

Detecção de Violações de Transparência a Partir de Feedbacks de Usuários em Repositórios de Aplicativos

Helbert Nunes¹, Lesandro Ponciano¹

¹Instituto de Informática – Pontifícia Universidade de Minas Gerais (PUC MINAS)
Belo Horizonte – MG – Brasil

helbertnunes97@gmail.com, lesandrop@pucminas.br

***Resumo.** Com a popularização do uso e desenvolvimento de aplicativos móveis, as lojas de aplicativos fornecem a possibilidade dos usuários se comunicarem com os desenvolvedores, mesmo que indiretamente, através de comentários. Neste contexto, o seguinte trabalho se propõe a investigar relatos de usuários acerca de violações de transparência dentro de aplicativos. A transparência é uma característica presente em um software, que, quando quebrada pode vir a gerar desconforto ou dúvida em um usuário. Através de uma análise feita sobre os feedbacks deixados pelos usuários nas lojas que disponibilizam os aplicativos. Busca-se identificar violações de transparência e associá-las à quebra de comunicabilidade. O estudo baseia-se em analisar uma base de comentários coletados na Google Play Store através do Método de Avaliação de Comunicabilidade e análise de polaridade, a fim de encontrar as violações de transparência nos comentários. Através da análise de polaridade e associação com a quebra de comunicabilidade, foi possível encontrar as violações de transparência nos comentários e perceber o impacto que causam no usuário final do aplicativo. Através dos resultados obtidos, é possível perceber que a transparência dos aplicativos deve ser analisada com mais calma pelos desenvolvedores, pois a falta de transparência pode levar os usuários a não mais utilizar os aplicativos.*

1. Introdução

A transparência é uma característica relevante nos mais diversos contextos, isso não é diferente quando se trata de software [Leite e Capelli 2010]. Usuários estão cada vez mais preocupados com que fim terão suas informações disponibilizadas em aplicativos e sistemas. Atualmente, é comum os usuários terem dados pessoais salvos em seus dispositivos. Qualquer aplicativo que tenha a necessidade de pedir permissão sobre os dados salvos, deve ser transparente quanto a essa necessidade. Por exemplo, uma calculadora, não têm motivo aparente para ter acesso à galeria de fotos armazenadas no celular, pois o funcionamento dela não está associada a fotos. Dessa forma, a solicitação desse tipo acesso, pode gerar desconfiança por parte do usuário em relação ao funcionamento do aplicativo. Esse fator pode gerar desconforto, ou até a não utilização do software por parte dos usuários.

Enquanto há diversos estudos sobre transparência de software [Leite e Capelli, 2010], pouco se sabe sobre transparência na perspectiva do usuário. Ou seja, como o usuário ao usar software experimentam e reportam problemas de transparência que experimentam. Nesse contexto, o problema a ser tratado neste trabalho é a **falta de informações sobre violações de transparência a partir de relatos dos usuários**. Relatos de usuários estão presentes em diversos ambientes como, comentários em aplicativos disponibilizados em lojas como Google Play Store, comentários em redes sociais e fóruns de pergunta e resposta. No contexto deste trabalho, ênfase é dada aos comentários de usuários em lojas de aplicativos.

Este trabalho tem como objetivo geral **investigar a importância da transparência no contexto de software a partir de relatos feitos por usuários em lojas de aplicativos**. Para atingir o objetivo geral, são propostos esses três objetivos específicos: i) obter relatos dos usuários de aplicativos relevantes à análise da transparência em aplicativos; ii) propor estratégias para identificar onde ocorrem violações na transparência em um software; iii) entender como a violação da transparência pode afetar na aceitação e uso do software pelos usuários.

O restante do trabalho está estruturado em 6 seções. A seção 2 apresenta o referencial teórico, que descreve os conceitos relevantes para o estudo. A seção 3 mostra os trabalhos relacionados que foram levados em consideração durante o estudo. A seção 4 descreve a metodologia utilizada no desenvolvimento do trabalho. A seção 5 descreve os resultados obtidos através da metodologia proposta e a seção 6 mostra as conclusões.

