

The Value of Crowdsourcing for Complex Problems: Comparative Evidence from Software Developed By the Crowd And Professionals

Abhishek Tripathi* and Deepak Khazanchi

College of Information Science and Technology, University of Nebraska at Omaha

Received: September 13, 2016; Accepted: September 18, 2016; Published: October 07, 2016

***Corresponding author:** Abhishek Tripathi, PhD Candidate, College of Information Science and Technology, University of Nebraska at Omaha, College of IS&T, 1110, 67th St, Omaha, NE, 68106, Tel: 402-955-9222; E-mail: atripathi@unomaha.edu

Abstract

Crowdsourcing is a problem solving model. In the context of complex problems, conventional theory suggests that solving complex problems is a province of professionals, that is, people with sufficient knowledge about the domain. Prior literature has indicated that the crowd, in addition to professionals, is also a great source for solving problems such as product innovation and idea generation. However, this assumption has yet to be tested. Adopting a quasi-experimental approach, this study uses a two-phase process to investigate this question. In the first phase we compare the development of a software by the crowd and professionals. In the second phase we evaluate the software developed by the crowdsourcing business model and professionals in terms of key perceived quality dimensions assessed by users of the systems. Quality is measured in terms of pragmatic quality, hedonic quality stimulation, and hedonic quality identification. Our study results suggest that there is a statistically significant difference between the software developed by a crowdsourcing business model and professionals in terms of hedonic quality stimulation and hedonic quality identification but there is no difference in terms of pragmatic quality. This research offers a first assessment of whether a crowdsourcing business model can be used to develop software with better user experience than professionally-developed software.

Keywords: Crowdsourcing; Pragmatic Quality; Hedonic Quality; Complex-Problem; Software Development

Introduction

Increasingly, organizations are tapping the wisdom of crowds to solve “complex” problems [7, 13, 27, 49]. This phenomenon is called crowdsourcing, a term coined by Howe [26]. The wisdom of crowds refers to the accumulation of information in groups that can be processed for collective wisdom, which is often considered to be better than professional wisdom. Surowiecki [49] has suggested that the collective wisdom of a group of less skilled individuals is more informative and creative than that of a few specialized people. The core of crowdsourcing ideas originated from the notion that the wisdom of crowds may be better than solutions created by professionals or small groups. Various definitions of crowdsourcing have been presented.

Crowdsourcing is “the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call” [10, 27, 28, 53]. Crowdsourcing has also been described as a problem-solving model [3, 9, 14, 15]; gaining input from many unknown and unconnected contributors [25]; and distributed production models that ask for contributions via open calls from an undefined large network of people [3, 54]. The common attribute of crowdsourcing in all these definitions is that it is a collaborative effort enabled by people-centric technology. Crowdsourcing business models benefit organizations by providing cheap labor and by tapping geographically dispersed crowds.

Critics of the wisdom of crowds suggest that collective wisdom may be only useful for simple problems, and may be difficult to use for complex problems such as software development. As the practice of problem solving with crowdsourcing becomes increasingly common, it is essential to identify whether the wisdom of crowds can be applied to solve complex problems.

There are two alternative streams of research that focus on the legitimacy of the crowd’s/customer’s complex problem-solving abilities. One stream suggests that crowds are mostly novice and do not have sufficient domain expertise to participate in and solve complex problems such as product innovation and development [33, 43, 5]. The other stream argues that “innovation is being democratized” [51], meaning that crowds/customers of product and services know about their requirements, are able to contribute toward the development of a product, and can solve complex problems [9, 32, 51].

Research on Complex Problem Solving (CPS) has revealed a wide variety of thoughts and insights about the characteristics and operationalization of complex problems [17]. The research community is still debating which definition should be widely accepted by the scientific community, what is “complex” in CPS, and how to evaluate the complexity of problems [42]. In group environments, CPS addresses challenges such as coordination of tasks, lack of domain expertise by community members, lack

of motivation, and sustainability of the community [31]. Such difficulties with CPS are never attempted in a crowdsourcing domain. Although the crowdsourcing business model supports creativity and problem solving [32], use of crowdsourcing for software development is different from general crowdsourcing [55]. These research gaps suggest that research on complex problem solving in crowdsourcing environments is valuable in addressing the question of whether the wisdom of crowds produces quality solutions for complex problems such as software development.

