

CICD VS. "Classic," way of deploying

The background features a large, circular diagram representing the Modern Software Development Cycle (CICD). The cycle is composed of five main stages, each represented by a chevron-shaped segment: 'Build' (with a wrench and screwdriver icon), 'Test' (with a gear and circular arrow icon), 'Deploy' (with a server rack icon), 'Monitor' (with a magnifying glass over a document icon), and 'Operate' (with a person at a laptop icon). The stages are connected by a continuous loop of arrows, indicating a continuous process. The entire diagram is rendered in a light blue color against a darker blue background.

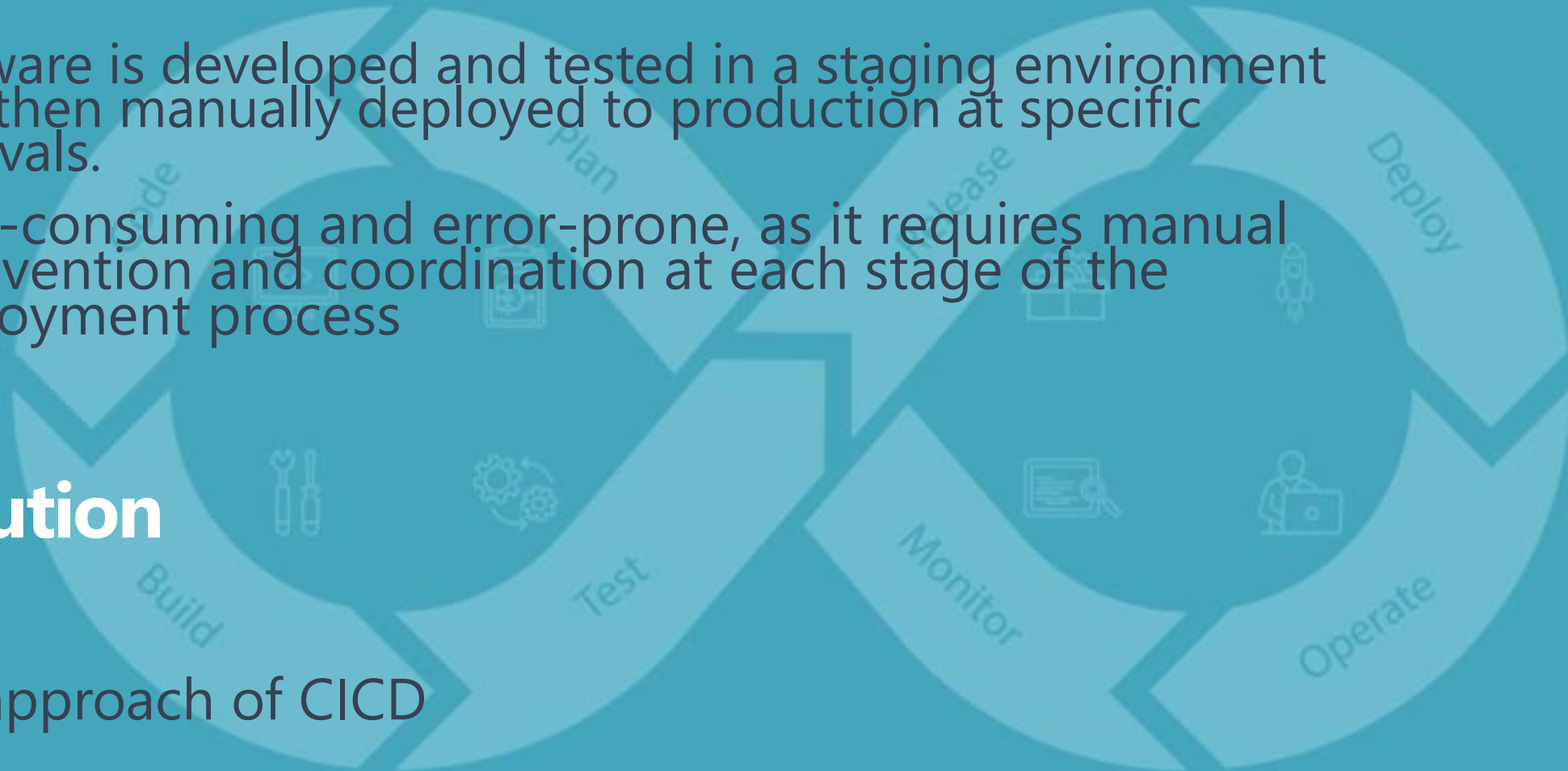
By : Mahmoud Gawish

bottlenecks

- software is developed and tested in a staging environment and then manually deployed to production at specific intervals.
- time-consuming and error-prone, as it requires manual intervention and coordination at each stage of the deployment process

Solution

the approach of CICD



Benefits

CICD involves integrating code changes into a central repository and automatically building, testing, and deploying the code to production.

CICD allows for a faster and more efficient deployment process compared to traditional approaches.

CICD offers more granular and efficient testing, as every code change is tested before being deployed to production.

CICD helps to catch errors and bugs earlier in the development process.

Continuous integration (CI):

- CI is a software development practice in which developers frequently commit changes to a central repository, and automated builds and tests are run on the updated codebase.
- The goal of CI is to detect errors and bugs early in the development process and to allow developers to work in a more collaborative and agile manner.

Build

Test

Analyze

Continuous delivery (CD):

- CD is a software engineering approach in which code changes are automatically built, tested, and deployed to production.
- The goal of CD is to increase the speed and frequency of software releases and to reduce the risk of errors and downtime in production.

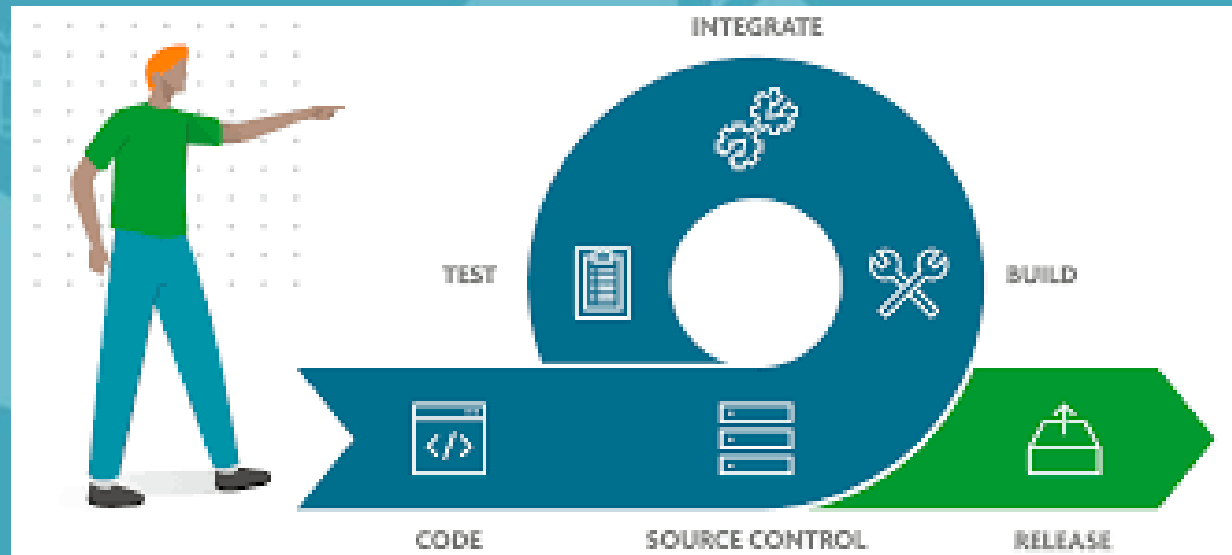
Deploy

Verify

Promote

Principles of Continuous Integration

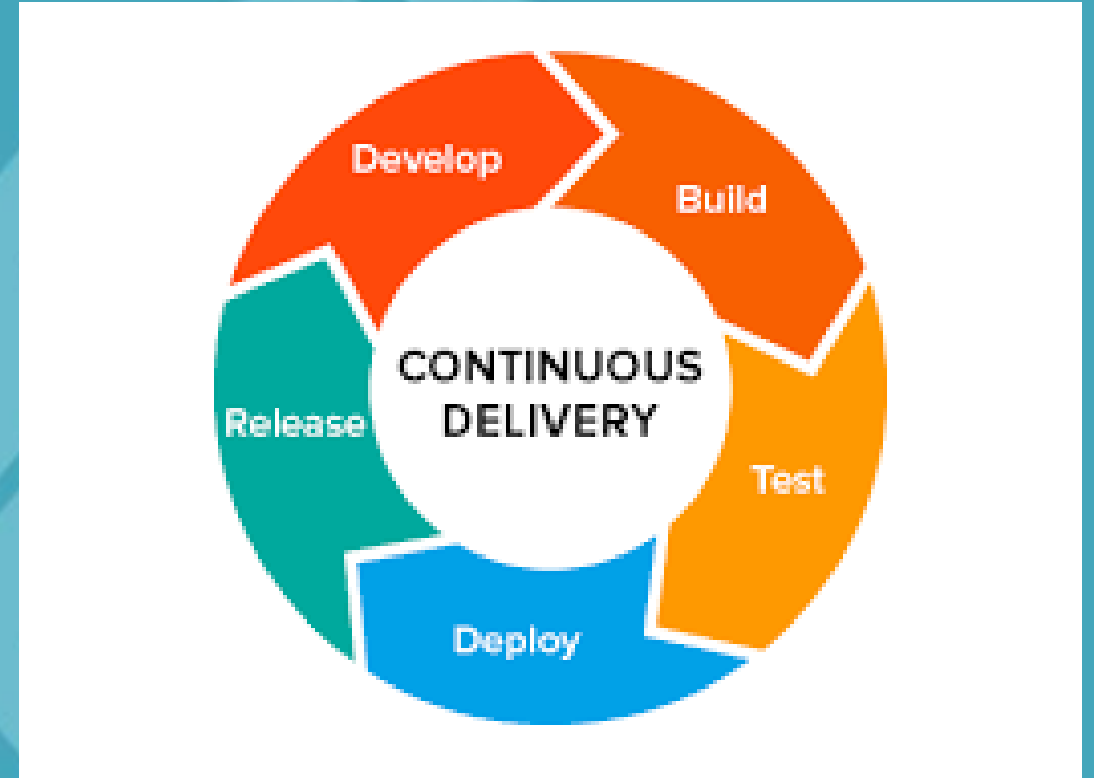
- Run Unit / integration tests
- Compile/ lint code
- Static Code Analysis
- Merge code from all sources(branches and commits)
- Check libraries and dependencies versions



The Modern Software Development Cycle

Principles of Continuous Delivery

- Repeatable Reliable Process
- Automate Everything
- Version Control Everything
- Bring the Pain Forward
- Build-in Quality
- "Done" Means Released
- Everyone is Responsible
- Continuous Improvement



continuous integration:

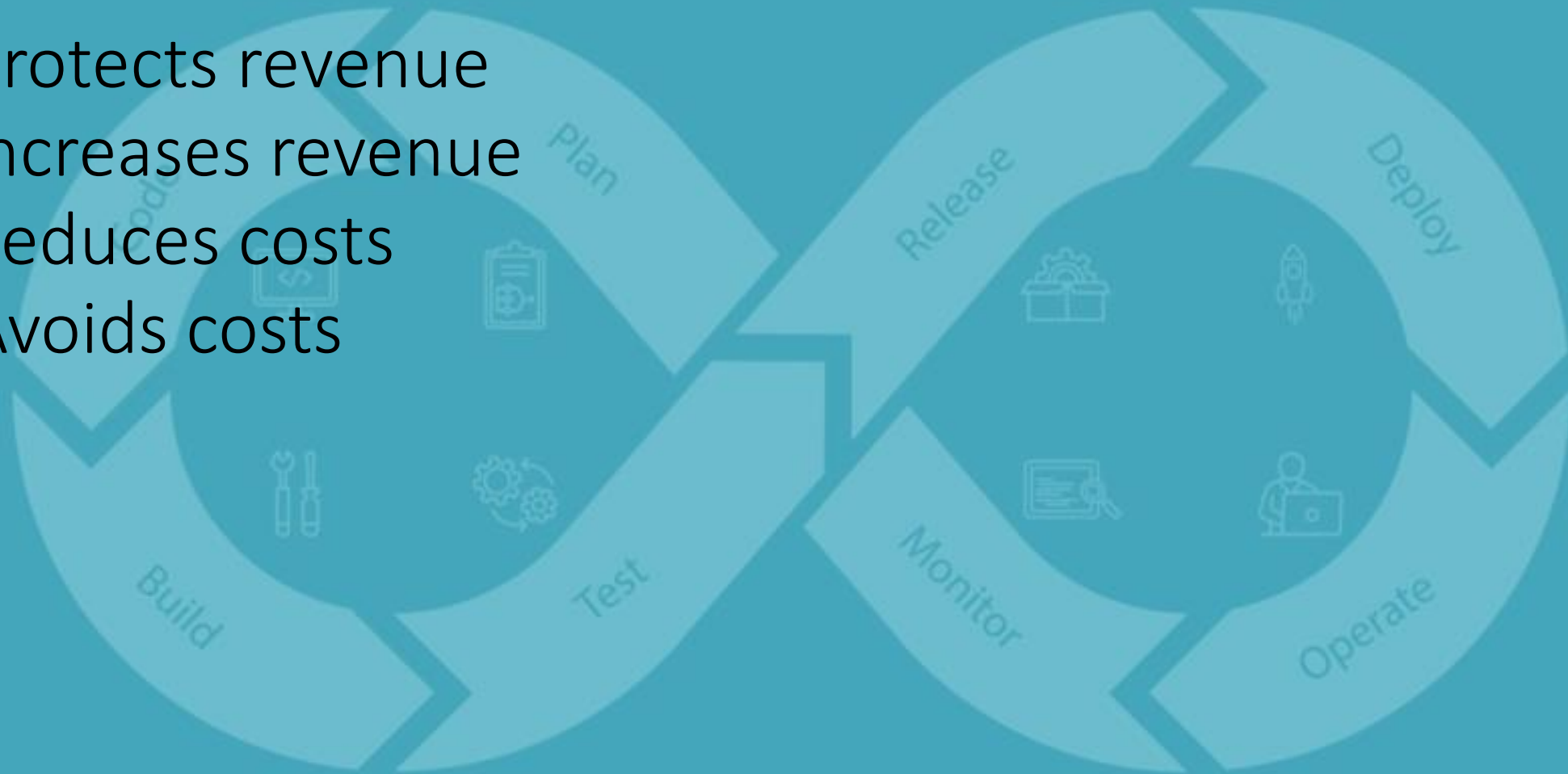
- Helps to detect errors and bugs early in the development process
- Allows developers to work in a more collaborative and agile manner
- Improves the overall quality and reliability of the software
- Reduces the time and effort required to release new features

continuous delivery:

- Increases the speed and frequency of software releases
- Reduces the risk of errors and downtime in production
- Allows for more flexible and agile response to changing business needs
- Increases collaboration between development and operations teams

Benefits of CI/CD

- . A. Protects revenue
- . B. Increases revenue
- . C. Reduces costs
- . D. Avoids costs



The Modern Software Development Cycle

A. Protects revenue

- Allows for quicker identification and resolution of issues
- Reduces downtime and improves availability of software
- Improves overall quality and user experience of software, leading to increased customer satisfaction and loyalty

B. Increases revenue

- Allows for faster delivery of new features and updates
- Enables more frequent and efficient releases, allowing for more opportunities to monetize new capabilities
- Facilitates experimentation and innovation through easier testing and deployment of new ideas

c. Reduces costs

- Automates manual and time-consuming tasks, freeing up development resources
- Reduces the need for emergency hotfixes and costly rollbacks
- Facilitates faster recovery from issues, reducing lost revenue

D. Avoids costs

- Prevents costly errors and mistakes by catching issues earlier in the development process
- Reduces the need for expensive emergency deployments and rollbacks
- Improves overall efficiency and speed of the development process, reducing the need for additional resources

Conclusion

CI/CD has the potential to significantly impact the revenue and cost of a business through improved software quality, faster delivery, and increased efficiency.