2. Referencial Teórico

Nesta seção, estão descritos os principais conceitos relacionados a transparência de software, e que são relevantes para uma melhor compreensão do trabalho. Estes conceitos foram divididos em três tópicos, que são: Transparência e Engenharia semiótica.

2.1. Transparência

Hosseini et al (2017) caracterizam a transparência como um requisito chave nos negócios modernos, auxiliando no processo de tomada de decisão dos *stakeholders* e aumentando a confiabilidade da informação. Porém, é necessária cautela para que a transparência não traga efeitos negativos. Houssini et al (2017) mostram através dos modelos de referência (definidos em seu trabalho), como mensurar e gerenciar a transparência como um requisito que gera informação relevante para os *stakeholders*.

De acordo com Leite e Capelli (2010), a transparência de software é um conceito novo e importante no contexto de desenvolvimento de software, e poucos trabalhos lidam com este conceito. Ainda no trabalho de Leite e Capelli (2010), eles apontam que se o software for implementado seguindo um modelo transparente, o código será transparente. Com esta afirmação, o desafio de se modelar um software transparente fica sob responsabilidade da modelagem de requisitos. A transparência de um software deve

ser tratada como requisito não funcional do sistema, visto que sendo transparente ou não, a funcionalidade do sistema não será afetada.

2.2. Engenharia semiótica

Barbosa e Silva (2010), apontam que a engenharia semiótica é uma teoria centrada na comunicação, com o foco na comunicação entre designers, usuários e sistemas. Na engenharia semiótica, as aplicações computacionais são caracterizadas como artefatos de metacomunicação, os artefatos são uma mensagem do designer para o usuário, sobre como o sistema pode e deve ser usado, e que efeito é gerado com sua utilização.

Os designers de Interação Humano-Computador (IHC), buscam a melhor forma de transmitir ao usuário sua visão sobre o sistema em uma ou mais mensagens. Como mostram Barbosa e Silva (2010), essa mensagem é codificada em palavras, gráficos, comportamento, ajuda on-line e explicações. Como o designer não estará ao lado do usuário durante a utilização esta mensagem é unilateral, e precisa ser bem clara, através da mensagem o designer “fala” pelo sistema. A mensagem codificada pelo designer, é responsável pela interação do usuário com o sistema, quando a mensagem é bem codificada o usuário consegue ter a percepção que era desejada pelo designer.

Para se encontrar as violações de transparência e falhas de comunicabilidade foram usadas as etiquetas do Método de Avaliação de Comunicabilidade (MAC) [Prates et al 2000], essas etiquetas são usadas para encontrar falhas na comunicação entre o sistema e o usuário. Como esse trabalho aborda violações de transparência no software, o tema está diretamente ligado à falha de comunicabilidade do sistema com o usuário, por isso a definição da engenharia semiótica se faz necessária no contexto abordado.

3. Trabalhos Relacionados

Nesta seção estão apresentados os trabalhos relacionados a transparência de software. Os resultados encontrados nos trabalhos a seguir apresentam diversas informações que nortearam a proposta de pesquisa deste artigo.

Leite e Capelli (2010), descrevem como a transparência pode vir a ser um requisito relevante dentro da engenharia de software. Para exemplificar, foi feito um estudo de caso usando o sistema “transparência olímpica”, (sistema criado com o intuito de “prestar contas”, à população acerca dos gastos das olimpíadas de 2016, realizadas na cidade do Rio de Janeiro). Foi feito um filtro em blogs e no *twitter* com palavras-chave para recolher o feedback dos usuários do sistema. Para ilustrar e definir o que seria a transparência no software estudado foram feitos três *Softgoal Interdependence Graph* (SIGs). Após a análise os autores apontaram os quatro principais problemas, que viram na aplicação da transparência no software, que são: A confiança (os autores apontam a dificuldade de confiar e acompanhar que os processos de transparência definidos serão cumpridos), o custo (aumentar os requisitos sem modificar os custos é um desafio adicional aos engenheiros), a performance e os usuários.

Hosseini et al. (2016), apontam como a transparência pode ser classificada como um requisito não funcional em um sistema. Existem casos onde a transparência é um requisito crítico e que precisa ser modelado com cuidado. Os autores propõem um *framework* baseado em três abordagens: *crowdsourcing*, *feedback* estruturado e

adaptação social. E estas seriam usadas para a melhor elicitación e evolução dos requisitos de transparência. Como a informação tirada de uma única fonte pode ser vazada, utilizando de vários stakeholders, essa informação se torna mais segura e confiável, além de promover um maior engajamento dos *stakeholders* acerca dos requisitos da transparência, por isso foi escolhido o *crowdsourcing*. Segundo os autores, a utilização do *feedback* estruturado se mostrou interessante para o estudo do *feedback* adquirido devido ao uso do *crowdsourcing*. Visto que, o *feedback* estruturado facilita o processamento, agregação, análise e cálculo dos dados. A adaptação social facilita a evolução do software à partir das outras duas abordagens citadas. Os autores apontam que essas três abordagens se completam, e que o framework criado traz benefícios para modelagem e evolução dos requisitos de transparência.

Hosseini et al. (2017) propõem quatro modelos de referência (*framework* abstrato para auxiliar no melhor entendimento nos relacionamentos importantes entre entidades de um fenômeno, propriedade ou sistema) para apoiar os engenheiros de requisitos a avaliar e modelar os requisitos de transparência de seus sistemas, que são: *Transparency Actors Wheel* (Identificação dos atores relevantes na troca de informações), *Transparency Depth Pyramid* (Divisão dos requisitos de transparência em três categorias: dados, processos e políticas), *Transparency Achievement Spectrum* (Transparência útil) e *information quality in transparency* (Qualidade da informação na transparência). Foi realizado como prova de conceito, uma análise do *United Kingdom Freedom of Information Act* (Lei do parlamento do Reino Unido), mostrando a importância da análise da perspectiva de quem recebe a informação, para a mensuração da transparência. Eles apontam também a importância desse estudo para outras áreas que podem utilizar da mensuração da transparência.

Kaika, et al. (2018) focam em mostrar, que, caso o usuário não atinja a expectativa que foi criada pelo desenvolvedor, o software não atinge seu objetivo. Os autores propuseram um método de elicitación de requisitos para evitar a violação de expectativas dos usuários. Utilizando o KAOS (do inglês, *Knowledge Acquisition in automated specification*), que é um método de modelagem de objetivo. Os autores analisam o motivo de cada expectativa gerada pelo sistema não ter sido atingida. Como não é possível controlar as ações dos usuários, foi usada a transparência como método de influência indireta nos usuários. Utilizando desse método de influência o esperado é que os usuários atinjam as expectativas do software voluntariamente.

Fu et al. (2013) mostram a relevância dos comentários dos usuários de aplicativos dentro das lojas de aplicativos. Através da coleta de mais de 13 milhões de avaliações de usuários em mais de 170 mil aplicações, foi feita uma avaliação quantitativa dos comentários, onde eles associaram palavras chave como: incrível, adorei, ótimo, horrível e etc, a um peso. Com este peso foi medido o grau de satisfação dos usuários em relação ao aplicativo avaliado. Com a coleta e análise do feedback foi possível entender quais fatores tornam o aplicativo mais atrativo para os usuários de determinados aplicativos, como jogos pagos.

Chen et al. (2014) apontam a dificuldade de se obter informações relevantes através do feedback dos usuários, e fazer a separação do que é um feedback informativo de um não informativo. Em seu estudo, foi apontado que apenas 35,1% dos feedbacks são relevantes para os desenvolvedores, e podem ajudar diretamente em melhorias para os aplicativos. Além disso, em aplicativos onde o número de usuário é muito grande, a

quantidade excessiva de avaliações de usuários torna ainda mais difícil a recuperação de informação útil dos feedbacks.