Lanier argues that crowd wisdom is inadequate to solve a creative or innovative problem; collective wisdom is useful when a problem is inadequately defined, a solution is simple, and the collective is aggregated by quality control which depends upon individuals to a high degree [33]. Other researchers suggest that crowdsourcing can be used for solving complex problems [8, 20, 30]. It should be noted here that software development is also a complex and creative activity. The production of a tangible product (software) may require various processes such as requirements analysis, design, coding, and testing [55].

To address the veracity of the two alternative claims about crowdsourcing, we propose to address the following specific research question: Does software developed by the crowdsourcing business model provide the same or better Perceived Quality (PQ) than software developed by professionals?

Theoretical foundation and Conceptual model

Organizations increasingly tap the wisdom of crowds to solve their problems [7, 13, 27, 49]. This phenomenon is called crowdsourcing, a neologism (a compound contraction of 'crowd' and 'outsourcing') coined by Howe [26]. The term crowdsourcing, like any IS fashion as suggested by Baskerville & Myers [2], is quickly gaining attention from academics and practitioners' circles. As an emerging research topic, crowdsourcing research roots span various disciplines such as economics, psychology, organizational behavior, management, and information systems, as well as in diverse directions [40]. Consequently, our research study builds on relevant research in the area of complex problem solving, user experience, and crowdsourced software development.

Complex Problems

A problem becomes complex when its solution requires responses that deviate from common solutions or from previously learned ones [36]. In the case of a complex problem, the problem is known but the solution is either unknown or there may be multiple solutions. The goal is not yet clear, but upon agreement, the complex problem may transform to a simple problem. Complex problems differ from simple problems in the availability of information about the problem, the precision of goal definition, the complexity of a problem in terms of number of variables, the degree of connectivity among variables, the type of functional relationship, time dependencies over the course of achieving the goal to solve a problem, and the richness of the problem's semantic embedding [47]. For example, an organization may

want strategic and competitive advantages. The problem is clear if they can define what is meant by "strategic and competitive advantages," but understanding how to solve the problem is far from clear. Deriving solutions to complex problems often requires "organizational learning" [44].

A popular use of crowdsourcing is to perform various micro tasks (routine tasks) which are easier to perform by the humans but rather difficult for machine [16]. Micro tasks are those that are executed in minutes and repetitive in nature, e.g., identifying a person in a photo, phone number verification, or writing reviews. In these types of problems, the solution is known and the objective is clear. Though organizations are also using crowdsourcing model to solve complex problems such as software development, current research in the crowdsourcing and complex-problem solving field lacks a systematic experimental investigation [34]. Leicht et al performed a structured literature review based on top IS and software engineering journals and conferences to review the current state of crowdsourced software development research and concluded that research on crowdsourced software development is still in a nascent phase. They reported that almost 60% of the research in crowdsourcing software development is from a systems perspective, about 40% of the research is on crowdsourcing applications in software development, and only one paper dealt with user perspectives.

Crowdsourced Software Development

Software development is considered one of the most complex, challenging, and creative processes. Software development involves various stakeholders, requirements analysis, design, and architecture, coding, and testing [55]. In addition, the software development life cycle span is becoming shorter, while software complexity is increasing and budgets are stagnant [35]. Software engineering has a substantial number of techniques and tools, and yet the field is still seeking new technologies and techniques as it faces new challenges every year [55]. One promising solution to developing software is by the crowdsourcing business model. IT industry leaders such as Fujitsu-Siemens [19] and SAP [4] have already leveraged the crowdsourcing business model for innovation management [35]. Lakhani et al report on a crowdsourced programming contest in which about 75% of the solutions to solve an immunogenomic problem outperformed the industry standard, at a total cost of \$6,000 [34]. To support crowdsourced software development, various commercial crowdsourcing platforms have emerged. These platforms use different types of open call formats, such as online competition; on-demand matching, in which the workers are chosen by registrants; and online bid, where developers bid for tasks before starting to work [37]. The World Quality Report (2014), the premier report for software testing practices, indicates that more than half of the surveyed organizations already employ crowdsourcing for their software testing process [35].

