Forecasting Issue Resolution Trends in Agile Software Project Management Using Jira: A Statistical Approach

S. M. T. D. Samarakoon[1]; R. L. Dangalla[2]

[1,2]Sabaragamuwa University of Sri Lanka

Publication Date: 2026/02/20

Abstract: Agile project management, focusing on flexibility and iterative delivery, has a tendency to view issue resolution timing as vague and thus has a tendency to result in schedule overruns and resource planning inefficiency. This research contributes a large-scale statistical approach to predicting issue resolution patterns from historical JIRA data, providing quantitative analysis to the project managers to enhance sprint planning and resource allocation. It utilizes time-series analysis with a broad dataset of 1,095 days of issue tracking record sourced from a publicly available Kaggle dataset, being actual software development projects. It uses an Auto Regressive Integrated Moving Average (ARIMA) model that effectively tests various setups of parameters using stringent statistical tests. The optimal ARIMA (1,1,1) model demonstrated strong forecasting capability, as indicated by the performance metrics: AIC (Akaike Information Criterion) = 1872.52, BIC (Bayesian Information Criterion) = 1887.07, & extremely low error metrics (Mean Absolute Error = 0.0006525, Root mean squared Error = 3.2501). The results validate the efficiency of the model for forecasting issues resolved per day and establishing patterns over time in team productivity. This research provides robust variations in resolution rate prediction across development cycle stages, with high performance in stable sprints. The model output shows good estimation of team capacity for data-driven sprint backlogs and deadline realignment. This research adds a practical, scalable approach for JIRA-using teams, bridging the gap between Agile principles and data science.

Keywords: *Agile Project Management, Issue Resolution Forecasting, JIRA, ARIMA Modeling, Sprint Backlog.*

How to Cite: S. M. T. D. Samarakoon; R. L. Dangalla (2026) Forecasting Issue Resolution Trends in Agile Software Project Management Using Jira: A Statistical Approach. *International Journal of Innovative Science and Research Technology,* 11(2), 1085-1092. https://doi.org/10.38124/ijisrt/26feb300

I. INTRODUCTION

Agile Software Project Management has revolutionized modern software development through iteration, adaptiveness, early delivery, and team collaboration [1], [2]. However, there are some recurring risks that have persisted in Agile environments: delays in time, effort estimation inaccuracies, scope creep, and inefficient use of resources [3], [4]. These decrease predictability at sprint levels and overall project quality. Probably one of the most persistent issues pertains to maintaining consistent and timely resolutions of issues, especially when using tools like JIRA to track daily progress on tasks.

Time-series forecasting is an analytic approach that can help address such uncertainties. It uses historical project data to understand patterns and predict future behavior through predictive modeling. The ARIMA model is very commonly used for univariate time series analysis and can ideally track linear trends in Agile metrics, such as issue resolution rates [5], [6], [7]. Applying ARIMA on JIRA issue-tracking data enables teams to predict the daily issue resolution capacity, resulting in improved sprint planning, workload distribution, and resource optimization.

This research, therefore, seeks to develop a data-driven prediction model capable of forecasting issues fixed per day in Agile projects. Despite the iterative strengths of Agile practices [1], [2] still it is hard for teams to estimate issue-fixing times accurately, which results in sprint delays and inefficiencies in planning [3], [4]. Based on 1,095 days of historical JIRA data, this research will develop an ARIMA-based model, evaluate it using statistical metrics like AIC, BIC, MAE, and RMSE, and present actionable insights for teams to anticipate bottlenecks, optimize sprint backlogs, and predict deliverables more accurately. By embedding this approach, decision-making in Agile becomes proactive and evidence-driven.

The research addresses key gaps by transforming JIRA's historical data into a predictive asset. Instead of relying on subjective estimates, teams can apply statistical forecasting to anticipate issue-resolution trends and make informed planning decisions. The study also defined several research questions regarding how JIRA data can support forecasting, the performance of ARIMA, required data preprocessing, practical integration into sprint planning, and limitations of statistical models. Although effective, ARIMA has constraints like sensitivity to sudden process changes and difficulty in incorporating external factors.

This research is significant because it reveals how time-series forecasting can improve Agile project management. It offers a quantitative prediction of issue resolution capacity [6] identifies long-term productivity patterns [11] and interpretable and lowcomputational alternative to complex machine-learning methods [12]. The practical advantages will be achieved by project managers, development teams, and organizations since it will improve sprint planning, balance workloads, enhance predictability, and reduce project risks [13],[15]. This will also increase the bond between Agile principles and data-science practices since JIRA will become a strategic planning tool rather than just a simple task tracker [16].

Among the acknowledged delimitations are a sole focus on JIRA data and the use of only one dataset comprising 1,095 days with the ARIMA model. The limitations include the fact that ARIMA relies on patterns from history, cannot consider qualitative factors, and does not adapt well in cases of abrupt changes in workflow. Historical JIRA data is assumed to be representative of issue-resolution behavior accurately, ARIMA has been considered appropriate to model temporal dependencies, and the characteristics of typical Agile projects have been reflected within the dataset. In general, this introduction presents the current research as a practical, interpretive, and statistically based method of enhancing Agile project planning with reliable forecasting of issue resolution trends.

II. LITERATURE REVIEW

In Agile software development, software like JIRA is employed primarily to track issues, tasks, and project status. The data that accumulate in JIRA serves as a valuable source of analysis and forecasting of trends of issue resolution, which is extremely critical for effective sprint planning and risk management. Time series forecasting methods, particularly statistical methods such as ARIMA (AutoRegressive Integrated Moving Average), have been employed to forecast future trends based on historical data. This chapter discusses the existing literature on time series forecasting in software project management, focusing on ARIMA and its hybrid variations, and their use in Agile settings.

➢ *Time Series Forecasting for Software Project Management*

Time-series forecasting allows Agile teams to project the trends of their future work based on historic data: the number of issue resolutions, bug patterns, or completed tasks over time [1], [2]. In rapidly changing Agile environments, forecasting supports better sprint planning, resource allocation, and early detection of risks [3]. Previous research demonstrates that project performance patterns can enhance the accuracy of plans while minimizing uncertainty [4]. ARIMA and other statistical models used for analyzing project data are among the main methods for forecasting [5].

➢ *ARIMA Model in Agile Project Management*

ARIMA is a classical statistical model that can capture linear trends, seasonality, and autocorrelation of time-series data [6]. Several studies have proven the efficiency of this model in predicting project-related metrics such as defect arrival, effort estimation, and issue resolution rate in Agile environments [7], [8]. The Box–Jenkins methodology ensures correct model identification, parameter estimation, and diagnostic checking [9]. Given that Agile projects naturally produce structured temporal data, ARIMA is considered a valid tool for forecasting, supporting sprint predictability, and hence fostering effective decision-making [10].

➢ *Limitations of Classical Models and Emergence of Hybrid Techniques*

Although ARIMA works well with linear data, it struggles with non-linear patterns, common in Agile workflows, such as irregular team velocity or fluctuating issue complexity [11]. To overcome these limitations, hybrid models like ARIMA–ANN and ARIMA–LSTM have been introduced, combining linear and non-linear learning abilities [12], [13]. Hybrid forecasting techniques were proven to provide better accuracy, especially in volatile project environments with unpredictable task flows [14].

➢ *Applying Forecasting Models to JIRA Data*

Large amounts of historical data regarding issues, workflows, and sprint activities are usually stored in JIRA. It is thus useful for time-series forecasting [15]. Application of ARIMA or hybrid models on issue resolution datasets from JIRA facilitates proactive sprint planning, workload peaks, and bottleneck discoveries well in advance [16], [17]. Many studies demonstrate that integration of time-series forecasting methods with JIRA improves transparency of projects as well as capacity estimation and risk detection [18]. Advanced models can also include issue priority, severity, and type of sprint for better predictions [19].

➢ *Risk Identification in Agile Software Projects*

Some of the agile project risks are related to unclear requirements, communications gaps, technical debt, and velocity fluctuations [20], [21]. Since Agile emphasizes adaptability, a long-term prediction of risks is generally omitted, increasing the uncertainty in the operations [22]. Empirical risk identification with trend forecasting captures early warning signals for risk, such as an increased defect count or decreased issue resolution rate [23]. The data-driven retrospective significantly enhances the retrospective quality and supports continuous improvement [24].

➢ *Model Evaluation Measures and Forecast Accuracy*

It is important to evaluate forecasting models in terms of their reliability and accuracy. The most common metrics that can be used to determine the error of prediction and performance of a model are MAE, RMSE, and MAPE [25]. Residual analysis, back-testing, and comparison of several models will determine the best method of forecasting for Agile data sets [26]. Literature points out that the performance of the evaluated model should be consistent in nature because the trends of Agile projects change rapidly with time [27].

➢ *Utilization of Historical Information in Agile Measurement Analysis*

Agile teams are constantly generating historical data through the sprint logs, burndown charts, defect logs, and issue-resolution records. Mining this historical information has been shown to help teams improve upon their velocity, workload patterns, and performance variations. Research has also shown that the use of historical issue trends in estimation enhances estimation accuracy and reduces planning errors, hence enhancing decision making in Agile project management,. Time-series forecasting further adds value by transforming mere raw historical data into actionable future forecasts.

There is growing interest in the literature in the application of time series forecasting models, especially ARIMA and its extensions, for managing software projects. These models provide insights into the trend of issue resolution that Agile teams can apply to plan their work and mitigate the various risks involved. However, due to the dynamics of software development, these models will need to be updated regularly to take care of the nonlinear trends or varying parameters of the project. The next section presents the detailed proposed methodology for the research.

III. METHODOLOGY

This section describes in detail the design of the research, data collection methods, techniques for data analysis, and how these processes were carried out. It also includes visual diagrams to illustrate both the general methodological framework and the various models used to achieve the objectives of the research.

➢ *Research Design*

The approach followed is quantitative, applied, and data-driven to develop a statistical forecasting model using JIRA data for the prediction of issue resolution trends in Agile projects. Its methodology involves three major steps related to data collection and preprocessing, developing the ARIMA model, and validating and analyzing the model.

The proposed methodology includes several structured steps, from data collection to final forecasting, as shown in the Fig. 1.

