

Crowdsourcing Software Development: Concept, Benefits and Adoption

Asiegbu Baldwin C.¹, Oluigbo Ikenna V.*², Ajakwe Simeon O.³, Onyike Gerald O.⁴

¹Department of Information Management Technology, Federal University of Technology Owerri, Owerri, Nigeria

²Department of Information Management Technology, Federal University of Technology Owerri, Owerri, Nigeria

³Department of Information Management Technology, Federal University of Technology Owerri, Owerri, Nigeria

⁴Department of Computer Science, Federal Polytechnic Nekede, Nekede, Nigeria

*Corresponding Author's Email: ikenna.oluigbo@futo.edu.ng, Tel: +2348052891410

Available online at: www.isroset.org

Received 26th Apr 2017, Revised 04th May 2017, Accepted 28th May 2017, Online 30th Jun 2017

Abstract: Traditional outsourcing – the contracting out of services and technology to a vendor firm, is pretty much a household name. Overtime, traditional software development outsourcing has been laced with certain problems; ranging from overreliance on a single vendor outsourcing firm and over budgeting, to financial loss incurred due to delay in software project delivery. It is therefore imperative to consider alternative outsourcing means. Crowdsourcing is a form of outsourcing that is emerging and evolving. Crowdsourcing, an emerging online business model, harnesses the creative solutions of a geographically displaced network of individuals through what amounts to an open call for proposals. Crowdsourcing software development entails contracting the development of software to an online community of skilled developers; the aim being to produce reliable, effective, high quality and inexpensive products by harnessing the power of crowd; who are in turn rewarded financially or through social incentives. This paper is aimed at exploring the concept of software development crowdsourcing, outlining its benefits over the traditional means of outsourcing software development, the models and methodologies of Crowdsourcing, as well as laying viable claim as to why it is a better and a reliable option; and thus should be adopted.

Keywords: Crowdsourcing, Crowdsourcing Software Development, Open Call Tasks, Software Development, Software Outsourcing.

I. INTRODUCTION

Outsourcing Information Technology components is fundamentally changing the landscape of global business operations, by lowering business operational cost and allowing firms focus on other potential business segments [1]. Crowdsourcing can be mirrored through outsourcing. Outsourcing entails delegating routine activities or tasks from an internal firm to an external entity such as subcontractor. It is the process of placing the development of system components in another development organization with the aim of achieving cost reduction and a gain in the time of development [2].

Software development outsourcing is among the common practice in the global business today. A common understanding of software outsourcing involves the transfer of software activities and series of processes (software design and implementation) by client organizations, i.e. outsourcers to specialized providers, i.e. vendors, with a primary motive of operating costs reduction [3].

Crowdsourcing is a phenomenal boom that emanates from the rise of online community composed of like-minded

entrepreneurs and enthusiasts who work together, with creative and innovative solutions to business processes, while reducing production cost [4]. The idea is to take the services of voluntary online community to build software rather than taking the services of traditionally employed workers [5]. The objective of software development crowdsourcing is to develop and implement quality software with little or no cost involved, by harnessing the power of crowd. To meet this objective, the crowd workers who agree to work on the task are given some financial or social incentives [6]. The tasks could be executed in a collaborative or competitive manner based on the organization style. Wikipedia and Linux are viewed as well-known collaborative crowdsourcing examples ([7]; [4]). Developing a software through crowdsourcing breaks the barrier that exists between a software developer and a user, and follows a co-creation principle [8].

Competitive crowdsourcing is reward based and has been used for variety of tasks and processes involved with software development [9]. With the ever increasing level of interest in crowdsourcing software development, it is significant to analyse the development process methodology used by crowdsourcing platforms, as well as the merits of

crowdsourcing software development over traditional software development outsourcing.

The aim of this paper is to enlighten and discuss the concept of crowdsourcing in software development processes. In specifics, this paper is intended to achieve the following objectives:

1. Briefly review the concept of traditional software development outsourcing
2. Identify the challenges and draw-backs of the traditional software development outsourcing
3. Discuss the concept of crowdsourcing software development and the benefits to software industry and other software users
4. Discuss the key elements and models of crowdsourcing, as well as methodologies involved in crowdsourcing software development process.
5. Identify the factors for successful software development crowdsourcing.
6. To make policy recommendation based on the subject matter.

II. LITERATURE REVIEW

Traditional Software Development Outsourcing

Before systems development methods came into being, the development of new systems or products was often carried out by using the experience and intuition of management and technical personnel. However, as a result of the growing complexity of modern systems and computers, there is the need for a clear and well-ordered software development process.

According to [10] and [11], Traditional Software development outsourcing involves sub-contracting or "farming out" or seeking resources for part or all of the software development phases and/or activities to/from independent, third-party companies or individuals, instead of keeping those functions in-house. An organization would use outsourcing for functions ranging from infrastructure acquisition to a wide scope software development, implementation, maintenance and support. The overall aim of this is to save cost and share risk with third party firms. Most large organizations only outsource a portion of any given IT function. [12] defined it as contracting out the development, planning, management, training, maintenance or operation of software services, skills, products or applications. Previous research shows that software development outsourcing gained a dramatic boom after 2001 and is still growing continuously due to economic downturn [13]. Outsourcing Software Development offers many benefits like access to leading-edge technology, skilled human resource, cheaper and high quality software development [11]. To avoid rip off in opportunities and certain business threats in a software outsourcing relationships, both client and the software

outsourcing vendor need a mutual understanding and trustworthy relationships.

