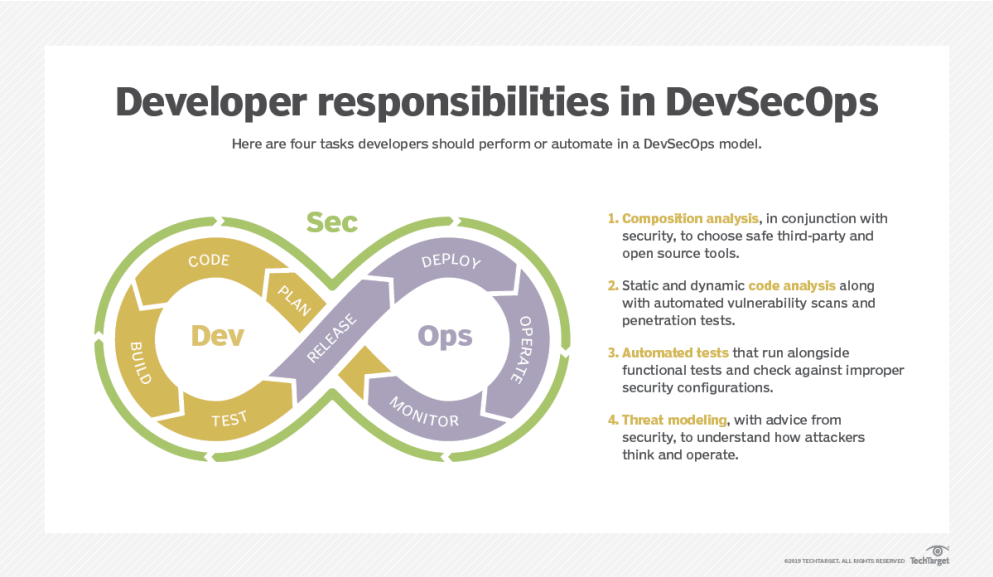
- What is DevOps?

The word *DevOps* is a combination of the terms *development* and *operations*, meant to represent a collaborative or shared approach to the tasks performed by a company's application development and IT operations teams, In its broadest meaning, DevOps is a philosophy that promotes better communication and collaboration between these teams -- and others -- in an organization. In its most narrow interpretation, DevOps describes the adoption of iterative software development, automation, and programmable infrastructure deployment and maintenance.

- What problems does DevOps solve?

Each company faces its own challenges, but common problems include releases that take too long, software that doesn't meet expectations and IT that limits business growth, DevOps solves communication and priority problems between IT specializations. To build viable software, development teams must understand the production environment and test their code in realistic conditions. A traditional structure puts development and operations teams in silos.



- What is a pipeline?

Before we talk about CI or CD, it's best to first understand what a pipeline is. Effectively, a pipeline is a group of events or jobs that are connected together in a sequence. But it's a bit more complicated than that because each sequence (or stage) is dependent upon the output of the previous stage. So:

Stage 1 > Output > Stage 2 > Output > Stage 3 > Output

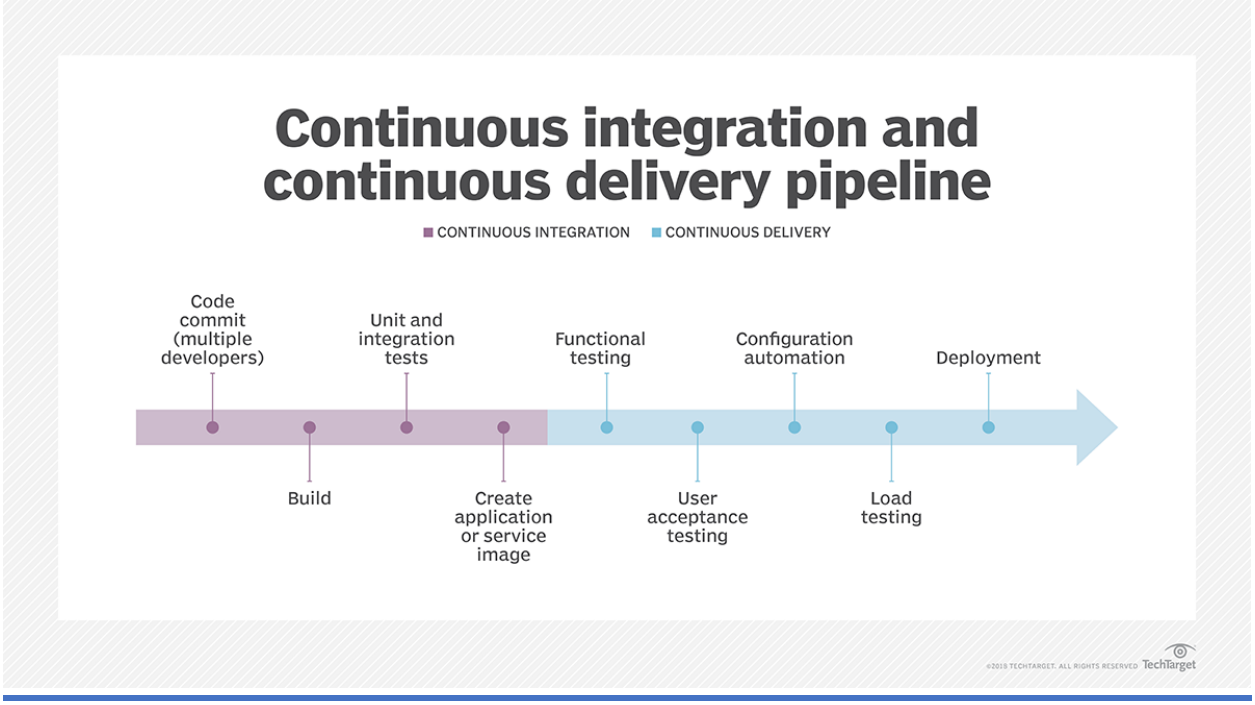
If the output of any stage fails, the next stage will also fail. The CI/CD pipeline is all about automation: Initiating code builds, automated testing, and automated deploying to the staging or production environments. It's a complex, but incredibly fast pipeline.

- What is Continuous Integration (CI)?

Continuous Integration is the process that allows developers to integrate new code into a shared repository (such as GitHub) throughout the day. That continuous submission of commits enables the system as a whole (typically by means of automation) to detect any integration bugs at the initial stage of committing so issues can be resolved immediately.

- What is Continuous Delivery (CD)?

Continuous Delivery (CD) occurs at the end of the CI cycle and is responsible for the automated delivery of the integrated code from the development to the production stage. CD is not only tasked with the automated delivery of the integrated code, but ensuring the delivered code is without bugs or delays.



- What are the benefits of CI/CD?

1- Faster time to market

The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently. Tech giants may have led the way, adopting Agile and DevOps techniques to transform their development processes and deliver constant improvements to their users, but with many smaller organizations following suit the landscape is becoming increasingly competitive.

2- Reduced risk

Having a shorter time to market doesn't just help you keep up with the competition. Rapid releases provide an opportunity for product managers and marketing professionals to engage more closely with the development process.

3- Better code quality

Testing your code's behavior is an essential step in the software release process but doing it thoroughly can also be extremely time consuming. A central part of any CI/CD pipeline is a series of automated tests that are run on each and every build. Although writing automated tests requires an investment of time and expertise, doing so pays significant dividends.

4- Smoother path to production

As we all know, practice makes perfect, and what's true of shooting hoops or mastering scales also applies to software releases. Adopting CI/CD is best done incrementally, starting with CI practices and building up your pipeline over time. As you start deploying changes more frequently you'll identify pain points and steps in your current process that slow you down,

5- Measurable progress

Many of the tools available to support an automated CI/CD pipeline also instrument the process, providing you with a whole host of metrics from build times to test coverage, defect rates to test fix times. Armed with this data you can identify areas that might need attention so you can keep improving your pipeline.

6- roll back changes quickly

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.

- Is CI CD good?

A well-built CI/CD process makes software development easier, faster and safer, which means DevOps teams have the time and energy to think outside the box. Attract and retain talent, humans aren't great at thinking long-term; So, it's not surprising that we favor short-term wins and churning out features as fast as we can.

But in order to maintain this pace, we need to slow down and fix our process. And I realize that sounds counter-intuitive, but it's necessary if we ever want to get out of technical debt and back into what we all want to be doing: building software that customers love.

- COST REDUCTION :

As per one of the recent Forbes Insights surveys, "Three out of four executives agree that the amount of time, money, and resources spent on ongoing maintenance and management—versus new project development or new initiatives—is affecting the overall competitiveness of their organization."
Delivering high quality tailor-made solutions to address business specific challenges requires a way to meet the rapid time constraints imposed by rivals. Embracing CI/CD is the perfect fix to shorten the time to finish a project and market new features. The shorter the development cycle, the higher the chances are to meet ambitious time-to-market goals

The average cost of data center downtime is $5,600 per minute, and the average downtime is 90 minutes, according to a Ponemon Institute study, so that's more than half a million dollars per incident.

Go back to your root cause analysis data and work out how many of your production incidents were caused by provisioning errors. You can make the calculation to find the potential revenue losses from your own business with a simple formula, as outlined in this Evolven blog.

LOST REVENUE = (GR/TH) x I x H GR = gross yearly revenue TH = total yearly business hours I = percentage impact H =number of hours of outage

Apply this formula and you can illustrate the potential revenue savings when you reduce outages with CD.

- EASY MAINTENANCE & UPDATES

Downtime often results in irreparable damage to customer trust and the company's reputation. Catching bugs in the early stages of deployment helps you fix the bugs promptly without any delay in shipping the build. While testing takes place in the background, developers can save a lot of time, which enhances productivity.