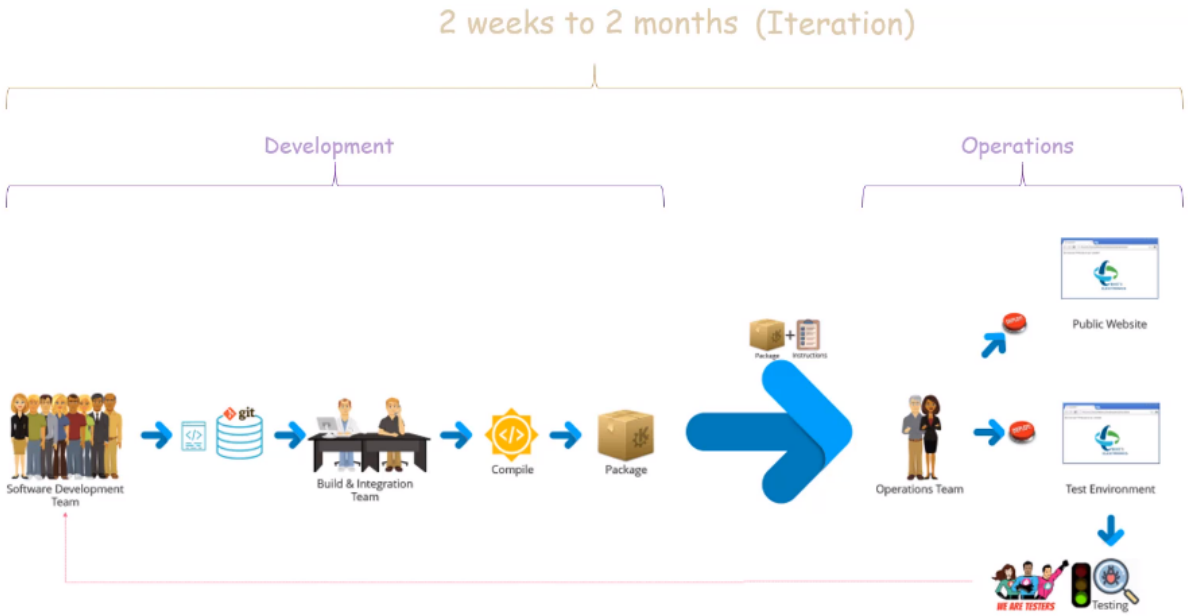
UD: Learn the concepts of Continuous Integration, Continuous Delivery / Deployment, DevOps & pipeline

▼ Old School SDLC



▼ Old School Integration Pain Points

Pain Point #1: Integration is Painful and Effort Consuming

Pain Point #2: Fixing Issues At The End Of Iterations

Pain Point #3: Intermediate Merge Issues Can Hold Up Teams

Pain Point #4: Long Feedback Cycle For Functional Defects

Pain Point #5: Long Iterations

▼ Bringing in CI

Continuous Integration is a development practice that requires developers to integrate code into a shared repository several times a day.

Cardinal principles of CI:

A single central repository where the code lives.

Developers check-in/commit their code frequently.

Build should be triggered every time a developer checks in code.

Build should be automated and fast.

Build should compile the code as well as run automated.

Fixing a failed build should be top priority for the developers.

Build results should always be communicated to all developers.

▼ Solving the pain points with CI

Pain Point #1: Integration is Painful and Effort Consuming.

CI: Integration is automated and quick.

Pain Point #2: Fixing Issues At The End Of Iterations.

CI: Issues show up early , because of frequent integration.

Pain Point #3: Merge Issues Can Hold Up Teams.

CI: Broken builds are fixed with immediate priority by developers.

Pain Point #4: Long Feedback Cycle For Functional Defects.

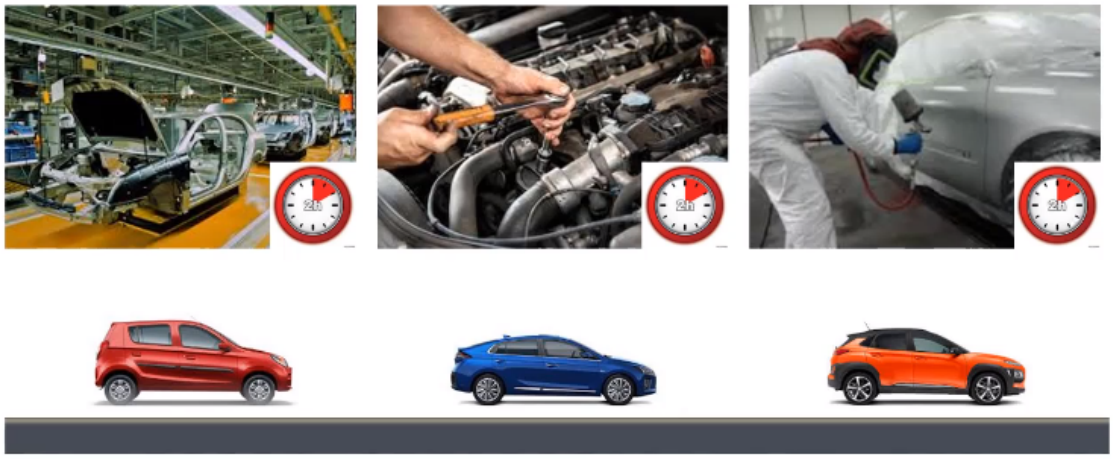
CI: Shorter feedback cycle – Developer is notified immediately.

Pain Point #5: Long Iterations

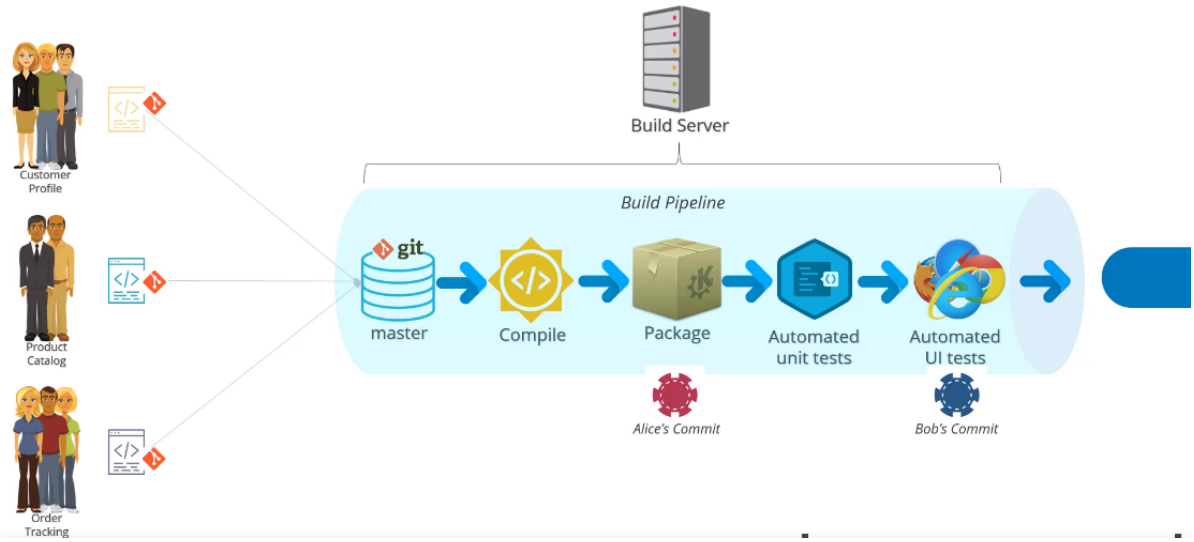
CI: Shorter Iterations. Faster time-to-market.

▼ Pipelines Explained

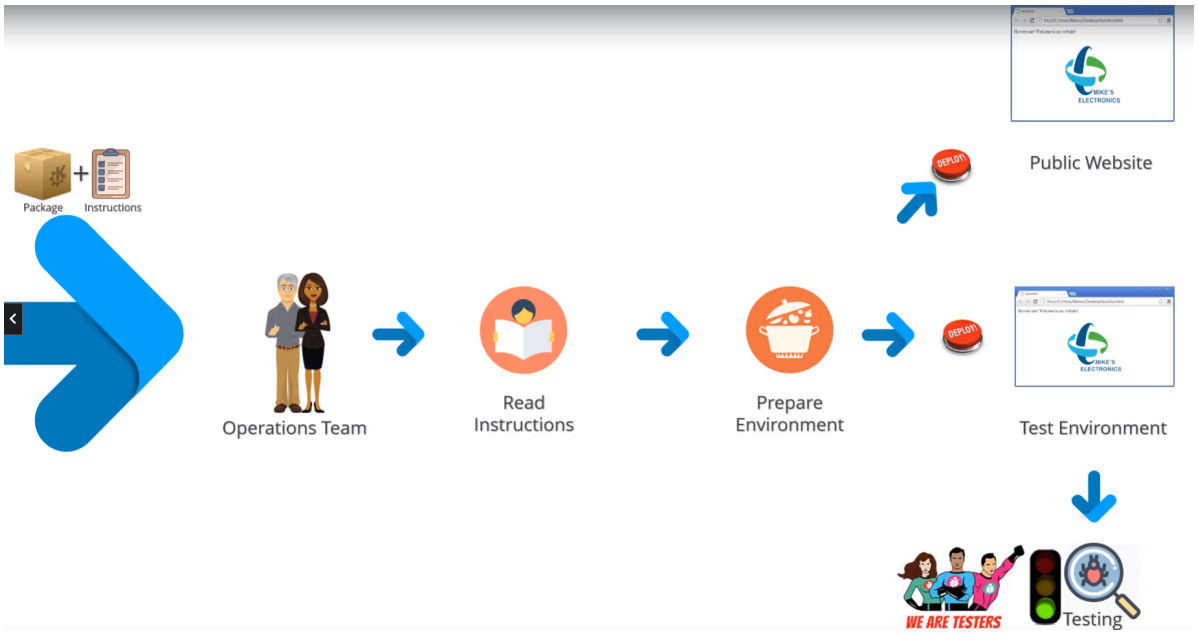
Car Assembly Line



No of cars in 24 hours = 10 cars



▼ Old School Operations



▼ Old School Operations - Pain points

Pain Point #1: Correctness of Instructions

Pain Point #2: Difference in instructions across environments

Pain Point #3: Error prone nature of manual tasks

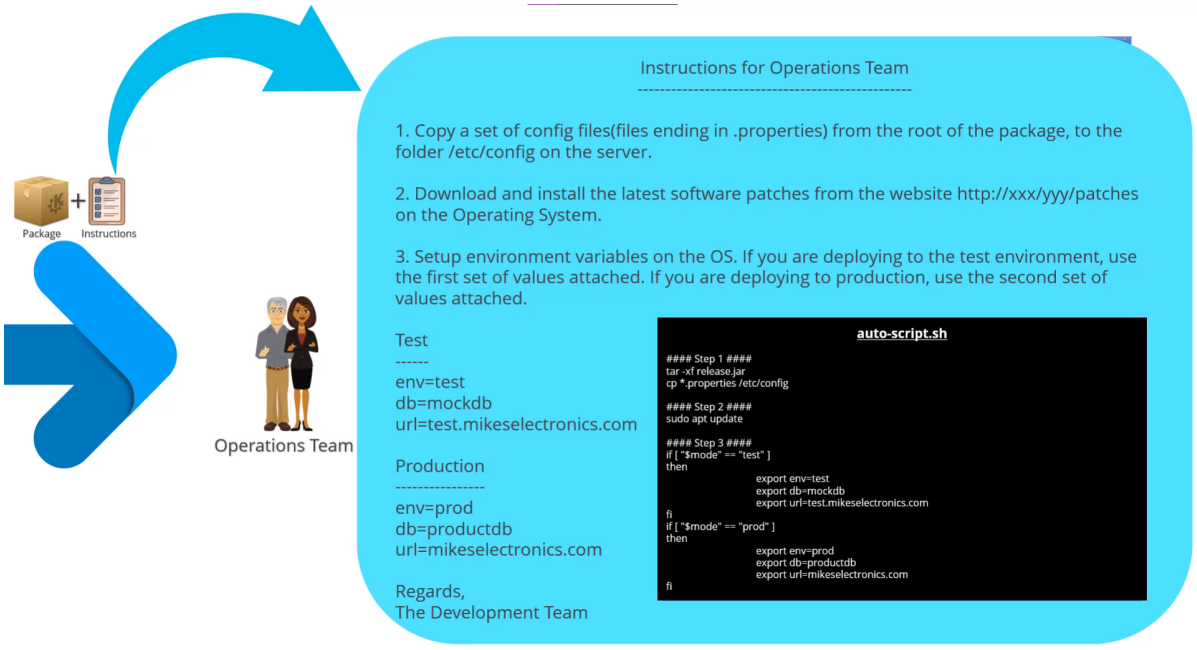
Pain Point #4: Deployments are sophisticated, high-impact with downtime