Irrespective of the benefits of traditional software development outsourcing, there are still certain key problems faced. According to [14], traditional outsourcing of software development has a number of specific risks that can influence the choice of crowdsourcing:

1. **Uncertainty about software development success.** This is a problem of a growing concern. The first thing that provider tends to assure a client in is the successful development and implementation of the outsourced software, as well as a secure handling of essential information about client's core business process.
2. **Reluctance to access outside expertise.** An outsourcing provider will always feel comfortable with its own software development expertise, whether it is adequate or not. Usually, there is a reluctance by these providers to seek for external expertise to augment its own expertise in a case of deficiencies [37].
3. **Uncertain financial payback.** While the provider can have some guarantees of his cash flow set by contract conditions, the client is uncertain about the payback on the product. The delivered software can meet all the technical requirements, and in the same time it might turn out to be obsolete, or do not support some important technologies and standards that appeared during the development period. A sudden bankruptcy, changes in local or foreign legislation, political instability and other factor endanger provider's income as well.
4. **Increased complexity of management.** Software Development Outsourcing is considered to be a solution to decrease complexity of management of the software functions. But often management of OSD project suffers from a lack of control since a single provider is in charge of the software development project.
5. **The lack of control over the outsourced resources.** A situation when provider has to develop a software that will be used over a heterogeneous IT infrastructure of a client composed of a variety of devices, legacy systems, which cannot be replaced or upgraded by the contractor's will. This creates specific technical challenges that can reduce outcome of an OSD contract.

Crowdsourcing Software Development (CSD)

Software development is no longer a common practice among small groups of developers, but increasingly takes place in organizations and niche involving a number of persons ([15]; [16]). One emerging approach to getting work done is crowdsourcing, a sourcing strategy that emerged in the 1990s [17]. As a result of the limitless opportunities of the World Wide Web (WWW), organizations can tap into a workforce consisting of anyone (even remotely apart) with

an Internet connection. Customers, or software development requesters, can advertise chunks of work, functions or activities on a crowdsourcing platform, where suppliers (i.e., individual workers) select those tasks that match their interests and abilities. Crowdsourcing have been adopted in a wide variety of domain – education, health care, and social context; but for this study, we shall lay emphasis on crowdsourcing in software development.

The term crowdsourcing is an aggregation of the words “outsourcing” and “crowd”. Crowdsourcing is first described by [18] as a new Web-based business model that harnesses the creative solutions of a distributed network of individuals through what amounts to an open call for proposals” [18]. This definition is quite vague. A second more extensive definition of [18] posits that “crowdsourcing represents the act of a company or institution taking a software function/duties once performed by employees and outsourcing it to an unknown, geographically dispersed network of individuals with different skills in the form of an open call. This can take the form of peer-development (collaborative or team work), but it is sometimes often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential labourers.” This second definition fits better than the first one, because it states clearly that the job can be performed collaboratively, but also individually and that a company uses an open call to reach the large network of potential experts.

There are certain similarities between crowdsourcing and open innovation. Open Innovation is the use of purpose-driven acquisition and dissemination of knowledge to accelerate internal innovation and creativity, and to expand the markets for external use of innovation, respectively). Both ways of innovating use (external) people instead of own employees to solve problems or to invent new products or services. But there is a major difference: companies which use a crowdsourcing approach, in contradiction to companies that use an open innovation approach, do not use a predefined group of experts or companies. They outsource functions to an unknown set of remotely located individuals in the form of an open call, where companies with an open innovation approach use a predefined network of knowledgeable experts to collaborate with.

Incorporating the definition of crowdsourcing by [4], the researcher can thus define software development crowdsourcing as the act of a company or institution taking a software development function/activities, and outsourcing it to a large, undefined, unknown and geographically distant network of individuals in the form of an open call. It can also be defined as the process of contracting part or all of an organization’s software development functions to an unknown, undefined and a large network of persons (crowd),

in other to assume responsibility over those functions [10]. In addition to reliably developing the required software, crowdsourcing software development (CSD) can enable an organization coordinate human endeavours, help social interactions, and power creativity, as well as to solve potential inherent problems and issues [19]. CSD utilises an open call format to attract online developers (Usually known as developers in the cloud) to accomplish various types of software development tasks such as architecture, component design, component development, testing, evaluation, documentation and bug fixing [20]. Software development crowdsourcing eliminated the separation between software developers and users, allowing for a co-creation paradigm - that is, a regular user becomes a co-designer, co-developer, and co-maintainer. This is a paradigm shift from conventional traditional software development, with developers distinct from users, to a crowdsourcing-based, peer-production software development, in which many users can participate [21].

A cloud (online, undefined and geographically dispersed developers) provides a scalable platform with sufficient resources, including computing power and software databases, for a large crowd of developers. A cloud significantly reduces the amount of manual labour needed to set up software production environments and empowers peer developers to perform software crowdsourcing tasks efficiently in design, coding, and testing [21].

