How to cite this paper:

Hani Barjes Al-bloush & Badariah Solemon. (2017). Users' intellectual property rights in crowdsourced software engineering tasks in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 369-376). Sintok: School of Computing.

USERS' INTELLECTUAL PROPERTY RIGHTS IN CROWDSOURCED SOFTWARE ENGINEERING TASKS

Hani Barjes Al-bloush[1], and Badariah Solemon[2]

[1]*Universiti Tenaga Nasional, Malaysia, hanibarjas@yahoo.com*
[2]*Universiti Tenaga Nasional, Malaysia, badariah@uniten.edu.my*

ABSTRACT. Online crowdsourced software engineering (CSE) platforms users comprise of task requesters and workers (or participants). It is critical for the users to acquire the Intellectual Property Rights (IPRs) for CSE tasks in order to maintain and encourage software crowdsourcing practices. This paper aims to examine the different CSE tasks and IPRs to help researchers understand better how these IPRs can be effectively applied to CSE platforms. A mapping table between CSE tasks and their IPRs is offered in this paper as the results of a comprehensive review of IPRs (copyrights, patents, trade secrets, trademarks, industrial design and database rights) and CSE tasks (in different phases of software engineering) from a number of relevant researchers. The results of the study are expected to be beneficial to crowdsourcing participants in understanding their rights and to existing crowdsourcing platforms owners as well as companies that are planning to launch, particularly, new CSE platforms to leverage the crowd power of the software engineering society.

Keywords: Crowdsourced Software Engineering (CSE), Intellectual Property Rights (IPRs), CSE tasks, participation

INTRODUCTION

The guarantee of the rights of users of online crowdsourcing platforms is critical and plays a vital role in increasing turnout of participants and in encouraging participation in online open call format, which is a good indicator for software engineering crowdsourcing platforms (Afuah & Tucci, 2012; Burger-Helmchen & Pénin, 2010; Majchrzak & Malhotra, 2013). Participation rights are considered in this study as Intellectual Property Rights (IPRs) of crowdsourcing users for accomplished tasks on online platforms (Mao, Capra, Harman, & Jia, 2015; Thuan, Antunes, & Johnstone, 2016; Vinaja, 2016). The importance of IPRs is highlighted in fields that witness many creations and innovations that require a guarantee of users' ownership rights for crowdsourcing tasks, including Crowdsourced Software Engineering (CSE) tasks (Ford, Richard, & Ciuchta, 2015; LaToza, Towne, Adriano, & Van Der Hoek, 2014; Mao et al., 2015; Stol & Fitzgerald, 2014). CSE has increasingly achieved a prominent place in the domain of software engineering, providing access to software engineering experts for various tasks through online platforms, such as *TopCoder (TC)* (Archak, 2010), *Amazon Mechanical Turk (AMT)* (Begel, DeLine, & Zimmermann, 2010) and *99Test* (Begel et al., 2010; Stol & Fitzgerald, 2014). Numerous concerns and risks surrounding crowdsourcing platforms regarding IPRs include awareness of intellectual property, intellectual property ownership (Ford et al., 2015), and knowledge remaining in the

crowd (Afuah & Tucci, 2012). The two main purposes of this study are: (1) to identify the tasks of the CSE and understand IPRs implications, and (2) to examine the different types of IPRs which may typically arise in CSE tasks. A review of the literature on CSE tasks and IPRs is also presented in this paper.

The remainder of this paper is structured as follows. Section 2 presents IPRs in software engineering, and Section 3 identifies and classifies the software engineering tasks in online crowdsourcing. Section 4 classifies the IPRs that may arise in CSE tasks, and lastly, some concluding remarks and recommendations for future work are made in Section 5.