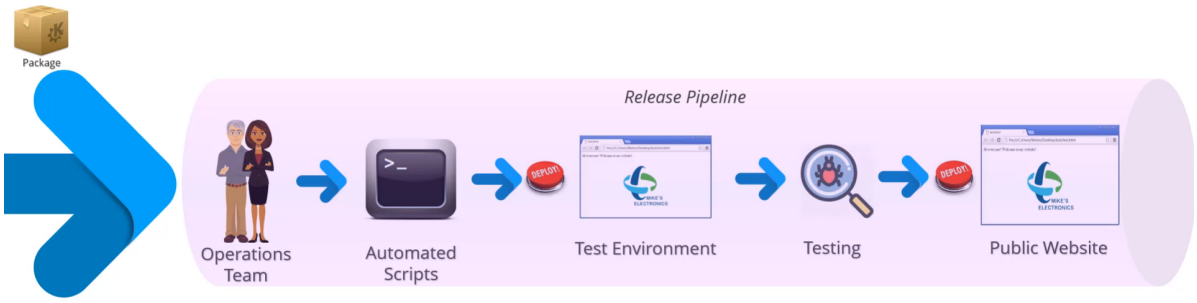
▼ Bringing in CD

Continuous delivery is a software development practice where software can be released to production at any time.

(Continuous integration is a pre-requisite.)

can use scripts instead of manual intructions:





▼ Soling the pain points with CD

Pain Point #1: Correctness of Instructions.

CD: Correctness of automated scripts can be verified at creation time.

Pain Point #2: Difference in instructions across environments.

CD: Automated scripts can easily pick the tasks for each environment.

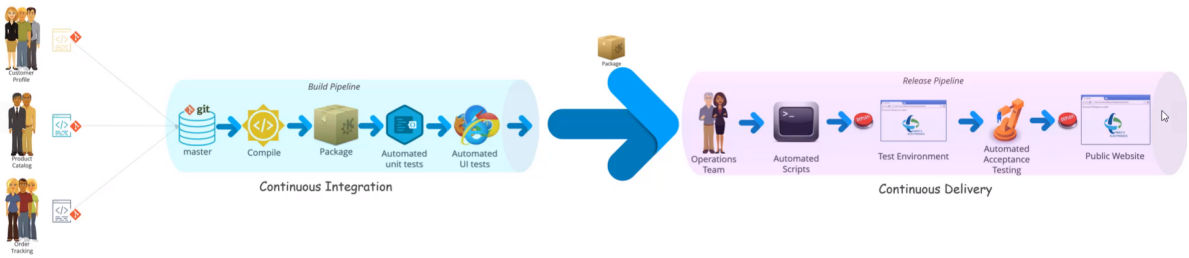
Pain Point #3: Error prone nature of manual tasks.

CD: Automation prevents the occurence of human errors.

Pain Point #4: Deployments are sophisticated, high-impact with downtime.

CD: Automated deployments, easily repeatable, lesser time-to-market.

▼ Maturing to CD



Continuous Delivery

&

Continuous Deployment

Continuous Delivery is a software development practice where software can be released to production at any time.

Continuous Deployment is a software development practice where software is automatically released to production continuously.

Continuous Deployment is not common

Only enterprises whose IT department has reached a very high maturity level, will go with Continuous Deployment, because of the risks involved.

Can do phased rollouts with Continuous Deployment - can also rollback

How many releases does Facebook push to production daily on an average?

Facebook does 50 daily production releases on an average.

That means an average of 1 production release every 30 minutes!

▼ DevOps

Term was coined by combining the terms 'DEVelopment' & 'OPerationS'

Breaking wall between dev team and ops team > goal is to reduce time to market