ISSN 2664-9969

UDC 004.02
DOI: 10.15587/2706-5448.2024.310411

Kostyantyn Kharchenko,
Oleksandr Beznosyk,
Bogdan Bulakh,
Bogdan Kyriusha,
Vadym Yaremenko

FORECASTING SOFTWARE DEVELOPMENT COSTS IN SCRUM ITERATIONS USING ORDINARY LEAST SQUARES METHOD

During scrum iterations, it is possible to apply cost forecasting for software testing and operation, if the data from previous iterations are known. Since the data for estimating the scope of work and the deadline within one sprint are accumulated during the project execution, it is possible to use such data to build a forecasting algorithm for the estimated parameters of the subsequent sprints.

The approach is based on refining the assessment provided by the development team and the scrum master in a specific metric. The main parameters for evaluation are the execution time and the amount of work performed. As a result of forecasting, it is possible to obtain clarifications for the team's assessment regarding the scope of work for the next sprint. This estimate is based on planned and actual data from the previous sprints.

The article discusses the method of least squares and the proposed code for a machine learning model based on this method. An example and graphs for iterations in scrum and corresponding forecasting for the next sprints are presented.

The use of the least squares method allows creating a mathematical model that can be adapted to different project conditions, providing flexibility and accuracy in forecasting. For example, the study uses the real data from the previous sprints, which includes the team's resource assessment and actual expenditures. Based on these data, a model was built that demonstrates a high correlation between predicted and actual costs, confirming the effectiveness of using the least squares method.

So, the least squares method is an effective tool for forecasting software development costs in scrum iterations. This method allows development teams to better plan their resources and timelines, contributing to the overall efficiency of the project.

Keywords: *cost forecasting, scrum, machine learning, least squares method, iterations, software development.*

Received date: 13.07.2024
Accepted date: 23.08.2024
Published date: 26.08.2024

How to cite
Kharchenko, K., Beznosyk, O., Bulakh, B., Kyriusha, B., Yaremenko, V. (2024). Forecasting software development costs in scrum iterations using ordinary least squares method. Technology Audit and Production Reserves, 4 (2 (78)), 30–33. https://doi.org/10.15587/2706-5448.2024.310411

1. Introduction

As mentioned in [1], it is possible to optimize testing costs during scrum iterations. In this work, it is possible to consider the possibility of forecasting costs at each iteration using scrum.

The relevance of this research stems from the need to improve the accuracy of cost forecasting in software development within the scrum [2] framework, which is crucial for effective project management in the global and Ukrainian IT sector.

As the input data, it is possible to take two important components. The first component is the team's estimation of resources to perform certain scrum iteration. The second component is the real amount of resources spent on such iteration.

If to accumulate pairs of such estimates and real costs over several scrum iterations, then it is possible to build a method of forecasting costs in subsequent iterations. In this method, the input is the estimate of the resource for the next iteration, and the prediction result gives a correction to the proposed estimate, either higher or lower [3–8].

Ukraine, as a country with a developed IT sector, has the potential to successfully apply global experience in cost forecasting for software projects. The advantages of implementing such forecast methods, like Ordinary Least Squares (OLS) [4], in our context lie in their ability to significantly improve cost planning accuracy and enhance resource management within the scrum framework. This can help Ukrainian companies compete more effectively on the international market. However, the specific characteristics of Ukrainian projects, such as limited resources, the large number of startups with uncertain business models, and changes in the regulatory environment, may complicate the direct application of international experience without adaptation.

The aim of this research is to develop and implement a method for forecasting software development costs during scrum iterations using the OLS method. This approach aims to identify patterns between planned and actual costs, enabling more accurate estimations of resources for future iterations based on historical data. By applying this method, the study seeks to enhance the understanding of how to predict deviations in resource usage, thereby providing a systematic approach to improving the accuracy of project planning in Agile environments.

2. Materials and Methods

It is possible to suppose that the *planned* estimate of the resource [1] at each scrum iteration is set by a constant, denote it as S (1). Then the set of input estimates at several previous scrum iterations is:

$$S=\{P_1+Q_1,\ P_2+Q_2,\ P_3+Q_3,\ \ldots,\ P_n+Q_n\}, \tag{1}$$

where P and Q are some metrics used by a development team, and the *actual* received values of spent resources look like:

$$R=\{R_1,\ R_2,\ R_3,\ \ldots,\ R_n\}. \tag{2}$$

So, at each scrum iteration, there is the *planned* value of total costs S_i and the corresponding *real* value R_i (2), which will be known after the iteration.

Then, let's denote a *predicted* estimate of the real costs of the resource in the next scrum iteration as R_{n+1}.

