Bot enabled Continuous Integration & Delivery for DevOps workflows

**A Mindtree thought paper**

**Background**

DevOps is a term that has gained enough traction over the last four years that it can now be considered mainstream and no longer the buzzword. The backbone of DevOps is its Continuous Integration and Delivery framework. Software Development cannot afford to go back to models that aren’t as quick or risk mitigating as the combination of Agile and DevOps. The success for a business application is now measured against the number of successful code commits and releases that happen into production per day and per week. Successful code commits and releases are the important triple underlined term here.

Multiple companies face challenges in adopting a robust Continuous Integration and Delivery system. Due to the number of moving parts, Organisation cultural changes within app development and operations teams, and multiple teams performing activities that require a solid communication feedback, even a good implementation of DevOps can run into efficiency and functional issues. With a myriad number of tools that an organization can leverage for their DevOps requirements, it is easy for the App Dev and IT ops teams to end up in a tools and platform fatigue. The app development team, testing team and operations (deployments and provision) team eventually end up spending more time tweaking and reworking various systems to deal with the increase in errors and changes to the systems.

This leads to them falling short of reaching the maturity level that they hoped to achieve with their DevOps implementation.

Automation alone is not the answer here.

Intelligent automation, using the advances in AI and predominantly in Machine learning, can enable a seamless Continuous Integration and delivery framework. This will reduce the burden of constant maintenance of the app development mechanisms and with Machine Learning built-in, the framework will only be getting smarter and more robust as the teams develop and deploy code.

**Continuous Integration and Delivery Framework**

DevOps is an IT Service Delivery methodology where App development and Service Operations entities merge their functional principles towards the common goal of quicker Time to Market while maintaining a stable delivery environment.

To fully realise DevOps, every CIO needs to invest in a strong Continuous Integration and Delivery foundation. Application development teams are responsible for code builds, deployments and testing while IT Ops are responsible for Environment builds, production deployments and platform support. Continuous integration and delivery provides the requisite platform for both App Dev and IT Ops teams to converge their practices, reducing the time taken between testing and release of applications to the end users. Both teams stand to gain with Continuous Integration and delivery.

The App dev teams are offered greater visibility of their end-to-end managed environments and have a quicker set-up and build times. IT Ops team see the downstream effects of Dev teams activities, and enable Dev team to anticipate issues effectively by way of continuous delivery. The framework provides greater synergy between the two teams through the discovery, fine-tuning and optimization of repeatable processes.

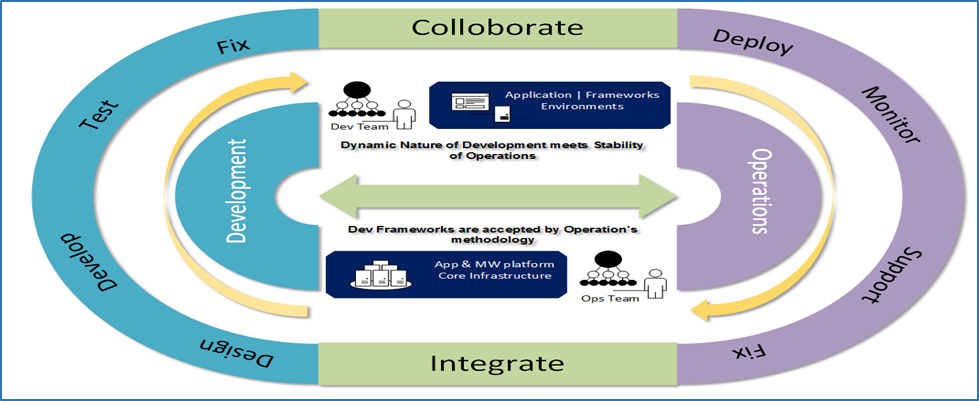


Figure 1: Continuous Integration and Delivery

The first challenge encountered by the organisation in its DevOps journey is the misalignment of App Dev and IT Ops teams. This usually leads to bridging gaps between the tools and practices used by both teams. Having tool segmentation occur in the beginning phases of establishing any new framework leads to months of FTE spent in providing work arounds or rationalising the system. The market currently has a plethora of tools to accommodate all the varying factors presented in Figure 1. Each subset of the framework has tools designed to solve challenges and help along the process. It is no surprise therefore that every team within an organisation ends up having their own set of tools and platforms.

