**DevOps Maturity Report**

**The DevOps practices used in our project:**

1. Planning the next iteration of the product’s development
2. Version Control
3. Building the code
4. Testing and deploying to the production environment
5. Delivering product updates
6. Monitoring and logging software performance
7. Gathering customer feedback

**Steps taken in team to ensure DevOps Maturity Model**

* Our dev, ops, service management, security, QA teams are distinct and independent.
* QA is embedded with dev teams to ensure high quality in operations
* Security is interwoven into the development process
* Our architecture is designed for operations and scale
* Development gets it to work and hands off to ops to finish the job
* Continuous deployment to production augmented with experimental releases
* Continuous integration, automating code into the main branch

**Benefits gained by using this practice:**

Companies that incorporate DevOps practices get more done, plain, and simple. With a single team composed of cross-functional members all working in collaboration, DevOps organizations can deliver with maximum speed, functionality, and innovation.

****

**Challenges faced**

**Inadequate risk management**

Having the foresight to identify potential ‘what if’ scenarios and making up contingency plans is an important aspect of project management. Projects rarely go exactly as planned because there are so many variables that can create unlimited possibilities.

**How we dealt with it:**

It is the job of every project manager to come up with alternate plans that the team may adopt if the project begins to spiral out of control. Having a project risk management system helps in identifying the types of risks and mitigating them. Having a contingency plan in place is critical. This plan should identify all risks that the course of action to be taken if they materialize. So, we identified the risks and simplified them.

**Six Challenges in DevOps Adoption:**

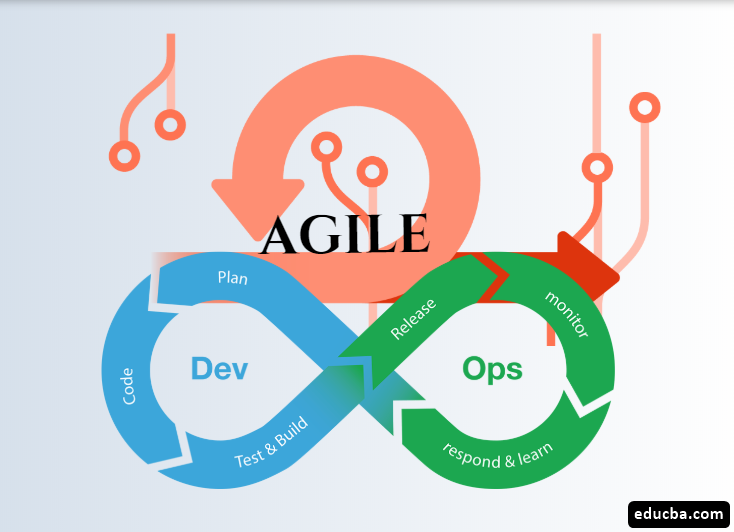
1. Moving from Traditional Infrastructure to Microservices.
2. Integrating Tools of Different Domains.
3. Changing Well Defined Processes to More Efficient Ones.
4. Separate Tools Set for Dev and Ops Team.
5. Adoption of New Tools.
6. Resistance to Change.

**Enhancements in current practices:**

1. Think agile, stay agile Release management practices
2. Don’t stop innovating
3. Use the right tools
4. Monitoring devops pipelines and applications
5. Using containers to drive microservices
6. Automating configuring and provisioning with infrastructure as code (IAC)
7. Automating release management with continuous integration and deployment

**Agile Practice and DevOps:**

**Automation of workflow** is another part of Agile combining with DevOps.



**Features and advantages of DevOps practices and processes used in the team**

**Microservices**

The microservices architecture is a design approach to build a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism. You can use different frameworks or programming languages to write microservices and deploy them independently, as a single service, or as a group of services.

Organizations may also use a microservices architecture to make their applications more flexible and enable quicker innovation. Typically, each service is paired with a small, agile team who takes ownership of the service.

**Continuous integration and continuous delivery**

DevOps practices such as CI/CD let DevOps teams deliver rapidly, safely, and reliably. CI is a software development practice where developers regularly merge their code changes into a central repository, followed by automated builds and tests. The key goals of CI are to find and fix bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates. CD expands on CI by deploying all code changes to a testing or production environment after the build stage.

**Monitoring and logging**

By capturing and analysing logs generated by applications, DevOps teams can better understand how software changes or updates may affect users.

**Building a secure DevOps model**

Moving to DevOps and DevSecOps is not a destination. It is a journey. DevOps is fundamentally changing how development and operations are done today. You can use the DevOps practices, process, frameworks, and workflow, based on the DevOps philosophy, to build security into your software development life cycle at speed and scale without sacrificing safety, while minimizing risks, ensuring compliance, and reducing friction and costs. DevOps and DevSecOps allow development, operations, and security teams to balance security and compliance with speed of delivery, and to build security into the full SDLC.

**Extent of CI/CD with Agile and DevOps:**

