## A software ecosystem for project management in BIM environments assisted by artificial itechniques.

Pedro E. Piñero Ramírez2[0000-0002-4498-0755], Iliana Pérez Pupo1[0000-0003-1433-0601], Pedro Y. Piñero Pérez1[0000-0002-7635-8290], Yosvani Marquez Ruiz2, Yulia Fustiel Alvarez2, Roberto García Vacacela3[0000-0002-1834-6806]

1 Intelligent Software Ecosystems Group, BusinessRedmine corporation (Canada) pepineror@gmail.com

2 Grupo de Investigaciones en Gestión de Proyectos. Universidad de las Ciencias Informáticas (La Habana, Cuba, CP 54830) iperez@uci.cu, [ppp@uci.cu](mailto:ppp@uci.cu), pppyob@gmail.com

3 Facultad de empresariales. Universidad Católica Santiago de Guayaquil (Ecuador)  
[roberto.garcia@cu.ucsg.edu.ec](mailto:roberto.garcia@cu.ucsg.edu.ec), [nicolas.villavicencio@cu.ucsg.edu.ec](mailto:nicolas.villavicencio@cu.ucsg.edu.ec)

**Abstract.** This paper presents a new platform called BusinessRedmine for project management in BIM environments. The platform constitutes a software ecosystem that combines traditional project management techniques with soft computing techniques for making decisions in projects. The proposed platform is business intelligence tools and consist of a set of software actives that cover the different processes of project management. The platform allows the management of multiple processes such as: scope management, planning management, schedule construction, risk management, stakeholder management, communications and quality. In addition, it facilitates integration with tools for 2D and 3D design using the IFC format. The platform includes facilities for managing project files aligned with the ISO 19650 series of standards. One of the fundamental elements it provides is the application of soft computing techniques to artificial intelligence for decision-making in project cuts. We compare the algorithm FISBR proposed in BusinessRedmine systems for project evaluation with other algorithms reported in bibliography based on neural networks and genetic algorithms techniques. The algorithms selected learn from a database that contains projects already evaluated and after that classify other projects. In experiments, we applied cross validation techniques combined with Friedman test and Wilcoxon test. Finally, a comparison with different project management tools is presented.

**Keywords:** Soft computing. Project Management, Artificial intelligence, BIM

### Introduction

There is a growing demand for tools with high added value for the management of project-oriented organizations. Within this context there is a group of elements that must be carefully planned to achieve the project objectives with an adequate balance of cost, time and quality. Two spheres of knowledge that exemplify this situation are the growing development of Building Information Modeling (BIM) technologies [1] [2] [3] for construction projects and the development of software projects [4].

In particular, in the construction sector, BIM technologies are aimed to simulate the complete execution of projects in virtual environments. The use of BIM methodologies and tools is still not very standardized and not generalized at a global level [5]. They constitute an open field for the development of new research [6] [7] [8]. At the current moment of BIM development, progress has been made fundamentally in the integration and parameterization of the designs or plans of the works with the planning activities. In this context, it is proposed that around 34% of the resources invested in a project are wasted and that approximately 95% of the information that is generated later is underused and is not used as lessons learned in new projects [9]. In these same sources, it is stated that the main factor that affects the inappropriate use of resources is found in inefficient planning. Considering this context, the development of new algorithms and techniques that allow the optimal or quasi-optimal construction of project schedules and take into account the limitations of resources and the variability in the ways of executing the tasks, remains a necessity.

Similarly, it happens in the development of software projects. In this area of ​​knowledge, it is identified by several sources that, in medium-sized projects, approximately 26% need to be renegotiated and 31% are canceled [10] [11] [12]. This scenario is characterized by high dependence on human resources and their skills, accompanied by tasks that can be performed in various ways. Among the fundamental causes of project failure are: insufficient training of human resources and frequent errors in planning processes, whether in scope, time or logistics, and poor management and inadequacies in control and monitoring processes [13] [14] [15]. In addition, it is identified that there is a group of factors that influence this problem related to the inadequacies of the available computer tools [13].

In this sense, a study of the tools reported in the bibliography was carried out and around 125 solutions associated with project management were identified [13]. Authors analyze 125 solutions, they identified that only 25 are based on free software, which represents 20% of the total. A characterization of these tools regarding the coverage of project management processes is shown in Table I.

Tabla I: Analysis of project management tools based on open source.

|  |  |
| --- | --- |
| Features that include | Percent |
| Portfolio management | 44 % |
| Planning management | 60 % |
| Human and non-human resource management | 56 % |
| Document management | 60 % |
| Control and monitoring reports | 32 % |

From the detailed analysis of the tools reported in the bibliography, the following shortcomings could be identified:

* Need to use numerous tools with different interfaces and low integration to manage the data they generate.
* Only 44% of the tools reported in the bibliography include facilities for the management of indicators for data analysis.
* Among the most frequent indicators in the tools are those associated with cost management described in the PMBOK [16], such as: earned value (EV), cost performance index Cost Performance Index (CPI), the Schedule Performance Index (SPI), Cost Variance (CV), Planned Cost (PV), and Actual Cost (AC).
* The few tools that allow the management of project portfolios are notable, in this case only 11 tools with this functionality.
* Although some tools include indicators for evaluating costs and time, they do not propose specific mechanisms for managing uncertainty in their calculation.
* In general, the existing tools do not allow treatment of the imprecision, vagueness or uncertainty contained in the information.
* High cost of solutions and low level of customization for different scenarios.
* Little availability of software ecosystems with solid architectures that take advantage of free software for reuse and rapid development of low-cost solutions.
* Among the open source or free solutions, only 8 sufficiently include the control and monitoring reports that they represent. In this way, a significant insufficiency is identified in the decision-making support processes in project management.

This paper presents a software ecosystem, based on free software technologies called BusinessRedmine for project management in different production environments. The second section presents a brief study of the state of the art associated with software ecosystems. The third section presents the architecture proposal of the proposed software ecosystem. Then in the fourth section is the analysis of results in the application of the ecosystem. Finally, the last section presents the conclusions of the work.

