e-mentor

DWUMIESIĘCZNIK SZKOŁY GŁÓWNEJ HANDLOWEJ W WARSZAWIE

WSPÓŁWYDAWCA: FUNDACJA PROMOCJI I AKREDYTACJ KIERUNKÓW EKONOMICZNYCH

2021, nr 5 (92)

Pałys, M. (2021). The role of open source software in the process of implementing social innovation in SMEs. *e-mentor*, *5*(92), 45–52. https://doi.org/10.15219/em92.1542

Marcin Pałys

The role of open source software in the process of implementing social innovation in SMEs

Abstract

Nowadays, small and medium-sized enterprises (further SME) look for ways to overcome pressing social problems. An open-software-based application can be considered a powerful tool for this task, because of functionalities such as data management, BI operation, and efficient channels of communication. A review of literature shows that there is a shortage of studies concerning connections between usage of Free/Libre or Open Source Software (further FLOSS) and implementing Social Innovation (SI). The aim of this research is to investigate how the use of FLOSS relates to the ability of SMEs to implement SI. The author aims to underline key areas in which FLOSS supported implementation of SI. A set of recommendations for future initiatives is created based on the experience of successful implementation. Based on a review of literature, the author created a set of hypotheses which are validated by the in-depth surveys with three companies in the SME sector. The article structure is as follows: first the author presents literature findings concerning the subject, and next the areas of the model and survey answers. Finally, the author performs a critical review of the model, confirming that open software can stimulate the process of developing social innovation in SMEs, especially in cost reduction, flexibility, and community support.

Keywords: social innovation in SMEs, FLOSS, social application, software-based solutions, innovation

Introduction

Nowadays, *social innovation* is a term gaining more and more recognition, especially in the SME sector, which often operates more locally Business proprietors have a personal connection with their clients, employees, and social surroundings and, for this reason, pressing problems such as wastage of food, exclusion or discrimination are more visible to them (Begonia et al., 2016). Fulfilling these issues can present a great value (Prahalad et al., 2009). Moreover, managerial staff can presumably become the benefactor of the innovation they introduced (Graddy-Reed & Feldman, 2015). To meet recognized social needs, SMEs are turning to software solutions with functionalities that give them stable communication channels, adequate data management, or broad business intelligence operations (Bhatt et al., 2016). There is a shortage of research papers on this issue, especially in the view of SMEs. The aim of this paper is to fill that research gap and to investigate how the use of Free/Libre Open Software Solution (further FLOSS) relates to the ability to implement Social Innovation (SI) by Small and Medium Enterprises (further SMEs). The research focuses on an application developed by SMEs that meets social needs and which can also support itself financially. The author performed a set of in-depth queries to identify the key areas of benefits from FLOSS usage (cost reduction, safety, flexibility) and expand knowledge about their impact on the process of implementing SI. The research findings might play an interesting role in operation of SMEs, underlining the importance of a FLOSS-based code in the process of implementing SI. The research gathers benefits and drawbacks of FLOSS, from other areas, underlined in the work of other researchers, and tests the applicability and compliance in SMEs. What is more, broader knowledge of the FLOSS application in these circumstances (SMEs implementing SI), provides tools for simple problem-solving for enterprises, which may not be aware of existing possibilities.

Marcin Pałys, University of Economics in Katowice, Poland, http://orcid.org/0000-0002-9740-5068

Review of the literature

Social innovations

OECD research published in 2007 stated that the ability to introduce innovation would be the main source of competitive advantage for enterprises in the following decades (OECD, 2007). Nowadays it is hard not to agree with this research. Innovations play an important role in driving the economy and stimulating the growth of contemporary markets. Research shows that intensified economic activity also creates a potential for creating innovation thus far stimulating the growth of the global economy (Galindo-Martin & Méndez-Picazo, 2014). In theory, the development of enterprises through innovation should make societies wealthier and fulfil their needs. However, the nature of challenges that European countries are facing today is hardly conquered with innovations in their traditional form[1] (Howaldt & Schwarz, 2010). Solving problems such as corruption, food wastage, or overpopulation are rarely part of traditional business model enterprises (Bund et al., 2013). In addition, many government organizations lack necessary instruments and resources to overcome social challenges of this kind. The fact that these problems are difficult to tackle does not make them invisible for societies and organizations. It is possible to identify a shift in values towards a sustainable way of living, and thus far actions taken to resolve social problems are in great demand. A new type of innovation is needed, focused on dealing with challenges that are considered important but are not perceived in conceptualizing traditional innovation. The term used to describe this type of action is *social innovation* (Prahalad et al., 2009).

The concept of social innovation (SI) resolves around "creating new solutions (products, services, models, markets, processes etc.) that simultaneously meet a social need (more effectively than existing solutions) and lead to new or improved capabilities and relationships and/or better use of assets and resources" (Bund et al,, 2013). This definition highlights several key characteristics of SI. The most important one is basing the meaning of innovation on a social need that has to be fulfilled. Usually, they relate to economic development or health and education of excluded social groups (Datta & Ishaswini, 2011). Moreover, the aspect of the novelty of the solution is underlined, binding the term social innovation with a traditional view of innovations. The aspect of better usage of resources and assets leading to lack of waste and encouragement to seek potential in items seemingly useless is also considered meaningful. Another definition formulated by the European Commission underlines the fact that SI may be introduced by various actors. The European Commission (2013) defines social innovation as

> a new response to pressing social demands, which affect the process of social interactions. (…). They are innovations that are not only good for society but also enhance individuals' capacity to act. They rely on the inventiveness of citizens, civil society organizations, local communities, businesses, and public servants and services. (p. 7)

What this means is that social innovation is often introduced by smaller entities (SMEs), having better knowledge of social problems around them and more capabilities to stimulate other members of the group to act accordingly.

The term *SI* is gaining a lot of attention as a supposed tool to resolve crucial social problems and formulate a response to challenges that either way could not be addressed. SI are considered to stimulate connection between society actors, create new a solution, and therefore promote reforms (Hubert, 2010; Moulaert & Mehmood, 2011). However, according to the International Telecommunication Union (ITU), a key factor in solving a modern social problem is access to the means of transmitting information – proper infrastructure which allows connection of social actors and provides a platform for actions (International Telecommunication Union, 2013). Entities search for a free to use, easy to develop and cheaply maintained application that could transfer information necessary to implement SI. The idea of Free/Libre or Open Source Software (further FLOSS) and the ethos connected with it seems to adhere to those needs.

FLOSS

For software to provide value for an organization, it is necessary to provide an opportunity to achieve goals, reduce the operational cost or improve efficiency (Weerawarana & Weeratunga, 2004). Software consists of set of instructions for hardware on how to perform an envisaged task, to fulfill assumed functions. Nowadays, software is a part of almost every product and service, such as cars, electronics, consumables, etc. Moreover, software solutions are a basis for an organization's day-to-day operation, supporting processes of management, information exchange and decision-making (Lippoldt & Stryszowski, 2009).

FLOSS is an umbrella term for a set of software created and released in a particular way[2]. Unlike propri-

[1] Innovation is defined by Schumpter as launch of a new product or a new species of an already known product, application of new methods of production or sales of a product (not yet proven in the industry), opening of a new market (the market for which a branch of the industry was not yet represented), acquiring of new sources of supply of raw materials or semi-finished goods, or creating new industry structure such as the creation or destruction of a monopoly (Shumpeter, 1934).

[2] There is a difference between free and open software (Kelty, 2008), but for the sake of clarity for this article, the author decided to use a term comprising both of these types of software.

etary software, a developer using FLOSS benefits from four freedoms (Stallman, 2002). The freedom to:

- run the program for any purpose,
- study how the program works and change it – access the source code,
- redistribute copies,
- distribute copies of modified versions.

A FLOSS solution can be developed in the same way as standard software, but in most cases it is produced by teams of unstructured developers working together in organizationally and geographically distant environments (Lee & Cole, 2003). This community-based development process can consist of professionals paid to work on a particular FLOSS solution, freelancers – enthusiasts contributing to society, or students learning how to develop software and sharing their code (Lakhani & Wolf, 2003). In some cases, an organization can become a leader and set a direction for development of a particular solution (Fitzgerald, 2006), but many projects are conducted with no management structure. Expanding of FLOSS software led to creation of specialized platforms such as GitHub or SourceForge which are used for storing code, organizing projects, and sharing effects with the wider community (Henkel, 2006). Because of its nature, FLOSS can be defined as a "privately-produced public good" (O'Mahony, 2003).

