2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)
Karnataka, India. Dec 23-25, 2022

2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) | 978-1-6654-5499-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/SMARTGENCON56628.2022.10083722

Efficient Digital Assistant Workflow for Ticket Monitoring and Dispatching

Mohamed Naveen
Developer, S/4 CDX CD S4 Cloud Operations
SAP Labs India Pvt. Ltd.
Bangalore, Karnataka, India
mohamed.naveen@sap.com

Suman De
Product Manager, S4 Product Insights & Experience
SAP Labs India Pvt. Ltd.
Bangalore, Karnataka, India
suman.de@sap.com

***Abstract*— In a larger cloud operation team, there are groups working on Monitoring and Dispatching whose primary work is monitoring the ticket queue, understanding the ticket, and assigning it to a respective workgroup. In a shift, at least one colleague must be working in the Monitoring and Dispatching role, and in most scenarios, it's an if-else work. In S/4HANA, we hire people specifically for the Monitoring and Dispatching role, contributing to the total cost of ownership. It's not quality work for the people working on the Monitoring and Dispatching role. To solve this problem, we can develop a solution that can understand the sentiment of the tickets, know about the workgroups in a team, and assign them automatically. If we can achieve this, we can reduce the total cost of ownership, provide quality work to our colleagues, avoid manual misses, and have intelligent and optimized dispatching. The vision is to build a Digital assistant with continuous ticket monitoring, intelligent ticket dispatching based on the user workload, and understanding the context of the ticket to give more insights about the ticket like any similar issues solved, what the solution provided last time, and who solved the similar issue last time.**

Keywords*— *Cloud Operations, Monitoring, and Dispatching, tickets, Digital Assistant

I. INTRODUCTION

The generic need for operations-related activities will always be an individual or a unit working on monitoring and dispatching issues for specific areas in an engineering team. In large organizations with users from various sectors ranging from cloud to on-premise solutions, operations play a vital role in streamlining the certainty of making applications available. A minimum of one individual must continue to work in the tracking and allocating role during a shift. In the majority of situations, monitoring and dispatching actions are condition-based, which means that their main responsibility is to keep an eye on the ticket queue. Once a ticket is received, the tracking and allocating group member will review its content, attempt to understand its scope, and then deploy it to the appropriate person working on the relevant area. We recruit employees, particularly for the Monitoring and Dispatching job in S/4HANA, which raises the Total Cost of Ownership (TCO) and degrades the quality of work for those employed in the position. The Monitoring and Dispatching layer is being overspent, the personnel in this function are not happy with their day-to-day responsibilities, and there is a manual error in the ticket service-level agreement.

According to the planned study, a digital assistant will be created with continuous ticket monitoring, intelligent ticket dispatching based on user workload, continuing to keep an eye on the terms of the ticket service level agreement, and alerting the team as necessary when any of the terms are broken. comprehends the context of the ticket to offer more information regarding the ticket, such as any related issues resolved, the solution offered before, and the person who handled the related issue previously. Detailed live reporting on the operations ticketing component should be sent to management. As a result, the operation team may remove the Monitoring and Dispatching Layer with the help of the users, who can then concentrate on interesting projects rather than Monitoring and Dispatching, and their happiness with routine jobs increases.

This paper proposes an idea that has been in the concept phase with pilot implementation proposals. This paper covers the proposed solution to the mentioned problem with a subsequent section taking a systematic approach to explain the entire work done by the author. The paper covers similar work done in this field highlighting the growth of research in this area followed by the proposed workflow, use cases solved by the proposed idea, and the advantages associated with the usage of this research and solving this problem for major industry vendors of enterprise software like SAP.

II. LITERATURE SURVEY

Ticket monitoring and usage of Digital Assistants have an important significance in research areas of Machine Learning and Artificial Intelligence. In this section, we explore scientific works performed in these fields for different use cases. For the purpose of controlling resources available and delivery deadlines, it's critical to determine how long a defect will take to repair. Therefore, anticipating bug-fix timelines with extreme precision is seen as being challenging. According to our opinion, these problems result from the absence of continuous or posterior assessment depending on the activity of existing stakeholders once an issue is originally claimed. Throughout this article, they transform the issue of predicting the duration of bug fixes into a continuous update of estimations with new actions. In order to update forecasts throughout the duration, they record information about issue-related actions that are transmitted to an issue monitoring system. They put forth DASENet, a two-staged activities streaming encoding model that utilizes deep learning, which uses a merged network for extracting contextual characteristics from varying sorts of records, and a series architecture for examining the temporal relationships between the streams. We demonstrate that DASENet maintains steady efficiency by using trials using bug tracking system datasets from open source projects such as Firefox, Chromium, and Eclipse. For example, regarding the Firefox

database, top-1 accuracy for DASENet is 4.6 to 8.5% over previous state-of-the-art studies. The DASENet model, tested with a short collection of roughly 900 observations (2% of an original dataset), may demonstrate competitive performance to the other models with a complete dataset thanks to the transferrable design that our technique offers. This work was one of the initial approaches for combining both. [1]

Every development initiative must include support for programming errors. The assignment of bug reports is crucial for effectively resolving issues. It might be challenging to delegate defect elimination responsibilities to the right programmer in the case of open-source software developments and big projects in which several programmers are working on various elements of computer programming. This procedure of triaging identified issues becomes more challenging when there are more computer programmers and identified issues. Within those circumstances, defect diagnosing could be laborious and prolong the defect-throwing time (BTL). Since the human assignment of bug reports is tedious, expensive, and time-consuming, a computerized method for triaging regression testing may minimize BTL. BTL and user happiness will suffer if defect tracking is given to an irrelevant developer with the requisite qualifications to address the problem. The procedure of automated defect tracking has a lot of opportunities to benefit from message categorization approaches. To identify the suitable programmer or programmers to address the identified defects, several Information Retrieval and Machine Learning techniques are employed in this study. The triaging of bugs was automated within that research using deep learning methods like the Bidirectional Long Short-Term Memory System. Data or content linked to the bug information may be found within event logs. The pretrained GloVe model is used in this study to convert the data sets from logs into vectors. This mechanism keeps track of programmers' actions depending on their prior employment. Several sizable datasets—Net-Beans, Eclipse, and Mozilla—are utilized for evaluating the suggested methodology. It was shown that the suggested method outperformed more established Machine Learning methods overall defect complaint suggestion in terms of accuracy, recall, precision, and f-measure. [2]

Software testing is a term used to describe the process of finding and identifying problems in programs. The majority of application problems result from human errors and omissions within the architecture or the runtime environment of a program. Because defects may take many different types since utilizing computer programs, it can be challenging for an individual/user of an application to identify and handle them effectively. To manage and monitor the identified defects in software, defect monitoring solutions were created as a solution to this problem. Numerous proprietary and open-source bug tracking systems are available presently, and several current concepts have also been developed to address the wide range of issues that continue to appear as a result of the evolving needs of programs. Given the numerous kinds of computer programs that are always changing and advancing on a daily basis, there is a need for a tool that will assist in correcting and keeping track of defect repairs. Throughout this research, a comparison of five defect monitoring devices mix of open source and proprietary systems—has been carried out. A brand-new fault monitoring system called "Bugtrac" is being presented, which takes into account all different kinds of software applications and the potential flaws that may arise from them. If contrasted with the current ones, the suggested approach is anticipated to become a more promising defect monitoring system. [3]

Integrating embodied virtual agents into the routine work of the computer regression assessment is the topic of this visionary study. The capacity of autonomous simulated entities to gather knowledge from their surroundings in addition to readily accessible databases and information services is a crucial characteristic. The construction of an intelligent virtual assistant for the application regression screening raises a number of study obstacles and problems. [5] In many sectors, fresh emerging patterns are appearing as business-specific plans move more and more toward digitization. The safeguarding of the company's digital value chains against these dangers is necessary for the achievement of the business objectives. This defense can be efficiently strengthened by a security operations center (SOC). Depending upon inventory, policy, and the expertise of the senior leadership of the corporation, AI can provide SOC Architecture Models. Hence, businesses using virtual assistants may be capable of achieving their corporate targets in a manner that is simultaneously highly productive and beneficial from a financial standpoint. [6]

III. PROPOSED METHODOLOGY

The following assumptions are made for an explanation of the idea and proposed solution in this document. The vision is to build a Digital assistant with the below features:

- Continues ticket monitoring.
- Intelligent ticket dispatching based on the user workload.
- Understand the context of the ticket to give more insights about the ticket like any similar issues solved, what the solution provided last time and who solved the similar issue last time.
- Continues monitoring of ticket service-level agreement parameters and notifies the team accordingly when any of the ticket parameter breaches.
- Provide detailed live report to the management on the Operations ticketing component.

The above solution can help the operation team to remove the Monitoring and Dispatching Layer and allow the user to work on interesting topics rather than Monitoring and Dispatching and improve user satisfaction in day-to-day work. By removing the Monitoring and Dispatching layer team can reduce the total cost of ownership. The following components make up the proposed system as shown in Figure 1.

A. Ticket Handler

Users can configure any external ticketing tool for monitoring the queue. The ticket handler will look for open tickets on a regular interval and once we receive a new ticket, the header and body of the ticket are sent to Service Ticket Intelligence service to get ticket classification and recommended solutions, and processor suggestions. The ticket handler also updates the Ticket Policy handler to start monitoring the new incoming tickets on service-level agreement parameters.

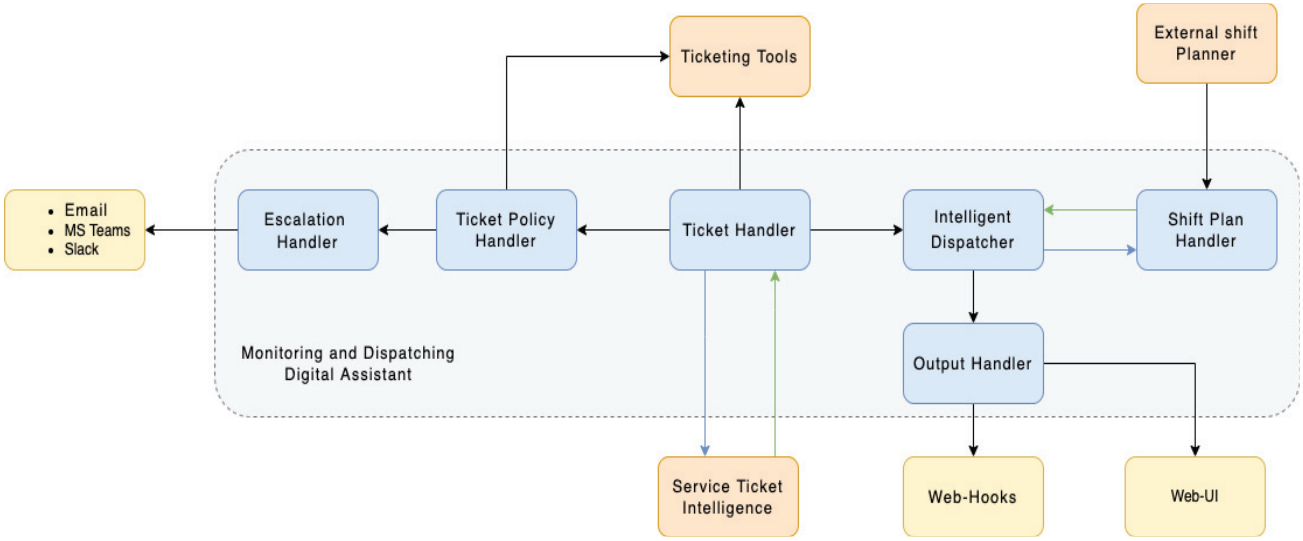

Fig. 1. Illustration of the proposed architecture with each of its sub-components

B. *Ticket Policy Handler*

Users can configure the ticket policies based on the service-level agreement. Once the Ticket policy handler receives the ticket, it starts monitoring the tickets IRT - Initial Reaction Time, MPT – Maximum Processing Time, APT – Action Plan Time, and ORT – Ongoing Response Time based on the configuration. If any of the mentioned parameters are about to breach, it initiates the process of contacting the respective agent and initiating an escalation process.

C. *Escalation Handler*

Users can configure the escalation hierarchy. Whenever there is an SLA breach or about to breach based on the escalation hierarchy, it will try to contact the respective agent via the configured media.

D. *Intelligent Dispatcher*

Users can configure dispatching rules based on the team's requirements, like how many tickets can be handled by an agent at a time and the combination of priority tickets to be dispatched. For Example:

- Agent can only handle a maximum of 5 tickets at a time.
- Agent can only handle 1 maximum priority ticket, 2 Medium priority, and 2 Low priority tickets at a time.

The dispatcher will route open tickets to the agent based on the configuration and also consider the agent recommendation from Service Ticket Intelligence, check the available agent for the particular shift from the Shift plan handler, and send a routing request to the output handler.

E. *Shift Plan Handler*

Users can plan the agent's shift in the Monitoring and Dispatching Digital Assistant through a web interface or Users can integrate their existing shift planning application.

F. *Output Handler*

Users can configure webhooks to receive ticket information / Dispatch alerts or use the inbuild web interface.

G. *Service Ticket Intelligence*

Service Ticket Intelligence helps you to build a self-driven customer service powered by machine learning. The service classifies incoming tickets automatically as a basis for routing them to the right agent. The agent is then provided with recommended solutions to improve operational efficiency. Service Ticket Intelligence is part of the SAP AI Business Services portfolio.

IV. Implementation Scenarios

The proposed workflow described in the previous section covers specific scenarios as follows:

A. *Dispatching of an open ticket to an agent*

- Scheduler triggers sync on a regular interval to the external ticketing tool.
- In a sync call, if we receive any open tickets, the ticket will be monitored in the ticket policy handler to identify an SLA breach and gives a heads-up before the breach.
- The ticket title and subject are sent to STI to get ticket category, agent suggestion, and ticket insight.
- Along with the STI response and ticket information is forwarded to the Intelligent dispatcher.
- Intelligent dispatcher checks the dispatching rules and checks the available agent in the shift forward to the output handler.
- The current queue status and dispatching information will be displayed in Web-UI, and if the webhook is configured, the dispatching information is pushed to the configured webhook.

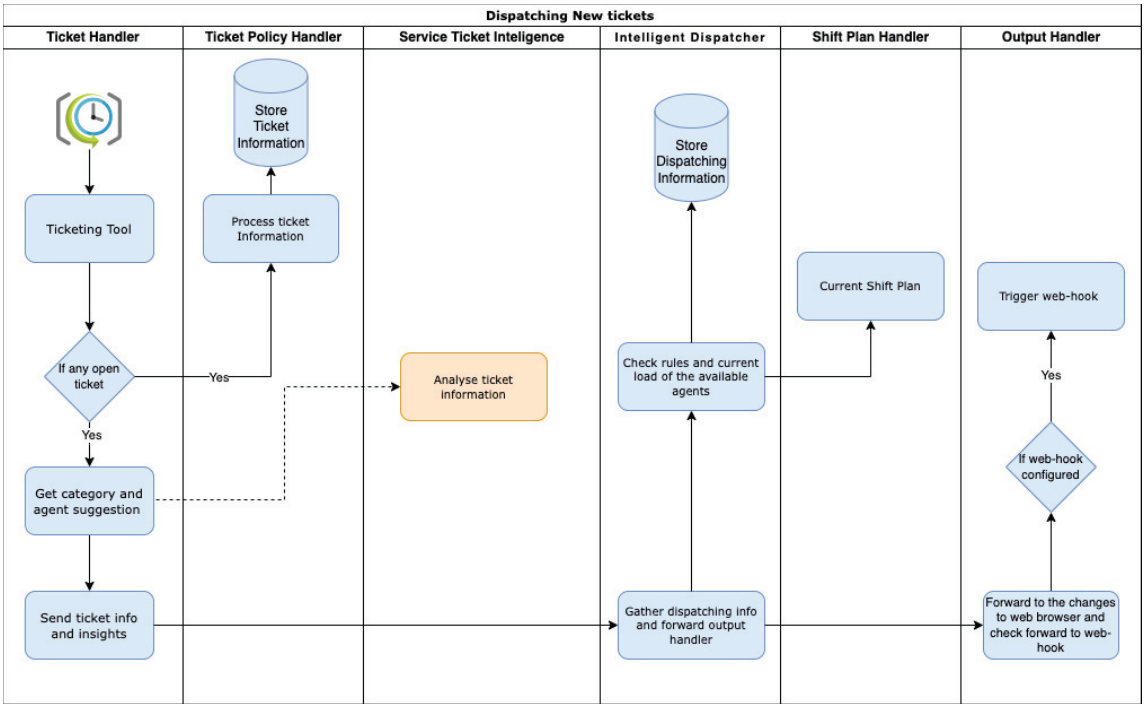

Fig. 2. Illustration of the workflow highlighting how an open ticket is dispatched to an agent