IPRS IN SOFTWARE ENGINEERING

Creativity is highly significant in almost all aspects of software engineering; including the code, database, user interface, testing, evaluation, etc. (Liu, Feng, Li, Jing, & Yang, 2016; Mao et al., 2015; Sharp, 2016). Therefore, it is not surprising if the software protection under intellectual property is critical to the software industry as well (Suh & Oh, 2015). The term 'intellectual property' refers to multiple essential intangible Intellectual property term refers to multiple intangible nature rights of ownership in an essential, such as software functionalities rights of ownership, such as software functionalities (Menell, Lemley, & Merges, 2016; Mossoff, 2014). Each of the different IP rights are available to create value for software products. The law gives distinctive techniques to ensuring these rights of ownership in light of their types. The various types of IPR relevant to software include: *copyrights, patents, trade secrets, trademarks, industrial design* and *database rights* (Cohen, 2014; Saxena, Deodhar, & Ruohonen, 2017). Each type of IPR has its own legal protection. *Copyrights, patents, trade secrets, industrial design,* and *database rights* can be utilized to protect the software itself (Bently & Sherman, 2014; SHARMA, 2014; Suh & Oh, 2015). Even though *Trademarks* do not protect software, this IPR protects the words, names, symbols or designs (mark) used to recognize an item in the marketplace (Blackett, 2016; SHARMA, 2014; Stim, 2016). Table 1 provides a summary of the types of IPR which, might typically appear in, or be influenced by, tasks of software engineering.

Table 1. A Summary of IPRs in Software Engineering

Rights	Software Components	Protection	References
Copyright	Source codes; Binary codes; Supplementary documents	Protect software from unauthorized access "software developers and publishers generally employ copyrights to protect their final products from illegal copies by end users".	(Bouchoux, 2012; Boyle & Jenkins, 2014; Suh & Oh, 2015)
Patent	Ideas; Functionalities	The software has technological features which reflect innovation that requires patent protection. "Software developers exploit patents to shield key technological features of software from market competitors." The aim of patents is to protect the functional dimension of the software, by providing potential ownership of new and innovative technology implemented by programs.	(Bouchoux, 2012; Capatina, Bleoju, Matos, & Vairinhos, 2016; Mossoff, 2014; Suh & Oh, 2015)
Trade Secrets	Formula; Pattern; Program; Compilation; Method; Technique; Process; Device	A particular and identified data can be protected through contractual arrangements such as a Non-Disclosure Agreement (NDA) or Proprietary Information Agreement (PIA).	(Bouchoux, 2012; Boyle & Jenkins, 2014; Duston & Ross, 2013)
Trademarks	Words; Names; Symbols	Software brand (via either a logo or a word) may be protected by securing trademark registration in order to prevent competitors from using similar marks.	(Bouchoux, 2012; Stim, 2016; Zeidman, 2011)
Industrial	Graphical User Interfac-	"GUI designs may be protected under a	(Corell, 2015;

Design	es(GUI)	number of legal regimes: copyright, and design patents."	Hook, 2016; Stigler, 2014)
Database Rights	Outputs of the software process	The outputs of the software process "include any kind of data produced, such as databases of raw data, tables, graphics, pictures or whatever else" can be protected by SUI generis database protection.	(Bellantuono & Lara, 2015; Guibault & Wiebe, 2013)

CSE TASKS

Crowdsourcing is changing the manner in which software is being developed, through an online open call format for participation in any task of software engineering, including, but not limited to: *analysis, design, coding, testing, documentation and project management* (LaToza & van der Hoek, 2016; Leicht, Durward, Blohm, & Leimeister, 2015; Stol & Fitzgerald, 2014; Suganthy & Chithralekha, 2016). These tasks are usually conducted either by software institution employees or the workforce contracted by the institution. However, in crowdsourcing software, any tasks can be assigned to an unknown workforce online (Mao et al., 2015; Stol & Fitzgerald, 2014; Vinaja, 2016). For example, the tasks of crowdsourcing software analysis where the participations of an unknown workforce (crowds) are being used in the provision of innovative ideas for new product development, capturing the user requirements and for converting the user requirements into software features (Lim, Quercia, & Finkelstein, 2010; Naparat & Finnegan, 2013; Suganthy & Chithralekha, 2016). These participations are conducted via online platforms (Brabham, 2013; Schenk & Guittard, 2011) such as *Topcoder* and *Upwork*, where software crowdsourcing users can either post tasks (i.e., work requesters) or select those tasks (i.e., workers) that meet their abilities and interests (Kittur et al., 2013). These suggest crowdsourcing benefits of easy access to a wide range of workers, diverse solutions, lower labour rates and reduced time-to-market (Mao et al., 2015).