The task is: if there are previous values $S_1,\ldots,S_n$, for the new *planned* value S_{n+1} to estimate the *real* value of costs for the future iteration R_{n+1}:

$$R_{n+1}=F(\{S\},\ \{R\},\ S_{n+1}). \tag{3}$$

Below is the list of the well-known methods of forecasting:

– Autoregression (AR);
– Moving Average (MA);
– Autoregressive Moving Average (ARMA);
– Autoregressive Integrated Moving Average (ARIMA);
– Seasonal Autoregressive Integrated Moving-Average (SARIMA);
– Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX);
– Vector Autoregression (VAR);
– Vector Autoregression Moving-Average (VARMA);
– Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX);
– Simple Exponential Smoothing (SES);
– Holt Winter's Exponential Smoothing (HWES).

Let's use the Ordinary Least Squares (OLS) method from the Python library Statsmodels [3, 4] as a forecasting method.

Suppose that the set of input estimates on several previous scrum iterations consists of two sets P and Q (the appropriate metrics are chosen by the project manager before starting work). In [1], P and Q represent the costs at each iteration and the budget for covering unforeseen losses, respectively. Let the planned resource estimates be represented at each iteration as the sum of the elements of two arrays:

$$S_i=P_i+Q_i, \tag{4}$$

$$P=\{1,\ 2,\ 1,\ 3,\ 2,\ 1,\ 5,\ 3,\ 2,\ 1\},$$

$$Q=\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1\}.$$

Then let's apply a set to the input of the forecasting method:

$$S=\{1.1,\ 2.2,\ 1.3,\ 3.4,\ 2.5,\ 1.6,\ 5.7,\ 3.8,\ 2.9,\ 2.0\}. \tag{5}$$

Accordingly, the actual cost of resources at each iteration, for example, is:

$$R=\{1.65,\ 2.75,\ 0.85,\ 3.95,\ 3.05,\ 2.15,\ 6.25,\ 4.35,\ 2.65,\ 1.55\}. \tag{6}$$

The actual R resources will differ from the S resources planned by the project manager. Having pairs of R_i and S_i as input, it is possible to predict real costs for future iterations. Let's use the following python code for forecasting:

```
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

def prepare_data(S_data):
  """ Prepare data for the model """
  return sm.add_constant(S_data)

def fit_model(X, Y):
  """ Fit an OLS model and return the summary """
  model = sm.OLS(Y, X).fit()
  return model

def plot_data(real_data, planned_data, forecast_data,
new_data_length):
  """ Plot the project data """
  plt.figure(figsize=(10, 5))
  total_length = len(planned_data)
  plt.plot(range(1, total_length + 1), planned_data,
'g', label='Planned by PM')
  plt.plot(range(1, total_length + 1), forecast_data,
'r', label='Forecast')
  plt.plot(range(1, len(real_data) + 1), real_data,
'b', label='Real Data from Project')
  plt.legend()
  plt.show()

# Data preparation
S = np.array([[1, 2, 1, 3, 2, 1, 5, 3, 2, 1], [0.1, 0.2,
0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]]).T
R = np.array([1.65, 2.75, 0.85, 3.95, 3.05, 2.15, 6.25,
4.35, 2.65, 1.55])
Planned_S = S.sum(axis=1)

print ("Planned S=", Planned_S)

# Model fitting
S_const = prepare_data(PQ)
model = fit_model(S_const, R)
print(model.summary())

# Forecasting
new_S = np.array([[1, 4, 2], [0.1, 0.5, 0.2]]).T
new_S_const = prepare_data(new_S)
forecasted_R = model.predict(new_S_const)
print(forecasted_R)

# Data for plotting
YY = np.concatenate((R, forecasted_R))
P_S = np.concatenate((Planned_S, new_S.
sum(axis=1)))

# Plot
plot_data(R, P_S, YY, len(new_S))
```

3. Results and Discussion

Then, for the example data above, it is possible to set new planned cost data:

$$P' = \{1,\ 4,\ 2\},$$
$$Q' = \{0.1,\ 0.5,\ 0.2\}, \quad (7)$$
$$S' = \{1.1,\ 4.5,\ 2.2\}.$$

For this data, it is possible to make a forecast for real costs using the Ordinary Least Squares (OLS) method [3]. Test execution result is shown in Fig. 1, 2.

As a result of the calculations, it is possible to get the following values for the predicted data:

$$R' = \{1.49982097,\ 5.17076726,\ 2.71531969\}. \quad (8)$$

```
==============================================================================
Dep. Variable:                y   R-squared:                       0.953
Model:                      OLS   Adj. R-squared:                  0.940
Method:           Least Squares   F-statistic:                     71.55
Date:          Fri, 15 Dec 2023   Prob (F-statistic):           2.19e-05
Time:                  09:27:48   Log-Likelihood:                -2.9573
No. Observations:            10   AIC:                             11.91
Df Residuals:                 7   BIC:                             12.82
Df Model:                     2
Covariance Type:      nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2843      0.313      0.910      0.393      -0.455       1.024
x1             1.1910      0.103     11.553      0.000       0.947       1.435
x2             0.2445      0.438      0.558      0.594      -0.791       1.280
==============================================================================
Omnibus:                    1.479   Durbin-Watson:                   2.030
Prob(Omnibus):              0.477   Jarque-Bera (JB):                0.401
Skew:                      -0.491   Prob(JB):                        0.818
Kurtosis:                   2.982   Cond. No.                         10.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[1.49982097 5.17076726 2.71531969]
```

Fig. 1. The text output of test

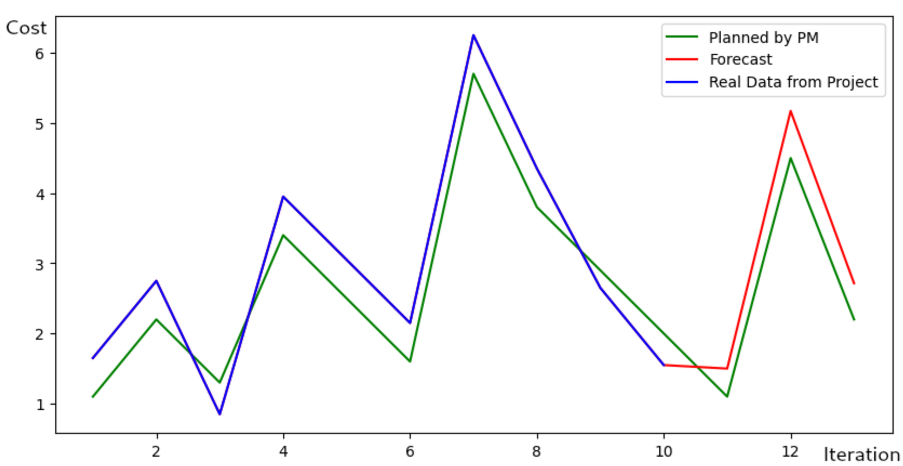

Fig. 2. Graph of planned costs (green color), actual costs at each iteration (blue color) and predicted value (red color)

As it can be seen from the obtained data, it is possible to calculate a forecast graph from the 11th to the 13th iteration (red color) in Fig. 2. It differs slightly from the schedule planned by the project manager (green color) taking into account manual planning errors in previous iterations. The accuracy of the method is sufficient, compared to the deviation of the planned and actual values (blue and green graphs from the 1st to the 10th iterations).

Since actual data on costs for the forecasted section (iterations 11–13) is unavailable, let's propose to consider the difference between the planned data (Planned by PM) and the forecasted values (Forecast) as shown in Fig. 2. The analysis showed that the general pattern of errors indicates a systematic underestimation by the project manager. On average, the forecasted data exceeds the planned estimates by approximately 5–7 %, suggesting to PM the need for adjustments in the cost estimation approaches to achieve greater accuracy.

This study makes a relevant contribution to the understanding of cost forecasting processes within scrum iterations using machine learning techniques. The least-squares analysis demonstrates the model's ability to adequately estimate future costs based on historical data. This highlights the importance of accurate data input and careful selection of model parameters, which is critical to ensuring accurate predictions.

The given example of forecasting opens up space for further research. In particular, attention should be paid to the variability of project conditions, which can affect the effectiveness of forecasting. Such variability requires more flexible machine learning methods that can adapt to changes in data and project conditions [5–7].

The least squares model used in this work showed acceptable prediction accuracy on a data set limited by the number and nature of projects. Such results indicate a potential of the method for use in a variety of settings, however, additional experiments are needed to verify its effectiveness in a wider range of project scenarios.

In addition, it is worth noting that taking into account additional factors, such as changes in team composition, technological updates, as well as external economic conditions, can significantly increase the accuracy of forecasts. Addressing these factors will require the integration of more sophisticated prediction models based on machine learning algorithms that can efficiently process large and diverse data sets.

Among the shortcomings of the method, it is worth noting the impossibility of accurate forecasting for a large number of iterations.

The research results lead to the development of methodological recommendations for practicing project managers who could use forecasting to optimize project management processes. This, in turn, can contribute to a better understanding of the potential costs and resources required to successfully complete projects within planned time frames and budgets.