Based on its organizational style, software development crowdsourcing can be either competitive or collaborative. In competitive crowdsourcing, only winning participants are rewarded. In collaborative crowdsourcing, people cooperate with each other on various aspects, including funding, concept development, user interface design, code, test, and evaluation. The process design (such as activities, duration, and number of participants), support infrastructure, and software projects are different for these two approaches. Basically, there are three key defining elements of software development crowdsourcing; “Outsourcing”, “Crowd” and “Software Development”

a. Outsourcing

Outsourcing, in its most basic form, can be conceived of as the purchase of a good or service that was previously provided internally from outside providers. Many information systems researchers also define it as contracting of various internal business needs or functions to outside service providers [22]. Except for the fact that crowdsourcing is an emerging type of “small-scale outsourcing” [5], both outsourcing and crowdsourcing share similar objectives in that they source in their business needs from outside entities to achieve their business goals. Outsourcing entails the delegation of tasks or jobs from internal production to an external entity such as subcontractor. It is the process of

placing the development of system components in another development organization with the aim of achieving cost reduction and a gain in the time of development.

b. The Crowd

In the software development crowdsourcing model, the crowd is the online community that is expected to play the role of “service providers” as producers, innovators, developers and problem solvers [23]. The benefits for individuals in the crowd are most of the time uncertain. Besides that, companies sometimes earn big profits by executing the ideas of individuals or a crowd. The crowd needs to be diversified as much as possible to ensure that the crowd renders optimally. But in practice the crowd seems to be undiversified. [24] states that the crowd currently is likely to be white, middle- or upper-class, English-speaking, higher educated, with high-speed Internet connections in the home. Moreover, the most productive individuals in the crowd are also likely young in age, certainly under 30 and probably under 25 years of age, as this age group is most active in Web 2.0 environment of massive content creation.

It is interesting to know why they want to participate, because in most cases, no financial rewards will be paid to consumers. [19] stated that there are two types of motivations individuals can trigger to participate, namely extrinsic and intrinsic motivations. An extrinsically motivated person performs an activity in order to obtain some kind of external reward, e.g. recognition for work done, fame, benefits for one’s career, the satisfaction of pursuing common goals or the opportunity to receive financial rewards. An intrinsically motivated person wants to participate in a crowdsourcing project for his or her own sake. Possible motivations could be a possible employment or working experience.

c. Software Development

Software development is the computer programming, documenting, testing, and bug fixing involved in creating and maintaining applications and frameworks involved in a software release life cycle and resulting in a software product [25]. The term refers to a process of writing and maintaining the source code, but in a broader sense of the term it includes all that is involved between the conception of the desired software through to the final manifestation of the software, ideally in a planned and structured process.

In a nutshell, software development process utilizes the crowdsourcing model to tap the collective knowledge of the community and/or to harness the crowd to directly develop, implement and maintain the software.

III. SOFTWARE DEVELOPMENT CROWDSOURCING MODELS

Software development crowdsourcing can be characterized in terms of the crowd size, software scale and complexity,

development processes, and competition or collaboration rules. Formal models for designing and modelling software crowdsourcing can have the following foundations:

Game Theory: The nature of contests in competition-based crowdsourcing can be analysed via game theory. For example, we can determine the reputation reward value based on the number of participants and the reward price; participants are often willing to compete to gain reputation rather than receiving a reward price determined according to a Nash equilibrium [26].

Economic Models: Economic competition models provide strategies and incentives to generate crowd participating, and reward-structuring rules for organizers to maximize crowdsourcing returns. The recently developed contest theory [27] introduces a new mathematical tools to describe the synergy among individual efforts, competition prize structure, and product quality.

Optimization Theory: Due to the competitive and dynamic nature of software development crowdsourcing processes, it can be a very challenging process to coordinate virtual teams, optimize the partition and allocation of development tasks, and striking an equilibrium between costs, time, and quality.

Technology Acceptance Model (TAM): Technology Acceptance Model, developed by [28], is one of the most influential research models in studies of the determinants of information systems and information technology acceptance to predict intention to use and acceptance of information systems and information technology by individuals. In the Model, there are two determinants including perceived ease of use and perceived usefulness. Perceived usefulness is the degree to which an individual believes that using a particular information system or information technology would enhance his or her job or life performance. Perceived ease of use is the degree to which a person believes that using a particular information system or information technology would be free of effort. Perceived ease of use and perceived usefulness positively affect the attitudes toward an information system; and further, positively affect the individuals’ intentions to use and the acceptance of the information system.

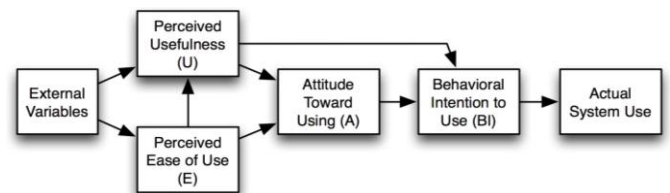


Figure 1: Technology Acceptance Model (Source: [29])

Knowledge Based Theory (KBT): This theoretical perspective suggests that crowdsourcing arrangements serve as a vehicle for utilizing the crowd’s complementary skills and expertise [30]. The first knowledge-based motivation for crowdsourcing is the desire to gain access to and exploit technical knowledge that the client firm does not possess [31]. Greater technical knowledge increases the ability of clients to more precisely spell out contract terms and to effectively monitor and supervise the crowd, increasing their relative bargaining power.