4. Metodologia

O estudo apresentado neste trabalho é de caráter qualitativo. Através da coleta de comentários em lojas de aplicativos espera-se: compreender e identificar as violações de transparência de software na perspectiva dos usuários. Nesta seção estão detalhadas as etapas seguidas para que seja atingido o objetivo da pesquisa.

4.1 Coleta e Processamento de Dados

Para a análise, foram coletados 16.567 comentários de 30 aplicativos distintos, os comentários e scripts usados para coleta e tratamento estão disponíveis em um repositório público do *GitHub*. O critério de seleção utilizado para a escolha dos aplicativos foi a popularidade, sendo selecionados 5 aplicativos, por categoria, a partir da seção “Em alta” na Google Play Store. A seção “Em Alta”, é atualizada em tempo real de acordo com as estatísticas da loja, por isso os aplicativos sempre variam.

A seção “Em Alta”, é dividida em 6 categorias, que são: Top Apps Gratuitos, nesta categoria estão presentes os aplicativos gratuitos mais baixados; Top Apps Pagos, nesta categoria estão presentes os aplicativos pagos mais baixados; Apps Mais Rentáveis, nesta categoria estão presentes os aplicativos gratuitos ou pagos, que geram mais receita com vendas dentro do aplicativo; Top Jogos Gratuitos, nesta categoria estão presentes os jogos gratuitos mais baixados; Top Jogos Pagos, nesta categoria estão presentes os jogos pagos mais baixados; e Jogos Mais Rentáveis, categoria que contém os jogos que mais geram receita em vendas dentro do jogo. Através desta seleção foi possível a coleta de aplicativos completamente distintos para tornar a análise dos comentários mais confiável.

Foi escolhido coletar os comentários no idioma inglês para se ter uma quantidade maior de comentários coletados, e também para que a análise de sentimento fosse mais confiável dos comentários. Com o objetivo de encontrar as palavras mais utilizadas nos comentários e fazer uma comparação entre essas palavras, foi gerado um mapa de palavras para análise. Para se gerar o mapa de palavras foram removidas as palavras de parada (do inglês *stop words*), pois elas são irrelevantes para uma análise. Exemplos de palavras de parada são: e, a, as, de, com e etc. As palavras de parada foram removidas somente para a criação do mapa de palavras, nas demais análises elas continuam presentes.

4.2 Análise de Polaridade

Para se encontrar as quebras de transparência e falhas de comunicabilidade se fez necessária uma divisão de polaridade nos comentários. Para fazer essa divisão foi usado o algoritmo: *Valence Aware Dictionary and Sentiment Reasoner* (VADER) [Hutto e Gilbert 2014]. O algoritmo VADER foi escolhido por ser tratar de algoritmo voltado para análise de comentários em mídias sociais, sendo, portanto, adequado para o presente estudo. Foi utilizada a biblioteca *vaderSentiment*, disponível na linguagem *Python*, para a análise.

O algoritmo VADER recebe como entrada um texto e funciona analisando sentença por sentença do texto e retorna uma pontuação formada por três dimensões: negativa, positiva e neutra. Cada dimensão recebe um valor específico entre 0 e 1. Para cada entrada, a soma das pontuações das três dimensões sempre é 1. Por exemplo, a frase “O telefone é super legal” (Do inglês, “*The phone is super cool*”), após a análise realizada pelo algoritmo VADER, as pontuações retornadas para cada uma das dimensões são: Negativo 0.0, Positivo 0.674 e Neutro 0.326. A soma das três dimensões é 1.

No presente trabalho, os comentários coletados na Google Play Store são classificados a partir da pontuação gerada pelo algoritmo VADER. Cada comentário é analisado e é atribuído a cada um deles uma polaridade de acordo com a dimensão que possui maior pontuação. Por exemplo, a frase “*The phone is super cool*” seria classificada como positiva, pois após a análise do VADER a pontuação da dimensão positiva foi a de maior valor.

