

# Crowdsourcing, Open-sourcing, Outsourcing and Insourcing Software Development: A Comparative Analysis

Nitin Naik

Defence School of Communications and Information Systems  
Ministry of Defence, UK  
Email: nitin.naik100@mod.uk

**Abstract**—The software development industry invariably embraces different models and techniques to develop the best user-oriented and cost-effective solutions. The selection of an appropriate software-sourcing model is one of the key factors that influences the entire development process. The various software-sourcing models Crowdsourcing, Open-sourcing, Outsourcing and Insourcing have been adopted to achieve the best software products. Every software-sourcing model possesses several strengths and may be effective in certain types of projects; however, its limitations may restrain the use of it in some other projects. It is very essential for an organisation to investigate the pros and cons of a potential sourcing-model before going to adopt it for their project. This paper presents a comparative analysis of Crowdsourcing, Open-sourcing, Outsourcing and Insourcing models in order to find out the best fit environment and suitability of each model.

**Keywords**—*Software-Sourcing Model, Crowdsourcing, Open-sourcing, Outsourcing, Insourcing, Software Development*

## I. INTRODUCTION

Enterprises always look for a solution to the software-sourcing dilemma whether to build, buy or hire [1], [2]. While each software-sourcing model facilitates technology, expertise and resources to the enterprise to achieve their goals, they are somewhat different regarding management of commercial risks, financial commitments, intellectual property and human resources [3]. Companies adopt different software-sourcing models due to many reasons, ranging from cost-effectiveness, specialised task and expertise, IT infrastructure and time limit. However, the change in software-sourcing model largely results from a shift in the global business strategy and competitiveness. There are four software-sourcing models largely used in the software industry are: Crowdsourcing, Open-sourcing, Outsourcing and Insourcing.

One of the major differences between these software-sourcing models is the employment of an army of developers. Insourcing software development model uses in-house employees to carry out all the development tasks. Outsourcing software development model hires an external service provider (company or individual) to carry out all or part of the development tasks. Crowdsourcing software development model hires skilled workers from the crowd using an open-call to carry out all or part of the development tasks. Open-sourcing

software development model invites skilled volunteers from the open-source community to carry out all or part of the development tasks. Among these software-sourcing models, recently, crowdsourcing has been emerged as one of the most powerful business models involving a large number of people combining to form a crowd to perform tasks. The modern crowdsourcing term was reinvented by Jeff Howe and Mark Robinson in 2006 [4]. Crowdsourcing is an approach to outsourcing tasks to a large undefined crowd through an open-call [5]. Crowdsourcing software development emerged from this crowdsourcing concept and inherited most of the features from it. However, it adapted its nature according to the requirements of the software engineering techniques and technologies to perform various software engineering tasks such as requirement analysis, design, coding, and testing [6]. Every software-sourcing model possesses several strengths and may be effective in certain types of projects, however, its limitations may restrain the use of it in some other projects.

The selection of an appropriate software-sourcing model is very vital for the success of a software project and business. Therefore, this paper presents a comparative analysis of these four software sourcing models: Crowdsourcing, Open-sourcing, Outsourcing and Insourcing. Firstly, it performs a relative analysis of these four software-sourcing models based on some crucial criteria to demonstrate their strengths and limitations at a broader level. Secondly, it exhibits a comparative analysis of these four software-sourcing models based on several criteria to clarify their differences at a detailed level. Finally, it accomplishes a suitability analysis of these four software-sourcing models to bespeak their potential conditions of usage.

The remainder of this paper is organised as follows: Section II elucidates the theoretical background of four software-sourcing models: crowdsourcing, open-sourcing, outsourcing and insourcing; Section III presents a relative analysis of all software-sourcing models for showing a bigger picture; Section IV demonstrates a comparative analysis of all software-sourcing models for providing detailed information about them; Section V clarifies the possible area and suitable conditions of their usages; Section VI concludes the paper and suggests some future work.

## II. THEORETICAL BACKGROUND

This section presents the theoretical background of crowdsourcing, Open-sourcing, outsourcing and insourcing software development models. Before explaining software-sourcing models, it clarifies the meaning of crowdsourcing to provide the better understanding of crowdsourcing software development model.