Although the crowdsourcing business model supports creativity and problem solving [32], use of crowdsourcing for software development is different from general crowdsourcing [55]. According to Wu et al software crowdsourcing needs

to support the rigorous engineering disciplines of software development; stimulate creativity in software development tasks through the wisdom of the crowd; address the psychological issues of crowdsourcing such as competition, open, sharing, collaboration, and learning; address the financial aspects and recognition for various stakeholders; ensure the quality of the software product; and address liability issues in case of failure [55]. A key feature of software crowdsourcing is that it is a contest-based crowdsourcing model. In a contest-based crowdsourcing model, a problem owner who faces an innovation-related problem posts this problem to a large independent crowd and then provides a reward to the agent who produces the best solution [50]. While competitions promote creativity and support quality software development, they may reduce the massive competitions [55]. A contest-based crowdsourcing model also promotes the min-maxing nature of game playing by different people with different roles [50].

This research simplifies and adapts crowdsourcing in a software development context, in particular a website development project. Understanding and managing of website structures is a complex task [11]. Like any other software development effort, website development processes can involve requirements analysis, design, and implementation, which make this also a complex, challenging, and creative process [55].

IT/IS Professionals

Human factor analysis is one of the most important areas in the software engineering [6]. According to Boehm human factor is the second most important factor after product size to determine the effort required for development of software [6]. We know from previous research that IS professional team's plays an important role for effective and efficient development of Information Systems [45]. Siau, Tan, & Sheng identified fifty-nine unique characteristics classified into eight dimensions that determine the important characteristics of software development team members [45]. Attitude/motivation, knowledge, interpersonal/communication skills, and working/ cognitive ability are the most important characteristics.

User Experience

In this research, we define perceived quality from the user experience perspective. Design and development challenges have shifted from providing efficient, reliable, secured, usable functionalities with a competitive price toward providing users with pleasurable experiences. User experience should therefore exceed expectations and support fulfillment of human needs, such as identification, past memory evocation, and stimulation (proliferation of knowledge and development of skills) through a product [38]. Consequently, good functionality and usability have become axiomatic features and are not enough when designing a successful product [21, 38, 39]. A valid and reliable measure of UX could be useful in the evaluation of crowdsourced software.

Hassenzahl and Tractinsky define user experience as a "consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of

the designed system (complexity, purpose, usability, functionality, etc.) and the environment within which the interaction occurs (organizational/social setting, meaningfulness of the activity)" [22]. According to Alben "all the aspects of how people use an interactive product (feeling, understanding, sensations) fit to the context" [1]. According to Forlizzi and Batterbee "emotion is at the heart of any human experience and an essential component of user-product interactions and user experience" [18]. In fact, "UX is a momentary, primarily evaluative feeling (good-bad) while interacting with a product or service" [23]. Various definitions and concepts of UX have been proposed, but the common theme in all the definitions is that UX is an outcome of interactions between a user and a product in the form of the user's perceptions and emotions.

Although these two premises are the same, researchers have used two different concepts to define user experience. One group suggests uncovering the objective in the subjective, and developed a model-based approach (a reductionist approach). The other group suggested that UX is very subjective in nature and should be inherent to the concept of UX, and thus developed a framework of thought (phenomenological approach).

Hassenzahl presented a hedonic/pragmatic model of user experience. This model suggested that users first perceive product features, such as content, presentation, functionality, and presentation style to view a personal version of the apparent product character (pragmatic attributes and hedonic attributes) [21]. This apparent product character leads to consequences, such as a product's appeal (good-bad), its emotional consequences (satisfaction and pleasure), and its behavioral consequences (increased usage). The consequences are not always the same and may be moderated by specific usage situations.

Pragmatic quality refers to a product's perceived ability to support the fulfillment of functions or intended tasks. Hassenzahl refers to these functions or tasks as "do goals" or instrumental goals (in which software is performing intended tasks) [23]. Pragmatic quality focuses on the utility and usability of products in terms of intended tasks. Hedonic quality refers to individual psychological well-being. Hedonic quality is mostly associated with pleasure. According to Hassenzahl [23], hedonic quality refers to a product's perceived quality to achieve the "be goals," such as being "competent" related to others. Hassenzahl emphasized that good UX stems from the fulfillment of the human needs for autonomy, competency, stimulation (self-oriented), relatedness, and popularity (other-oriented) through interacting with a product or service [23].

In summary, we propose a conceptual model adapted from Hassenzahl to address the research question in this paper (refer Figure 1) [21]. A conceptual model is a graphical lens for communicating the specification of things, events, or processes [52]. First, drawing on previous theoretical studies, this study proposes that the development approach (by crowdsourcing or IT professionals) has an impact on perceived quality, which is moderated by the complexity of the problem. Perceived quality of the product is measured in terms of pragmatic quality, hedonic quality stimulation, and hedonic quality identification.

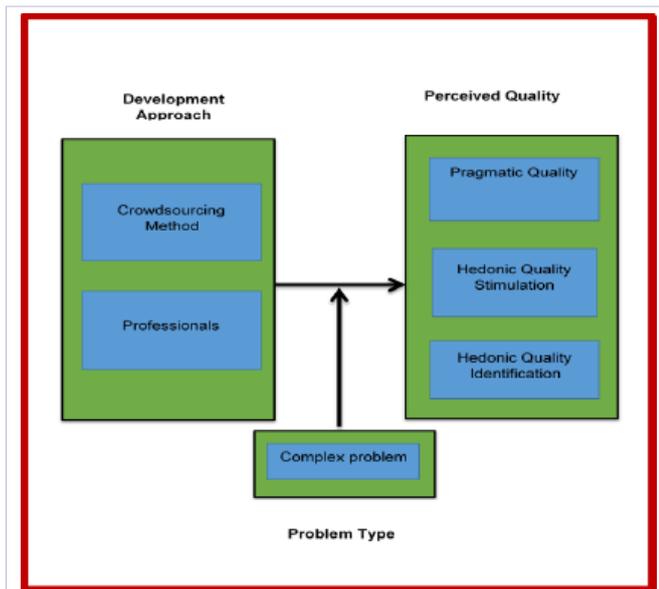


Figure 1: Conceptual model.

Research Method

To empirically test the research question, a quasi-experimental research design using a survey questionnaire is performed. A quasi-experimental design is used in situations where it is not possible to exercise a full experimental control, and where there is a lack of the full control of a true experimental design or randomized controlled trials [46]. In this study, the random assignment of subjects to treatments (crowds and professionals) is not feasible. An experimental design is useful to explore the decision performance and characteristics of the information system developed by the crowd and professionals [29]. The survey questionnaire is used to operationalize the various outcome constructs and data collected is used to compare and contrast the results.

We use a two-phase process to investigate the research question. The first phase used the development of a software by the crowd and professionals. In the second phase we evaluate the software developed by the crowdsourcing business model and professionals in terms of key perceived quality dimensions, including pragmatic quality, hedonic quality stimulation, and hedonic quality identification to compare the quality of software developed by crowds and professionals.

Participants

The participants in this study consisted of a crowd of students and a professional web development community at University of Nebraska at Omaha (UNO). The students at UNO in this research form the crowd; and the professional group is represented by the UNO's web development community called the Attic. Attic is a group of undergraduate and graduate students who have or are applying skills in web development and multimedia presentation technologies under professional supervision to commercial projects. The Attic group has successfully completed more than

twelve projects of considerable complexity ranging from complex website development to mobile app development.

Tasks

Previous studies have identified task as an important variable and have used it as a lens to study CPS. Problem-solving is mostly a task-centered activity and some researchers believe that "tasks" and "problems" are synonymous to each other [42]. In this study the research task is to design and develop a website to promote the Alumni Association, and UNO in general. The UNO Alumni Association wanted to establish a website for the UNO Alumni members to submit images of themselves with a UNO flag wherever they are in the world. This website would allow users to upload a picture which would then be approved by a content administrator to finalize the submission. The pictures would then be shown on a map. We crowdsourced this problem to the UNO student community and also had this problem solved by the Attic professionals.

Measurement

In this study we measured the perceived quality of the software developed by the crowd and by the professionals. The variables used in this study are listed below:

- i. Independent Variable: development approach
- ii. Dependent Variables: pragmatic quality, hedonic quality stimulation, and hedonic quality identification

Upon the development of the software the students at UNO, who were not part of the crowd that developed the software, were asked to participate in a survey designed to measure the perceived quality of the systems. We used existing measures to evaluate the pragmatic quality, the hedonic quality stimulation, and the hedonic quality identification. In order to evaluate the perceived quality, we used the survey questionnaire developed by Hassenzahl [24]. The survey instrument is composed of a 7 point Likert-scale items designed specifically to measure the perceived quality of the software product.

