**Title** : files manager

**Students Names**: Tomer Cohen, Tal Shwartz, Yonatan Uzan

**Source**: <https://www.mabl.com/blog/what-is-cicd> {title: **CI/CD Tools}**

With the move to DevOps, there has been a surge of new automation tools to help with the CI/CD pipeline. These automation tools typically integrate with various established popular developer tools, including code repository systems like GitHub and bug tracking systems like Jira. As SaaS has become a more popular delivery model, many of these tools are running in the cloud, the same place where many modern developers are running their apps ([including the ones at mabl!](https://www.mabl.com/blog/why-we-chose-google-cloud-platform-over-aws-at-mabl)).

The most popular automation tool is [Jenkins](https://jenkins.io/) (formerly Hudson), which is an open source project supported by hundreds of contributors as well as a commercial company, [Cloudbees](https://www.cloudbees.com/" \t "_blank). Cloudbees even offers [several different Jenkins training programs](https://www.cloudbees.com/jenkins/training) and [product add-ons](https://go.cloudbees.com/plugins/).

Besides open source projects, there are several modern commercial software automation products available including [CircleCI](https://circleci.com/" \t "_blank), [Codeship](https://codeship.com/" \t "_blank), and [Shippable](https://www.shippable.com/). Each of these has several different advantages and disadvantages for specific workflows. To really understand which will work for you, I’d encourage trying each of them specifically within your developer workflow to see how they work in your environment (how they work with your tools, your cloud platform, your container system, etc.).

Workflow is a description of a work process within an organization.

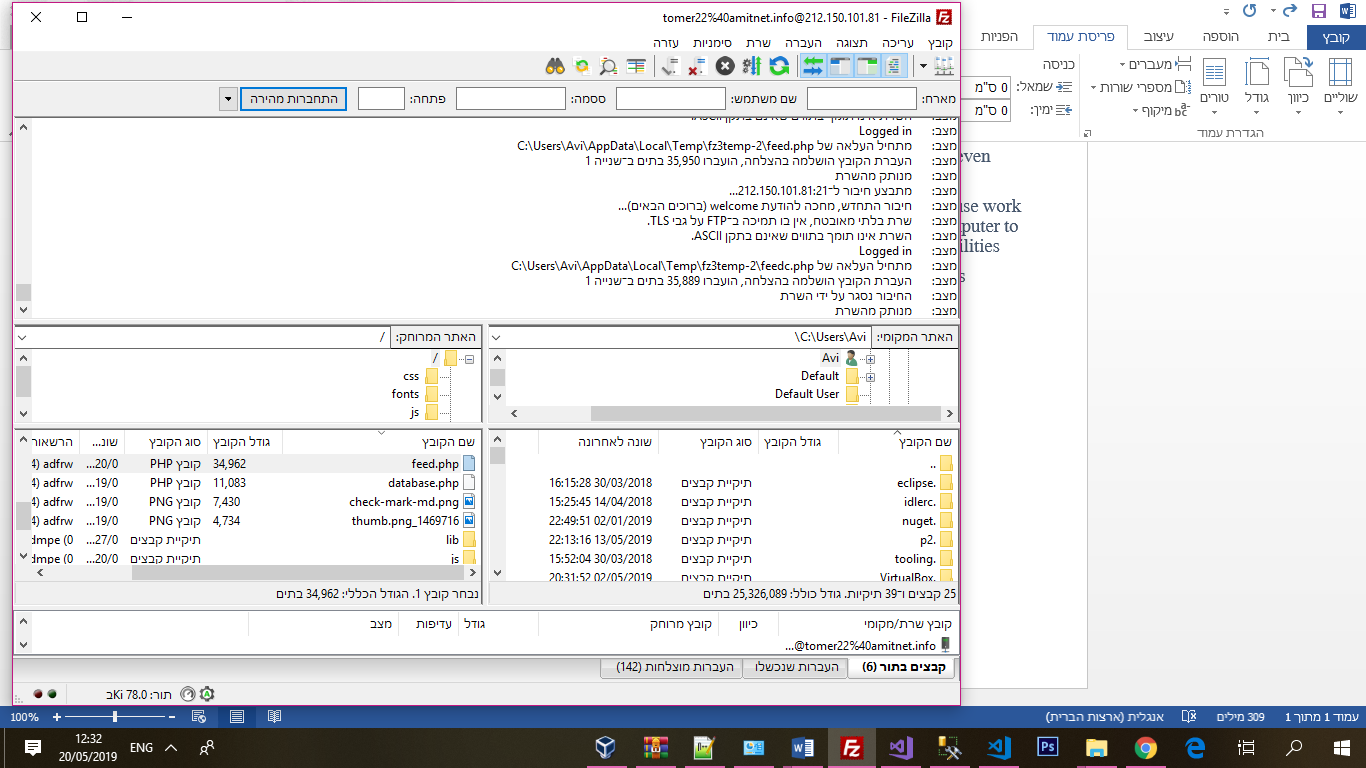
Workflow is a "mini" of predefined applications whose function is to optimize and automate a wide range of business processes. for example:

Collection of signatures, feedback, round of approvals Serial or parallel certificates for programs and documents, routine status monitoring, document routing to multiple destinations, and more.

The work process can be related to different departments, different employees and even between different workflows.

Many organizations spend a lot of money on faulty and inefficient workflows because work processes are often managed manually. Working with SharePoint enables your computer to monitor your work processes and save time and money. Moreover within the capabilities

The SharePoint Workflows can be configured to automate and automate workflows.



**Title** : Code Commit and Version Control

**Students Names**: Tomer Cohen, Tal Shwartz, Yonatan Uzan

**Source**: <https://www.mabl.com/blog/what-is-cicd> {title: **CI/CD Tools}**

**Code Commit and Version Control**

With *Git* as the code repository, we adopted the following process for committing the code and triggering a *pre-build*:

* Created a *pull request* to check for periodic updates from individual developers, and commit the code to the code repository with appropriate versioning. *Git* generates a dynamic version number, based on the branches and the changes that were committed prior to the current build
* Using *NuGet*, a Microsoft-specific tool to identify and pull together all dependent code, a *NuGet*package is generated, which is a single ZIP file with the *.nupkg* extension, containing compiled code (DLLs), other files related to that code, and a descriptive manifest with version number and a date-time stamp.
* Any new code commits triggers an automated*pre-build* to check the sanctity of the code that is checked in.
* As part of *pre-build*, *impacted tests* (unit tests) are run automatically.
* If the *pre-build* is unsuccessful, an *automatic build failure notification* is generated

**[Related :  Independent software testing is an Oxymoron in the age of DevOps and continuous delivery](https://www.comakeit.com/blog/independent-software-testing-is-an-oxymoron-in-the-age-of-devops-and-continuous-delivery/" \t "_blank)**

**Continuous Integration**

*Continuous Integration (CI)* is a development practice that integrates code created by multiple developers through an automated framework. We used Microsoft *Visual Studio Team Services (VSTS)* as the CI/CD tool, and adopted the following process:

* If the *pre-build* is successful, an automated *merge-request* is generated.
* The *merge-request* triggers a new build, which builds the solution and merges with the master branch.
* Automated regression tests are executed on the solution. The Visual Studio Test task automatically runs tests included in the app assemblies, but there is a wide range of configuration options that can be used to run only specific tests. Refer [*Run Tests using Visual Studio task*](https://github.com/Microsoft/vsts-tasks/blob/releases/m109/Tasks/VsTest/README.md)for more information.
* If the regression tests are unsuccessful, an *automatic failure notification* is generated.

