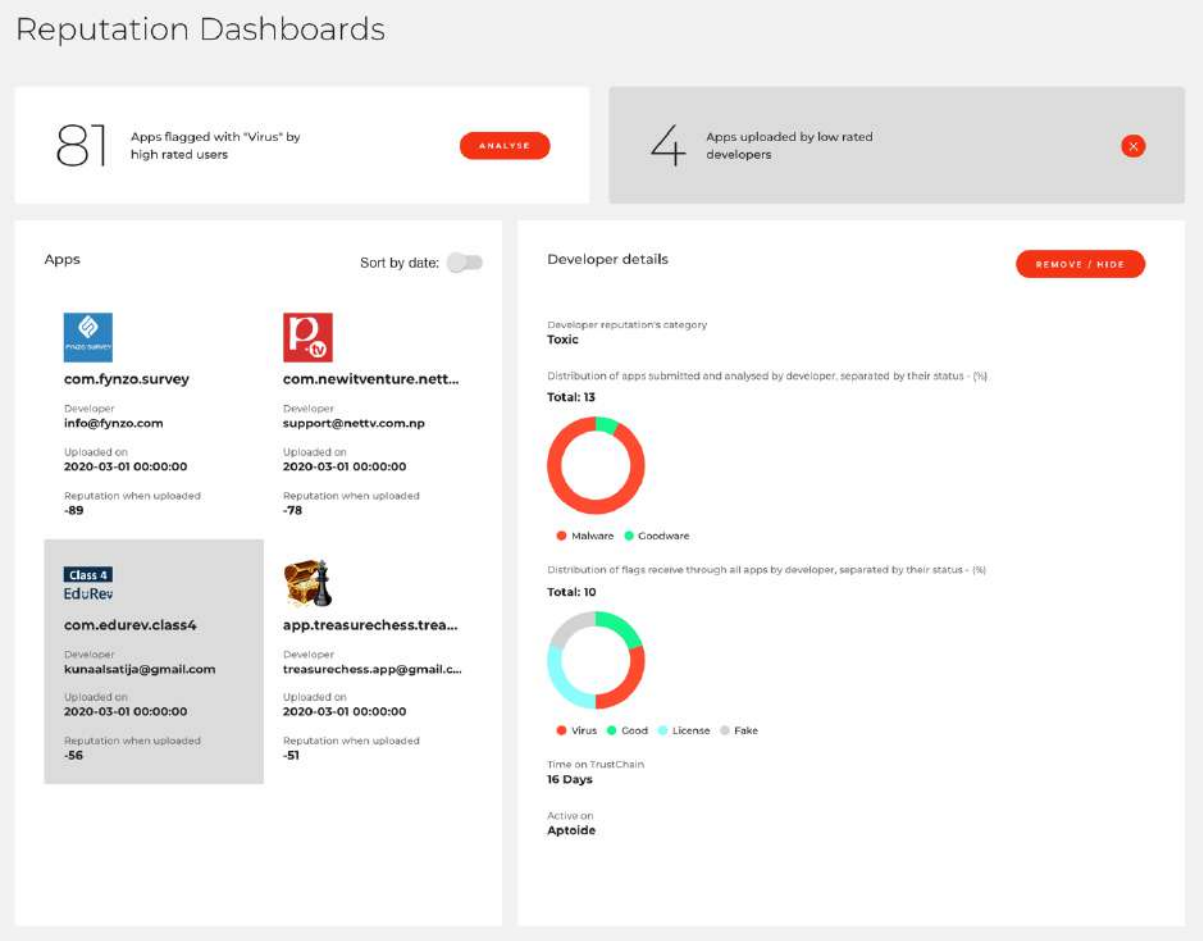

Figure 3 - Dashboard screen regarding apps uploaded by low rated developers.

As in the previous interface, this module is also divided in two components, one with the list itself, another with the developer's details.

The cards in the list contain the package name of the app, the developer and the reputation of the developer. Then, the developer's details present the analyst with the distribution of apps submitted by the developer, regarding their classification. The analyst can also observe for how long the developer has been active on the store, and on how many stores is he active.

Management of the system

Understanding the impact that the reputation system is having on Aptoide's cybersecurity processes is no easy task, as it is crucial that the QA manager can efficiently understand this impact, through pre-defined KPIs, and, if necessary, adapt the reputation model to improve the system. So, when developing the visualizations that empower this task, we carefully designed them to provide the necessary information related with quality control and cybersecurity, without overloading the QA manager with too much information.

To facilitate the process of the system evaluation, we have divided this KIPs page in 3 parts: 1) rapid overview of the system KPIs; 2) overview of the reputations landscape; 3) overview of cybersecurity processes.

On top of our KPI page, we have the information about apps pending analysis by the QA professionals: on the left the number of apps flagged by top users more than 24 hours ago and have not been analysed yet, and on the right the number of apps uploaded by low rated developers more than 24 hours ago.

Figure 4 -Widget representing the number of flagged apps by top users and uploaded apps by low rated developers, both with more than 24 hours without analysis

Both cases represent potential vulnerabilities in the security of the app store, so this is high priority information. This will help the manager to quickly understand the potential threats in the app store.

The visualization of the reputation landscape is a crucial feature for a QA Manager, as it allows to evaluate how the reputations of users is evolving, and the type of users that are present in the system.

To visualize this landscape, we provide a 6-month window for the QA manager to observe how the reputation of consumers and developers are distributed over the 5 different reputation levels, as displayed in Figure 5, for consumers, and Figure 6 for developers. By observing the distribution of reputations, the manager can understand the percentage of users that can be trusted, as well as the percentage of bad users in the system.

Ideally, with a reliable reputation system, and the necessary techniques to engage the users in behaving well, the manager would observe an increase in green and blue colors, with the corresponding decrease in red and orange.

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

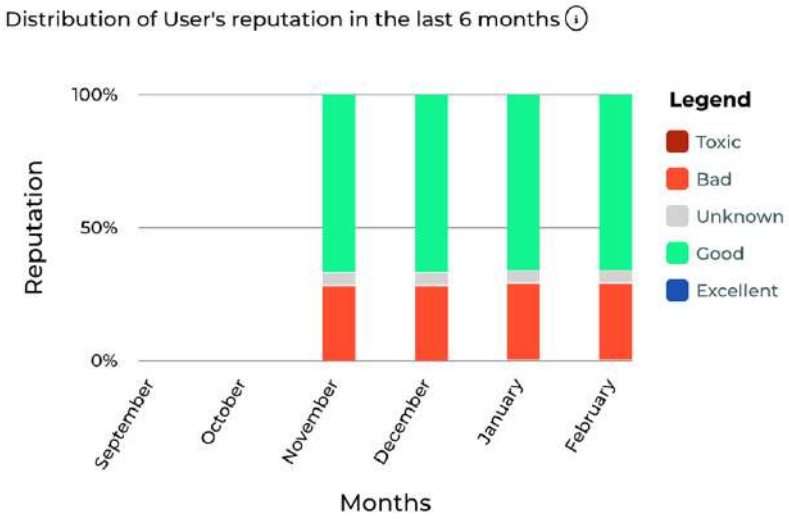

Figure 5 - Chart representing the distribution of User's reputation in the last 6 months

In the case of the visualization of the developer's reputation distribution, the QA manager also gains the ability to predict how many issues should appear in the near future, with a higher number of low rated developers indicating that there is a higher probability of low-quality apps entering the app store.

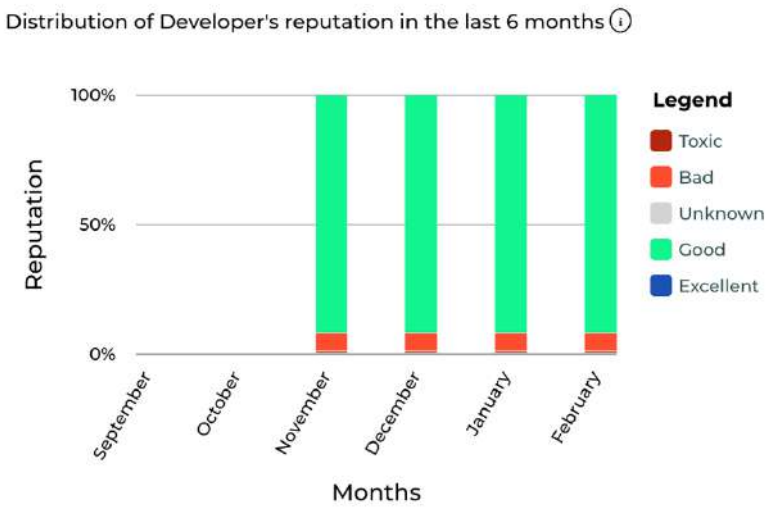

Figure 6 - Chart representing the distribution of Developer's reputation in the last 6 months

In order to have a more detailed analysis about the distribution of scores attributed to users or developers respectively, two graphs were developed to illustrate the number of users / developers who obtained a given score. The following Figure 7 and Figure 8 representing that distribution.

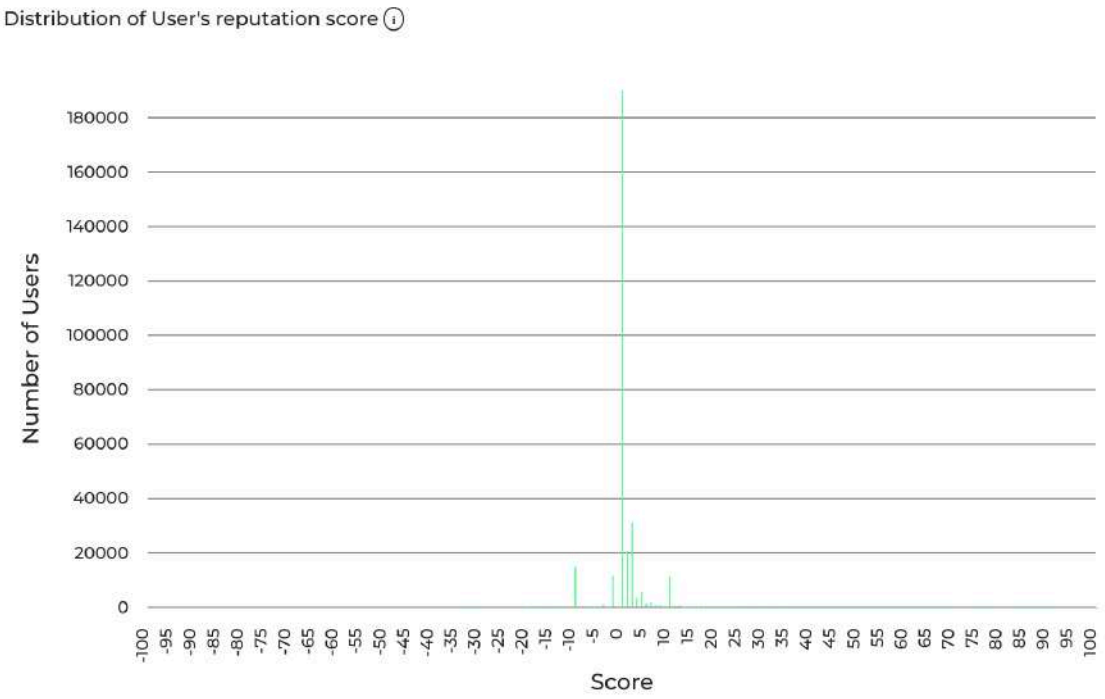

Figure 7 - Chart representing the number of users with a respective reputation score

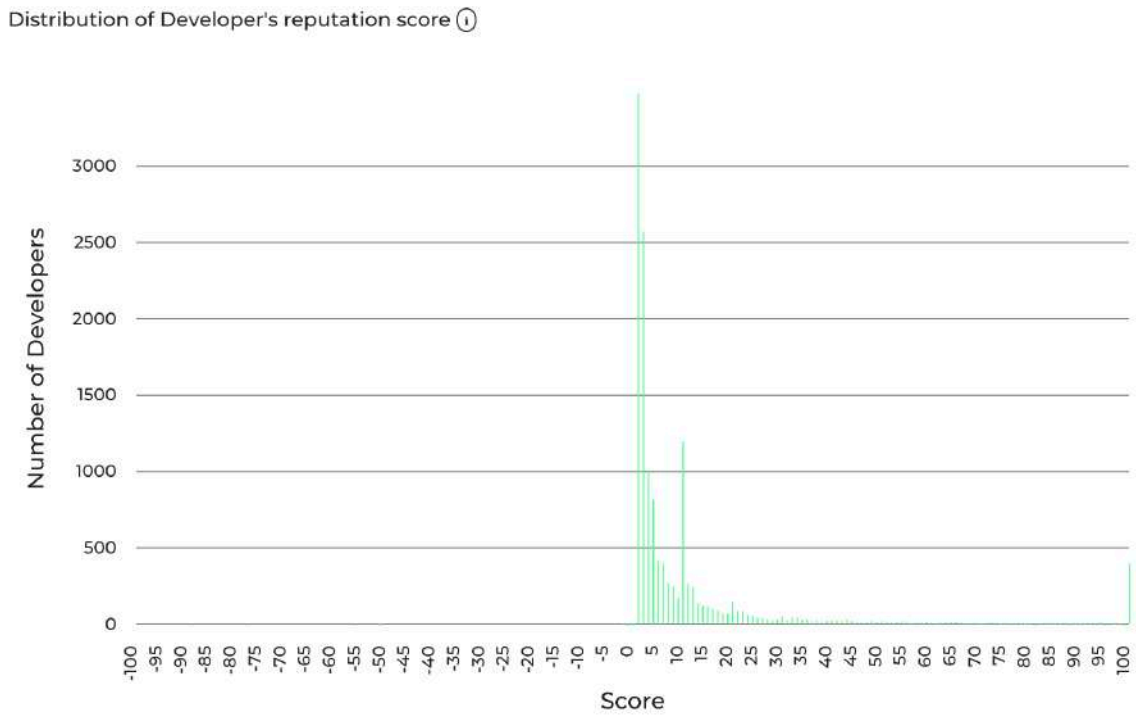

Figure 8 - Chart representing the number of developers with a respective reputation score

To evaluate the efficiency and efficacy of the cybersecurity processes associated with the Trustchain, we provide to the QA manager a dedicated section for this analysis.