### A. Crowdsourcing

Crowdsourcing is the act of a company or institution taking a project/job once performed by employees and outsourcing it to an undefined nexus of people in the form of an open call [4]. This project/job can be done by self-organizing communities of individuals as a collaborative job. However, most often, it is done by sole individuals. The key requirement is the use of an open-call format and the large network of potential workers [5]. Crowdsourcing challenges a large group of people with diverse backgrounds located remotely to create an entire solution or project [7]. Crowdsourcing can be used to achieve numerous goals and solutions. Jeff Howe and Mark Robinson coined the term crowdsourcing [4], and Jeff Howe used this term in his article in the Wired Magazine in 2006 [8]. Since then, researchers have been investigating the various models of crowdsourcing and their impacts and successes. However, crowdsourcing, an ancient term was first popularly used by the English Royal Navy to solve the problem of measuring longitude. They organized a competition with a price of 10000 pounds for the man who could solve the problem. It was a crowdsourcing project and every body took part from the community including the greatest scientists such as Newton. However, a carpenter John Harrison found the solution: a watertight clock. Therefore, diverse community is the base of crowdsourcing [9].

### B. Crowdsourcing Software Development

Software crowdsourcing is derived from crowdsourcing, and uses an open-call to hire online skilled workers to perform various tasks requirement analysis, design, coding, and testing [6]. This open call is normally an online competition, where all the developers from around the world compete for a task and try to win it, which they prefer to do [10]. Software crowdsourcing can be explicated by using its own word "CROWD" (Community, Remote, Open-call, Web, Diverse) and defined as an open-call for diverse and remote community to participate and collaborate in the various software development activities on the web. This innovative software development model has been proved successful in many big projects by improving the overall efficiency and reducing the development time and cost. However, software development process is based on engineering techniques modelling, simulation, prototyping, testing and inspection and, therefore, every kind of software project/task may not be suitable for crowdsourcing [11], [12]. Software crowdsourcing largely works well with less complex and stand-alone software development tasks without interdependencies [13]. Due to the large anonymous crowd, it is difficult to deal with several software development issues in software crowdsourcing such as quality, liability, intellectual property rights, information security, privacy, and law enforcement.

Some prominent software crowdsource platforms are: Top-Coder, AppStori, Mob4Hire, eLance, oDesk, vWorker, Guru, TaskCity, and Taskcn [14].

### C. Open-sourcing Software Development

In open-source software development, software is normally created as a collaborative effort in which software engineers/programmers refine the software/code and share the changes within the open-source community. The open-source community is a large group of programmers and experts those are not interested in any financial reward or proprietary ownership. The open-source concept is based on 3R: Review, Refine, and Redistribute source code, therefore, software evolves continuously and gets better and better. The main aim of the open-source community is to provide an alternative solution to expensive proprietary software. Therefore, everyone can use it free of charge or at minimum cost. The Open Source Initiative (OSI) is a big movement that protects and promotes open-source software. Generally, open-source software is distributed under licenses that comply with the Open Source Definition (OSD) [15]. There are several popular open-source software available for use: Linux, MySQL, PHP, Blender, MongoDB, git, Android and Mozilla Firefox.

### D. Outsourcing Software Development

In outsourcing software development, an organisation sub-contracts all or part of a software project to a specialist external service provider (third-party) that internal employees might otherwise perform. The third-party bears the responsibility of timely completion of the assigned task for an organisation. However, the ultimate responsibility of the completion of that task remains with the management of an organisation, which is delegating the work. An organisation subcontracts all or part of a software project because of many reasons such as: an organization cannot handle all software development tasks internally due to unavailability sufficient expertise or resources, several tasks are temporary, and the organization does not intend to hire in-house professionals for short-term to perform the tasks, or simple for cost-cutting [16]. Despite many benefits of outsourcing, it may be a risky adventure if not outsourced to the right service provider. Therefore, a thorough investigation on an external service provider and their expertise is the key part of the successful outsourcing software development strategy.

### E. Insourcing Software Development

In insourcing software development, an organization develops and manages a software project/task using its in-house expertise and resources instead of subcontracting to an external service provider. It is just opposite of outsourcing software development model. Insourcing development can be done in various ways such as relocating employees from a department to another or from a project to another, hire new qualified staff to fill in temporary needs, or train the existing staff in order to be able to perform different tasks, that otherwise would have been outsourced [17]. There are many outsourcing issues that impel organisation to perform insourcing software development activities such as inopportuneness or insufficient ROI (return on investment) with previously outsourced job, operational difficulties, communication/language barrier, long

learning curve, time difference between the countries, erosion of in-house IT expertise/innovation. However, insourcing has several challenges such as additional recruitments, training, infrastructure and licenses.