IV. CROWDSOURCING PERSPECTIVES

Crowdsourcing software development generally involves three types of actors [32]: customers, who have software development work that needs to be done; workers, who participate in developing software; and platforms, or brokers, who provide an online market place where customers and workers can meet. Each of these three actors will have a different perspective on crowdsourcing software development. We briefly discuss them below.

Customers: Customers, or requesters, are organizations or individuals who have software tasks that need to be developed. Customers can have different motivations for crowdsourcing software development; for instance, an organization may temporarily want to increase their software capacity. By ‘outsourcing’ some of the work to the ‘crowd,’ an organization can become quite flexible in scaling up and down their production capacity.

Workers: Workers are the individuals who perform the work - they develop the chunks of software that are ‘outsourced’ by a customer. Little is known about who these workers and there are many open questions that would be of great interest. Some initial work has been done on workers’ motivation, but we need a better understanding of why participants engage in this ‘insecure’ form of employment. Furthermore, while ‘normal’ development jobs typically require certain levels of formal education, little is known about the background of crowdsourcing workers.

Platforms: Platforms provide an online ‘marketplace’ for workers and customers to meet. The largest platform for crowdsourcing software development is TopCoder, with a developer community of more than 600,000 as of January 2014 [32]. However, there have also been indications only a small fraction of its registered users are actually participating in software development. Platforms may have different participation models. TopCoder for instance uses a competition based model—in TopCoder’s model, the winner gets paid a certain prize, and the runner-up receives 50% of this prize.

Crowdsourcing Support Tools

Given software development crowdsourcing’s distributed nature, it needs a powerful development environment to

facilitate software design, coding, test, and deployment across distributed and heterogeneous infrastructures. All participating team members should be able to access the same software development environment customized for a specific project. Moreover, numerous submissions from these team members must be quickly screened, evaluated, reviewed, and integrated. Thus, software crowdsourcing efforts require enhancements to common software development tools for coding, testing, and deployment to support automatic project building, integration, performance analysis, and security checking. Table 1 summarizes the various tools supporting crowdsourcing:

Software Development Tools: An IDE for crowdsourcing integrates tools for requirements, design, coding, compilers, debuggers, performance analysis, testing, and maintenance. Software development tools support large-scale system log administration and analysis, and help team members resolve software problems and enhance system reliability using log messages.

Social Networks and Collaboration Tools: Facebook, Twitter, wikis, blogs, and similar Web-based tools let participants communicate for sharing and collaboration. For example, organizers can use Facebook profiles to form a virtual crowdsourcing team, even if the participants don’t know each other.

Project Management Tools: Crowdsourcing project management should support project cost estimation, development planning, decision making, bug tracking, and software repository maintenance, all specialized for the context of the dynamic developer community. In addition to these regular functions, it must incorporate ranking, reputation, and award systems for both products and participants.

Table 1: Software Development Crowdsourcing Support Tools (Source: Research)

Crowdsourcing Support Tools	Characteristics
Software Development Tools	Streamlines the environment configuration for online-oriented software development and testing Supports large-scale system log administration and analysis Web-based IDE that helps simplifies and integrates online software development
Social Networks and Collaboration Tools	A team-based collaboration tool for content creation and sharing
Project Management Tools	A crowdsourcing tool for online development that ranks developers’ skills and organizes contests for development tasks A project tool that supports collaborative software development, and can serve as code repository for online-based software crowdsourcing An online software development tool that integrates project hosting, social networking, and programming education

Crowdsourcing Software Development Methodology

The study of [20] proposed a typical methodology for software development crowdsourcing:

a. Development Process

Crowdsourcing software development process generally follows a waterfall model, which is shown in Figure 2. Each of the phases is achieved by the open competition format. The development process commences with a requirement phase. During this phase, the project manager, who comes from the crowd is responsible for managing the following phases and communicating with the client companies to identify their project goals, task plan and estimated budget. Then the requirements specification is defined and passed as the input to the next phase. The subsequent architecture phase decomposes the application into a set of components.

The component design activity produces a full set of design documentations such as UML diagrams and component specifications. These specified design components are then implemented in the subsequent development phase. The component development activity may incorporate pre-built reusable components. The finished components are combined together in an assembly phase and are further certified by system-level testing activities. Finally, the fully functioning solution is deployed into the customer's quality assurance environment in the deployment phase. After a period of user acceptance testing, all developed assets are delivered to the client.

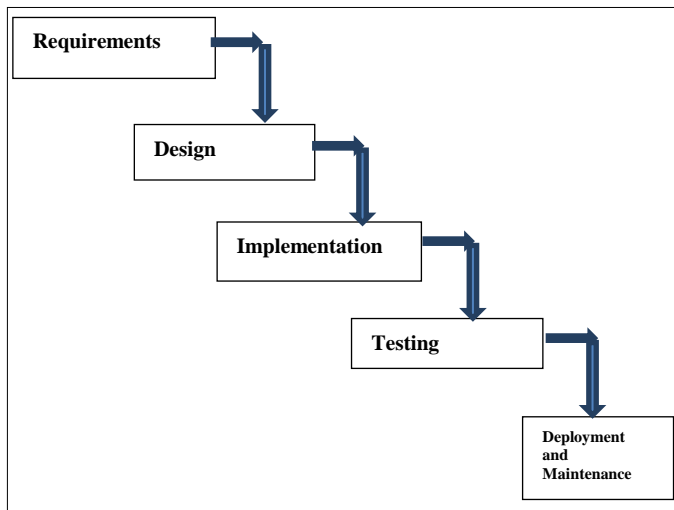


Figure 2: The Waterfall System Development Model (Source: Research)

b. Open Call Tasks

Crowdsourcing utilises an open call format to attract online workers to make contributions to the posted tasks. Usually, the open call format is in the form of a competition. Each crowdsourced development task is organised as an open contest. The typical open call task phases are divided into a series of task competitions. The duration of the whole process for a single task is usually 1-2 weeks. The process to run an open call task is illustrated in Figure 3. To start with, a task categorised by its development type is posted on the website with information such as task descriptions, payment amount and time lines. The time lines include two important dates: registration deadline and submission deadline. All developers who are willing to participate should announce their decision publicly by registering the contest. This means online developers are allowed to observe opponent developers' information including historical performance and skill ratings.

After the submission deadline, all submitted solutions are collected to be evaluated by peer review according to pre-defined screening and review scorecards. The solutions that passed screening are scored (usually by three community experts from different perspectives, e.g., performance on accuracy, stress and failure tests). Once the review process is finished, the developers are notified privately. If they are not satisfied with the results they have one chance to argue with the reviewers (called the appeal phase). Finally the revised scores are announced on the website and contestants are ranked by the average score given by all reviewers. The 1st place winner gets the full payment and the runner-up gets half of this amount.

c. Participation Incentives

The prize money is an important factor in motivating the crowd participation [20]. But not everyone registers the competition for winning the prize. There are multiple types of incentives for developers to undertake CSD tasks. According to [20], these incentives include gaining skills, getting feedback, making friends, earning money, having fun, getting peer recognition and getting a sense of accomplishment. Since usually only the top two winners of the tasks can be rewarded with money, the developers often get nothing monetary for their effort. But through competing in the task competition, developers may find it exciting and their submissions are tested by screening and further reviewed by the experts of the community. In this way they are rewarded with fun, acknowledgement and experts' feedback for improving their skills.

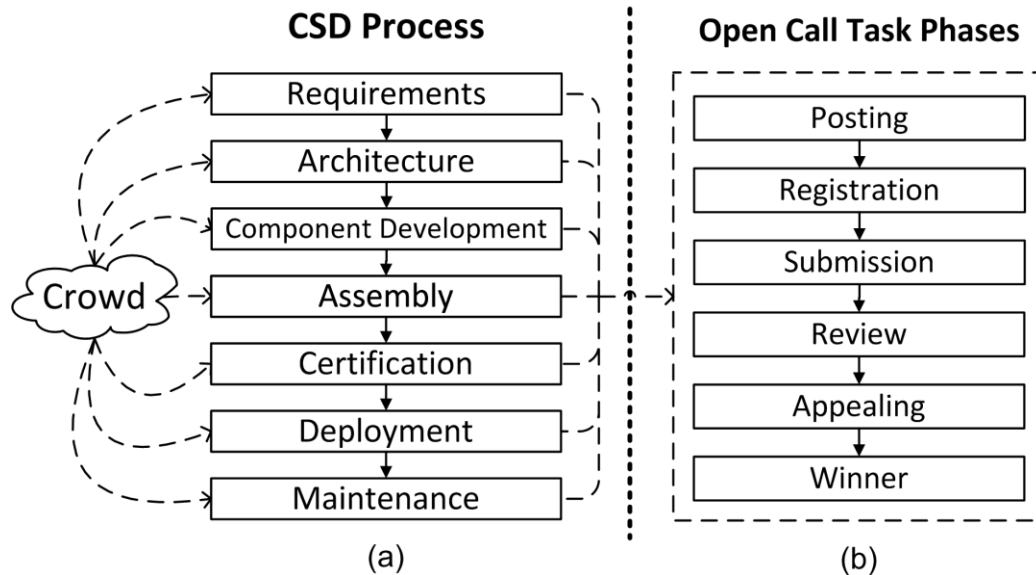


Figure 3: Crowdsourced Software Development Process and Task phases (Source: [20])

Benefits of Crowdsourcing

A number of potential benefits have been linked to the adoption of crowdsourcing, and these benefits would also be applicable in the context of software development specifically [32]:

1. **Cost Reduction:** This is achieved through lower development costs for developers, and also through the avoidance of the extra cost overheads typically incurred in hiring private software developers. In addition, a crowdsourcing organizer can acquire a software at a much reduced cost because only those individuals or team that successfully completes the software development function would be paid, and could even pay below-market costs, given that participants might seek reputation rewards rather than monetary ones.
2. **Faster Time-to-Market:** Through accessing a skilled network of talents, who can achieve a great deal of development across various software phases, as well as parallel development on tasks, and who are typically willing to work at weekends.
3. **Higher Quality:** Through broad participation the ability to get access to a broad and deep pool of development talent who self-select on the basis that they have the necessary expertise, and who then participate in contests where the highest quality 'winning' software is chosen. Such software comes from competent participants who try to outdo their peers in submitting innovative concepts, design, code, or tests.
4. **Creativity and Open Innovation:** The large network and variety of skilled expertise available ensures that more creative solutions can be explored, and the best result adopted.

5. **Talent Identification:** An organizer might be interested in identifying talented developers, as demonstrated by their performance in competitive efforts.