B. Notification to an agent before a possible breach

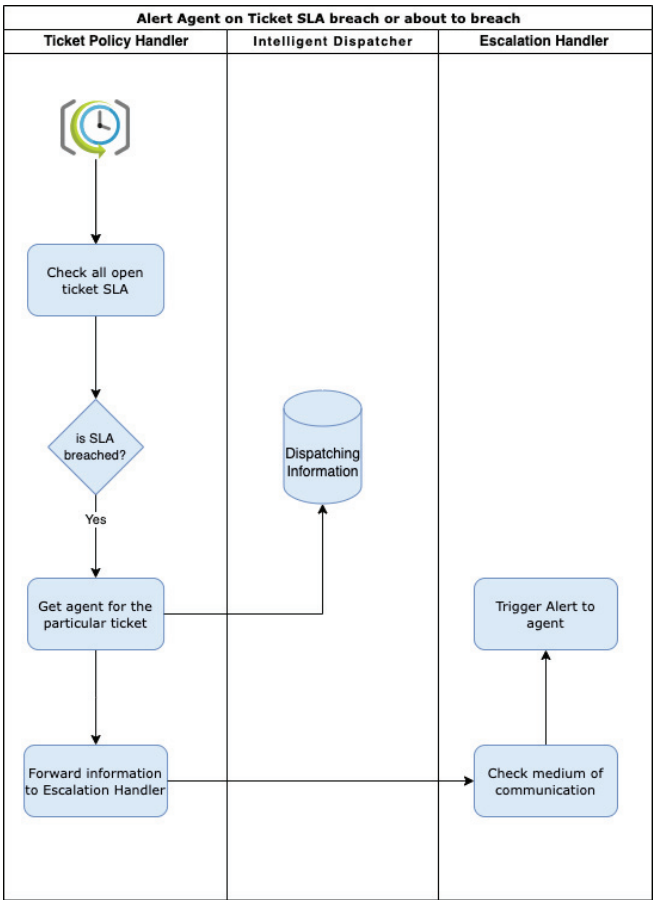

Fig. 3. Illustration of the workflow highlighting how an agent is notified when a ticket is about to breach

- Scheduler triggers sync on a regular interval to the external ticketing tool for the monitored tickets' current status and information.
- If the monitored tickets SLA is about to breach, get the tickets assigned agent from the dispatcher database.
- Forward the information to the escalation handler and check the configured communication medium.
- Trigger the alert to the agent.
- If the agent still doesn't respond, it will try to contact the next agent in the escalation hierarchy.

V. Driving Intelligence using Machine Learning

The proposed workflow is driven by the basic concepts of Artificial Intelligence of Constraint Satisfaction Problem (CSP) and Scheduling. This is a classical resource scheduling problem, and one possible way of solving the same is by using Graph Theory. From prior research done on this by the authors in finding an efficient algorithm for avoiding conflicts between available resources. [7] In the case of ticket monitoring and dispatching, the realization of the problem as a CSP can be referred from another similar done for resource allocation and scheduling and cloud. [8] This technique uses Graph coloring to allocate resources based on various priorities, which is addressed through its chromatic number. The chromatic number forms the basis of the reassignment of tickets to resources, considering the least waiting time is in question for the proposed workflow. Depending on the queue priority and proper utilization of resources, the ticket resolution time can be halved or less than the previous time taken based on resource availability. The process would reject any allocation which enhances the waiting time taken to solve a particular set of incoming incidents for the ticketing system.

Fig. 4. Process flow for Service Ticket Intelligence [9]

Users may create a machine learning-powered, self-driven service experience with the use of service ticket intelligence. Inbound tickets are dynamically categorised by the services in order to route them to the appropriate agent. The agent is then given suggestions for enhancing production efficacy. Service Ticket Intelligence is a component of the offering of SAP AI Business Services.

Deep learning neural networks that have been educated on massive quantities of past records are used by Service Ticket Intelligence. The algorithm recognises the meaning of unorganized incident communications, groups the issue according to the best probable groups, and then suggests to the agent solutions or knowledge base articles from similar issues that have already been resolved. The algorithm becomes improved over time as additional service issues are addressed and as consumers provide comments. [9]

VI. Advantages and Business Implications

Ticket monitoring and dispatching is a major concerns for any operations team. It requires 24X7 availability of support not just for the self-organization but also for customers who can be varied and may have a diverse geographic presence. The number of end users affected by a specific defect can even reach millions depending on the application and may halt critical business processes. This challenge requires an agile mindset from the operations team to maintain complete support in such situations through upgraded monitoring and dispatching mechanisms. Organizations like SAP, Microsoft, etc. with a large customer base for their ERP solutions require this aspect of their business to run smoothly without disruption. The proposed workflow in this paper as a result caters to the problem and highlights the following advantages of its implementation:

- Reduce the Total Cost of Ownership by not hiring specifically for monitoring and dispatching.
- Team members working on monitoring and dispatching can be assigned to any quality work and improve employee satisfaction on day-to-day tasks.
- Improve ticket handling and avoid human errors by having continuous IRT, MPT, APT, and ORT tracking and avoid human errors.
- Provides additional information to the ticket processor which can improve overall resolution time for the unit or system.
- Intelligent ticket distribution, the ticket is assigned based on the current workload of the ticket processor and rules set by the admin.
- Improved routing of tickets leads to better utilization of available resources.

This is a major advantage of the traditional mindset but also puts forward challenges. Some of the key limitations that require to be addressed are as follows:

- Escalation untracked due to the unavailability of resources.
- During migration of bot or any operational activity, there is a halt in the auto dispatching and monitoring process.
- Manual dependency in setting the right priorities needs to be averted.
- Deployment of the aforementioned solution can require different services in the platform that is considered for implementation which could result into extra cost.
- Integration with the Machine Learning capabilities also mandates the presence of Service Ticket Intelligence which may bear cost.

Hence, considering the above aspects, it can be clearly stated that the future of ticket monitoring and dispatching is going to be very different from the traditional means but it needs to consider multiple challenges before a perfect digital assistant is put into place.

VII. Conclusion

Every operations team must deal with the critical issue of ticket monitoring and dispatching. It requires support to be available 24 hours a day, seven days a week for both the self-organization and for customers who may be diverse and have a wide geographic presence. Depending on the application, the number of end users affected by a particular fault might reach millions and potentially stop vital company activities. This research developed a solution to this issue that can recognize the workgroups within a team, comprehend the emotion of the requests, and allocate them automatically. If we do this, we are able to lower the total cost of ownership, give our coworkers better work, prevent human errors, and have intelligent and efficient dispatching.

The digital assistant continuously monitors tickets, intelligently distributes tickets based on user workload, and comprehends the context of each ticket to provide more information about it, such as whether any related issues have been resolved, what kind of solution was offered the last time, and who provided that solution. The intelligent capabilities provided through the usage of a previous research done by the authors on allocation of resources and conflict management using graph coloring provide an essence of making the workflow simpler. It is also complemented by the SAP solution of Service Ticket Intelligence solution that uses ML to perform a manual driven client support system. The advantages with these

integration is multifolds and harmonizes the entire ticket monitoring and dispatching workflow.

VIII. Future Works

This solution is still in the concept phase and it is currently intended to use only within the S/4HANA cloud operations team. The plan is to build the digital assistant on the Django framework and host it on S/4HANA Converged Cloud. This can be tried in other landscapes and used for academic purposes of operational efficiency and improving customer support through intelligent product support.

Acknowledgment

This paper was possible thanks to the support of our parent organization, SAP Labs India. This paper is also possible thanks to the support from the S/4HANA cloud operations team which acts as frontrunners for the knowledge gained on the topic as well implementation of the proposed idea. We would like to thank techNxt for creating a research community for budding authors and also to our managers and colleagues for extending their support in our research journeys.

References

[1] Y. Lee, S. Lee, C. -G. Lee, I. Yeom and H. Woo, "Continual Prediction of Bug-Fix Time Using Deep Learning-Based Activity Stream Embedding," in IEEE Access, vol. 8, pp. 10503-10515, 2020, DOI: 10.1109/ACCESS.2020.2965627.

[2] T. S. Mian, "Automation of Bug-Report Allocation to Developer using a Deep Learning Algorithm," 2021 International Congress of Advanced Technology and Engineering (ICOTEN), 2021, pp. 1-7, DOI: 10.1109/ICOTEN52080.2021.9493515.

[3] M. K. Gopal, G. M, P. Chandra, P. Shetty and S. Raj, "Bugtrac – A New Improved Bug Tracking System," 2022 IEEE Delhi Section Conference (DELCON), 2022, pp. 1-7, DOI: 10.1109/DELCON54057.2022.9753390.

[4] S. De, "A Study on Chaos Engineering for improving Cloud Software Quality and Reliability," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021, pp. 289-294, DOI: 10.1109/CENTCON52345.2021.9688292.

[5] I. Itkin, A. Novikov and R. Yavorskiy, "Development of Intelligent Virtual Assistant for Software Testing Team," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2019, pp. 126-129, doi: 10.1109/QRS-C.2019.00036.

[6] B. Tureczki and K. Szenes, "Interdisciplinary Optimization of Security Operations Centers with Digital Assistant," 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2021, pp. 397-402, DOI: 10.1109/SACI51354.2021.9465632.

[7] S. De and V. Vijayakumaran, "An Efficient Algorithm in Project Management for Resource Scheduling and Conflict Management using Graph Coloring Technique," 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020, pp. 1-6, DOI: 10.1109/INOCON50539.2020.9298193.

[8] S. De, “An efficient technique of resource scheduling in cloud using graph coloring algorithm”, Global Transitions Proceedings, Volume 3, Issue 1, 2022, Pages 169-176, ISSN 2666-285X, https://doi.org/10.1016/j.gltp.2022.03.005.

[9] “What Is Service Ticket Intelligence?”, SAP Help Documents, Available: https://help.sap.com/docs/SERVICE_TICKET_INTELLIGENCE/934ccff77ddb4fa2bf268a0085984db0/2f0e49ac91c24d54acb694d967e0cfc0.html?locale=en-US, Last Accessed: 26.09.2022.

[10] D. Chen, B. Li, C. Zhou and X. Zhu, "Automatically Identifying Bug Entities and Relations for Bug Analysis," 2019 IEEE 1st International Workshop on Intelligent Bug Fixing (IBF), 2019, pp. 39-43, doi: 10.1109/IBF.2019.8665494.

[11] H. Mahfoodh and Q. Obediat, "Software Risk Estimation Through Bug Reports Analysis and Bug-fix Time Predictions," 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), 2020, pp. 1-6, doi: 10.1109/3ICT51146.2020.9312003.

[12] Q. Liu, H. Washizaki and Y. Fukazawa, "Adversarial Multi-task Learning-based Bug Fixing Time and Severity Prediction," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021, pp. 185-186, doi: 10.1109/GCCE53005.2021.9621355.

Author Profiles

Mohamed Naveen is currently part of the S/4 CDX CD S4 Cloud Operations. With over 10 years of experience in different organizations, he has heavily contributed as an SAP ABAP Developer and then as a DevOps Engineer. He is a skilled Cloud Ops Engineer with 4+ years of hands-on experience supporting, automating, and optimizing mission-critical deployments in SAP converged cloud, HEC, and GCP. Also, he has a strong knowledge and experience in developing Python, Django, and REST API.

Suman De is a Product Manager for SAP S4 PM Product Insights & Experience and has been with SAP since 2015. He is a research scholar and is currently pursuing his Doctorate from IIT KGP-CCDS. He has published over 35 research papers in International conferences of IEEE, Springer, and Elsevier, including 4 book chapters. Suman has been associated with multiple conferences and journals of International repute in various roles and has provided Guest Lectures in various institutions. He has been associated with different initiatives at SAP as a mentor and has won the Thought Leadership Champion award a record six times since 2020.