In essence, there are 5 broad subsets of Continuous Integration and delivery framework -

* Code commit and build
* Unit testing
* Code Analysis
* Application Packaging
* Deployment configuration and scripting

As mentioned before, an organisation usually ends up with multiple tools between various App Dev teams and IT Ops teams, leading to tools fatigue. Automating each of the following subsets is among the best ways to reduce issues while working with multiple tools or even a rationalized set of tools for each subsets. IT Ops teams automate the Application Packaging and Deployment subsets while App Dev teams automate unit testing and code analysis.

While Automation does help move the app development and operations along with straightforward overview, tool segmentation would make the platform and the framework unstable, leading to numerous errors. Fixing the errors takes time and brings multiple changes to the platform, thereby making the teams lose faith in their own systems.

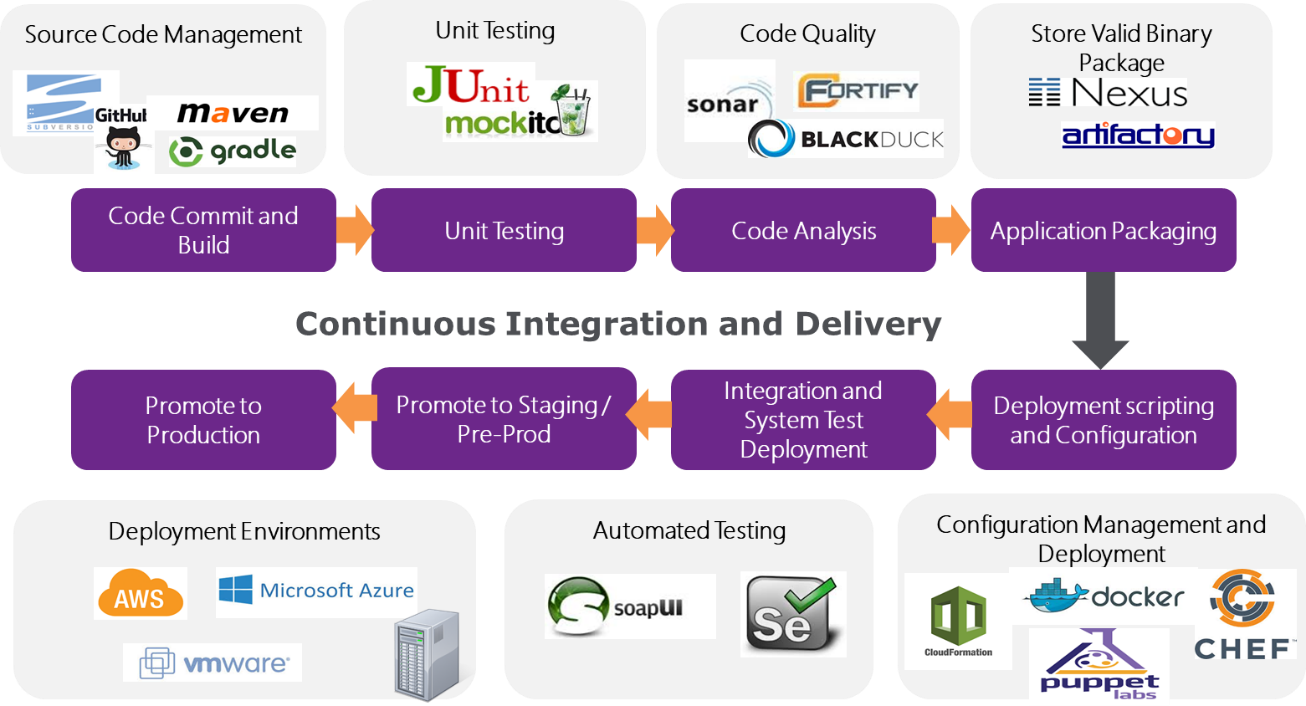


Figure 2: Subsets of Continuous Integration and Delivery with associated tools and environment details

**Machine learning enabled Bot for Continuous Integration & Delivery**

Artificial Intelligence broadly unifies multiple specializations that enable automated machines to perform cognitive functions and independently evolve heuristic capabilities. To enable AI within any industry, engineers use Machine learning or Deep learning algorithms to enable programs to perform independent tasks. When looking at the context of AI within the DevOps philosophy, Machine learning is the way to go.

Machine learning (ML) is the ability for computer programs to self-learn from experience without any explicit programming for said learning. The term was coined by Arthur Samuel, a researcher computer gaming and a pioneer in Artificial intelligence, in 1959 while working at IBM [1]. Great strides are being made in the world of Artificial intelligence with Google, Microsoft, IBM and Facebook leading the charge. Google’s Alphabet company Deepmind and IBM’s Watson are among the most famous examples which showcase how Machine Learning is helping overcome challenges within the real world.

