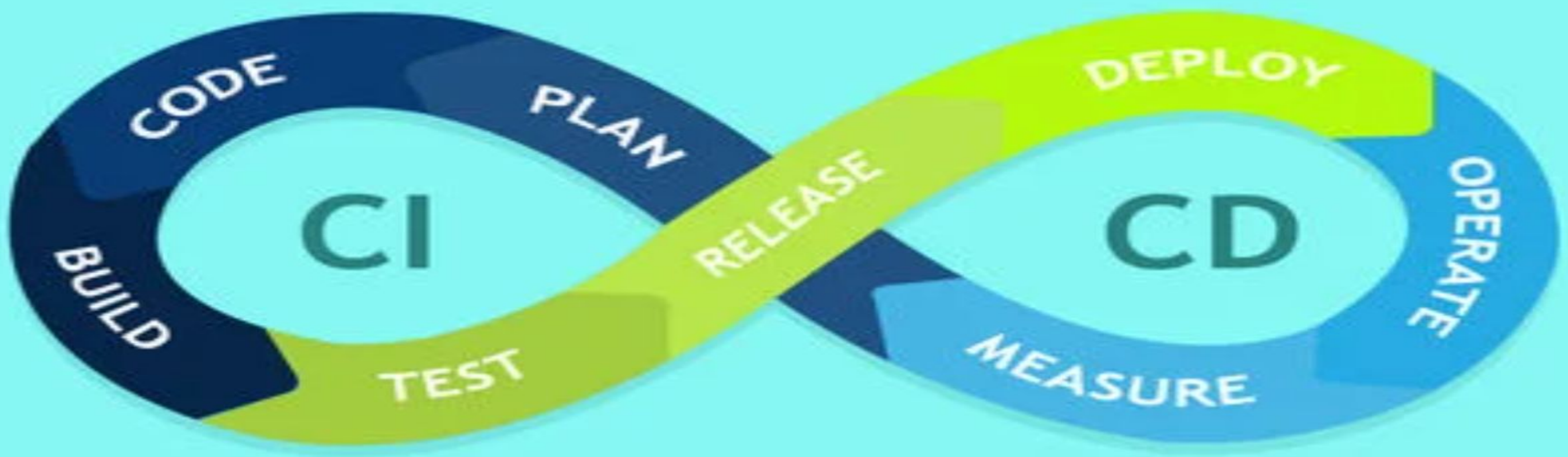
CI/CD

How does it generate revenue and keep developers happy?

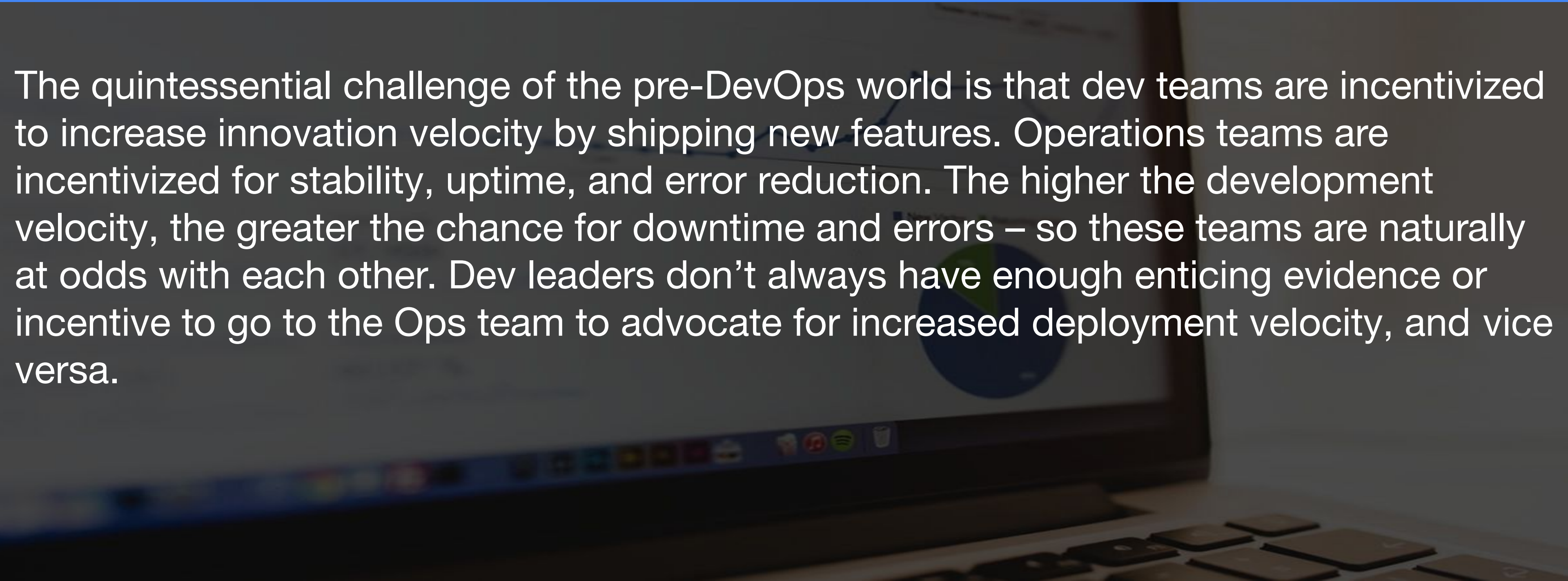
Continuous Integration is the practice of integrating code into a shared repository and building/testing each change automatically, as early as possible - usually several times a day.

Continuous Delivery adds that the software can be released to production at any time, often by automatically pushing changes to a staging system.



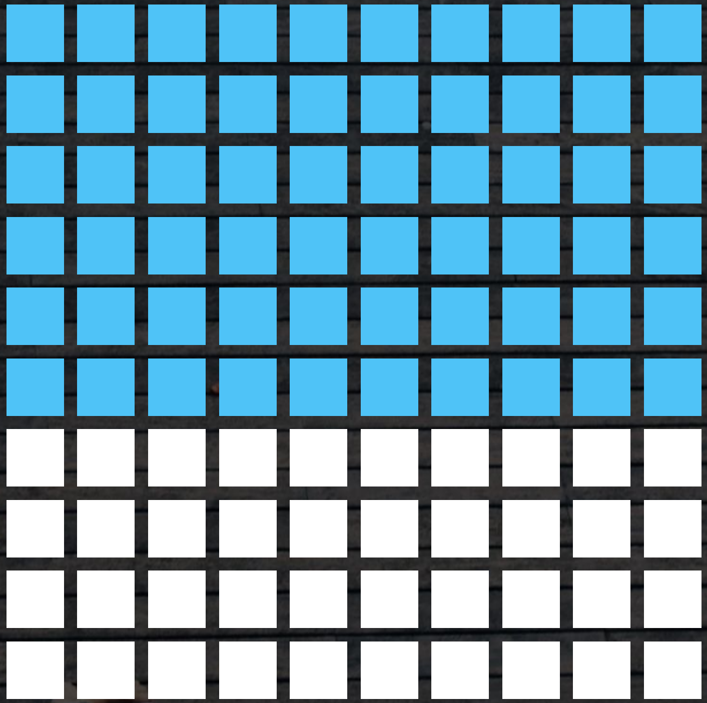
Continuous Deployment goes further and pushes changes to production automatically.

Problems with bad or non-existent CI/CD



The quintessential challenge of the pre-DevOps world is that dev teams are incentivized to increase innovation velocity by shipping new features. Operations teams are incentivized for stability, uptime, and error reduction. The higher the development velocity, the greater the chance for downtime and errors – so these teams are naturally at odds with each other. Dev leaders don’t always have enough enticing evidence or incentive to go to the Ops team to advocate for increased deployment velocity, and vice versa.

A large percentage of the overall IT budget goes to support teams of engineers needed to integrate and maintain a complex toolchain. An enterprise company with 1,000 developers could need up to 40 engineers just to maintain the DevOps toolchain instead of allocating these resources towards delivering business value.



The solution: CI/CD

More Automation

Increased speed of innovation and better ability to compete in the marketplace are setting the pace for everyone else.

More Productivity

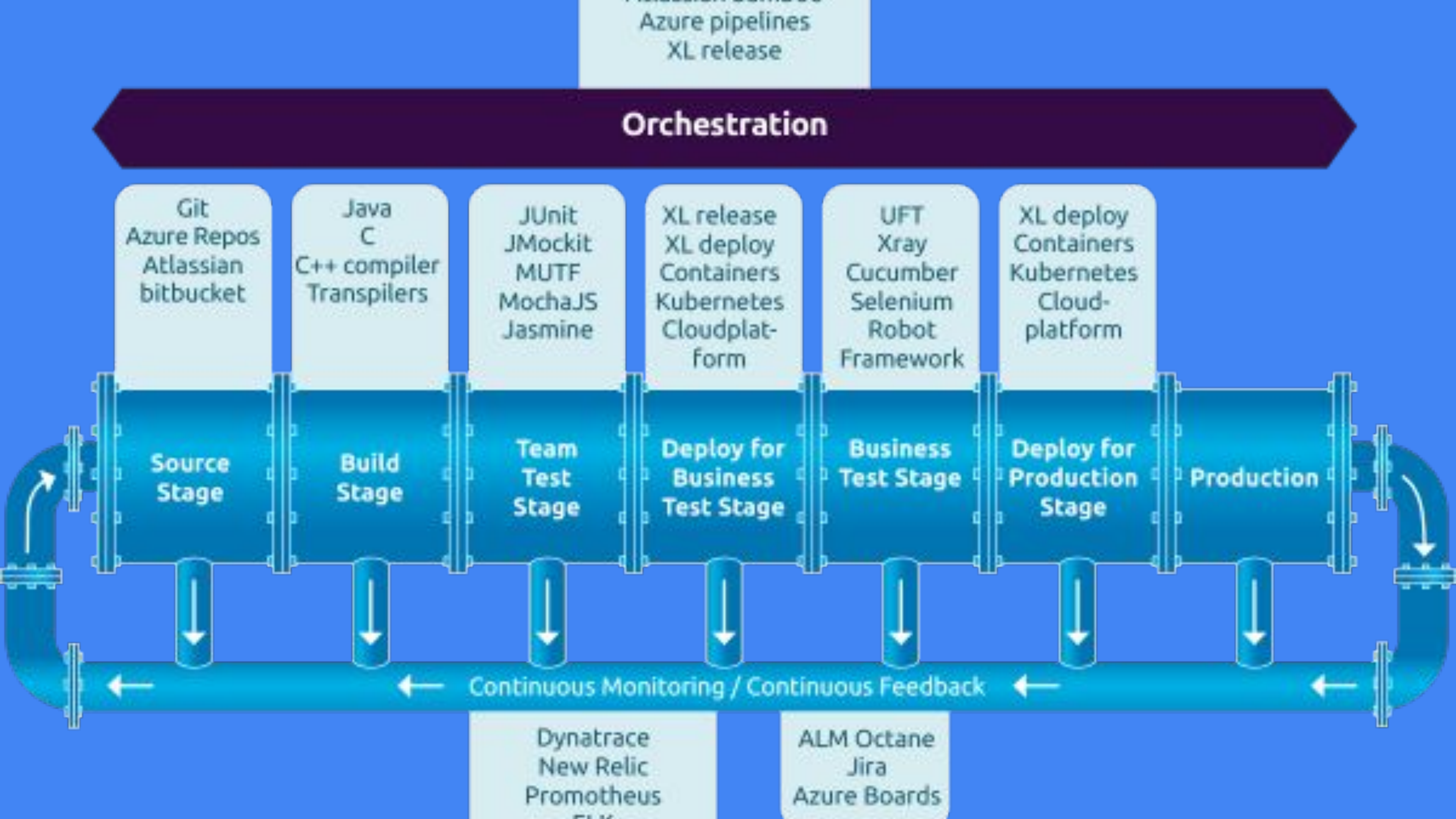
Code in production is making money instead of sitting in a queue waiting for manual checks to be deployed.Human error and delays have been eliminated from the process.

Lesser Turnover

Great ability to attract and retain talent as engineers can work on important projects and not worry about time-consuming manual tasks.

Focus on Strengths

Higher quality code and operations due to specialization.Developers can focus on the code , and operations doesn’t have to feel like a gatekeeper or a barrier



Atlassian bamboo
Azure pipelines
XL release

Orchestration

Git
Azure Repos
Atlassian bitbucket

Java
C
C++ compiler
Transpilers

JUnit
JMockit
MUTF
MochaJS
Jasmine

XL release
XL deploy
Containers
Kubernetes
Cloudplatform

UFT
Xray
Cucumber
Selenium
Robot Framework

XL deploy
Containers
Kubernetes
Cloud-platform

Source Stage → Build Stage → Team Test Stage → Deploy for Business Test Stage → Business Test Stage → Deploy for Production Stage → Production

← Continuous Monitoring / Continuous Feedback ←

Dynatrace
New Relic
Promotheus
ELK

ALM Octane
Jira
Azure Boards