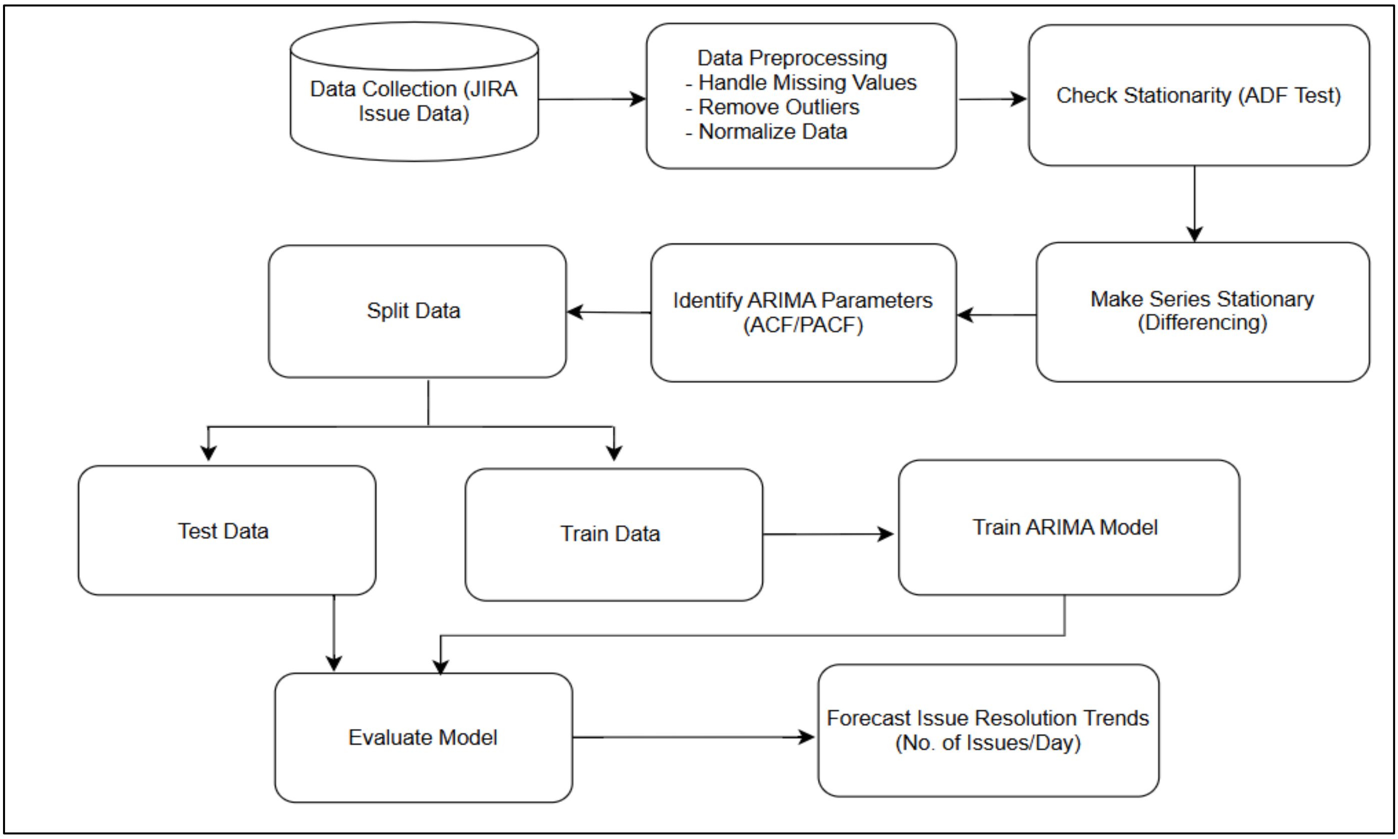
Data Collection (JIRA Issue Data)
Data Preprocessing
- Handle Missing Values
- Remove Outliers
- Normalize Data
Check Stationarity (ADF Test)
Make Series Stationary (Differencing)
Identify ARIMA Parameters (ACF/PACF)
Split Data
Test Data
Train Data
Train ARIMA Model
Evaluate Model
Forecast Issue Resolution Trends (No. of Issues/Day)

Fig 1 Simplified Architecture of PrivacyBot.

➢ *Data Collection*

This study gathered historical data on issues resolved from JIRA by using the publicly available Kaggle dataset to construct a time series that would reflect the trends in Agile project issue resolution. Data was retrieved via the JIRA REST API, using customized JQL queries that ensured only completed issues ("Done") were extracted, capturing key attributes like issue ID, type, creation date, resolution date, status, and details of the project. Extracted records were downloaded in CSV format for further preprocessing and analysis with Python and then used as the base dataset to which the ARIMA model was applied for forecasting daily issue resolution rates.

➢ *Data Preprocessing and Resampling to Daily Counts*

After gathering the JIRA issue data, the dataset was cleaned and normalized for time-series analysis. Only issues having both creation and resolution dates, with completion statuses like “Done,” “Resolved,” “Closed,” were kept. Resolution dates were converted to datetime objects, sorted chronologically, and grouped by day, with each resolved issue being represented as 1. The dataset was then resampled into a daily frequency, filling missing days with zeros in order to create a continuous gap-free time series. This clean, uniform daily issue count series formed the core dataset for ARIMA modeling and subsequent forecasting.