Bots can be classified as automatons designed to perform specific operations, wherein they mimic a human interface. Common examples include Chatbots for customer representative domains, network bots that index web pages or even Clippy from Microsoft. Mindtree uses its own developed Chatbot Maci for issue resolution among its17000 global employee base. Bots can also be programmed in the IT Ops space to perform rudimentary operations such as unlocking user accounts, creating new user accounts or providing accesses to systems based on secure approval processes.

Mindtree has been a pioneer in the field of DevOps, being one of the first Managed Services providers to start its own DevOps services team, experienced with almost the entirety of DevOps specific tools. Taking off from this experience, and leveraging the AI practice within, Mindtree is working on developing Machine learning enabled Bots that will oversee automations within the Continuous Integration and Delivery space, by being the core programs controlling tools for the 5 subsets mentioned above. Instead of having the teams perform specific operations for every release or maintenance activities, the DevOps bots are programmed to handle requests and perform the appropriate actions on scripts developed by the specific teams.

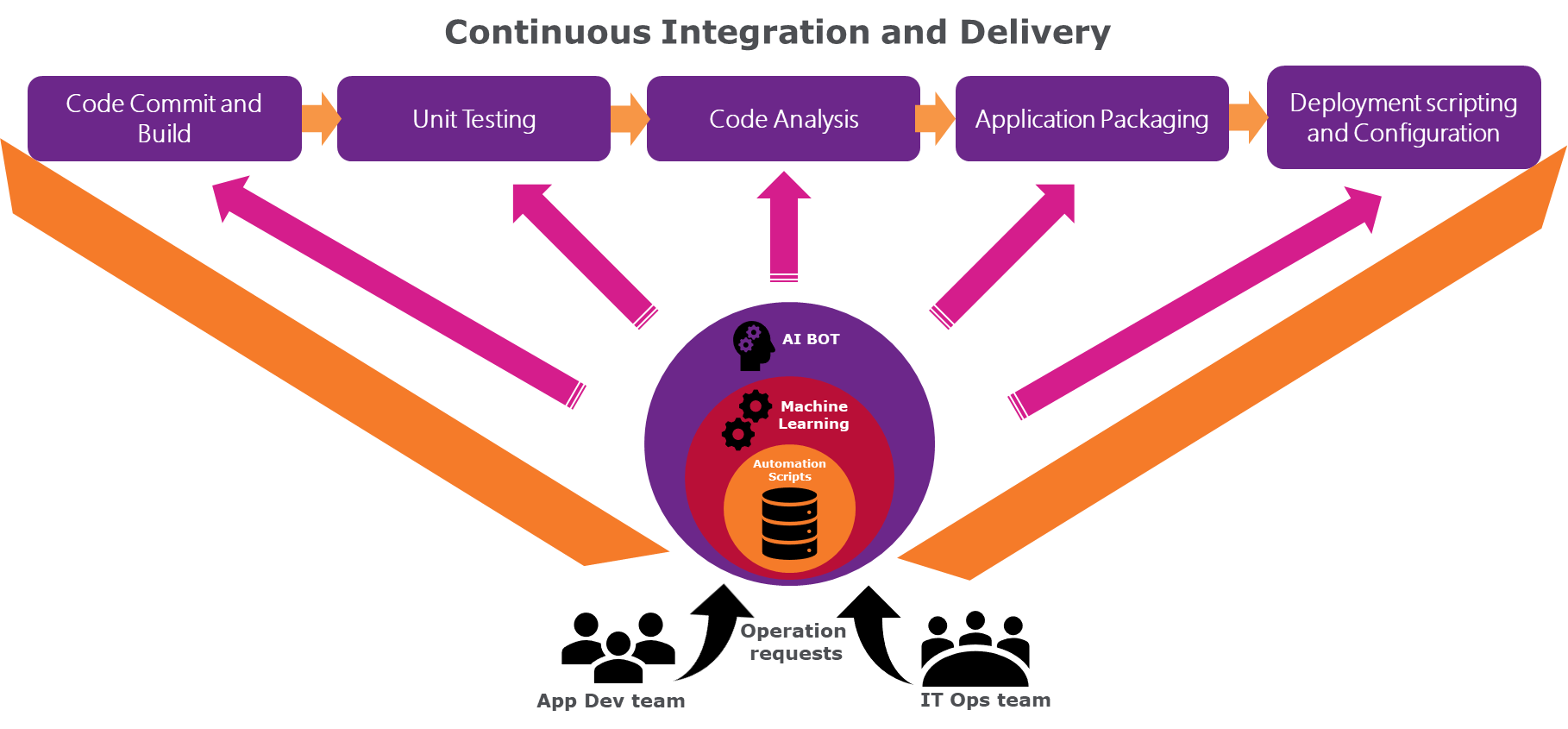
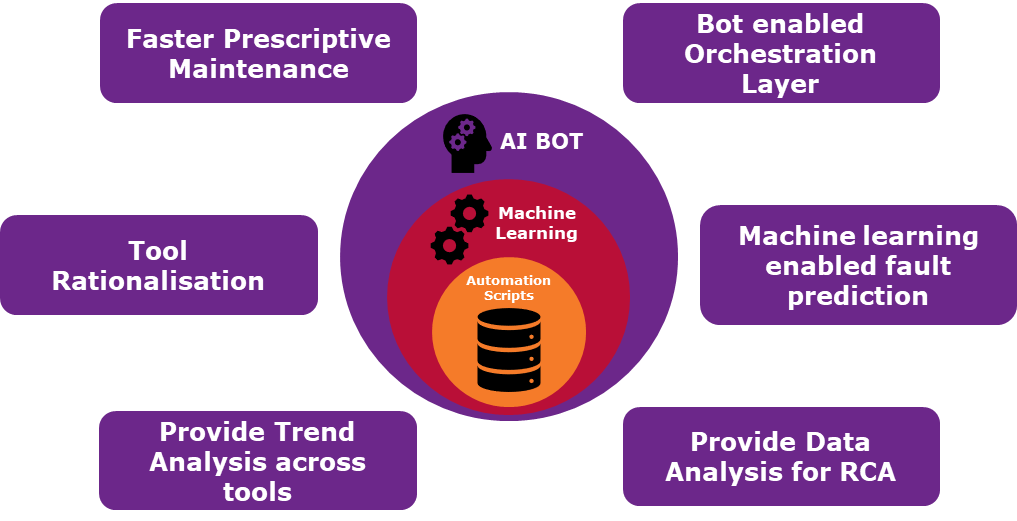


