

Review

Freelancers in the Software Development Process: A Systematic Mapping Study

Varun Gupta ^{1,2,*} , Jose Maria Fernandez-Crehuet ¹  and Thomas Hanne ² 

¹ Departamento de Ingeniería de Organización, Administración de empresas y Estadística, Universidad Politécnica de Madrid, 28006 Madrid, Spain; josemaria.fernandez-crehuet@upm.es

² School of Business, University of Applied Sciences and Arts Northwestern Switzerland, 4600 Olten, Switzerland; thomas.hanne@fhnw.ch

* Correspondence: varun.gupta@alumnos.upm.es

Received: 25 August 2020; Accepted: 25 September 2020; Published: 27 September 2020



Abstract: [Context] Freelancers could catalyze the software development process by providing their niche skills to generate high quality outputs. They could help companies (including startups) to foster innovations by suggesting creative ideas and providing their expertise in implementing them (for instance, designing solutions, coding solutions etc.). Freelancers could effectively and efficiently work as a virtual member of the software development team. The company must make informed decisions about which task to allot to the freelancer, which freelancer to select, pricing the task, and evaluating the submitted work. On the other hand, the freelancer should make an informed decision about evaluating the monetary value of the task to be charged, trusting the requester, analyzing the skills requirement of the task (finding matches between skill requirement and skills processed), selecting the best task, and maintaining the highest level of reputation. However, the literature does not provide freelancers and the companies the guidelines that support their decision making. However, if freelancers are selected carefully for the most suitable task, the companies will benefit a lot in terms of improved software development metrics. [Objectives] The objective of this paper is to provide the research community the research trends in freelancer-supported software development. This helps to understand that which software development areas have higher concentrations of research efforts, which area has the support of empirical evidence to support management decision makings, and which area requires the research attention. [Method] The systematic study is conducted by planning the mapping protocol, executing the protocol, and reporting the findings using various visualization tools like bar charts and pie charts. The search process was planned to be executed using set of inclusion and exclusion conditions on four bibliographic databases (IEEEExplore, Springerlink, Sciencedirect, and ACM digital library). The relevant papers are selected by applying inclusion and exclusion conditions. The google citations of the relevant papers are subject to the inclusion and exclusion conditions again to include the more relevant papers. Finally, the systematic schema was created and populated after analyzing the studies abstracts. [Results] The results indicate the following (a) The research focus is on generic software development (78%) rather on individual life cycle activities. (b) The number of empirical studies is limited (25%). (c) A number of studies proposing solutions and evaluating on live cases in industrial settings are missing from the literature. This is in comparison to the validation approaches (72%) i.e., solutions tested in laboratory settings. (d) At present, the literature has limited ability to provide the software companies (including startups) with the guidelines (in the form of opinions and experience reports) for involving freelancers in the software development process. (e) The reported challenges include Collaboration and Coordination (33%), Developer Recommendation (or selection) (19%), Team Formulation (14%), Task Recommendation (allocation) (14%), Task Decomposition (11%), Privacy and Security (Confidentiality) (11%), Budget Estimation (8%), Recognition (8%), Trust Issues (8%), Market Dynamism (6%), Intellectual Property Issues (6%), Participation of Crowd Worker (6%), and Capacity Utilization (3%). These challenges are highly interactive, and each challenge impacts all other challenges. (e) Recent

focus of the researchers (total 7 studies in 2019) is on generic software development handling the collaboration and coordination (3 studies out of 7), Developer recommendation (2 studies out of 7), and task recommendation (2 studies out of 7). **[Conclusion]** The freelancer-driven software engineering research area has got the attraction of the researchers, but it will take a long time to gain maturity. This puts an urgent call for more empirical studies and evaluation-based solution research that could help companies (including startups) to foster innovations. Further, the research focus should be well distributed among the various development phases to address the unique challenges associated with individual activities. The accurate management of the freelancer in the software development could help companies and startups to foster innovations and remain competitive in the marketplace.

Keywords: freelancers; innovation; crowdsourcing; software development; systematic mapping study

1. Introduction

Freelancers the self-employed people who have a short term, task-based association with the employer and thus they are not a part of the company workforce (i.e., they are not the employees). Their association with the company is only until the allocated task is successfully completed and thus they do not have long term commitments (or working associations) with the company as expected from the full time employees. They have an obligation to complete the task with high quality standards within the agreed deadline, in exchange for the money (agreed price of the outsourced tasks). During the time the freelancer has been executing the outsourced task, they are free to undertake more freelancing tasks with different requesters (different companies), with different contractual conditions. In other words, the freelancers undertake multiple projects in parallel. Freelancers do not have strict and lengthy written employment contracts as exist in full time employment. The outsourcing of the tasks (also termed as projects) in the freelancing world (gig economy) works as follows:

- (a) The company requests for the services of the freelancers. This could be accomplished in either of the following ways:
 - Publishing task description, deadline, price, and other conditions on the freelancing platforms or on the company website or on social media or on the platforms managed by the company itself. This approach requires open call for freelancer participation and is thus crowdsourcing-based freelancing.
 - The freelancers in the professional networks of the software company (or startups) are directly approached with the outsourcing request. This approach is usually undertaken if the deadline for execution of the project is too strict that prohibits the idea of outsourcing to the crowd. This approach does not require open call for participation of freelancers in undertaking outsourcing tasks and is thus non-crowdsourcing-based freelancing. The company could also maintain their own panel of expert freelancers which could be approached in crowdsourcing or non-crowdsourced manner, which is termed as panel-based freelancing model in [1]. Authors in [1] have classification the freelancing association models into task-based, panel-based, or hybrid. Either of the models could be executed in crowdsourced or non-crowdsourced manner.
- (b) Freelancers registers their expression of interest in undertaking the outsourced tasks.
- (c) The company performs the selection of the freelancers for assigning the task for execution. The selection is quite simple in non-crowdsourcing-based models. In such a model, the approached freelancer (which is capable as per company previous working relations) and the requesting company undergo negotiations. If such negotiations are successful, the task is assigned to

the freelancer. In Crowdsourced models, the freelancer selection for the pool of interested freelancer list could be accomplished in a variety of the ways as mentioned below:

- Screen the profiles of the interested freelancers by analyzing their reputation and their capability to produce high quality work. The ratings, previous reviews, sample of previously executed tasks etc. could be helpful. The freelancers could also be interviewed to make final decisions.
 - Evaluate freelancers based on the brief abstract solution provided by them, finally leading to their selection. The abstract solution is the partial solution about the task to be outsourced. To accomplish this, the company must request for abstract solutions at the time the request for the participation was registered on the platform.
 - Evaluate the solutions given by freelancers and select the best one for the reward. This requires the company to request for the complete solutions of the task at the time the request for the participation was registered on the platform. The best solutions are rewarded. This crowdsourcing takes the form of the competition, but it is usually undertaken for the tasks that are not computationally hard and effortful (for instance, logo designing). As the software engineering tasks are effortful, computational hard problems to solve and complex, a competition-based approach is likely to be less feasible here.
- (d) The freelancer submits the work, which is evaluated by the company. After the positive outcome of the evaluation, the freelancer is rewarded with the money (outsourcing task price).

Thus, there are different ways of inviting interest, screening applicants, selecting and evaluating the work. However, the researchers categorize freelancing as either crowdsourced or non-crowdsourced. In the crowdsourced model, the company sends an open call to a crowd of freelancers. Freelancers could collaborate (usually never happens), take part in a competition, or undertake microtask independently. In the non-crowdsourced model, the company selects one freelancer to independently work on the problem (task) to generate a solution. Freelancing activity in software engineering is challenging, as the freelancer should have perfect skills and competencies to handle the technical task (skill match). It is very effortful to invite the interest of a freelancer, select a freelancer from the big pool (or select tasks from the big pool), negotiate the terms and conditions, communicate during the task execution, evaluate submitted work, and trust that payment will be made. The entire process is highly uncertain and risky. This is because the submission of the work and its quality is probabilistic.

Software companies need to continuously innovate their products (or services) to maintain competitive advantage in highly volatile markets i.e., by continuously providing more and more benefits to the software users. The companies also continuously try to do process innovation to improve the software development processes that lead to the benefits to the firms which could be passed on to the users. Thus, the process innovation could also help the company to improve their products and increase the value they provide to its markets. The growth of the companies in the markets helps them to enhance their resources but their organizational processes become more and structured. For instance, the big software company will have a very organized organizational structure with well-defined processes for everything. This limits the ability of the companies to innovate as the rigid structures will resist the changes required to innovate. Further, technologies and customer needs are changing so fast that the big companies may find it very costly (or even harder) to employ the software engineer with niche skills for undertaking the task in hand. These companies may also wish to employ the specialist for short term basis if the requirement for the specialization is for short term only. The startup phase of the companies is more challenging because of the need to continuously innovate because of resource limitations and an ever-changing environment [2]. However, this phase has very flexible organizational structure and processes which are most favorable for supporting the innovation activities. The outcome of the discussion is that the companies (starting from their startup phase) must continuously innovate by overcoming the restrictions imposed by resources and agility of the organizational processes and structures. Innovation not only require the gathering

of the creative ideas but also their implementation and release in the markets to generate value to the customers. For instance, the company may have a brilliant idea about a new product feature, but they need to implement it by executing software engineering process which include generic activities like designing, coding, testing, and release. The resource limitations and flexibility issues with the organizational processes and structures will be effectively managed by outsourcing the innovative idea implementation tasks to the freelancers. Freelancers could also be meaningful sources for suggesting innovative ideas to the companies that could be implemented inhouse or outsourced to the freelancers [3].

