1) Continuous Integration tool: **Jenkins**

   - **Notable features:** Jenkins is an open-source Continuous Integration server. Jenkins helps development teams with running automated tests. Jenkins will continuously tes a project build and show errors early into the development process.
   - **Documentation:** Jenkins provides great in-depth documentation on how to get started and use their product. The 3 main categories of documentation include, (User handbook, Tutorials, and Resources). Also, Jenkins provides developer documentation on how to develop your own Jenkins plugins. The documentation starts with a Guided Tour… (https://www.jenkins.io/doc/pipeline/tour/getting-started/) making it easy to get started.
   - **Popularity:** Jenkins, founded in 2011, is the number 1 option for development teams. The market shared for Jenkins is 51.25% and has more than 6070 companies using the product. This is significant when compared to CircleCI, which only has a market share of 6.41% and 691 companies using it. Jenkins has both a github and linkedin account which shares its recent updates and work. It has a MIT License which makes it user friendly and open source friendly making it a very popular option.

2) Real Time Error Monitoring tool: **Raygun**

   - **Notable features:** Raygun is a cloud based platform that provides error, crash, and performance monitoring for web applications. Raygun allows development teams to visibly see errors that their users encounter and provides code-level detail into root causes. With Raygun, development teams can both identify and resolve real user issues and show how the application performed for each user session and page load.
   - **Documentation:** The documentation for raygun is user friendly and provides a lot of information on how to get started. It includes **(**language guides, product guids, Raygun for team development, accounts and billing, privacy and security, and offers further support**)**.

Raygun offers a free trial and additional resources such as demos, blogs, and customer stories.

- **Popularity:** Raygun, founded in 2007, is an award-winning application monitoring company for modern software teams. They have both a github and a linkedin account which documents their recent work and updates. Its notable clients include Coca-Cola, Domino's Pizza, Microsoft and Samsung. It has been recognised as a finalist for Wellington Gold Awards 2016 & 2019, won NZ Hi-Tech Awards Start-up of the Year 2015, and won the Innovative Hi-Tech Software Product of the Year 2014 at the NZ Hi-Tech Awards

µs = microsecond

extraLargeArray results:
insert 1.1357913 s
append 3.4422 ms

tinyArray results:
insert 40.3 µs
append 94.2 µs

smallArray results:
insert 52.5 µs
append 109.3 µs

mediumArray results:
insert 188.5 µs
append 148.3 µs

largeArray results:
insert 9.555 ms
append 647.4 µs

As the arrays get larger the run time increases significantly. The runtime for the .push() function seems to be running faster depending on the array size. The 2 smallest arrays have a faster .push() runtime compared to the unshift() runtime. According to the data I gathered, the tiny and small arrays have a faster .push() runtime compared to their .unshift() runtime. Yet, when the array becomes medium sized or bigger, the runtime for unshift() is quicker than the runtime for .push().

push adds to the **end** and rarely needs reallocated memory+copy over.
unshift adds to the **start** and *always* needs to reallocate memory and copy data over