Nowadays, FLOSS is a mainstream way of developing software rather than an academic curiosity. A number of active FLOSS projects are estimated in thousands and many broadly used programs benefits from it (Chrzanowski & Zawada, 2018). Among the most recognized is the Linux operating system Apache Web Server, many users applications (Mozilla Firefox, Open Office), web-building frameworks (WordPress), programming languages compliers, and integrated development environments (Atom, PyCharm) or enterprise systems (openCRX).

FLOSS has several benefits for an organization, which are distinguished from usage of closed-source software. Firstly, with open communities, software can be easier and cheaper to develop, because an organization can use premade pieces of code (named *libraries*) and create the program that is needed. The open sourced nature of FLOSS means that software can be reconstructed and adjusted to an organization's expectation. Therefore, developers working with FLOSS can react faster to changing environmental factors (such as market demand) and software is more capable of being regenerated (Yildrim & Ansal, 2011). This characteristic means that costs can be cut significantly and improves flexibility of operation. Secondly, developing with FLOSS is easier because vast communities often offer support and advice for developers. Many FLOSS projects include message boards and a mailing list, where authors can ask for a solution to a problem, debugging, or testing of their software (Fuggetta, 2003). Moreover, often, the FLOSS code is available for scrutiny to more developers, and thus is more secure. Security vulnerabilities are identified quicker and could be removed faster. Additionally, because of the open source nature of the solution, an organization can gain an insight into the program and adjust its security policies (Payne, 2002). FLOSS can also serve as a tool stimulating education and research. The cost of obtaining, installation and maintenance is often very low, and in some cases there is no cost, and because of that an organization can support software used solely for educating its members or benefactors or for conducting experiments and developing new software (Lakhani & Wolf 2003).

Using FLOSS has a variety of benefits, beginning with a possibility of cost reduction, more flexibility, and a higher level of security. Moreover, open sourced communities, emerging around FLOSS, grant developers a platform to learn and receive answers for their questions and problems. As stated previously, their purpose is not always to achieve financial success but often to fulfill a social need. To test this theory, the author conducted a series of interviews with three SMEs creating social innovation and asked them about the usage of FLOSS. The methodology and results are presented in the next paragraph.

Methodology

The aim of this research is to investigate how the use of FLOSS relates to the ability of SMEs to implement SI. The author aims to underline key areas in which FLOSS has supported implementation of SI. A set of recommendations for future initiatives is created based on the experience of successful implementation. An in-depth survey was picked as a validation method because the topic of connections between usage of FLOSS and there has been little research into the ability of SMEs to implement SI. Therefore, broad, qualitative research is needed. Organizations for the survey were chosen based on the impact of SI implementation and relevance to the research topic. All the research attendees created applications which were the driver of SI. Surveys were conducted via the online communication platform Google Meet. Each interview was filmed and saved on the university server. Meetings took place from 20th August to 7th September. Each meeting lasted around 45 minutes. After the interviews, data was analyzed regarding other literature findings by the author. After a review, conclusions were drawn for the paper.

Based on a literature review the following research questions were posed:

1. What was the driver of the social need that led to introduction of SI?
2. Did the use of open software lead to cost reduction of the applications?
3. Was FLOSS flexible?
4. Is the application more secure due to using open software?
5. Did the FLOSS community help in overcoming problems at the application development stage?

The questions presented above were asked in three SMEs responsible for introducing SI. Every company

delegated one key employee to answer the research questions. The companies were as follows:

- **Cozato** (platform A) – an application dedicated to the ecological trend "zero waste" and barter exchange among neighbors. The main goal of platform A is to promote reusage of goods and allow members of a social group to help each other and combat poverty. This is achieved by creating a platform on which people can submit offers of old, unused items which can be bartered or given to a person in need. Items can also be sold for a small amount of money. Unlike other solutions, which allow barter trading (Facebook groups, olx.pl), platform A is a platform designed only for this purpose and thus is much more developed, with functionalities supporting the operation (more flexible search engine or segregation and validation of offers). Software is a key factor in this SI because it is responsible for maintaining the platform, gathering, and structuring data, BI processes, and displaying the user interface. The software used in this project was the frameworks NextJs and Django. The software was created by people with professional experience in software development.
- **Zagadkowy.pl** (platform B) – an initiative which emerged during the COVID-19 pandemic. The company implementing the platform Zagadkowy.pl is the Silesian company Cudotwórnia, operating in the entertainment sector, focusing mainly on creating city games, questings[3] and escape rooms. Because of sanitary restrictions in Poland, it was impossible to conduct activities which were a common practice of integration among members of a social group or employers of a company. To support the needs of its clients, Cudotwórnia created platform B, a platform which consists of several online city games and escape rooms. The games use images, videos, sounds, and Google Maps to recreate situations and mechanics of a real-life event. Participants can take part in social activities in their homes, adhering to sanitary rules. Platform B addresses the need for social interaction, prevents alienation, and provides secure entertainment in times of a pandemic. It is especially important for people in quarantine. In this project, software is used to maintain the platform B website, maintain the payment system, analyze data, and store multimedia. Cudotwórnia uses the WordPress framework for its platform. Creation of the site was outsourced to a third party and then the company began to maintain and adjust platform B to their needs. The software was created by people without professional experience in software development.
- **Makeabetterplace.com** (platform C) – a platform inspired by the Georgian concept of social bars – places serving the purpose of stimulating creativity, innovation, and entrepreneurship among citizens. The main idea behind platform C is to connect people locally and support their bottom – up initiatives, by providing g them with a space for organizing meetings, testing business ideas, and finding specialists for their projects. The platform aims at strengthening relations between members of local groups and encouraging them to fulfill social and commercial needs. The main purpose of using software in this initiative is to store and analyze data, provide a way of communication, and unify access to information. The software used in this project is original software named GROT, with some addons based on FLOSS solutions. The software was created by people with professional experience in software development.

Findings

In the section below, the author presents the gathered answers to research questions and the conclusion drawn from them:

- **What was the driver of the social need that led to introduction of SI?** – in all the cases, the presented social need was observed by the entity and the will to solve social problems was the main drive behind introducing SI. Enterprises were looking for a way to capitalize on fulfilling a social need, creating a sustainable solution. The developer of platform C states:

  > We have the human-factor-focus principle, in which we design systems (existing architecture) with people's needs and not with developers playing Chinese whispers (requirements team, substantive design team, architecture team, programming team, test team) and finally the client asks why they built something other than that expected.

  Analyzing the answers to the question, it is worth underlining that SMEs, because of the smaller scale on which they operate, are a vital part of a society, well connected with other entities. The existence of social need is known to entrepreneurs; they are willing to find a solution. With appropriate technology, SMEs could implement a SI. Therefore, to stimulate the process of introduction of SI by SMEs, entrepreneurs need to be educated and presented with technologies able to uphold their ideas.

[3] Questing is a type of geocaching activity. Participants, in small groups search for a particular item or place in the city. Most often a GPS device is used to support the group.

- **Did the use of open software lead to cost reduction of the applications?** – owners of platform A state that thanks to FLOSS it was possible to significantly cut the cost of their platform, mainly by cutting the working hours of developers, because of usage of premade software libraries to fulfill their needs for functionalities. Moreover, platform A could cut the testing time, therefore lowering the cost even more. According to owners of platform B, cost reduction was also significant because without FLOSS not only the writing of the code, but also maintaining the code would have been outsourced. Also, the prices of third-party services concerning known frameworks are lower than the price of implementing closed software. The owner of platform B states:

  > Using closed software would generate enormous costs for our company. We checked it and we priced it, but in an era when everyone was tightening their belts, especially our clients, and especially us, it was not possible to build something from scratch. I think it would be reinventing the wheel, creating functionalities that already exist and are present on the market.