Figure 3: AI Bot handling automations and maintenance requests from App Dev and IT Ops teams

With AI Bots handling DevOps operational requests from both App Dev team, while performing maintenance tasks prescribed by IT Ops team, the following advantages start making themselves evident –

* **Faster Prescriptive Maintenance**: Bots are quicker in identifying and isolating environment issues and perform prescriptive maintenance activities before the issues have larger impact.
* **Tool Rationalisation**: Bots can be enabled to act as Manager of Managers, correlate various integrations between tools and provide a combined metric view for teams that want to perform trend analysis.
* **Provide Trend Analysis across tools**: Bots will be able to provide valuable insight and correlation between multiple tools. This will help both teams understand and rationalise tools.
* **Provide Data Analysis for RCA**: Root cause capturing by Bots helps track down issues and understand deeper problems, thereby providing metrics based data to App Dev team.
* **Machine learning enabled fault prediction**: Bots enabled with Machine learning can use historical data of code commits, issues, failure and root cause to identify probable faults and enable IT Ops team to rectify issues before they happen.
* **Bot enabled Orchestration Layer**: Bots can provide self-service enabled server orchestration on demand and even help the infrastructure scale up to demand through Data analysis.



**Conclusion**

Mindtree is leveraging its experience as a pioneer in DevOps Managed services offerings and with its AI Practices organization to develop Machine learning enabled Bots that handle automated tasks within the DevOps framework. Identifying the challenges faced by multiple companies and the associated teams with their Continuous Integration and Delivery systems, Mindtree has great insight into addressing the problems through cutting edge technical solutions.

Automated Bots, with Machine learning protocols at its core, can become the interface between the DevOps Continuous Integration and Delivery tools and the App Dev/IT Ops team, providing far better agility, faster deployments and increased fault tolerance.

The Bots solution also reduces Tools and Platform fatigue, Maintenance times and manual tasks, enabling the App Dev teams to concentrate on better product development and faster Time to Market.

Mindtree has been working on the latest Machine Learning platforms and has a deep understanding of the vast benefits this revolution will bring and has the required value driven expertise in the AI space. Mindtree currently is servicing various businesses in employing this framework and is an enabler in their AI success stories.

**References**

[1] Samuel, Arthur (1959). [*"Some Studies in Machine Learning Using the Game of Checkers"*](https://doi.org/10.1147/rd.33.0210). IBM Journal of Research and Development.