

# Detailed Report

## PROJECT DETAILS

### IPL Dashboard

#### Group - 6

Harshitha S	2020HS70006
Remya Ravindran	2020HS70039
C Divya Darshini	2020HS70030
Renu Sri Matam	2020HS70040
Sushmitha S	2020HS70054

#### Description

The Dashboard provides the feature to browse any IPL team and get the access to their history of the games played.

It also displays the number of games played, the history of wins and losses ratios and the tournament year.

#### Project Structure

1. GitHub repo contains the front-end and backend code for IPL dashboard written in React Js.
2. Backend is a Maven spring boot project.
3. Frontend is built on ReactJs and the backend runs on Java – SpringBoot.
4. Building automation is done using Jenkins.
5. Data for backend is taken from Kaggle (data is considered from year 2008 to 2020 of IPL games).

#### Feature Enhancements

As a future enhancement we can introduce a new section that shows newly added teams as well as teams which are not active from one or more seasons of IPL league.

We can add option for live updates of the match.

## Agile/DevOps Team structure

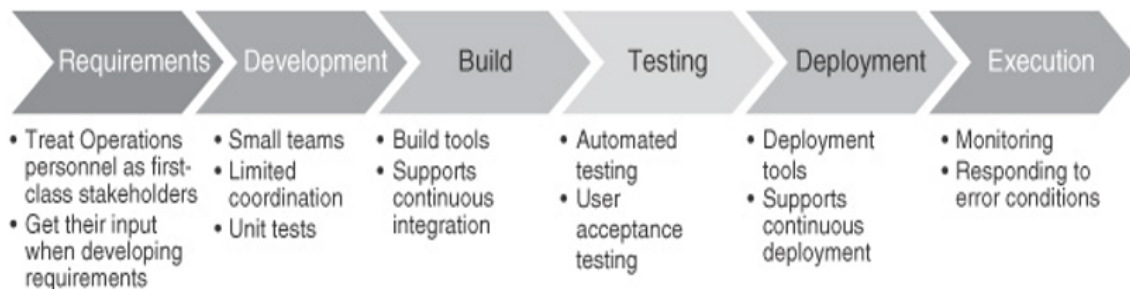
“DevOps teams are usually made up of **people with skills in both development and operations**”

On the path to practicing DevOps, we realized that it's important to understand that teams structure depending on the greater context of the Project and its appetite for change

## Team Size

Based on our current project its 5 and Devops team size varies based on organizations requirement.

## DevOps life cycle processes



## Responsibilities, Collaboration

The primary roles and responsibilities of our DevOps team were to

1. communicate effectively,
2. improve visibility across the CI/CD pipeline
3. constantly learn new things.
4. highly focused on developing new features and services quickly without sacrificing reliability or customer experience.

## How frequently do you deploy on production / staging?

By considering that team follows Agile methodology i.e Sprint planning and Sprint review so by end of each sprint we deploy it to the production /staging.

Internally for the testing of any specified features QA/developers deploy it and test it locally but for the production /staging it will be by end of each sprint. Usually, sprint lasts for 15 days.

## **Cycle Time to Build / Test / Deploy**

As it follows Continuous Integration and Continuous Deployment, when we pushed the code to the respective GitHub repository the build happens if any errors are found it was reported to the developer and was fixed.

Internally for the testing of any specified features we deployed it and tested it locally.

## **DevOps practices**

- Treat Ops as first-class citizens from the point of view of requirements
- Make Dev more responsible for relevant incident handling
- Enforce the deployment process used by all, including Dev and Ops personnel
- Use continuous deployment
- Develop infrastructure code, such as deployment scripts

## **Extent of CI / CD**

We have used Jenkins to integrate with our project as it is one of the Continuous Integration Tool.

Using Jenkins, we created builds and pipelines.

## **Toolsets used for standardization**

Devops Tools used in the completion of the project

- |                |                              |
|----------------|------------------------------|
| 1. GitHub      | Source code management       |
| 2. Maven       | Building                     |
| 3. Junit       | Testing                      |
| 4. Jenkins     | Automation                   |
| 5. SonarQube   | Code Analysis                |
| 6. IntelliJ/VS | IDE                          |
| 7. Chrome      | Web Browser                  |
| 8. ELK         | Collect logs, show dashboard |

## **Challenges faced**

1. Integrating github and jenkins together using webhooks.
2. Automating the deployment from Jenkins.
3. Deployment of SonarQube and integrating it in the pipeline.

## **Enhancements to Current Practices**

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan