https://doi.org/10.48009/3_iis_2017_116-128

Issues in Information Systems
Volume 18, Issue 3, pp. 116-128, 2017

THE TAXONOMY OF ESTIMATION IN SOFTWARE DEVELOPMENT PROJECTS

Rosarito Sánchez-Morcilio, University of Puerto Rico, Río Piedras Campus, rosarito.sanchez@upr.edu
Francisco Quiles-Torres, University of Puerto Rico, Río Piedras Campus, francisco.quiles@upr.edu

ABSTRACT

Truthful estimation represents a critical role to managers of software projects. An in-depth analysis of literature was performed concerning estimation of software development projects. Statistical models were studied to find estimation variables that are relevant to project managers. A taxonomy of estimation was created to present the insights of effective estimation in software development projects. The taxonomy arranges a list of considerations to successfully manage software development projects. In addition, straight forward recommendations are provided for project managers to update their knowledge in estimation of software development projects. This article gives direction towards adequate estimation in software development projects.

Keywords: Project Management, Software Development, Estimation, Considerations for Estimation.

INTRODUCTION

Organizations are now using technology to conduct their operations more efficiently. Advanced and state of the art operations only take place as a result of successful software development projects. Due to increased demand for software developers, it is projected to grow 17% from the years 2014 to 2024, much faster than the average of all occupations (Bureau of Labor Statistics, 2017). Most of the project management currently conducted is related to technology requiring software development activities. Project management processes, which include initiation, planning, executing, monitoring and controlling, and closing (Project Management Institute, 2013), are important to conduct software development undertakings effectively. Successful project management by the government of a country leads to economic prosperity (Dutta, Geiger, & Lanvin, 2015).

Estimating in software development projects, which involves cost and time, is challenging (Bardsiri, Hashemi, & Razzazi, 2015). Predicting cost and time to execute a project that involves new concepts is even more challenging due to its nature (Bouras, Aqle, Alsardy, & Nizam, 2016). Software development projects are known for runaway costs, making the estimation process critical for success (Uzzafer, 2017). There are many articles and books about estimation in software projects; the project manager needs to decide which techniques he will use on his given project to avoid being overwelmed by too much information or be inaccurate due to the lack of it. It takes more than a simple set of variables to be precised in estimation. For instance, using historical estimations may not be the only variable to consider since most software development projects are new concepts, unique and complex in nature (Siddique & Hussein, 2014, and Khatibi & Bardsiri, 2015, Bardsiri, Hashemi, & Razzazi, 2015). Moreover, software developers do not feel they are experts in estimating because both feel presure to give a competitive price for customers while feeling humble enough to recognize the magnitude of their effort (Siddique & Hussein, 2014). Using statistical models to come up with accurate estimations is ideal, but it's not always feasible from a managerial perspective (Datey, 2015).

The purpose of this study is to show a taxonomy of estimation in software development undertakings. The taxonomy arranges a list of considerations project managers can understand and follow when dealing with software development project estimations. First, definitions and important aspects about software development projects are presented, followed by the methodology, findings, conclusion and recommendations for future work. The main contribution of this paper is presented in the discussion section, which is the taxonomy for the reviewed estimation on software development projects.

BACKGROUND

Definitions

A *software developer* is a professional who writes computer programs, which include applications to do specific tasks or to control devices or networks tactically and creatively (Bureau of Labor Statistics, 2017). A *project manager* is a professional who leads the team responsible for achieving the project objectives and who has knowledge about project management (Project Management Institute, 2013). *Project Management* (PM) is a short-lived venture, limited by time, budget, resources and scope, to create a unique product or service according to the client's requirements (Project Management Institute, 2016). An *estimate* represents an approximation of a quantity based on information available at the time recognizing that further relevant evidences are uncertain or undetermined (Business Dictionary, 2017). *Knowledge* is, in the organizational context, the sum of what is known and resides in the intelligence and the competence of people (Business Dictionary, 2017). *Taxonomy* is the study of the general principles of scientific classifications (Merriam-Webster, 2017).

Software development projects are knowledge intensive

Software development is a knowledge intensive process completed by a well-educated workforce (Vasilchenko & Morrish, 2011). Software development projects provide innovation by knowledge skills, which represent a competitive advantage for firms (Kärreman, 2010). These types of projects create value and add superior judgment to produce useful applications that are required to improve the performance of operations. Therefore, applications (developed software) become commodities to organizations. The transfer of knowledge among team members of a project being deployed globally can be complex and can affect positively or negatively the overall success of the project. Moreover, language variations, cultural dissimilarities, time zone changes, distinct procedures and practices and distinct hardware and infrastructure can lead to challenges when shifting knowledge (Betz, Oberweis, & Stephan, 2014).

Software development projects and intellectual property rights

Many ideas developed through the creation of software may be assets susceptible to intellectual property rights. Chen, Bharadwaj, and Goh (2017) explained there are three main intellectual property rights: *ownership, usage*, and *control*. Under *usage* property rights there are three types*: right to use*, *right to deploy*, and *right to sell*. *Right to use* means that the developed software can be used, but not be sold, modified, or distributed. *Right to deploy* means that the know-how and the information gained may be used for other endeavors. However, this residual intellectual asset is difficult to be clearly specified in many contracts. *Right to sell* means that the developed software can be sold in certain locations and businesses. Deciding the extents in the contract, especially those related to intellectual property rights, is crucial before starting a software development project. A study conducted by Chen, Bharadwaj, & Goh (2017) found fifty three percent (53%) of contracts offering sole ownership of the developed software to the project's client. Ethics in software development projects should also be taken into consideration. Calluzzo & Cante (2004) concluded that there is a significant confusion and lack of clarity with respect to behaviors associated with enterprise property, software, and hardware technology.

Project management methods to conduct software development projects

Software development must start with effective project management planning (Siddique & Hussein, 2014), following project management principles (Project Management Institute, 2013) and the appropriate methodology. The variety of project management methodologies include Projects in Controlled Environments (Prince2), Rational Unified Process (RUP) taxonomy, Six Sigma, Waterfall, and Agile (Schwalbe, 2015). The Waterfall method is well-known and suitable when clear goals and requirements can be set at the very early stages of the project (Kaleshovska, Josimovski, Pulevska-Ivanovska, Postolov, & Janevski, 2015). The Waterfall method was initially widely used in managing software development projects (Dawson & Dawson, 2014). However, it became ineffective due to the fact that its rigid structure did not allowed changes in the software during the project execution (Kaleshovska et al., 2015).

Agile techniques are particularly useful to manage software development projects (Lasek & Adamus, 2014) since those techniques are characterized by the division of tasks into short phases of work and frequent reassessment and

adaptation of plans (Schwalbe, 2015). Dawson and Dawson (2014) explained agile techniques allow the user to be actively involved in the development of processes and continually assessing on the software requirements. The iterative activity continues until the additional cost of the system outweighs the benefits. The authors concluded that due to the agile techniques flexibility, many software development projects are managed successfully today by following this methodology. Cient, Roberto, Avalia, Review, & Revis (2014) emphasizes that client involvement and experience, as well as knowledge, are key to successful agile practices. Agile techniques are important to successful software development projects, though not at enhancing customer responsiveness (Recker, Holten, Hummel, & Rosenkranz, 2017). Agile techniques include Scrum, extreme programming (XP), Test-driven development, adaptive software development, Agile Modeling, Agile Data Method, Lean Development (Schwalbe, 2015, and Kaleshovska et al., 2015). Scrum, a popular variation of agile technique, is a project methodology that provides flexibility and responsiveness contributing successfully in managing software projects, according to Kaleshovska et al. (2015). Scrum is an iterative framework that allows responding assertively to changes during the project evolution and welcomes the customer's active participation in the project.

When a customer requirement variation or an improved idea arises, that new change is included in a later stage of the project, in the following Scrum sprint or iteration. Scrum daily meetings allows to progress tracking and control of the project. Self-dedication and empowerment are given to the project team because better communication and project transparency are shown in those daily meetings. As a result, there is more cooperation and support among team members. User involvement in the software development process is very high. Every sprint, Scrum allows the possibility of re-planning and adjusting the project as opposed to traditional project management techniques that only permit planning during the project start. Kaleshovska et al. (2015) concludes scrum is highly recommended for managing software development projects.

Statistics on the estimation of software development projects

Estimation is an intelligent guess of a number (Business Dictionary, 2017). Statistics are strongly advised to obtain accurate estimations. Many authors have recently written about the software development estimation from the statistical standpoint (Azzeh & Nassif, 2016, Dragicevic, Celar, & Turic, 2017, Idri, Hosni, & Abran, 2016, Nassif, Azzeh, Capretz, & Ho, 2016, Sandoval & Ruiz Ascencio, 2016, Satapathy, Kumar, & Rath, 2016, Subramanian, Pendharkar, & Pai, 2017, Velarde, Santiesteban, Garcia, & Casillas, 2016a, and Velarde, Santiesteban, Garcia, & Casillas, 2016b), as well as the eight (8) articles reviewed in this study (Khoshgoftaar, Gao, Chen, & Napolitano, 2015, Rath, Acharya, & Satapathy, 2016, Popovic, Bojic, & Korolija, 2015, Siddique & Hussein, 2014, Uzzafer, 2017, Bouras et al., 2016, Datey, 2015, and Khatibi & Bardsiri, 2015). Estimation techniques for project planning used today by managers are basically closely related to statistics. Some of the estimation techniques are poker, user stories/points, Delphi, parametric/non-parametric estimates, expert judgment analog and comparison, three point estimations, and work breakdown structure + weighted effort estimation in hours (Siddique & Hussein, 2014). These statistical estimation techniques developed by researchers are difficult to implement for commercial use (Datey, 2015). Most statistical models for estimation are run by comparing data of past projects with new project requirements (Khatibi & Bardsiri, 2015).

Software datasets, a statistical model that uses historical data and past experiences, helps researchers make more accurate estimations assuming the newly developed software has similar requirements as those in the previous developed software (Bardsiri, Hashemi, & Razzazi, 2015). However, from the managerial standpoint, the variables used in those statistical models are relevant for the estimation process. Although statistical practices on estimation of software development project are reviewed in this paper, its main focus is to create a reviewed estimation of software development projects useful for project management.

Research question for the study

This study has the purpose of presenting an up-to-date version from the project management's view of the considerations on estimation of software development undertakings. A taxonomy of the reviewed/updated considerations on estimation of software development projects will be provided and discussed. The taxonomy of estimation is relevant to executives and academia alike. Project managers can review or learn, if new to the field, the current considerations on estimation of software development projects. Startup managers can follow the guideline considerations when developing software, as those are presented in this paper. Stakeholders, in general, can become

aware of the latest information about estimation in software delevopment projects. Information Systems or Project Management students can learn the specific details related to the estimation process within the software development environment. The research question stated in this study is: "What are the current variables used in statistical models associated to the estimation of software development projects?"

