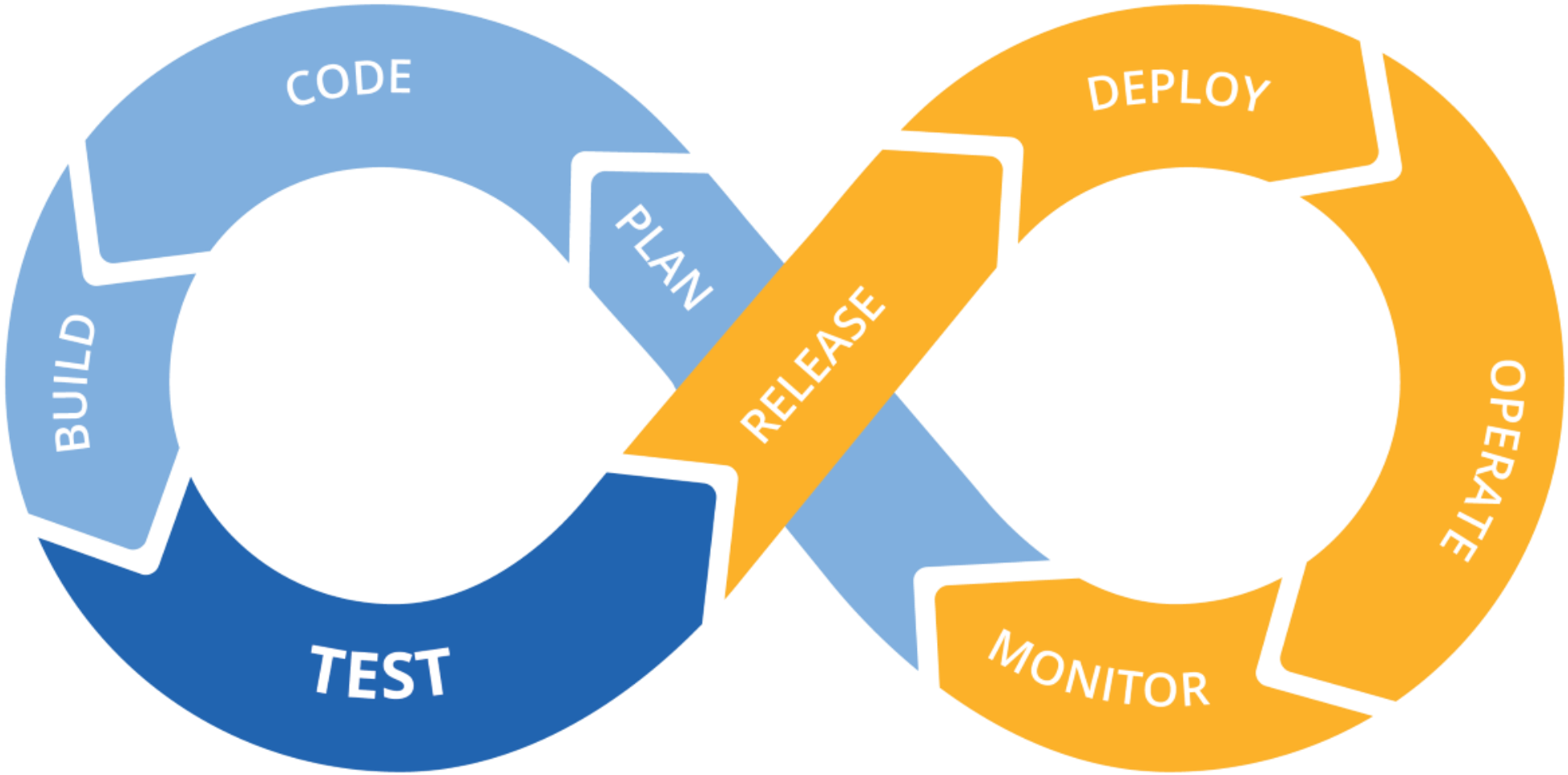
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR

What is CI/CD

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.

Why should we use CI/CD

CI/CD allows organizations to ship software quickly and efficiently. CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

What is CI

Continuous integration is the practice of merging all developers' working copies to a shared mainline several times a day, which is then automatically built and tested before its merged with the shared repository

What is CD

Continuous delivery is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, without doing so manually. It aims at building, testing, and releasing software with greater speed and frequency

Benefits of CI/CD

- Faster time to market
- Reduced risk
- Shorter review time
- Better code quality
- Smoother path to production
- Faster bug fixes
- Tighter feedback loops
- Measurable progress

Benefits of CI/CD to the business

- Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD.
- Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly.
- Save time: You will be saving time because everything will be automated, and you can watch the build process in Circle CI and because of this you will know as soon as failures happens.

- Constant releases: There is no waiting for one big release for a feature now small features will continuously go out to customers with every commit to GitHub
- Build quality: When you build your tests, they will be checking to make sure that everything is working as expected before it ever moves forward.
- Quick build failures: This is important because the more time you have hardware running the higher the cost.
- Higher security:You will be able to avoid security loopholes because you will have tests in place to check for security loopholes.