Other goals include solution diversity, idea creation, broadening participation, marketing, and participant education, such as encouraging people to use or learn specific tools. To ensure a good outcome from software crowdsourcing, organizations can leverage a cloud infrastructure to accelerate the process of setting up the development environment and enabling distributed and large-scale development by a highly dynamic community.

In addition to the benefits of crowdsourcing discussed above, organizations should consider the following five (5) factors in making crowdsourcing a success [32]:

1. **Include a diverse audience:** The strength of crowdsourcing lies in its ability to engage a large, diverse group of people in collaborative problem solving. Research shows that significant advantages arise from including people with a diverse range of problem-solving approaches, backgrounds, and personal experiences [32]. Often times, those not considered expertise with solving a problem often are more successful because they look at the issue from a fresh perspective
2. **Provide a Clear Purpose:** To ensure that crowdsourcing does not descend into chaos when there are divergent or relevant or irrelevant ideas, crowdsourcing organizers should provide a specific challenge for the company. An explicit purpose should give employees a catalyst, a community, and an outlet to unlock untapped innovation

3. **Motivate to Participate:** While employees are typically enthusiastic about providing input for new initiatives, there is no guarantee of universal participation. Participation is more likely to be self-generating in companies that have a trusting environment in which employees are engaged and encouraged to contribute.
4. **Allow Time to Innovate:** Companies demand of employees to innovate, yet are unwilling to grant them the time to do so. Allow for freedom of thought and action, and people will surprise you.
5. **Do not throw out Ideas:** Diverse audiences will deliver diverse solutions, and the most viable and financially promising suggestions will typically win out. A key to successfully fostering innovation is the ability to turn failures into lessons and future successes. To truly evolve, companies must imbibe a culture in which even a failed idea is encouraged to continually push the boundaries.

V. CROWDSOURCING IN PRACTICE

[4] describes six types of crowdsourcing: collective intelligence, crowdcreation, crowdvoting, crowdcreativity, cloud labour, and crowdfunding.

1. **Collective Intelligence:** A first definition of collective intelligence was designed by Douglas Hofstadter [18]. Nowadays collective intelligence has more a focus on communications technology, namely the internet. Web 2.0 has enabled interactivity and thus, users are able to generate their own content. Collective Intelligence leverages on this to enhance the social vast network of existing knowledge. "According to Howe "collective intelligence is not only merely a quantitative contribution of information from all cultures; it is also qualitative".
2. **Crowdcreation:** [18] state that another type of crowdsourcing is crowdcreation. [33] describes this type as "crowdsourcing for product design or configuration". This is because sometimes a crowdsourcing call is used to gather users for the creation or design of a product that totally depends on their input.
3. **Crowdvoting:** Crowdvoting occurs when a website seeks the opinion and judgement of a large group of unknown individuals on a particular subject matter. A typical example is a prediction market that gathers crowds' views on football, and ensures accuracy by having participants bet a certain amount of money based on the outcome of the football result.
4. **Crowdcreativity:** Harnessing the innovative pool of skilled individuals and online communities to design and develop original products and concepts.

5. **Cloud Labour:** Leveraging of a distributed virtual labour pool, available on-demand to fulfil a range of tasks from simple to complex. Crowdsourcing is used to connect labour demand and supply. Virtual workers perform activities that range from simple to specialized tasks.
6. **Crowdfunding:** While crowdsourcing is mainly focused on the knowledge and preferences of consumers, crowdfunding is about the financing of project and people by large crowds [4]. Crowdfunding occurs for any variety of purposes, from disaster relief to citizen, to individuals seeking support online, to political campaigns.

Critical Success factors for Crowdsourcing

The critical success factor model for crowdsourcing is based on [34], [35], and [36] while developing the peripheral factors which affect the motive alignment of the crowd towards the crowdsourcing initiative. In the model, motive alignment of the crowd is the central idea whereas the vision & strategy of the crowdsourcing initiative, linkages & trust, external environment, infrastructure and human capital are the peripheral factors. This model is aimed at exploring the space of crowdsourcing and provide an overview of the important considerations to be kept in mind while implementing a crowdsourcing initiative [4].

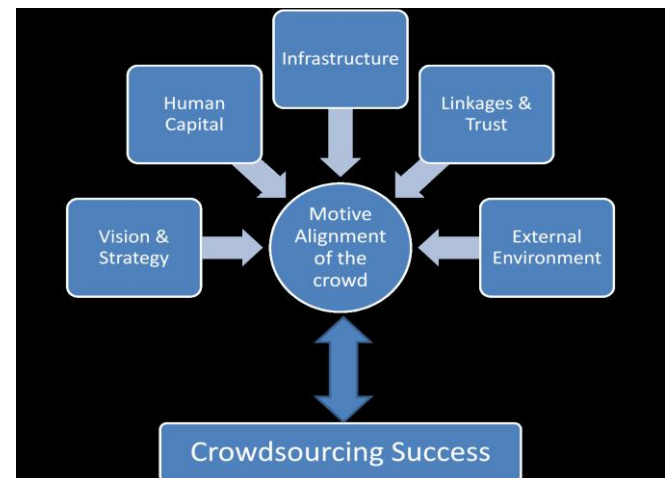


Figure 4: Crowdsourcing Critical Success Factor Model (Source: [4])

The Future of Crowdsourcing

It is almost certain crowdsourcing will dramatically change the nature of work and creativity in the future. As crowdsourcing continues to make previously scarce resources become much more abundant, what customers are willing to pay for will change and evolve dramatically. This will likely have very far reaching implications. Crowdsourcing harnesses the power of today's communication technologies to liberate the potential which

exists in large pools of people. It will shift the way work gets done.

