**Phases and tools used CI AND CD CYCLE:**

**Continuous Development:** This phase primarily focuses on project planning and coding. It’s a continuous process where developers are required to code whenever any changes occur in the project requirement or in case of any performance issues. **Tools:** Git, CVS, Subversion, Mercurial, Maven, Garden.

**Continuous Integration:** This phase involves integration of updated code and add-on functionalities to the existing code and furthermore bugs are detected through unit testing. **Tools:** Jenkins, CircleCI, Buddy.

**Continuous Testing:** This phase involves automated testing for bug in software and if found the code is sent back for integration phase for modification. **Tools:** Selenium, Ranorex, Test.ai, TestSigma, TestNG, Junit.

**Continuous Deployment:** This phase includes configuration management to make the deployment of code on servers accurate and smooth. The teams release the code to the servers and schedule updates for servers, keeping it consistent throughout. **Tools:** UrbanCode Deploy, DeployBot, Juju, GitLab, CircleCI, Travis CI, Puppet, Chef, Ansible.

**Continuous Monitoring:** This phase involves monitoring the functionality of the code to detect system errors, low memory on servers etc. **Tools:** SolarWinds, Sematext, Nagios, New Relic, Redgate, Datadog, CloudZero, ProsperOps, Xosphere.

**Continuous Feedback:** This phase involves analysis and improvement of the code with regard to customer’s feedback. The team can either use structural or non-structural method to collect feedback from customer. **Tools:** Slack, Teams, Zoom, JIRA, Asana, Celoxis.

**Continuous Operation:** This involves automating the application’s release and all these ipdates that help you keep cycles short and give developers and provide more time to focus on developing. **Tools:** BigR.io, JIRA, Opsgenie, Statuspage.

**Difference between Continuous Development and Continuous Deployment:**

|  |  |
| --- | --- |
| **Continuous Development** | **Continuous Deployment** |
| Continuous Development involves planning and coding the product the team is developing. | Continuous Deployment involves code changes to the application released automatically into the production environment. |
| **Stages:** Integration, Testing, Deployment, Monitoring. | **Stages:** Version Control, Acceptance tests, Independent Deployment, Production Deployment. |
| It helps faster software development as it removes code conflicts and code incompatibilities early. | It enables organizations to respond to market demands for high quality solutions in a lesser amount of time |

**Continuous Integration:**

This phase involves integration of updated code and add-on functionalities to the existing code and furthermore bugs are detected through unit testing. One of the main benefits of integrating regularly and testing each integration is that you can detect errors more quickly and locate them easily. Since each integration or update to codebase is usually small, pinpointing the exact change that causes the error can be done quickly. The CI runs after developer checks in the code and throws out the results in seconds. So, it allows the developers to know immediately if his or her code has successfully built or broken.It also lets the developer know if his code has successfully integrated with the other’s code or broken, that something that another team member has done to a different part of the code base. Hence, CI does the quicker code analysis and makes the later merges simpler and error free.

A continuous integration server easily models and visualizes complex workflows (enabling continuous delivery) and provides an intuitive interface for building continuous delivery pipelines. A continuous integration server offers the ability to do the following:

* Run automatic builds, tests and releases in a single place.
* Deploy any version, anytime
* Keep configuration orderly
* Support plug-ins to enhance functionality
* Monitor your project’s repository
* Pull changes and perform tasks you defined for successful commit
* Send feedback to the relevant project members with details of the build

**The CI process includes:**

1. Merging of all the Developers code to the main line
2. Triggering a build
3. Compiling the code and making a build.
4. Carrying out the unit test.

**Importance:** Continuous Integration enables better transparency and farsightedness in the process of software development and delivery. It is beneficial across all phases of DevOps.

**Benefits:**

1. Smaller code changes
2. Fault isolations
3. Faster Mean time to Recovery
4. More Test Reliability
5. Faster release rate
6. Smaller Backlog
7. Customer Feedback Integration