4.3 Análise de Transparência e Comunicabilidade

Para se encontrar as violações de transparência e falhas de comunicabilidade foram utilizadas as etiquetas MAC [Prates et al 2000], apresentadas na Tabela 1. Como o campo de comentários dos aplicativos nas lojas se trata de um ambiente informal, para cada etiqueta, foram geradas variações semânticas e sintáticas, apresentadas na Tabela 2, para que sejam encontradas suas ocorrências com mais precisão entre os comentários coletados.

Tabela 1. Etiquetas usadas no método de avaliação de comunicabilidade e suas descrições

Etiqueta	Descrição
<i>Where Is?</i> <i>What Now</i>	O usuário parece estar procurando por alguma função específica mas demonstra dificuldade em localizá-la. Então ele começa a procurar nos menus pela função sem executar nenhuma ação. Nesta categoria a etiqueta <i>What Now?</i> que se aplica a quando o usuário claramente procura por uma pista do que fazer em seguida na expectativa de atingir seu objetivo.
<i>What's this?</i> <i>Object or Action?</i>	O usuário parece estar explorando as possibilidades de interação para obter mais entendimento sobre alguma função específica. Nesta categoria também se inclui casos onde os usuários se confundem com os componentes sendo associados a objetos ao invés de ações, ou vice versa.
<i>Oops!</i> <i>I can't do it this way</i> <i>Where am I</i>	Esta categoria abrange os casos onde o usuário performa alguma ação para atingir um estado esperado mas a saída não corresponde à expectativa. O que leva o usuário a desfazer a

	ação, ou a seguir com uma série de ações equivocadas levando à etiqueta <i>I can't do it this way</i> . Esta categoria também inclui a etiqueta <i>Where am I?</i> onde o usuário realiza uma ação que deveria ser executada em outro contexto.
<i>Why doesn't it?</i> <i>What happened?</i>	Como a anterior, nesta categoria o usuário espera uma saída e não a recebe. No cenário seguinte ele insiste no mesmo caminho, acreditando que esta é a maneira correta de executar a função. O cenário alternativo <i>What Happened?</i> é quando o <i>feedback</i> recebido pelo sistema é diferente do esperado.
<i>Looks fine to me...</i>	O usuário atinge algum resultado que ele acredita ser o esperado. Às vezes o resultado atingido não é o correto e não foi compreendido pelo usuário.
<i>I can't do it</i>	O usuário não consegue atingir o objetivo proposto pela função, seja por motivo de não saber como, ou por não possuir recursos (tempo, paciência, vontade, etc.).
<i>Thanks, but no, thanks</i> <i>I can do otherwise</i>	O usuário ignora o caminho proposto pelo sistema e parte para uma solução alternativa para atingir seu objetivo. <i>I can do otherwise</i> se aplica a quando o objetivo é alcançado de outra maneira. <i>Thanks, but no, thanks</i> é uma recusa explícita do caminho apresentado pelo sistema.
<i>Help</i>	O usuário utiliza o sistema de ajuda do aplicativo

Tabela 2. Etiquetas usadas no método de avaliação de comunicabilidade e suas variações sintáticas e semânticas que foram acrescentadas na coleta de comentários

Etiqueta	Variações Semânticas/Sintáticas
<i>Where is it?</i>	<i>Where is it</i> <i>Where's</i> <i>Where is</i> <i>Wheris</i>
<i>What now?</i>	<i>What now</i> <i>Now What</i> <i>Wat now</i> <i>Now wat</i> <i>What to do</i> <i>What to do now</i> <i>What i do</i>