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

First, a plot to understand the ratio of the number of apps analysed against the total number of apps to be analysed (Figure 9). Here, the manager can choose to visualize this information for the last 6 months, 6 weeks or 6 days. This granularity is important, as it allows the manager to understand the rate of new apps, against the resources employed to analyse them, in different times spans.

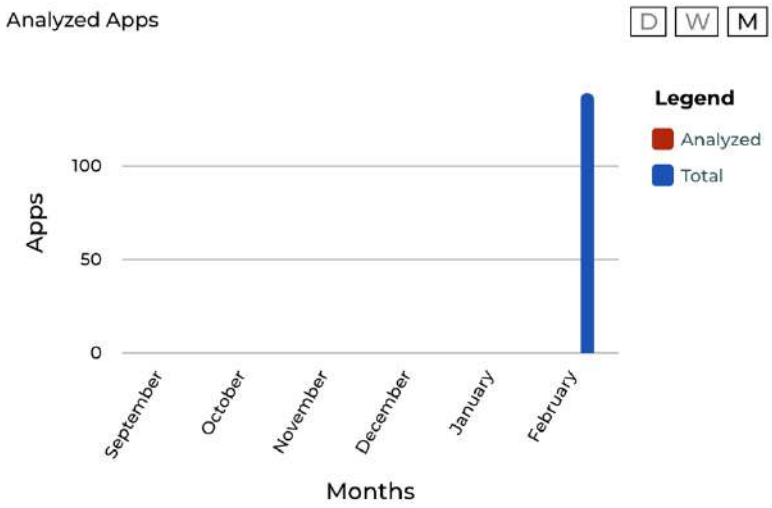

Figure 9 – Chart representing the number of analysed apps

To understand how valuable is the feedback given by the consumers, besides the plot with the reputations, the QA manager can also evaluate this through the feedback truthfulness plot, depict in Figure 10. Here, we plot the total number of feedbacks given by consumers (in blue) with the total number of feedbacks that were true (in red), in the last 6 months.

An increase in the difference between these numbers indicates that there should be an overall degradation of the reputation levels of the user base, with the opposite also being true.

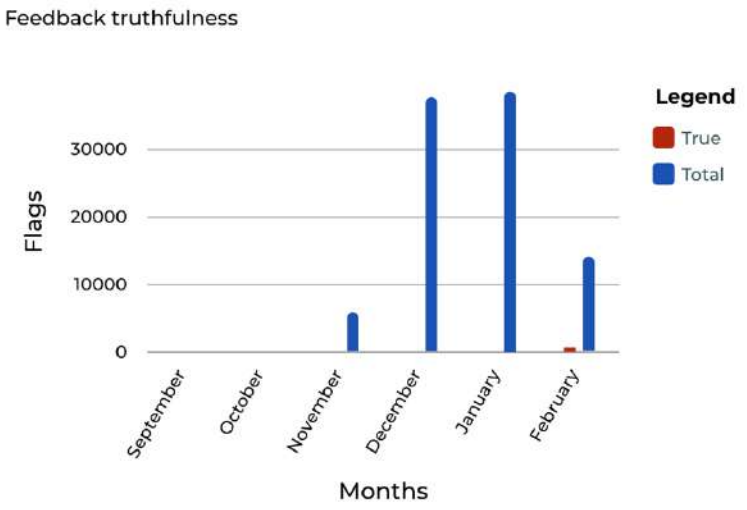

Figure 10 – Chart representing the distribution feedback truthfulness in the last 6 months

With the tools provided, it is expected that there will be a stabilization of the reputation levels which will in turn potentiate our system and create a safer app store. Ideally, this should be followed by a decrease in negative feedbacks by the users, as the Aptoide's is becoming faster in responding to real threats. To evaluate this, we present the number of negative feedbacks by high rated users in our system in different timescales, Figure 11, ideally decreasing over time as developers are more accurately classified as low rated, and their submissions quickly analysed and removed if unsafe, while also having an additional layer of security provided by the network of high rated users that identify unsafe apps from more unexpected sources. The activity from these users can be seen in a visualization with a configurable time window to understand how the activity by these users is evolving.

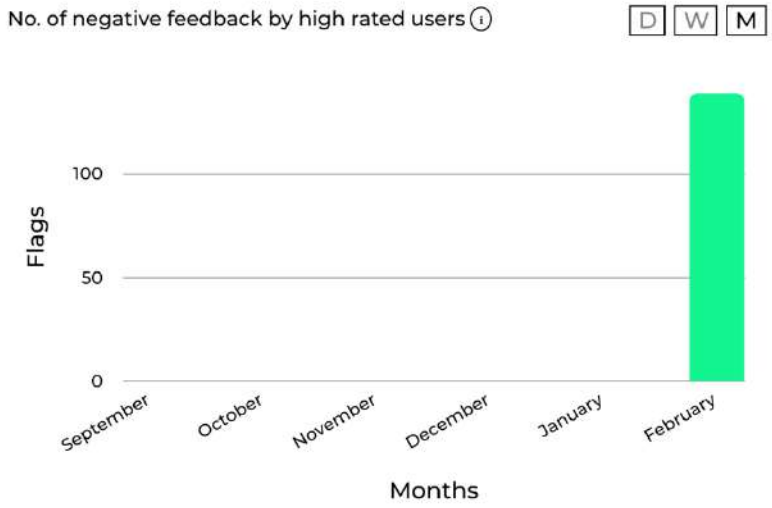

Figure 11 – Chart representing the number of negative feedbacks by high rated users

We also provide the QA Manager with a visualization of the number of apps left to analyse in the system submitted by low rated developers, Figure 12, either in the last 6 months, 6 weeks, or 6 days. With this information the manager can deduce specific periods where either the workforce was overloaded or there was an unusually high amount of potentially dangerous submissions.