A list of the crowdsourcing tasks that might typically be requested in all phases of the software engineering process and their descriptions is provided in Table 2. The tasks in each software engineering phase include innovative idea generation, capturing user requirements and converting user requirements into software features (analysis); representing software requirements into UML diagram, algorithm writing, software design, receiving design, feedbacks and critiques, receiving software logo, and receiving user interface design proposals (design); writing codes for the design and reviewing the codes (implementation); functional test, performance test, usability test, localization test and GUI test (testing); documenting all software phases (documentation); and workflow design, workflow management (project management).

Table 2. A Classification of Crowdsourced Software Engineering Tasks

Software Engineering Phases	Crowdsourcing Tasks	Description	References
Analysis	Innovative idea generation	Participation of the crowd is beneficial for innovative concepts and ideas for novel product development, considering the user requirements and converting them to software features.	(Afridi, 2012; Bari, Johnston, Wu, & Tsai, 2016; Jeff, 2009; Naparat & Finnegan, 2013; Suganthy & Chithralekha, 2016)
	Capturing user requirements		
	Converting user requirements into software features		
Design	Representing software requirements into UML diagram	Crowd design is beneficial for formulating UML diagrams, algorithm design and component specifications for software requirements. The crowds	(Mao et al., 2015; Naparat & Finnegan, 2013; Suganthy & Chithralekha, 2016; Wu,
	Algorithm writing		

	Software design Receiving design feedbacks and critiques Receiving software logo Receiving user interface design proposals	offer worthy suggestions in the design of the user interface. Moreover, in particular scenarios, the criticism they offer is innovative and informative.	Tsai, & Li, 2013a, 2013b; Xu & Bailey, 2011)
Implementation	Writing codes for the design Reviewing the codes	Crowd coding is useful in developing a code for a particular software task or solving a given problem. Also, crowdsourcing allows multiple developers to review the code changes made by other developers, reducing the risk (bugs) and missed requirements.	(Mao et al., 2015; Suganthy & Chithralekha, 2016; Wu et al., 2013a, 2013b)
Testing	Functional test Performance test Usability test Localization test GUI test	Crowd testing recruiting both professional testers and end users to support on-demand testing services, such as GUI testing. This helps in testing quickly and in less time to the market.	(Leicht et al., 2015; Mao et al., 2015; Wu et al., 2013a, 2013b)
Documentation	Documenting all software phases	Crowd documenting helps in comprehensive understanding in all software phases, such as API, algorithm and code. Crowds can generate rich documentation through contributed questions and answers.	(Leicht et al., 2015; Suganthy & Chithralekha, 2016; Wu et al., 2013a, 2013b)
Project Management	Workflow design Workflow management	Crowdsourcing supporting workflows which include a project plan, software development and testing. Crowds support workflow design and execution of tasks involving human and machine activities	(Afridi, 2012; Brambilla, Ceri, Mauri, & Volonterio, 2015) (Kittur, Khamkar, André, & Kraut, 2012; Minder & Bernstein, 2012)

IPRS IN CSE TASKS

A particular focus on the context of crowdsourcing for profit recommends that tasks must be crowdsourced only if the IPRs can be clearly specified (Vinaja, 2016). Various IPRs could be applied to CSE tasks in order to guarantee adequate protection for all users of crowdsourcing platforms. Table 3 shows the mapping table of various potential IPRs (all types of IPRs typically appear in, or be influenced by, tasks of software engineering (see Table 1), which could be requested in all phases of software engineering as listed in Table 2). As can be seen in the mapping table, most CSE tasks are protected under the copyright IPR except several tasks including receiving software logo, codes review and all types of software testing tasks (functional test, performance test, usability test, localization test, and GUI test), which are protected under trademark or trade secret. Several tasks are protected under the copyrights are further protected with patent, trade secret and/or trademark including innovative idea generation, converting user requirements into software features, representing software requirements into UML diagram, algorithm writing, software designs, receiving user interface design proposals, and writing codes for the design CSE tasks. Like copyright, a database right does not have to be requested as the right exists as soon as the database exists in a recorded form. However, database rights are not covered in the existing research most probably because creation of databases is not one of commonly known CSE tasks.

Table 3. A Classification of IPRs in CSE Tasks

CSE Tasks	IPRs	References

denotes available (/)	Copyright	Patent	Trade secret	Trademark	Industrial Design	Database Rights	
Innovative idea generation	/	/	/				(Anderson, 2011; Poetz & Schreier, 2012; Stobbs, 2012)
Capturing user requirements	/						(Suh & Hwang, 2010; Suh & Oh, 2015)
Converting user requirements into software features	/	/					
Representing software requirements into UML diagram	/		/				(Di Penta, German, Guéhéneuc, & Antoniol, 2010)
Algorithm writing	/	/	/				(Suh & Hwang, 2010; Suh & Oh, 2015; Verma, 2012)
Software designs	/	/					(Stim, 2016)
Receiving software logo				/			(Stim, 2016)
Receiving design feedbacks and critiques	/						(Suh & Hwang, 2010; Suh & Oh, 2015)
Receiving user interface design proposals	/	/			/		(Suh & Hwang, 2010)
Writing codes for the design	/	/					
Reviewing the codes			/				(Freibrun, 2012)
Functional test			/				(Riungu, Taipale, & Smolander, 2010; Zogaj, Bretschneider, & Leimeister, 2014)
Performance test			/				
Usability test			/				
Localization test			/				
GUI testing			/				
Documenting all software phases	/						(Suh & Oh, 2015)
Workflow design	/						
Workflow management	/						

CONCLUSION AND FUTURE WORK

IPRs management and control for CSE tasks increases users' ability to participate in online crowdsourced software engineering platforms. This paper presents the different CSE tasks and IPRs that can be obtained from CSE tasks, which could help researchers understand better how these IPRs can be effectively applied to CSE platforms. The mapping table between CSE tasks and their IPRs is proposed and explained in this paper as the results of a comprehensive review of IPRs (copyrights, patents, trade secrets, trademarks, industrial design and database rights) and CSE tasks (in different phases of software engineering) from a number of relevant researchers. Based on the results presented in this paper, the authors further examine the legal document documents of the platform to understand how such IPRs have been implemented in almost 50 purely CSE platforms such as Topcoder and Innocentive as well as partially software engineering platforms that are not designed for software engineering in particular, but still can be used to support various activities of the software engineering, such as AMT, Upwork, and Freelancer. Our initial findings in that study show majority platforms acquired IPRs ownerships of contents submitted by participants. The authors also found out that there is no guideline to manage IPRs protection of platforms that supports CSE practices. Therefore, in the future, we plan to create a new guideline to manage and control IPRs that protect users' participation in online CSE platforms. In particular, instead of adopting agreements and instructions built on weak foundations in a number of CSE platforms, we plan

to formulate the IPRs applicable to the CSE tasks. With this new CSE guideline, IPRs for tasks can be managed and controlled in CSE platforms.

REFERENCES

Afridi, A. H. (2012). *Workflow engineering for crowdsourcing in project management towards a human-computers services.* Paper presented at the Engineering, Technology and Innovation (ICE), 2012 18th International ICE Conference on.

Afuah, A., & Tucci, C. L. (2012). Crowdsourcing as a solution to distant search. *Academy of Management Review, 37*(3), 355-375.

Anderson, J. J. (2011). Secret Inventions. *Berkeley Technology Law Journal, 26*(2), 917-978.

Archak, N. (2010). *Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com.* Paper presented at the Proceedings of the 19th international conference on World wide web.

Bari, E., Johnston, M., Wu, W., & Tsai, W.-T. (2016). *Software Crowdsourcing Practices and Research Directions.* Paper presented at the 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE).

Begel, A., DeLine, R., & Zimmermann, T. (2010). *Social media for software engineering.* Paper presented at the Proceedings of the FSE/SDP workshop on Future of software engineering research.

Bellantuono, G., & Lara, F. T. (2015). *Law, Development and Innovation* (Vol. 13): Springer.

Bently, L., & Sherman, B. (2014). *Intellectual property law*: Oxford University Press, USA.

Blackett, T. (2016). *Trademarks*: Springer.

Bouchoux, D. E. (2012). *Intellectual property: The law of trademarks, copyrights, patents, and trade secrets*: Cengage Learning.

Boyle, J., & Jenkins, J. (2014). *Intellectual Property: Law & the Information Society: Cases & Materials*: Center for the Study of the Public Domain.

Brabham, D. C. (2013). *Crowdsourcing*: Mit Press.

Brambilla, M., Ceri, S., Mauri, A., & Volonterio, R. (2015). *An Explorative Approach for Crowdsourcing Tasks Design.* Paper presented at the Proceedings of the 24th International Conference on World Wide Web.

Burger-Helmchen, T., & Pénin, J. (2010). *The limits of crowdsourcing inventive activities: What do transaction cost theory and the evolutionary theories of the firm teach us.* Paper presented at the Workshop on Open Source Innovation, Strasbourg, France.

Capatina, A., Bleoju, G., Matos, F., & Vairinhos, V. (2016). Leveraging intellectual capital through Lewin's force field analysis (the case of software development companies). *Journal of Innovation & Knowledge*.

Cohen, J. E. (2014). What Kind of Property Is Intellectual Property. *Hous. L. Rev., 52*, 691.

Corell, J. (2015). How liability for intellectual property defects in computer software transactions affect an efficient allocation of resources. A journey in the protection, transaction, infringement and uncertainty of copyright protected software.

Di Penta, M., German, D. M., Guéhéneuc, Y.-G., & Antoniol, G. (2010). *An exploratory study of the evolution of software licensing.* Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1.

Duston, T., & Ross, T. (2013). Intellectual property protection for trade secrets and know-how. *Intellectual Property Organization*.

Ford, R. C., Richard, B., & Ciuchta, M. P. (2015). Crowdsourcing: A new way of employing non-employees? *Business Horizons, 58*(4), 377-388.

Freibrun, E. (2012). Intellectual Property Rights in Software: What They Are and How the Law Protects Them.

Guibault, L., & Wiebe, A. (2013). Safe to be open. Study on the protection of research data and recommendation for access and usage.

Hook, S. A. (2016). Protecting Content Online: The Interface of Copyright and Design for Websites, Apps, and GUIs.

Jeff, H. (2009). Crowdsourcing: why the power of the crowd is driving the future of business. *Crown Business*.

Kittur, A., Khamkar, S., André, P., & Kraut, R. (2012). *CrowdWeaver: visually managing complex crowd work.* Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.

Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., . . . Horton, J. (2013). *The future of crowd work.* Paper presented at the Proceedings of the 2013 conference on Computer supported cooperative work.

LaToza, T. D., Towne, W. B., Adriano, C. M., & Van Der Hoek, A. (2014). *Microtask programming: Building software with a crowd.* Paper presented at the Proceedings of the 27th annual ACM symposium on User interface software and technology.

LaToza, T. D., & van der Hoek, A. (2016). Crowdsourcing in Software Engineering: Models, Opportunities, and Challenges. *IEEE Software*, *33*(1).

Leicht, N., Durward, D., Blohm, I., & Leimeister, J. M. (2015). Crowdsourcing in Software Development: A State-of-the-Art Analysis.

Lim, S. L., Quercia, D., & Finkelstein, A. (2010). *StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis.* Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2.

Liu, L., Feng, L., Li, Y., Jing, D., & Yang, H. (2016). Flourishing creativity in software development via Internetware paradigm. *Science China Information Sciences*, *59*(8), 080103.

Majchrzak, A., & Malhotra, A. (2013). Towards an information systems perspective and research agenda on crowdsourcing for innovation. *The Journal of Strategic Information Systems*, *22*(4), 257-268.

Mao, K., Capra, L., Harman, M., & Jia, Y. (2015). A Survey of the Use of Crowdsourcing in Software Engineering. *RN*, *15*, 01.

Menell, P. S., Lemley, M. A., & Merges, R. P. (2016). Intellectual Property in the New Technological Age: 2016-Chapters 1 and 2. *Intellectual Property in the New Technological Age*.

