



IT6039 SOFTWARE
TESTING &
MAINTENANCE

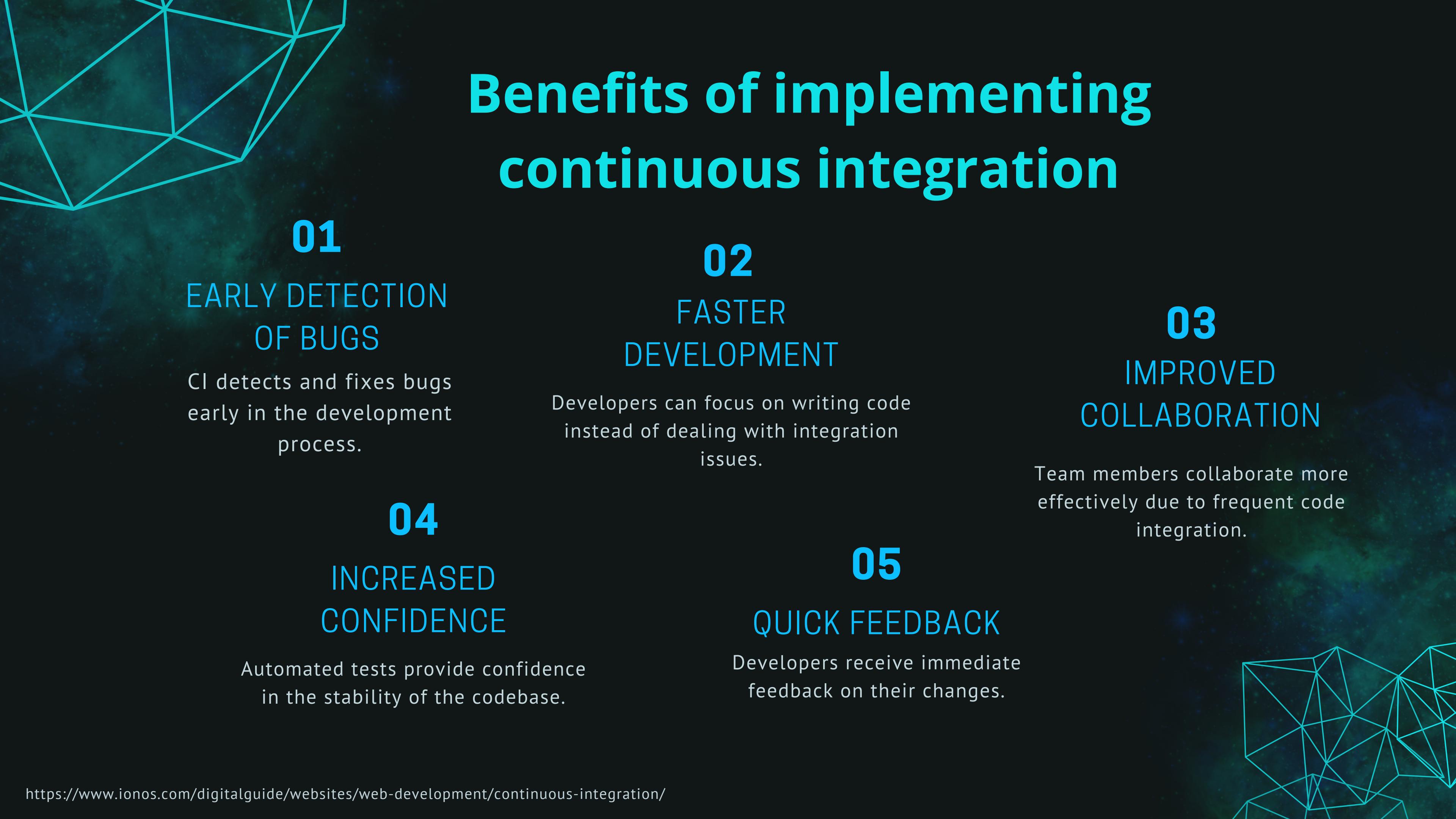
GUILLERMO OZAN

What is continuous integration?

(CI)

Continuous Integration is a software development practice where code changes from multiple developers are automatically integrated into a shared repository several times a day. Each integration triggers automated tests and builds, ensuring that the new code does not break the existing codebase.





Benefits of implementing continuous integration

01

EARLY DETECTION OF BUGS

CI detects and fixes bugs early in the development process.

02

FASTER DEVELOPMENT

Developers can focus on writing code instead of dealing with integration issues.

03

IMPROVED COLLABORATION

Team members collaborate more effectively due to frequent code integration.

04

INCREASED CONFIDENCE

Automated tests provide confidence in the stability of the codebase.

05

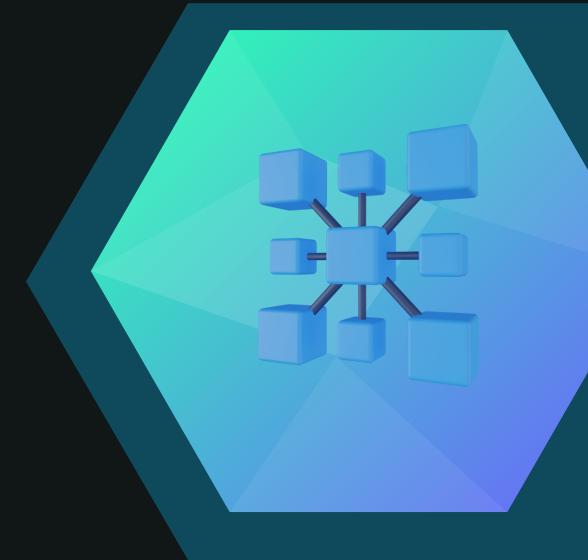
QUICK FEEDBACK

Developers receive immediate feedback on their changes.

Limitations & Conclusion

Despite its positive characteristics, continuous integration is often also shown to have some disadvantages in everyday working life.

While it poses challenges, its benefits, such as early bug detection, faster development, and improved collaboration, outweigh the drawbacks. The discipline and effort invested in CI significantly enhance the overall software quality and developer productivity, making it an indispensable practice in the software development lifecycle.



COMPLEXITY

Setting up CI infrastructure can be complex and time-consuming.



INTEGRATION CHALLENGES

Some projects with large codebases might face challenges in integrating changes smoothly.



REQUIRES DISCIPLINE

Developers need to follow best practices to ensure successful CI, which requires discipline and coordination.

Continuous Delivery and Continuous Deployment

- **CONTINUOUS DELIVERY**

Extends CI by automatically preparing code changes for release to production. It ensures that the codebase is always in a deployable state. However, the actual deployment to production is done manually.

- **CONTINUOUS DEPLOYMENT**

Continuous Deployment takes the concept further by automatically deploying code changes to production after successful CI and testing. It eliminates the need for manual intervention in the deployment process.

- **RELATION TO CONTINUOUS INTEGRATION**

Continuous Integration is the foundation for both Continuous Delivery and Continuous Deployment. CI ensures that code changes are continuously integrated, tested, and validated



Continuous Integration Practices



Continuous integration, delivery and deployment are software development practices born out of the DevOps movement. They make the process of building, testing and releasing code more efficient and get working product into the hands of users more quickly than traditional methods. Done well, a build pipeline enables teams to deliver working software at pace and get timely feedback on their latest changes.

- **AUTOMATED TESTING**

Automated testing ensures that new changes do not introduce bugs, providing fast and reliable feedback to developers, leading to higher confidence in the codebase.

- **VERSION CONTROL SYSTEM**

Using a version control system such as Git allows tracking changes, collaborating effectively, and reverting to previous states, ensuring code integrity and facilitating seamless collaboration among developers.

- **CONTINUOUS CODE INTEGRATION**

Frequent code integration leads to early bug detection, smoother collaboration, and reduced integration challenges, ensuring a stable codebase.

TESTING IN CONTINUOUS INTEGRATION



Involves running automated tests (unit tests, integration tests, etc.) whenever code changes are integrated. Testing ensures that new changes do not break existing functionality and maintains the overall integrity of the codebase. It provides immediate feedback to developers, allowing them to fix issues promptly, ensuring a reliable and stable software product.

CONTINUOUS INTEGRATION TOOLS ANALYSIS AND COMPARISON



JENKINS

Jenkins is an open-source automation server. It provides hundreds of plugins to support building, deploying, and automating any project.

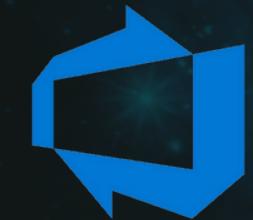
Also offers extensive plugin support and flexibility, making it suitable for a wide range of projects.



GITLAB

GitLab CI/CD is integrated into GitLab, a web-based Git repository manager. It provides a seamless experience for managing source code, CI/CD pipelines, and project collaboration.

*GitLab provides an integrated environment, allowing developers to manage code repositories and CI/CD pipelines in one place.



Azure DevOps

AZURE DEVOPS

Azure is Microsoft's cloud infrastructure platform, the Microsoft equivalent to Amazon Web Services.

Like the aforementioned AWS CodePipeline, Azure offers a CI Tool that fully integrated into the Azure suite of hosting tools

Azure DevOps is Microsoft's agile CI/CD solution with Azure integration, Windows support, container-friendly features, and GitHub compatibility.

Continuous Integration Tool Overview

JENKINS

1. Basic Tool Terminology:

- a. Jenkins jobs are individual tasks or processes that can be executed.
- b. Plugins extend its functionality, allowing integration with various tools and technologies.
- c. Jenkins pipelines define the entire build, test, and deployment process as code.

1. Main Features:

- a. They offer a vast array of plugins for integrating with different tools, enabling customization of CI workflows.
- b. The tool supports distributed builds, allowing workload distribution across multiple machines for faster build times.
- c. Jenkins supports defining pipelines using code, enabling version control and easy sharing of build configurations.
- d. Jenkins integrates with version control systems, build tools, and deployment platforms, providing a comprehensive CI solution.



RESOURCE REFERENCING.



WEBSITES USED

<https://www.jenkins.io/>

<https://about.gitlab.com/>

<https://azure.microsoft.com/en-us/products/devops>

<https://about.gitlab.com/solutions/continuous-integration/>

<https://www.ionos.com/digitalguide/websites/web-development/continuous-integration/>

<https://www.jetbrains.com/teamcity/ci-ci-guide/ci-ci-best-practices/>

<https://www.atlassian.com/continuous-delivery/continuous-integration/tools>

<https://blog.jetbrains.com/teamcity/2023/07/best-ci-tools/>

THANK YOU

IT6039 SOFTWARE TESTING AND MAINTENANCE

STUDENT ID 20220676

GUILLERMO OZAN