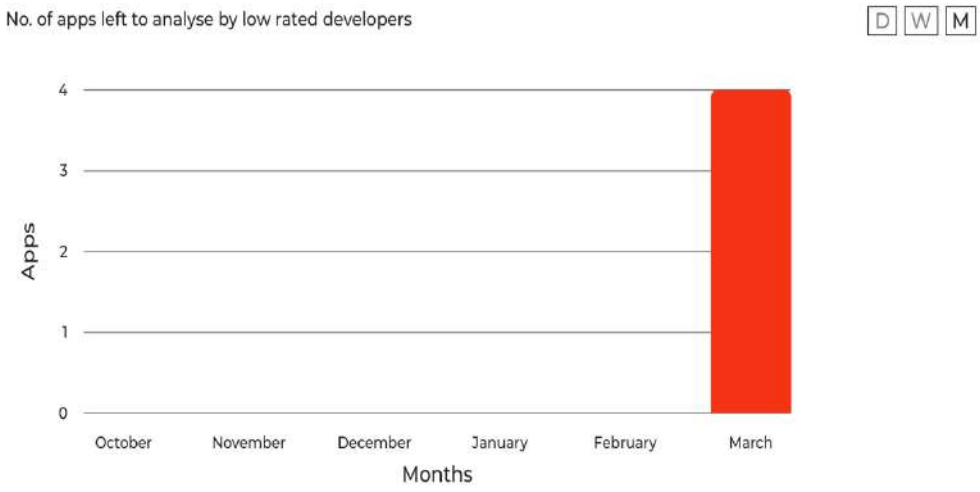

Figure 12 – Chart representing the number of apps left to analyse in the system submitted by low rated developers

In order to understand how the distribution of different types of user's actions has evolved over time, has been created a graph to represents this distribution. The following Figure 13 shows this chart.

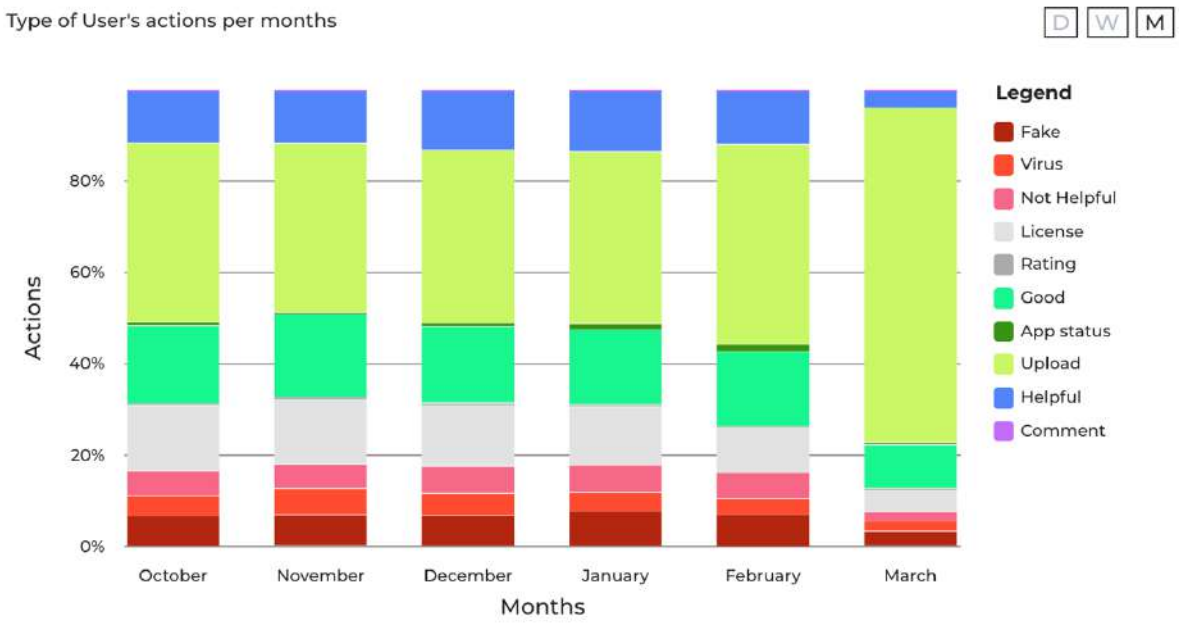

Figure 13 - Chart representing the distribution of the different type of user's actions

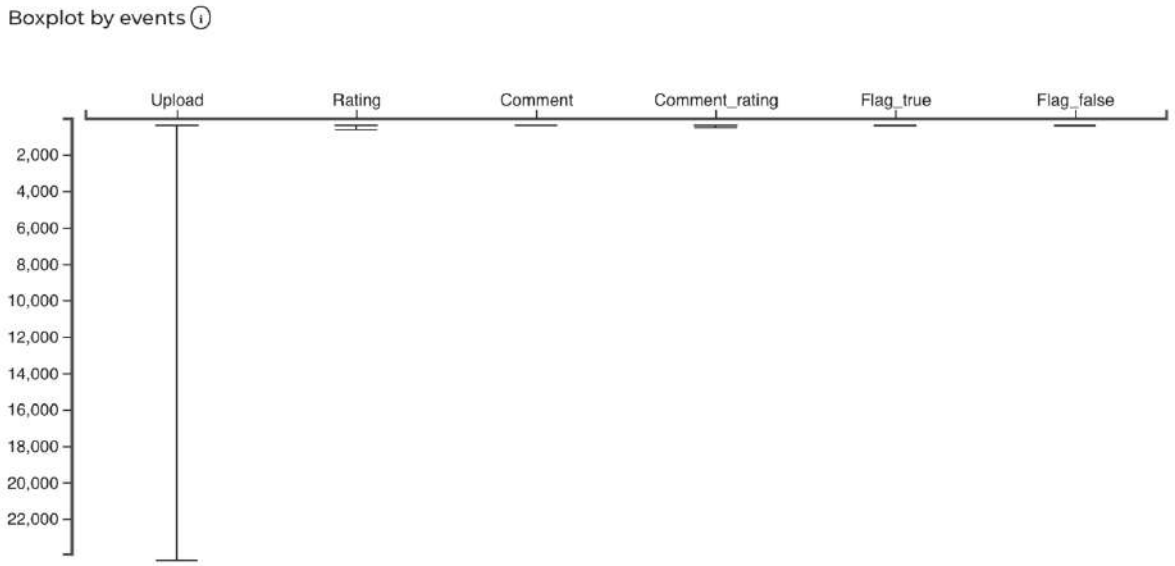

Figure 14 - Chart representing a boxplot by events.

Management of consumers and app developer's reputation' consolidation mechanisms

Adaptability is very important in reputation systems, as the behaviour of the users may change, or the business may evolve, leading to mandatory changes in the reputation system, to fine tune the variables, in accordance to the analysis done using the KPIs shown previously.

With the reputation being computed as a sum of integer values associated with actions, the impact of each action is clear. Also, a system based on individual actions and their associated points has the advantage of allowing the QA professional to easily adapt the variables values, and continuously adjusting and iterating on existing models.

Two screens for managing the variables were implemented, on for the consumers, Figure 15, and one for the developers, Figure 16. Here, it is also possible to observe the previous models.

