

## **HOW CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY HELPS IN PROJECT MANAGEMENT**

Project's lifecycle consists of 5 process groups such as: initialization, planning, execution, monitoring & controlling, closing [1]. Inside monitoring & controlling group team checks that application does not have defects and verifies that application meets project requirements. Inside closing group we build and distribute application. Activities in these groups can be time consuming depending on project complexity. More over, if project is developed with hybrid or adaptive workflow, then more time will be spend, because team needs to repeat these processes again and again. To resolve this issue and to save a time it is recommended to use Continuous Integration and Continuous Delivery approaches.

Continuous integration (CI) is the strategy of building and validating a software-based solution automatically whenever a file is checked into your configuration management system. Validation can occur via several strategies such as automated regression testing and even static or dynamic code and schema analysis. CI enables developers to develop a high-quality working solution safely in small, regular steps by providing immediate feedback on code defects.

CI includes next steps such as [2]:

- ensure build readiness – ability to check the status in web UI, notify team via email or message;
- obtain current source – only specific branch from source code is used for building and distribution, so there is a small chance that wrong source code will be used;
- perform static code/schema analysis – ability to run linter or 3<sup>rd</sup> party tool like SonarQube etc. that will help to improve your source code and avoid potential bugs;
- build solution – predefined steps with secret credentials will be used. In result you will not miss anything and there is no need to share secret credentials with a team;
- run regression test suites – ability to run unit test, integration tests or end-to-end tests. It will help to save time for testing;
- report results – ability to get build status, code analysis and testing report.

It brings next benefits such as [3]:

- reduced risk – you will not build broken source code or code with defects;
- shorter review time – manual work will be automated;
- faster bug fixes – avoid human mistakes, faster build and test cycles.

Continuous Delivery (CD) – is a strategy of building and delivery of application to clients automatically [4]. It brings next benefits such as [5]:

- low risk releases – automated build and test process will prevent from distributing low quality product;
- faster time to market – tools for distribution will do all manual and routine work for you;
- lower costs – you don't need to use human resources for work that can be automated;
- efficient infrastructure – it is easier to develop the app because all infrastructure is established;
- measurable progress – you can predict and estimate time for automated things, in result your estimation will be accurate;
- tighter feedback loops.

There are a lot of tools and services that will help you to integrate CI/CD to your project. Different tools have different features and pricing. Here is a list of most popular tools: Jenkins, CircleCI, TeamCity, Bamboo, GitLab, Buddy, Travis CI, Codeship, GoCD, Wercker, Semaphore, Nevercode, Spinnaker, Buildbot [6].

According to the statistic using of CI and CD in your workflow can save up to 20% of time [7]. When CI/CD is used, code quality is improved and software updates are delivered quickly and with high confidence that there will be no breaking changes.

### **References:**

1. <https://thedigitalprojectmanager.com/projects/pm-methodology/project-management-life-cycle/>
2. <https://www.pmi.org/disciplined-agile/agile/continuousintegration>
3. <https://www.jetbrains.com/teamcity/ci-cd-guide/benefits-of-ci-cd/>
4. Назаров О.Ю. Використання технологій та методів CI/CD для розгортання коду в хмарному середовищі. Збірник наукових праць ЛОГОС. <https://doi.org/10.36074/logos-30.04.2021.v1.54>
5. <https://continuousdelivery.com/>
6. <https://katalon.com/resources-center/blog/ci-cd-tools>
7. <https://www.techaheadcorp.com/blog/how-ci-cd-save-app-development-time/>