  Company C used their own developed technology GROT for platform C because in their view using FLOSS would increase the cost. Developers would have to learn a new coding language or use a tool in which they are not proficient. Because of their expertise in GROT, it was beneficial to use a closed-software solution. However, they decided to add FLOSS addons to speed up the development of chosen functionalities (such as reading QR codes or gaining access to fast-rotating government data). Moreover, the developers of platform C realize that due to the need of creation of an online shop or banking functionalities their GROT language would not succeed because of malfunction possibilities and the costly process of testing. Hypothesis two was confirmed.
  It is certain that FLOSS can reduce costs on two levels. According to the answers from platforms A and C, FLOSS enables the number of developers to be cut and testers' working hours to be cut using the premade code. What is more, FLOSS offers a solution for most common functionalities of an app, allowing developers to concentrate on more complicated issues during the development process. According to platform B, FLOSS usage leads to minimization of outsourcing and use of in-house resources, reducing costs. Creating many features of app B would not be possible or would have been outsourced without FLOSS. On the contrary, the usage of FLOSS could lead to increased costs if developers have to learn a new coding tool instead of using the known solution. Therefore, the cost benefits of implementing FLOSS may vary depending on the knowledge of the developers and their ability to learn new skills.

- **Was FLOSS flexible?** – platform A owners perceived their FLOSS solution as a flexible one but only if the developers had enough expertise to adjust the framework to their needs. In project A this could be accomplished but platform A states that when picking a framework, it is often the case of the amount of learning in relation to the generated value. Moreover, they point out that many functionalities are similar across platforms so they could use software libraries prepared by other FLOSS community members. Developers of platform A state:

  > The FLOSS we chose turned out to be flexible because we gained experience in it. The team had a lot of experience in React, but we chose Next for technological reasons. This led to some problems in the beginning, when we had to understand the software a bit, but in the end it turned out that choosing this FLOSS was a plus, but it definitely took some time.

  Developers of platform B underline that their framework consists of a vast number of addons ready for use without modification, which fulfilled their need for functionalities, and FLOSS made it easy to change the platform while it was already operational and adjust it to the needs of clients. The platform had basic functionalities which were already covered by the framework. In platform C, the developers perceived the FLOSS solution as flexible for the small task it was responsible for, but would not decide to transform the project entirely into FLOSS, because the GROT language was perceived as more suitable for their needs. Hypothesis three was confirmed.
  FLOSS flexibility is based mostly on the type of the framework of FLOSS and the expertise of the developers. Moreover, it is important to highlight two characteristics of a framework: the spectrum of functionalities and the ease of adjustability. The framework used by platform B is to be considered narrower in spite of functionalities, but even an unskilled developer can quickly adjust the platform to the needs of the client. On the other hand, frameworks used by platform A are to be considered broad in functionalities but proper usage and adjusting requires appropriate knowledge of the developer. Companies like C, already using a custom-made solution, may perceive FLOSS as inflexible because the smaller frameworks do not have the required functionalities, and broader frameworks require too much learning time.

- **Is the application more secure due to using open software?** – developers of platform A state that in their view FLOSS provides a high level of security because of constant testing and regular updates from the community. The work of many developers translates into a regularly updated base of exploits and malfunctions. Potential threats are noticed faster, and many entities are working on preventing them doing damage. The owners of platform B underline that FLOSS made their platform much more secure than they ever could with their lack of specialized knowledge. The developers of platform C state that because of using an unknown to vast community software their platform is more secure, because hostile entities do not have access to GROT specifications to prepare an attack, and it would cost a lot of resources to create malware which would be dangerous to their platform. On the other hand, the security of platform C depends only on its developers and their expertise. Hypothesis five was confirmed.
  The case of software security plays an important role in the development of software-based SI. Because of the lack of knowledge in this area, enterprises like to outsource the issue and mimic that act using FLOSS. Many entities work on the safety of the software and entrepreneurs are benefactors of the process without the need to pay for it. What is more, the safety of the solution is in all its users' interest, so the community contributes to the process willingly and eagerly to make their future projects more secure. On the other hand, a custom-made solution might not be in the scope of most common attacks thus far providing a higher level of security, but it requires a certain amount of knowledge in software security to make this kind of solution secure. Furthermore, a targeted attack, focused on breaching a particular platform, can still be dangerous for custom software.
- **Did the FLOSS community help in overcoming problems at the application development stage?** – during the process of creating platforms A and B, community support was beneficial for the project. Developers solved several problems with the support of a local coding group, internet forums or FLOSS documentation. Moreover, developers could discuss the functionalities of their platform with users of this technology, asking for advice. The developers state: "It's hard to program without Stack overflow[4] and, in addition to official documentation, there is often salvation in finding out that someone has had a problem like ours and it was solved." Support concerning premade addons was beneficial for them. Platform C creators could hardly consult other entities on their problems, and relied mainly on company developers. Hypothesis six was confirmed.
  Based on the answer to this question, it is possible to highlight the two types of community support an entrepreneur could benefit from. One is passive support acquired through documentation and internet activity of other developers (blogs, fora, threads on stack overflow). This provides a deeper insight into the functioning of the FLOSS and basic problem solution. For a more complicated issue, developers seek guidance in active support – consulting the community about a particular issue/functionality in their application. This kind of support demands a developer to be an active community member, participating in knowledge trading.