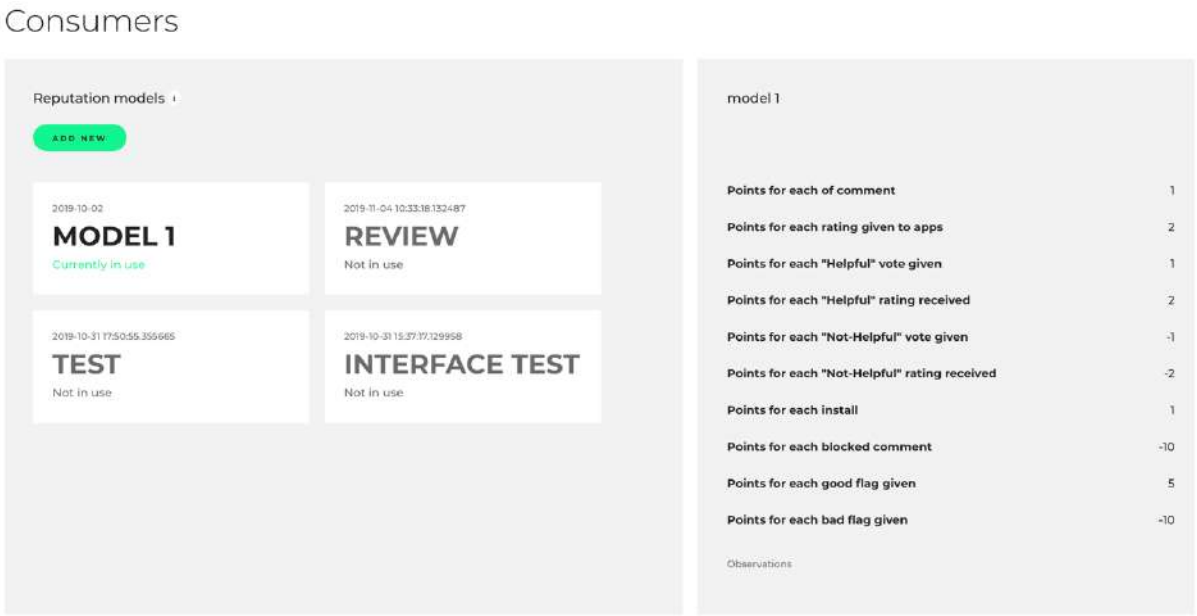

Figure 15 -Configuration screen regarding Consumer's reputation models.

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

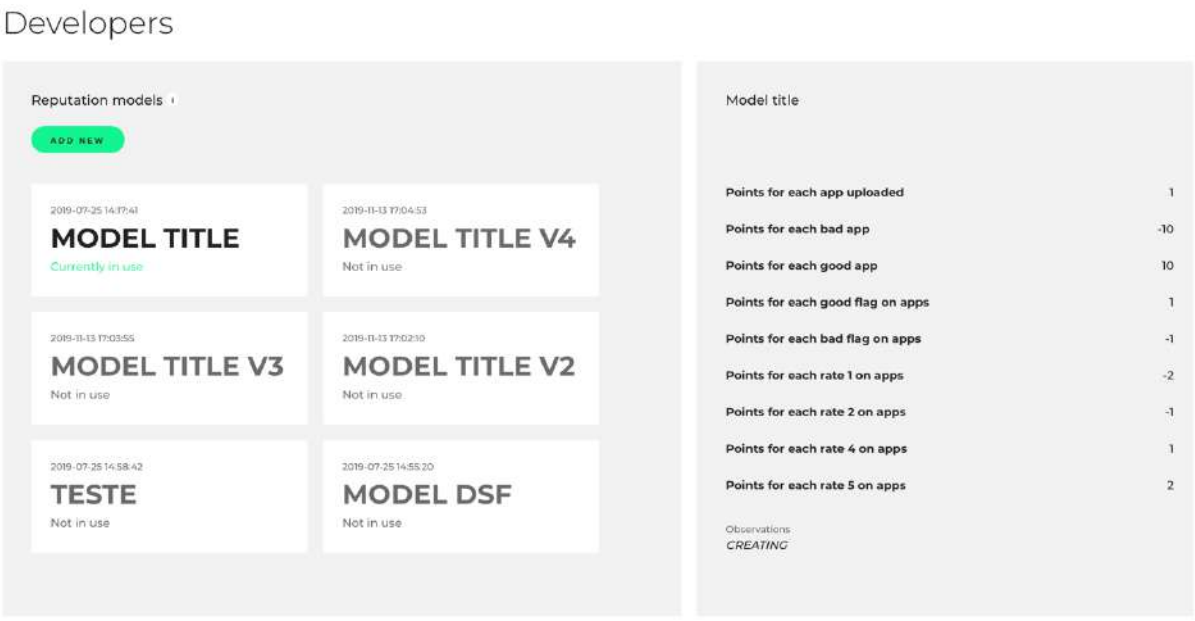

Figure 16 - Configuration main screen regarding Developer's reputation models.

Upon the creation of a new model, the QA manager simply has to assign the necessary points as he sees fits (Figure 15 and Figure 16) and the models, after saving, goes into production immediately.

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

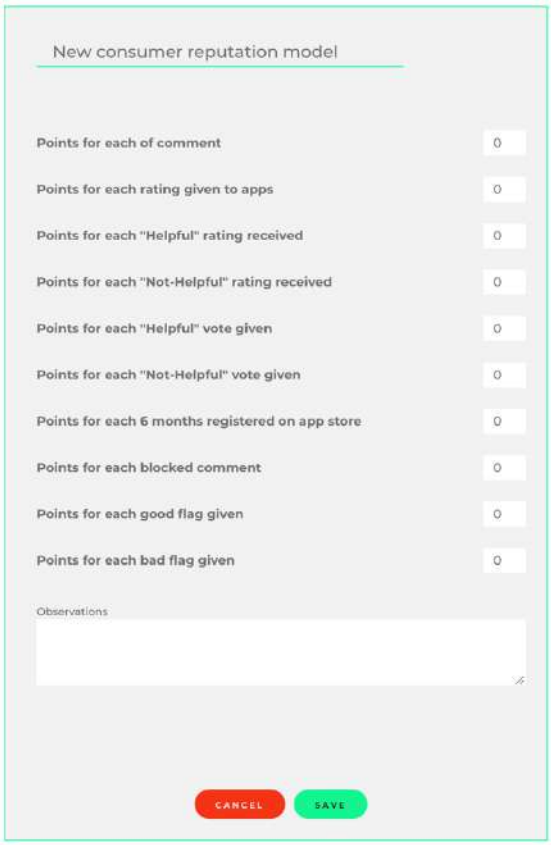

Figure 17 – Interface section where the QA Manager could define a new Reputation Model for a Consumer.

