Let's look at five benefits of implementing a CI/CD pipeline to understand why many organizations have shifted toward this approach.

1. Reduce risk

Finding and fixing bugs late in the development process is expensive and time-consuming. This is especially true when there are issues with features that have already been released to production.

With a CI/CD pipeline, you can test and deploy code more frequently, giving testers the ability to detect issues as soon as they occur and to fix them immediately. You are essentially mitigating risks in real time.

2. Deliver faster

Organizations are moving toward releasing features multiple times a day. This is not an easy task; only a handful of companies like Netflix, Amazon, and Facebook have been able to achieve this goal. But, with a seamless CI/CD pipeline, multiple daily releases can be made a reality.

Teams can build, test and deploy features automatically with almost no manual intervention. This is accomplished using various tools, frameworks, and systems like Travis CI, Docker, Kubernetes, and LaunchDarkly.

3. Expend less manual effort

To align with the shift-left paradigm, we need automation right from the start. This is also a vital component of having a successful CI/CD implementation. Once you build features and check in code, tests should be automatically triggered to make sure that the new code does not break existing features and that the new features are working correctly.

After the tests run, the code gets deployed to different environments, including QA, staging and production. Throughout this process, you will be getting constant notifications through different channels, giving you plenty of information about the build, test and deploy cycles.

4. Generate extensive logs

Observability is one of the biggest aspects of DevOps and CI/CD integration. If something is wrong, you need to understand why. You need a mechanism to study the system in production over time and identify key performance metrics. Observability is a technical solution that helps in this effort.

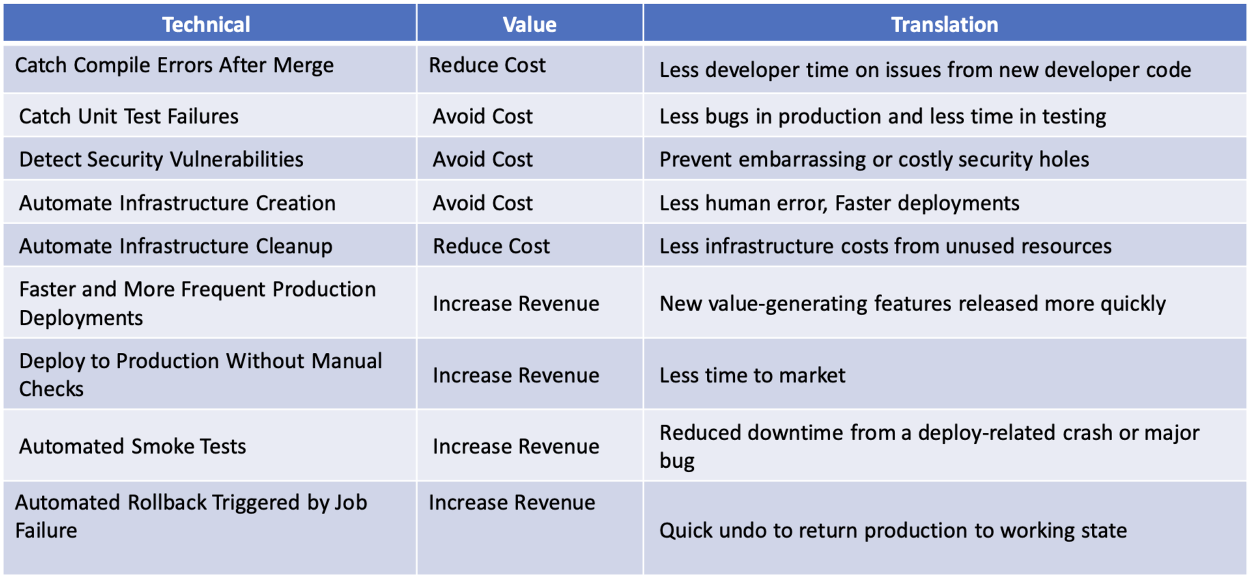
One key aspect of observability is logging information. Logs are a rich source of information to understand what is happening beneath the UI and study application behavior.

With a CI/CD pipeline, extensive logging information is generated in each stage of the development process. There are various tools available to analyze these logs effectively and get immediate feedback about the system.

5. Make easier rollbacks

One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly. If any new code changes break the production application, you can immediately return the application to its previous state. Usually, the last successful build gets immediately deployed to prevent production outages.

The world is moving toward rapid release cycles, and CI/CD pipelines have accelerated the release rate. With careful planning and implementation, such a pipeline can help you find defects faster, implement fixes immediately, and increase overall customer satisfaction.



Technical	Value	Translation
Catch Compile Errors After Merge	Reduce Cost	Less developer time on issues from new developer code
Catch Unit Test Failures	Avoid Cost	Less bugs in production and less time in testing
Detect Security Vulnerabilities	Avoid Cost	Prevent embarrassing or costly security holes
Automate Infrastructure Creation	Avoid Cost	Less human error, Faster deployments
Automate Infrastructure Cleanup	Reduce Cost	Less infrastructure costs from unused resources
Faster and More Frequent Production Deployments	Increase Revenue	New value-generating features released more quickly
Deploy to Production Without Manual Checks	Increase Revenue	Less time to market
Automated Smoke Tests	Increase Revenue	Reduced downtime from a deploy-related crash or major bug
Automated Rollback Triggered by Job Failure	Increase Revenue	Quick undo to return production to working state