Michael Roseweir

07/23/2021

CS Assessment Alternative Tools Research

1. Do some internet research and find two new tools to explore– one for Continuous Integration, and one for Real Time Error Monitoring. You cannot use the ones you used earlier this week.

    a. The tool that I selected for CI is Jenkins, a community driven and open source CI/CD tool. Jenkins is also an on-premise software which would be extremely useful in environments where sensitive information is stored.

    b. The tool that I selected for Real Time Error Monitoring is called Raygun. A live error tracking tool that supports all languages and all frameworks with extremely competitive pricing.

2. Record the unique value-add, or notable features for each tool. Imagine you were an engineer trying to convince your manager to use each tool. What would you say to convince them?

    a. Being that Jenkins is an on-premise software, I feel as though it adds an extra layer of security. This is very useful for companies that are handling sensitive customer information, and need to ensure that all the information in your website is kept as secure as possible. Another interesting feature is that Jenkins is both open source, and runs on community-driven updates. I believe this would contribute to more realistic updates and features implemented by individuals who actually use the software on a day to day basis.

    b. Raygun has several notable features that I liked when researching the product. One of the ones that stood out to me when comparing a couple different products was the ability to track errors through both desktop and mobile devices. It also logs all of the user's workplace environment when an error is recorded, allowing you to see the version, host, OS, browser and more. You can even sort the errors by the environments in order to see if it is a specific environment that is causing issues for your application.

3. For each tool, find the Getting Started instructions for how to begin using the tool. Evaluate the quality of these instructions. Is there an easy way to get started with the tool? Is the process well-documented? Are there any special tutorials or sandboxes available to make trying it out or learning the tool easier? Include a summary of how to get started with the tool, helpful links, and any other notable resources for this process.

    a. Jenkins documentation is very impressive. Their website (jenkins.io) has a massive documentation library and gives several different options when it comes to installation. Jenkins uses Docker as one of the options for running the software in an isolated environment. There are very thorough walkthroughs and instructions on how to install and begin using Jenkins. Including giving you the lines of code that you just need to run in your local environment. Here is the link to the Jenkins Installation Guide.

    b. Rayguns website also boasts a great set of documentation when it comes to both installation as well as features. When you are looking for the installation instructions you can easily search by the language that you are planning to use the application for, and it gives detailed instructions on how to to get the application running. The documentation gives you the lines of code necessary to paste into your document to begin tracking with Raygun. Including a block of code that is designed to catch errors that are thrown while the page is loading. Here is the link to the React Installation Guide for Raygun.

4. How long has this tool been around? How popular is it? Summarize the maturity and market share of each tool. To answer these questions, check out any public official Github repos for the tool (are they in active development? what is the date of the earliest commit?), the tool's public website for any notable mentions of current companies that use the tool, and any other information that will help you determine if each tool is compelling to other companies and how new it is.

    a. Jenkins is a tool that has been around for a long time and has a bunch of community support on Github. It has over 17k Stars, 31,000 commits as well as almost 7,000 forks. Looking at the overview of the master/main branch, some of the oldest commits date back to 13 years ago. While some of the newest commits were done less than 24 hours ago. This shows me that it is still extremely active in the community, and they are in a constant state of development. Reading the notes on their Github showed me that Jenkins has over 1,700 plugins that support automating almost anything. Being that Jenkins has been around for so long, it seems as though the maturity for the product is high.

b. After looking through Raygun's Github, I was able to see that they have commits dating back about 6 years, with the most recent commits being within the past 3 months. The Github for Raygun is a little bit more spread out across a few different repositories, so I am mainly looking at the raygun4js repo. The repository has far less stars, forks, and commits when compared to Jenkins, but it does show active updates and constant development. The Github also has several sets of documentation within the repo. Including documentation for both React and Angular.