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.

   Jenkins

   RayGun

2. Start a new document (Google Docs or similar) and 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?

   Jenkins is one of the most popular tools for Continuous Integration, it has an open source that is automated to servers. Meaning that it can handle any tasks such as building, testing, deploying and delivering the software.

   It has a very friendly UI and the documentation is easy to read and follow. Best part of it all? It's free and always updated.

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.

   Jenkins

   https://www.jenkins.io/doc/book/

   https://www.jenkins.io/doc/pipeline/tour/getting-started/

   https://www.youtube.com/watch?v=LFDrDnKPOTg

   RayGun

   https://raygun.com/documentation/

   https://raygun.com/documentation/language-guides/overview/

   https://www.youtube.com/watch?v=LxHzvSNGJSU

To get started on these tools, here's 2 lists of links that I've provided that will help get the installation of the software into our VSCode. Both straight forward and easy to follow. I also included a Youtube video to help understand what Jenkins does and how it can be very beneficial to use for development.

Same thing goes for RayGun, however. RayGun is not free and will have to pay for it to use some of their features. But it is worth the investment fo the value that it provides and how it can help handle the errors in real time. It supports many programming languages from PHP, Node.js, Vue.js, Python and much more. I have provided a link that will show more frameworks that it supports. So if there is ever a need to switch programming languages, this will make it easier for us as a whole.

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.

   https://github.com/jenkinsci

   Jenkins was created back in 2004, offering 1,700 plugins and being able to be used anywhere in the world. Till this day, there are many DevOps Engineers and Web Developers that are contributing and using Jenkins. I provided a link above to their Github profile to see the active users there.

   https://github.com/MindscapeHQ

   There is no specific release date when RayGun was created, however. They do have a very active user base. From what I was able to find, they have made UI improvements, have a page dedicated of their “best releases” and showing what they have improved on. Giving user feedback and listening to their reports.

```
// TINYARRAY SUMMARY
// This was faster compared to the extralargearray and mediumarray
// since it has a smaller array and objects to deal with,
// it will process the information faster then its 2 counter parts
// the start was at 34 ms, however. It came back fast in 90 ms
```

```
// MEDIUMARRAY SUMMARY

// Although not as fast as tinyarray, since it has more objects in the
array
```

```
// It is still faster then the extralargearray and still was able to process

// the information fairly quickly, but not as fast as tinyarray

// the start was at 167 ms, however. It came back fast in 125 ms
```