Empirical Analysis and Results

For the crowdsourcing task, we received two partial solutions and one full solution. Out of the three, one of them was a prototype. The prototype solution contained only static features and only few of the features were incorporated in the solution. The possible explanation for partial solution development is either the absence of the motivation such as a reward for participation or the lack of the specific guidelines in terms of the expectations in the final solution.

Table 1 summarizes the descriptive statistics results while Table 2 and Table 3 shows the results of Multivariate and Univariate analysis.

A total of 66 students from UNO participated in the survey. The participants were either undergraduates or graduate belonging to different departments at UNO. Table 1 shows that the average response rate for the hedonic quality identification (HQL) as 5.2 and the hedonic quality stimulation (HQS) as 4.6, which is more

Table 1: Descriptive Statistics

Descriptive									
		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
HQIL	1	66.0000	4.2778	1.3491	.1661	3.9461	4.6094	1.2222	6.8889
	2	66.0000	5.2037	.7530	.0927	5.0186	5.3888	3.2222	6.6667
	Total	132.0000	4.7407	1.1834	.1030	4.5370	4.9445	1.2222	6.8889
PQL	1	66.0000	4.9515	1.0949	.1348	4.6824	5.2207	2.0000	7.0000
	2	66.0000	5.2212	.7885	.0971	5.0274	5.4150	3.0000	6.8000
	Total	132.0000	5.0864	.9600	.0836	4.9211	5.2517	2.0000	7.0000
HQSL	1	66.0000	4.0354	1.2758	0.1570	3.7217	4.3490	1.3333	7.0000
	2	66.0000	4.6465	1.0984	0.1352	4.3765	4.9165	1.3333	6.6667
	Total	132.0000	4.3409	1.2249	0.1066	4.1300	4.5518	1.3333	7.0000

Table 2: Multivariate Analysis

Effect		Value	F	Hypothesis df	Error df	Sig.
Intercept	Pillai's Trace	0.977	1840.310 ^a	3.000	128.000	.000
	Wilks' Lambda	0.023	1840.310 ^a	3.000	128.000	.000
	Hotelling's Trace	43.132	1840.310 ^a	3.000	128.000	.000
	Roy's Largest Root	43.132	1840.310 ^a	3.000	128.000	.000
Cat	Pillai's Trace	0.157	7.960 ^a	3.000	128.000	.000
	Wilks' Lambda	0.843	7.960 ^a	3.000	128.000	.000
	Hotelling's Trace	0.187	7.960 ^a	3.000	128.000	.000
	Roy's Largest Root	0.187	7.960 ^a	3.000	128.000	.000

Multivariate Tests^b

a. Exact statistic

b. Design: Intercept + Cat

Table 3: Univariate Analysis

Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	HQIL	28.29 ^a	1	28.29	23.705	.000
	PQL	2.4 ^b	1	2.4	2.637	.107
	HQSL	12.32 ^c	1	12.32	8.697	.004
Cat	HQIL	28.29	1	28.29	23.705	.000
	PQL	2.4	1	2.4	2.637	.107
	HQSL	12.32	1	12.32	8.697	.004
Error	HQIL	155.16	130	1.19		
	PQL	118.34	130	0.91		
	HQSL	184.22	130	1.42		
	HQSL	2,683.89	132			
Corrected Total	HQIL	183.45	131			
	PQL	120.74	131			
	HQSL	196.55	131			

Tests of Between-Subjects Effects

a. R Squared = .154 (Adjusted R Squared = .148)

b. R Squared = .020 (Adjusted R Squared = .012)

c. R Squared = .063 (Adjusted R Squared = .055)

than the average response rate for the crowdsourcing model-based approach which recorded as 4.27 and 4.03, respectively. For the pragmatic quality (PQL), the average response rate observed was 5.2 for the professional-development approach

and 4.95 for the crowdsourcing model-based approach.

In order to compare the perceived quality of the website developed by the crowdsourcing model against the one developed by the professionals, we conducted a multivariate test (MANOVA) because there were three dependent variables. The three dependent variables were namely HQIL, HQSL and PQL. The alternative hypothesis in our study is that the development approach (crowdsourcing method and professionals' method of software development) has an effect on all the three dependent variable.