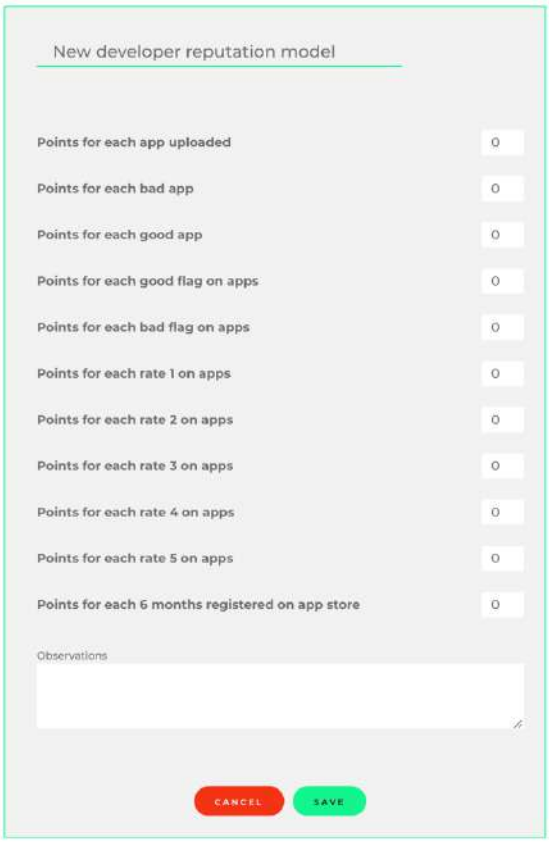

Figure 18 - Interface section where the QA Manager could define a new Reputation Model for a Developer.

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

User Testing Results

We continuously evaluated our interface using the defined metrics and methods, which led to the following results.

Regarding effectiveness, it was always 100%, as the users were always able to complete the tasks. However, they clearly had some difficulties, as the efficiency was very low (Figure 19). Here, we defined a baseline for each scenario, defined by us, as expert users. Figure 19 contains the comparison between the average times and the baseline, for the first iteration, and **Error! Reference source not found.** for the second one. Except on scenario 4 and 6, it is possible to observe that, from one iteration to another the users took, on average, less time to complete the tasks, thus closer to our defined baseline. Scenario 4 kept the times because the users had to analyse the KPIs, which does require some time for non-expert users. As for the last scenario, the users had to change the reputation model, a scenario with a contrary tendency, as we see an increase in the time to perform the tasks. However, we do not consider this increase a negative sign, since changing the values of the reputation model requires some time. This increase reflects the greater knowledge that QA managers have regarding reputation systems, thus spending more time considering the values to be changed, as we did not ask for specific changes in the model, but rather to just change it.

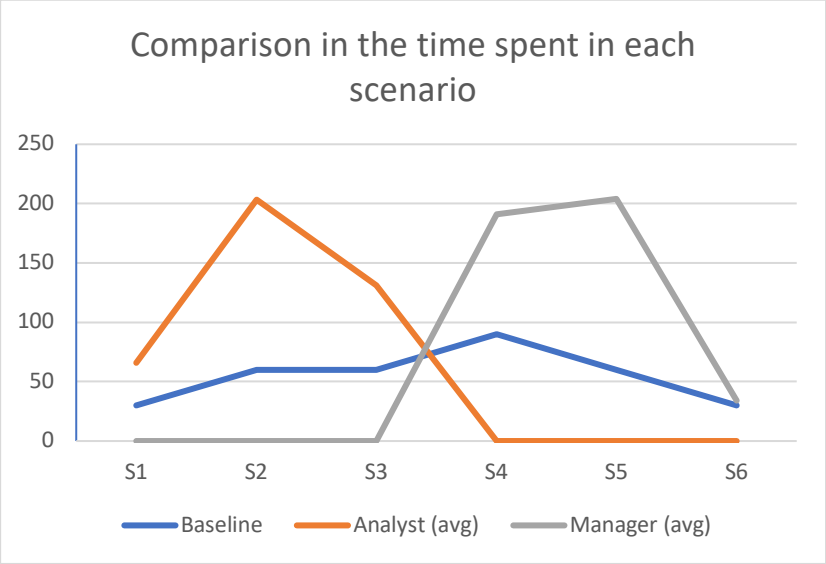

Figure 19 - Comparison of the average times for the first user testing. The y-axis represents the seconds, while the x-axis are the scenarios.

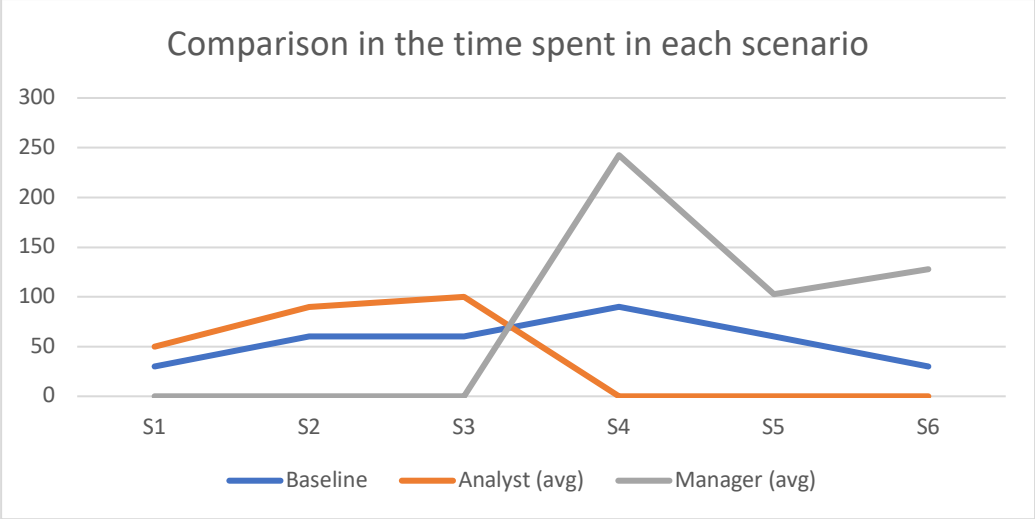

Figure 16 - Comparison of the average times for the second user testing. The y-axis represents the seconds, while the x-axis are the scenarios

To compare the satisfaction between iterations, we plotted the answers of each iteration and compared (Figure 20 to Figure 25). As it is possible to observe, the user satisfaction increased constantly, showing how we were able to meet the user's requirements and address their concerns.

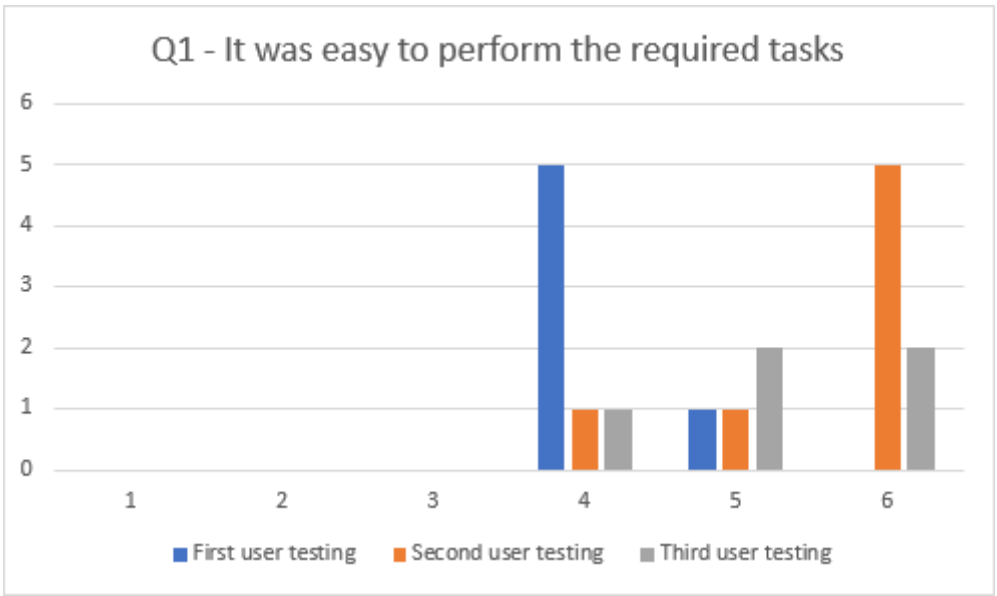

Figure 20 - Evolution of satisfaction regarding how easy it was to perform the tasks.