### III. RELATIVE ANALYSIS OF CROWDSOURCING, OPEN-SOURCING, OUTSOURCING AND INSOURCING MODELS

This section presents a relative analysis of these four software-sourcing models. It analyses the two related criteria to provide corresponding strengths and limitations of each model. These software-sourcing models are very extensive and their relative comparisons are very difficult because they have been evolved through different processes and needs. However, this relative comparison is based on a linguistic range “Lower” and “Higher” to render a nimble and broader view of each model with respect to other models. These comparison are relative and might vary due to many other factors such as size of project, team size and infrastructure.

#### A. Development Time vs. Development Cost

Fig. 1 shows the relative comparison of these software-sourcing models based on the development time and cost. The graph illustrates that insourcing incurs the highest development time and cost, then it decreases for the other models with open-sourcing incurring the lowest development time and cost. However, as previously mention it does not include other parameters that might affect this comparison. Based on this relative comparison, their difference of time and cost is quite obvious and clear.

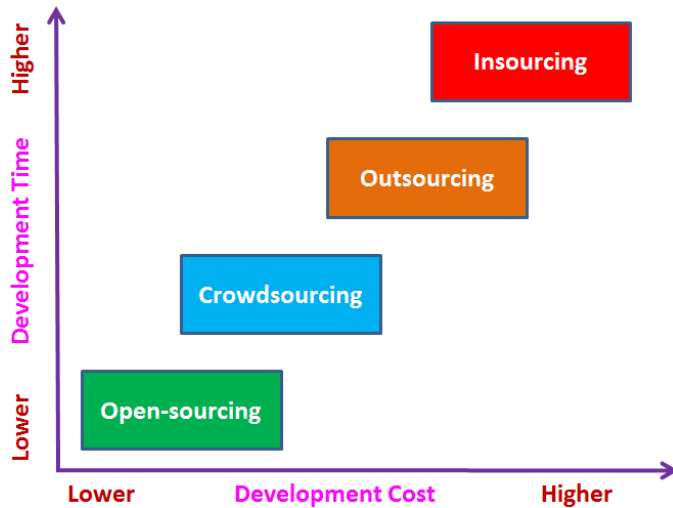


Fig. 1: Development Time vs. Development Cost

#### B. Structured Development vs. Tailored Product

Fig. 2 exhibits the relative comparison of these software-sourcing models based on the use of structured development approach and extent of tailored product. The graph highlights the heavy employment of structured development techniques in insourcing and outsourcing, whereas, crowdsourcing and open-sourcing are more focused on developing or products rather than using a structured approach. However, crowdsourcing and

open-sourcing can employ any rapid development approach such as an agile methodology. Despite less structured and engineering methodology, crowdsourcing and open-sourcing potentially delivers tailored product than outsourcing and insourcing models.

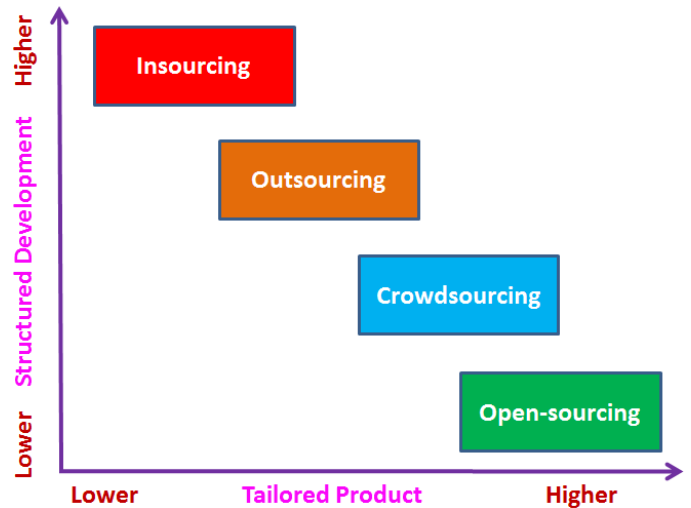


Fig. 2: Structured Development vs. Tailored Product

#### C. User Participation vs. Transparency/Openness

Fig. 3 elicits the relative comparison of these software-sourcing models based on user participation and level of transparency in the software development process. The graph reveals that insourcing and outsourcing are professional-led development and, therefore, user participation and transparency in the development process are very low, whereas, crowdsourcing and open-sourcing are mainly based on user participation and leads a transparent development process.