Minder, P., & Bernstein, A. (2012). *How to translate a book within an hour: towards general purpose programmable human computers with crowdlang.* Paper presented at the Proceedings of the 4th Annual ACM Web Science Conference.

Mossoff, A. (2014). A Brief History of Software Patents (and Why They're Valid). *Arizona Law Review Syllabus, December*, 14-41.

Naparat, D., & Finnegan, P. (2013). Crowdsourcing Software Requirements and Development: A Mechanism-based Exploration of 'Opensourcing'.

Poetz, M. K., & Schreier, M. (2012). The value of crowdsourcing: can users really compete with professionals in generating new product ideas? *Journal of Product Innovation Management*, *29*(2), 245-256.

Riungu, L. M., Taipale, O., & Smolander, K. (2010). *Software testing as an online service: Observations from practice.* Paper presented at the Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on.

Saxena, K. B. C., Deodhar, S. J., & Ruohonen, M. (2017). Intellectual Property Management in Software Business *Business Model Innovation in Software Product Industry* (pp. 109-127): Springer.

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics & Management*(1), 93-107.

SHARMA, D. K. (2014). INTELLECTUAL PROPERTY AND THE NEED TO PROTECT IT. *Indian J. Sci. Res*, *9*(1), 084-087.

Sharp, H. (2016). *Creativity and Collaboration in Software Design and Development.* Paper presented at the Proceedings of the European Conference on Cognitive Ergonomics.

Stigler, R. (2014). Ooey GUI: The Messy Protection of Graphical User Interfaces. *Nw. J. Tech. & Intell. Prop.*, *12*, i.

Stim, R. (2016). *Patent, copyright & trademark*: Nolo.

Stobbs, G. A. (2012). *Software patents*: Aspen Publishers Online.

Stol, K.-J., & Fitzgerald, B. (2014). *Two's company, three's a crowd: a case study of crowdsourcing software development.* Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Suganthy, A., & Chithralekha, T. (2016). *Application of crowdsourcing in software development.* Paper presented at the Recent Trends in Information Technology (ICRTIT), 2016 International Conference on.

Suh, D., & Hwang, J. (2010). An analysis of the effect of software intellectual property rights on the performance of software firms in South Korea. *Technovation, 30*(5), 376-385.

Suh, D., & Oh, D.-h. (2015). The role of software intellectual property rights in strengthening industry performance: Evidence from South Korea. *Technological Forecasting and Social Change, 92*, 140-154.

Thuan, N. H., Antunes, P., & Johnstone, D. (2016). Factors influencing the decision to crowdsource: A systematic literature review. *Information Systems Frontiers, 18*(1), 47-68.

Verma, S. (2012). IP Protection of Software and Software Contracts in India: A Legal Quagmire! *Journal of Intellectual Property Rights, 17*, 284-295.

Vinaja, R. (2016). Software Sourcing in the Age of Open: Leveraging the Unknown Workforce by Ågerfalk, Pär J., Fitzgerald, Brian, and Stol, Klaas-Jan: (2015). Heidelberg, Germany: Springer International Publishing. 77 pp. $59.99. ISBN: 978-3-319-17265-1. *Journal of Global Information Technology Management, 19*(1), 73-74.

Wu, W., Tsai, W.-T., & Li, W. (2013a). Creative software crowdsourcing: from components and algorithm development to project concept formations. *International Journal of Creative Computing, 1*(1), 57-91.

Wu, W., Tsai, W.-T., & Li, W. (2013b). An evaluation framework for software crowdsourcing. *Frontiers of Computer Science, 7*(5), 694-709.

Xu, A., & Bailey, B. P. (2011). *A crowdsourcing model for receiving design critique.* Paper presented at the CHI'11 Extended Abstracts on Human Factors in Computing Systems.

Zeidman, B. (2011). *The Software IP Detective's Handbook: Measurement, Comparison, and Infringement Detection*: Prentice Hall Professional.

Zogaj, S., Bretschneider, U., & Leimeister, J. M. (2014). Managing crowdsourced software testing: a case study based insight on the challenges of a crowdsourcing intermediary. *Journal of Business Economics, 84*(3), 375-405.