RESEARCH METHODOLOGY

This is a qualitative research article based on formal review of literature.

Criteria to select the literature for this study

The "peer reviewed" articles used in this study were found in the University Library System, specifically in the *Business Source Ultimate* database. The first search was under the key words "software development" and "estimation" within the article subject terms. The second search was under the key words "project management" and "estimation". Articles published between 2014 to present (2017) were considered. This exercise provided enough material to find the factors related to estimation on software development projects.

Criteria to select the variables for this study

We carefully read the description of each statistical model for estimation in software development projects presented in the selected literature. We then underlined the name of the variables used in the statistical models. Since those variables were relevant for estimation, we adopted those for our study by citing its authors, its publication year, and by identifying them as subcategories.

Microsoft Access® (2016) database was used for the classification of the variables, or subcategories, into the seven (7) categories, which represents the considerations for estimation in software development projects. Figure 1 shows the query, or collection of tables, used to classify data (the variables) for the taxonomy of estimation.

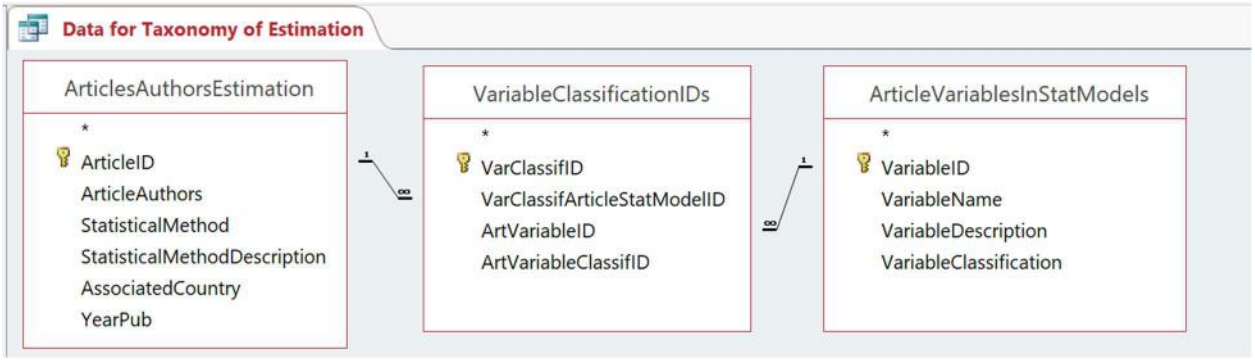

Figure 1. Microsoft Access® Query to classify data for the taxonomy of estimation

Criteria to select the categories for the study

Since many variables for estimation are expected in the selected literature, grouping those variables into categories is a great idea to summarize the concepts that are key in estimation. The categories for this study represent the considerations for estimating in software development projects. Those categories were selected from grouping the variables found in the selected literature by using a form created from the query shown in Figure 1.

FINDINGS

For the first search ("software development" and "estimation"), a total of five (5) articles were found by Khoshgoftaar et al. (2015); Rath, Acharya, & Satapathy (2016); Popovic, Bojic, & Korolija (2015); Siddique & Hussein (2014); Khatibi & Bardsiri (2015). For the second search ("project management" and "estimation"), a total of four (4) articles were found by Uzzafer (2017); Bouras et al. (2016); Datey (2015); Khatibi & Bardsiri (2015). The Khatibi & Bardsiri (2015) article was found in both searches. All articles were directly related to estimation using statistics, except for the Datey (2015) article, which focused on project management software usage.

After revising the selected literature (Bouras, Aqle, Alsardy, & Nizam, 2016, Datey, 2015, Khatibi & Bardsiri, 2015, Khoshgoftaar, Gao, Chen, & Napolitano, 2015, Popovic, Bojic, & Korolija, 2015, Rath, Acharya, & Satapathy, 2016, Siddique & Hussein, 2014, and Uzzafer, 2017), the data found were organized in the Microsoft Access® tables shown in Figure 1. A total of fifty (50) variables on the statistical models analyzed. For the purpose of this study, those variables, or characteristics, are identified as subcategories, which were then classified by using a form constructed from the query presented in Figure 1. Seven (7) groups, or variable classifications, were found. Those seven (7) groups are the key factors in the estimation process, and are identified, in this study, as the categories or considerations for estimation. The names of the categories are estimation cost, project, software, hardware, team members, time and customer. Those categories are the considerations for the estimating in software development projects.

The research question stated in this study is: "*What are the current variables used in statistical models associated to the estimation of software development projects*?" Fifty (50) variables, or subcategories, were found in the selected literature. Since presenting fifty (50) variables would be overwhelming for managers to take into consideration when estimating for software development projects, a classification was performed by organizing data into a Microsoft Access® form. Seven (7) groups or categories were found. Those are the considerations for estimation in software development projects.