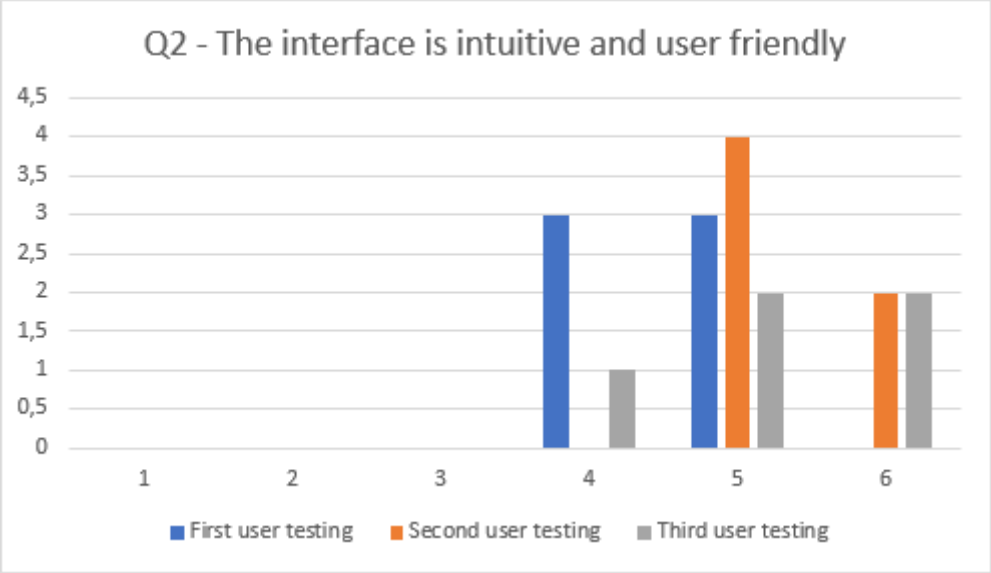

Figure 21 - Evolution of satisfaction regarding the interface.

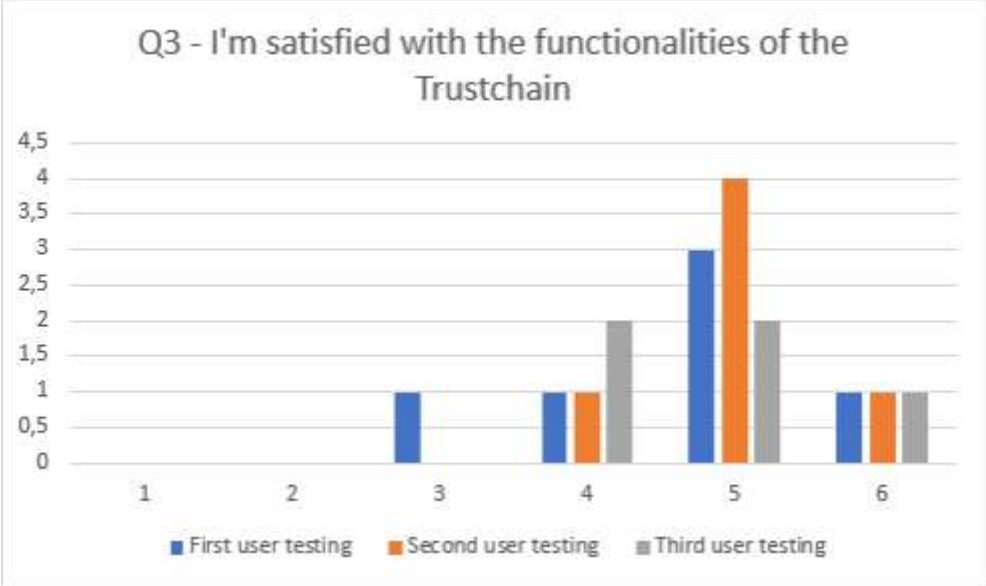

Figure 22 - Evolution of satisfaction regarding the functionalities provided.