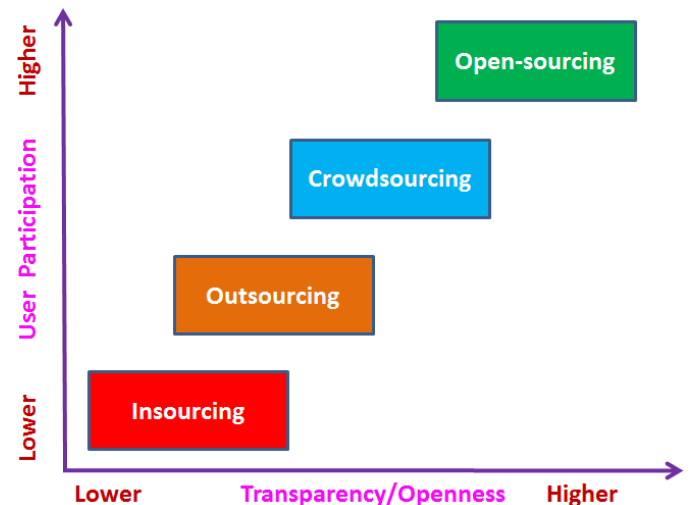


Fig. 3: User Participation vs. Transparency/Openness

#### D. Confidentiality vs. License Provision

Fig. 4 displays the relative comparison of these software-sourcing models based on the level of confidentiality of

information and strong licensing provision. The graph divulges that insourcing maintains the highest level of confidentiality of information because of in-house development. Crowdsourcing and open-sourcing involve a large crowd or community, therefore, they are perhaps not suitable for keeping information confidential. Insourcing is more inclined to private licensing, patents and copyrights, while crowdsourcing and open-sourcing predominantly support wider community and open-source licensing provision.

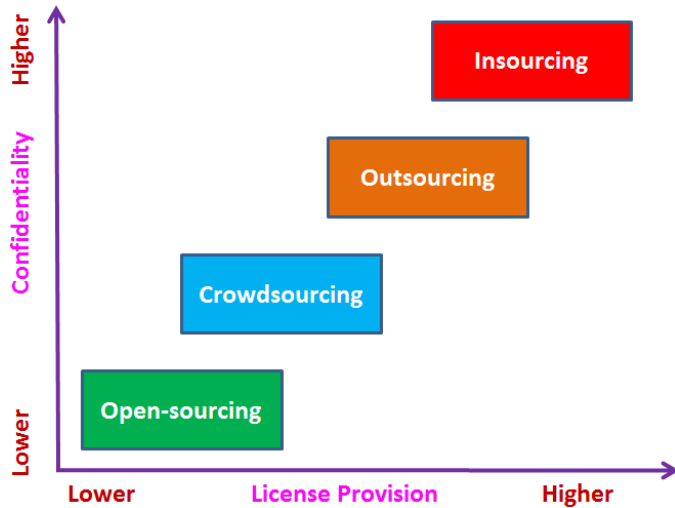


Fig. 4: Confidentiality vs. License Provision

E. Scalability vs. Business Risk

Fig. 5 demonstrates the relative comparison of these software-sourcing models based on the range of scalability and sharing of business risk with others. The graph discloses that open-source and crowdsourcing are easily scalable without much investment and potentially an easy option for business expansions, though this is an expensive affair in insourcing and outsourcing. Also, crowdsourcing and opensourcing can share their business risks with crowd or community across the development process and is safer than insourcing and outsourcing where the complete risk is borne by the organisation only.

F. Operational Control vs. Flexibility/Mobility

Fig. 6 expresses the relative comparison of these software-sourcing models based on the management and operational control and working flexibility. The graph indicates that insourcing has the highest level of management and operational control of the project development with least working flexibility for employees. Crowdsourcing and opensourcing models are based on maximum freedom and working flexibility with the little control mechanism.