Data for Taxonomy of Estimation

VariableClass	VariableName	VariableDescription	StatisticalMethod	ArticleAuthors
Software	changeability	capable of adapting to many uses or	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Project	individual size	the developed software impacts onl	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Team Members	experience programmer	developed by experience programm	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Software	software effort	load of effort to develop software w	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Hardware	concurrency	the number of simultaneous process	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Software	code reusability	the number of times code is used in	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Team Members	special training	the requirement of a special training	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Hardware	distributed system	the number of computers the newly	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Customer	end-user efficiency	how productive is the user as a resu	Random Forest Technique	Rath, Acharya, & Satapathy (2016)
Hardware	installation ease	smoothness of the newly developed	Random Forest Technique	Rath, Acharya, & Satapathy (2016)

Figure 2. Microsoft Access® Query to organize data for taxonomy of estimation (partial view)

Figure 2 shows a partial view of the query that was used to organize data for taxonomy of estimation. The VariableClass refers to the category, the VariableName refers to the variable found in the statistical models, VariableDescription refers to the description for the variable, StatisticalMethod refers to the name of the statistical model used by the authors, and the ArticleAuthors refers to the name of the authors of the article where the variables were found and the publication year.

Table 1 presents the findings of this study in detail. It is a report constructed from the Microsoft Access® query shown in Figure 2. Table 1 includes the category name or consideration for estimation, the subcategory name, the subcategory description, and the author or authors for the subcategory. Subcategories are the estimation characteristics found in literature. Each subcategory was defined in Table 1, as the authors intended. A few of the subcategories were described using online dictionaries. For example, the concept "parallel" is a subcategory name, which was defined according to the Technopidia Dictionary (2017).

Data from Table 1 was analyzed carefully to develop the taxonomy of estimation in software development projects, shown in Figure 1. It shows the seven (7) categories for estimation, which are cost estimation, project, customer, team members, software, hardware, and time. Due to the fact that cost estimation is the main point in the figure, the only subcategories presented in the figure are the ones belonging to it, which are project cash inflow, initial cost, cost variations, project cash outflow, and net cash flow.

DISCUSSION

This study provides new insights that advance the best practices in the estimation process of software development projects. It extends the existing theory by synthesizing the estimation process into the taxonomy of estimation in software development projects presented in Figure 3.

The taxonomy of estimation in software development projects

Figure 3 presents the taxonomy of estimation in software development projects. It clarifies current literature about what considerations are relevant to improve effectiveness when estimating for that kind of project. It identifies seven (7) considerations, or categories, that are significant in estimation. The seven (7) considerations were carefully selected from categorizing the estimation variables found in the statistical models that were evaluated for this study.

Explanation of Figure 3 The taxonomy of estimation in software development projects sketch

Figure 3 is composed of two (2) circles, and a center point. The *center point* is Cost estimation, which is the main consideration. The *inner circle* represents the five (5) variables related to the cost estimation consideration, which are project cash inflows, initial costs, cost variations, project cash outflows, and net cash flow. The *outer circle* shows the rest of the considerations: project, software, hardware, team members, time and customer. The project manager must review both circles periodically, whether it is weekly, every two weeks, etc., depending on the agile technique chosen.

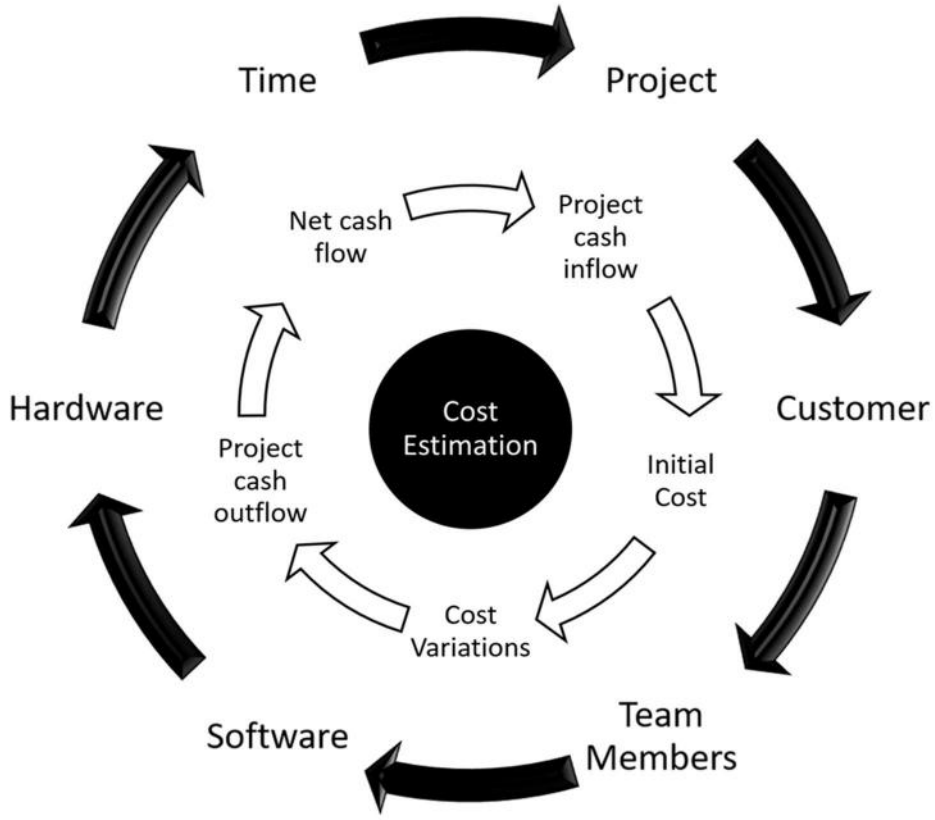

Figure 3. The taxonomy of estimation in software development projects

Description of Table 1 (Appendix) Findings from the study used to develop the taxonomy of estimation in software development projects