Discussion

The presented studies are viably important on the spectrum of other FLOSS oriented papers. There are no studies in literature of a similar approach (FLOSS-based Social Innovation in SMEs) therefore the author focused on findings concerning general usage of FLOSS and searched for common ground and new findings. According to Borzaga and Bodini (2014), successful implementation of software depends to a great extent on the understanding of the complexity of the problem being solved by it. The findings of the study correspond to that approach, claiming that the drive for introducing SI in researched SMEs was the social need observed in the company's immediate surroundings. SMEs are implementing social innovation because they see the social need in their surroundings – among their family, friends, employees, or other social groups – and understand it well. Software functionalities also provided companies with more possibilities to conduct their operations and fulfill social needs. In the area of cost and flexibility, it is a common theory that FLOSS allows for a significant cut in expenses because of usage of predeveloped frameworks/libraries which are free for exploitation (Ågerfalk & Fitzgerald, 2008; Kilamo et al., 2012; Reed et al., 2012). Moreover, flexibility of the solution does not have to be limited. Research responders agree with the theory that using FLOSS for standardized, small functionalities translates into time and cost reduction but, in some cases, developing a solution using self-written code can be more efficient and adhere to the user needs better. It depends on the level of professional expertise of the staff of the company. Based the example of makeabetterplace.com, it is possible to underline that if a company has a skilled developer focused on developing self-written technology, it is not efficient to use FLOSS because of the time and effort which must be put into learning a new technology. On the other hand, as the example of Zagadkowy.pl and Cozato proves, FLOSS can lower

[4] Stack overflow – a social networking site where programmers can ask questions about software development.

the cost of a project significantly without limiting its functionality. According to other research, FLOSS solutions are considered safer than closed-software solutions. Exploit and bugs are more likely to be noticed and are fixed faster. Moreover, some security measures are taken even without user knowledge (updates of WordPress framework, fixing known exploits) (Erturk, 2012; Thompson & Wagner 2017; Walden et al., 2009). Research responders expressed two different approaches. In the first two companies, the developers are benefactors of the work of other community members, using broadly known, regularly updated and tested FLOSS code to grant their platform security. This is a convenient solution for SMEs without a high level of programing knowledge. The second approach, mentioned by the third company, is that FLOSS is much more popular, better known, and easier to breach, so self-written or closed software is more secure. The FLOSS code is available for everyone, thus creating malware able to attack the application is simpler. The developers focus on a more expensive solution that can go unnoticed by potential hackers. Importantly, in this approach security depends solely on the expertise of the developers. In the area of community support, other research states that open access to code promotes a process of creating large communities around FLOSS, often in the form of fora, blogs, or code repositories. It creates a lot of source material for developers interested in using that solution, helpful in the process of implementing functionalities or debugging. The process helps to empower developers, and give them freedom of creation (Bhatt et al., 2016). All responders agree that this is a positive factor concerning FLOSS which stimulates the process of introducing SI. Moreover, according to study findings, there is a passive type of support, acquired by publicly available blogs, fora etc., and an active type of support, achieved by reaching out for the help of community members. Passive support is available freely, but active support often requires a contribution for the support group (being a part of a broader initiative, helping other developers etc.).