IV. COMPARATIVE ANALYSIS OF CROWDSOURCING, OPEN-SOURCING, OUTSOURCING AND INSOURCING MODELS

This section presents a comparative analysis of these four software-sourcing models based on several criteria to clarify their differences at a detailed level. This complete comparative study is shown in Table I:

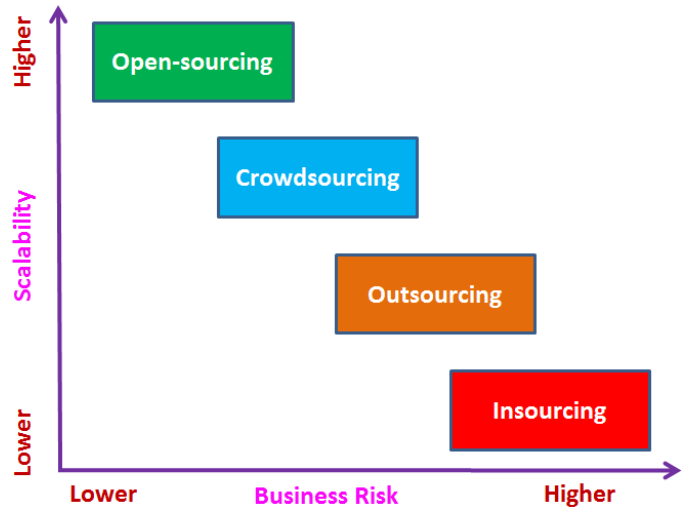


Fig. 5: Scalability vs. Business Risk

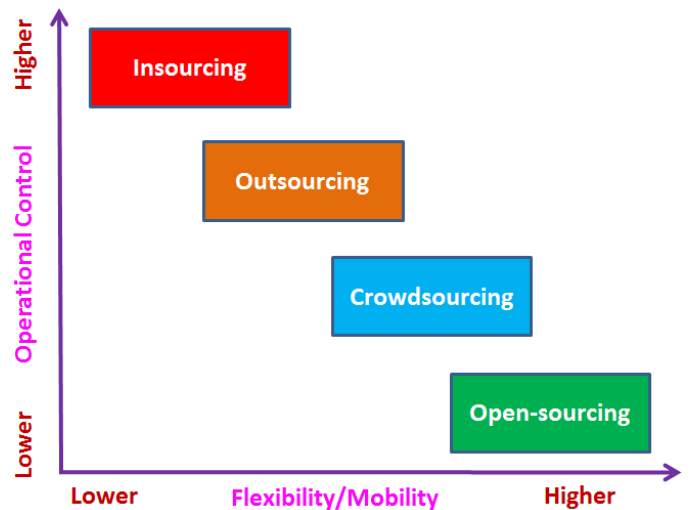


Fig. 6: Operational Control vs. Flexibility/Mobility

V. SUITABILITY ANALYSIS OF CROWDSOURCING, OPEN-SOURCING, OUTSOURCING AND INSOURCING MODELS

Crowdsourcing opens up the software development process to a much broader base of ideas [18]. Crowdsourcing software development is best suited for small and non-sensitive tasks. For example, it is a great option for web designers and designers, in general, to get some usability feedback on their work before making it public [19]. The main application areas are education, healthcare, non-profit and retail business sectors, which are directly related to the public. Therefore, a greater voice of the public can be listened for their problem solving and accelerate innovation [20]. The main attraction has been its comparatively lower price, than the price for hiring a dedicated professional in insourcing or subcontracting to a third-party in outsourcing.

Open-sourcing opens up the entire design/build lifecycle to the public who are not the part of an organization to have their

TABLE I: Comparison among Crowdsourcing, Open-sourcing, Outsourcing and Insourcing Software Development Models