Table 1 presents details regarding these considerations, or categories for estimation, the subcategory, or variable name, the variable description, and the variable's authors and year of publication. For instance, *Project* is associated with business goals, capability of analysts, changes in priorities, expert judgment, individual size, industry size, knowledge-based systems, non-technical factors, project effort, project scope, project size, and requirement stability. *Customer* consideration is linked to the variables access, end-user efficiency, functionality, and operational ease. *Team members* are related to experienced programmer, group development, individual development, motivation, special training, and team stability. *Software* is matched to application performance, changeability, code reusability, object-oriented, online data entry, online update, and place of development, portability, programming language difficulty, security features, software effort, and software quality. *Hardware* is related to compatibility with existing systems, complexity of internal processing, concurrency of the processor, data communication, distributed system, platform, rational unified process, and technical aspects. The *time* consideration in estimation is associated to implementation and schedule, while the *Cost estimation* is linked to project cash inflow, initial costs, cost variations, project cash outflow, and net cash flow.

The main contribution of the article

The main contribution of the article is the answer to the research question: "*What are the current variables used in statistical models associated to the estimation of software development projects*?". There are are fifty (50) variables, or subcategories, found in the statistical models of the selected literature. Those fifty (50) variables were grouped by similarities, and are presented in the Subcategory Name column (Variables found in the statistical models) from Table 1. Seven (7) groups or categories came up. Those categories are the considerations, facts or judgement necessary for better estimation process. The considerations are shown in Figure 1, the taxonomy for estimation in software development projects. The taxomomy is relevant for project managers since it summarizes what is significant when estimating for software development projects.

Recommendations for effective estimation on software development projects

The recommendations for effective estimation on software development projects are focused on some of the categories identified, which are project, team members, time, and cost estimation. To be more effective in the estimation, on cost and time, for software development projects, it is advised that the project manager:

Project

- Keep revising estimates as new priorities and insights arise over the time span of the project, by adopting agile project methodologies (Siddique, & Hussein, 2017).
- Identify intellectual property opportunities in each project, as most developed software may become a unique invention and an asset for the organizations (Chen, Bharadwaj, & Goh, 2017).
- Select the project management method that suits the project best. Agile techniques are advantageous for software development projects as those constantly assess the software requirements (Lasek & Adamus, 2014, Schwalbe, 2015, Dawson & Dawson, 2014).
- Use project management software, besides Microsoft Excel, to control the cash flow of the project, setting cash inflows and outflows as milestones (Datey, 2015).
- Constantly perform the project assessment according to the taxonomy of estimation in software development projects, presented in Figure 1. A review of both the inner and the outer circle considerations to guarantee the estimation process is effective.

Team Members

- Motivate the project team, managers, testers, developer and designers, to be involved in the estimation process (Siddique & Hussein, 2014).
- Reward project team members for the knowledge intensive work and promote continuous training to the well-educated workforce (Vasilchenko & Morrish, 2011).

Time

- Must continually assess estimations since the software development project requirement may change over time (Dawson & Dawson, 2014).

Cost estimation

- Include a cost project manager, to perform the usual tasks and to focus on assessing estimates within each agile process circle so that cost is maintained and controlled in the most accurate way.
- Use Scrum, an agile technique, to assess estimations every time the iteration is run (Lasek & Adamus, 2014, and Kaleshovska et al., 2015).
- Maintain historical data from past projects (Bouras et al., 2016), even though that is not the only variable to be considered for estimation due to the fact that most software development projects are distinctive in nature (Chen et al., 2017, Siddique & Hussein, 2014, and Khatibi & Bardsiri, 2015, Bardsiri, Hashemi, & Razzazi, 2015).
- Adopt statistical techniques for estimation (Bardsiri, Hashemi, & Razzazi, 2015), considering relevant estimation variables, presented in Figure 1.
- Plan in conformity to the taxonomy of estimation in software development projects, presented in Figure 1. Table 1 details the variables definitions.
- Use the taxonomy of estimation in software development projects as guidelines to adequately assess estimations effectively.

Conclusion

The task of developing software is knowledge intensive. In most instances, it results in unique inventions, which are susceptible to intellectual property rights (Chen et al., 2017). The process of accurately estimating software development projects is challenging; simply using past data from similar projects for estimation is not advisable because many of these projects are distinctive in nature (Siddique & Hussein, 2014). There are numerous statistical models for estimation purposes, notwithstanding, those models are difficult to be implemented from the managerial standpoint. Datey (2015) proposes adopting project management software to control the cash inflows and outflows in the estimation process. This paper proposes taxonomy with the considerations that are relevant for estimation on software development projects. The considerations are the categories that were found by grouping the variables used in statistical models for estimation. Those considerations are especially valuable in estimation from the project management standpoint. Beyond estimation purposes, the given considerations provide guidelines to plan and execute a software development project.

Implications

When managing software projects, estimations must be verified periodically during the execution phase due to the many uncertainties that existed in the previous phases of the project (Datey, 2015). This study contributes to the academy in two important aspects. The *first aspect* is the taxonomy of the considerations for estimation in software development projects. The *second aspect* is the list of recommendations for estimation in software development projects. These two aspects are relevant suggestions to take into account when estimating for software development projects. Project managers can use the contribution of this paper as guidelines for improved estimation in the planning phase of the project management.