### Brief analysis of software ecosystems

Traditional software development models manage projects independently, an element that causes each project to be presented as a custom development and production costs rise. These traditional models include cascade development, prototype-based development, original component-based model, and spiral development. As an alternative to these traditional models, others such as software product lines (LPS) and software ecosystems have emerged. These new models with a focus on architecture and reuse enhance cost reduction and increase quality in productions.

The software product lines constitute industrial models of software development, inspired by models of chain production used in other spheres of society. They are created from the evolution of development models based on components and software factories, improving the organization of processes and the division of responsibilities between the departments of the organization. This development model encourages the management of project programs, over and above the management of independent projects.

The elements that define an LPS are: domain, product families, the set of reusable assets, the mode of production and the architecture, as a central element that guarantees the integration of solutions and reuse [17] [18].

Software ecosystems, for their part, are aimed at creating alliances between different organizations, focusing on outsourcing the relationships of organizations, enhancing productivity and sharing market sectors [19-23]. According to Bosh, a software ecosystem constitutes a set of computer systems that enable, automate and support a certain business need [20]. Other authors, for their part, consider that the basis of software ecosystems is in the establishment of solid business architectures that are supported by the integration of their components [24].

Among the important elements to guarantee the sustainability of ecosystems is the need to define the income models that regulate the relationships between the different entities of the ecosystem. Let's see below different entry strategies that can be used and the trends in this regard within free software [20]:

* Custom development: in this scheme, users pay for the development of customized software for their specific requirements. It is used by 1.8% of those surveyed.
* Services and support: generally supported by phone calls, training, and consulting contracts. Employed by 7.9% of those surveyed.
* Business licensing: a licensing under a business license in the traditional way. Used by 24.6% of those surveyed.
* Software as a service (SaaS): users pay to access and use the software via the internet. Used by 5.3% of those surveyed.
* Embedded Software: Open source is embedded in commercial end-to-end solutions packages. Used by 5.3% of those surveyed.
* Embedded hardware: Open source is embedded in hardware that is generally distributed under commercial licenses. Used by 9.7% of those surveyed.
* Advertising: the software is free to be used under advertising conditions. Employed by 2.6% of those surveyed.
* Subscriptions: where support services and updates are contracted, for a period of time that is generally annual. Employed by 34.2% of those surveyed.
* Other products and services: this model refers to the fact that open source is not used directly to generate income, but other products and services that are generally proprietary are those that generate the income and contain parts based on open source. This model is used by 8.8% of those surveyed.

It is also important that the licensing strategies for the ecosystem products are defined, promoting good relations between the entities and guaranteeing the legal protection of the products. In this sense, the following strategies are presented [25]:

* Open Source - The solution is released under an open source license only. Employed by approximately 24.6%.
* Assembled Open Source: The product or service includes code from multiple open source projects using multiple licenses. Scheme used by 16.7% of those surveyed.
* Closed: The product is based on open source, but it is not available under open source licensing. Scheme used by 5.2% of those surveyed.
* Open and closed licensing: open source projects are developed from the union of independent parties under commercial licensing, developed as closed source. Method used by 14.9% of those surveyed.
* Open-core licensing: the core of the solution is open under an open source license. But the professional versions of the products include open and closed source developments and are commercially licensed. Scheme used by 23.7% of those surveyed.
* Dual licensing: the base code is licensed to different users as open source or as commercial licensing. Method used by 14.9% of those surveyed.

Note that the open-core-based licensing model is one of the most widely used, taking advantage of the advantages of free software, but managing the professional versions of the products under commercial licenses, which include closed source.

Another important element to consider is the development strategies that are the basis for the licensing of the solutions. Furthermore, these strategies have a direct influence on the level of code reuse and the reduction of development costs [25]:

* Open source community development: the software is distributed using open source licenses and is publicly developed by a community of supplying individuals or organizations. Method used by 9.7% of those surveyed.
* Open source development as a single provider: the software is distributed using a free software license and all contributions and extensions are published but development and is dominated by a single provider organization. Employed by approximately 21.1% of respondents.
* Mixed open source development: the product or service is provided by a single provider and is based on the combination of projects developed in free software, publicly developed by communities or other organizations. Employed by approximately 19.3% of respondents.
* Completely closed development: all software components and functionalities are closed by the producing entity and are not based on free software.
* Hybrid development: refers to development models where the software is distributed using open source licenses, but some functionalities are developed behind closed doors, the following three categories stand out in this regard:
  + Hybrid community: most of the software is developed publicly by a community of individuals or providers, while some functionalities are developed behind closed doors by providers of the product or service. Scheme used by 7.9% of those surveyed.
  + Hybrid providers: development is dominated by employees of a single organization, and some functionalities are developed behind closed doors. Scheme used by 33.3% of those surveyed.
  + Mixed hybrid: the product or service is based on the combination of projects that use open source developments developed publicly by communities or multiple providers and some functionalities are developed behind closed doors. Scheme used by 8.8% of respondents.

Based on the analysis of the models and the different strategies considered, a software ecosystem is proposed for the development of solutions and its use is proposed to help decision-making in project management.

An example of software ecosystems is the Microsoft Office suite, made up of a set of IT solutions for office management. Which includes facilities for project management among others. This ecosystem is also made up of all the relational knowledge of the Microsoft company that enhances the integration of its products with different technologies and solutions. This ecosystem is based on a commercial licensing model.

Another example of an ecosystem is the development of the Eclipse package. This is a set of solutions specialized in the development of IDEs for programming computer solutions [26]. It is an ecosystem based on open source solutions.

### Architecture of the BusinessRedmine Software Ecosystem.

The proposal constitutes a software ecosystem, called BusinessRedmine [27], for the full-cycle management of the integrated project management process. In this research the authors define the software ecosystem as follows:

**Definition of software ecosystem**: a software ecosystem is formed from the system of relationships between different entities with the aim of sharing clients, market segments or increasing efficiency and effectiveness in the production processes they develop. It is supported by the definition of common architectures and platforms that facilitate the integration of solutions, the exchange of information, resources, artifacts and assets in general. These relationships must be supported by economic-financial relationships, work agreements or development models that promote the exchange of data and codes. Its main components are:

* Domain of application.
* Family of products that make up the portfolio of solutions.
* Ecosystem assets that comprise the set of all reusable elements of the organization, both code components and documentation.
* Architecture that enhances the integration of assets, specifying and development strategies.
* Set of the modes of production of the entities of the ecosystem.
* The set of relationships between the entities that comprise it (relational knowledge), the planned entry strategies and the licensing and marketing strategies of the solutions.

This definition constitutes the basis of the architectural proposal developed. Some elements that characterize the BusinessRedmine ecosystem are.

* Allows help in decision-making in project management by combining traditional methods with different artificial intelligence techniques, facilitating help in decision-making.
* The processes that the BusinessRedmine ecosystem models are aligned with the elements that describe international standards such as: PMBOK [16], SCRUM, the ISO 21500 standard [28] and the ISO 19650 series [2].
* The BusinessRedmine ecosystem is based on web technologies and designed for installation in private or public clouds, guaranteeing a high level of security.
* The architecture that supports the ecosystem is based on free software and docker technologies that facilitate the updating and maintenance of solutions

The ecosystem is made up of a set of assets organized in the following layers:

* Layer associated with the high availability of solutions.

1. It is formed from the integration of HAProxy and KAlive tools that enhance high availability and platform performance. These facilities may be integrated into nginx or APACHE servers.

* Security layer:

1. Includes the assembly of LDAP server with interface for management from the WEB or integration with existing LDAP servers in the organization.
2. Rupcatcha that allows integration with captcha management that enhances secure access to information for each of the project participants.
3. Logstash's security configured to capture the logs associated with ecosystem security.
4. Elastic search security aimed at integrating with log stash and allowing the structuring of security logs with the platform for managing them.
5. Grafana's security configured to manage alarms and display the platform's security logs. It allows the classification of logs associated with application errors, user access to the platform and user behavior within the platform.
6. Firewall to control access to ecosystem components.

* Project management layer:

1. Base platform for managing programs and projects (work objects). It allows the intelligent management of different stages of project development and their schedules. Facilitates integration with the first dimension of BIM.
2. Subsystem for the feasibility analysis of programs, projects and project portfolios. It combines traditional methods with soft computing techniques to better model uncertainty and ambiguity in feasibility analysis processes. It allows integration with the sixth dimension of BIM, associated with the analysis of sustainability of investments.
3. Subsystem for strategic management that allows the modeling of strategic objectives from the management of indicators of physical and financial execution of projects.
4. Subsystem for risk management at the organization level and at the project level with treatment of uncertainty. It facilitates the qualitative and quantitative analysis of risks using different artificial intelligence techniques.
5. Balanced scorecard subsystem to help business decision-making in project management. Includes integration with ERP systems. It takes the best practices proposed by Kaplan and Norton integrated with alert systems based on artificial intelligence techniques.
6. System for the management of control indicators and monitoring of support projects at different management levels. It includes facilities for augmented analytics and the reuse of different solutions reported in the bibliography. It includes a set of indicators aligned with the main project management standards and agile development methodologies. Furthermore, it incorporates the following graphs and indicators: Burn-up, Burn-Down, S Curve and CPI, SPI, EV, EAC indicators among others.
7. Subsystem for managing project participants and their competencies.
8. CRM subsystem for managing clients, suppliers, shareholders and contracts.
9. Subsystem for the construction of optimal or quasi-optimal project schedules. It integrates with the fourth dimension of BIM, associated with time.
10. System for the management of logistic stock to support decisions of productive chain. It enhances the integration with the fifth dimension BIM associated with the cost of investment projects.
11. Subsystem for billing and sales management with integration with point of sale that provides a friendly interface for the company's clients at all times of the negotiation process.
12. Subsystem for making financial decisions and integration with accounting management systems, enhancing the integration of the fundamental activity of the park with basic economic management.
13. Subsystem for pre-investment analysis and estimation of initial costs of projects integration with the first dimension of BIM

* Data sharing environment layer:

1. Database manager subsystem that allows the assembly and administration of the data servers associated with the ecosystem.
2. Shared Data Environment Subsystem for integration with project file repositories. It includes facilities for the storage and development of advanced searches. Based on technologies that facilitate the integration of outputs from different BIM tools. Includes advanced document management and project files.
3. Subsystem for integration with distributed version control systems. That facilitate project management from the development process itself and the successful application of agile development techniques.

* Collaboration and continuous training layer:

1. Subsystem for the assembly of conversational systems applicable to support management and with potential for its application in different sectors of society. This asset is expected to be developed from the reuse, adaptation and evolution of other solutions reported in the bibliography and from characterizing the principles and best practices of conversational systems.
2. Subsystem for the support and virtual training of the clients of the system. It will allow the setting up of consulting and training services for client entities with facilities for remote work.

* Integration layer with BIM tools and advanced data analysis:

1. Subsystem for integration with the BIM tools control and monitoring ecosystem. This facilitates the development of the second and third dimensions of BIM associated with the basic engineering and detailed engineering of work objects.
2. Subsystem for the integration of the detailed schedules of the dashboard platform with BIM tools that relate the design models with the work execution schedules. And they allow the integration of BIM dimensions in the ecosystem.

The fundamental process that models the proposed ecosystem is the control and monitoring of the projects. This process has a strong integration with Artificial Intelligence techniques, as shown below. In this process, the ecosystem provides a guide based on the following steps:

1. Calculation of indicators from data.
2. Product quality review.
3. Automatic evaluation of the project from the indicators.
4. Detection of the fundamental difficulties of the project and areas of knowledge with difficulties.
5. If there are no problems, finish the check and congratulate.
6. In case of difficulties, go down the cascade of indicators and analyze their causes.
7. Having identified the causes, proceed to decision-making and the final evaluation of the project.

In step 1, the ecosystem calculates the indicators proposed in Table I

**Table** 4.1 Indicators for decision making

|  |  |
| --- | --- |
| Analysis area | Indicators |
| Project Integration management | * Comprehensive evaluation of the project * IAP (Compliance with court agreements) |
| Client management and commitments | * IE (Execution Index) * IRE (Execution Performance Index) |
| Time management | * SPI (scheduling performance index) |
| Cost management | * CPI (cost performance index) * PV, AC, EV (earned value) |
| Procurement Management | * IRL (Logistics Performance Index) |
| Human Resource Management | * IRRH (Human Resources Performance Index) |
| Scope and quality management | * IREF (Efficcacy Performance Index) |
| Information consistency | * ICD (Data Quality Index) |

In step 3 the BusinessRedmine platform introduces a fuzzy inference system FISBR, as a classifier for the automatic evaluation of projects that is described in details in [13]. Algorithm FISBR constitutes a hybrid neuro-fuzzy system of the Sugeno type. With this algorithm, we intend to take advantage of the potentialities of neural networks and the interpretability of fuzzy inference systems. The proposed FISBR uses a node-oriented multilayer network architecture.

As part of the experimentation of this research, the results' analysis section presents the results of the comparison of the proposed algorithm with other classifiers reported in the bibliography.

The platform is being applied in several real scenarios. And the algorithms proposed for control and monitoring were validated by experimenting with their behavior in a database of completed projects. The design of the experiments and their results are listed below.

### Results analysis

As part of the validation of the impact of the proposed ecosystem, the following design of experiments was carried out:

* Experiment 1: Comparison of the BusinessRedmine ecosystem with other systems for managing projects and project-oriented organizations.
* Experiment 2: analysis of the implementation of the system in different scenarios.
* Experiment 3: comparison of the classifier used in the evaluation of projects, based on the fuzzy inference system, with other classifiers reported in the bibliography.

#### Experiment 1 Comparison of the proposal with other tools.

In this experiment, the proposed ecosystem is compared with other tools reported in the bibliography. It is compared to tools for project management or project-oriented organizations Table 2.

Table 2 Comparison of the proposal with other tools.

|  | BusinessRedmine | Clarizen | MS Project Server | Jira | Basecamp | Redmine |
| --- | --- | --- | --- | --- | --- | --- |
| Open source | Yes | No | No | No | No | Yes |
| Collaborative work | Yes | Yes | No1 | Yes | Yes | Yes |
| Task management | Yes | Yes | Yes | Yes | No | Yes |
| Resource management | Yes | Yes | Yes | No | No | No |
| Finance management | Yes | Yes | Yes | No | Yes | No |
| Stakeholders management | Yes | Yes | Yes | Yes | Yes | No |
| Contract management | Yes | No | No | No | No | No |
| Risk management | Yes | Yes | No | No | Yes | No |
| Feasibility analysis | Yes | No | No | No | No | No |
| Automatic notifications (email) | Yes | No | No | Yes | Yes | Yes |
| Project management indicators | Yes | Yes | Yes | Yes | Yes | No |
| Kanban method integration and SCRUM agile methods | Yes | No | No | Yes | Yes | Yes |
| Learning curve (Easy, Medium, Complex) | Easy | Mean | Easy | Hard | Easy | Easy |
| Ability for customization and Sovereign development | Yes | No | No | No | No | Yes |

1 Needs to integrate with other Microsoft platforms like Visual Studio and Microsoft teams to facilitate collaboration

The table identifies the versatility of the proposed platform and its capacity for process integration. Furthermore, the fact that it was developed using free software technologies allows stakeholders to have access to the four freedoms of free software. The proposed platform incorporates a greater number of functionalities than the rest of the platforms used in the comparison.

#### Experiment 2 analysis of the system implementation process in different scenarios.

The results of the system implementation process are presented in the following four scenarios:

* Implementation in the Network of Centers of the University of Informatics Sciences since 2010. In this scenario, it was installed under the commercial name GESPRO. The calculated availability of the overall assembled system is around 92%.
* Implementation in the CERVEM Company, for the management of business processes under a project approach since 2018.
* Implementation in Engineering and Project Management Network (RCGP) with an impact on the training of the Master in Project Management postgraduate program, since 2020.
* Implementation of the ETECSA company for the management of research and innovation projects in 2021

As part of the analysis of the results, a survey was carried out with 32 clients of the system to analyze the variable "time spent analyzing projects and generating status reports." The distribution of the professionals surveyed was as follows: 14 project managers, 5 project planners, 7 project management office inspectors and 6 project quality reviewers. Measurements were made of the time spent analyzing projects and preparing status reports before and after the implementation of the solutions proposed by the BusinessRedmine ecosystem.

Respondents selected one of the following responses:

* less than 30 minutes.
* between 1 and 2 hours to prepare a report.
* between 3 and 7 hours to prepare a report.
* about 24 hours to prepare a report.
* about 32 hours to prepare a report.
* more than 40 hours to prepare a report.

The results were satisfactory, achieving in some cases a reduction in time from 1 week to minutes. As part of the validation of the proposed model, the increase in productivity was calculated during the development of product customizations. Productivity is calculated from equation 1.

(1)

* Product scope is expressed in the number of use cases that were necessary to develop in the implementation process.
* Amount of Resources: refers to the amount of human resources that participated in the implementation project in a specific scenario.
* The time variable indicates the total time it took to set up the application in the specific scenario.

Table V represents a sample of the data associated with the development of the different customizations of the ecosystem products, ordered by the order of their occurrence over time.

Table V: Increase in productivity with the use of the ecosystem in customizations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Real sceneriesimplementations | Product scope | Human resources | Time (days) | Efficiency indicator |
| UCI | 64 | 10 | 45 | 1.16 |
| CERVEM | 16 | 3 | 10 | 1.23 |
| RCGP | 32 | 3 | 10 | 2.46 |
| Telecomunications company | 32 | 2 | 7 | 3.56 |

As can be seen in Table V and in Figure 5, there is an increase in productivity motivated by the use of software ecosystems and LPS that promote the reuse of all assets.

Figura 5: Increase in productivity by reuse of BusinessRedmine software assets in different scenarios.

#### Experiment 3 analysis of the behavior of the project evaluation subsystem

One of the key elements in the work of the platform is the introduction of artificial intelligence techniques to help decision-making by project managers, users of the platform. This element is specified in all the functionalities related to the visualization of indicators in each of the management processes that the ecosystem supports. One of these functionalities being the project evaluation process that is based on a fuzzy inference system that in this experimentation we will identify as FISBR. In this section, an experiment is carried out to compare FISBR with other classifiers that allow classification and can be applied in the automatic evaluation of projects.

For this experiment, a set of classifiers representing different AI techniques are chosen, including neural networks, the descending gradient or evolutionary algorithms, and compared with the proposal made based on fuzzy inference systems FISBR.

* FIR.DM algorithm suggested by [29]. This algorithm uses a method based on the descending gradient to construct and optimize a Sugeno Degree Zero fuzzy inference system.
* FS.HGD algorithm recommended by [30]. It is based on the combination of the heuristic search and the descending gradient.
* GFS.LT.RS algorithm proposed in [31] for the generation of fuzzy rules in the evaluation of projects. This technique implements evolutionary lateral adjustment of the membership functions in order to obtain linguistic models with higher levels of precision. It uses a rule representation model based on 2-tuples linguistic representation that allows side-scrolling of labels (slight left / right shifts of the original membership function).
* GFS.THRIFT algorithm submitted in [32], This algorithm generates fuzzy rules and then optimizes them using a basic genetic algorithm (GA).
* HyFIS algorithm recommended by [33] is a hybrid neuro-fuzzy model used to build and optimize fuzzy systems. It also combines the prediction capabilities of neural networks with fuzzy inference systems, generating Mamdani-type rules.
* Algorithm based on WM: recommended by Wang and Mendel [34] in our particular case, it generates a set of candidate fuzzy rules for the evaluation of projects that are refined in a second phase. The generated rules are of the Mandani type. In the exploitation stage, it uses a fuzzy Mandani-type inference system and is able to show the rules that influenced the final decision.

The three techniques used for the validation of the results are listed below:

* Pre-experiment technique without initial observation: analysis of the results of applying the model in a case study. The objective of this pre-experiment is to evaluate the potential of the platform with the end users. For this experiment, a group of end users of the application, experts in project control and monitoring, was selected. Then a survey was applied to them related to their experiences in the use of the system and the facilities for the control and monitoring of the projects. Finally, the word computing technique was applied to identify the experts' evaluation.
* Normality analysis technique: techniques were applied to analyze the normality of the data.
* Experimental technique with initial and final observation: Comparison of the different algorithms proposed in the control and monitoring module in an experimental database. The objective of this experiment is to find the most efficient algorithm for project evaluation. In our context, the most efficient algorithm is the one that makes the fewest errors in the evaluation of projects, taking as a reference the previous classification by a human expert.

**Analysis of the results of the pre-experiment technique without initial observation.**

To evaluate the proposed model from experts, a questionnaire was developed in which 53 specialists dedicated to the control and monitoring of projects were involved. Specialists include project managers, project management office specialists, and development center managers. The objective of the questionnaire was to know the existing criteria about the quality of the proposed model and the indicators. This questionnaire was based on four criteria:

* Is the proposal for project evaluation applicable in different scenarios?
* Do you consider the proposal understandable?
* Is the proposed model usable, comfortable to be applied?
* Do the proposed indicators cover the knowledge areas of project management?

To validate the reliability of the prepared questionnaire, it was subjected to validation by 14 experts and using the SPSS tool, the Cronbach's Alpha coefficient test was executed on the data collected. This type of test is applicable to questionnaires with Likert-type responses, where each item or question is answered according to numbers that represent degrees of agreement or disagreement (Very High, Very High, High, Medium, Low, Very Low, None).

Cronbach's alpha coefficient was equal to 0.873. According to the literature, it is considered that alpha values ​​greater than 0.7 are sufficient to guarantee the reliability of the scale. Given the positive result obtained in this test, it was considered valid to apply the aforementioned questionnaire.

Once the questionnaire is applied, computer techniques with words are applied to consolidate the criteria obtained in it. The following steps were followed:

* + 1. A basic set of linguistic terms (LBTL) is defined for the evaluation of the criteria. LBTL = {not at all, very low, low, medium, high, very high, perfect}.
    2. The specialists evaluate each criterion using some linguistic terms, see Table II.

Table II: Structure of the evaluation of the specialists for each criterion. Ci represents the criteria and represents the specialists.

Para la validación de la fiabilidad del cuestionario confeccionado se sometió el mismo a validación por 14 expertos y utilizando la herramienta SPSS se ejecutó la prueba del coeficiente Alfa de Cronbach sobre los datos recolectados. Este tipo de prueba es aplicable a cuestionarios con respuestas de tipo Likert, donde cada ítem o pregunta se responde de acuerdo a números que representan grados de acuerdo o desacuerdo (Altísimo, Muy alto, Alto, Medio, Bajo, Muy Bajo, Ninguno).

El coeficiente Alfa de Cronbach fue igual a 0.873. Según la literatura se considera que valores del alfa superiores a 0.7 son suficientes para garantizar la fiabilidad de la escala. Dado el resultado positivo obtenido en esta prueba se consideró válido aplicar el citado cuestionario.

Una vez aplicado el cuestionario se aplican técnicas de computación con palabras para consolidar los criterios obtenidos en el mismo. Se siguieron los siguientes pasos:

1. Se define un conjunto básico de términos lingüísticos (LBTL) para la evaluación de los criterios. *LBTL ={ nada, muy bajo, bajo, medio, alto, muy alto, perfecto}.*
2. Los especialistas evalúan cada criterio usando alguno de los términos lingüísticos ver Tabla II.

Tabla II: Structure of the evaluation of the specialists for each criterion. *Ci* represents the criteria and ej represents the specialists.

|  |  |  |  |
| --- | --- | --- | --- |
| **Critera** | **Experts** | | |
|  | … |  |
|  |  | … |  |
| … | … | … | … |
|  |  | … |  |
|  |  | … |  |
| … | … | … | … |
|  |  | … |  |
| … | … | … | … |
|  |  | … |  |

1. Expert preference is transformed into fuzzy sets based on the linguistic variable.
2. Following the 2-tuples model [35] of computation with words, the evaluations of the experts are added, consolidating them for each criterion to be evaluated. See equation 1

 

1. Analyze the results see Tabla III

Tabla III Evaluation results.

|  |  |
| --- | --- |
| Criteria | Summarized expert evaluation |
| Applicability in different scenarios | High |
| Comprehensibility of the proposal | Very High |
| Usability of the proposal | Mean |
| Coverage of the indicators of the areas of knowledge of project management. | High |
| **Final aggregation** | **High** |

The criterion with the lowest evaluation was "Usability of the proposal" motivated by the use of non-traditional methods and the need to have platforms such as R that implement the proposed algorithms. The best evaluated criterion was the comprehensibility and applicability of the proposal. This is a positive element that shows the acceptance of the experts consulted for the application of the model in different scenarios. In general, the final evaluation given by the experts to the proposal was High.

**Analysis of the results of the normality analysis technique.**

A knowledge base of finished projects from the software project development environment of the University of Computer Sciences was experimented with, with more than 10 production centers dedicated to this activity. Each record in the database corresponds to a cut of a project where control and monitoring indicators are collected. A project cutoff is a periodic evaluation activity of a project, usually weekly. In this activity, the project's progress indicators are reviewed and decisions are made.

The knowledge base used presents the following distribution: total of 7,843 projects, 40% evaluated as Good, 14% evaluated as fair and 46% evaluated as bad. Each project cut-off was evaluated by human experts in such a way that there is a standard for comparing the quality of the classification results [36].

The experts evaluated the projects considering the indicators described in Table I. The assumption of normality of the samples was verified through the Shapiro – Wilk test. This showed that for each result of the metrics, at least one technique has probability values ​​(p-value) lower than 0.05, so the idea of ​​normal distribution is rejected with 95% confidence. Since not all samples follow a normal distribution, the use of non-parametric tests is chosen to verify significant differences.

**Analysis of the results of the comparison experiment technique of the different algorithms proposed**.

This experiment is applied on the same database of finished projects explained in the previous section and as a continuity of the analysis of the behavior of the algorithms suggested on it. For the comparison of the algorithms, the following set of metrics are defined as measurement criteria:

* Percent correct classifications: average achieved among the 20 partitions created for random cross-validation
* Number of false positives
* Number of false negatives
* MSE, following equation
* SMAPE, following equation:

Where n is the number of validation cases, is the expected output value and is the output calculated by the system. The metrics are calculated on the cases selected for the tests, once the algorithm learned from the training cases.

For the comparison of the algorithms proposed in the model, the random cross-validation method is applied with k = 20 test iterations and n = 7, this last parameter representing the 7 proposed algorithms. For each iteration, the data set is randomly divided into a training set with 70% of the cases and a test set with the remaining 30%.

As they are related samples, the Wilcoxon and Friedman tests are used. In both cases, significant differences are considered when p-value <0.05. In the case of the Wilcoxon test, the Monte Carlo method is applied for 99% of the confidence interval. During the application of the tests, in the cases where significant differences were found between the techniques, different groups of techniques were formed.

The groups formed meet the following property: the techniques of the same group do not have significant differences between them. Groups also have the following property: Group1 <Group2 <… Group\_n. The techniques in the smaller groups are those that report the best results, while those located in the higher groups have the worst results.

Regarding the metric "Percent correct classifications", the Friedman test was applied, comparing the samples obtained from applying the algorithms. In this context, the best result is considered to be the technique that statistically contributes the greatest value to the metric. As a result, significant differences are found between the techniques with p-value = "7.8547750364149e-18". The Wilcoxon test is applied, conveniently comparing two-by-two techniques, finally obtaining the following groups

Group 1: FISBR, FS.HGD, FIR.DM

Group 2: GFS.THRIFT

Group 3: GFS.LT.RS

Group 4: WM

Group 5: HyFIS

Regarding the metric “Number of false positives”, the Friedman test was applied, comparing the samples obtained from applying the algorithms. In this context, the best result is considered to be the technique that statistically contributes the lowest value of this metric. As a result, there are significant differences between the techniques with p-value = "1.57230970124996e-17". The Wilcoxon test is applied, conveniently comparing two by two techniques, finally obtaining the following groups:

Group 1: FISBR

Group 2: WM, HYFIS, FIR.DM,

Group 3: FS.HGD, GFS.LT.RS, GFS.THRIFT

In this case, the FISBR obtained the best results because it was the technique that obtained the least false positives. Getting false positives (rating Good when the project is actually rated Bad) in the area of ​​project management is detrimental; Evaluating the Good of a project that is not excellent can cause damage in decision-making and cause unrealistic planning.

Regarding the metric "Number of false positives" in this context, the best result is considered to be the technique that statistically contributes the lowest value of this metric. The Friedman test was applied, comparing the samples obtained. As a result, significant differences are found between the techniques with p-value = "5.19919647410999e-14". The Wilcoxon test is applied, conveniently comparing two by two techniques, finally obtaining the following groups:

Grupo 1: WM

Grupo 2: FISBR, GFS.LT.RS

Grupo 3: FS.HGD, HYFIS, GFS.THRIFT

Grupo 4: FIR.DM

It can be affirmed that the WM technique obtains significantly better results in terms of the number of false negatives on the set of test cases.

Obtaining false negatives (rating Bad when the project is actually rated Good) in the project management area is not as damaging as false positives, but evaluating a project that is not actually rated as Bad may lead to taking bad decisions.

Regarding the “MSE” metric, in the context of this experiment, the technique that statistically contributes the least value to the metric is considered the best result. The Friedman test was applied, comparing the samples obtained. It was shown that there are significant differences between the techniques with p-value = "1.45922235866844e-16". The Wilcoxon test is applied, conveniently comparing two by two techniques, finally obtaining the following groups:

Group 1: FISBR, FIR.DM

Group 2: GFS.THRIFT, WM

Group 3: GFS.LT.RS

Group 4: FS.HGD

Group 5: HyFIS

It can be stated that the FISBR and FIR.DM techniques obtain significantly better results in terms of the mean square error over the set of test cases.

Regarding the “SMAPE” metric, in this context, the best result is considered to be the technique that statistically contributes the lowest value of this metric. The Friedman test was applied comparing the samples obtained. As a result, significant differences are found between the techniques with p-value = "1.45922235866844e-16". The Wilcoxon test is applied, conveniently comparing two by two techniques, finally obtaining the following groups:

Grupo 1: FISBR

Grupo 2: WM, FIR.DM

Grupo 3: GFS.LT.RS, GFS.THRIFT

Grupo 4: FS.HGD

Grupo 5: HYFIS

It can be stated that the FISBR algorithm obtains significantly better results in terms of the percentage error of the symmetric absolute mean over the set of test cases. FISBR technique generates the best system for project evaluation since it achieves the best results regarding the metrics: percentage of correct classifications (97%), number of false positives, MSE and SMAPE.

The fuzzy rules generated from the algorithms allow the projects to be evaluated efficiently, taking into account the high percentage of correct classification obtained in them. In addition, with the use of the proposed techniques, the treatment of phenomena such as imprecision, vagueness and uncertainty contained in the primary data is achieved, as well as the changing conditions in management styles and the level of maturity reached by the organization that provides the data

### Conclusions

From the analysis of the results previously presented, the following conclusions are reached:

* The development of a software ecosystem to aid decision-making for the integrated management of projects based on open source technologies and free software was identified as an opportunity. The BusinessRedmine ecosystem was developed with functionalities aligned with international standards in the domain of project management.
* The introduction of the results into social practice made it possible to significantly reduce time and costs in the control and monitoring of projects in the beneficiary organizations.
* Artificial intelligence techniques are incorporated into the proposed ecosystem to aid decision-making, enabling a better treatment of phenomena such as imprecision, vagueness and the uncertainty contained in the primary data. The proposed ecosystem makes intensive use of these techniques both for the classification of projects and for the visualization of the results.
* The use of algorithms such as FISBR allow the system to learn from evidence and can then use the knowledge to classify new instances. It facilitates the adaptation of systems to changing environments associated with the management styles and the level of maturity reached by the organizations that manage the projects.
* It was found that the FISBR technique had a significantly better performance than other techniques reported in the bibliography, showing 97% effectiveness in classification.

As future work, the authors propose the combination of systems to help the automatic classification of projects with recommendation systems that improve the usability of the models and platforms for project management.

### References

* + 1. Christophe, C., Daan, A. y Ingrid, A.: BIM and ISO 19650 from a project management perspective (2018). [en línea]. S.l.: European Federation of Engineering Consultancy Associations. [Consulting: 10 septiembre 2021]. Disponible en: https://www.efcanet.org/sites/default/files/2020-01/390764\_BIM%20booklet.pdf.
    2. ISO 19650-1: Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modelling (BIM)—Information Management Using Building Information Modelling. Part 1: Concepts and Principles. ISO (2018)[en línea]. [Consulting: 10 septiembre 2021]. Disponible en: <https://www.iso.org/standard/68078.html>
    3. Mesároš, P., Mandičák, T. y Behúnová, A.: Use of BIM technology and impact on productivity in construction project management. Wireless Networks (2020) [en línea], [Consulting: 23 august 2021]. ISSN 1572-8196. DOI 10.1007/s11276-020-02302-6. Disponible en: https://doi.org/10.1007/s11276-020-02302-6.
    4. Saeed Mahdi Gaafar Sadeq, Algoritmos de estimación de distribuciones con tratamiento de restricciones para la construcción de cronogramas de proyectos, Tesis de Doctorado, Departamento de Investigaciones en Gestión de Proyectos, Universidad de las Ciencias Informáticas (2021).
    5. Loyola, M.: Encuesta Nacional BIM 2019: Informe de Resultados. [en línea]. Santiago: Universidad de Chile: ISBN: 978-956-401-111-0. (2019). Disponible en: www.bim.uchilefau.cl.
    6. Cudzik, J. and Radziszewski, K., Artificial Intelligence Aided Architectural Design [en línea]. S.l.: eCAADe (Education and Research in Computer Aided Architectural Design in Europe) and Faculty of Civil Engineering, Architecture and Environmental Engineering, Lodz University of Technology. Proceedings of the 36rd eCAADe Conference. ISBN 978-94-91207-15-0. (2018) [Consulting: 11 septiember 2020]. Disponible en: <https://mostwiedzy.pl/pl/publication/artificial-intelligence-aided-architectural-design,146441-1>
    7. González, C.F.L., Metodología BIM (Building Information Modeling) aplicada a la prevención de riesgos laborales (PRL). Journal Bim & Construction Management, **1**(1), pp. 20-30. ISSN 2659-6962, (2019)
    8. Hoar, C., Atkin, B. y King, K., Artificial intelligence: What it means for the built environment [en línea]. London: Royal Institute of Chartered Surveyors (RICS). (2017), Global/august 2017/DML/22396/RICS INSIGHTS. Disponible en: www.rics.org.
    9. Dave, B., Buda, A., Nurminen, A. y Främling, K.: A framework for integrating BIM and IoT through open standards. Automation in Construction, **95**, pp. 35-45. ISSN 0926-5805. DOI 10.1016/j.autcon.2018.07.022. (2018)
    10. DunbaR, G., Project Management Failures - Standish (Chaos) reports. (2016) [en línea]. [Consulting: 21 october 2021]. Disponible en: <https://www.linkedin.com/pulse/project-management-failures-standish-chaos-report-2015-dunbar>.
    11. Lynch, J., Project Resolution Benchmark for IBEX Financial Corp. The Standish Group International, Inc, pp. 11, (2018)
    12. Johnson, J.: CHAOS Report: Decision latency theory: It is all about the interval. S.l.: Lulu. com. ISBN 978-0-692-04830-6, (2018)
    13. Pérez Pupo, I., Vacacela, R.G., Piñero Pérez, P.Y., Saeed Mahdi, G. S. and Peña Abreu, M.: Experiencias en el uso de técnicas de softcomputing en la evaluación de proyectos de software. Investigación Operacional, **41** (1), pp. 108-119, (2020)
    14. Mossalam, A. y Arafa, M., The role of project manager in benefits realization management as a project constraint/driver. Housing and Building National Research Center, HBRC Journal, pp. 56-67, (2014)
    15. Pacelli, L., Grandes Errores en la Gestión de Proyectos. Resumen autorizado de: The Project Management Advisor: 18 Mayor Projetc Screw-Ups, and Cut Them off the Pass. *Financial Times Prentice Hall*, (2004).
    16. PMI, The standard for Project Management and a guide to the Project Management Body of Knowledge (PMBoK guide) Seventh Edition / Project Management Institute. Project Management Institute, (2021).
    17. Northrop, L., & Jones Lawrence, M. G: Introduction to software product lines adoption. SPLC (2013).
    18. Krueger, C W. Mechanical product lifecycle management meets product line engineering. SPLC, pp. 316-320. (2015).
    19. Bosch-Sijtsema, P. M., & Bosch, J.: Plays nice with others? Multiple ecosystems, various roles and divergent engagement models. Technology Analysis & Strategic Management, **27(**8), Pp. 960-974, (2015).
    20. Bosch-Sijtsema, P., & Bosch, J. Aligning innovation ecosystem strategies with internal, R&D. Proceedings of the 2014 IEEE ICMIT, 7th international conference on management, of innovation and technology. pp.424-430, http://dx.doi.org/10.1109/ICMIT.2014.6942464). Singapore. (2014).
    21. Bosch, J. ESAO: towards data-and ecosystem-driven R&D. Proceedings of the 18th International Software Product Line Conference **1**, Pp.363-363). ACM. (2014)
    22. Jansen, S., Handoyo, E., & Frota Alves, C.: Scientists' Needs in Modelling Software Ecosystems. Proceedings of the 2015 European Conference on Software Architecture Workshops, pp 1-44 http://doi.acm.org/10.1145/2797433.2797479, DOI 10.1145/2797433.2797479). Dubrovnik/Cavtat, Croatia. (2015)
    23. Krueger, C., & Clements, P. Second generation systems and software product line engineering. SPLC 2015, pp. 388-389. (2015)
    24. Maninkas, K., & Marius Hansen, K. Software ecosystems a systematic literature review. Journal Systems and Software, **86** (5), pp: 1294-1306. DOI=http://dx.doi.org/10.1016/j.jss.2012.12.026. (2013)
    25. Group 451, Open Source IsNot a Business Model, CAOS REPORT url: https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CBsQFjAAahUKEwicyuimusfIAhWFJh4KHctEBSc&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F0%2F4%2F2%2F04246FB1-0BF6-44F4-BC44-4CCB220E1711%2F451\_-\_08\_OCT\_-\_CAOS\_Report\_9\_Open\_Source\_is\_Not\_a\_Business\_Model.pdf&usg=AFQjCNHxtYWz4J9z-jVQDZteoXP180IxBQ&bvm=bv.105039540,d.dmo . [Consulting November 2021].
    26. The eclipse foundation. The Eclipse Foundation - open source community website. In. Ottawa, Ontario, Canada: The Eclipse Foundation, (2012)
    27. Piñero Pérez, Pedro .Y.., et al., BusinessRedmine Sistema para la toma de decisiones empresariales y la gestión de proyectos, Número de Registro de Propiedad Intelectual: Cuba. CENDA 4076-12-2019
    28. Team, SEI CMMI Production. CMMI for Development v1. 3. Lulu. com, (2010).
    29. Nomura, H., Hayashi, I. & Wakami, N. 1992. A Learning Method of Fuzzy Inference Rules by Descent Method. IEEE International Conference on Fuzzy Systems. San Diego, CA
    30. Ishibuchi, H., Nozaki, K., Nozaki, K., Tanaka, H., Tanaka, H., Hosaka, Y., y otros. (1994). Empirical study on learning in fuzzy systems by rice taste analysis. Fuzzy Sets and Systems, 64(2), 129-144.
    31. Alcalá, R., Alcalá, J., & Herrera, F. (2007). A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. IEEE Transactions on Fuzzy Systems, 15(4), 616 – 635
    32. Thrift, P. 1991. Fuzzy logic synthesis with genetic algorithms. Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA91). San Diego (United States of America).
    33. Kim, J., & Kasabov, N. (1999). HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Networks, 12(9), 1301-1319
    34. Wang, L.-X., & Mendel, J. (1992). Generating fuzzy rule by learning from examples. IEEE Transactions Systems, Man, and Cybernetics, 22(6), 1414-1427
    35. Herrera F. and L. Martınez. A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Transactions on Fuzzy Systems, 8(6):746–752, 2000.
    36. S. Torres, Y. Piñero, P. Y. Piñero, L. F. Capretz: Creation and evaluation of software teams - A social approach. International Journal of Manufacturing Technology and Management 12/2014; 28(4/5/6). DOI:10.1504/IJMTM.2014.066695 <https://www.researchgate.net/publication/270703524_Creation_and_evaluation_of_software_teams_-_A_social_approach>