Crowdsourcing represents a fundamentally different way to make and sell stuff, and the phenomenon is still in its infancy. If an individual or organization aspires to be involved in the future, these rules should be kept in mind:

1. Pick the Right Model
2. Pick the Right Crowd
3. Offer the Right Incentives
4. Keep employing people
5. Find benevolent Dictators
6. Keep things Simple
7. Be Prepared for fluff
8. Look for diamond in the rough

VI. CONCLUSION AND RECOMMENDATION

Based on the literature review and its discussions, the researchers draw the following conclusion:

1. That traditional software outsourcing process has certain limitations as outlined. A better alternative is crowdsourcing. Crowdsourcing taps into a wider range of skilled talents of the crowd in overcoming the challenges facing traditional outsourcing process.
2. Barring a little concern about crowdsourcing processed, it provides a viable alternative for individuals and organizations alike. An open call for innovativeness and creativity from a network of expert is recommended as this can also result in reduced cost and higher productivity.
3. Furthermore, organizations and users that adopts crowdsourcing can enjoy potential high reduction in cost for software development, higher quality, high creativity and identification of great talents.
4. If organizations and other software industry can include a diverse audience, provide a clear software development purpose, motivate users to participate, allow time to innovate, not throw out ideas from the crowd; then there can be success in crowdsourcing.

The researchers thus recommends that organizations and software industries adopt crowdsourcing of their software development to enjoy the benefits (large pool of knowledge, reduced overheads amongst others) involved as well as the ease and flexibility in developing a software.

REFERENCES

- [1] B.C. Asiegbu, I.V. Oluigbo, S.M. Nzotta, "Determining Success Factors of Software Development Outsourcing for Nigeria: A Data Reduction Technique", International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, Vol.3, Issue.4, pp.481-490, 2016
- [2] I.V. Oluigbo, B.C. Asiegbu, S.M. Nzotta, G.N. Ezeh, "A Two Step Clustering of Critical Success Factors for Software Development Outsourcing Peculiar to Selected Nigerian Sectors", International Journal of Science Technology Management and Research, Vol.1, Issue.6, pp. 1-10, 2016
- [3] H. Smuts, "Critical success factors for Information Systems outsourcing management: A software development lifecycle view", Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, US, pp. 304-313, 2010
- [4] J. Howe "Crowdsourcing: Why the power of the crowd is driving the future of business", Crown Business, New York, pp. 77-99, 2008
- [5] D. Gefen, E. Carmel, "Is the world really flat? A look at offshoring at an online programming marketplace" MIS Quarterly, Vol. 32, No.2, pp. 367-384, 2008
- [6] L. Hoffmann, "Crowd Control", Communications of the ACM, Vol.52, No.3, pp.16-17, 2009
- [7] A. Doan, R. Ramakrishnan, A.Y. Halevy, "Crowdsourcing systems on the World-Wide Web", Communication of the ACM, Vol. 54, No.4, pp. 86, 2011
- [8] W.T. Tsai, W. Wu, M.N. Huhns, "Cloud-Based Software Crowdsourcing", Journal of Internet Computing, Vol. 18, No. 3, pp. 78-83, 2014
- [9] K. Stol, B. Fitzgerald, "Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development", Proceedings of ICSE 2014, USA, pp. 187-198, 2014
- [10] Rob Aalders, "The IT Outsourcing Guide", John Wiley & Sons, Inc. USA, pp.10332, 2001.
- [11] B. Vangie "IT Outsourcing: The Platform Vision of API Giants", webopedia, UC, pp.1-19, 2012.
- [12] B. Nicholson "Global software outsourcing: The solution to the IT skills gap", Anglo-German Foundation for the Study of Industrial Society (Anglo-German Foundation), UK, pp.45-57, 2001.
- [13] U.K. Siffat, U.A. Asad, "Knowledge Sharing Management Model (KSMM) for Software Development Outsourcing Vendors", International Journal of Advanced Science and Technology, Vol.45, Issue.7, pp.135-147,2012
- [14] I. Oleg, "Outsourcing of Software Development": Humboldt Universität zu Berlin, Berlin, pp.1-25, 2005
- [15] A. Biegel, J. Bosch, M.A. Storey, "Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder", IEEE Software, vol. 30, No.1, pp.52-56, 2013.
- [16] D. Tamburri, P. Lago, H.V. Vliet, "Organizational social structures for software engineering", ACM Computing Surveys, vol. 46, No 1, pp.1-3, 2013.
- [17] S. Greengard, "Following the Crowd", Communication of the ACM, vol. 54, No.2, pp.20-22, 2011.
- [18] J. Howe, "The rise of crowdsourcing", Wired magazine, Vol.14, No 6, pp.53-61, 2006.
- [19] M. Parameswaran, A.B. Whinston. "Social Computing: An Overview", Communications of the Association for Information Systems, Vol.19, Issue.1, pp. 762-780, 2007.
- [20] K. Mao, Y. Yang, Q. Wang, Y. Jia, M., "Harman Developer Recommendation for Crowdsourced Software Development Tasks", 2015 IEEE Symposium on Service-Oriented System Engineering, CA, pp. 347-35, 2015.
- [21] S.L. Mewada, U.K. Singh, P. Sharma, "Security Enhancement in Cloud Computing (CC)", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.1, pp.31-37, 2013.
- [22] R. Kishore, H. Rao, K. Nam, "A relationship perspective on IT outsourcing", Communications of the ACM, Vol.46, No.12, pp.87-92, 2003.

- [23] G.D. Saxton, O. Oh, R. Kishore, "Rules of Crowdsourcing: Models, Issues and System of Controls", Journal of Information Systems Management, Vol.30, Issue.1, pp.1-2, 2010.
- [24] K.Vetrikodi and V.Geetha, "Efficient Analysis of Differential Query Services in Mobile Crowd and Its Location Privacy", International Journal of Computer Sciences and Engineering, Vol.3, Issue.8, pp.168-175, 2015.
- [25] R. Klopper, S. Gruner, D. Kourie, "Assessment of a framework to compare software development methodologies", Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, South Africa, pp. 56-65, 2007
- [26] W. Wu, W.T. Tsai, W. Li, "An Evaluation Framework for Software Crowdsourcing", Frontiers of Computer Science, Vol.7, No.5, pp.694-709, 2013.
- [27] L.C. Corchón, "The Theory of Contests: A Survey", Review of Economic Design, vol.11, No.2, pp. 69-100, 2007.
- [28] F.D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", MIS Quarterly, Vol. 13, No. 3, pp. 319-340, 1989
- [29] S.C. Chen, S.H. Li, C.Y. Li, "Recent Related Research in Technology Acceptance Model: A Literature Review", Australian Journal of Business and Management Research, Vol. 1, No. 9, p. 124-127, 2011
- [30] R. Grant, C. Baden-Fuller, "A knowledge accessing theory of strategic alliances", Journal of Management Studies, Vol. 41, No. 1, pp. 61-84, 2004
- [31] N. Levina, J. Ross, "From the vendor's perspective: Exploring the value proposition in information technology outsourcing", MIS Quarterly, Vol. 27, No 3, pp. 331-364, 2003
- [32] J.S. Klaas, F. Brian, "Researching Crowdsourcing Software Development: Perspectives and Concerns", ICSE-2014, India, pp.58-64, 2014.
- [33] F. Kleemann, G.G. Vob, K. Rieder, "Underpaid innovators: the commercial utilization of consumer work through crowdsourcing", Science Technology and Innovation Studies, Vol.4, No.1, pp.1-5, 2008
- [34] R. Heeks, B. Nicholson, "Software Export Success Factors and Strategies in Developing and Transitional Economies", Competition and Change, UK, pp. 267-302, 2004
- [35] E. Carmel, "The New Software exporting nations: Success Factors", Electronic Journal of Information System Development, Vol. 13, No. 4, pp. 1-12, 2003
- [36] D. Farrell, "Smarter Offshoring", Harvard Business Review, US, pp. 84-92, 2006
- [37] I.V. Oluigbo, B.C. Asiegbo, G.N. Ezech, O.C. Nwokonkwo, "Group Membership Prediction of an Outsourced Software Project: A Discriminant Function Analysis Approach", International Journal of Scientific Research in Multidisciplinary Studies, Vol. 3, Issue 4, pp. 12-18, 2017.

AUTHOR'S PROFILE

Professor Asiegbo Baldwin C. is a professor of Economics and Information Management Technology in the department of Information Management Technology, FUTO. He had his BSc, MSc and Phd in Economic Cybernetics from Kharkov State University, Ukraine. He is a member of Nigerian Economic Society, Information Technology Financial Management Association, and an affiliate member of the Computer



Professionals of Nigeria. His Research Interests includes Applied (Managerial) Economics and Business Management, as well as Information Management Technology. He has published over 35 research papers and Technical Reports in international journals and conferences.

Oluigbo Ikenna Victor has a B.Tech and MSc in Information Technology from Federal University of Technology Owerri, Nigeria (FUTO). He is a seasoned IT Researcher and a lecturer in the department of Information Management Technology, FUTO. He is a member of International Association of Computer Science and Information Technology, International Association of Engineers, Institute of Research Engineers and Doctors. In addition, he is a reviewer member of International Journal for Innovative Research in Science and Technology, and International Journal of Trends in Research and Development. He has published over fifteen research papers in recognised international journals. His research interests includes Network Administration and Security, Data Science and Big Data, Green Technologies and Energy Efficient Networks, Impact of ICT and Internet Technologies, Research Methodologies, Quantitative and Qualitative Analysis.



Ajakwe Simeon O. is a research fellow and doctoral candidate in the department of Information Management Technology, FUTO. He is a member of IEEE, NCS. Qualifications: MSc-IMT, BSc-Comp Sci, CITP(Chartered Information Technology Professional), OCA(Oracle Certified Associate).His research interest from Software engineering, outsourcing, Big Data analytics, knowledge management and Artificial Intelligence, Information System.



Onyike Gerald Obinna is currently a lecturer in the Department of Computer science, Federal polytechnic Nekede, Owerri. He holds a BSc in computer science from Imo State University, Owerri and PGDE from the National Teachers Institute (NTI). He is currently undergoing a Master's degree programme in information Technology at the National Open University of Nigeria. He is a member of computer professionals Registration council of Nigeria (CPN). He is happily married with children.