Criteria	Crowdsourcing	Open-sourcing	Outsourcing	Insourcing
1. Development/Maintenance Team	Selected from the Public	Anybody from the public	Third-party/contractor	Private company/contractor
2. Management and Operational Control	Mostly distributed control on the project but it could be a combinational as well	Distributed and informal policy-based control on the project	Partial and distributed control on the project	Centralised and policy-based control on the project
3. Legal Contract	Legal agreement is difficult and loose among multiple parties	Mostly no legal agreement	Legal agreement among all parties	Legal agreement among all parties
4. License Management	Closed/Open/Dual depending on the negotiation	Open source (with varying degree of openness)	Mostly closed and proprietary	Mostly closed and proprietary
5. IP Protection	Only possible when appropriate warranties and indemnities are included in contracts with the contributors	Open source license and creative commons license protection	Contracts and charters corporation, copyright and patents protection	Contracts and charters corporation, copyright and patents protection
6. User Participation	User-led development	User-led development	Expert/professional-led development	Expert/professional-led development
7. Competitiveness	Highly competitive	No competition, as it is a voluntary contribution	Highly competitive	Highly competitive
8. Financial Rewards	Reward depending on the project/task and its completion	No/minimum financial reward	Financial contract based on company's policies	Financial contract based on company's policies
9. Working Flexibility/ Mobility	Mostly flexible - anytime and anywhere	Most flexible model - anytime and anywhere	Less flexible, dependent on third-party's infrastructure and policies	Less flexible, dependent on company's infrastructure and policies
10. Development Approach	Company can adopt any structured approach, however, crowds focus on developing a product	Not necessarily structured approach-more focused on developing product rather than engineering approach	Mostly structured approach but selection of engineering model is based on project size and company's adaptability	Mostly structured approach but selection of engineering model is based on project size and company's adaptability
11. Documentation	Fragmentary documentation	Insufficient documentation-rely on discussion forum	Detailed documentation	Detailed documentation
12. Development Cost	Low-cost solutions/services	Minimum cost/normally free for all	More expensive than crowdsourcing but cheaper than insourcing	Most expensive solutions/services
13. Development Time	Less development time than outsourcing and insourcing	Minimum development time	Less time than insourcing but higher than other models	Higher than any other software development model
14. Product/ Service/ Satisfaction	Tailored solutions/services	Tailored solutions/services	Less customised solutions/services than crowdsourcing and open-sourcing	Less customised solutions/services than crowdsourcing and open-sourcing
15. Interoperability	Mostly interoperable and based on open standards	Mostly interoperable and based on open standards	Mostly less interoperable than crowdsourcing and open-source	Mostly less interoperable than crowdsourcing and open-source
16. Transparency/ Openness	Greater transparency than outsourcing and insourcing	Greatest transparency than any other models	Limited transparency/ closed development	Least transparency/closed development
17. Reliability	Depending on the nature of the crowd	Reliable	Highly reliable	Highly reliable
18. Scalability	Easily scalable	Easily scalable	Used to obtain scalability to some extent	Used less scalable than any other models
19. User-Friendliness	User-friendly but may be for specific domain of crowd	More technical skills and training are needed	Mostly user-friendly	Most user-friendly
20. Confidentiality	Generally less confidential/dependent on how it is managed with the crowd	Least confidential as it is an open-source and available for all	Confidentiality dependent on how much sensitive information is shared with third-party	Greater confidentiality than any other models
21. Business Risk	Risk shared with the crowd	Risk shared with the open-source community	Risk shared with partners	Higher risk than any other development models

own design ideas and to execute them [18]. Open-sourcing software development is best suited for products where testing, enhancement, scalability, reliability, interoperability are required at minimum/no cost. If software product requires interoperability with other products, platforms, and businesses, then open-sourcing is the most suitable option. Another, preferred choice of open-sourcing is for those software products which company wants to launch as a “try before you buy a product. Akin to the crowdsourcing software development, price is the biggest motivation behind the popularity of open-source software development. It produces products at minimum cost (sometimes basic versions are free), which may also include some other costs such as virus protection, support charges and ongoing upgrade expenses.

Outsourcing is less open approach than crowdsourcing and open-sourcing but more open than insourcing. Outsourcing software development is best suited for products where company wants to keep control over the design and execution processes to a lesser extent with cost-effectiveness [18]. It is the most common sourcing model to manage all or part of the design from an insource perspective with more options to explore. While, the finished build comes back to the organization and is ultimately issued to the market under the name of the original organization. It is also useful for the task where specialised persons are required to achieve the high quality. Similar to the two previous sourcing models, the cost is another driving factor here, and it is cheaper than insourcing, however, it is not cheaper than crowdsourcing and open-sourcing models. While it reduces the internal costs, allowing the company to focus on its core competencies. In a fast moving technological world, it is unlikely that a company will be able to design a whole spectrum of technical software. Therefore, it is judicious to outsource some of the complex tasks while keeping intact the core business processes [21].

Insourcing is the closed approach and completely different from all the previous open approaches. Insourcing software development is best suited for products where company wants to keep complete control over the entire software development process at a higher cost [18]. It is also the most preferred method for dependable and trustworthy software, which involve critical features such as reliability, safety, security, and availability. It is also important for software companies that want to enrich their in-house IT expertise and innovation for the growth of the business. Especially, in cloud development environment, insourcing is very vital for SaaS (Software-as-a-Service) companies. Consequently, they can develop their in-house products because technology and innovation are the core IP (Intellectual Property) of their business. Also, several venture firms would require insourcing as a condition of investment [22].

