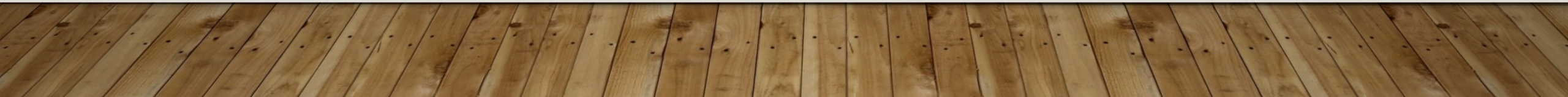
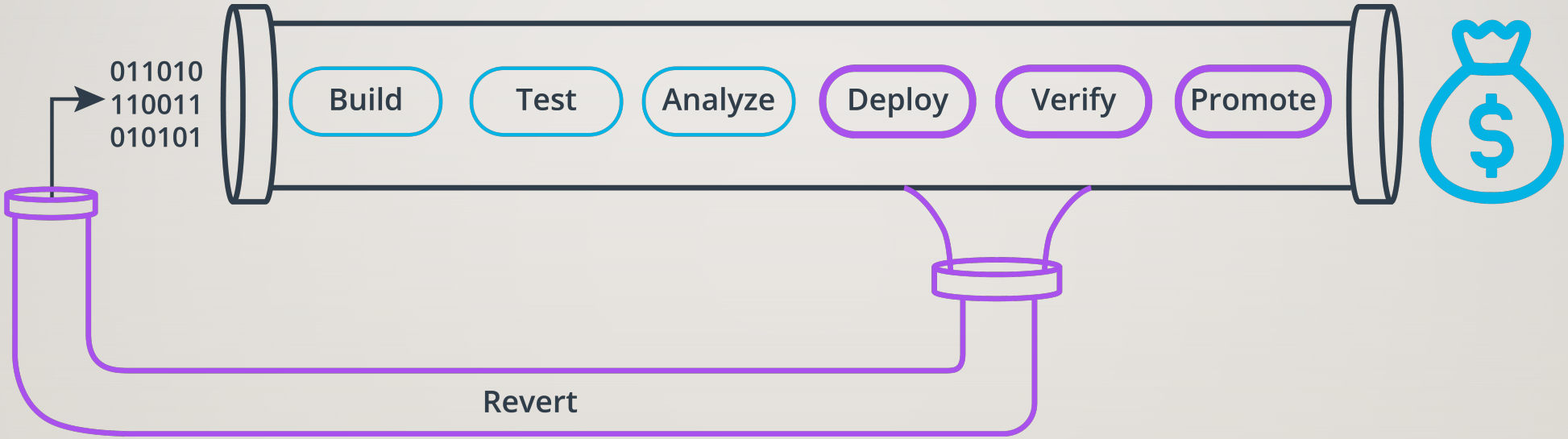
CI/CD AS A MEANS TO DRIVE BUSINESS VALUE

HOW CONTINUOUS INTEGRATION & CONTINUOUS DEPLOYMENT SAVES MONEY AND DELIVERS VALUE



CONTINUOUS DELIVERY = CONTINUOUS INTEGRATION + CONTINUOUS DEPLOYMENT



- An automated process to deploy and verify the stability of code, infrastructure, and the overall application performance.
- Allows engineers to focus on providing smaller chunks of value to clients at a quick pace by reducing the time spent by engineers manually error checking, monitoring, deploying the product.
- Less engineering time reduces costs and increase productivity!

CONTINUOUSLY DELIVERY (CD) PRINCIPLES

CD Principles	Business Value
Repeatable Reliable Process	Increases predictability, which facilitates planning and meeting goals, and ultimately controls costs
Automate Everything	Reduces time to delivery to clients. Improves customer satisfaction, which protects revenue.
Version Control Everything	Enables the ability to revert to a previous working application in case of errors. Reduces downtime and saves money.
Bring the Pain Forward	Solutions to pain points increases efficiency, which results in smoother and quicker time to delivery. This allows the team to control costs.
Build-in Quality	Each step includes quality checks and ways to recover if errors occur. Results in less down time of the application, which controls costs.
"Done" Means Released	Clear definition of success makes planning and meeting goals easier. This predictabiligy controls costs.
Everyone is Responsible	Encourages practices that reduce errors, which ultimately reduces costs.
Continuous Improvement	Frequent value is being delivered to clients. Increases customer satisfaction, which protects revenue.

CONTINUOUS INTEGRATION

- What: The practice of merging developers' code to a shared code repository several times a day
- Tools:
  - Code Repository & Versioning Tool: Git, Gitlab, Github (Recommended)
- Requirements:
  - Clear feature specifications that focus on customer value
  - Features are defined in smaller chunks
  - Unit Tests and Code Style Guide to insure consistency and
- Impact: Standardizes code and identify errors early in the process that can be quickly fixed by the person who introduced them. Improves products and reduces time spent on errors, which controls costs and protects revenue.

CONTINUOUS DEPLOYMENT (CD)

- **What**: A software engineering approach in which the value is delivered frequently through automated deployments.
- **Tools**:
  - IAC: CloudFormation (recommended), Terraform
  - Monitoring/Logging – DataDog, CloudWatch
  - Alerts: Prometheus (recommended), CloudWatch
- **Requirements**:
  - Determine infrastructure needed to run and consistently deploy infrastructure. Deploy infrastructure through code.
  - Determine a deployment strategy (i.e., Canary, Blue-Green Candidates)
    - Avoids downtime of the application
- **Impact**: Repeatable process that avoids downtown time of the application. Controls costs and protects revenue.

SUMMARY

- Implementing CI/CD drives business value while reducing costs.
- It provides consistency around development and deployment which significantly controls costs.
- Allows engineers to focus on designing features that provide client value, rather than dealing with errors.
- Support team collaboration, which increases client retention, which also support controlling costs by reducing turnover.