CI/CD ADOPTION

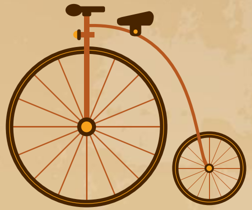
A Look at the Economics

COMMON CHALLENGES IN SOFTWARE PROJECTS

Software project **failures are expensive** and unfortunately common, the causes are varied, but amongst the most common are:

1. Unrealistic or unarticulated project goals
2. Poor reporting of the project's status
3. Unmanaged risks
4. Sloppy development practices
5. Poor coordination between developers on tasks dependencies
6. Poor communication among customers, developers, and users

HOW CAN CI/CD HELP US SUCCEED?



AVOID COSTS

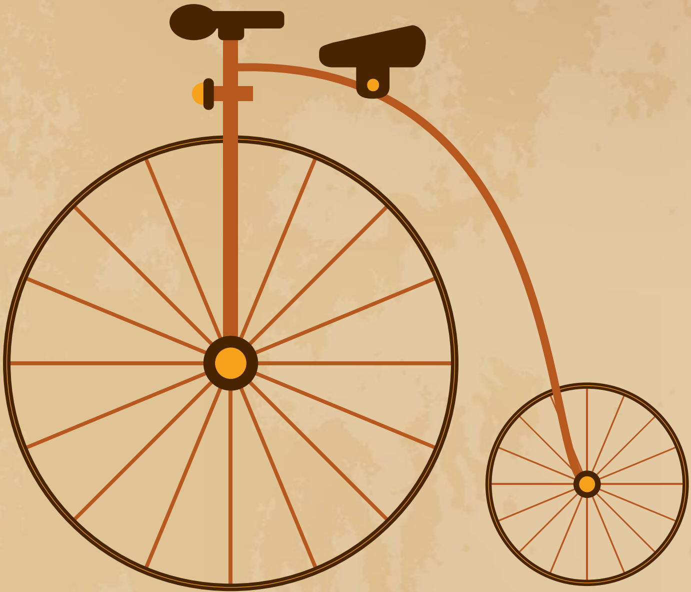
Less bugs
Reduce conflicts between developers

REDUCE COSTS

Pay only for Infra used
Less developer time spend on toil tasks

PROTECT OR INCREASE REVENUE

Less time to market
Less unplanned downtime
Easy rollback to previous working version



AVOID COSTS

A CI/CD automates the development cycle's best practices, aligning the incentives to the desired behaviors.
By not allowing direct push of code to the release branch, we ensure development branches meet a minimal quality level before considering them as release candidate code.
CI/CD ensures clear responsibilities, and developers won't waste hours or even days because of test failure in the main branch.

We could go one step forward and reject the code automatically if, upon merging it, the release candidate branch had any test failures.

This technology-supported process reduces conflicts between team members since it is clear and unappealable.
Only clean code will make it to the release, and a regular release schedule eliminates the rush to push features that are not yet ready.

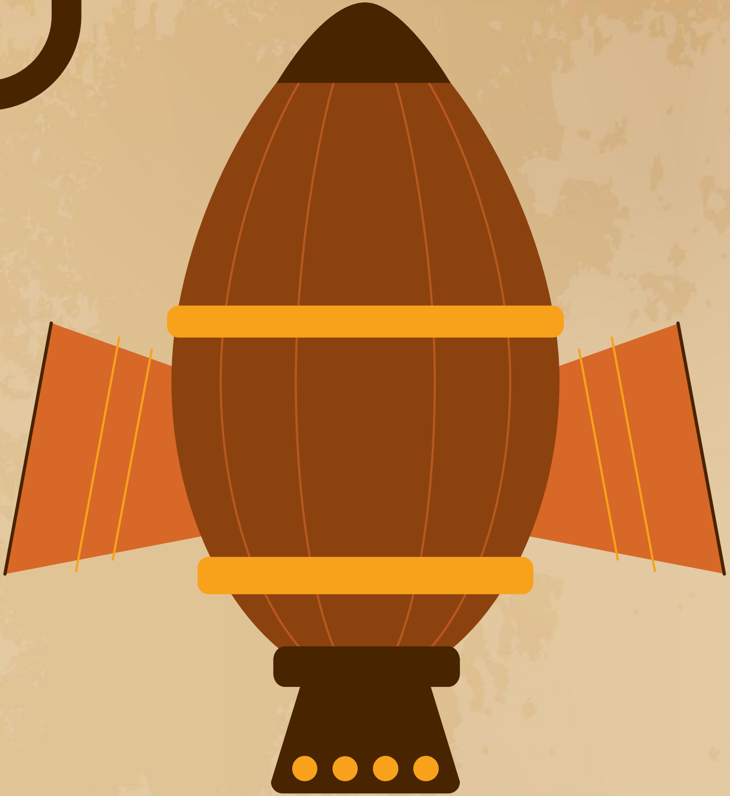
REDUCE COSTS

Using Circle CI and AWS to support the CI/CD infrastructure will help us optimize the budget. Instead of dedicated build machines with administration and maintenance needs, we will use the Just In Time capabilities to deploy the resources.

Ensuring the developers' needs are met, they won't waste time waiting for build machines to be free. And the business won't spend money buying and maintaining oversized development infrastructure.

Since the workflow is automated, developers will be enabled to spend time writing quality code to meet the business requirements, instead of setting up development environments or manually running build, integration, or deployment tasks.

Building and running tests on each code push ensure errors are caught earlier on smaller changesets, allowing an easy resolution and a lot less time investigating bugs on more significant combined patches on the release candidate branch.



PROTECT OR INCREASE REVENUE

Smaller and more frequent releases mean new features will start having a return on investment faster. They will also enable the business to react quicker to the competition or customer wishes.

Easy and fast rollback to the previous functioning version protects the brand name and value, by minimizing the impact of production bugs in customer satisfaction.

Making sure bug reports are resolved only with the addition of new tests will ensure production errors happen only once, driving the code base's continuous improvement.

Sales would get a clear view of new features and possible deployment dates, allowing them to set realistic customer expectations, preventing user frustration and attrition waiting for the next big release.

THANKS!

Do you have any questions?

youremail@freepik.com

+91 620 421 838

yourcompany.com

CREDITS: This presentation template was created by **Slidesgo,** including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

Please keep this slide for attribution