The practical significance of this research lies in the creation of an effective tool that allows development teams to more accurately plan resources and timelines, which in turn contributes to the overall efficiency of project management within the scrum framework. By refining the estimation processes and reducing the uncertainty in planning, this tool can help project managers optimize resource allocation, minimize project risks, and ensure the timely delivery of software products. The implementation of this forecasting method can ultimately lead to better project outcomes, improved team performance, and higher client satisfaction in software development projects.

Further research may include developing more complex models [7–10] that take into account a larger number of parameters and are able to adapt to changes in project conditions, as well as testing other machine learning methods to provide more accurate predictions.

4. Conclusions

In the course of the conducted research, the possibilities of forecasting the costs for software development in terms of scrum iterations were considered. The Ordinary Least Squares Model (LSM) was used as a basis, which allows estimating future costs for subsequent iterations on the basis of historical data on previous sprints. The study proves the feasibility of using LSM for such purposes, especially taking into account the fact that forecasting is based on a limited number of parameters (resources and time).

The results of using the model showed a sufficient level of correlation between the predicted and real values, which indicates the adequacy of the selected forecasting method for managing projects using the scrum methodology. The constructed graphs demonstrate that the model is able to adequately reflect the dynamics of cost changes and thereby help project teams to plan resources more effectively.

It is important to note that the use of machine learning methods, in particular the method of least squares, requires accuracy in the selection of input data and their processing. It is necessary to ensure high quality of data, as well as take into account possible changes in the conditions of design and implementation of projects, which may affect the forecasting results.

Finally, the proposed approach proved to be effective on test data, which gives reason to recommend it for use in real projects. Such a method can become a valuable tool in the hands of scrum teams aimed at increasing the efficiency of software project management and cost optimization.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, including financial, personal, authorship, or any other, that could affect the study and its results presented in this article.

Financing

The study was conducted without financial support.

Data availability

The manuscript has no associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the presented work.

References

1. Kharchenko, K., Beznosyk, O., Bulakh, B., Ishchenko, G., Yaremenko, V. (2023). The development of the method of optimizing costs for software testing in the Agile model. *Technology Audit and Production Reserves, 6 (2 (74)),* 10–14. https://doi.org/10.15587/2706-5448.2023.293067
2. Sutherland, J., Schwaber, K. (2017). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.* Available at: https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf Last accessed: 05.06.2024
3. *Statistical models, hypothesis tests, and data exploration.* Version 0.14.0. Available at: https://www.statsmodels.org/stable/ Last accessed: 01.06.2024
4. *Statsmodels. Ordinary Least Squares.* Version 0.14.0. Available at: https://www.statsmodels.org/stable/generated/statsmodels.regression.linear_model.OLS.html Last accessed: 03.06.2024
5. Ramessur, M. A., Nagowah, S. D. (2021). A predictive model to estimate effort in a sprint using machine learning techniques. *International Journal of Information Technology, 13 (3),* 1101–1110. https://doi.org/10.1007/s41870-021-00669-z
6. Alshammari, F. H. (2022). Cost estimate in scrum project with the decision-based effort estimation technique. *Soft Computing, 26 (20),* 10993–11005. https://doi.org/10.1007/s00500-022-07352-w
7. Coleman, J. (2022). *Talking about Sizing and Forecasting in Scrum.* Available at: https://www.infoq.com/articles/sizing-forecasting-scrum/ Last accessed: 05.06.2024
8. Mallon, S. (2020). *AI Advances Facilitate SCRUM Teams For Agile Development.* Available at: https://www.smartdatacollective.com/ai-advances-facilitate-scrum-team-construction-for-agile-development/ Last accessed: 07.06.2024
9. Omeyer, A. (2023). *4 ways Scrum Masters can Leverage AI today + tool recommendations.* Available at: https://www.step-size.com/blog/ways-scrum-masters-can-leverage-ai-today/ Last accessed: 10.06.2024
10. *Leveraging the Power of Artificial Intelligence in Agile Development* (2023). Available at: https://www.cognixia.com/blog/leveraging-the-power-of-artificial-intelligence-in-agile-development/ Last accessed: 07.06.2024

Kostyantyn Kharchenko, *PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0002-7334-8038*

✉***Oleksandr Beznosyk,*** *PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0003-2775-6070, e-mail: beznosyk.oleksandr@lll.kpi.ua*

Bogdan Bulakh, *PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-5880-6101*

Bogdan Kyriusha, *PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-7343-1387*

Vadym Yaremenko, *Assistant, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-8557-6938*

✉*Corresponding author*