**Continuous Integration** :

(CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early

(<https://www.thoughtworks.com/continuous-integration>).

By regularlyintegrating, one can detect errors quickly, and locate them more easily(<https://www.thoughtworks.com/continuous-integration>).

How does continuous integration works:

* Developers check out code into their private workspaces
* When done, commit the changes to the repository
* The CI server monitors the repository and checks out changes when they occur
* The CI server builds the system and runs unit and integration tests
* The CI server releases deployable artefacts for testing
* The CI server assigns a build label to the version of the code it just built
* The CI server informs the team of the successful build
* If the build or tests fail, the CI server alerts the team
* The team fixes the issue at the earliest opportunity
* Continue to continually integrate and test throughout the project

(<https://www.thoughtworks.com/continuous-integration>)

Practices of Continuous Integration:-

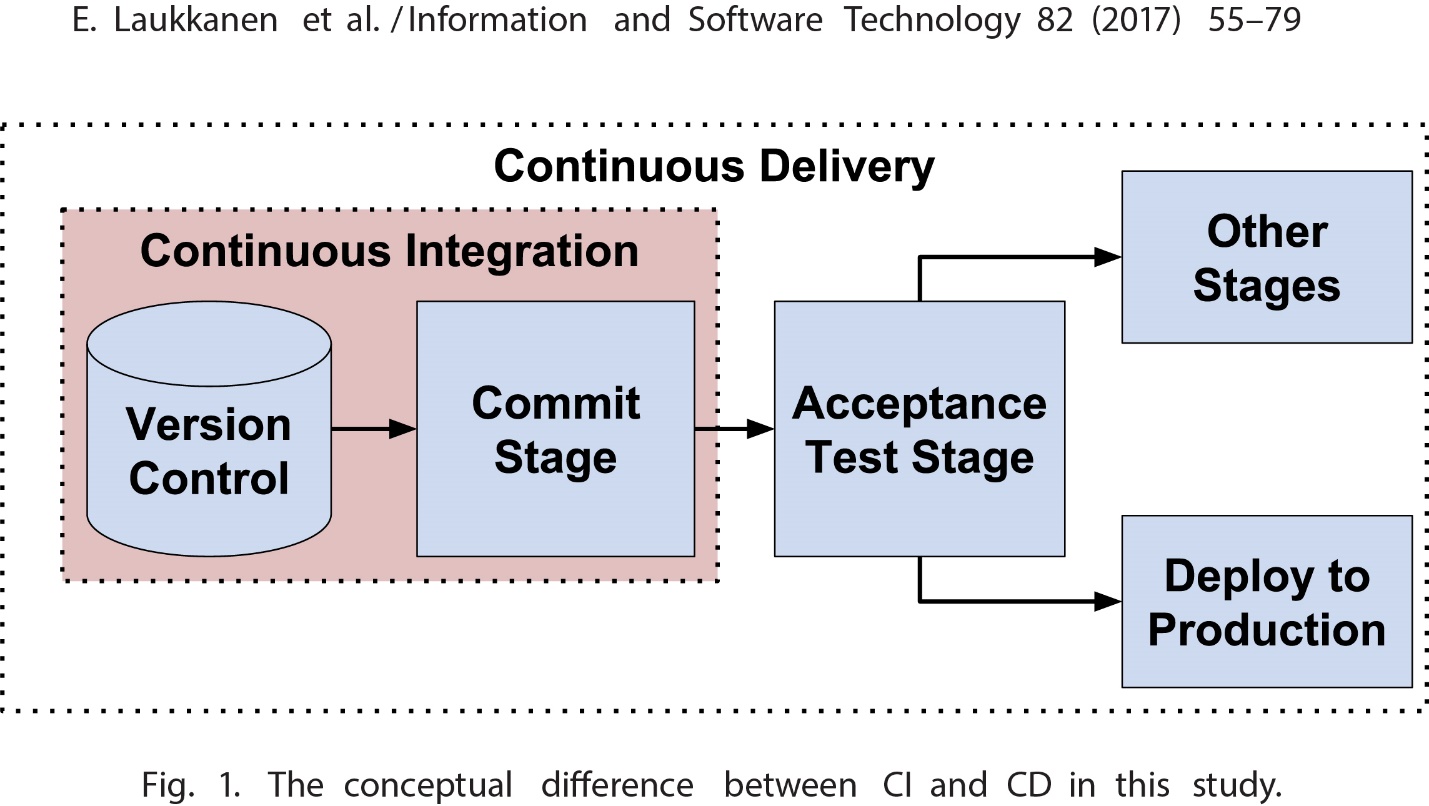
* Maintain a Single Source Repository.
* Automate the Build
* Make the Build Self-Testing
* Everyone Commits To the Mainline Every Day
* Every Commit Should Build the Mainline on an Integration Machine
* Fix Broken Builds Immediately
* Keep the Build Fast
* Test in a Clone of the Production Environment
* Make it Easy for Anyone to Get the Latest Executable
* Automate Deployment

(Martin Fowler, 1 May 2006, https://www.martinfowler.com/articles/continuousIntegration.html)

**Continuous delivery (CD):**

Continuous delivery (CD) is a software development discipline in which software can be released to production at any time (] M.Fowler,ContinuousDelivery,2013) . The discipline is achieved through optimization, atomization and utilization of the build, deploy, test and release process (J. HUmble and D.Farley , Continouse Delivery : Reliable software Releases through build , test , and Deployment Automation,2010) .

CD extends CI by continuously testing that the software is of production quality and by requiring that the release process is automated. The difference between CI and CD is further highlighted in Fig. 1 where it is shown that while CI consists of only a single stage, CD consists of multiple stages that verify whether the software is in releasable condition.



(EeroLaukkanen ,JuhaItkonen , Casper Lassenius),

Characteristic of Continuous Delivery :-

1. Valuable software

Developing valuable software is a goal that has long been on the Agile manifesto (Beck et al.,2001). However , it Is not an easy goal to achieve .Before adopting CD ,some of our teams had been using an Agile method called Kanban(Anderson,2010); however, due to delivery problems , we still had situations where a team had completed a feature but could not deliver it to production to obtain users’ feedback. Consequently , they built additional functionalities on top of that feature, simply assuming it was useful .Unfortunately, when they finally delivered the software to the users, they found out that the feature was not what the users needed .Even worse , by that point , significant effort had been spent on the feature and the additional functionalities. An important objective of our CD implementation was to alleviate this problem . We want teams to build valuable software rather than spend time on features that users do not need(Lianping Chen , Continuous Delivery : Overcoming adoption challenges ,2017).