ONE PAGE SUMMARY

IT & Software › DevOps › Continuous Delivery

Introduction to Continuous Delivery

Streamline software updates with Continuous Delivery.

DEFINITION

> Continuous Delivery is a software development practice that enables frequent, reliable releases. It ensures that code changes are automatically prepared for production deployment.

KEY CONCEPTS

Automated Testing

Automated tests verify code changes quickly, ensuring they work as intended before deployment.

Deployment Pipeline

A series of automated processes that code changes go through from development to production.

Version Control

A system that tracks changes to code, allowing teams to collaborate and manage revisions effectively.

Release Readiness

The state of code being always ready for deployment, minimizing the risk of failures.

Feedback Loops

Continuous feedback from users and automated systems helps improve the software rapidly.

EXAMPLES

→ Deploying a new feature to production every week.

→ Running automated tests after each code commit.

→ Using version control to manage team contributions.

MEMORY TIPS

★ Think of 'CD' as 'Code Delivered' for easy recall.

★ Remember 'Pipeline' as a flow of changes moving smoothly to production.

★ Associate 'Automated Testing' with 'Safety Net' for code.

COMMON MISTAKES

✗ Neglecting to write sufficient automated tests.

✗ Skipping stages in the deployment pipeline.

✗ Failing to integrate feedback from users.

QUICK RECAP

Continuous Delivery allows for frequent and reliable software releases through automation. Key elements include automated testing, deployment pipelines, and version control to ensure code is always ready for production.