<i>What's this?</i>	<i>What is this Whats this Wat is this Something going on</i>
<i>Oops!</i>	<i>Ops Whops</i>
<i>I can't do it this way</i>	<i>I cant do it this way Cant do this way</i>
<i>Where am I?</i>	<i>Where am I Where i am Where iam Were iam Where is it Where it goes</i>
<i>Why doesn't it?</i>	<i>Why does it Why do this</i>
<i>What happened?</i>	<i>Wat happened Something happened Something happens</i>
<i>Looks fine to me</i>	<i>Looks fine Looks good</i>
<i>I give up</i>	<i>I give up I gave up</i>
<i>I can do otherwise</i>	<i>I made it in another way I can make in another way Oderwise works do otherwise</i>
<i>Thanks, but no, thanks</i>	<i>No thanks Thanks but no</i>
<i>Help</i>	<i>Help Help me Help out Need help</i>

5.2 Relação de Comentários Coletados por Polaridade

Na Tabela 3, é demonstrada a relação dos comentários coletados, divididos por polaridade após a análise utilizando o algoritmo VADER. Como se tratavam dos aplicativos mais populares da loja, nota-se uma quantidade baixa de comentários negativos em relação aos outros.

Tabela 3. Quantidade de comentários por polaridade

Polaridade	Quantidade
Neutro	10256
Positivo	5813
Negativo	498
Total	16567

A Figura 2 apresenta a quantidade de comentários encontrada para cada polaridade onde é possível notar a superioridade de comentários neutros encontrados na base de dados em relação aos comentários polarizados. É importante notar a pequena quantidade de comentários negativos em relação aos outros, como foram analisados os aplicativos da aba “Em alta” este resultado era esperado.

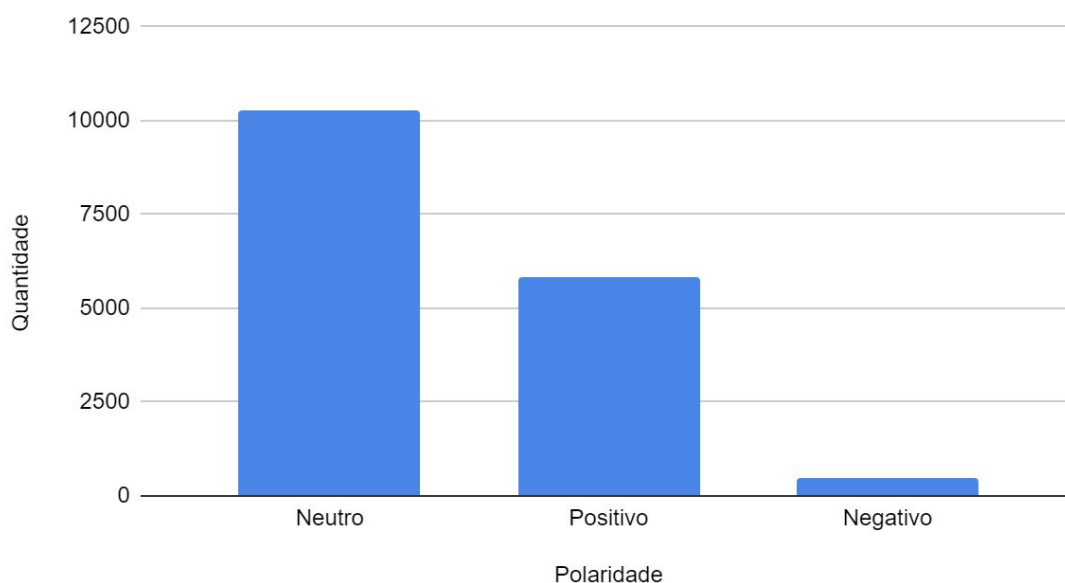


Figura 2. Representação gráfica da quantidade de comentários agrupados por polaridade após a análise do algoritmo VADER

5.3 Agrupamento por Etiqueta do MAC

Na Tabela 4, estão demonstrados a quantidade de comentários, divididos por polaridade, com sua ocorrência em cada etiqueta do MAC as variações semânticas e sintáticas já estão agrupadas a sua etiqueta correspondente. Conforme mostra a Tabela 2, a maioria dos comentários onde se encontram as etiquetas se caracteriza como neutro. O que era esperado, visto que as etiquetas apontam violações de comunicabilidade e não críticas ou reclamações.

