**CONTINOUS INTEGRATION:**

**what is continuous integration?**

Continuous Integration (CI) is a development practice that requires developers to commit code into the shared repository several times a day. Each commit in the repository is verified and automatically build allowing teams to detect problems in the earliest.

By integrating regularly developers can detect the errors easily and locate them easily.

Because CI is so frequent there is less back-tracking to discover where things went wrong. So, developers can spend more time on building features.

Continuous integration is cheap. If you don’t follow a continuous approach you will have a longer period between development and integrations. This makes it more difficult to find and fix problems. Such integration problems can easily make the project fall into off-track or may fail altogether.

**Benefits of continuous integration to the organization:**

* Say bye to long and tense integrations.
* Increase visibility makes it to enable greater communication.
* Spend less time in debugging and more time in developing.
* It stops waiting to find out whether the code works.
* Reduce integration problems which makes it to deliver rapidly.

**Continuous Integration is more than a process:**

CI is backed by several principles and practices

The practices include:

* Maintain a single source repository.
* Automate the build.
* Make you build self-testing.
* Every commit should build on an integration machine.
* Keep the build fast.
* Test in a clone of the production environment.
* Make it easy for anyone to get the latest executable version.
* Everyone can see.
* Automate deployment.

**How to do the CI:**

* Developers check out their code into private workspaces.
* When done commit the changes into the repository.
* The CI server checks for the commits in the repository.
* The CI server builds the code and does unit and integration tests.
* The CI server releases deployable artifacts for testing.
* If the build fails, the CI servers’ alerts team.
* The team fixes the code at the earliest.
* Continue to continuously integrate and test throughout the project.

**Team responsibilities in CI:**

* Commit the code frequently.
* Don’t commit broken code.
* Don’t commit untested code.
* Don’t commit when the build is broken.
* Don’t go home after committing until the system builds successfully.

**Quote:**

*Continuous integration doesn’t get rid of bugs in the code, but it does make dramatically easier to find and remove them.*

* **Martin fowler.**