The Old School !!

- Large multi-changes old-school scheduled deployment had always been the only way to deploy new application features and changes to customer platforms and portals.
- Though an experienced operation team and well-organized process would smooth the deployments to an extent, the old-school way has always been a messy activity to handle, with manual process leading to unexpected behaviors and errors, in itself causing major setbacks as: Lost cost, degraded customer satisfaction and affected company profile.
- A big gap between the development team and operation team, who though having same goal, yet different area of expertise and responsibility.

Setbacks

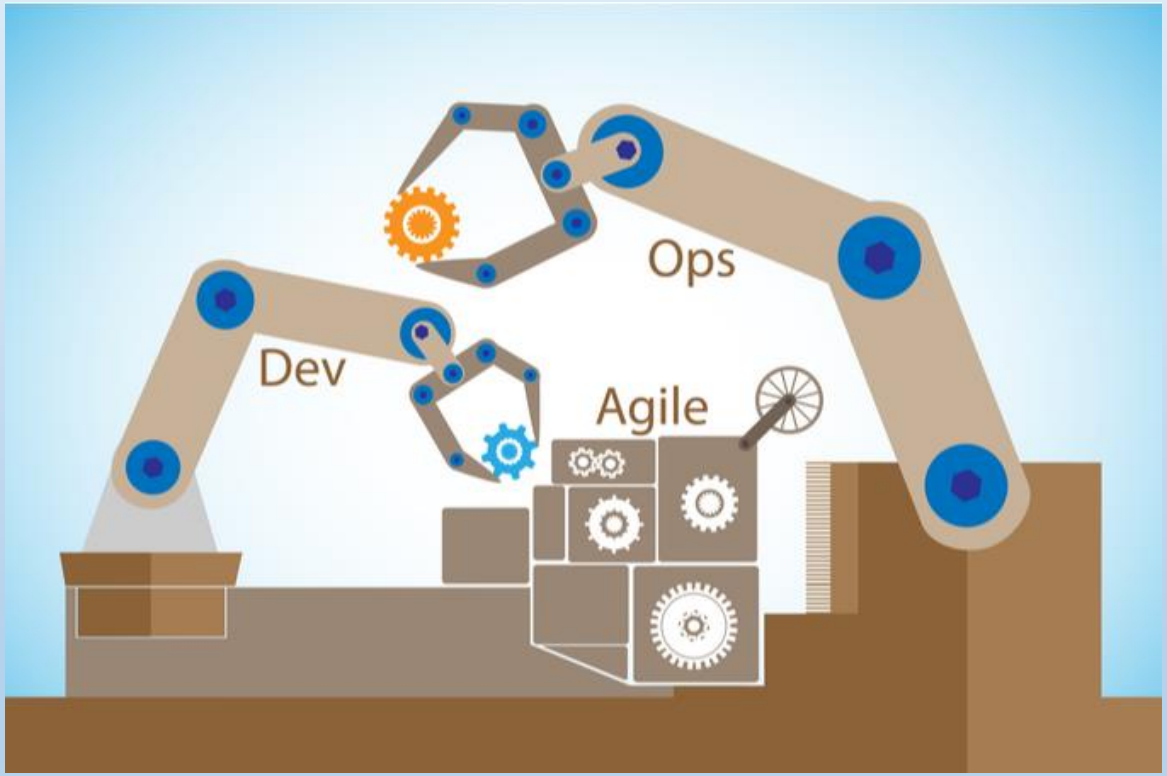
- A new feature would always find its way to the production environment, but the time it takes and the way it get deployed has its own costs and effects
    1. New deployment require downtime, so increasing time of deployment increases as well duration of unavailability of the application, leading to probable contractual penalties
    2. Delayed deployment due to unexpected errors, leads to broken deadlines, huge customer dissatisfaction, and unfulfilled contract features.
    3. Lengthy and complicated deployments leads to integration issues between application different features and unexpected behavior and errors, which requires more involvement from the operation and development team, leading to more labor time, and less concentration on developing and adding values to existing solutions.

The Question ..?

- Operation experts have concluded that the answer for such complication lies in resolving 2 area of conflicts:

    1. Creating an agile environment where daily repetitive small changes are tested, verified, and deployed on different environment before reach production, avoiding known complications linked with big long-planned deployments.
    2. Creating an automatic process for deployments and working towards reaching the paradigm of removing any manual human intervention in the whole process, leading to persistent automated behavior for every deployment.

The Saviour



- The answer for all our questions is: CI/CD pipelines [a.k.a Continuous integration / Continuous deployment].
- A CI/CD pipeline helps us to reach the paradigm of automating the whole process of the deployment and the transformation of the developers code to a business value.
- CI/CD is the result of a collaborating community called devops, which long to remove the gap mentioned between the development and operation team, consisted of wide various stack of technologies integrated together to help us reach our goal.
- The CI part is responsible for automating all code-related needed action for deployment [Compiling – Testing – Find security issues – Conversion of code to machine binaries]
- The CD part integrate with the CI, taking its automated outcome and perform the responsibility of provision underlying infrastructure and the part of application deployment

CI/CD effect

- Having a full automated CI/CD pipeline implemented helps towards the following:
    1. Small Iterative deployments with exact consistent behavior which reduce the time of unavailability of application during deployments, as well meeting pre-agreed deadline, which lead to avoiding possible penalties and better customer satisfaction.
    2. A development team who can focus all his attention on adding value on the present solution instead of burning-out in unimportant activities.
    3. Clean unstressful deployment which lead to a healthier and more productive environment.
    4. Better company profile and customer satisfaction due to gained consequences of smooth deployments.