UdaPeople:
Continuous Integration and Deployment

Reinforcing and expediting development and delivery at lesser costs

Continuous Integration and Deployment

What is it?

- Continuous Integration is a mechanism by which developer code is integrated, several times a day, into a central codebase that is subjected to frequent automated testing.

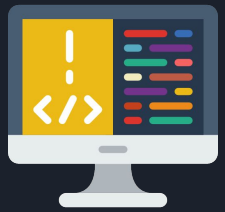
- Continuous Deployment is a mechanism by which the central codebase is released to the customer quickly and frequently, bringing the latest and greatest faster than ever before.

Together, with both CI and CD, we're offering Continuous Delivery to our customers. Our customers can expect...

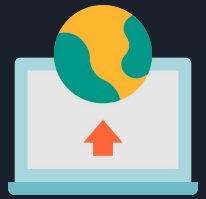
- Shorter turnaround times on defects, new features, and major infrastructure changes.
- Safer releases with less breaking changes.

CI/CD Saves Money

Continuous Integration and Deployment saves money in every step of UdaPeople, from developer to customer.



- Frequent tests in the central codebase identifies bugs sooner rather than later, reducing the time investment for defect work and resolution across the lifecycle.



- Automated deployment reduces mistakes, takes less time, and is a less esoteric process, allowing developers to help with more than just code.

CI/CD Gives Everyone Options

Continuous Integration and Deployment gives UdaPeople stakeholders flexibility…

- Continuous Integration provides a central codebase that undergoes heavy testing.
  - Developers can choose testing requirements and test how they want. The pipeline performs the tests, not a dedicated tester.
- Delivery to the customer can be done immediately, after automated tests, after customer testing and approval, or a combination thereof.
  - Automated deployment makes these processes easy to perform or change.
- Customers can request updates to delivery cycles with little impact to developer schedules.
  - Hourly, nightly, weekly, or monthly builds are automated the same way, just at different intervals.

Initial Expenses of CI/CD

CI/CD saves money over the lifetime of UdaPeople, but has some upfront costs.

- Developers will need to be trained on CI/CD principles, workflows, and technologies.
- Existing processes need to be refactored to align with meet CI/CD criteria.
- The shift towards CI/CD is itself continuous; Rome wasn't built in a day.
- Initial efforts towards CI/CD should be performed in parallel with our existing processes.