|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **WHITE** | |  | **YELLOW** | |  | **GREEN** | |  | **BLACK** | |  | **RED** | |  |
|  | |  | |  |  | |  |  | |  |  | |  |
| Teams with **AWARENESS** | | Teams with an **INITIATIVE** | |  | Automate **COMPLETE** deployment | |  | **INTEGRATE** SaaS, IaaS, PaaS | |  | **PIONEER** new tech | |  |
| DevOps dojo white belt teams have foundation and a high-level understanding of DevOps, CICD & Automation. A DevOps taxonomy to help familiarize and provide you with an understanding of DevOps concepts and principles Prerequisites: none | | DevOps dojo yellow belt teams have hands-on experience with an interactive usage with EDSE Pipeline. These teams have begun onboarding their applications, and able to test CICD pipelines in DEV or UAT environment, and have put comprehensive checks to track Agile work with DevOps | |  | DevOps dojo Green belt teams are running their projects in the production environment using CICD pipeline. They are using industry standard DevOps practices in better collaboration and speed to market. | |  | DevOps dojo black belt teams share a passion for end to end automation using new tools. Have continuously improved their People, Process and Technology. These teams have undergone Project to Product transformation, iterating and involving in the Development to Production stages. | |  | Teams have an expertise in DevOps to their use case, ability have a greater impact across organization and ability to drive business with niche expertise. Use of new ways to implement new concepts such as Machine Learning, Blockchain, Data Science, Quant, etc. | |  |
| **INITIATIVE** badges | | **EXPLORATION** badges | |  | **ESTABLISH** badges | |  | **EXPANSION** badges | |  | **EXPERTISE** badges | |  |
| **Culture and Lean Product** | | **Value Stream Mapping** | |  | **Containerization** | |  | **Infrastructure as a service** | |  | **Continous ML Scaling** | |  |
|  | People, process, and products to enable continuous delivery of value to our end users |  | Optimize your processes for value delivery & speed |  |  | Use of docker, Kubernetes in your project, manage your containers |  |  | Infrastructure management and dynamic infrastructure management using Infrastructure as a Code |  |  | Ability to scale Machine Learning models & manage accuracy in the production environment |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Design Thinking** | | **Test Driven Development** | |  | **Site Reliability Engineering** | |  | **Incident Gap Analysis** | |  | **Intelligent IOT Security Platform** | |  |
|  | Design thinking in product engineering |  | Use TDD & BDD to create tests first & code second |  |  | Service Level Agreement, Service Level Incidents, Service Level Objective |  |  | Analyze Incident lifecycle & continuously improve reaction time |  |  | Enable IOT Security & Monitoring the Infrastructure |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Agile Planning** | | **Leading Change** | |  | **Continuous Testing** | |  | **SecOps** | |  | **ChatOps** | |  |
|  | Implementing Agile & SAFe |  | Leading change in DevOps process, empower people to lead change & make impact at different levels in an organization |  |  | Execute Automated Tests from a CICD pipeline, monitor quality and speed |  |  | Security must be considered from the beginning & continuously assessed. Shift Left on Security. E.g Threat Modelling |  |  | Integrating ML for cognitive chat response. Faster collaboration by integrating chatbots to CICD and servers |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Continuous Learning** | | **Version Control** | |  | **DevOps Kaizen** | |  | **Immunity Testing** | |  | **Compliance as a code** | |  |
|  | Learning new DevOps taxonomy, for Continuous Planning and Continuous Integration, Agile Planning and CAPEX estimation. |  | Use Gitlab or other SCM to records changes to a file or set of files over time so that you can recall specific versions later |  |  | Track events to continuously improve process |  |  | Using resiliency & Chaos Monkey |  |  | compliance & audit measures in your code with Verification as a service |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Pairing** | | **Continuous Integration** | |  | **Release Management** | |  | **Configuration Management** | |  | **Threat modelling** | |  |
|  | Implement Pair programming, Peer review & document review processes in agile process |  | The process of merging work from all the developers in a team into the master branch as and when required. |  |  | Managing releases with pipeline, measure work with code commits & merges into CICD. Manage deployment through tools like Spinnaker |  |  | Manage Environment using Chef/Puppet/Ansible. Shift Left on container configuration with Kubectl & Helm Charts |  |  | Use advance security concepts into your architecture with Encryption as a service |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **Pipeline monitoring** | |  | **Continous monitoring** | |  | **DevBizOps** | |  |
|  |  |  |  |  |  | Monitoring & control of your pipeline with a dashboard |  |  | Real-time dashboards to evaluate your releases in the dashboards |  |  | Business IT alignment from the Biz loop by automatically aligning all downstream activities with the goals and objectives needed to deliver the expected business value. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |