Journal of Applied
Technical and Educational Sciences
jATES
ISSN 2560-5429
http://jates.org

Web Application for Simulation of Agile Software Projects – Stage One: Team Builder

Emira Mustafa Moamer Alzeyani [1], Peter Baláž [2,3], Csaba Szabó [1]

[1] *Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, {emira.mustafa.moamer.alzeyani|csaba.szabo}@tuke.sk*

[2] *Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, peter.balaz.2@student.tuke.sk*

[3] *Deutsche Telekom IT Solutions Slovakia, Žriedlová 13, 040 01 Košice, Slovakia*

Abstract: *Training of software development specialists is facing the main problem of providing a good enough amount of practical experience over a variety of real project situations. One approach is using simulators that model scenarios based on real projects but make them repeatable for a better teaching outcome: allowing experimentations on "what if" cases without any real costs or risk. In this work, we analyse the agile development of products, artificial intelligence and data used in machine learning. We implemented a web application, which has the task of mediating agile product development. This application is aimed at product owners in order to facilitate their work through appropriately integrated artificial intelligence. When testing the application, we evaluated different versions of the prediction models and verified web interface with direct testing. The results and data from the testing are evaluated and interpreted in the chapter on verification of the proposed solution.*

Keywords: *agile development; artificial intelligence; product owner; web application*

1. Introduction

Software development goes deep into the past. Software has been developed by poor and inappropriate practices. To improve software development, new development methods and procedures have been developed to speed up and improve this process (Salameh, 2014).

In today's world, the most popular method for development is agile methodology, where the main feature is the flexible team. When developing software, the key task is to communicate with the client, which determines the requirements for product development. Agile methods use different procedures, but they all try to secure quality software and deliver all client requests.

The main intermediaries of clients' requirements are the product owners, which persons have the task of communication with the client. There are often misunderstandings in communication with the client and based on that, software development errors arise.

Since artificial intelligence can help in various industries, we tried to find a way how to use it appropriately to facilitate agile development. So, we will try creating a web application and train the model for prediction that we can properly integrate to improve the agile development for product owners. By removing one of the product owner's obligations, such as distribution tasks and left it to the application, we would help and facilitate the work of the product owner.

2. Goals

The aim of this work is to research agile development, artificial intelligence and data processing so that we have a sufficient understanding of the issue. We will try to create a web application that could use suitable artificial intelligence to improve, speed up and facilitate agile development for product owners.

3. Background

3.1. Artificial intelligence

Artificial intelligence (AI) is increasingly being used in the web environment because users want exceptional applications that have customized content and are easy to use. Application of artificial intelligence leads to a process that begins with analysing the data and then search for specific patterns in the data set, since AI creates a model that can predict similarity in patterns (Alsolai & Roper, 2020).

There are many development frameworks that deal with supporting artificial intelligence such as CNTK, ONNX, Apache Mahout or Caffe. But very often, there is the construction, training and validation of AI models executed in Python environment such as Jupyter notebook. Resulting models are converted into models to be useable in JavaScript environment for the web only after that. TensorFlow framework came up with the offer of the library for training and using models directly in JavaScript, which uses high-level and low-level API, with the support of inclusion of Python-trained models as well.

3.2. Data processing for machine learning

It is often important to obtain information from the data using data analysis or knowledge through a process (Han & Kamber, 2006). We need to get raw data in the beginning, and we must process them using various processes such as cleaning up unnecessary data, filtering data or their standardization. The data modified in this way then usually enters the methods that help us using data to describe the past, whether to predict the future or help us answer the questions we are looking for (Cuesta, 2015).

The data that contains the knowledge can take various forms. We can divide them into three parts, namely structured, semi-structured and unstructured.

These can be categorized into several parts (Holcik, 2012):

- symbolic
- numerical
- categorical
- nominal
- binary
- multimedia

Under the term big data, we perceive a huge amount of data and because of the processing of such an amount of data, we would need very expensive resources and computing units that are difficult to access.

Data mining is the process by which we examine what functions we need to apply, so that we can gain data knowledge reliably and efficiently. We need to find out what challenges we can face when creating a data mining process.

3.3. Agile development

Agile development is the process where software project managers are facing challenge to effective management of the team at software development process. Software development was characterized not only as a technical process, but also as a social one, which requires effective relationship management to facilitate critical skills and expertise of team members (Aruping, Venkatesh, & Agarwal, 2009).

Examples of methods that fall under agile are extreme programming, Scrum, crystal clear, feature driven development, dynamic system development methodology and Kanban (Merzouk, Cherkaouia, Marzak, & Nawal, 2021).

When it comes to agile development, agile principles and values must also be adopted by project teams and must create a specialized team that stimulates agile actions and consists of the following roles (Loiro, Castro, Avila, Cruz-Cunha, Putnik, & Ferreira, 2019):

- product owner

- team leader
- team members

The product owner is the person who acts as the intermediary between the client or other colleagues and his team, (s)he is considered an expert in project management, where it mainly focuses on the client's needs.

A team leader is a person who is responsible for leading and supporting team members, solves external problems to keep the team focused.

Team members are a group that organizes itself and has different functions with different skills, which create the product. In our models, we consider seven work positions for team members, namely:

- front-end
- back-end
- design
- database
- IT security
- tester
- documentation.

3.4. AI support for agile development

Agile software project management generates data as all managements do. According to the characteristics presented above, these data could be related to e.g. activities of a specific person, team or sprint.

Ona can try to substitute expert systems for temporarily unavailable domain experts or other stakeholders to support continuous development (Annaiahshetty & Prasad, 2013). The rule base is very critical because different experts used to apply the rules differently.

Simulators built upon project data can be used to address risk assessment (Lunesu, Tonelli, Marchesi, & Marchesi, 2021) or process improvement (Sobiech, Eilermann, & Rausch, 2015), (Maxim, Kaur, Apzynski, Edwards, & Evans, 2016) and (Tanveer, Guzmán, & Engel, 2016) as well. Authors address the whole process or the planning phase, which is known to be the most critical. AI support in these simulators solves the problem of complicated analytical models.

4. Implementation and Results

4.1. Data collection

In order to be able to train the prediction model, we must first obtain data. For our work we need to get data that contains the task for the team member and the position to which this task was assigned. The best way to obtain data was from open-source git projects and their commits, which are freely accessible such as (Chen, 2022) and (Markovtsev & Arrazolo, 2021). Such raw data that we obtained from these sources can be hardly analysed. As first step in this process we need to filter data based on keywords, so we have enough specific data for given roles, then we need to remove inappropriately, disruptive data. This process should ensure that we will have good enough data to have a well-trained model.

4.2. Model training

To train our prediction model we used the JavaScript TensorFlow library which we will use in NodeJS environment. We also used the sentence encoder TensorFlow library, which searches for similarities in sentences and creates tensors from them, which are further inserted into the model. For the prediction model, we chose a sequential model, in which we inserted one dense layer with an active SoftMax function and a tensor, which we created from the given data. The model was compiled using the categorical cross-entropy loss function and Adam optimizer and we choose learning rate 0.001.

As the next step we trained the model with batch size of value 4 and we choose validation split of training data to be 10 % of the whole data set, and shuffling data with every epoch. We trained the model on 250 epochs because it is the number of steps, we can assure we have enough data to train the model for prediction and we do not over-train the model.

4.3. Verification of trained models

Models were trained on different numbers of records, and we also tried to train several models for prediction of separate positions. The groups were divided into 250, 500 and 1000 records of trained models for each position. Each model is tested using the two loss functions categorical cross-entropy and mean squared error (MSE), respectively, and we were using activation functions such as RELU, Sigmoid, SoftMax and Linear. The units have the form of training data, where each layer of outputs is the input to the next layer. Every model is tested using the Adam optimizer with learning rate values of 0.01 and 0.001.

In Tab. 1, we can see two of the best and worst trained models for each group.

Table 1. Evaluated models sorted by average prediction success

Records	Epochs	Optimizer	Learning rate	Loss function	Activation function	Average prediction success (%)
1000	250	Adam	0.01	MSE	RELU	78.9
1000	250	Adam	0.01	Categorical cross-entropy	Sigmoid	79.5
250	250	Adam	0.01	Categorical cross-entropy	SoftMax	81.7
250	250	Adam	0.01	Categorical cross-entropy	Sigmoid	82.3
1000	250	Adam	0.001	MSE	RELU	82.6
500	250	Adam	0.01	Categorical cross-entropy	Linear	82.7
1000	250	Adam	0.001	Categorical cross-entropy	Sigmoid	83.1
500	250	Adam	0.01	Categorical cross-entropy	RELU	83.6
250	250	Adam	0.001	MSE	Linear	86.5
250	250	Adam	0.001	Categorical cross-entropy	SoftMax	89.7
500	250	Adam	0.001	Categorical cross-entropy	Sigmoid	91.2
500	250	Adam	0.001	Categorical cross-entropy	SoftMax	93.1

As our testing shows, the model that passed the best was trained on 500 records with the Adam optimizer, loss function Categorical cross-entropy and activation function SoftMax with an average prediction success rate of 93.1%.

The worst model is the one that was trained on 1000 records, using the loss function of MSE and the RELU activation function with an average prediction success rate of 78.9%.

4.4. Testing multiple models for prediction

When training the models, we created a version where we trained the model for each work position in the team. So, we had seven models that we trained using the parameters that from previous testing came out as the most appropriate. We have used the Adam optimizer with a learning rate of 0.001, categorical cross-entropy loss function and SoftMax activation function.

We also tested this version on 70 tasks. The tasks had a distribution to allow building of 10 teams of 7 members at different work positions. As can be seen in Tab. 2, resulting prediction success rate was 81.4 %. That is why we decided not to use this version, due to the low value of the successful prediction.

Table 2. Evaluation of multi-model solution

Position	Successfully predicted positions out of 10 expected
front-end	8
back-end	7
design	10
database	8
IT security	7
tester	9
documentation	8

4.5. Integration to created application

We created an application to prove that our trained models are suitable for use in agile development. Application will be using several models of Scrum methodology as project, sprints, stories, and tasks, as shown on Fig.1.

Our application provides a support for the common process of agile planning of these models. Starting with project creation, followed by defining the stories and their tasks as it is with agile requirements modelling. Sprints can be then created based on task selection. Detailed flow of activities is shown on Fig. 2.

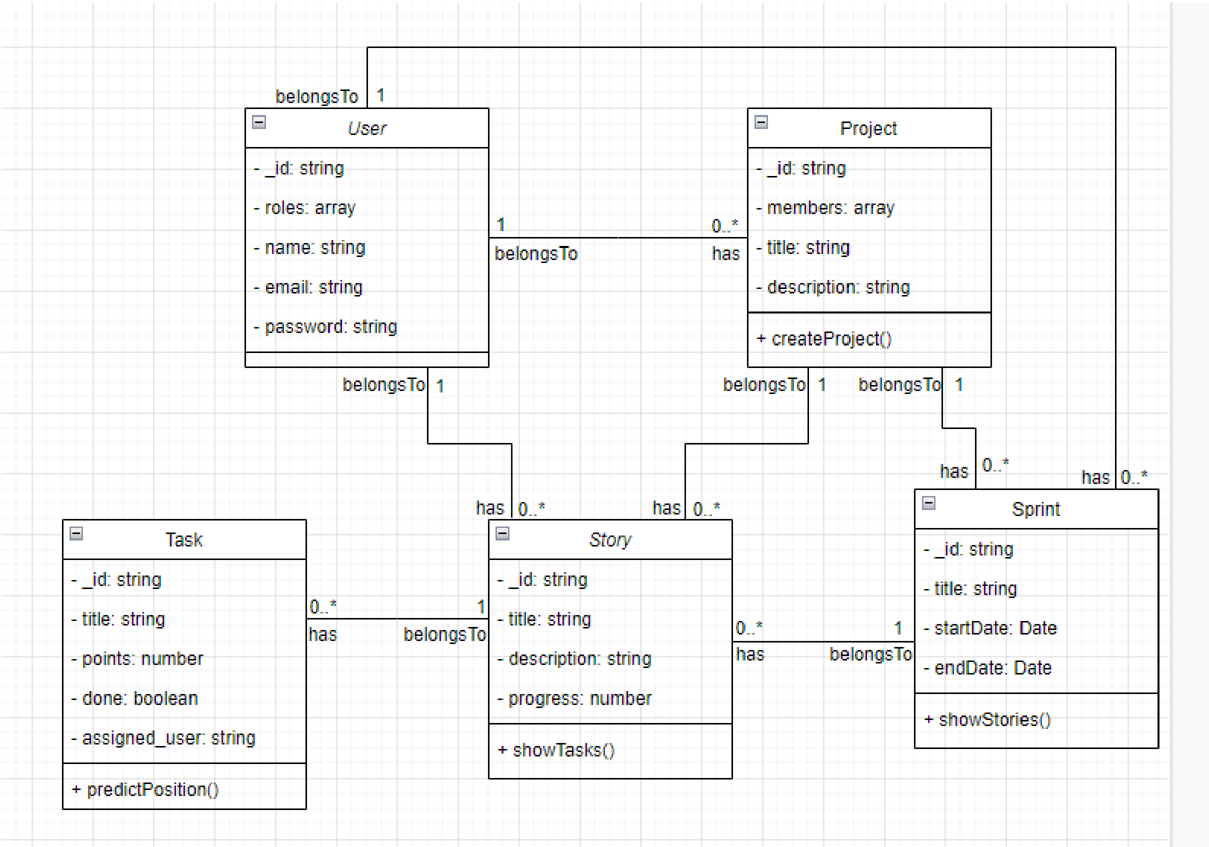

Fig. 1. Relations between main project objects

For implementation of the application, we used frameworks such as ReactJS, NodeJS, MongoDB and for the integration of artificial intelligence we chose a TensorFlow framework.

We have decided that we will use our prediction model for tasks, where when user creates a task, server will start a process where our model predicts a position of given task. As soon as we get all predictions done, the final selection of the team member to which the task will be assigned takes place by selection according to occupancy of the members. That is, if the member with the highest priority has many tasks, then this task is assigned to the next one in the predicted suitability order.

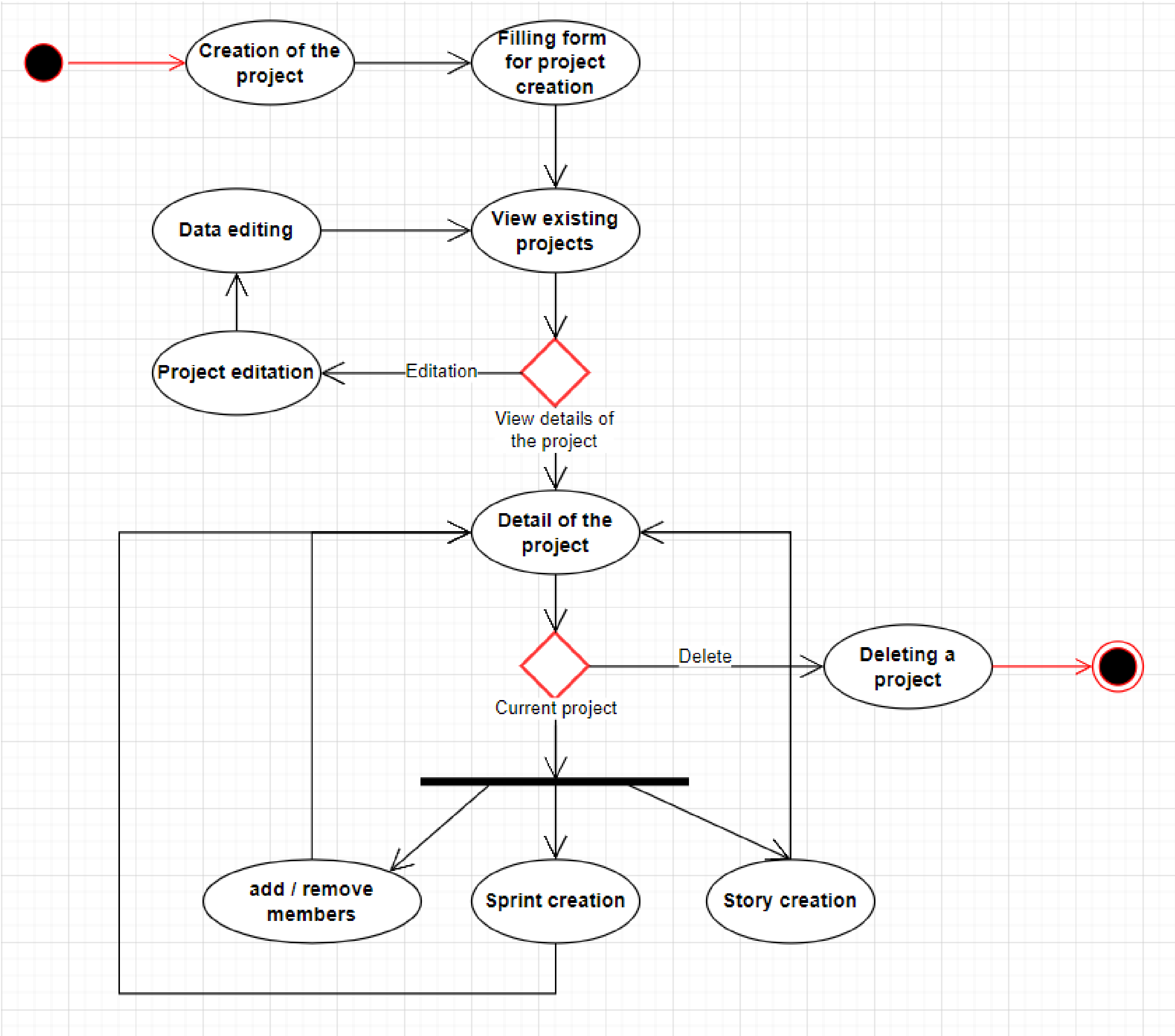

Fig. 2. Activities implemented in our application

5. Conclusion

The main goal of this work was to use AI and efficient methods appropriately to be able to improve agile development for product owners which can also be used in general in the development of different products and services (Dukan 2013) and improve problem solving abilities (Kovari 2020). In the first part, we analysed the frameworks dealing with artificial intelligence, data processing for machine learning and agile development. We used the knowledge we gained during analysis to data processing with which we trained a model for predicting tasks for team members.

We trained several models using various parameters, which we tested and selected the most suitable parameters, and we used them to train the model. We have therefore implemented a web application that will be able to perform agile development and through the integration of our prediction model make it easier for product owners to take on one of their responsibilities.

This application was based on the design of a conceptual model and a sequence of screens. We have created a user-friendly interface, which has been subsequently tested with the aim to improve to ensure an improved user experience. Our application should be able to mediate agile development with an orientation towards the product owner, and by taking one of the responsibilities it would be able to facilitate their work.

In our future work, we will reconsider the cases we compared the single model implementation with the multi-model one. Main reasons switching to the version using the collection of encoders could be the easier way of model extension – re-training of separate tensors/encoders instead of re-training of a huge model. Specially, if considering that software developers gain new knowledge and experience during each task implementation (almost) independently on their colleagues, even if they fail to complete the assigned task.

References

Annaiahshetty K., & Prasad, N. (2013) Expert System for Multiple Domain Experts Knowledge Acquisition in Software Design and Development, 2013 UKSim 15th International Conference on Computer Modelling and Simulation, 2013, pp. 196-201, doi: 10.1109/UKSim.2013.124.

Alsolai, H., & Roper, M. (2020) A systematic literature review of machine learning techniques for software maintainability prediction. J. Inf. Softw. Technol., 119, 106-214. https://doi.org/10.1016/j.infsof.2019.106214

Aruping, L. M., Venkatesh, V., & Agarwal, R. (2009) A control theory perspective on agile methodology use and changing user requirements. Information Systems Research, 20(3), 377–399. https://www.jstor.org/stable/23015471

Chen, X. (2022) AwesomeData: Awesome Public Datasets. Date of access: 2022-01-14. [Online]. Available: https://github.com/awesomedata/awesome-public-datasets

Cuesta, H. (2015) Data analysis in practice. ISBN-10: 1783280999, ISBN-13: 978-1783280995

Dukan, P. (2013). Cloud-based smart metering system. In 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics, pp. 499–502.

Holcik, J. (2012) Data analysis and classification. Academic publishers CERM. ISBN 978-80-7204-793-2

Han, J., & Kamber, M. (2006) Data mining: Concepts and techniques. 2nd Edition, Diane Cerra Publisher, San Francisco.

Kovari, A. (2020). Study of Algorithmic Problem-Solving and Executive Function. Acta Polytechnica Hungarica, 17(9), 241–256.

Loiro, C., Castro, H., Avila, P., Cruz-Cunha, M. M., Putnik, G. D., & Ferreira, L. (2019) Agile project management: A communicational workflow proposal. Procedia Computer Science, 164, 485–490. https://doi.org/10.1016/j.procs.2019.12.210

Lunesu, M. I., Tonelli, R., Marchesi, L., & Marchesi, M. (2021) Assessing the Risk of Software Development in Agile Methodologies Using Simulation. IEEE Access, vol. 9, pp. 134240-134258, 2021, doi: 10.1109/ACCESS.2021.3115941.

Markovtsev, V., & Arrazolo, S. (2021) 452M commits on GitHub, Date of access: 2022-01-14. [Online]. Available: https://data.world/vmarkovtsev/452-m-commits-on-github

Maxim, B. R., Kaur, R., Apzynski, C., Edwards D., & Evans, E. (2016) An agile software engineering process improvement game. 2016 IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-4, doi: 10.1109/FIE.2016.7757682.

Merzouk, S., Cherkaouia, A., Fatima-zahra, Marzak, A., & Nawal, S. (2021). Smart Irrigation: Case study for Hybrid-SIX methodology. Procedia Computer Science, 191, 524-529. https://doi.org/10.1016/j.procs.2021.07.077.

Salameh, H. (2014). What, When, Why, and How? A Comparison between Agile Project Management and Traditional Project. European Centre for Research Training and Development UK, 52-74.

Sobiech, F., Eilermann, B., & Rausch, A. (2015) A heuristic approach to solve the elementary sprint optimization problem for non-cross-functional teams in Scrum. SIGAPP Appl. Comput. Rev. 14, 4 (December 2014), 19–26. https://doi.org/10.1145/2724928.2724930

Tanveer, B., Guzmán, L., & Engel, U. M. (2016) Understanding and improving effort estimation in Agile software development: an industrial case study. In Proceedings of the

International Conference on Software and Systems Process (ICSSP '16). Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2904354.2904373

About Authors

Emira Mustafa Moamer ALZEYANI received her M. Sc. from Faculty of Electrical Engineering and Informatics, Technical University of Košice in 2018, then she was working as assistant lecturer at the College of Electronic Technology, Tripoli, Libya. She is a first-year PhD student at Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice. Her research interests include agile software project management improvements and artificial intelligence in agile projects.

Peter BALÁŽ received his M. Sc. from Faculty of Electrical Engineering and Informatics, Technical University of Košice in 2022. He is a software developer at Deutsche Telekom IT Solutions Slovakia. His research interests include agile project management improvements.

Csaba SZABÓ received his M. Sc. from Faculty of Electrical Engineering and Informatics, Technical University of Košice (TUKE) in 1998 and PhD in Informatics from TUKE in 2008. He is an associate professor at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice. His research interests include classic and agile software project management, green software and intelligent systems.