Tabela 4. Número de ocorrências das etiquetas MAC nos comentários separados por sua polaridade.

Etiqueta	Neutros	Positivos	Negativos
Where is it?	16	0	0
What now?	31	0	0
What's this?	8	0	0
Oops!	40	1	0
I can't do it this way	0	0	0
Where am I?	2	0	0
Why doesn't it?	0	0	0
What happened?	3	0	0
Looks fine to me	0	0	0
I give up	1	0	0
I can do otherwise	0	0	0
Thanks, but no, thanks	7	0	0
Help	247	44	1

5.4 Comunicabilidade e Transparência

Após a separação dos comentários por etiqueta MAC, foi feita a separação dos que poderiam conter quebras de transparência associada às etiquetas. Na Tabela 5 estão contidos alguns exemplos dessa quebra de transparência. Pelos comentários foi possível notar uma relação da etiqueta “*Where's it?*” com a quebra de transparência, quando um software é atualizado, o aplicativo recebe a atualização e não é transparente com o usuário sobre o que foi alterado de uma versão para outra, gerando desconforto.

Tabela 5. Exemplos de comentários que possuem quebra de transparência relacionados à etiqueta MAC relativa junto com uma explicação da quebra de transparência.

Etiqueta MAC	Comentário	Quebra de Transparência
<i>Where is it?</i>	<i>I want 4 v 4 clash squad back. Where is it? I want it back. I want that back.</i>	O aplicativo não foi transparente na troca de versões o que gerou confusão no usuário.
<i>Where is it?</i>	<i>Updated. hmm Okay.... I apologize for being rude. but the app is not working correctly still. auto boost, ping boost, and FPS overlay in the advance section of settings don't do anything. They're not working. ALSO, where's the game tuner?? I can't tune my game at all.... there are no settings to increase performance of the game nor any things to change resolution. I actually want this app to work please.... and in the future I have some suggestions, email me if u want my ideas :- </i>	O usuário questiona onde está a funcionalidade descrita no aplicativo. O aplicativo não foi transparente quanto a essa funcionalidade.
<i>Where is it?</i>	<i>Where is the main apps that have to download. Only the license that I already buy. Are you bankrupt?</i>	O aplicativo não foi transparente com a necessidade de baixar outro aplicativo para usá-lo
<i>What now?</i>	<i>The game keeps crashing, it won't let me generate worlds, keeps deleting my old ones, and when I try to go to another's world it won't pop up. It even glitches sometimes after force stopping the app because of so much problems. I rated this a 5, two years ago. Clearly it has gone downfall from that. Please help me, I have no idea what to do. I really want to fix this because I know how well and enjoyable this app is. I love this app. It sucks if it doesn't work forever.</i>	Apesar de gostar do aplicativo o usuário não sabe o que fazer para manter seus dados salvos, quebra de transparência sobre haver ou não essa possibilidade.

	<i>Thank you!!!</i>	
<i>What's this?</i>	<i>The app is asking me the following literally (allow panda to access camera to take photos or videos)!!!!!!! What is this for real are you kidding me?! Why would you want to take photos or videos?!?!?!?</i>	O aplicativo solicitou acesso à câmera do usuário sem motivo relacionado às suas funcionalidades. Violação clara de transparência.

6. Conclusão

Este estudo se propôs a entender os seguintes pontos: A possibilidade de se encontrar violações de transparência a partir de relatos de usuários nas lojas de aplicativos, entender como as violações da transparência afetam a aceitação do aplicativo pelos usuários, assimilar a quebra de transparência com a violação de comunicabilidade através do estudo da engenharia semiótica. Foi proposta uma coleta de comentários de aplicativos distintos, para uma análise ampla e imparcial. Da coleta de comentários foi proposta uma análise com base no Método de Avaliação de Comunicabilidade para se relacionar a violação de transparência com a comunicabilidade.