Software development involves many effortful, complex, and hard activities like requirement engineering, designing solutions, coding solutions, testing solutions and release, which must be executed continuously (evolutionary software engineering). Each activity imposes different constraints like skills requirement, environment challenges, coordination issues, collaboration issues, need for documentation, etc. The software engineering task to be outsourced should be assigned to the freelancers with the necessary competencies required for the task to be successfully executed. The employing of freelancers is a good solution provided that the effort of managing freelancing is exceeded by the cost and time reductions it brings to the development.

The software companies expect from the research community the effective solutions that could help them to involve freelancers in suitable software development activity and empirical guidelines that could help them to take decisions in the directions. The support of empirical evidences will foster the decision making process of the company management about different aspects including (a) hiring the freelancer vs. undertaking the task in house; (b) identification of the software engineering activity that should be outsourced to freelancers; (c) establishing price estimated; (d) deciding the model for freelancer associations; (e) decision about the crowdsourcing platforms etc. The overall objective is to include freelancers in the software engineering activities to foster innovations in the company to remain competitive in the marketplace.

This paper aims to undertake a systematic mapping study (Section 2) to understand the trends in freelancer-supported software development by structuring this area around software development activity focused, challenges and issues reported, type of research undertaken by the researchers and the temporal trends (publication trends). The objective is to explore this area to maximum breadth, using an efficient mapping protocol to identify the research gaps or the areas that require urgent research focus. This needs the research to bifurcate the complex domain of freelancer supported software engineering into meaningful information to identify the state of art. This state of art will help to identify the research gaps that could motivate the researchers to bridge those gaps by providing empirical guidelines, tools, and techniques for fostering freelancer involvement in various software engineering activities. This will help the software companies (including the startups) to adopt the research solutions provided by the researchers as per the working context unique to their companies to improve software success rates and overcoming resource limitations.

Providing a holistic view of the research area justifies the execution of the research study as a systematic mapping study over a systematic literature review. This is because systematic mapping studies are well suited to provide classification and counting of the broad research areas rather providing empirical evidence-based on the synthesis of the literature [4]. The systematic mapping study conducted in this paper is based on the guidelines disseminated in [5].

This paper is structured as follows; the background related to systematic mapping is given in Section 2. The planning of the mapping protocol is given in Section 3 by highlighting research questions, search string, bibliographic databases, inclusion and exclusion conditions and data aggregation and synthesis. Section 3 executes the mapping protocol and highlights the search results. The data contained in search results are extracted to answer research questions in the same section. This leads to discussions in Section 4, with threats to validity in Section 5. Finally, the paper is concluded and directions for future work are mentioned in Section 6.

2. Background

The systematic mapping studies are undertaken to structure the research area into categories (classification schema), that represents the research trends/focus in the area. Such studies are conducted to get a broad overview of the research available in the research area by undertaking the classification (identifying categories and relationships) and counting (number of publications within classification scheme). This helps the researchers to have a holistic view of the research efforts made by other researchers in the area and identify the research gaps. As this study involves a broad view of the research area, the research questions are formulated broadly (mappings are broad studies rather deeply conducted). The much focus of these studies is to not provide empirical evidence by deeply analyzing the research studies, but rather to highlight the research trends. Thus, the number of studies that forms the basis of systematic study are usually very large and are synthesized by analyzing their abstracts (and conclusion if abstracts are ambiguous). The reliability of such studies is a big issue due to bias involved [6]. The guidelines to conduct the mapping studies are disseminated in [4,5].

Authors in [4] suggested that the mapping studies should involve the following steps:

- Definition of Research Questions (Defining research scope).
- Conduct search.
- Screening of the papers.
- Keywords using abstracts.
- Data extraction and Mapping process.

Research questions are broadly formulated, and bibliographic databases are searched (conducting a search), resulting in a large number of papers. These papers are screened according to the conditions laid down in mapping protocol (inclusion and exclusion criteria) to filter out the non-relevant ones. The relevant ones are subjected for further analysis to create the classification schema. The information extracted by abstracts (and conclusion) is used to populate the classification schema. The mapping process thus consists of three stages i.e., Planning the mapping protocol, executing the protocol, and reporting the findings.

To the best of the researcher's knowledge, the literature lacks the mapping studies that investigate the trends in employing freelancers for the software development. The software development involves a large number of complex activities with high uncertainties, high costs, and time delays. Freelancers could be useful to undertake such challenging tasks, which provides companies the opportunity to take advantage of the freelancer's niche skills, minimization of cost and time, and increased quality. However, these benefits are incurred only if the freelancers are carefully selected to undertake the tasks for which they have the best match to the skills required by the task to be executed.

3. Research Method

3.1. Research Questions

The systematic mapping study protocol tries to find answers to the research questions with the objective of analyzing the research trends of employing the freelancers for the software development tasks. The research questions are framed broadly to meet the objective of capturing the research patterns over the years, focusing on researchers focus, patterns of reported issues, and challenges. The Population and Interventions of the PICO framework are only used to frame research questions, to be less restrictive in conducting the search process.

Research Questions

RQ1. Which software development process activities have involved the freelancers?

RQ2. Which Issues and challenges are associated with the freelancer involvement in the software development process?

RQ3.How does freelancer supported software development research can be classified in terms of research type?

RQ4.How does the freelancer supported software development research evolved over the time?

3.2. Search String and Bibliographic Databases

The systematic mapping protocol triggers four bibliographic databases against the search string formulated as (Freelancer*) AND (“Software Engineering” OR “Software Development”). The databases were triggered for the publications published from 2015 to 2020 to yield an initial list of publications on 17th March 2020.

The use of the three keywords i.e., freelancer*, software engineering, and software development, extracts those papers that specially mention freelancer (or freelancers) in the context of software development and/or software engineering. This excludes the studies focused on crowdsourcing in general without focusing on freelancers in particular. The objective of this paper is to analyze the studies, especially based on freelancers and not on generic crowdsourcing.

The initial list is subjected to the selection process in accordance with the Inclusion and exclusion criteria to get a final list of studies to answer research questions. Following that, four Bibliographic databases were searched for studies to meet the objectives of this research paper.

- IEEExplore.
- ACM Digital Library.
- Springerlink.
- Sciencedirect.

The google scholar citations of the final papers were subjected to inclusion and exclusion criteria. Those satisfying the conditions are added to the list of selected studies that are considered in this paper.

3.3. Inclusion and Exclusion Criteria

Execution of the systematic mapping protocol search string yields a large number of papers. All these papers do not have the ability to answer the research questions that are of prime interest for the researchers i.e., does not answer within the research scope. The extracted papers are subjected to the selection process by executing the conditions in inclusion and exclusion criteria. Further, the google citations of the final papers were subjected to filtering on the basis of the inclusion and exclusion criteria. The final papers are analyzed to answer the framed research questions. The systematic mapping protocol for this paper specified the following conditions for the inclusion and exclusion of research studies for further analysis.

The objectivity of the inclusion and exclusion criteria is ensured by testing it over the same bibliographic databases, using the same search string but only for the year 2020. The resulting papers selected by each researcher are compared with each other to match the level of agreements between researcher’s selection decisions. Although there was high agreement, minor changes were made in some inclusion and exclusion criteria to enhance its objectivity.

Inclusion Criteria

- Papers must be related to software engineering and freelancers (both crowdsourcing and non-crowdsourcing-based freelancers)
- Papers must be able to answer either of research questions.
- Papers with clear and understandable abstract, making researchers able to answer either research question.
- Paper must be written in English.

Exclusion Criteria

- Papers written in any other language except English,

- Papers not able to answer either research question.
- Papers not based on Software Engineering and freelancers (i.e., filtering out generic crowdsourcing papers)
- Papers mentioning the freelancers in software engineering only at abstract levels.
- All except the one occurrence of the Papers appearing multiple times in various bibliographic databases are excluded.
- Papers with minimal details about the findings about freelancers in software development, requiring researchers read the entire paper. For example, consider the paper with an abstract, mentioning the following: researchers report various challenges in employing freelancers for small projects in the paper. Such papers are filtered out as they require researchers to read the entire paper to gather reported challenges.

3.4. Data Aggregation and Synthesis

The abstracts of the final papers will be subjected to analysis. The analysis will result in the identification of related categories that answers the framed research questions. This is termed as structured schema. During the process of reading abstracts, for each research question, the related keywords are identified and merged to create this schema. To ensure validity of the results, the data extracted from the studies were first added to the data extraction form and then mapped to the conceptual schemas. The structure of data extraction form is given in Annex-I by all the researchers. Any disagreements were resolved in the consensus meetings.

