

Artificial Intelligence: Automating Jira Requests

CS4991 Capstone Report, 2025

Grant Costello
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ded5nh@virginia.edu

ABSTRACT

When a problem in the Markel Systems occur, underwriters create uninformative Jira Tickets. My team utilized an internal AI tool and the Jira API to automate this process and add relevant information. The underwriter prompts the AI and the AI collects information about the problem and formats it in a helpful way for the service desk technician to understand and solve. This expedites the time need to solve the issue. The next step is to train the AI to format the Jira Ticket in an easily readable way.

1. INTRODUCTION

The insurance side of the Markel Group exists in the specialty insurance space. They once insured one of the pairs of ruby slippers that Dorothy wore in the original *Wizard of Oz* movie. The slippers ended up being stolen and an entire investigation had to be undertaken to verify that no insurance fraud took place. Markel has insured concert venues, sports cars, mansions, and pretty much anything that there is some risk to owning.

Because Markel insures such big unique items, underwriters rely on underlying systems coded with complicated rules to determine what the policy looks like and what forms are attached. These systems and the rules within them sometimes break. Currently, there are two ways for the groups that interact with these systems to indicate something is broken. The first group is the underwriters. When they find a problem, they submit a form on an intranet

website called ServiceNow. The ticket is then sent to the right people to fix the problem. They then create a Jira ticket for the problem and it is put on the backlog. However, when a developer discovers a problem, they create a ticket in Jira.

This current workflow for system problems is not generalized and makes the whole process slower. When the ticket is received by the correct team, it is formatted differently and the description of the problem can be difficult for the team to understand without reaching back out to the party that discovered the problem. This makes solving these problems time-consuming and less efficient than it could be. My team decided to tackle this problem and utilize AI to do so.

2. RELATED WORKS

According to Schmelzer (2025), Artificial Intelligence can be utilized to help analyze complex problems built on data. When determining risk or how much to insure, much is unknown, so AI can play a part by taking in all known data and making a more accurate guess than current models used by insurance companies. Schmelzer also mentions that insurance operations can be a place where AI is utilized. By making the processes more efficient, those results can be passed to the customer and make their experience better. My project aimed to utilize insurance operations and make that efficiency felt by the customer.

According to Godinho (2025), flexibility is the key to AI driven development. It allows software development teams to work smarter. It is not automating the software development process but adapting to a new tool that can be utilized to stay ahead of competition and stay efficient. My project's goal was to use AI to adapt and make the software development, primarily debugging, process easier for developers.

3. PROJECT DESIGN

The first step of designing how the Jira Ticket Automation would work was to research the current processes showing how a Jira Ticket was created, claimed and completed. We looked at the two processes, both ServiceNow and Jira directly. We emailed and met with some underwriters to understand how the ServiceNow process worked and we asked them about flaws in the current process and about their ideas for improvement.

Then we looked at the other part of the process used by the developers, who both create Jira tickets for developers on other teams and complete Jira tickets. We talked to the developers on the Document Solutions team, who gave us a list of requirements for creating an automated ticketing process and told us where to look for certain data.

We were tasked with adding the automated ticketing functionality to the Insight Hub AI, a tool being developed by the Document Solutions Team to assist underwriters and developers. We had several meetings with the document solutions team, looking “under the hood” at the source code of the Insight Hub AI and how it fit into the overall Insight Hub website.

We looked at different ways to add this functionality to the Insight Hub AI. Other common AIs, ChatGPT, Gemini, and CoPilot all had similar user interfaces but with some

differences. We liked the way CoPilot had a list of user prompts and wanted to implement a similar feature, enabling the user to press a button and see different prompt options, one of which could create a Jira ticket.

After my team had all the requirements for researching and figuring out how to actually implement our project, we split the project up into three different parts: first prompting the Insight Hub AI; then pulling and formatting the data related to the problem; and finally, making the API call to Jira.

We decided to start researching the API call and taking a backend to frontend approach to the project. The Jira API documentation was very lengthy and all over the place with what was needed for our project, though we ultimately found the correct POST request to use. My team did not have much experience with Jira and all of its abilities, since this internship was the first time we had to use it; so we had to learn how the Markel IT Department used Jira. I got in touch with a director in the IT department that is in charge of the Markel Jira Standards and she walked me through the standards needed for a story versus a bug and the fields we needed to include in the API call. We took the API call and found the fields we needed and practiced the call using Postman. It was difficult to find all the required fields because Markel had lots of custom fields in Jira that were not documented and we had to search for that information.

After the API call, we looked at ways to pull data from the Insight Hub AI and include it in the Jira Ticket, while also formatting it in a way that the Jira API could handle and understand. We utilized the Insight Hub AI by asking for any data related to the user's prompt and gave the AI a template showing how we wanted the response returned to us. This part was tricky because, like ChatGPT, even with

clear instructions, the system would sometimes give us information we did not want or which was formatted in the wrong way. Another difficulty was having to learn typescript and C# because the Insight Hub was written with an Angular frontend and a .NET backend.

We had the most trouble with the user interface. While we were working on the Jira call, the Document Solutions Team was continuing development on the Insight Hub AI. We could not create a prompt button because they had not finished with the overall look of the AI, so we resorted to creating a button under the prompt bar. In this way, if the user pressed it, it would start the Jira ticket automation process, taking all of the data from the prior conversation and pulling any related information. The system did not give the user a chance to explain the problem well, but we had to work with what was within our control. On the day of our presentation, we had to fix our code because the Document Solutions team had accidentally deleted a line that we wrote, making development very difficult.

4. ANTICIPATED RESULTS

We were not able to see the results of the automated Jira project we implemented into the Insight Hub AI. However, we expect that it will decrease the amount of time developers spend fixing problems instead of developing new products. Every bug that is found adds time to both the underwriter who finds a bug and has to request a fix and the developer who has to fix it.

Bugs are not avoidable, so our new process should decrease the wasted time dealing with bugs. Underwriters will save time because they can just type the problem into the Insight Hub AI and press a button, creating a Jira Request Ticket. They can then spend more time selling insurance, making Markel more money. We expect that developers will save

time fixing these bugs because of the formatted data that we included in the Jira Ticket. The Insight Hub AI is now adding all information about the problem into the Jira request, so the developer does not have to search for the problem, enabling them to focus on solutions instead.

5. CONCLUSION

The importance of the automated Jira Ticket creation process lies with the time saved doing mundane tasks. Underwriters and Developers both save time by using the Insight Hub AI to complete the creation process entirely. The AI has access to all the data related to the underlying systems and is able to pull from that data when an error is found, creating a much better Jira ticket than an underwriter or developer every could by knowing everything related to the problem and perhaps in some cases, the place to look to solve the problem.

6. FUTURE WORK

For future work, the Insight Hub AI needs some user interface work. When developing our process, the Insight Hub AI's UI was being developed at the same time. Because of those changes, we had to add a button to start the process. To improve in the future, we could add a prompt for the user to type in to start the process.

In addition, the Insight Hub's responses need to be more standardized. When asking the Insight Hub AI for data, it was sporadic in its variety of responses and sometimes it didn't understand the problem, so we need to take it and make it respond in a more similar way each time to make it easier for the developers to solve the problem.

REFERENCES

Schmelzer, R. (2025, February 13). *From reducing risk to improving operations: The role of AI in insurance.* Forbes. <https://www.forbes.com/sites/ronschmelz>

er/2025/02/13/from-reducing-risk-to-improving-operations-the-role-of-ai-in-insurance/

Godinho, N. (2025, February 6). *Council post: AI and flexible team models: Empowering instead of replacing*. Forbes. <https://www.forbes.com/councils/forbestechcouncil/2025/02/06/ai-and-flexible-team-models-empowering-instead-of-replacing/>