Conclusion

The findings of the research correspond to the existing theory concerning the relationship between social innovation and FLOSS usage. It was proven that FLOSS stimulates the process of implementing SI and should be considered an important variable in evaluation of its business potential. Moreover, studies confirm that challenges regarding implementation of SI in SMEs in other parts of the earth are like challenges for Polish enterprises[5]. However, research findings provide more insight into the nature of the process of implementing SI in SMEs with FLOSS. A study shows that the amount of developer skills in the company plays a role in determining the method of using the software. Regarding a characteristic of small and local entities, without professional know-how and a high budget, FLOSS could be a solution for implementing expected innovation. To stimulate the process, one should present the means and educate the employers of SMEs about possibilities regarding FLOSS thus far providing them with tools for their needs.

The presented research acts as a focus study and should be broadened to provide more universal findings. A study concerning quantitative research of SMEs is to be conducted to prove that conclusions drawn from three entities can be extended to cover a larger population. Moreover, FLOSS differs vastly in functionalities and capabilities, so research concerning the types of software could isolate factors and the strength of their influence on the process of implementation of SI by SMEs. The SME sector is also known for its diversity. FLOSS usage can vary from enterprise to enterprise, sector to sector. More in depth characterization of SMEs and a broader scale of the research could present interesting findings which did not emerge from the research because of its narrower scale.

References

Ågerfalk, P. J., & Fitzgerald, B. (2008). Outsourcing to an unknown workforce: exploring opensourcing as a global sourcing strategy. *MIS Quarterly*, *32*(2), 385–409. https://doi.org/10.2307/25148845

Begonja, M., Čićek, F., Balboni, B., & Gerbin, A. (2016). Innovation and business performance determinants of SMEs in the Adriatic region that introduced social innovation. *Economic Research-Ekonomska Istraživanja, 29*(1), 1136–1149. https://doi.org/10.1080/1331677X.2016.1213651

Bhatt, P., Ahmad, A. J., & Roomi, M. A. (2016). Social innovation with open source software: User engagement and development challenges in India. *Technovation*, *52–53*, 28–39. https://doi.org/10.1016/j.technovation.2016.01.004

Borzaga, C., & Bodini, R. (2014). What to make of social innovation? Towards a framework for policy development. *Social Policy and Society, 13*(3), 411–421. https://doi.org/10.1017/S1474746414000116

Bund, E., Hubrich, D., Schmitz, B., Mildenberger, G., & Krlev, G. (2013). *Blueprint of social innovation metrics. Contributions to an understanding of opportunities and challenges of social innovation measurement*. A deliverable of the project: The theoretical, empirical and policy foundations for building social innovation in Europe (TEPSIE), European Commission – 7th Framework Programme. European Commission. https://archiv.ub.uni-heidelberg.de/volltextserver/18700/1/D2.4_final.pdf

Chrzanowski, M., & Zawada, P. (2018). *Otwarte innowacje i ich wykorzystanie w przedsiębiorstwach typu start-up*. Oficyna Wydawnicza Politechniki Rzeszowskiej.

Datta, S., & Ishaswini. (2011). Pro-environmental concern influencing green buying: A study on Indian consumers. *International Journal of Business and Management, 6*(6), 124–133. https://doi.org/10.5539/ijbm.v6n6p124

[5] e. financial aspects, professional expertise, flexibility of the solution.

Erturk, E. (2012). A case study in open source software security and privacy: Android adware. In *World Congress on Internet Security (WorldCIS-2012)* (pp. 189–191). IEEE.

European Commission. (2013). *Guide to social innovation.* http://ec.europa.eu/regional_policy/sources/docgener/presenta/social_innovation/social_innovation_2013.pdf

Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly, 30*(3), 587–598. https://doi.org/10.2307/25148740

Fuggetta, A. (2003). Open source software – an evaluation. *Journal of Systems and Software, 66*(1), 77–90. https://doi.org/10.1016/S0164-1212(02)00065-1

Galindo-Martin, M. A., & Méndez-Picazo, M. T. (2014). Entrepreneurship, economic growth, and innovation: Are feedback effects at work? *Journal of Business Research*, *67*(5), 825–829. https://doi.org/10.1016/j.jbusres.2013.11.052

Graddy-Reed, A., & Feldman, M. P. (2015). Stepping up: an empirical analysis of the role of social innovation in response to an economic recession. *Cambridge Journal of Regions, Economy and Society, 8*(2), 293–312. https://doi.org/10.1093/cjres/rsv008

Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy, 35*(7), 953–969. https://doi.org/10.1016/j.respol.2006.04.010

Howaldt, J., & Schwarz, M. (2010). *Social innovation: Concepts, research fields and international trends*. https://www.asprea.org/imagenes/IMO%20Trendstudie_Howaldt_englisch_Final%20ds.pdf

Hubert, A. (2010, July). *Empowering people, driving change: social innovation in the European Union.* https://ec.europa.eu/migrant-integration/library-document/empowering-people-driving-change-social-innovation-european-union_en

International Telecommunication Union. (2013). *Measuring the information society.* https://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2013/MIS2013_without_Annex_4.pdf

Kelty, C. (2008). *Two bits: The cultural significance of free software.* Duke University Press. https://twobits.net/pub/Kelty-TwoBits.pdf

Kilamo, T., Hammouda, I., Mikkonen, T., & Aaltonen, T. (2012). From proprietary to open source – Growing an open source ecosystem. *Journal of Systema and Software*, *85*(7), 1467–1478. https://doi.org/10.1016/j.jss.2011.06.071

Lakhani, K., & Wolf, R. (2003). *Why hackers do what they do: Understanding motivation and effort in free/open source software projects.* http://dx.doi.org/10.2139/ssrn.443040

Lee, G., & Cole, R. (2003). From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science, 14*(6), 615–758. https://doi.org/10.1287/orsc.14.6.633.24866

Lippoldt, D., & Stryszowski, P. (2009). *Innovation in the Software Sector.* OECD. http://ict-industry-reports.com.au/wp-content/uploads/sites/4/2010/10/2009-Innovation-in-the-Software-Sector-OECD-Nov-2009.pdf

Moulaert, F., & Mehmood, A. (2011). Spaces of social innovation. In A. Pike, A. Rodriguez-Pose, & J. Tomaney (Eds.), *Handbook of local and regional development* (pp. 212–225). Newcastle University UK.

OECD. (2007). *Innovation and growth. Rationale for an innovation strategy.* https://www.oecd.org/sti/39374789.pdf

O'Mahony, M. (2003). The role of information and communications technology inputs. *National Institute Economic Review, 184*(1), 58–59. https://doi.org/10.1177/0027950103184001005

Payne, M. (2002). The politics of systems theory within social work. *Journal of Social Work, 2*(3), 269–292. https://doi.org/10.1177/146801730200200302

Prahalad, C., Nidumolu, R., & Rangaswami, M. (2013). Why sustainability is now the key driver of innovation. *Engineering Management Review, 41*(2), 30–37. IEEE. https://doi.org/10.1109/EMR.2013.6601104

Reed, S., Storrud-Barnes, L., & Jessup, L. (2012). How open innovation affects the drivers of competitive advantage: Trading the benefits of IP creation and ownership for free invention. *Management Decision, 50*(1), 58–73. https://doi.org/10.1108/00251741211194877

Schumpeter, J. (1934). The theory of economic development. An inquiry into profits, capital, credit, interest and the business cycle. *Harvard Economic Studies, 46*.

Stallman, R. M. (2002). Free software definition. In *Free software, free society: Selected essays* of *Richard M. Stallman* (pp. 43–46). GNU Press.

Thompson, C., & Wagner, D. (2017). *A large-scale study of modern code review and security in open source projects.* Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering (pp. 83–92). https://doi.org/10.1145/3127005.3127014

Walden, J., Doyle, M., Welch, G. A., & Whelan, M. (2009). *Security of open source web applications*. Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 545–553). IEEE.

Weerawarana, S., & Weeratunga, J. (2004, January). *Open source in developing countries*. Sida. https://cdn.sida.se/publications/files/sida3460en-open-source-in-developing-countries.pdf

Yildirim, N., & Ansal, H. (2011). Foresighting FLOSS (free/libre/open source software) from a developing country perspective: The case of Turkey. *Technovation*, *31*(12), 666–678. https://doi.org/10.1016/j.technovation.2011.07.004

Marcin Pałys is an employee of the University of Economics in Katowice and entrepreneur. As a researcher he combines interests in economy and entrepreneurship with a fascination with the possibilities offered by modern technologies. The main areas of his research are Business Intelligence and Big Data, as well as the possibility of using these technologies in the SME sector.