The data extracted after abstract analysis is used to populate the individual schema, which answers the individual formulated research questions under investigation. The extracted data for the categories of schema are maintained in google excel sheet to classify and count the research types associated with them (research questions 3 and 4). This excel sheet helps to count the artefacts of interest to the researchers to answer research questions 3 and 4, which involve further categorization and counting.

4. Results

4.1. Search Results

The execution of the systematic mapping protocol resulted in 17 studies being selected from four bibliographic databases. The google citations of 17 papers were analyzed and 19 of these citations were found to satisfy the inclusion conditions as laid down in mapping protocol. A total of 36 papers were considered for analysis for meeting the objective of this research. The result of execution of the search process, first over individual databases and then over citations as per the protocol, are given in Table 1.

Table 1. Execution of Systematic Mapping Protocol.

Bibliographic Databases	# Initial Papers	# Papers (After Applying Inclusion & Exclusion) (a)	# Papers Citations (b)	# Papers (After Applying Inclusion & Exclusion on (b)) (c)	# Final Papers (a + c)
IEEEExplore.	07	04	45	10	14
ACM Digital Library.	82	05	45	04	09
Springerlink.	228	07	13	05	12
Sciencedirect	120	01	0	0	01

4.2. Studies Selected for Review

The following papers were finally selected for further analysis to answer the framed research questions (Table 2).

Table 2. Selected research studies.

Paper Details	Reference ID	Bibliographic Database	Year	Citations	Number of Google Citations Selected
Dwarakanath, A.; Chintala, U.; Shrikanth, N.C.; Virdi, G.; Kass, A.; Chandran, A.; Paul, S. Crowd build: A methodology for enterprise software development using crowdsourcing. In Proceedings of the 2015 IEEE/ACM 2nd International Workshop on CrowdSourcing in Software Engineering, Florence, Italy, 19 May 2015; pp. 8–14.	[7]	IEEE	2015	27	08
Bernabé, R.B.; Navia, I.Á.; García-Peñalvo, F.J. Faat: Freelance as a team. In Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 07–09 October, 2015; pp. 687–694.	[8]	ACM	2015	08	Nil
Song, F.; Chen, H.; Fu, Y. An approach to rapid worker discovery in software crowdsourcing. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015), Melbourne, VIC, Australia, 9–11 December 2019; pp. 370–382.	[9]	Springer	2015	02	02
Murray-Rust, D.; Scekkic, O.; Lin, D. Worker-centric Design for Software Crowdsourcing: Towards Cloud Careers. In Crowdsourcing; Springer: Berlin/Heidelberg, Germany, 2015; pp. 39–50.	[10]	Springer	2015	03	Nil
Shao, W.; Wang, X.; Jiao, W. A developer recommendation framework in software crowdsourcing development. In National Software Application Conference; Springer: Singapore, 2016; pp. 151–164.	[11]	Springer	2016	05	03
Abhinav, K.; Dubey, A. Predicting budget for Crowdsourced and freelance software development projects. In Proceedings of the 10th Innovations in Software Engineering Conference, Jaipur, India, 5–7 February 2017; pp. 165–171.	[12]	ACM	2017	05	Nil
Al-bloush, H.; Solemon, B. Intellectual Property Challenges in the Crowdsourced Software Engineering: An Analysis of Crowdsourcing Platforms. In International Conference of Reliable Information and Communication Technology (IRICT 2017), Malaysia, 22–23 September 2017; pp. 875–884.	[13]	Springer	2017	0	Nil
Chen, Y.; Lee, S.W.; Xie, Y.; Yang, Y.; Lasecki, W.S.; Oney, S. Codeon: On-demand software development assistance. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 06–11 May 2017; pp. 6220–6231.	[14]	ACM	2017	23	03
Lee, S.W.; Chen, Y.; Klugman, N.; Gouravajhala, S.R.; Chen, A.; Lasecki, W.S. (2017). Exploring coordination models for ad hoc programming teams. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, Denver Colorado USA, 06–11 May 2017; pp. 2738–2745.	[15]	ACM	2017	05	01
Abhinav, K.; Dubey, A.; Jain, S.; Virdi, G.; Kass, A.; Mehta, M. Crowd Advisor: A framework for freelancer assessment in online marketplace. In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Buenos Aires Argentina, 20–28 May 2017; pp. 93–102.	[16]	IEEE	2017	10	01
Dubey, A.; Singi, K.; Kaulgud, V. Personas and redundancies in crowdsourced testing. In Proceedings of the 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), Buenos Aires Argentina, 22–23 May 2017; pp. 76–80.	[17]	ACM	2017	01	Nil

Table 2. Cont.

Chatterjee, A.; Varshney, L.R.; Vishwanath, S. Work capacity of regulated freelance platforms: Fundamental limits and decentralized schemes. <i>IEEE/ACM Trans. Netw.</i> 2017, 25, 3641–3654.	[18]	IEEE	2017	05	Nil
Dubey, A.; Abhinav, K.; Virdi, G. A framework to preserve confidentiality in crowdsourced software development. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires Argentina, 20–28 May 2017; pp. 115–117.	[19]	IEEE	2017	04	01
Arenas, C.; García, V.H.M.; Otálora, J.E. Crowdsourcing as a knowledge management mechanism in the elicitation of mission software requirement. In Proceedings of the International Conference on Knowledge Management in Organizations, Zilina, Slovakia, 6–10 August 2018; Springer: Cham, Switzerland, 2018; pp. 288–298.	[20]	Springer	2018	01	Nil
Tsikerdekis, M. Persistent code contribution: A ranking algorithm for code contribution in crowdsourced software. <i>Empir. Softw. Eng.</i> 2018, 23, 1871–1894.	[21]	Springer	2018	02	Nil
Zhang, Z.; Sun, H.; Zhang, H. Developer recommendation for Topcoder through a meta-learning based policy model. <i>Empir. Softw. Eng.</i> 2020, 25, 859–889.	[22]	Springer	2019	0	Nil
Ivan, I.; Budacu, E.; Despa, M.L. Using profiling to assemble an agile collaborative software development team made up of freelancers. <i>Procedia Comput. Sci.</i> 2019, 162, 562–570.	[23]	ScienceDirect	2019	0	Nil

The mapping process is executed in the two stages. In the first stage, the studies were selected from four bibliographic databases and in the second stage, the studies were selected from google citations of the papers selected in the first stage. Conditions of selections of the studies were laid down in mapping protocol. Table 3 gives the studies selected from google citations.

Table 3. Selected Google Scholar citation studies.

Paper Details	Paper Reference ID	Parent Paper Reference ID	Publication Year
Dubey, A.; Abhinav, K.; Taneja, S.; Virdi, G.; Dwarakanath, A.; Kass, A.; Kuriakose, M.S. Dynamics of software development crowdsourcing. In Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Orange County, CA, USA, 2–5 August 2016; pp. 49–58.	[24]	[7]	2016
Long, F.; Yang, C.; Rong, H.G.; Li, J.F. A user-oriented resource scheduling method for improving agile software pattern in cloud environment. <i>J. Cent. S. Univ.</i> 2016, 23, 2906–2916.	[25]	[7]	2016
Machado, L.; Kroll, J.; Marczak, S.; Prikladnicki, R. Breaking collaboration barriers through communication practices in software crowdsourcing. In Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Orange County, CA, USA, 2–5 August 2016; pp. 44–48.	[26]	[7]	2016

Table 3. Cont.

Sharma, S.; Hasteer, N.; Van-Belle, J.P. An exploratory study on perception of Indian crowd towards crowdsourcing software development. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 901–905.	[27]	[7]	2017
Machado, L.D.S. Empirical Studies about Collaboration in Competitive Software Crowdsourcing; 2018, PhD Thesis, Pontifical Catholic University of Rio Grande do Sul, Brazil, Link: http://tede2.pucrs.br/tede2/bitstream/tede/8256/2/LETICIA_SANTOS_MACHADO_TES.pdf	[28]	[7]	2018
Sari, A.; Alptekin, G.I. A Prize Determination Approach for Crowdsourced Software Development. In Proceedings of the 2018 2nd European Conference on Electrical Engineering and Computer Science (ECCS), Bern Switzerland, 20–22 December 2018; pp. 450–453.	[29]	[7]	2018
Hasteer, N.; Bansal, A.; Murthy, B.K. Barriers to crowdsource software development: Evaluating the intertwined effects. Int. J. Bus. Inf. Syst. 2019, 32, 127–144.	[30]	[7]	2019
Illahi, I.; Liu, H.; Umer, Q.; Zaidi, S.A.H. An Empirical Study on Competitive Crowdsourced Software Development: Motivating and Inhibiting Factors. IEEE Access 2019, 7, 62042–62057.	[31]	[7]	2019
Karim, M.R.; Yang, Y.; Messinger, D.; Ruhe, G. Learn or Earn? Intelligent Task Recommendation for Competitive Crowdsourced Software Development; 51st Hawaii International Conference on System Sciences, Hawaii, USA, 2–6 January 2018; pp. 5604–5613.	[32]	[16]	2018
Johnson, M.; Stevens, P. Confidentiality in the process of (model-driven) software development. In Proceedings of the Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming, New York, NY, USA, 9–12 April 2018; pp. 1–8.	[33]	[19]	2018
Kaur, H.; Gordon, M.; Yang, Y.; Bigham, J.P.; Teevan, J.; Kamar, E.; Lasecki, W.S. Crowdmask: Using crowds to preserve privacy in crowd-powered systems via progressive filtering. In Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing, Québec City, QC, Canada, 23–26 October 2017.	[34]	[14]	2017
Lee, S.W.; Krosnick, R.; Park, S.Y.; Keelean, B.; Vaidya, S.; O’Keefe, S.D.; Lasecki, W.S. Exploring real-time collaboration in crowd-powered systems through a UI design tool. Proceedings of the ACM on Human-Computer Interaction; ACM, New York, NY, USA, 2018; pp. 1–23.	[35]	[14]	2018

Table 3. Cont.

Aghayi, E.; LaToza, T.D.; Surendra, P.; Abolghasemi, S. Implementing Microservices through Microtasks. <i>arXiv Prepr.</i> 2019, arXiv:1903.01977.	[36]	[14]	2019
Binzagr, F.; Medjahed, B. Crowdmashup: Recommending crowdsourcing teams for mashup development. In <i>Proceedings of the International Conference on Service-Oriented Computing</i> , Hangzhou, China, 12–15 November 2018; Springer: Cham, pp. 679–693.	[37]	[15]	2018
Li, S.; Chen, H.; Shu, Y.; Hu, F. The verification technique of complex tasks' nonfunctional specifications in software crowdsourcing. In <i>Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT)</i> , Chang chun, China, 10–11 December 2016; pp. 166–170.	[38]	[9]	2016
Shu, Y.; Chen, H.; Li, S.; Hu, F. The verification approach to complex tasks' functional specification in software crowdsourcing. In <i>Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT)</i> , Changchun, China, 10–11 December 2016; pp. 171–176.	[39]	[9]	2016
Karim, M. Data Analytics for Optimized Matching in Software Development; 2017, PhD Thesis, University of Calgary, doi:10.11575/PRISM/25515.	[40]	[11]	2017
Yu, D.; Zhou, Z.; Wang, Y. Crowdsourcing Software Task Assignment Method for Collaborative Development. <i>IEEE Access</i> 2019, 7, 35743–35754.	[41]	[11]	2019
Yu, X.; He, Y.; Fu, Y.; Xin, Y.; Du, J.; Ni, W. Cross-Domain Developer Recommendation Algorithm Based on Feature Matching. In <i>Proceedings of the CCF Conference on Computer Supported Cooperative Work and Social Computing</i> , Kunming, China, 16–18 August 2019; pp. 443–457.	[42]	[11]	2019

Table 4 highlights the relation between the parent reference ID and the child reference ID (Google citation included for meeting the research objectives of this systematic mapping study).

4.3. Answer to Research Questions

The selected studies (Tables 2 and 3) are analyzed by reading their abstract followed by execution of the thematic analysis over abstracts to (1) structure the research area into schemas and (2) populate the categories mentioned in the schema, which represents an answer to the framed research question. The answers to research questions are as given below.

Table 4. Relation between Parent & Child studies.

Parent Reference ID	Child Reference ID (Google Citation)
[7]	[24–31]
[9]	[38,39]
[11]	[40–42]
[14]	[34–36]
[15]	[37]
[16]	[32]
[19]	[33]

RQ1. Which software development process activities have involved the freelancers?

The different software development areas where the freelancers are employed are given in Table 5. The categories of development areas are identified through the analysis of abstracts. However, many papers did not mention the development area. Such studies are categorized in the “Generic Development” (meaning software development process as a whole) category in this paper.

Table 5. Software Development Areas Focused by researchers.

Software Development Area	Research Study
Project Planning	[25]
Requirement Management	[20]
Programming/Coding	[14,21,37]
Software Testing	[7,17]
Software Integration	[7]
Debugging	[40]
Software Evolution	[40]
Generic development	[8–13,15,16,18,19,22–24,26–36,38,39,41,42]

The activities mentioned in Table 5 are the only software engineering activities where freelancers were involved as reported by the literature studies. The formulation of the category labels and classification of the studies in these categories were made on the basis of the software engineering activity that the literature studies have focused on.

For some studies, the identification of categories and their assignment to those categories was the biggest challenge. To accomplish this, some assumptions were made in the categorization process. For instance, the two labels, i.e., “software evolution” and “requirement management” as used in this study, consider the categorization driven by the difference of the focus of the studies on the software evolution. For instance, the study as given in [40] is categorized under the label “software evolution” as it suggests the recommendation of the software features to be released in the next version of the software system (focus on software evolution).

However, the studies focusing on requirement management without any focus on software evolution (focusing on managing requirement for a particular version only rather than driving next version development), are assigned to Requirement management category. The study as given in [27] deals with resource management and is hence assigned to the Project Planning category.

Table 5 shows that the majority of the research studies disseminates the aspects related to freelancing for the overall software development. These studies are not particularly based on specific software development areas. Programming/Coding is the main focus of studies that are based

on specific development activity, followed by software testing. However, the number of software development activity specific studies are limited, requiring more emphasis by researchers.

RQ2. Which Issues and challenges are associated with the freelancer involvement in the software development process?

Table 6 gives the patterns of issues and challenges as identified by researchers in the studied research papers.

Table 6. Patterns of research issues and challenges.

Issues & Challenges	Research Study	Percentage Reported (%)
Budget Estimation (Task value estimation)	[12,21,29]	8%
Capacity Utilization	[18]	3%
Collaboration and Coordination	[14,15,17,20,25,26,28,30,35–37,41]	33%
Communication	[26,28,30]	8%
Developer Recommendation (or selection)	[9–11,16,22,23,37]	19%
Intellectual Property Issues	[13,30]	6%
Participation of Crowd Worker (Freelancer)	[27,31]	6%
Privacy and Security (Confidentiality)	[19,30,33,34]	11%
Recognition	[10,21,31]	8%
Task Decomposition	[7,36,38,39]	11%
Team Formulation	[8,10,15,23,37]	14%
Task Recommendation (allocation)	[17,32,40–42]	14%
Trust Issues	[10,24,30]	8%
Market Dynamism (<i>Ex. Needs, task complexity, skills requirement, etc.</i>)	[12,25]	6%

Table 6 lists the issues and challenges as reported by studied research papers. Few studies are associated with multiple rows of the Table 6 (i.e., they had reported multiple issues and challenges). However, they are treated as individual studies for each row (i.e., individual challenge and issues) for calculation of the percentage studies reporting particular challenges and issue.

The majority of studies report coordination and collaboration (33%) as one of the important issues in undertaking the crowdsourcing tasks by freelancers. The idea is that the software development is the collaborative activity and the team must coordinate their efforts to jointly contribute toward solutions. Executing the software development tasks require the team of software engineers to jointly work towards a unified solution to the problem. In a distributed team with freelancers as one of the members, managing such a team and establishing coordination and collaboration is very challenging.

The communication between the freelancer and the software engineering team is another issue reported by the 8% studies. The objective is to establish the channels for effective communication which is hindered by the global distances.

This implies that the task decomposition (another most reported issue) (11%) will be very challenging, as making them self-contained (making them more independent without the need for high collaboration) will require much expertise and effort. The reason is that the decomposition of the complex task into independent tasks to be outsourced is very challenging because the companies may have limited resources to accurately perform the decomposition. The challenges are further amplified because the decomposition team must decompose the complex task not only considering independence among them, but also, they need to hide the project-related confidential information from each task description.

Developer recommendation (19%), i.e., recommending developers to the requesters (companies requesting the freelancer's participation), is another highly reported issue. The developer

recommendation system will recommend suitable freelancers to the requesters after the task is outsourced in the crowd of freelancers requesting for their participation. The recommendation is made on the optimal combination of the values representing the freelancer's ability to execute the task. This may include their skills, previous ratings, recent feedback, average time to completion, average charges charged, etc. The developer recommendation is a challenging activity because it is hard to identify parameters for the recommendation, dynamically updating the parameter values, matching task requirements with developer competencies, etc. Most of the studies either propose a recommendation system or report the challenges with recommendations.

Team formulation is reported by 14% of the studied papers. The challenge associated with the Team formulation is the difficulty experienced by software companies to form the team of freelancers for undertaking the crowdsourced tasks. This may be due to the coordination and communication issues, which leads to trust issues. Further, freelancers work from their workplaces as per their own schedule, so establishing the co-located team working is very difficult.

Trust issues (8%) emerge when requesters are not sure if the freelancers will be able to work together and will submit the requested solutions. The trust issues emerge because both freelancers and requestors do not know each other. The freelancers are not sure if requestors will pay them after completion of the work. The requestors are not sure if the freelancers will submit the solutions at all or if they do, then whether the submission will be of acceptance quality standards and within the deadline. The freelancing platforms have maintained the work profile of both freelancers and the requestors, to help each other to establish trust. Further, the requestors deposit the freelancer payments with the platform, which is only released to freelancers after the requestors have positively evaluated the quality standards of the deliverables delivered by the freelancers.

On the opposite side of developer recommendation, there is task recommendation (14%). Stated in other words, this will recommend suitable tasks to the developer, which otherwise will be quite effortful to search amongst the numerous tasks. Privacy and security (11%) associated with the software artefacts are another observed challenge as reported in the studies. The objective is to maintain privacy of the objects related to the project described in the outsourced task and securing the company or project related information that could be meaningful to the competitors. This could be accomplished in many freelancing platforms by signing non-disclosure agreements with the freelancers to avoid sharing the project related information with third parties. Additionally, the tasks are designed in a manner that gives minimal information about the project background. Confidentiality associated with the task to be crowdsourced is reported in 11% of the studies.

Intellectual Property Issues had been reported by 6% of the studies. The objective here is to establish the ownership of the solution submitted by the freelancers. The requestors should be the owner of the solution, but this issue need to be resolved before outsourcing the task, i.e., whether the intellectual property ownership will remain with the platform, freelancer, or with the requestor.

Market dynamics (6%), i.e., changes in the market, affects the activities related to the outsourcing of the tasks. For instance, continuous changes in the market like supply of niche skills, ongoing market rate, etc. affects the prices to be offered to freelancers. Participation of the crowd workers, i.e., freelancers had been reported in 6% of the studies. The challenge is to ensure that the freelancer participates in the freelancing activities continuously. The opportunities to learn, gain money, social interaction, etc. should motivate them to continuously take part in the execution of the outsourced tasks.

These challenges had a direct impact on each other. For example, communication helps to enhance coordination and collaboration, which enhances trust. This impacts team formulation, which impacts trust issues.

RQ3. How does freelancer supported software development research can be classified in terms of research type?

The authors categorize the research studies in different types using the various categories of different types of research studies as mentioned in [43], with the following interpretations:

Solution papers (papers with new solution for the problem with no strong validation or evaluation, a small hypothetical example could be a part of it), **Validation Research papers** (solutions must be validated using experimental data, simulations, or in laboratory settings), and **Evaluation research papers** (solutions must be executed (validated) in real scenarios). **Empirical research** is added as the new category which is equivalent to the evaluation category used in [43].

The studies could also be categorized as Opinion papers, Experience reports, and Philosophical Papers. However, the categorization of the paper to either of the category depends on the contribution made in the paper by a relevant category. For example, if a paper proposes a new solution, validates in laboratory, and finally suggests a framework, then this paper will be categorized depending on the relative contributory share of each category in it.

The temporal trends of type of research for software development are given in Figure 1, with the help of a chart.

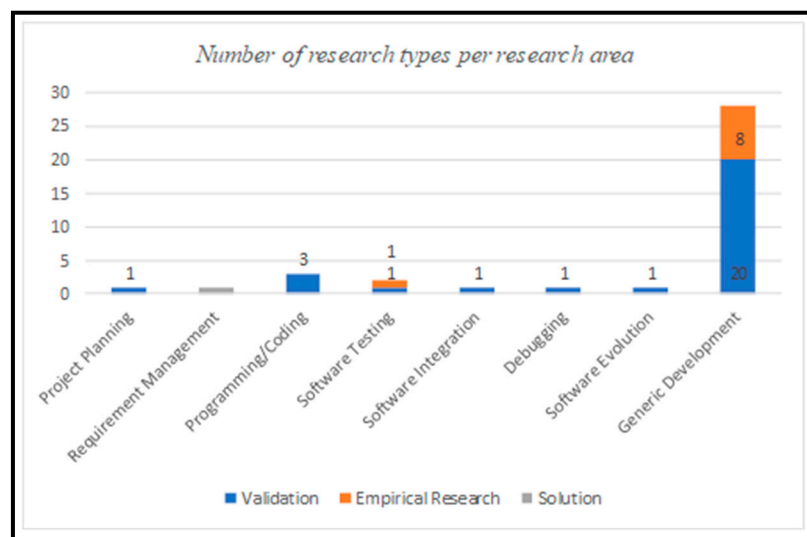


Figure 1. Research study types for each research area (software development).

Figure 1 highlights the following points:

- Researchers had focused on employing freelancers (or crowdsourced freelancers) for generic software development. This means that the research studies do not focus on particular software development activity. This accounted for 78% of studied papers in this research (28 out of 36).
- The studies focusing on generic software development (78%) amongst total studied in this research, the validation-based approaches outweigh empirical studies by 150%. Stated in other words, empirical studies got less attraction of the researchers as such studies are 60% less than the validation-based approaches. The researchers had focused on proposing the solutions and validating them in a laboratory setting. The research results are not validated in Industrial settings as live scenarios.
- The number of studies reporting solution type research (with no validation) are limited i.e., only 1 out of 36 studies (3%) had proposed solutions which are illustrated with small or hypothetical examples (no validation).
- There had been little focus on programming or coding focused studies. The studies are usually validation-based approaches.
- Other areas that got attraction (but little) were project planning, software integration, debugging, and evolution.

The temporal trends of type of research for reported issues and challenges are given in Figure 2, with the help of a chart.

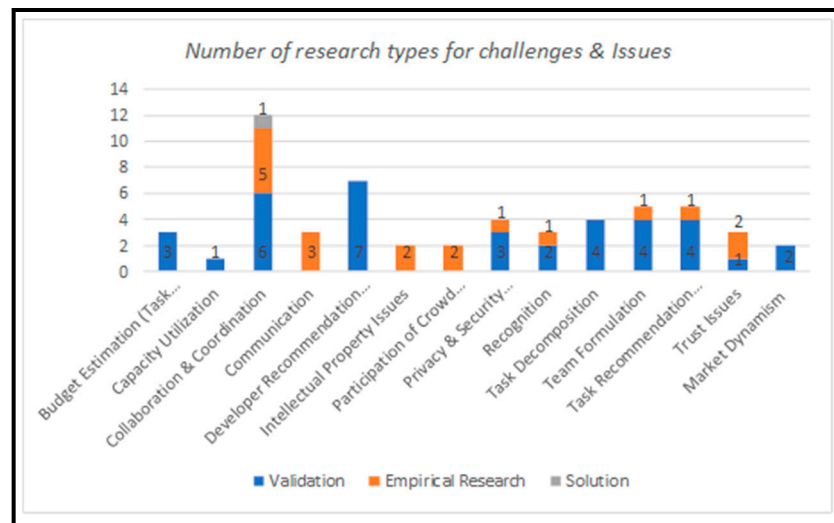


Figure 2. Research study types for each addressed issue and challenges.

Figure 2 highlights the following points. However, an individual paper had highlighted multiple issues and challenges. The paper had been counted as a single paper for each highlighted point.

- Most of the researchers had highlighted the issue of collaboration and coordination in their studies (reported in 12 studies). The validation approaches are marginally larger than the empirical research. One study is solution-based research.
- Developer recommendation is the second highest reported and addressed issue in studies (7 studies).
- Team formulations with crowdsourced freelancers and task recommendations are third highest reported and addressed issues (5 studies each).
- Task decomposition and confidentiality is reported in 4 studies each, followed by trust issues (1 validation, 2 empirical), budget estimation, and communication (reported in 3 studies each).
- The validation-based approaches outweigh the empirical research studies. The pie chart (Figure 3) shows that validation-based approaches (72%) are conducted by researchers that use empirical validation-based research (25%). Solution-based research is not much focused on (3%).

The pie chart shows the numerical proportion share of the type of research studies amongst the studied research papers.

Figures 1–3 collectively highlight the various focused software engineering research areas and highlighted (or addressed) issues and challenges. It shows that the research work is much focused on proposing and validating the solutions in a laboratory setting. The focus on validating solutions in industrial settings (using real scenarios) is very limited.

There is not much focus of the researchers on performing empirical studies, which shares that just 25% of the research studies were conducted using freelancers in software engineering. Further, these empirical studies are mostly conducted in generic software development and the studies addressing various software development phases and providing empirical evidence for artefacts under investigation are very limited.

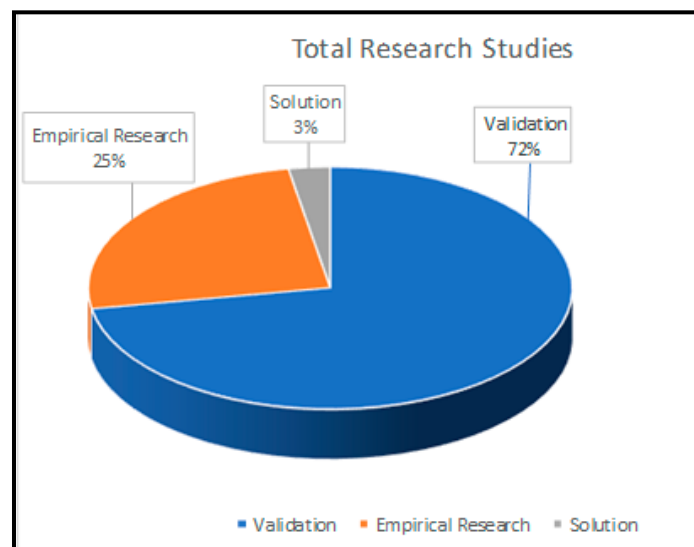


Figure 3. Number of studies categorized into different research types.

RQ4. How does the freelancer supported software development research evolved over the time?

This research question is answered by categorizing the research answer to the previous research question (RQ 3), year wise (publication year).

Figure 4 shows the year wise trends of the focus of the studied research papers on the various software development activities. This figure helps to understand which activity got researcher attention and when.

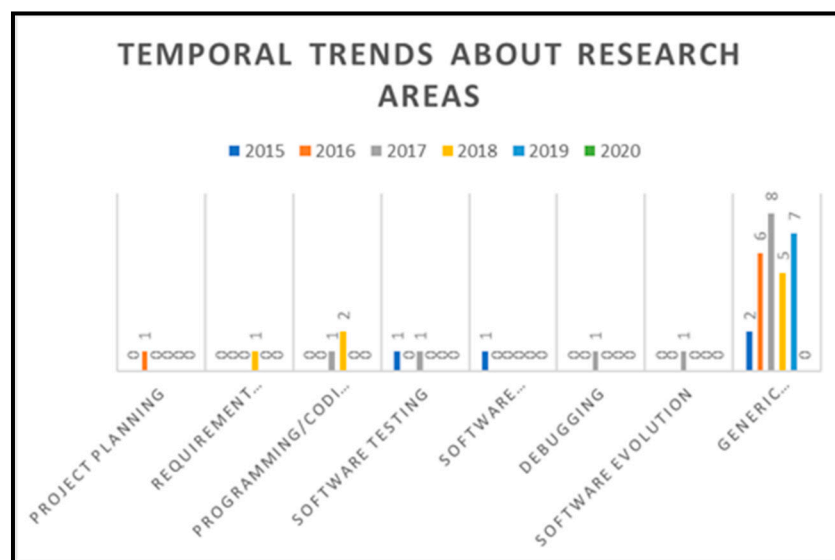


Figure 4. Temporal research trends about focused research areas (software development).

The facts related to the researcher focus on various software development activities and types of research understood by them is already explained using Figure 1. The main points are relevant to Figure 4 are as follows:

- The year 2020 does not yield any paper that focuses on studying the empirical evidence and implementing solutions for involving the freelancers in the process of the software development.
- The recent work done by researchers had been targeted towards freelancers in generic software development. Hence, they are not focused on any individual development activity.

Figure 5 shows the temporal trends in various issues and challenges addressed by researchers that were identified by employing the freelancers for software development activities. This figure helps to understand which issues and challenges are still getting researcher focus.

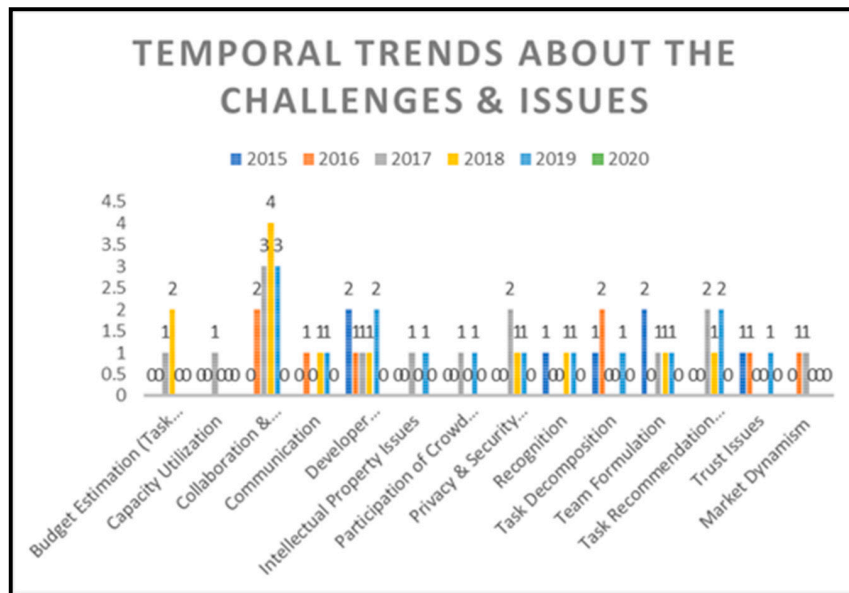


Figure 5. Temporal trends about focused issues and challenges.

The facts related to the researcher focus on various issues and challenges are already explained in Figure 2. The main points relevant to Figure 5 are as follows:

- The year 2020 does not yield any research study. This may be because the systematic mapping was conducted in March 2020, which is at the beginning of the year.
- Collaboration and coordination is still an active research issue in the year 2019 (addressed by 3 studies).
- Developer recommendation and task recommendation ranks second in the list of the most addressed issues in the year 2019 (addressed by 2 studies).
- Collaboration and coordination, developer recommendation, and task recommendation had been addressed by most of the studies (Figure 2). This had been addressed in 2019 also. This implies that more research is required in these areas.
- Team formulation and task decomposition got less attraction in the year 2019.

5. Discussion

The research studies had been focused on improving the generic software development activities by employing freelancers rather employing them for different development activities (Figure 1). The software development activities require different technical skills and expertise. The software companies may not have immediate availability of the human resources with the niche skills to undertake the particular task. The small companies and the startups may have limited resources with multiple roles to perform. The limited resources limit the ability to implement innovative ideas. Freelancers seem to be a good option as they bridge the skill gaps and provide ability to these firms to implement innovations. The successful associations with the freelancers depend on the accurate selection of the freelancer from the pool of interested freelancers. Individual software development activities require different skills and processes different research challenges to address. The same freelancer pool cannot serve the requirements of all development activities. The researchers' focus on adopting the freelancers involvement in all software engineering activities uniformly will help to

provide empirical evidences, tools, and techniques to the companies that will help them accurately design tasks to be outsourced, accurately select freelancer, accurately establish working relations in the distributed team, maintain ethical standards, and find value generating solutions to the problem. The current state of art is that the freelancer's involvement across different development activities is non-uniform, with a major focus on generic software development (Figures 1 and 2).

For instance, the failure rates of the startups are due to requirement related issues [44–47] and the product development issues [48]. The state of research in the domain of the requirement engineering in the startup context is still in early stages. The literature has limited ability to support the startups in the form of empirical evidence, evaluated solutions, etc. that could be adopted as per their working context and finally implemented [2]. Big companies also do a lot of marketing research to identify the innovative ideas for product innovation i.e., changing the product to enhance value to the customer. The startups and big companies could involve the freelancers in requirement engineering activity not only to generate new product ideas but also to create requirement models and mapping models to the requirement specification documentation [3]. The freelancers could be of prime importance in undertaking requirement management activity since they are globally distributed over the globe. They could be used as the source of the requirements for future software versions and as a source of effective feedback for supporting the software evolution [3].

However, employing freelancers for requirement management will require empirical studies to investigate issues like trust, confidentiality, validation of freelancer feedback, value-based pricing, motivations to support their active participation (especially for financially constrained startups), etc. The solution strategies will be challenging to be evaluated in live settings as simulating the live experience will be effortful and difficult.

Further, the number of empirical studies focusing on individual and overall software development are very limited (Figures 1–3). This limits the empirical evidence that could attract researchers' interest, highlight unique challenges, and suggest research gaps. The lack of experience reports and opinions (which are shared by other companies) in the literature limits the sharing of the knowledge (lesson learned) that could be beneficial to the companies seeking advice about freelancer's involvement. The literature provides 72% validation studies (Figure 3), meaning that solutions are validated in laboratory settings only. Their effectiveness is yet to be established in real settings. This implies that the existing solutions must be checked for their feasibility to be used in industrial settings to gather the real time metrics associated with their impact on the overall business. Limited studies signify that the area has got the attraction of the researchers, but it will take a long time to gain maturity. This puts urgent call for more empirical studies and evaluation-based solution research that could help companies (including startups) to foster innovations. Further, the research focus should be well distributed among the various development phases to address the unique challenges associated with individual activities.

The studied research papers identified various challenges and issues that are incurred in involving freelancers. The reported challenges are ranked in order of number of studies reporting them. However, there exists a relation among the challenges and issues. For example, coordination and collaboration affect team formulation, trust, quality of work, communication and so on. This implies that ranked challenges could represent the prime parameters for targets of future research, yet the researchers must consider their interrelationships.

The literature does not focus on different models of freelancer associations with the company as identified in [1]. The crowdsourced freelancing is only focused while the non-crowdsourced-based associations are not available in the literature. However, freelancing could be crowdsourced or non-crowdsourced, meaning all crowdsourcing models are freelancing models but not vice versa. The challenges associated with the different freelancing models are unique and thus put different constraints on their execution for software development.

In particular, this mapping study highlights the need for more empirical research followed by a solution-based research with strong applicability in the industrial settings. The research must focus, in particular, on the individual development life cycle activities and in general the overall development.

6. Threat to Validity

The requirement for the consistent and complete reporting of validity threats in software engineering research had been reported in [5,6]. However, this paper addresses the threats to four types of validity as identified in [5]. This includes the descriptive validity, theoretical validity, generalizability, and interpretive validity. The overall objective to address the threats is to enhance the validity of the study by minimizing the bias.

However, the researchers ensure the validity for the result reproducibility under the systematic review protocol as planned and executed in this paper and the assumptions made. Assumptions relate to freelancer association models considered by researchers, keywords categories, and their descriptive labels, and so on. However, improper or ambiguous use of terminologies by studied research paper authors may be considered a threat. Details about each type of validity is as given under:

- (a) **Descriptive Validity:** The ability of the research study to accurately describe the gathered information ensures descriptive validity. To handle descriptive validity, the data extraction form was used (Appendix A) to accurately record the information, which was synthesized to populate the conceptual schemas leading to meet the objectives of the study.
- (b) **Theoretical validity:** Theoretical validity is the ability of the researchers to capture through the systematic mapping study the information that they intended to capture [5]. The mistakes made for the selection of the studies, extracting data from them, and finally accurately classifying them results in outcomes that deviates from what researchers wish to investigate. To address this concern, all the researchers participated in the planning and execution of the mapping protocol. Random checks were also made to cross check the information extracted from the execution of the mapping protocol. The process of the formulation and execution of the mapping protocol was arranged as the series of the milestones. Each milestone is checked and agreed once they are reached, before moving ahead towards bigger milestones. For example, before creating a big structure schema for individual research questions, the sub schemas were created and considered as small milestones. All the authors were involved in validating the milestones. The random efforts were also made in the last to recheck all milestones to ensure bias free research to the maximum possible extent. Any disagreements among the researchers were resolved using consensus meetings.
- (c) **Generalizability:** The results of the mapping study are highly generalizable because the mapping study had ensured the coverage of a broad research area for a longer time interval for all types of diverse sized software companies.
- (d) **Interpretive validity:** The ability of the researcher to accurately interpret the gathered data without subjective judgement is called interpretive validity. In other words, the conclusion drawn should be driven by the data gathered from the studies. Analysis of the data by multiple researchers should result in similar interpretation of the data. The joint involvement of all researchers in the formulation and execution of the mapping protocol helped to resolve issues with single subjective judgments. Results were drawn considering the agreements among multiple perspectives of the researchers which helped to ensure interpretive validity.
- (e) **Repeatability** is the ability of the mapping study to be repeated by another researcher which yields similar results. The mapping study is executed using available guidelines and the mapping protocol is specified in this paper to ensure reproducibility. The assumptions made during formulation of the conceptual schemas and assignment of the studied research papers to these categories are also mentioned. Provided the mapping conditions like search strings, inclusion and

exclusion criteria, and interpretation of the assumptions remains the same, the repeatability is likely to prevail.

7. Conclusions and Future Work

The systematic mapping study resulted in identifying the valuable information about the research trends in employing the freelancing team for undertaking the software development activities. The valuable information is concluded by structuring it separately for each research question.

RQ1.Which software development process activities have involved the freelancers?

The distribution of the research efforts is non uniform across the software development life cycle areas. The majority of studies focused on generic software development as a whole rather on individual development activities. This limits the solutions addressing the freelancer adoption across individual software development activities by overcoming the constraints which are specific to the activities. Stated in other words, if the company requires freelancers in the requirement prioritization phase then literature does not provide the solutions to manage the outsourcing to the freelancers. The researchers had focused much on bringing solutions for freelancer involvement for generic software development by addressing the various challenges for the freelancer involvement (as identified in RQ 2), followed by validations in laboratory settings (rather in a real world setting). The wide array of freelancers is available for undertaking a variety of complex tasks but effectively managing them could bring value to the firm and hence to its customers.

RQ2.Which Issues and challenges are associated with the freelancer involvement in the software development process?

Various challenges and issues were reported including Collaboration and Coordination (33%), Developer Recommendation (or selection) (19%), Team Formulation (14%), Task Recommendation (allocation) (14%), Task Decomposition (11%), Privacy and Security (Confidentiality) (11%), Budget Estimation (8%), Recognition (8%), Trust Issues (8%), Market Dynamism (6%), Intellectual Property Issues (6%), Participation of Crowd Worker (6%), and Capacity Utilization (3%). These challenges are interconnected, i.e., each challenge affects other challenges. These interrelationships are usually ignored while designing solutions for freelancer involvement in software development. In other words, solutions addressing one of more challenges may be hindered by the challenges that are not under the consideration of the researchers. A holistic approach is required which provides effective solutions addressing complex interconnected challenges.

RQ3.How does freelancer supported software development research can be classified in terms of research type?

The studies are usually validation-based approaches (72%) (solutions evaluated in laboratory settings). This means that they still lack the evaluation in real world settings before being actually put to use in industrial settings (unless the R&D department of the companies wish to innovate them and adapt to their working context). The literature lacks empirical research, which just account for 25% of total studies analyzed by the researchers. Empirical studies should be undertaken to provide the research community with the research gaps in freelancer driven software development and for the companies to gain access to the lessons learned by other companies.

RQ4.How does the freelancer supported software development research evolved over the time?

The collaboration and coordination, developer recommendation, and task recommendation are still the active challenges as faced by researchers. Since limited studies are reported in the literature and this research area is in the initial phase, there is no sufficient evidence that effective research solutions for freelancer driven software engineering exists today. The studies focused

on emerging areas like motivating freelancers (using gamifications), improving recognition (example using blockchains), reducing technical debt, reaching product market fit in dynamic markets under resource constraints etc. are missing from the literature. Overall, the studies focus on broad challenges.

In future, it is expected that the research will suggest the efficient utilization of the freelancers in software development of various application domains. The research will help to provide evidence and effective solutions to employ freelancer teams in a cost-effective way.

Author Contributions: Conceptualization, V.G.; methodology, V.G., J.M.F.-C. and T.H.; software, V.G., J.M.F.-C. and T.H.; validation, V.G., J.M.F.-C. and T.H.; formal analysis, V.G., J.M.F.-C. and T.H.; investigation, V.G., J.M.F.-C. and T.H.; resources, V.G., J.M.F.-C. and T.H.; data curation, V.G., J.M.F.-C. and T.H.; writing—V.G.; writing—review and editing, V.G., J.M.F.-C. and T.H.; visualization, V.G., J.M.F.-C. and T.H.; supervision, J.M.F.-C. and T.H.; project administration, V.G., J.M.F.-C. and T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Appendix A (Data Extraction Form)

Paper Related Information

- Paper Title
- Paper ID
- Child IDs
- Author details.
- Publication Year.
- Google citations.
- Citations selected (ID).
- Research type.

Software Engineering Areas Related Information

- Software Engineering Activities supported.

Challenges and Issues Related Information

- Challenges and issues reported.

References

1. Gupta, V.; Fernandez-Crehuet, J.M.; Gupta, C.; Hanne, T. Freelancing Models for Fostering Innovation and Problem Solving in Software Startups: An Empirical Comparative Study. *Sustainability* **2020**. (Under Review).
2. Gupta, V.; Fernandez-Crehuet, J.M.; Hanne, T.; Telesko, R. Requirements Engineering in Software Startups: A Systematic Mapping Study. *Appl. Sci.* **2020**, *10*, 6125. [[CrossRef](#)]
3. Gupta, V.; Fernandez-Crehuet, J.M.; Hanne, T. Fostering Product Innovations in Software Startups through Freelancer Supported Requirement Engineering. *Results Eng.* **2020**. Volume 8, 100175. [[CrossRef](#)].
4. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*; BCS Learning & Development Ltd.: Swindon, UK, 2008; Volume 12, pp. 1–10.
5. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [[CrossRef](#)]

6. Petersen, K.; Gencel, C. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In Proceedings of the 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, 23–26 October 2013; IEEE: New York, NY, USA, 2013; pp. 81–89.
7. Dwarakanath, A.; Chintala, U.; Shrikanth, N.C.; Viridi, G.; Kass, A.; Chandran, A.; Paul, S. Crowd build: A methodology for enterprise software development using crowdsourcing. In Proceedings of the 2015 IEEE/ACM 2nd International Workshop on CrowdSourcing in Software Engineering, Florence, Italy, 19 May 2015; IEEE: New York, NY, USA, 2015; pp. 8–14.
8. Bernabé, R.B.; Navia, I.Á.; García-Peñalvo, F.J. Faat: Freelance as a team. In Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, Porto, Portugal, 7–9 October 2015; pp. 687–694.
9. Song, F.; Chen, H.; Fu, Y. An approach to rapid worker discovery in software crowdsourcing. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015), Melbourne, VIC, Australia, 9–11 December 2019; Springer: Berlin, Germany, 2019; pp. 370–382.
10. Murray-Rust, D.; Scekcic, O.; Lin, D. Worker-centric Design for Software Crowdsourcing: Towards Cloud Careers. In *Crowdsourcing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 39–50.
11. Shao, W.; Wang, X.; Jiao, W. A developer recommendation framework in software crowdsourcing development. In *National Software Application Conference*; Springer: Singapore, 2016; pp. 151–164.
12. Abhinav, K.; Dubey, A. Predicting budget for Crowdsourced and freelance software development projects. In Proceedings of the 10th Innovations in Software Engineering Conference, Jaipur, India, 5–7 February 2017; pp. 165–171.
13. Al-bloush, H.; Solemon, B. Intellectual Property Challenges in the Crowdsourced Software Engineering: An Analysis of Crowdsourcing Platforms. In *International Conference of Reliable Information and Communication Technology*; Springer: Berlin, Germany, 2017; pp. 875–884.
14. Chen, Y.; Lee, S.W.; Xie, Y.; Yang, Y.; Lasecki, W.S.; Oney, S. Codeon: On-demand software development assistance. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; pp. 6220–6231.
15. Lee, S.W.; Chen, Y.; Klugman, N.; Gouravajhala, S.R.; Chen, A.; Lasecki, W.S. Exploring coordination models for ad hoc programming teams. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; pp. 2738–2745.
16. Abhinav, K.; Dubey, A.; Jain, S.; Viridi, G.; Kass, A.; Mehta, M. Crowd Advisor: A framework for freelancer assessment in online marketplace. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Buenos Aires, Argentina, 20–28 May 2017; pp. 93–102.
17. Dubey, A.; Singi, K.; Kaulgud, V. Personas and redundancies in crowdsourced testing. In Proceedings of the 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), Buenos Aires Argentina, 22–23 May 2017; IEEE: New York, NY, USA, 2017; pp. 76–80.
18. Chatterjee, A.; Varshney, L.R.; Vishwanath, S. Work capacity of regulated freelance platforms: Fundamental limits and decentralized schemes. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3641–3654. [[CrossRef](#)]
19. Dubey, A.; Abhinav, K.; Viridi, G. A framework to preserve confidentiality in crowdsourced software development. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), Buenos Aires Argentina, 20–28 May 2017; IEEE: New York, NY, USA, 2017; pp. 115–117.
20. Arenas, C.; García, V.H.M.; Otálora, J.E. Crowdworking as a knowledge management mechanism in the elicitation of mission software requirement. In Proceedings of the International Conference on Knowledge Management in Organizations, Zilina, Slovakia, 6–10 August 2018; Springer: Cham, Switzerland, 2018; pp. 288–298.
21. Tsikerdekis, M. Persistent code contribution: A ranking algorithm for code contribution in crowdsourced software. *Empir. Softw. Eng.* **2018**, *23*, 1871–1894. [[CrossRef](#)]
22. Zhang, Z.; Sun, H.; Zhang, H. Developer recommendation for Topcoder through a meta-learning based policy model. *Empir. Softw. Eng.* **2020**, *25*, 859–889. [[CrossRef](#)]
23. Ivan, I.; Budacu, E.; Despa, M.L. Using profiling to assemble an agile collaborative software development team made up of freelancers. *Procedia Comput. Sci.* **2019**, *162*, 562–570. [[CrossRef](#)]

24. Dubey, A.; Abhinav, K.; Taneja, S.; Viridi, G.; Dwarakanath, A.; Kass, A.; Kuriakose, M.S. Dynamics of software development crowdsourcing. In Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Orange County, CA, USA, 2–5 August 2016; IEEE: New York, NY, USA, 2016; pp. 49–58.
25. Long, F.; Yang, C.; Rong, H.G.; Li, J.F. A user-oriented resource scheduling method for improving agile software pattern in cloud environment. *J. Cent. S. Univ.* **2016**, *23*, 2906–2916. [[CrossRef](#)]
26. Machado, L.; Kroll, J.; Marczak, S.; Prikladnicki, R. Breaking collaboration barriers through communication practices in software crowdsourcing. In Proceedings of the 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), Orange County, CA, USA, 2–5 August 2016; IEEE: New York, NY, USA, 2016; pp. 44–48.
27. Sharma, S.; Hasteer, N.; Van-Belle, J.P. An exploratory study on perception of Indian crowd towards crowdsourcing software development. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 901–905.
28. Machado, L.D.S. Empirical Studies about Collaboration in Competitive Software Crowdsourcing. Ph.D. Thesis, Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil, 2018. Available online: http://tede2.pucrs.br/tede2/bitstream/tede/8256/2/LETICIA_SANTOS_MACHADO_TES.pdf (accessed on 27 September 2020).
29. Sari, A.; Alptekin, G.I. A Prize Determination Approach for Crowdsourced Software Development. In Proceedings of the 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 20–22 December 2018; IEEE: New York, NY, USA, 2018; pp. 450–453.
30. Hasteer, N.; Bansal, A.; Murthy, B.K. Barriers to crowdsource software development: Evaluating the intertwined effects. *Int. J. Bus. Inf. Syst.* **2019**, *32*, 127–144.
31. Illahi, I.; Liu, H.; Umer, Q.; Zaidi, S.A.H. An Empirical Study on Competitive Crowdsourced Software Development: Motivating and Inhibiting Factors. *IEEE Access* **2019**, *7*, 62042–62057. [[CrossRef](#)]
32. Karim, M.R.; Yang, Y.; Messinger, D.; Ruhe, G. Learn or Earn? Intelligent Task Recommendation for Competitive Crowdsourced Software Development. In Proceedings of the 51st Hawaii International Conference on System Sciences, Waikoloa Village, HI, USA, 3–6 January 2018; pp. 5604–5613.
33. Johnson, M.; Stevens, P. Confidentiality in the process of (model-driven) software development. In Proceedings of the Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming, New York, NY, USA, 9–12 April 2018; pp. 1–8.
34. Kaur, H.; Gordon, M.; Yang, Y.; Bigham, J.P.; Teevan, J.; Kamar, E.; Lasecki, W.S. Crowdmask: Using crowds to preserve privacy in crowd-powered systems via progressive filtering. In Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing, Québec City, QC, Canada, 23–26 October 2017.
35. Lee, S.W.; Krosnick, R.; Park, S.Y.; Keelean, B.; Vaidya, S.; O’Keefe, S.D.; Lasecki, W.S. Exploring real-time collaboration in crowd-powered systems through a UI design tool. In *Proceedings of the ACM on Human-Computer Interaction*; ACM: New York, NY, USA, 2018; pp. 1–23.
36. Aghayi, E.; LaToza, T.D.; Surendra, P.; Abolghasemi, S. Implementing Microservices through Microtasks. *arXiv* **2019**, arXiv:1903.01977.
37. Binzagr, F.; Medjahed, B. Crowdmashup: Recommending crowdsourcing teams for mashup development. In Proceedings of the International Conference on Service-Oriented Computing, Hangzhou, China, 12–15 November 2018; Springer: Cham, Switzerland; pp. 679–693.
38. Li, S.; Chen, H.; Shu, Y.; Hu, F. The verification technique of complex tasks’ nonfunctional specifications in software crowdsourcing. In Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), Changchun, China, 10–11 December 2016; IEEE: New York, NY, USA, 2016; pp. 166–170.
39. Shu, Y.; Chen, H.; Li, S.; Hu, F. The verification approach to complex tasks’ functional specification in software crowdsourcing. In Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), Changchun, China, 10–11 December 2016; IEEE: New York, NY, USA, 2016; pp. 171–176.
40. Karim, M. Data Analytics for Optimized Matching in Software Development. Available online: <https://prism.ucalgary.ca/handle/11023/4254> (accessed on 3 March 2020).
41. Yu, D.; Zhou, Z.; Wang, Y. Crowdsourcing Software Task Assignment Method for Collaborative Development. *IEEE Access* **2019**, *7*, 35743–35754. [[CrossRef](#)]

42. Yu, X.; He, Y.; Fu, Y.; Xin, Y.; Du, J.; Ni, W. Cross-Domain Developer Recommendation Algorithm Based on Feature Matching. In Proceedings of the CCF Conference on Computer Supported Cooperative Work and Social Computing, Kunming, China, 16–18 August 2019; pp. 443–457.
43. Wieringa, R.; Maiden, N.; Mead, N.; Rolland, C. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.* **2006**, *11*, 102–107. [[CrossRef](#)]
44. Paternoster, N.; Giardino, C.; Unterkalmsteiner, M.; Gorschek, T.; Abrahamsson, P. Software development in startup companies: A systematic mapping study. *Inf. Softw. Technol.* **2014**, *56*, 1200–1218. [[CrossRef](#)]
45. Giardino, C.; Wang, X.; Abrahamsson, P. Why Early-Stage Software Startups Fail: A Behavioral Framework. In Proceedings of the International Conference of Software Business (ICSOB 2014), Paphos, Cyprus, 16–18 June 2014; pp. 27–41.
46. Cantamessa, M.; Gatteschi, V.; Perboli, G.; Rosano, M. Startups' Roads to Failure. *Sustainability* **2018**, *10*, 2346. [[CrossRef](#)]
47. Tripathi, N.; Oivo, M.; Liukkunen, K.; Markkula, J. Startup ecosystem effect on minimum viable product development in software startups. *Inf. Softw. Technol.* **2019**, *114*, 77–91. [[CrossRef](#)]
48. Crowne, M. Why software product startups fail and what to do about it. Evolution of software product development in startup companies. In Proceedings of the IEEE International Engineering Management Conference, Cambridge, UK; 2002; Volume 1, pp. 338–343. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).