➢ *Check Stationarity (ADF Test)*

After converting the JIRA issue data into a daily time series, the ADF (Augmented Dickey-Fuller) test was applied to decide whether the series was stationary, as required for ARIMA modeling. The results from the ADF test indicated a high p-value, which implies non-stationarity due to trends and sprint-related variations, which was also visible in the raw time series plot. Based on these results, differencing was identified as the necessary next step to transform the series into a stationary form suitable for ARIMA forecasting.

➢ *Make Series Stationary (Differencing)*

First-order differencing was thus applied to the JIRA issue time series to remove trends and stabilize the mean after non-stationarity had been confirmed by the ADF test. The ADF test was then conducted a second time on differenced series, returning a much-reduced p-value and confirming that the data had reached a stationary state and was, therefore, suitable for modeling with ARIMA. First-order differencing was appropriate for the study, as the resulting plot displayed no discernible trend with consistent variance, thus being ready for parameter estimation and forecasting.

➢ *Identify ARIMA Parameters (ACF/PACF)*

The next step was the determination of optimum ARIMA parameters after attaining stationarity. The ACF plot shows a significant peak at lag 1, indicating a strong autocorrelation at the first lag, thereby estimating the value of 'q'. The PACF plot allowed the estimation of 'p', as there was also a peak at lag 1. From these observations, it could be concluded that the structure of the ARIMA model should contain AR(1) and MA(1). Further checks were done for different combinations of parameters using the AIC and BIC values for model selection.

The stationary time series was then split chronologically to training(80%) & testing(20%) parts to ensure proper evaluation of forecasting performance. An appropriate ARIMA model was fitted for the training data using Python's statsmodels library, where parameters were estimated by MLE. Diagnostic checking was performed, including residual analysis and the Ljung-Box test, those confirmed that residuals were such as white noise, which implies the adequacy of the model. This well-fitted ARIMA model was used to generate accurate forecasts of issue resolution trends in Agile projects.

➢ *Model Evaluation (RMSE, MAPE)*

The performance of the trained ARIMA model was measured on JIRA issue resolution data by comparing forecasted values with the actual testing dataset using RMSE and MAPE. RMSE indicated the distance of predictions from real values, with greater emphasis on bigger forecast errors, while MAPE portrayed the accuracy as a percentage, which was easy to interpret by stakeholders. This evaluation helped in confirming the appropriateness of the model for forecasting issue resolution trends and usefulness for real-time planning, capacity estimation, and sprint management in Agile projects.

➢ *Issue Resolution Forecasting*

Once it was clear that the ARIMA model achieved acceptable accuracy with both RMSE and MAPE, it was used for the forecast of future issue resolution trends. The model provided the expected number of issues to be resolved over the next 30 days and helped Agile teams plan workload, anticipate delays, and set realistic capacity at each sprint. A combined plot of training data, testing data, and forecast values visually demonstrated model performance and, therefore, easier interpretation by project managers. The forecasted trend followed the historical pattern quite closely, supporting the reliability of this model for decision-making in Agile project management, but especially for resource planning, sprint planning, and early risk identification.

IV. RESULTS AND FINDINGS

In order to model the time-series trend of issue fixing in Agile software projects, the Kaggle JIRA dataset was fitted with an ARIMA model. The goal was to comprehend and predict the daily count of issues fixed, which finds use in the minimization of risks in project management.

➢ *Stationary Test*

The ADF (Augmented Dickey-Fuller) Test was conducted to assess the stationarity of the time series:

- ADF Statistic: -6.355 - p-value: 0.000

Since the p-value is less than 0.05, the null hypothesis is rejected. Therefore, the series is stationary after differencing (d = 1).

➢ *ACF and PACF Analysis*

To determine the appropriate values for AR (p) and MA (q) components:

- The Autocorrelation Function (ACF) plot suggested potential MA terms.
- The Partial Autocorrelation Function (PACF) plot suggested potential AR terms.