Limitations of the study

This study only considered articles published on the *Business Source Ultimate* university library database. The *Web of Science* library database was not used since it was not available at the university's library. In this study, as well as in many other studies, performing more analysis does not necessarily contribute to new knowledge. The variables obtained from the statistical models started repeating themselves as more articles were evaluated. Therefore, the results presented in this study are still very relevant for the project management of software development endeavors.

Future work

A great proposal for the future would be to develop a questionnaire using the considerations listed in the taxonomy, with the purpose of finding the level of agreement among real project managers involved in software development projects. Another purpose may be discovering new variables for estimation different from the existing ones. That

exercise can help advance this study further. Another future research can focus in comparing the diverse estimation techniques presented by (Siddique & Hussein, 2014).

REFERENCES

Azzeh, M., & Nassif, A. B. (2016). A hybrid model for estimating software project effort from Use Case Points. *Applied Soft Computing*, *49*, 981–989. https://doi.org/10.1016/j.asoc.2016.05.008

Bardsiri, A. K., Hashemi, S. M., & Razzazi, M. (2015). Statistical Analysis of the Most Popular Software Service Effort Estimation Datasets. *Journal of Telecommunication, Electronic and Computer Engineering*, *7*(1), 87–96.

Betz, S., Oberweis, A., & Stephan, R. (2014). Knowledge transfer in offshore outsourcing software development projects: an analysis of the challenges and solutions from German clients (1). *Expert Systems*, *31*(3). https://doi.org/10.1111/exsy.12005

Bouras, A., Aqle, A., Alsardy, A., & Nizam, R. (2016). Ontologies for better project planning. *Journal of Modern Project Management*, *3*(3), 74–79. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84956618013&partnerID=40&md5=9e796167e6496dd1d187d015a27174bc

Bureau of Labor Statistics (2017), U.S. Department of Labor, *Occupational Outlook Handbook, 2016-17 Edition*, Software Developers, on the Internet at https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm (visited *April 16, 2017*).

Calluzzo, V. J., & Cante, C. J. (2004). Ethics in Information Technology and Software Use. *Journal of Business Ethics*, *51*(Computerworld 37), 301–312. https://doi.org/10.1023/B:BUSI.0000032658.12032.4e

Chen, Y., Bharadwaj, A., & Goh, K. Y. (2017). An empirical analysis of intellectual property rights sharing in software development outsourcing. *MIS Quarterly*, *41*(1), 131–161.

Cient, I. E., Roberto, M., Avalia, P., Review, D. B., & Revis, O. J. S. (2014). Risk analysis for use in methods of agile software development project management. *Revista de Gestão E Projetos*, *5*(2), 90–102.

Datey, V. C. (2015). Cash Flow Estimation for Engineering Projects using Project Management Software. *Aweshkar Research Journal*, *XIX*(1), 95–109.

Dawson, C., & Dawson, R. (2014). Software Development Process Models : A Technique for Evaluation and. *Knowledge and Process Management*, *21*(1), 42–53. https://doi.org/10.1002/kpm

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, *127*, 109–119. https://doi.org/10.1016/j.jss.2017.01.027

Dutta, S., Geiger, T., & Lanvin, B. (2015). The global information technology report 2015. World Economic Forum (Geneva). Retrieved April 26, 2016 from http://www3.weforum.org/docs/WEF_Global_IT_Report_2015.pdf

"estimate, n." Business Dictionary (2017). Web. 17 April, 2017. http://www.businessdictionary.com/definition/estimate.html

Idri, A., Hosni, M., & Abran, A. (2016). Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles. *Applied Soft Computing*, *49*, 990–1019. https://doi.org/10.1016/j.asoc.2016.08.012

Kaleshovska, N., Josimovski, S., Pulevska-Ivanovska, L., Postolov, K., & Janevski, Z. (2015). The contribution of Scrum in managing successful Software Development Projects. *Economic Development*, (1–2), 175–194.

Kärreman, D. (2010). The Power of Knowledge: Learning from Learning by Knowledge Intensive Firm. *Journal of Management Studies, 47*(7), 1405-1416.

Khatibi, E., & Bardsiri, V. K. (2015). Model to estimate the software development effort based on in-depth analysis of project attributes. *IET Software*, *9*(4), 109–119. https://doi.org/10.1049/iet-sen.2014.0169

Khoshgoftaar, T. M., Gao, K., Chen, Y., & Napolitano, A. (2015). Comparing Feature Selection Techniques for Software Quality Estimation Using Data-Sampling-Based Boosting Algorithms. *International Journal of Reliability, Quality and Safety Engineering*, *22*(3). https://doi.org/10.1142/S0218539315500138

"knowledge, n." Business Dictionary (2017). Web. 17 April, 2017. http://www.businessdictionary.com/definition/knowledge.html

Lasek, M., & Adamus, A. (2014). When it is worth to use Agile Methodologies in Software Development Project Management practice? *Business Informatics*, *1*(31). https://doi.org/10.15611/ie.2014.1.13

Microsoft Access®, 2016. Microsoft Access 2016 for Windows. Redmond, WA: Microsoft Corporation.

Nassif, A. B., Azzeh, M., Capretz, L. F., & Ho, D. (2016). Neural network models for software development effort estimation: a comparative study. *Neural Computing and Applications*, *27*(8), 2369–2381. https://doi.org/10.1007/s00521-015-2127-1

"parallel, n." Technopidia Dictionary (2017) Web. 20 April 2017. https://www.techopedia.com/definition/8777/parallel-computing

Popovic, J., Bojic, D., & Korolija, N. (2015). Analysis of task effort estimation accuracy based on use case point size. *IET Software*, *9*(6), 166–173. https://doi.org/10.1049/iet-sen.2014.0254

Project Management Institute (2013). *A guide to the project management body of knowledge: PMBOK® guide, Fifth Edition (5/E)*. Newtown Square, Pa: Project Management Institute.

Project Management Institute (2016). What is Project Management? Project Management Institute. Retrieved April 26, 2016, from http://www.pmi.org/en/About-Us/About-Us-What-is-Project-Management.aspx

Rath, S. K., Acharya, B. P., & Satapathy, S. M. (2016). Early stage software effort estimation using random forest technique based on use case points. *IET Software*, *10*(1), 10–17. https://doi.org/10.1049/iet-sen.2014.0122

Recker, J., Holten, R., Hummel, M., & Rosenkranz, C. (2017). How Agile Practices Impact Customer Responsiveness and Development Success: A Field Study. *Project Management Journal*, *48*(2), 99–122.

Sandoval, M. del R. O. L., & Ruiz Ascencio, J. (2016). MUREM: A Multiplicative Regression Method for Software Development Effort Estimation. *Computación Y Sistemas*, *20*(4).

Satapathy, R., Kumar, M., & Rath, S. K. (2016). Optimised class point approach for software effort estimation using adaptive neuro-fuzzy inference system model. *International Journal of Computer Applications in Technology*, *54*(4). https://doi.org/http://dx.doi.org/10.1504/IJCAT.2016.10001320

Schwalbe, K. (2015). *An introduction to project management, Fifth Edition (5/E).* Minneapolis, MN: Schwalbe Publishing. ISBN-13: 978-1505212099.

Siddique, L., & Hussein, B. A. (2014). Practical Insight into estimation process in software projects using Agile

Methdos. In *Third International Scientific Conference on Project Management in the Baltic Countries* (pp. 272–284). Riga, Latvia.

Subramanian, G. H., Pendharkar, P. C., & Pai, D. R. (2017). An Examination of Determinants of Software Testing and Project Management Effort. *Journal of Computer Information Systems*, *57*(2), 123–129. https://doi.org/10.1080/08874417.2016.1183428

"taxonomy, n." Merriam-Webster (2017) Web 20 April 2017. https://www.merriam-webster.com/dictionary/taxonomy

Uzzafer, M. (2017). Cost estimation of software projects: A sub-additive approach. *Journal of Modern Project Management*, (April), 86–97. https://doi.org/10.19225/JMPM01210

Vasilchenko, E., & S. Morrish, (2011). The Role of Entrepreneurial Networks in the Exploration and Exploitation of Internationalization Opportunities by Information and Communication Technology Firms. *Journal of International Marketing, 19*(4), pp. 88105.

Velarde, H., Santiesteban, C., Garcia, A., & Casillas, J. (2016a). Analyzing the Effect of Variables in the Software Development Effort Estimation. *IEEE Latin America Transactions*, *14*(8). https://doi.org/10.1109/TLA.2016.7786366

Velarde, H., Santiesteban, C., Garcia, A., & Casillas, J. (2016b). Software Development Effort Estimation based-on multiple classifier system and Lines of Code. *IEEE Latin America Transactions*, *14*(8).

APPENDIX

Table 1. Findings from the study used to develop the taxonomy of estimation in software development projects

	Category name (Consideration for Estimation)	Subcategory Name (Variables found in statistical models)	Subcategory Description	Authors
1	Customer	access	provide access to third parties	Rath, Acharya, & Satapathy (2016)
2		end-user efficiency	how productive is the user because of deploying the newly developed software	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
3		functionality	the newly developed software can perform what it is supposed to do free of errors	Khoshgoftaar, Gao, Chen, & Napolitano (2015), Popovic, Bojic, & Korolija (2015), Siddique & Hussein (2014)
4		operational ease	smoothness of the newly developed system to be used in daily operations, able to function easily	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
5	Hardware	compatibility with existing systems	how well the newly developed software will interrelate with legacy systems, if any	Siddique & Hussein (2014)
6		complexity of internal processing	the amount of resources to needed to complete a task	Rath, Acharya, & Satapathy (2016), Siddique & Hussein (2014)
7		concurrency	the number of simultaneous process software handles	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
8		data communication	how the newly developed software handles the sharing of data among the whole system	Uzzafer (2017)
9		distributed system	the number of computers the newly developed software allows to be linked geographically, multiple sites	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
10		installation ease	smoothness of the newly developed system to be ready for use	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
11		platform	the group of technologies, applications and processes, where the newly developed software will operate	Khatibi & Bardsiri (2015)
12		rational unified process	stable joint procedures	Rath, Acharya, & Satapathy (2016)
13		technical aspects	the general environment where the newly developed software will operate	Siddique & Hussein (2014)
14	Funds	cost variations	the real cost after the execution of the project. It includes notes on the variations.	Bouras, Aqle, Alsardy, & Nizam (2016)
15		initial cost	the cost perceived at the initial phase of the project	Bouras, Aqle, Alsardy, & Nizam (2016)
16		net cash flow	the difference between project cash inflow and project cash outflow that arises periodically during the software development project	Datey (2015)
17		project cash inflow	the cash that is incoming periodically during the software development project	Datey (2015)
18		project cash outflow	the cash that is paid periodically during the software development project	Datey (2015)
19	Project	business goals	how well the newly developed software achieves the business goals of the organization	Siddique & Hussein (2014)
20		capability of analysts	The ability to examine the system's requirement	Rath, Acharya, & Satapathy (2016)
21		changes in priorities	while creating the software, new insights or requirements may come up with time	Siddique & Hussein (2014)
22		expert judgment	The knowledge and the experience of the professional involved in the project. It avoids many unpredictable errors during the completion of the project.	Popovic, Bojic, & Korolija (2015), Siddique & Hussein (2014), Uzzafer (2017)
23		individual size	the developed software impacts only few business divisions	Khoshgoftaar, Gao, Chen, & Napolitano (2015), Popovic, Bojic, & Korolija (2015), Rath, Acharya, & Satapathy (2016)
24		industry size	the developed software impacts industry	Khoshgoftaar, Gao, Chen, & Napolitano (2015), Popovic, Bojic, & Korolija (2015), Rath, Acharya, & Satapathy (2016)
25		knowledge-based system	how well the software development project contributes to future project plans on estimations accuracy	Bouras, Aqle, Alsardy, & Nizam (2016)
26		non-technical factors	the other factors in the organization, not related to technical issues, that may arise during the project	Siddique & Hussein (2014)

	Category name (Consideration for Estimation)	Subcategory Name (Variables found in statistical models)	Subcategory Description	Authors
27		project effort	the energy that is applied to complete the job successfully	Popovic, Bojic, & Korolija (2015), Siddique & Hussein (2014)
28		project scope	it is what the newly developed software is expected to perform. It is the main purpose of the project.	Bouras, Aqle, Alsardy, & Nizam (2016), Popovic, Bojic, & Korolija (2015), Siddique & Hussein (2014)
29		project size	the overall magnitude of the project	Khoshgoftaar, Gao, Chen, & Napolitano (2015), Popovic, Bojic, & Korolija (2015), Rath, Acharya, & Satapathy (2016)
30		requirement stability	well-defined characteristics that are expected to be performed every time the newly developed software is in use	Popovic, Bojic, & Korolija (2015), Rath, Acharya, & Satapathy (2016), Siddique & Hussein (2014)
31	Software	application performance	response or throughput referring to the amount of coding lines processed within a given time	Rath, Acharya, & Satapathy (2016), Uzzafer (2017),
32		changeability	capable of adapting to many uses or applications	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
33		code reusability	the number of times code is used in different procedures	Rath, Acharya, & Satapathy (2016), Uzzafer (2017)
34		object-oriented	whether the software implemented is object-oriented, or modern	Rath, Acharya, & Satapathy (2016)
35		online data entry	all the written lines of codes required while developing the software	Uzzafer (2017)
36		online update	how much effort is needed every time a change of software requirement arises	Uzzafer (2017)
37		place of development	software is developed in an organization such as industry, government, or laboratory	Khatibi & Bardsiri (2015), Khoshgoftaar, Gao, Chen, & Napolitano (2015)
38		portability	how easy the newly developed software is transferred from one computer environment to another. Environment includes operating systems, hardware, other software, users, and programmers, to another.	Rath, Acharya, & Satapathy (2016)
39		programming language difficulty	coding complexity	Rath, Acharya, & Satapathy (2016)
40		security features	cryptographic capabilities	Rath, Acharya, & Satapathy (2016)
41		software effort	load of effort to develop software with specific requirements, sometimes measured in hours	Bouras, Aqle, Alsardy, & Nizam (2016), Khatibi & Bardsiri (2015), Popovic, Bojic, & Korolija (2015), Rath, Acharya, & Satapathy (2016), Siddique & Hussein (2014), Uzzafer (2017)
42		software quality	software free of errors after testing a wide range of data samplings	Khatibi & Bardsiri (2015), Khoshgoftaar, Gao, Chen, & Napolitano (2015), Popovic, Bojic, & Korolija (2015), Siddique & Hussein (2014)
43	Team Members	experience programmer	developed by experience programmers, not beginners	Khoshgoftaar, Gao, Chen, & Napolitano (2015), Rath, Acharya, & Satapathy (2016), Siddique & Hussein (2014)
44		group development	working in groups of people	Bouras, Aqle, Alsardy, & Nizam (2016), Khoshgoftaar, Gao, Chen, & Napolitano (2015), Siddique & Hussein (2014)
45		individual development	working as stand-alone person	Khoshgoftaar, Gao, Chen, & Napolitano (2015)
46		motivation	programmer's energy and reason to perform the job	Rath, Acharya, & Satapathy (2016)
47		special training	the requirement of a special training to use the newly developed software	Rath, Acharya, & Satapathy (2016)
48		team stability	the possibility that the group of programmers' changes before the completion of the project impacting productivity	Siddique & Hussein (2014)
49	Time	implementation	the process of installing the newly developed systems	Popovic, Bojic, & Korolija (2015)
50		schedule	how much time is available against the work load of completing the project scope (software)	Siddique & Hussein (2014)