Software Reliability & CI/CD

Software Reliability

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 processes
5. Poor coordination between developers and operations
6. Lack of understanding for customer requirements

How CI/CD assists with software reliability

1. Avoid costs: Less issues and downtimes avoid additional development
2. Reduces costs: Keep your hosting costs manageable with fully automated and efficient processes
3. Protect or Increase revenue: Keep services running to keep business operating, also improves reputation which increases customer retention and trust.

How does it avoid and reduce cost?

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.

How does it 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.

Thank you!
@AbdulazizGH