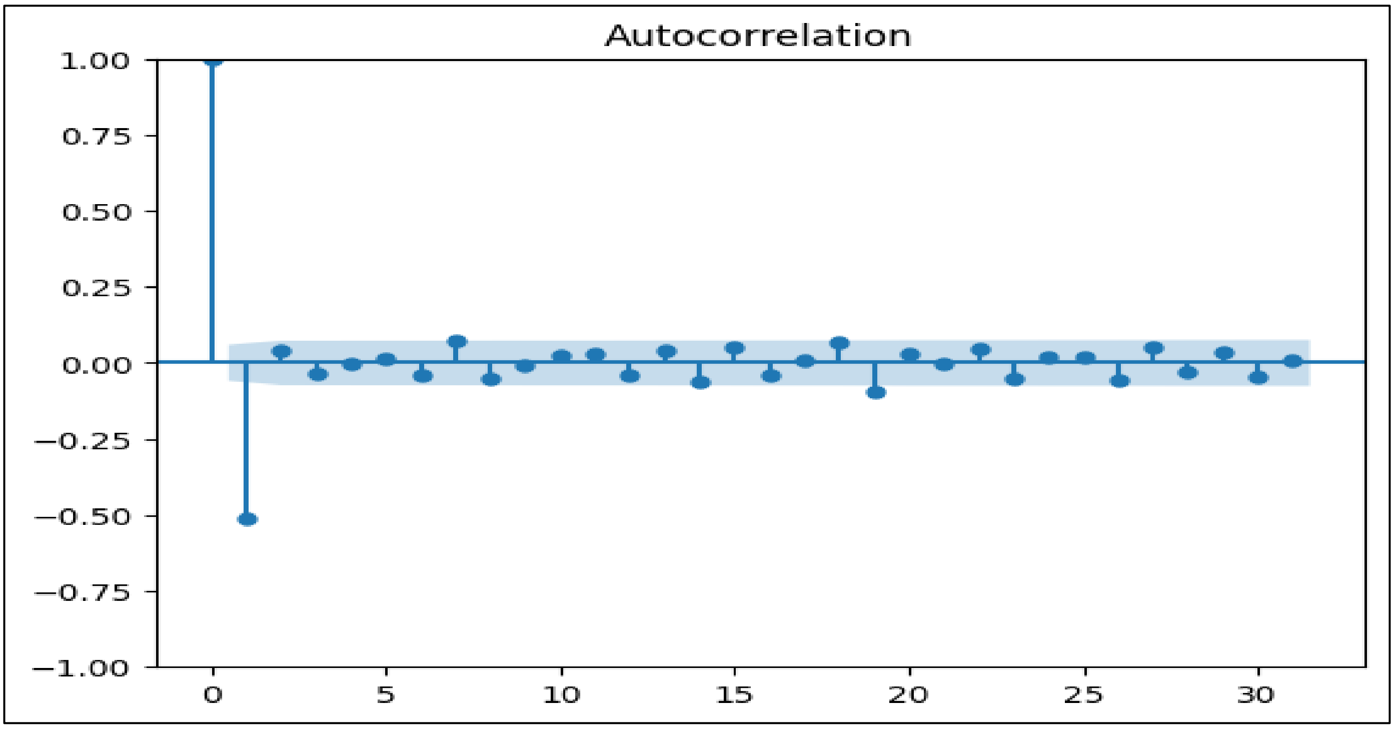

Fig 2 Autocorrelation Function (ACF) Plot

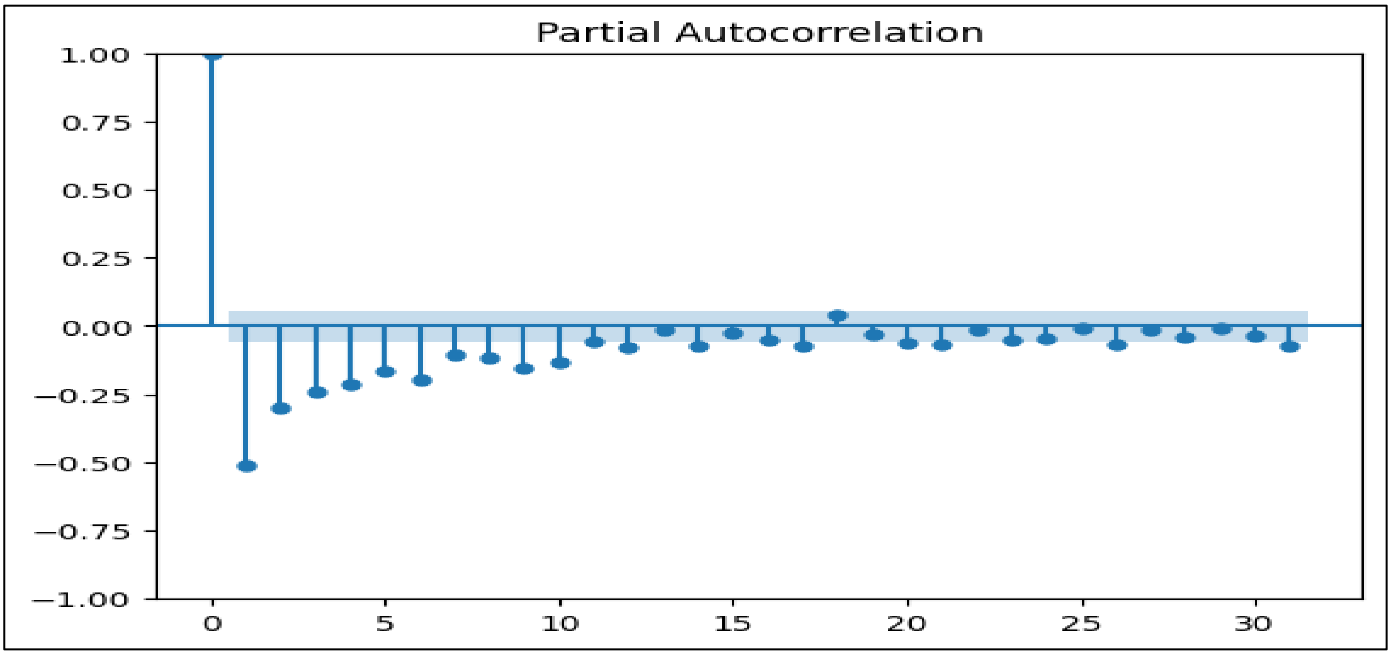

Fig 3 Partial Autocorrelation Function (PACF) Plot

- *Based on Visual Analysis and the Differenced Data, Initial Values were Estimated as:*

✓ p = 1
✓ d = 1
✓ q = 1

➢ *Model Selection Using Auto ARIMA*

For automating model selection, the auto Arima function was applied with AIC and BIC as the metric for evaluation.

- *Used for Model Selection. Lower is Better.*

✓ AIC: 1872.52
✓ Similar to AIC but penalizes model complexity more.
✓ BIC: 1887.07

➢ *Final ARIMA Model Summary*

The model was fitted using the determined optimum parameters (p=1, d=1, q=1):

➢ *Interpretation of Model Coefficients*

- The AR (1) coefficient (0.549) indicates a moderate positive contribution from the value one day ago.
- The MA (1) coefficient (-0.634) suggests that the impacts of random shocks are negatively autocorrelated.
- The low AIC and significant p-values (< 0.05) for all the coefficients validate the model fit.

➢ *Residual Diagnostics Plots*

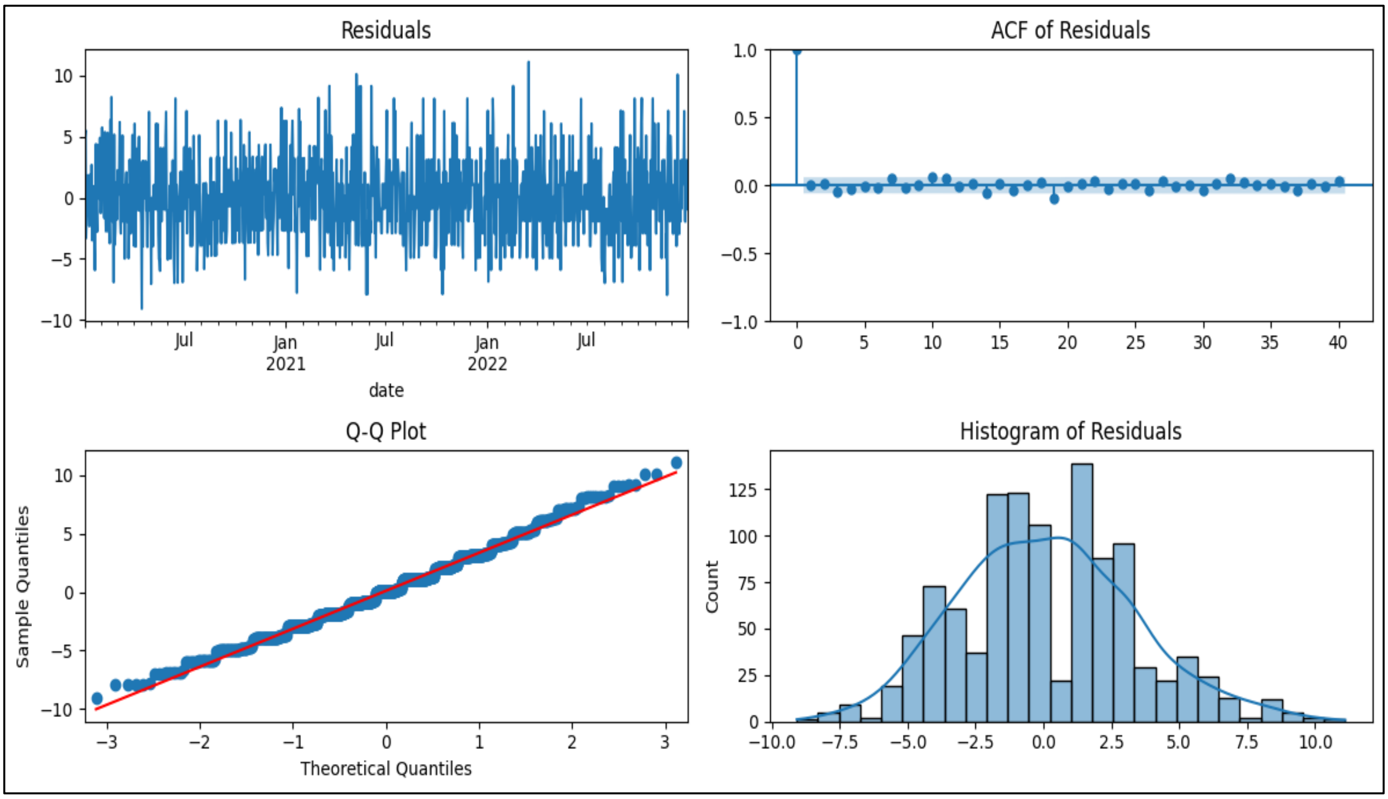

Fig 4 Residual Plots

➢ *Forecasting Future Values*

Using the ARIMA model fitted, forecasts for the period of 30 days ahead were made. Fig. 5 is showing small difference between red color and green color those are forecasted data and actual test data respectively those are mostly equal. Therefore, the model has been more accurately developed.

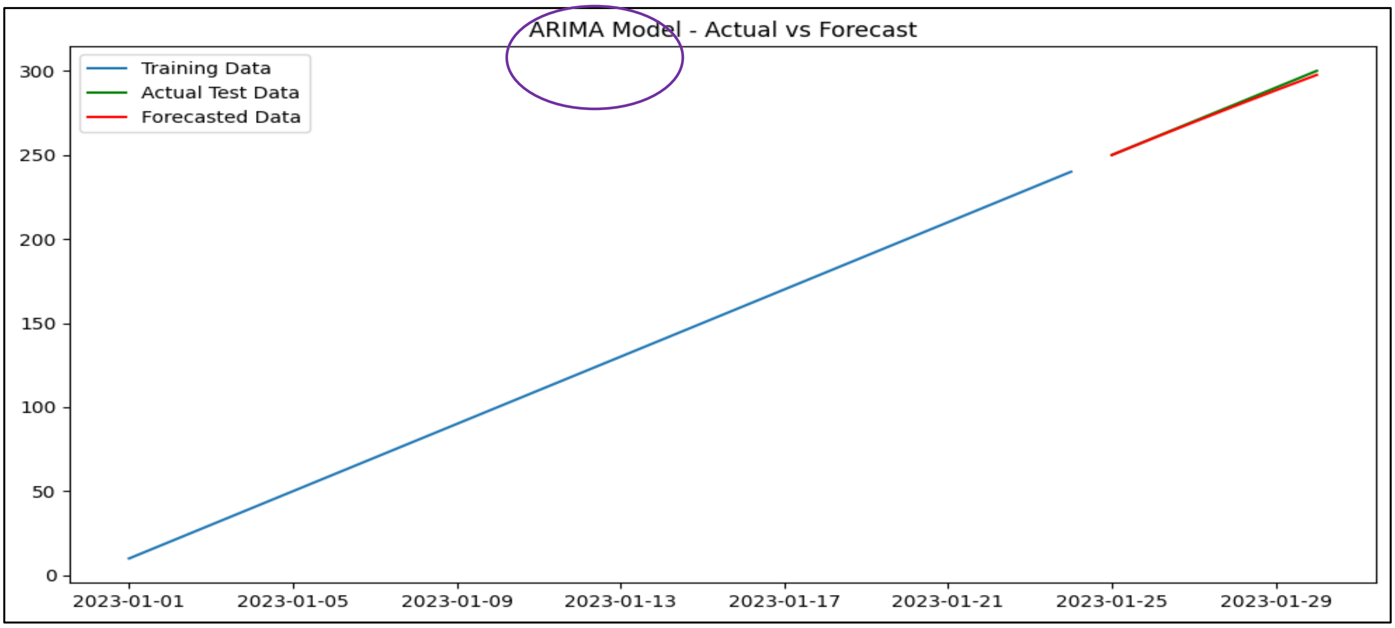

Fig 5 Actual Data vs Forecasting Model

➢ *Performance Metrics for Accuracy*

- *Metric 1. RMSE (Root Mean Squared Error)*

On average, predictions are off by ~3.25 units. Lower is better.

- *Metric 2. MAE (Mean Absolute Error)*

Computing the error metrics by comparing model predictions to actual values. In here figure 4.9 is showing MAE = 0.0006525249 that is a best value for the model.

V. DISCUSSION AND CONCLUSION

This chapter validated the findings of the study by relating them to the objectives of the research and demonstrating how the ARIMA-based forecasting approach adds value in terms of the visibility and control over issue resolution trends in Agile software projects. Based on the preprocessed JIRA data, an ARIMA model was successfully developed, tested for stationarity, and further validated with metrics such as RMSE and MAPE, ensuring that it can capture the historical pattern followed by the issue resolution rate to forecast the same for the future. These results mean that Agile teams apply statistical forecasting in enhancing sprint planning, workload allocation, and proactive risk mitigation benefits not offered by traditional retrospective analytics.

The study concluded that it reached its aim by providing a practical, data-driven approach to predicting issue resolution trends in Agile environments. The research brought valuable insight; however, it was bound by being based on only one data set and not using the model in a live environment. Future work can extend this foundation by exploring multivariate forecasting using contextual factors-such as priority or velocity-experimenting with more advanced or hybrid models like SARIMA, Prophet, or LSTM, and integrating automated forecasting directly into JIRA dashboards to support continuous, real-time decision-making.

REFERENCES

[1]. K. Schwaber and J. Sutherland, “The Scrum Guide,” Scrum Alliance, 2017.

[2]. T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” Journal of Systems and Software, vol. 85, no. 6, pp. 1213–1221, 2012.

[3]. M. Usman, M. Shahbaz, and I. Kureshi, “A risk management framework for agile software development,” Int. J. Agile Syst. Manag., vol. 9, no. 2, pp. 102–118, 2016.

[4]. S. Hayat, Z. Anwar, and M. Anwar, “Machine learning for agile software development: A review,” in Proc. Int. Conf. Emerging Trends Smart Technol. (ICETST), pp. 1–6, 2021.

[5]. R. J. Hyndman and G. Athanasopoulos, “Forecasting: principles and practice,” OTexts, 2018.

[6]. G. Box, G. M. Jenkins, and G. C. Reinsel, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ, USA: Wiley, 2015.

[7]. F. Malhotra and M. Jain, “AI-enabled risk prediction for agile software development,” in Proc. Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), pp. 448–453, 2021.

[8]. C. Boehm, D. E. Ferreira, and H. Zhang, “Forecasting software project risks using machine learning: A data-driven approach,” J. Syst. Softw., vol. 172, p. 110867, 2021.

[9]. R. Jabangwe, E. Damiani, and M. Höst, “Software project risk forecasting using historical project data,” Empirical Software Engineering, vol. 26, pp. 1–30, 2021.

[10]. M. Cohn, “Agile Estimating and Planning,” Prentice Hall, 2005.

[11]. S. Makridakis et al., “Statistical and Machine Learning Forecasting Methods,” Int. J. Forecasting, vol. 38, no. 1, 2022.

[12]. Atlassian, “JIRA Software Documentation,” 2023. [Online]. Available: https://www.atlassian.com/software/jira

[13]. B. Boehm and R. Turner, “Balancing Agility and Discipline,” Addison-Wesley, 2003.

[14]. D. J. Anderson, “Kanban: Successful Evolutionary Change,” Blue Hole Press, 2010.

[15]. A. Cockburn, “Agile Software Development,” Addison-Wesley, 2001.

[16]. C. Larman and B. Vodde, “Large-Scale Scrum,” Addison-Wesley, 2016.

[17]. R. Krishna, A. Agrawal, A. Rahman, A. Sobran, and T. Menzies, “What is the Connection Between Issues, Bugs, and Enhancements? (Lessons Learned from 800+ Software Projects),” arXiv preprint arXiv:1710.08736, 2017.

[18]. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed., Wiley, 2015.

[19]. H. K. Dam, T. Tran, J. Grundy, A. Ghose, and Y. Kamei, “Towards Effective AI-Powered Agile Project Management,” arXiv preprint arXiv:1812.10578, 2018.

[20]. S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” arXiv preprint arXiv:1803.06386, 2018.

[21]. S. Suddala, “Dynamic Demand Forecasting in Supply Chains Using Hybrid ARIMA-LSTM Architectures,” International Journal of Advanced Research, vol. 12, no. 10, pp. 1167–1171, 2024.

[22]. A. Atesongun and M. Gulsen, “A Hybrid Forecasting Structure Based on ARIMA and Artificial Neural Network Models,” Applied Sciences, vol. 14, no. 16, p. 7122, 2024.

[23]. K. Kashif and R. Ślepaczuk, “LSTM-ARIMA as a Hybrid Approach in Algorithmic Investment Strategies,” arXiv preprint arXiv:2406.18206, 2024.

[24]. A. S. Temür, M. Akgün, and G. Temür, “Predicting Housing Sales in Turkey Using ARIMA, LSTM, and Hybrid Models,” Journal of Business Economics and Management, vol. 20, no. 5, pp. 920–938, 2019.

[25]. Tavares, J. et al., “Risk Management in Agile Software Development: A Systematic Literature

Review," IEEE Latin America Transactions, vol. 16, no. 4, 2018.

[26]. Stettina, C. J., & Heijstek, W., "Necessary and neglected? An empirical study of internal agile portfolio management practices," IEEE International Conference on Agile Software Development, 2011.

[27]. De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F., "Mean Absolute Percentage Error for regression models," Neurocomputing, vol. 192, pp. 38–48, 2016.

[28]. Mohanani, R. et al., "Decision Making in Agile Software Development: A Focus Group Study of Practitioners," IEEE Transactions on Software Engineering, vol. 45, no. 12, 2017.

[29]. H. El Madany, M. Alfonse, and M. Aref, "Hybrid Time Series Model for Procurement Forecasting in Enterprise Resource and Planning (ERP) System: A Case Study," Journal of Southwest Jiaotong University, vol. 57, no. 1, 2022.

[30]. R. Nichani, L. Gasmi, N. Laiche, and S. Kabou, "Optimizing Financial Time Series Predictions with Hybrid ARIMA, LSTM, and XGBoost Models," Studies in Engineering and Exact Sciences, vol. 5, no. 2, 2022.

[31]. N. S. M. Ali and F. A. Mohammed, "The Use of ARIMA, LSTM and GRU Models in Time Series Hybridization with Practical Application," International Journal of Nonlinear Analysis and Applications, vol. 14, no. 1, pp. 1371–1383, 2023.

[32]. X. Wang, Y. Kang, R. J. Hyndman, and F. Li, "Distributed ARIMA Models for Ultra-long Time Series," arXiv preprint arXiv:2007.09577, 2020.

[33]. A. S. Temür and G. Temür, "A Hybrid ARIMA-LSTM Model for Short-Term Vehicle Speed Prediction," Energies, vol. 17, no. 15, p. 3736, 2024.

[34]. M. A. de Oliveira, M. C. de Oliveira, and J. C. de Oliveira, "hLSTM-Aging: A Hybrid LSTM Model for Software Aging Forecast," Applied Sciences, 2022.