## VI. CONCLUSION

This paper has presented the comparative study of different software-sourcing models: crowdsourcing, Open-sourcing, Outsourcing and Insourcing. It has conducted three types of analysis relative, comparative and suitability to expound numerous aspects of all four software-sourcing models. Finally, it has discussed the suitable applications and scenarios where each software-sourcing model offered benefits in the software development and where it can be applied. In future, it may

be interesting to perform critical analysis of these different sourcing models based on statistics and demographics.

## REFERENCES

- [1] K.-J. Stol, P. Avgeriou, M. A. Babar, Y. Lucas, and B. Fitzgerald, “Key factors for adopting inner source,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 23, no. 2, p. 18, 2014.
- [2] M. J. Schniederjans, A. M. Schniederjans, and D. G. Schniederjans, *Outsourcing and insourcing in an international context*. Routledge, 2015.
- [3] Outsourcing-law.com. (2016) Sourcing Models. [Online]. Available: <http://www.outsourcing-law.com/sourcing-models/>
- [4] J. Howe. (2006, June 2) Crowdsourcing: A Definition. [Online]. Available: [http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing\\_a.html](http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html)
- [5] —, “Crowdsourcing: Why the power of the crowd is driving the future of business,” *Crown Business, New York*, 2008.
- [6] K. Mao, L. Capra, M. Harman, and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” *Research Note*, vol. 15, no. 1, pp. 1–36, 2015, May 10.
- [7] N. Ang, L. Grant, and C. Liu. (2011) Software crowdsourcing development. [Online]. Available: <http://softwarecrowdsourcingdevelopment.wikispaces.asu.edu/Software+Crowdsourcing+Development>
- [8] J. Howe, “The rise of crowdsourcing,” *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.
- [9] E.-J. Pfauth. (2008, May 23) Jeff Howe: Crowdsourcing is all about diversity. [Online]. Available: <http://thenextweb.com/2008/05/23/jeff-howe-crowdsourcing-is-about-diversity/#gref>
- [10] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, “Developer recommendation for crowdsourced software development tasks,” in *Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on*. IEEE, 2015, pp. 347–356.
- [11] W. Wu, W.-T. Tsai, and W. Li, “Creative software crowdsourcing: from components and algorithm development to project concept formations,” *International Journal of Creative Computing*, vol. 1, no. 1, pp. 57–91, 2013.
- [12] H. Wang, G. Yin, X. Li, and X. Li, “TRUSTIE: A software development platform for crowdsourcing,” in *Crowdsourcing*. Springer, 2015, pp. 165–190.
- [13] K.-J. Stol and B. Fitzgerald, “Two’s company, three’s a crowd: a case study of crowdsourcing software development,” in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 187–198.
- [14] K. Mao, Y. Yang, M. Li, and M. Harman, “Pricing crowdsourcing-based software development tasks,” in *Proceedings of the 2013 international conference on Software engineering*. IEEE Press, 2013, pp. 1205–1208.
- [15] Opensource.org. (2016) Open Source Initiative. [Online]. Available: <https://opensource.org/>
- [16] R. Mukherjee. (2014) Advantages and disadvantages of outsourcing. [Online]. Available: <https://blog.udemy.com/advantages-and-disadvantages-of-outsourcing/>
- [17] Sbp-romania.com. (2010) To outsource or to insource? [Online]. Available: <http://www.sbp-romania.com/Articles/to-outsource-or-to-insource.aspx>
- [18] D. Stevenson. (2009) Crowdsourcing: Open source for proprietary software. [Online]. Available: <http://it.toolbox.com/blogs/original-thinking/crowdsourcing-open-source-for-proprietary-software-30453>
- [19] Hongkiat.com. (2016) Crowdsourcing: Pros, Cons, and More. [Online]. Available: <http://www.hongkiat.com/blog/what-is-crowdsourcing/>
- [20] Crowdsourcingweek.com. (2015) What is crowdsourcing? [Online]. Available: <http://www.hongkiat.com/blog/what-is-crowdsourcing/>
- [21] Orientsoftware.net. (2016) 10 reasons why outsourcing software development works. [Online]. Available: <http://www.orientsoftware.net/software-outsourcing/why-outsourcing>
- [22] Notioncapital.com. (2015, September 11) Crowdsourcing: Pros, Cons, and More. [Online]. Available: <http://www.notioncapital.com/blog/outourcing-vs-insourcing/>