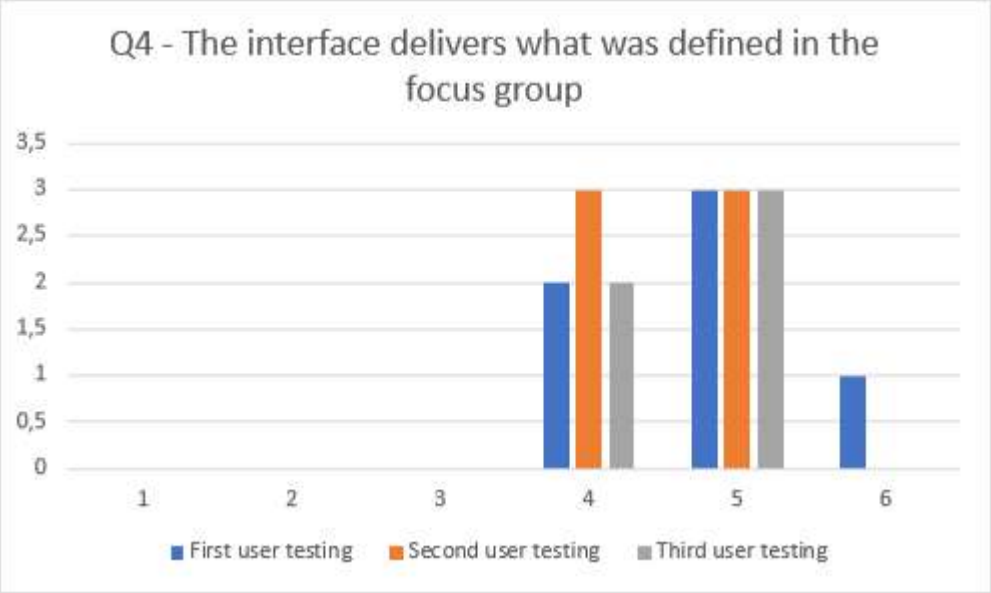

Figure 23 - Evolution regarding the delivery of functionalities

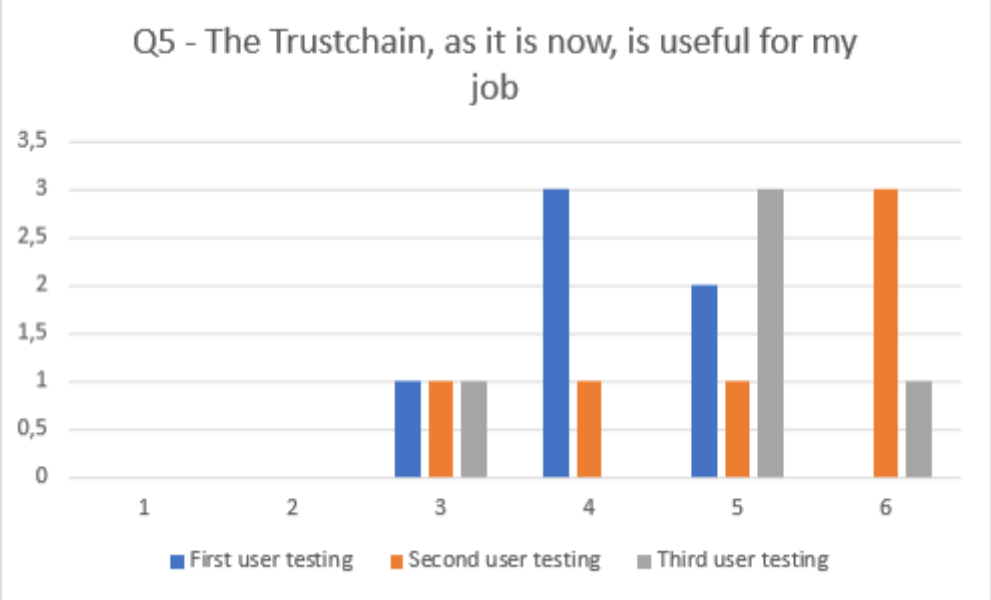

Figure 24 - Evolution of usefulness of the TrustChain.

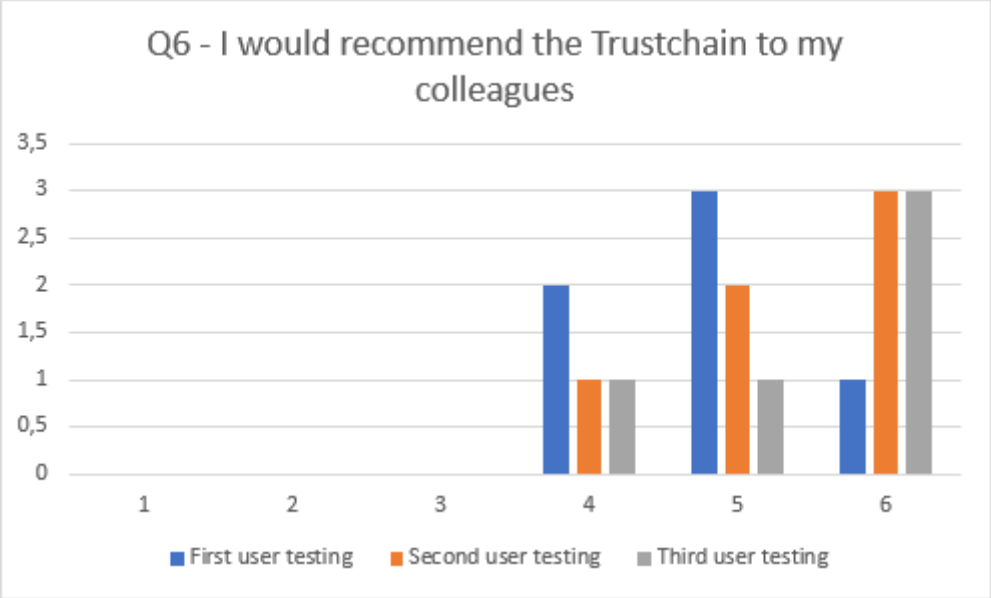

Figure 25 - Evolution of TrustChain as a recommendable system to other professionals.

Conclusion

Building the interface for the such novel system was no easy task. Besides the research associated with it, we also had to ensure that our users could use it, despite their lack of knowledge in reputation systems. That usability, as demonstrated, was achieved through a close relationship with the end users, the QA team. We constantly had this present in our development cycles, and we did not hesitate on hearing their concerns and suggestions. However, this approach come with its own adversities. Besides being a time-consuming task, as the QA is comprised of several members, with different backgrounds, these users did not have a clear idea of what may be the requirements of the project, as they do not know how reputation systems work.

We do considerer that the interface is not complete, and that there are still some improvements to make, especially regarding the visualization of KPIs, where there is still much that can be shown.

Lisb@20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

About Aptoide

Founded in 2011, Aptoide is the first decentralised and social Android App Store. With nearly 200 million users, 4 billion downloads and 1 million apps, Aptoide is a community-based platform that reinvents the app discovery experience through a social environment, tailored recommendations and the opportunity for users and partners to create and share their own personalised App Stores. Aptoide also provides white-label App Store solutions for developers, OEMs, Telcos and Integrators, through which they can upload and distribute their Android apps. Different versions of Aptoide are available for mobile, TV and VR devices and is accessible in over 40 languages. Aptoide is the 692nd most visited Website worldwide, according to Similarweb. Amongst its main partners are OEM's like Oppo and Vivo, and app publishers such as ZeptoLab and Goodgame Studios. It currently has more than 85 employees from 15 different nationalities, has its HQ in Lisbon and local offices in Shenzhen (China) and Singapore.

Bibliography

[1] TechCrunch, “U.S. consumers now spend 5 hours per day on mobile devices,” 2017. [Online]. Available: https://techcrunch.com/2017/03/03/u-s-consumers-now-spend-5-hours-per-day-on-mobile-devices/.

[2] J. Clement, “Mobile app usage - Statistics & Facts.” 2019.

[3] McAfee, “McAfee Mobile Threat Report Q1, 2019,” 2019.

[4] McAfee - Intel Security, “Mobile Threat Report,” 2016.

[5] Engadget, “‘Secure’ apps in Google’s Play Store are a crapshoot,” 2016. .

[6] Tech Desk, “Judy malware on Google Play Store: How to check if your Android device is safe,” 2017.

[7] R. Whitwam, “Security firm Check Point says millions infected with botnet malware via Play Store,” 2017. .

[8] F. Mercaldo, C. A. Visaggio, G. Canfora, and A. Cimitile, “Mobile malware detection in the real world,” in *Proceedings of the 38th International Conference on Software Engineering Companion - ICSE ’16*, 2016, pp. 744–746.

[9] Y. Ishii *et al.*, “Understanding the security management of global third-party Android marketplaces,” in *Proceedings of the 2nd {ACM} {SIGSOFT} International Workshop on App Market Analytics - {WAMA} 2017*, 2017.

[10] J. Abras, Chadia and Maloney-Krichmar, Diane and Preece, “User-centered design,” *User-Centered Des.*, pp. 445–456, 2004.

[11] ISO, “ISO 9241-11:2018 (en) - Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts,” 2018.

[12] britannica, “Likert scale.” [Online]. Available: https://www.britannica.com/topic/Likert-Scale. [Accessed: 08-May-2019].

Lisb@20 20 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional