Journal of Quality Engineering and Production Optimization

Vol. 7, No. 1, Winter & Spring 2022

http://jqepo.shahed.ac.ir

Research Paper

DOI: 10.22070/JQEPO.2022.15346.1214

An Integrating Machine Learning Algorithm and Simulation Method for Improving Software Project Management: A Case Study

Ayria Behdinian[1], Mohammad Amin Amani[1], Amir Aghsami[1,2], Fariborz Jolai[1*]

[1] *School of Industrial and Systems Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[2] *School of Industrial Engineering, K. N. Toosi University of Technology (KNTU), Tehran, Iran*

***Corresponding Author:** Fariborz Jolai (E-mail: fjolai@ut.ac.ir)*

Abstract *– Project managers analyze the factors that affected projects' success, signifying performing a project within the scopes (Time, Quality, and cost) defined in the initial step. The implication of each factor on project success is essential since several of them have been specified in this area. Employing all of them is not feasible, and it may impose outrageous expenses on the organizations. Therefore, this article aims to identify the factors that impact project accomplishment and pinpoint the most contributing factors to facilitate the project's implementation. The main contribution of this paper is representing a framework by combining Machine Learning algorithms and simulation models to detect the effectiveness of leading organizational factors on project accomplishment, beneficially leading to extracting the accurate analysis. A logistic regression algorithm was employed to build a predictive model. The predictive model was created based on independent variables to predict whether the software project would be successful or fail. Also, Gamification was determined as the most influential factor on the objective by the Logistic regression feature importance method. Then, Gamified and non-Gamified models were compared by the Simulation method and showed Gamification made a 36.26% improvement in the rojects cycle time and a 15% enhancement in the quality of employers' performance by decreasing the projects' bugs. For validating the simulation results, some projects were implemented in the real case study, and the model results clarified the Gamification potential in improving employee engagement leading to better work progress tracking and higher performance quality.*

Keywords*– Software Project Management, Machine learning, Simulation, Gamification*

MOTIVATION

Over the years, various methods in software project management have been presented and assessed; however, the vast majority of organizations are not profit-driven industries. The actual case (Software company) that we analyzed in this research dealt with financial obstacles due to project management inefficiency. Furthermore, there was no effective and efficient cooperation in order to carry out projects in the organization. Hence, utilizing approaches or changing factors to enhance the organization's performance can significantly impact workforce productivity in the firm mentioned above. Therefore, for better project management action, essential factors in the organization can be evaluated. The primary motivation for expanding this framework is to examine low-cost tools such as gamification elements for the software organization along with other features such as the skills of employees or project management support to analyze the impact of gamification, which has been considered in this research along with fundamental conventional project management factors.

Manuscript Received:28- Dec -2021 & Revised: 20- March -2022 & Accepted: 30- April -2022
ISSN: 2423-3781

I. INTRODUCTION

A software project includes software development from the earlier steps, such as requirements gathering, and afterward, it continues to the testing and maintenance phase. It is essential to manage software production efficiently to successfully carry out the project by considering the time, quality, and cost.

Project managers apply various methods to perform the tasks productively due to software projects' complexity (Hsieh et al., 2018). Project management procedures have changed considerably since handling the software projects' concludes the problem of resources management and scheduling, and the issue of the entire project life cycle should be taken into account to perform the project productively (Hamzeh et al., 2020). Plenty of software process models have been defined, namely Waterfall, Prototyping, Incremental, spiral, agile, scrum, etc., attempting to boost the project life cycle's productivity and quality. However, a broad number of projects still fail to meet the basic requirements considering time delays, cost overruns, and quality limitations (Gitinavard et al., 2020).

Companies have utilized Numerous elements to make progress, most of which are related to the human resource knowledge area (Niazi et al., 2016). The related knowledge, tools, and techniques are lacking in software process improvement; therefore, various methods should be applied to analyze software project management enhancement (Cerdeiral and Santos, 2019). Planning is a contributing factor in the project execution; hence every aspect of the project should be defined conspicuously in this phase. All the elements that affected the process are supposed to be empirically appraised to be weighed, and their effectiveness can be scrutinized (ul Hassan et al., 2018).

Multiple features seem to affect the managing process, such as project planning and scheduling, team collaboration, time tracking, and reposting; thus, project managers take advantage of procedures to enhance performance during the project steps, called a change management model for software project monitoring integrated with Earned Value Management (EVM). The model above paves the way to forecast the changes and subsequent reworking; furthermore, estimating project progress at any time could be applicable (Efe and Demirors, 2019). Many determining factors affect the Software projects' development; therefore, it is essential to consider all the contributing elements. Requirements uncertainty has to be analyzed for involved factors in the projects (Gitinavard and Mousavi, 2015); it is identified as the most noticeable factor since it has a considerable impact on the budget, time, and quality of the products (Haleem et al., 2021).

Some methods such as machine learning, simulation, parametric models, expert estimation, etc., can be used to analyze and estimate different factors and objectives in software projects and help to improve the project's performance.

A. Machine Learning

Machine learning is a data science method to find patterns in amounts of data, which can be utilized for different objectives such as project success, estimate costs, and other targets based on independent variables. Machine learning techniques have been used in prediction endeavors for many years (Amirkhalili et al., 2020). Pospieszny et al. (2018) used Artificial Neural Network (ANN), Support Vector Machine (SVM), and Generalized Linear Models (GLM), which are machine learning algorithms for software project's effort and duration estimation. Han et al. (2015) deployed machine learning algorithms to predict software project time. Masoud et al. (2017) employed a clustering method that is an unsupervised machine learning algorithm to cluster the software project into four classes based on resources and choose the project into the class with equal or less than the enterprises' resources. Bibi and Stamelos (2006) used machine learning algorithms for the software development cost prediction. Mahdi et al. (2020) utilized machine learning for software project risk assessment to reduce the project's fault. Rashid et al. (2012) employed a machine learning algorithm for software quality estimation.

B. Simulation

Evaluating the utilized methods Gives great value to research (Gitinavard, 2019); Simulation is one of the functional tools to evaluate Software processes (Liu et al., 2009). It is an effective way to verify and validate the organization's predicted model (França and Travassos, 2013). Carrying out the usefulness of the simulation model in an actual situation is highly important; therefore, it is essential to formulate the model in a simulated area to provide reports of the objectives before performing in real industrial environments (Kolkman, 2020).

The simulated model provides a situation to appraise the changes in different variables and design alternatives without the exorbitant cost of implementation in an actual condition. Even though these models were used in industrial zones on large scales, it is now effectively applied in software organization (García-García et al., 2020). When the required data is extracted from different field studies, it can be run multiple times to specify the effectiveness of various factors and increase the proposed method's credibility (Liu et al., 2020).

Project managers have a vast majority of problems after launching the project, and one of those obstacles in software areas is risk management. Liu et al. (2009) introduced the simulated models to decrease risk in a software project. Zhang et al. (2008) Mentioned that hybrid process simulation models had captured people's attention due to being more analogous to the real-world situation. The blended simulation model of System Dynamic and Discrete-event Simulation can genuinely affect the software development process. Many simulation models in software project management should be analyzed to concern their merits and demerits in the functional area (Dashti et al., 2021).

Bassil (2012) attempted to simulate a Waterfall method within a software organization in order to iron out the deficiencies and problems in terms of human resources productivity and indeed concerned the delay time reasons. Many management and engineering factors, such as the customer's requirements, should be concerned in developing the simulated model (Rus et al., 2014). The crowdsourcing method has correctly been added to the simulated model by Saremi and Yang (2015) to estimate its benefits and avoid the errors it may bring about.

By introducing state-of-the-art software development methods, it is required to estimate these approaches' functionality, one of which is named agile hardware design, which was analyzed by Amid et al. (2020) and they showed it is workable in alleviating the design costs. Including as many relevant elements as possible in a simulated model is valuable for analyzing the results. Therefore Dugarte Peña et al. (2020) employed many intervening factors in analyzing, designing, coding, and testing the software products, and finally, the initial estimation of the software project's time was feasible.

C. Gamification

One of the methods associated with software project management is Gamification, which is highly applicable in the process integrated with human resources to increment software projects' productivity. Besides, it can positively affect the project's result and facilitate the implementation of practices that impact software lifecycle achievement (Machuca-Villegas and Gasca-Hurtado, 2018a).

Gamification is the use of various game elements in non-game contexts (Deterding et al., 2011). Integrating play and work in a workplace has gotten attention in recent years since many managers are looking forward to involving employees in different aspects of the projects, and the premier way for performing this is by utilizing Gamification elements (Najjar et al., 2021). Gamification includes points, leaderboards, and badges, to name but a few, to motivate both the employees and customers (Rodrigues et al., 2019). According to Deterding et al. (2011), the term Gamification was broadcast on mass media in 2010 when it was defined as a valuable tool to the industrial zones. In 2012, Gamification turned out to be a strategic tool that can positively affect the organization's productivity, and it is proven that employees' engagement is positively shifted by employing this tool in the work situation (Singh, 2012). In addition to higher productivity in the organization, it has been shown that people involved in Gamification elements are more satisfied with what they do and positively it impacts people’s minds (Xi and Hamari, 2019).

The software development area was no exception, and gamification techniques were successfully included in software development from the initial steps to the final stage, deployment, to engage and motivate the employees (Moldon et al., 2020). Gamification elements' usefulness in a workplace to achieve organization targets and perform the project perfectly considering the budget, time, and quality is mentioned by Perryer et al. (2016). The feeling of workers about applied Gamification tools in the workplace has an undeniably direct correlation with the success of the projects; accordingly, Meder et al. (2013) performed research to demonstrate the effectiveness of these tools, and finally, it was concluded that a small number of isolated gamification concepts such as team competition, leaderboards, point systems, and badges could magnificently stimulate the workforces.

It is mentioned that a broad number of case studies showed that software project management could negatively affect the success of software development projects. Thus, it is readily apparent that employing some other methods must be intervened with project management to improve the process. The practical strategy to answer this problem is a gamified system (Machuca-Villegas and Gasca-Hurtado, 2018b).

Machuca-Villegas and Gasca-Hurtado (2018a) indicated that the most common area in project management is human resources with a frequency of 38.8%; hence, it is apparent that the integration of Gamification with the resources results in promoting collaboration among them which can be a crucial point in the software projects success. Apart from this area, the elements applied by several studies have a wide range, even though the most functional one is the point systems. Featherstone and Habgood (2019) mentioned real-world reward as a critical element in Gamification to motivate individuals to engage in the project voluntarily. Muszyńska (2020) referred to Gamification as a functional method to develop communication among the employees to provide high-quality products.

Various components have been introduced by Werbach and Hunter (2014) to be employed in different situations, such as teams, achievements, gifts, points, mission, social graphs, tables of classification, etc. One of Gamification's underlying factors is defining the same goal for members, and the feasibility of this is related to forming teams to collaborate and make progress (Koivisto and Hamari, 2019). According to Spanellis et al. (2020), setting up teams in an organization requires scrutinizing the knowledge and interaction level, to mention but a few, to implement the process of Gamification functionally.

A real-practice software case obstacle has been considered, and its principal problems have been recognized to be addressed. One of the difficulties in this organization is switching to a profitable method fastly without requiring a significant culture shift. Secondly, the lack of involvement in the organization may cause failure in the project; hence, project managers in this company are looking forward to ironing out this hurdle. The other substantial component is mainly related to the employees' optimum performance level since although many projects have been executed successfully, the organization's utilization is low. The project manager in the case study is looking forward to solving these issues to boost the company's efficiency, productivity, and working conditions. For this reason, previous studies were reviewed, which is summarized in Table 1.

Table I. Summarized literature review

Article	***Factors***					***Simulation***	***Machine learning***	***Case study***
	TS*	***PS****	***M****	***CCL****	***G****			
(Lalsing et al., 2012)	✓		✓					✓
(Serrador and Pinto, 2015)			✓	✓				✓
(Han et al., 2015)			✓				✓	
(Masoud et al., 2017)	✓		✓				✓	
(Machuca-Villegas and Gasca-Hurtado, 2018a)	✓	✓			✓			
(Rodrigues et al., 2019)	✓			✓	✓			
(Muszyńska, 2020)	✓	✓			✓		✓	
(Amid et al., 2020)	✓		✓			✓		

(García-García et al., 2020)	✓	✓	✓	✓		✓		
(Najjar et al., 2021)	✓		✓		✓			
(Li et al., 2021)	✓		✓	✓		✓		✓
This Research	✓	✓	✓	✓	✓	✓	✓	✓

***Note: CCL= Customer collaboration level, PMS= Project manager support, TS=Team skill, M=Method (agile, traditional),G=Gamification**

Due to Table 1, the research gap in this area can be considered by examining the factors together and determining the value of each factor. In addition, most of the surveys used a questionnaire for analysis, While methods such as machine learning are less utilized in this field. Various studies have cited success factors; However, the extent of their impact on the organization is less clear, which is applicable to employing simulation models. Applying project factors and the framework (Machine Learning and Simulation models) in a real case has not been taken into consideration in the field of Software Project Management.

In this paper, according to mentioned gaps, After determining the organization issues, a predictive model will be built by a Machine learning algorithm that can predict whether the project will be successful or fail based on the independent variables, which are Team skills, Customer collaboration, the project's manager support, using the agile or traditional method, and Gamification, and then the independent variable that has the most impact on the objective will be determined in order to address the organization problem mainly related to picking out profit-driven method and factors with higher performance and enhancing employees' involvement. Then the simulation model will be utilized to evaluate the improvement of the projects based on applying the most influential variable. A brief summarization of this paper's contributions is mentioned as follows:

- Applying a combination of machine learning and simulation methods to improve software project management factors(Time, Quality)
- Comparison of the factors influencing software project's accomplishment utilizing machine learning algorithm
- The effects of gamified approach on a software organization's productivity implementing a simulation model
- Employing Gamification elements (Team building, Leaderboard, Points) to enhance software project management
- The improvement of the software project's time cycle and quality by applying Gamification elements

The rest of the paper is organized as follows. Section 2 expresses the problem definition and the factors used to build the predictive and simulation models. Section 3 defines a Logistic regression algorithm which is a Machine learning algorithm and simulation method. In section 4, experimental results are presented. Section 5 expresses the conclusion.

II. PROBLEM DEFINITION

Software projects' success is directly correlated to meeting the time, quality, and cost defined at the projects' early steps. Therefore, project managers attempt to employ the factors and methods that could significantly improve the triangular element mentioned above (Davoudabadi et al., 2019). The alteration in the conventional organizations' system required a great deal of budget, effort, and harmonization; consequently, the actual case faced budget deficiency for this change. According to this issue, project managers are looking forward to opting for the crucial elements to improve the organization's performance. The company can extract the functional elements at a low cost along with higher accuracy in employing project manager factors by exploiting machine learning algorithms. A machine learning algorithm will be employed to build a predictive model based on independent variables to predict the goal, predict whether the project will be successful or fail, and analyze which independent variables have the most effects on the dependent variable. Then by a simulation model, the most influential independent variable's impact will be evaluated on the project's success. Fig 1 illustrates the problem and solving process for this research, in which the dataset is made based on an analysis of the projects by the project manager, and then machine learning algorithms are utilized to build a classifier, and the feature importance technique determines the influential factors. In the next step, simulation modeling is employed to analyze the impact of the crucial element on project success.

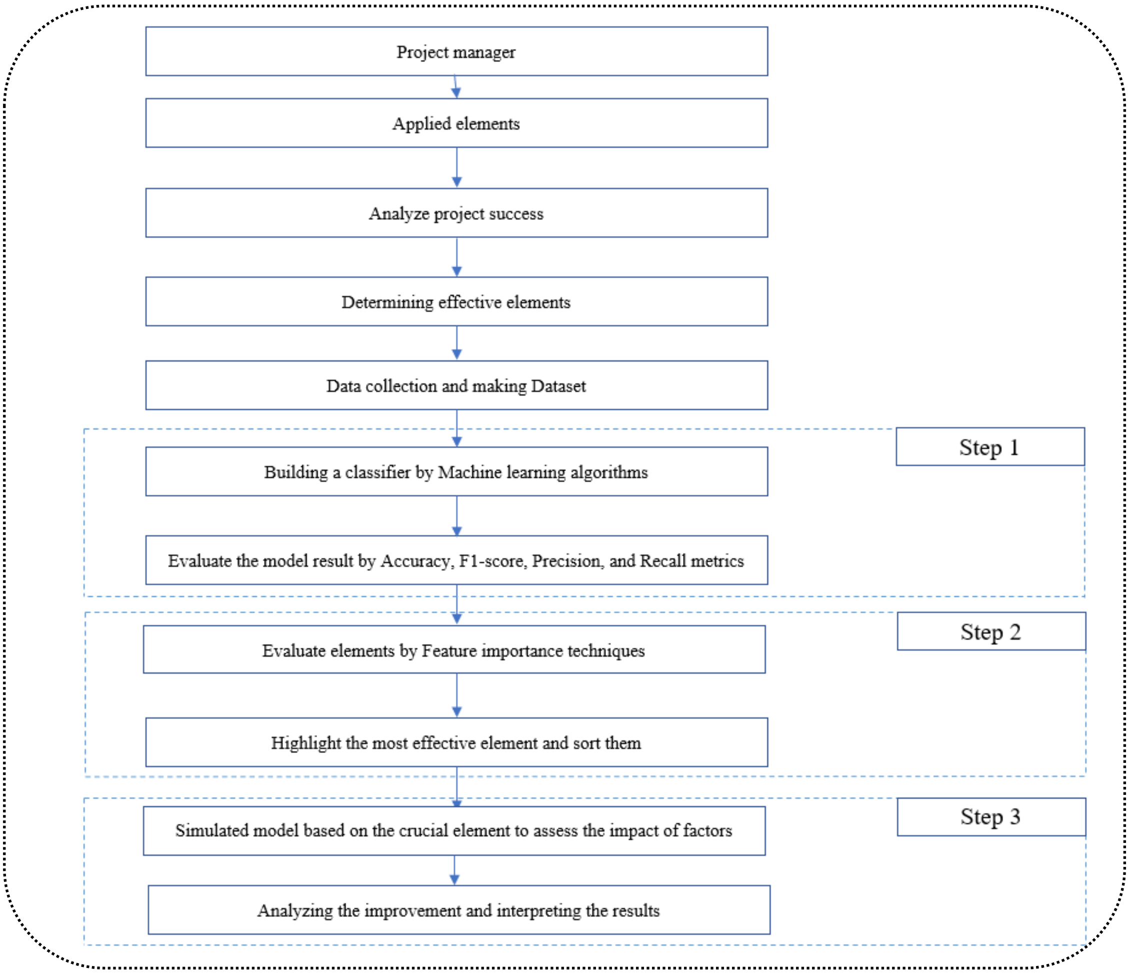

Fig 1. The Problem And Solving Process

For implementing the simulation method, it is highly effective to apply the real case data to analyze the actual situation's improvement for implementing the simulation method. Hence, according to that organization, the company's software development process and all the related variables are utilized to increment the model's credibility.

A. Case study

This study used a merged dataset combining relevant information from the workforce and projects by the Donya Pardazesh Abri Occupational informatics center. The database provides all information regarding the variables mentioned above and project results. The data of a software manufacturing company is extracted to analyze individual items. The company's projects revolve mainly around generating new software products related to Web-based tasks comprising some layers, namely, Analyze, Design, Programming, Quality Assurance, Documentation, and Deployment. The Software Company employed Jira software that is perfectly fitted to software-related tasks and paves the way for the managers to monitor software development (Fisher et al., 2013). The company, as mentioned above, has performed several projects;

however, for different aspects of this article, twelve developers, one designer, one analyzer, and two quality assurance have been appointed, and their projects have been analyzed. The project's success is coordinated to the total project time, the total cost of the project, quality of the software products, all of which have been elucidated in the initial phase.

B. Factors

Throughout the years, many factors have been defined as contributing items to the project's success. For managing the software development process, agile project management was applied in several companies; therefore, analyzing the agile vs. traditional method and its effectiveness on the project's success is remarkable (Lalsing et al., 2012). Customer collaboration with employees to advance the projects is another crucial factor influencing the project's success, and it is different according to the customer personality and project's scales (Serrador and Pinto, 2015). Even though the agile method proposes the customers engagement, it is not feasible for every project. For this specific case, in this paper, customer collaboration is grouped into three levels (Low, Average, High) concerning the time customers have been involved in analyzing the project.

There is an undeniably direct correlation between Team skills and projects' success; hence it should be considered an underlying element in every software project (Akgün, 2020). Supporting the team by project management empowers the teams in various projects (Sulaiman and Mansor, 2018). Employing Gamification elements in a company with the other factors to examine its influence on productivity is highly recommended in a software project since it considerably affects its results (Machuca-Villegas and Gasca-Hurtado, 2019). All the factors mentioned above can affect the project differently; thus, it is essential to examine their effects on the software project's success. The variables used by machine learning to build the predictive model and feature importance are mentioned in Table 2.

Table II. Selected Variables For Project Success Prediction

Variable	*Type*	*Value*	*Role*
Team skill	Categorical	High, Average, Low	Independent
Project manager support	Numerical	1, 0	Independent
Method	Categorical	Agile, traditional	Independent
Customer collaboration level	Categorical	High, Average, Low	Independent
Gamification	Numerical	1, 0	Independent
Project success	Numerical	1, 0	Dependent

C. Assumptions

According to the data, we considered assumptions for the implementation of the framework. Concerning project management support and the use of game elements, the decision is 0 and 1, which signifies that these factors can not be partially implemented. Regarding Customer collaboration and team competencies, three levels are considered for executing the models. According to the workload and phases defined for implementing every project in the organization, Agile and traditional which were utilized for plenty of projects, were assessed in this research. Even though there are multiple procedures to implement Agile, the scrum method is considered in this investigation.

III. METHODOLOGY

In this section, the Logistic regression algorithm, which is a Machine learning algorithm, will be expressed to see how it can be utilized to build the predictive model and find the most influential independent variable which will be used in the simulation model to find its impact on project improvement, then simulation method and its input and output variables will be expressed.

A. Machine learning

Supervised learning is one of three machine learning branches, which has two parts, namely, 1) Regression and 2) Classification. In classification, a class label is predicted based on independent variables. One of the most useful Classification algorithms is Logistic regression. Fig 2 illustrates the supervised learning process for this research where, in the data preprocessing part, the One-hot encoding will be employed to convert the categorical variables to numerical and, in the algorithm selection section, the Decision Tree classifier, Random Forest classifier, multi-layer perceptron (MLP) classifier which is a reliable and straightforward class of feed-forward ANN (Shariati et al., 2020), and Logistic regression classifier will be compared based on the accuracy to find the better classifier for the problem.

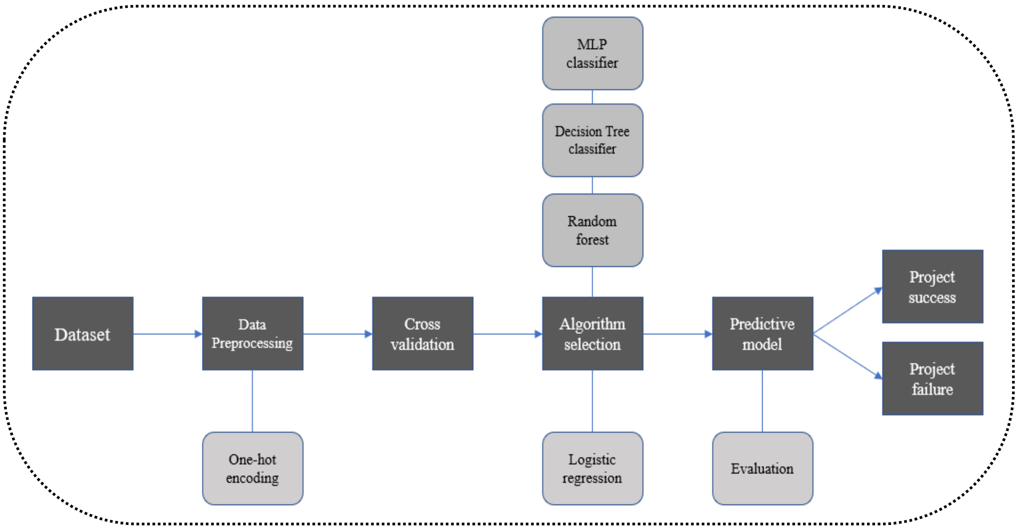

Fig 2. The Supervised Learning Process For This Research

Due to the statistic view, a function is linear if the parameters are linear (Amani et al., 2021). Logistic regression results depend on the sum of the inputs and parameters; therefore, Logistic regression is counted as a generalized linear model. The Linear regression hypothesis is as follows:

$$Z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_n X_n \tag{1}$$

Where θ is coefficient and X is the independent variable. In Logistic regression, the goal is to map the predicted values to probability which means mapping the values to a value between 0 and 1; for this work, the Sigmoid function is utilized. The Sigmoid function is as follows:

$$f\ (x) = \frac{1}{1 + e^{-(X)}} \tag{2}$$

By applying the Sigmoid function to the Linear regression hypothesis, the Logistic regression is made, which is mentioned in equation (3):

$$h_\theta(X) = \frac{1}{1 + e^{-(\theta^T X)}} \tag{3}$$

Where θ is coefficient and X is the independent variable. The threshold is chosen 0.5, which means if the probability calculated by equation (3) is equal to or bigger than the threshold ($h_\theta(X) \geq 0.5$), the classifier classified it to class 1, and if the probability is less than the threshold ($h_\theta(X) < 0.5$) the classifier classified it to class 0 .To find the best hyperplane to separate the classes and estimate the parameters, the logistic regression cost function should be minimized. The logistic regression cost function is as follows:

$$Cost\ (\ h_\theta(X)\ ,y)\ =\ \begin{cases} -\ log\ (h_\theta(X)) & if\ y\ =\ 1 \\ -\ log\ (\ 1 -\ h_\theta(X))\ & if\ y\ =\ 0 \end{cases} \tag{4}$$

A condensed version of the Logistic regression cost function, which is made of equation (4), can be formulated as follows:

$$J(\theta)\ =\ -\frac{1}{m}\sum_{i=1}^{m}[\ y^{(i)}log\ (h_\theta(X^{(i)}))\ +\ (1\ -\ y^{(i)})log\ (\ 1-\ h_\theta(X^{(i)}))] \tag{5}$$

There are several methods for validating the machine learning algorithms such as Train/test split, k-fold Cross-Validation (CV), Leave-one-out Cross-Validation (LOOCV), Leave-one-group-out Cross-Validation (LOGOCV), Nested Cross-Validation (NCV), Wilcoxon signed-rank test, Mc Nemar’s test, etc. among these methods, k-fold Cross-validation is one of the most applied methods which is utilized in this research too.

B. Simulation

After picking the gamification tool out as the most influential factor in project success, it is essential to evaluate its implications on the software company's performance. Two simulation models with the same inputs have been developed to make the assessment feasible. Even though five discrete factors, namely, team skill, project management support, agile approach, gamification tools, and customer collaboration level, have been examined by employing Machine learning, only one factor would be altered in both models: Gamification, since it is the most contributing element in project success. For simplicity and taking advantage of the models, just a module can be changed in a simulated system to pave the way for organizations to utilize that specific element to improve their performance (Lu et al., 2020). For this purpose, except for the gamification elements, the other factors would be constant. Developing a software product has shaped the model; however, forming groups as a gamification tool is demonstrated in the second model. The first model comprises some steps without employing the gamification component. After establishing groups in the company, it is feasible to use points and leaderboards, which are the most significant gamification method elements.

A conventional way of developing a software product comprises six steps: Analysis, Design, Implementation, Testing, Deployment, Maintenance (Hassan, 2016).

Analysis: In this step, the customers' information and the most feasible features applicable for generating new products will be analyzed. Furthermore, the requirements and goals of the projects are needed to be determined.

Design: This phase provides the situation to make the requirements visible by employing related software.

Implementation: In this step, the development team onset the coding process according to a roadmap, which is the most time-consuming period throughout the project.

Testing: After completing the task, it is required to determine all the defects; hence, quality assurance (QA) evaluates all the areas to specify the bugs.

Deployment: This phase refers to making the product available for customers to purchase or utilize.

Maintenance: it is noticeable that after delivering the product to the customers, it is vital to develop new features or fix customer bugs.

For developing both simulation models, all the phases from analysis to deployment have been considered to examine how much time is required to perform the same number of tasks for the models. This paves the way to compare the timeline of tasks being done in two discrete simulation models. Besides that, the number of bugs for each model would specify the product's quality improvement. The input and output for the models have been defined in table 3. The numbers 0,1 refer to utilizing that element or not.

Table III. Simulation input and output variables

Variable	*Role*
Team skill	Input
Project manager support	Input
Method	Input
Customer collaboration level	Input
Gamification	Input
Duration of simulation	Output
Cycle time for an activity	Output
Total bugs for specific activities	Output

Due to the fact that the structure of the organization and the work process is dissimilar in the two models (non-gamified and gamified model), the workflow and the steps of carrying out the tasks are described separately in simulation models for both cases to clarify the difference between the approaches as mentioned above concerning the procedures and resources since for implementing Gamified simulation models the resources are the teams; However, the non-gamified model resources consists of every employee in the organization.

B.A. Non-Gamified model

This model has considered one project management, analyzer, designer, backend programmers, front-end programmers, quality controllers, and human resources to assess their performance. The non-gamified duration functions are represented in table 4. The utilized parameters are extracted from Jira Software reports which the organization employed.

Table IV. Non-Gamified simulation parameters

Task Name	*Time (hours)*
Upload Stories in Jira Software	0.15
Time estimation	0.2
Story Analysis	1
Story design	2
Frontend coding	N*(12,1)
Backend coding	N(10,1)
Quality control	2
Debug backend	N(6,1)
Debug frontend	N(10,3)
Deployment	2

***Note: N=Normal distrubution**

The overview of the simulation model has been demonstrated in Fig 3. The tasks for each resource have been presented, and the resources aligned to that specific task are named in a circular shape. Each task's time would alter; therefore, the result would be diverse.

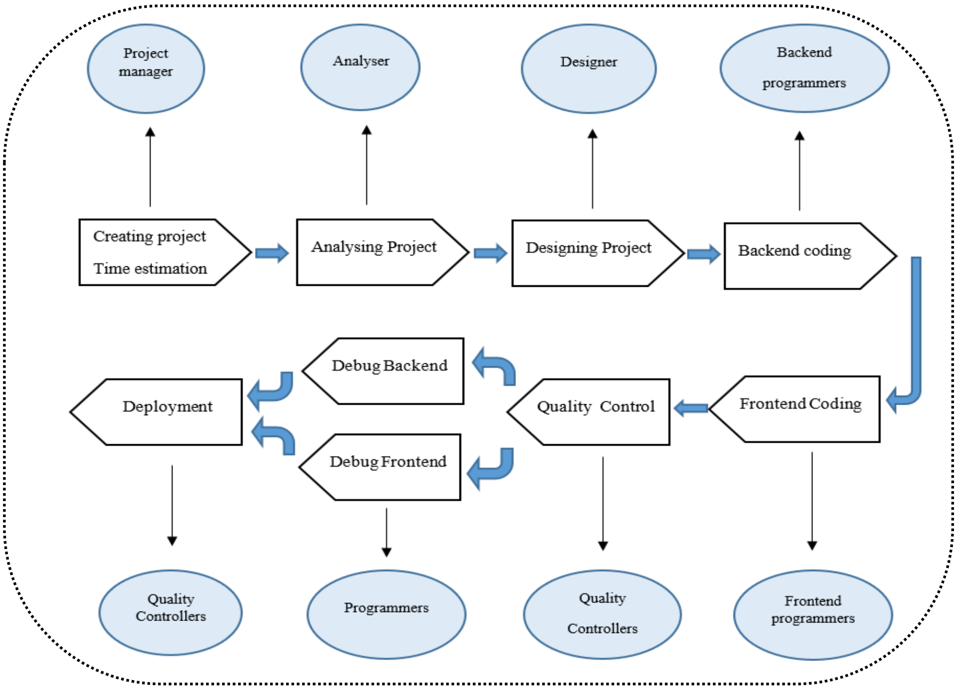

Fig 3. Non-Gamified Model

B.B. Gamified model

Every human Segment is of importance in the project's success. However, in this case, the programmers spend a great deal of time performing the project, and in most cases, the bottleneck in the whole process is coding. Therefore, the developers' department was the premier section for employing the gamification method to analyze the results. For utilizing point and leaderboard elements, 12 developers have been grouped into three discrete teams called A, B, and C. Each team has four members, consisting of two back-end programmers and two front-end programmers. The points the teams have gained, and the leaderboard for ranking them would be displayed to them at the meeting to motivate them to improve their performance. The non-gamified duration functions are demonstrated in table 5.

Table V. Gamified simulation parameters

Task Name	*Time (hours)*
Upload Stories in Jira Software	0.15
Time estimation	0.2
Story Analysis	1
Story design	2
Group A backend coding	N*(5,1)
Group B backend coding	N(6,1)
Group C backend coding	N(8,1)
Group A frontend coding	N(5,1)
Group B frontend coding	N(6,1)
Group C frontend coding	N(7,0.5)
Quality control	2
Debug Group A	N(2,0.5)
Debug Group B	N(3,1)
Debug Group C	N(5,1)
Deployment	2

***Note: N=Normal distrubution**

The gamified model process and their aligned resources are demonstrated in Fig 4.

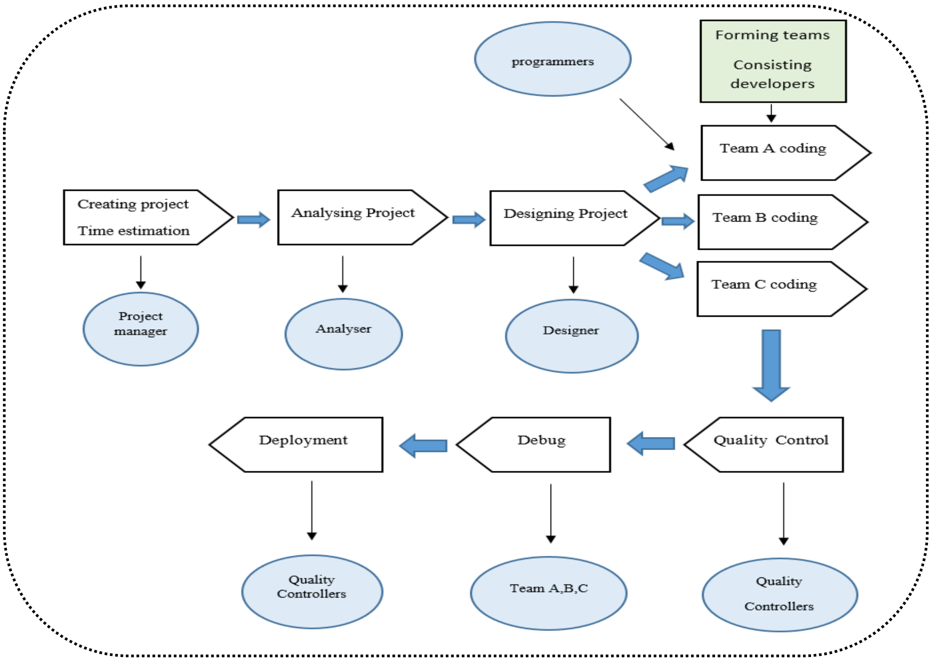

Fig 4. Gamified model

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, the predictive model will be built by logistic regression and will be evaluated by different metrics, and the most influential independent variable will be determined. Then the simulation model will be made to see the impact of the independent variable on the project improvement.

A. Building the predictive model and feature importance

In this section, the logistic regression classifier will be utilized to build a predictive model based on independent variables: team skills, customer collaboration, the project's manager support, using the agile or traditional method, and Gamification predict the software project will be successful or fail.

Before starting to train the model, the data preprocessing should be done to build a better model with the appropriate performance for dependent variable prediction; therefore, the one-hot encoding method is employed to convert the categorical features to numerical form.

The dataset is small; consequently, the K-fold cross-validation (CV) method is applied to the dataset to avoid overfitting (i.e., the model has trained appropriately but can't predict well). In this method, the training set is divided into K smaller parts, and then the model is trained using K-1 of the parts and, the remaining part is used for validating (i.e., this part is used for calculating the accuracy) and each time, the validation part and training folds changes and the test part is used for final evaluation; consequently, the average of accuracy is calculated (see Fig 5).

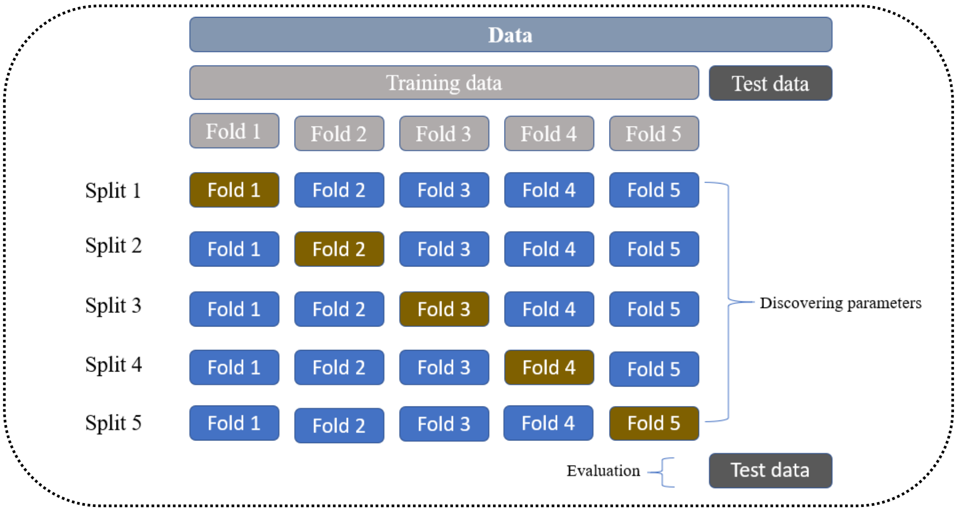

Fig 5. K-Fold Cross-Validation Structure

The Logistic regression algorithm makes the predictive model, and for the model assessment, Precision, Recall, F1-score, Accuracy, and Confusion Matrix are used. Fig 6 illustrates the Confusion matrix result.

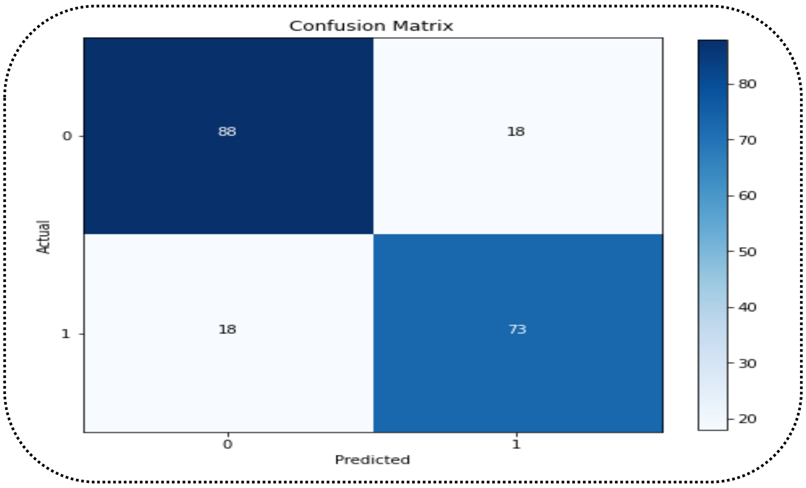

Fig 6. Confusion Matrix Result

The Precision, Recall, F1-score, and Accuracy metrics are calculated based on the Confusion Matrix in equations (6) – (9).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$F1 - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{9}$$

Where TP, TN, FP, and FN are defined as follows:

- True Positive (TP): The predictive model predicted positive, and the main value is positive.
- True Negative (TN): The predictive model predicted negative, and the main value is negative.
- False Positive (FP): The predictive model predicted positive, but the main value is negative (Type 1 error).
- False Negative (FN): The predictive model predicted negative, but the main value is positive (Type 2 error).

Before the logistics regression was chosen as a classifier, three algorithms including the decision tree classifier, random forest classifier, and multi-layer perceptron (MLP) classifier were tested, and the results were compared, then the logistic regression was chosen. The evaluation's outcome and algorithms comparison are presented in Table 6.

Table VI. Evaluation's Outcome And Algorithms Comparison

	Metric			
Algorithm	*Precision (%)*	*Recall (%)*	*F1-score (%)*	*Accuracy (%)*
Logistic regression	80.2	80.2	80.2	81.7
Decision Tree	74.4	70.3	75.1	72.3
Random Forest	74.1	72.5	75.6	73.3
MLP	79.3	75.8	79.6	77.5

Precision, Recall, and F1-score are calculated for each class by Classification-report, and results are presented in Table 7.

Table VII. Classification Report

	Metrics		
Class	*Precision (%)*	*Recall (%)*	*F1-score (%)*
0	83	83	83
1	80	80	80

The most effective independent variable on the project success will be determined based on the coefficients of the predictive model's independent variables. To find the most effective independent variable, the Euler number to the power of each independent variable's coefficient is calculated. Then the results are compared (see Fig 7), and the bigger number shows that the independent variable has more effect on project success which can be proved as follows:

$$\frac{P\,(y=1)}{P\,(y=0)} = \frac{P\,(y=1)}{1 - P\,(y=1)} = \frac{\frac{1}{1+e^{-z}}}{1 - \frac{1}{1+e^{-z}}} = e^{z} = e^{\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_n X_n} \tag{10}$$

Now, when making a change on any independent variable such as increasing X_1 by 1 unit, the prediction will change as follows:

$$\frac{y_{new}}{y} = \frac{e^{\theta_0 + \theta_1(X_1+1) + \theta_2 X_2 + \ldots + \theta_n X_n}}{e^{\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_n X_n}} =$$
$$e^{\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_n X_n - \theta_0 - \theta_1 X_1 - \theta_2 X_2 - \ldots - \theta_n X_n} \tag{11}$$

So, if the independent variable X_1 is increased 1 unit; therefore, the prediction will change e^{θ_1} which is the euler number to the power of the independent variable's coefficient.

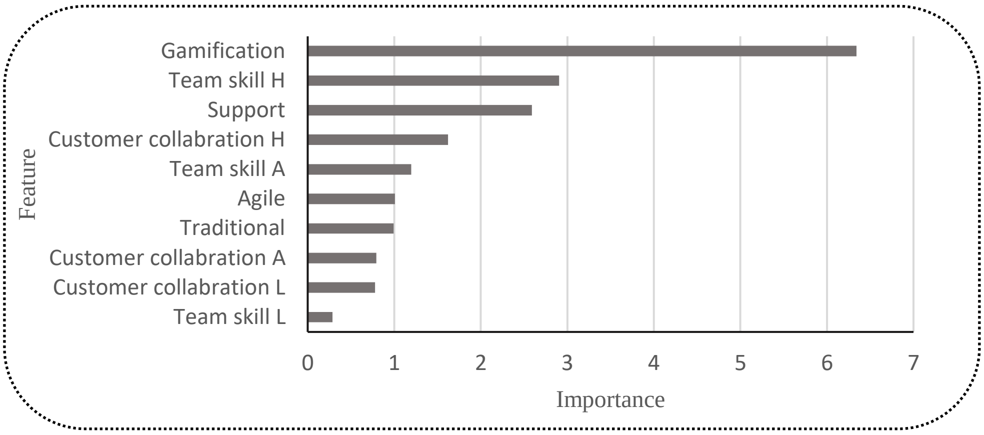

Fig 7. Feature Importance

Based on Fig 7, Gamification has the most impact on the project's success. The simulation model will determine the project improvement based on employing Gamification in the next section.

B. Simulation results

The simulation results are presented in this section. The results of twenty projects, picked out randomly from the dataset, have been displayed in table 8. The projects' data, called 1 to 20, is used as the simulation models' input factors. For each project, the non-gamified model and gamified model's cycles time have been written in the table; for more details, the minimum value and maximum value in addition to the mean value have been reported. To highlight the gamification tool as a contributing factor, it must specify the simulated model's improvement.

Table VIII. Evaluation Of Gamification Effectiveness On Project Improvement Based On Cycle Time

Project code	*CCL**	*Method*	*PMS**	*TS**	*Non-Gamified model time (hour)*			*Gamified model time (hour)*			*Improvement*
					min	*mean*	*max*	*min*	*mean*	*max*	
1	A*	agile	0	H	31.67	36.06	42.48	18.08	24.7	30.54	31%
2	H*	agile	1	H	29.5	33.59	39.84	16.89	22.25	28.64	33%
3	L*	traditional	0	H	35.16	39.73	47	22.53	28.38	36.43	28%
4	A	traditional	1	A	35.67	40.76	46.54	19.32	25.58	32.82	37%
5	H	agile	1	H	28.98	33.26	38.84	16.88	22.89	28.64	31%
6	H	agile	0	H	37.29	43.63	50.04	18.79	24.57	32.19	43%
7	A	agile	1	H	36.02	39.74	47.55	16.86	22.45	31.06	43%
8	A	traditional	0	H	38.25	43.26	50.65	18.3	25.63	32.71	40%
9	L	agile	0	H	35.72	40.4	48.68	17.28	25.11	29.85	37%
10	L	agile	1	H	33.42	38.43	44.76	17.51	22.92	28.8	39%
11	A	agile	0	L	35.46	40.35	45.94	18.97	24.59	29.78	39%
12	L	traditional	1	H	36.84	41.17	47.33	19.74	26.56	32.66	34%
13	A	agile	1	H	35.65	40.19	46.85	17.24	22.58	30.67	42%
14	A	agile	0	L	34.58	39.78	45.6	17.48	25.41	32.28	35%
15	L	agile	1	L	34.15	38.55	44.57	18.5	23.22	30.72	40%
16	L	traditional	0	L	38.75	43.84	50.63	21.8	28.51	35.27	34%
17	L	traditional	0	A	31.55	38.53	44.91	22.36	28.34	35.16	27%
18	L	traditional	1	A	37.13	41.59	49.26	21.63	26.77	34.3	36%
19	H	traditional	0	A	35	39.1	44.44	21.46	27.78	35.04	28%
20	A	agile	1	H	30.06	34.05	40.41	15.01	22.72	29.7	33%
		Average			34.54	39.3	45.81	18.83	25.04	31.86	36.26%

***Note: CCL= Customer collaboration level, PMS= Project manager support, TS= Team skill, A=Average, H=High, L=Low**

After reporting the time in hours, it is applicable to demonstrate the time cycle improvement percentage by comparing two models. To do this, the mean value is utilized, and ultimately, the results signify that the total time of the projects has been reduced noticeably from 786 to 500 hours, which presents a 36.2% improvement in this factor. For incrementing the accuracy of the results, 100 observation count from the Simphony software have been considered, and according to results gamified model can perfectly adjoin the organization to be utilized.

The quality assurance section is significant in the organization since it directly concerns the customers and other employees, and every bug detected by customers would be reported to this section. Furthermore, it is important to reduce the number of bugs in the project life cycle since fixing them requires considerable time and effort. The number of bugs in two discrete simulation models for each project has been demonstrated in Fig 8.

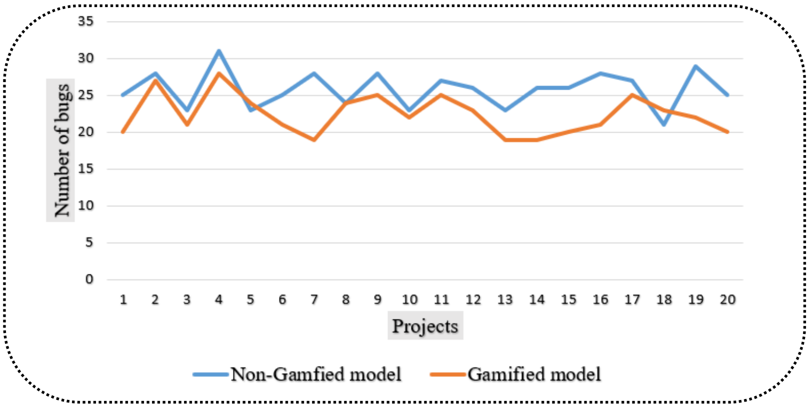

Fig 8. Number Of The Bug For The Simulation Model

By comparing two trends in Fig 8, it is recognizable that except for a small number of the projects, the number of bugs has declined in most of them out of 516 total bugs in the Non-gamified model to 448 in the gamified model, which showed a 15% enhancement in the quality of employers' performance.

C. Validate the simulation result by implementing in the case study

After implementing the simulation method and displaying the importance of gamification development in the organization, the execution of this method in the objective case study is highly significant to validate the results obtained through simulation. This software company's tasks and projects comprise making it possible for customers to leverage systems and devices to keep things running smoothly in their organizations and support customers whenever they face fault or dysfunctionality of software. However, the projects for this research are among those which is related to developing new software. Four projects were considered for validating this study results, which were executed in the organization, and their success was confirmed. To consider similar conditions, four projects were considered that were very similar to dataset projects. This time, the gamification method was employed to perform the projects to examine the results in the actual case. The inputs considered in this case are similar to the simulation model, and only the gamification factor was added to the organization. The results are shown in Table 9.

Table IX. The Results Of Implementing Gamification In The Case Study

	The gamified model Cycle time		*The gamified model number of bug*	
Project code	*Simulation*	*Case study*	*Simulation*	*Case study*
1	31%	29%	20	22
4	37%	25%	27	28
9	37%	30%	25	27
17	27%	19%	21	26
Average	33%	25.75%	23.25	25.75

The results in the real-practice resulted that the gamification method could perfectly fit the organization and reduce the cycle time in these specific projects. Even though the duration of projects has a 25.75% decrease on averagely, the number of bugs was not close to the prediction in one of the projects, and even it was higher than the number of bugs in the traditional method. By performing research revolving around that project, it was concluded that in some cases, the employees prefer the gamification point as compared to the quality; therefore, it is highly suggested that in this method, the top management should control the quality controller not to pass the projects to the customers when all the phases have not been analyzed perfectly. Ultimately, the other four projects showed a reduction in the number of bugs than the traditional method as predicted.

According to the real-practice case study, the gamified model attempted to convert the organization from static to dynamic and increment cooperation among employees and feedback from them to improve teamwork, which is applicable when employees have the same team goal. Specific rewards lead to motivating all team members more, and it is witnessed that every employee attempts to reach a high rank in the leaderboard, ideally causing to boost employees' work quality. In the organization ahead, many employees participated in the teamwork and organizational meetings motivationally in order to receive intrinsic and extrinsic rewards. They attempted considerably in the direction of team improvement, and also, the failure of the project due to practical team cooperation was significantly reduced.

From a practical viewpoint, taking advantage of the gamification method in the organization provides the conditions for employees to perform their duties to succeed. In this regard, team building, which is one of the Gamification elements, can be used in different parts of the organization, due to which cooperation among members would increase. To create a better team rank for team members, employees harness their ability and convey the knowledge to their co-workers, increasing all employees' ability. Factors such as increasing motivation and learning in the organization are project managers' goals since these factors improve the organization's performance. As a result, project time and cost would reduce, and the employee's work quality would increase.

D. Sensitivity Analysis

Concerning the results in Fig 7, except for the gamification factor, the next substantial factor which needs to be considered is team skill . To assess the effectiveness of team skill on project management factors (Time, Quality), a specific project (PMS=1, Method=Agile, CCL=Average, Gamification=0, Team skill=Low) is considered. Transformation in the situation is such that all factors except team skill are fixed. The amount of improvement in time and quality after changing Personnel skills from low to high in the simulation model is presented in Fig 9.

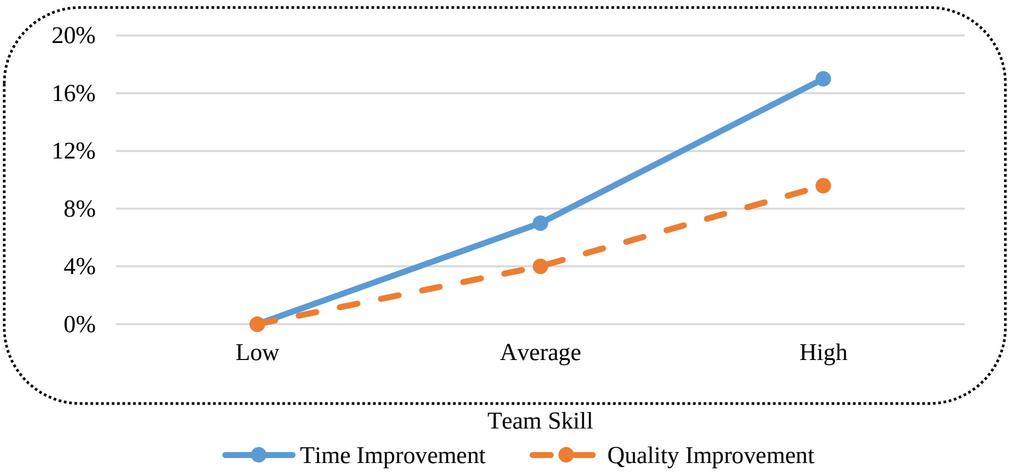

Fig 9. Team Skill Sensitivity Analysis

The results show that skilled personnel can positively affect the quality of work and reduce project time. However, It is not as influential as gamification elements.

V. MANAGERIAL INSIGHTS

The perspective of project managers is to accomplish successful projects and enhance the organization's performance. All organizational activities are in the direction with the predetermined goals of the organization. Exploiting the gamification elements paves the way for project managers to significantly increment the organization's financial gain, which is why numerous organizations utilize this approach. The phrase team assembling makes it possible for project managers to cultivate a habit of independent learning in the organization, which is one of the critical insights of project managers since no organization can make progress without ameliorating the knowledge of its employees. This teamwork to gain more points strengthens the interpersonal and communication skills within the organization, resulting in identifying obstacles in projects in the early stages. Project managers highlight potential collaboration across the groups due to the fact that not having much feedback within the organization leads to the discovery of fundamental flaws in the project after the completion of the project.

The proposed framework can be implemented forthwith, and financial investment for setting on Gamification is tiny commensurate to the benefits that can be gained. The feasibility of execution is interdependent with the organization's environment and resources. Managers should consider Gamification as a tool for encouraging employees according to firm goals since increasing pressure on the workforce would cause opposite results; Employees' concentration would be on gaining points instead of performing proper tasks. For successful implementation, staff knowledge regarding this approach should be increased, and team bonding should be employed as an influential factor in preparing each individual to enter the gamification method.

VI. CONCLUSION

The present paper was mainly arranged to present a predictive model which predicts whether the project will be successful or fail based on the independent variables and find the most important independent variable. A machine learning-based model was built by the logistic regression algorithm, which this algorithm was chosen among other machine learning algorithms such as the decision tree classifier, random forest classifier, and multi-layer perceptron (MLP) classifier by comparison based on their performance and accuracy. Also, the gamification factor was determined by the feature importance method based on independent variables coefficients in the predictive model as the most critical factor that has the most impact on the project success. In the next step, by comparing the gamified and non-gamified models with the simulation method, results showed Gamification made 36.26% improvement in the projects cycle time and 15% enhancement in the quality of employers' performance by decreasing the projects bugs. By considering this point, employing all of the factors is not feasible. It may impose expenses on the organizations, which assists decision-makers in focusing on utilizing Gamification to have better performance in the projects. According to the model results, it is notable that the improvement in the performance of the organization's teams is directly affected by game-making expressions, the possibility of creating more cooperation between employees, and the transfer of valuable information between them in this situation increases. Forming teams in the organization to receive more rewards and team points is highly valuable according to much more feedback from co-workers and troubleshooting the team to improve team rank. In order to validate the simulation results, the gamification method has been employed in the actual case, and the results reported that these tools could ideally fit the organization in order to reduce the project time cycle and boost the project quality; nevertheless, in a tiny proportion of the projects these tools can negatively affect the number of bugs. By analyzing this specific project type, it is concluded that the points are a significant factor that most of the employees consider; hence, the project manager should control the employees not to transfer the products, which does not have all the required, to the customers.

In this paper, some limitations affected the whole process. First, the captured data was from a software organization, and the proportion of data was unstructured. Hence, the number of features could be analyzed was limited; the dataset was formed according to project management reporting. Second, implementing the real projects within the organization required a great deal of budget and time which imposed researchers to execute a tiny number of projects in the actual workplace. For future research, it is essential to calculate the organization's costs in discrete simulation models and express the output as the organization's revenue in both gamification elements and its absence. From an economic point of view, input factors will be added to the model, including the cost of rent and facilities and human resources, so that the organization's total cost can be calculated. To compare the model, considering the cost of implementing gamification tools in the organization should be analyzed and used as input to the gamified model.

REFERENCES

Akgün, A.E. (2020) Team wisdom in software development projects and its impact on project performance. International Journal of Information Management 50: 228-243.

Amani, M.A., Ebrahimi, F., Dabbagh, A. et al. (2021) A machine learning-based model for the estimation of the temperature-dependent moduli of graphene oxide reinforced nanocomposites and its application in a thermally affected buckling analysis. Engineering with Computers 37(3): 2245-2255.

Amid, A., Biancolin, D., Gonzalez, A. et al. (2020) Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs. IEEE Micro 40(4): 10-21.

Amirkhalili,Y.S., Aghsami, A., Jolai, F. (2020). "Comparison of Time Series ARIMA Model and Support Vector Regression". International Journal of Hybrid Information Technology, 13(1), pp.7-18. doi:10.21742/IJHIT.2020.13.1.02

Bassil, Y. (2012) A simulation model for the waterfall software development life cycle. arXiv preprint arXiv:1205.6904.

Bibi, S. and Stamelos, I. (2006) Selecting the appropriate machine learning techniques for the prediction of software development costs. IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, 533-540.

Cerdeiral, C.T. and Santos, G. (2019) Software project management in high maturity: A systematic literature mapping. Journal of Systems and Software 148: 56-87.

Dashti, M.S., RezaZadeh, M., Khanzadi, M. et al. (2021) Integrated BIM-based simulation for automated time-space conflict management in construction projects. Automation in Construction 132: 103957.

Davoudabadi, R., Mousavi, S.M., Šaparauskas, J. et al. (2019) Solving construction project selection problem by a new uncertain weighting and ranking based on compromise solution with linear assignment approach. Journal of Civil Engineering and Management 25(3): 241-251.

Deterding, S., Dixon, D., Khaled, R. et al. (2011) From game design elements to gamefulness: defining" gamification". Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments. 9-15.

Dugarte-Peña, G.L., Sánchez-Segura, M.I., Medina-Domínguez, F. et al. (2020) Simulation of the software development process: an approximation using System Dynamics and the Larman Method.

Efe, P. and Demirors, O. (2019) A change management model and its application in software development projects. Computer Standards & Interfaces 66: 103353.

Featherstone, M. and Habgood, J. (2019) UniCraft: Exploring the impact of asynchronous multiplayer game elements in Gamification. International Journal of Human-Computer Studies 127: 150-168.

Fisher, J., Koning, D. and Ludwigsen, A. (2013) Utilizing Atlassian JIRA for large-scale software development management. Reportno. Report Number|, Date. Place Published|: Institution|.

França, B.B.N. and Travassos, G.H. (2013) Are we prepared for simulation based studies in software engineering yet? CLEI electronic journal 16(1): 9-9.

García-García, J., Enríquez, J.G., Ruiz, M. et al. (2020) Software process simulation modeling: systematic literature review. Computer Standards & Interfaces 70: 103425.

Gitinavard, H. (2019) Strategic Evaluation of Sustainable Projects based on Hybrid Group Decision Analysis with Incomplete Information. Journal of Quality Engineering and Production Optimization 4(2): 17-30.

Gitinavard, H. and Mousavi, S. (2015) Evaluating construction projects by a new group decision-making model based on intuitionistic fuzzy logic concepts. International Journal of Engineering 28(9): 1312-1319.

Gitinavard, H., Mousavi, S.M., Vahdani, B. et al. (2020) Project safety evaluation by a new soft computing approach-based last aggregation hesitant fuzzy complex proportional assessment in construction industry. Scientia Iranica 27(2): 983-1000.

Haleem, M., Farooqui, M.F. and Faisal, M. (2021) Cognitive impact validation of requirement uncertainty in software project development. International Journal of Cognitive Computing in Engineering 2: 1-11.

Hamzeh, A.M., Mousavi, S.M. and Gitinavard, H. (2020) Imprecise earned duration model for time evaluation of construction projects with risk considerations. Automation in Construction 111: 102993.

Han, W.J., Jiang, L.X., Lu, T.B. et al. (2015) Comparison of machine learning algorithms for software project time prediction. International Journal of Multimedia and Ubiquitous Engineering 10(9): 1-8.

Hassan MS (2016) Software Development Life Cycle.

Hsieh, M-Y., Hsu, Y-C. and Lin, C-T. (2018) Risk assessment in new software development projects at the front end: a fuzzy logic approach. Journal of Ambient Intelligence and Humanized Computing 9(2): 295-305.

Koivisto, J. and Hamari, J. (2019) The rise of motivational information systems: A review of gamification research. International Journal of Information Management 45: 191-210.

Kolkman, D. (2020) The usefulness of algorithmic models in policy making. Government Information Quarterly 37(3): 101488.

Lalsing, V., Kishnah, S. and Pudaruth, S. (2012) People factors in agile software development and project management. International Journal of Software Engineering & Applications 3(1): 117.

Li, D., Deng, L., Zeng, X. et al. (2021) Dynamic simulation modelling of software requirements change management system. Microprocessors and Microsystems 83: 104009.

Liu, D., Wang, Q. and Xiao, J. (2009) The role of software process simulation modeling in software risk management: A systematic review. 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, 302-311.

Liu, X., Li, D-Q., Cao, Z-J., et al. (2020) Adaptive Monte Carlo simulation method for system reliability analysis of slope stability based on limit equilibrium methods. Engineering Geology 264: 105384.

Lu, F., Chen, B., Qi, L. et al. (2020) PadMesh: a parallel and distributed framework for interactive mesh generation software. Engineering with Computers. 1-22.

Machuca-Villegas, L. and Gasca-Hurtado, G.P. (2018a) Gamification for improving software project management processes: A systematic literature review. International Conference on Software Process Improvement. Springer, 41-54.

Machuca-Villegas, L. and Gasca-Hurtado, G.P. (2018b) Gamification for improving software project: systematic mapping in project management. 2018 13th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 1-6.

Machuca-Villegas, L. and Gasca-Hurtado, G.P. (2019) Toward a Model based on Gamification to Influence the Productivity of Software Development Teams. 2019 14th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 1-6.

Mahdi, M.N., MH, M.Z., Yusof, A. et al. (2020) Design and Development of Machine Learning Technique for Software Project Risk Assessment-A Review. 2020 8th International Conference on Information Technology and Multimedia (ICIMU). IEEE, 354-362.

Masoud, M., Abu-Elhaija, W., Jaradat, Y. et al. (2017) Software project management: resources prediction and estimation utilizing unsupervised machine learning algorithm. International Conference on Engineering, Project, and Product Management. Springer, 151-159.

Meder, M., Plumbaum, T. and Hopfgartner, F. (2013) Perceived and actual role of gamification principles. 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. IEEE, 488-493.

Moldon, L., Strohmaier, M. and Wachs, J. (2020) How Gamification affects software developers: Cautionary evidence from a quasi-experiment on GitHub. arXiv e-prints. arXiv: 2006.02371.

Muszyńska, K. (2020) Gamification of communication and documentation processes in project teams. Procedia Computer Science 176: 3645-3653.

Najjar, M., Ghannouchi, S.A. and Ghannouchi, S. (2021) GaBPM: An attempt for the enhancement of BP performance through Gamification. Procedia Computer Science 181: 835-842.

Niazi, M., Mahmood, S., Alshayeb, M. et al. (2016) Toward successful project management in global software development. International Journal of Project Management 34(8): 1553-1567.

Perryer, C., Celestine, N.A., Scott-Ladd, B. et al. (2016) Enhancing workplace motivation through Gamification: Transferrable lessons from pedagogy. The International Journal of Management Education 14(3): 327-335.

Pospieszny, P., Czarnacka-Chrobot, B. and Kobylinski, A. (2018) An effective approach for software project effort and duration estimation with machine learning algorithms. Journal of Systems and Software 137: 184-196.

Rashid, E., Patnaik, S. and Bhattacherjee, V. (2012) Software quality estimation using machine learning: Case-Based reasoning technique. International Journal of Computer Applications 58(14).

Rodrigues, L.F., Oliveira, A. and Rodrigues, H. (2019) Main gamification concepts: a systematic mapping study. Heliyon 5(7): e01993.

Rus, I., Neu, H. and Münch, J. (2014) A systematic methodology for developing discrete event simulation models of software development processes. arXiv preprint arXiv:1403.3559.

Saremi, R.L. and Yang, Y. (2015) Dynamic simulation of software workers and task completion. 2015 IEEE/ACM 2nd International Workshop on CrowdSourcing in Software Engineering. IEEE, 17-23.

Serrador, P. and Pinto, J.K. (2015) Does Agile work?—A quantitative analysis of agile project success. International Journal of Project Management 33(5): 1040-1051.

Shariati, M., Mafipour, M., Ghahremani, B. et al. (2020) A novel hybrid extreme learning machine–grey wolf optimizer (ELM-GWO) model to predict compressive strength of concrete with partial replacements for cement. Engineering with Computers. 1-23.

Singh, S. (2012) Gamification: A strategic tool for organizational effectiveness. International Journal of Management 1(1): 108-113.

Spanellis, A., Dörfler, V. and MacBryde, J. (2020) Investigating the potential for using Gamification to empower knowledge workers. Expert Systems with Applications 160: 113694.

Sulaiman, MAH. and Mansor, Z. (2018) Critical success factors in agile enterprise architecture: A conceptual paper. Advanced science letters 24(7): 5220-5223.

ul Hassan, I., Ahmad, N. and Zuhaira, B. (2018) Calculating completeness of software project scope definition. Information and Software Technology 94: 208-233.

Werbach, K. and Hunter, D. (2014) Gamificación: revoluciona tu negocio con las técnicas de los juegos. Pearson Educación.

Xi, N. and Hamari, J. (2019) Does Gamification satisfy needs? A study on the relationship between gamification features and intrinsic need satisfaction. International Journal of Information Management 46: 210-221.

Zhang, H., Kitchenham, B. and Pfahl, D. (2008) Reflections on 10 years of software process simulation modeling: A systematic review. International Conference on Software Process. Springer, 345-356.