Referring to Table 2 we see that the p-value is very close to zero, which is less than the level of significance (alpha); therefore, the development approach (crowdsourcing method and professionals' method of software development) has an effect on all the three dependent variables namely HQIL, HQSL and PQL.

The MANOVA test also provides the ANOVA table to test the mean difference between the three dependent variables. Table 3 shows that the p-value for the HQSL and HQIL is close to zero, suggesting that the development approach has an effect on HQSL and HQIL. For PQL, the p-value is 0.107 and is greater than the level of significance, which suggests that PQL has no effect on the development approach.

Discussion / Suggestions

Design and development of software is an important and complex process. Due to the evolution of user-centric technologies the tasks previously solved only by the professionals can now be solved by the wisdom of the crowd. While various organizations such as Topcoder and Innocentive have achieved considerable maturity, solving complex problems such as software design and development by a crowdsourcing model is still a topic of debate among academics and practitioners.

Theoretical and Practical Implications

Our research has shown that the development approach (professionals or crowdsourced model) has a statistically significant effect on the overall perceived quality of solutions (software) developed. However, we found no significant difference in terms of pragmatic quality aspect of perceived quality. This suggests that focus on the products (software developed by professionals and crowdsourced model) in terms of its utility and usability in relation to potential tasks is same for both the methods while there is a significant difference in terms of the hedonic quality which refers to the “general human needs such as novelty and change, personal growth, self-expression and relatedness” beyond the utility and usability aspect of a product [23]. We also determined the fact that the software developed by the professionals has better hedonic attributes than the crowdsourcing model.

Although various scholars have studied the crowdsourcing research phenomenon from the lens of software development, more research is warranted from the user perspective [35]. To the best of our knowledge, this study is the only experimental study on crowdsourcing and complex problem solving.

Primarily, the user experience model has never been used in the IS discipline and in particular the crowdsourcing domain. We go beyond the existing studies in crowdsourced software development by offering a deeper understanding of the perceived quality not only in terms of the utility and the usability of the software but also in terms of the general human needs. Existing studies on crowdsourced software development have mostly addressed the phenomenon based on couple of crowdsourcing organizations such as Topcoder and Innocentive [34]. In this study we have answered the call of various researchers who have emphasized the need for a more detailed study on the crowdsourcing and complex problem, and on the crowdsourced software development [33, 34].

Secondly, a systematic literature survey based on the top IS conferences and journals reveals that the theoretical research that motivates the design of crowdsourcing related artifact is least common and there is still very little research on traditionally popular topics such as adoption and complex problem solving in the crowdsourcing context. The conceptual model provided in this study should provide a solid starting point for continuing the crowdsourcing research by extending our knowledge of traditional work arrangements of organizations and crowdsourcing based model to solve complex problems. Based

on the results of our experiment, we can argue for supporting the notion that at least instrumental goal can be achieved by the crowdsourcing based model i.e. crowdsourced software is able to fulfill the behavioral goals.

Conclusion and Future Work

Our experimental study appears to be a necessary first step for better understanding the phenomenon of crowdsourcing and complex problem solving. Our scope was limited to only one experimental study, but the results still have validity. We provide a conceptual model for better understanding the crowdsourcing and its complex problem solving abilities. Although limited in scope only to the college students based crowd simulation, this experiment provides a solid foundation from which one can build future research. Given that this is the first step, we acknowledge that additional study is needed to verify and refine our conclusions and findings. First, more complex problems are needed to consider for this experiment. We recognize that our experiment is limited in scope, and are currently pursuing an extension to conduct a field experiment that includes professional crowdsourcing organizations such as Topcoder and Innocentive to simulate a diverse crowd

Declaration

This research project is approved by the IRB and the IRB approval number is 737-13-EX.

References

1. Alben L. Defining the Criteria for Effective Interaction Design. 1996;3(3):11-15.
2. Baskerville R L, Myers M D. Fashion Waves in Information Systems Research and Practice. MISQ. 2009;33(4):647-662.
3. Baumel U, Georgi S, Ickler H. and Jung R. Design of New Business Models for Service Integrators by Creating Information-Driven Value Webs Based on Customers' Collective Intelligence. Proceedings of the 42nd HICSS. 2009:1-10.
4. Blohm I, Riedl C, Leimeister J M, Krcmar H. Idea Evaluation Mechanism for Collective Intelligence in Open Innovation Communities: Do Traders Outperform Raters?. Proceedings of the 2011 ICIS. 2011.
5. Bidault F, Cummings T. Innovating Through Alliances: Expectations and Limitations. R&D Management. 1994;24(2):33-45.
6. Boehm B W. Software Engineering Economics. Englewood Cliffs (NJ): Prentice-hall. 1981.
7. Bonabeau E. Decisions 2 0: The Power of Collective Intelligence. MIT Sloan Management Review. 2009;50(2):45-52.
8. Brabham D C. Moving the crowd at Threadless: Motivations for Participation in a Crowdsourcing Application. Information, Communication & Society. 2010;13(8):1122-1145.
9. Brabham D C. Crowd Sourcing the Public Participation Process for Planning Projects. Planning Theory. 2009;8(3):242-262.
10. Brandel M. Crowdsourcing: Are you ready to ask the world for answers? Computerworld. 2008;42(10):24-26.
11. Coda F, Ghezzi C, Vigna G and Garzotto F. Towards a Software Engineering Approach to Web Site Development. In Software Specification and Design. Proceedings. Ninth International Workshop on IEEE. 1998:8-17.

12. Colomo-Palacios R, Tovar-Caro E, García-Crespo Á and Gómez-Berbis J M. Identifying Technical Competences of IT Professionals. In Professional Advancements and Management Trends in the IT Sector. 2012;1:1-14.
13. Datta R. Collective Intelligence: Tapping into the Wisdom of Crowds. KM Review. 2008;11(3):3.
14. Davis J and Lin H. Web 3.0 and Crowdsourcing. In Proceedings of the 2011 AMCIS. 2011.
15. Doan A, Ramakrishnan R and Halevy A Y. Crowdsourcing Systems on the World-Wide Web. CACM. 2011;54(4):86-96.
16. Erickson L B, Petrick I, Trauth E M. Hanging with the Right Crowd: Matching Crowdsourcing Need to Crowd Characteristics. In: Proceedings of the Eighteenth Americas Conference on Information Systems. 2012.
17. Fischer A, Greiff S and Funke J. The Process of Solving Complex Problems. Journal of Problem Solving. 2011;4(1):19-42.
18. Forlizzi J, Battarbee K. Understanding Experience in Interactive Systems. In Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques ACM. 2004: 261-268.
19. Füller J, Mühlbacher H, Matzler K and Jawecki G. Consumer Empowerment Through Internet-Based Co-Creation. JMIS. 2009;26(3):71-102.
20. Guinan E, Boudreau K J and Lakhani K R. Experiments in Open Innovation at Harvard Medical School. MIT Sloan Management Review. 2013;54(3):45-52.
21. Hassenzahl M. The thing and I: understanding the relationship between user and product. In Funology Springer Netherlands. 2003:31-42.
22. Hassenzahl M, Tractinsky N. User Experience-a Research Agenda. Behavior & information technology. 2006;25(2):91-97.
23. Hassenzahl M. User experience (UX): Towards an Experiential Perspective on Product Quality. In Proceedings of the 20th International Conference of the Association Francophone d'Interaction Homme-Machine ACM. 2008.
24. Hassenzahl M, Eckoldt K, Diefenbach S, Laschke M, Len E and Kim J. Designing Moments of Meaning and Pleasure. Experience Design and Happiness. International Journal of Design. 2013;7(3).
25. Haythornthwaite C. Crowds and Communities: Light and Heavyweight Models of Peer Production. In Proceedings of the 42nd HICSS. 2009.
26. Howe J. The Rise of Crowdsourcing. Wired. 2006;14(6).
27. Howe J. Crowdsourcing, Why the Power of the Crowd is Driving the Future of Business. NY: Crown Business. 2008.
28. Huysman M and Wulf V. IT to Support Knowledge Sharing in Communities, towards a Social Capital Analysis. JIT. 2006;21(1):40-51.
29. Jarvenpaa S L, Dickson G W and DeSanctis G. Methodological Issues in Experimental IS Research: Experiences and Recommendations. MISQ. 1985;9(2):141-156.
30. Jeppesen L B, Lakhani K R. Marginality and Problem-Solving Effectiveness in Broadcast Search. Organization Science. 2010;21(5):1016-1033.
31. Kittur A, Smus B, Khamkar S and Kraut R E. CrowdForge: Crowdsourcing Complex Work. In Proceedings of 2011 ACM Symposium on User Interface Software and Technology. 2011:43-52.
32. Kittur A. Crowdsourcing, Collaboration and Creativity. ACM Crossroads. 2010;17(2):22-26.
33. Lanier J. You are Not a Gadget. NY: Random House Digital. 2010.
34. Lakhani K R, Boudreau K J, Loh P R, et al. Prize-Based Contests can Provide Solutions to Computational Biology Problems. Nature biotechnology. 2013;31(2):108-111.
35. Leicht N, Durward D, Blohm I and Leimeister J M. Crowdsourcing in Software Development: A State-of-the-Art Analysis. 28th Bled eConference. 2015.
36. Maier N R. Problem Solving and Creativity in Individuals and Groups. CA: Brooks/Cole Publishing Co. 1970.
37. Mao K, Capra L, Harman M and Jia Y. A Survey of the Use of Crowdsourcing in Software Engineering. RN. 2015.
38. Olsson, Thomas. User Expectations and Experiences of Mobile Augmented Reality Services. Tampereen teknillinen yliopisto. 2012.
39. Oppelaar E R, Hennipman E J, van der Veer G C. Experience Design for Dummies. In Proceedings of 15th European Conference on Cognitive Ergonomics: the Ergonomics of Cool interaction. 2008.
40. Pedersen J, Kocsis D, Tripathi A, et al. Conceptual Foundations of Crowdsourcing: A Review of IS Research. In Proceedings of the 46th HICSS. 2013.
41. Poetz M K and Schreier M. The Value of Crowdsourcing: Can Users Really Compete With Professionals in Generating New Product Ideas? Journal of Product Innovation Management. 2012;29(2):245-256.
42. Quesada J, Kintsch W and Gomez E. Complex Problem-Solving: a Field in Search of a Definition? Theoretical Issues in Ergonomics Science. 2005;6(1):5-33.
43. Schrader S and Gopfert J. Structuring Manufacturer-Supplier Interaction in New Product Development Teams: an Empirical Analysis. Elsevier Science. 1998.
44. Senge P M. The Fifth Discipline: The Art and Practice of the Learning Organization. New York: Doubleday. 1990.
45. Siau K, Tan X and Sheng H. Important Characteristics of Software Development Team Members: An Empirical Investigation Using Repertory Grid. Information Systems Journal. 2010;20(6):563-580.
46. Sproull N L. Handbook of Research Methods: A Guide For Practitioners and Students in the Social Sciences. Scarecrow press. 2002.
47. Sternberg R J and Frensch P A. Complex Problem Solving: Principles and Mechanisms. Mahwah, NJ: Lawrence Erlbaum Associates, Inc. 1991.
48. Ren J. Who's More Creative, Experts or the Crowd? In Proceedings of the 2011 AMCIS. 2011.
49. Surowiecki J. The Wisdom of the Crowds. NY: Anchor Books. 2005.
50. Terwiesch C and Xu Y. Innovation Contests, Open Innovation and Multiagent Problem Solving. Management science. 2008;54(9):1529-1543.
51. Von Hippel E A. Open Source Projects as Horizontal Innovation Networks-By and for Users. MIT Sloan School of Management. 2002.
52. Wand Y, Storey V C and Weber R. An Ontological Analysis of the Relationship Construct in Conceptual Modeling. ACM Transactions on Database Systems (TODS). 1999;24(4):494-528.
53. Whelan E. Exploring Knowledge Exchange in Electronic Networks of Practice. JIT. 2007;22(1):5-12.
54. Wiggins A and Crowston K. From Conservation to Crowdsourcing: A Typology of Citizen Science. In Proceedings of the 44th HICSS. 2011:1-10.
55. Wu W, Tsai W T and Li W. Creative Software Crowdsourcing: From Components and Algorithm Development to Project Concept Formations. International Journal of Creative Computing. 2013;1(1):57-91.