Pelos resultados obtidos, percebe-se que é possível associar a violação de transparência com a comunicabilidade, mesmo que em poucos casos. Também se nota que as violações ocorrem, em sua maioria, em comentários neutros o que leva a pensar que os usuários deixam críticas construtivas nos comentários, mostrando aos desenvolvedores que se trata de uma fonte rica de informações para melhorar seu aplicativo. Na análise é possível notar que em alguns casos, a violação de transparência, pode gerar um grande desconforto aos usuário. Como por exemplo no caso de acesso desnecessário à câmera do celular Este desconforto deve ser percebido rapidamente pelos desenvolvedores e ser corrigido, a fim de evitar a perda do usuário e outros mais que notam a violação e não deixam comentários, apenas deixam de usar o aplicativo.

Em relação a trabalhos futuros seria interessante fazer uma análise em aplicativos com uma avaliação ruim para se ter uma percepção diferente quanto às avaliações e comentários deixados pelos usuários. Talvez com essa abordagem seja possível associar o motivo de abandono dos usuários à quebra de transparência e falha na comunicabilidade, o que deixaria ainda mais relevante o que foi levantado no presente trabalho. Os dados e códigos utilizados neste trabalho foram disponibilizados em <<https://github.com/HelbertNunes/TCC>> para serem utilizados como ponto de partida para eventuais evoluções deste estudo.

Referências

- BARBOSA, Silva. *Interação Humano-Computador*. Elsevier, 2010
- CHEN, N. Lin, J. Hoi, S. Xiao, X. Zhang, B (2014) “AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace.” *Proceedings of the*

- 36th International Conference on Software Engineering - ICSE 2014*, 2014, doi:10.1145/2568225.2568263.
- FU, B. Lin, J. Li, L. Faloutsos, C. Hong, J. Sadeh, N. (2013) “Why People Hate Your App.” *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 13*, 2013, doi:10.1145/2487575.2488202.
- GITHUB, disponível em <https://github.com/HelbertNunes/TCC>. Acesso em 11 Novembro 2019.
- GOOGLE, Play Store. disponível em <https://play.google.com/store>. Acesso em 11 Novembro 2019.
- HOSSEINI, M. Shahri, A. Phalp, K. Ali, R. (2016)“Foundations for Transparency Requirements Engineering.” *Requirements Engineering: Foundation for Software Quality Lecture Notes in Computer Science*, 2016, pp. 225–231., doi:10.1007/978-3-319-30282-9_15.
- HOSSEINI, M. Shahri, A. Phalp, K. Ali, R. (2016) “Crowdsourcing Transparency Requirements through Structured Feedback and Social Adaptation.” 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), 2016, doi:10.1109/rcis.2016.7549330.
- HOSSEINI, M. Shahri, A. Phalp, K. Ali, R. (2017) “Four Reference Models for Transparency Requirements in Information Systems.” *Requirements Engineering*, vol. 23, no. 2, 2017, pp. 251–275., doi:10.1007/s00766-017-0265-y.
- HUTTO, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.
- KAIYA, H. Yoshioka, N. Washizaki, H. Okubo, T. Hazeyama, A. Ogata, S. Tanaka, T. (2018) “Eliciting Requirements for Improving Users’ Behavior Using Transparency.” *Communications in Computer and Information Science Requirements Engineering for Internet of Things*, 2018, pp. 41–56., doi:10.1007/978-981-10-7796-8_4.
- LEITE, J. Cappelli, C. (2010) “Software Transparency”, Published online: 2010-04-27. DOI 10.1007/s12599-010-0102-z
- PHYTHON, disponível em <https://www.python.org/>. Acesso em 11 Novembro 2019.
- PRATES, R. Barbosa, S. Souza, C. (2000) “A Case Study For Evaluating Interface Design Through Communicability ” *DIS’00 Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods and techniques*, 2000, pp. 308-316.
- VADERSENTIMENT, disponível em <https://github.com/cjhutto/vaderSentiment>. Acesso em 11 Novembro 2019.