Enabling DevOps on NonStop

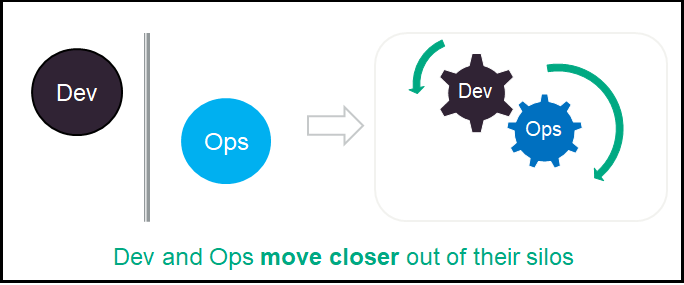
**Needs of High Performing Organizations Today**

With the on-demand nature of services today, enterprises require shorter development and deployment cycles to create and improve products in order to stay relevant. This means frequent deployments and faster recovery in case of failures. This requires **better communications** between the development, quality assurance and the operations teams, teams with **multiple competencies**, and **automation** to accelerate development and drive efficiency. This is where DevOps plays an important role.

**What is DevOps?**

The Wikipedia definition of DevOps says that “it a set of practices intended to reduce the time between committing a change to a system and the change being placed into production, while ensuring high quality“

DevOps has people at its core. DevOps as the name suggests, is a culture of collaboration than contracts. It is a set of principles, practices to build effective **collaboration** between Development & Operations teams. It breaks the patterns of working in silos, by bringing people together to collaborate in order to bring in efficiency.



This is made possible as DevOps leverages on solid foundations like Agile, Lean principles, extending them to entire organization. DevOps **increases software lifecycle predictability** with faster feature delivery and faster feedback enabled by automation.

In DevOps everything is continuous.

Continuous Integration (CI) - automates the build & testing of code every time a team member commits changes to version control

Continuous Delivery - Extends CI to release-process, gets new changes released quickly and reliably

Continuous Deployment - Extends Continuous Delivery automating the release-process

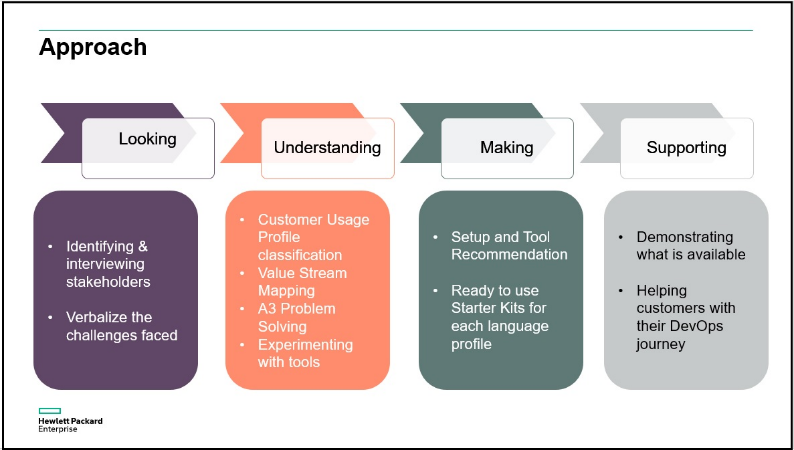
DevOps helps accelerate time from unmet market needs into making them available to business, and capture outcomes.

**DevOps Enablers**

People and Automation are the two enablers for DevOps. While a culture of collaboration and trust is important, equally important are the right choice of tools and automation. From Planning, to Code, Build, Test, Release, Deploy and Monitor, the right tools are required to successfully implement DevOps. In this article we will look at the DevOps enablers for NonStop mostly from the tools and automation point of view.

**DevOps for NonStop**

With the goal to enable NonStop customers adopt DevOps, used a four phase (Look, Understand, Make and Support) approach to identify challenges, addressing the gaps in tooling and providing tool chain recommendations and best practices.



The in first phase, we interviewed Solution Architects who interface with customers to learn about challenges faced. The development environment of NonStop customers is very different from the ones in our labs due to stringent compliance requirements. This was an important aspect which we had to keep in mind when we came up with recommendations.

With all the information gathered through interviews and simulation of customer experience today, we tried to put them in perspective in the understand phase where we tried to identify challenges and gaps using some lean principles like value stream mapping and A3 problem solving. There were two outcomes of this phase namely Language based classification of NonStop Customer Usage profiles ([CustomerUsageProfileClassification](https://github.com/HewlettPackard/NonStop/blob/main/nsdevops/images/CustomerUsageProfileClassification.jpg)) and Tool sets to support DevOps on NonStop.

NonStop supports multiple languages and runtimes and customers use one or more of these profiles to develop and deploy applications. There are three main profiles, GUARDIAN (C, COBOL, pTAL) , OSS (C/C++) and Platform Agnostic OSS, off which there are again two categories, partially platform agnostic (example Java + JNI and any polygot with C/C++ env on OSS) and fully platform agnostic (Java, Python). Each profile has its own set of **challenges which are unique to the profile**

For each of the language profile, we started identifying the tool sets and gaps there in.

Getting hands-on, in the making phase, we experimented with different types of setup and zeroed in on one setup that we could recommend to customers. The recommended setup ([RecommendedSetup](https://github.com/HewlettPackard/NonStop/blob/main/nsdevops/images/RecommendedSetup.jpg)) includes a centralized server (Windows/Linux) hosting most of the tools including Jenkins Master for Continuous Integration, source code management, cross compilers, build automation tools, static analysis tools, artifact repository, provisioning and orchestration tools.

**Starter Kits**

While tool recommendations and best practices guides help, we also wanted to provide some tangible resources for customers who wanted to get started with DevOps. By tangible, we meant something that is usable as is in the customer environment, that and get them started. The focus was on demonstrating pipelines and tools for a language usage profile, reusable scripts and pipelines, easy to get started and try out with customer applications,

So, in the make phase, we built starter-kits. These starter kits are language based that is customer usage profile specific, ready to use, developer-friendly and production-ready. Each starter-kit consists of A sample application (typically client/server), a set of pipeline scripts and a README file with usage instructions.

Currently these are hosted on GitHub <https://github.com/HewlettPackard/NonStop/tree/main/nsdevops>

Customers can clone the repository, follow the instructions in the [HPE NonStop ModernDevOps - Instructions for CI-CD setup](https://github.com/HewlettPackard/NonStop/blob/main/nsdevops/HPE%20Nonstop%20Server-Modern%20DevOps-Instructions-for-CI-CD-Setup%20Documnet_v1.2.pdf) , update the pipeline scripts according to their setup and get started. The scripts can also be re-used for applications with similar profile with very little changes to the GIT repository and build steps if any.

In the support phase, we have and continue to support our customers in their DevOps journey by helping them with tooling, coming up with new starter-kits for new use cases like usage with public cloud and also ensuring a rich set of tools are available to easy this journey.

To conclude DevOps is supported on NonStop and starter-kits are here to help you get started.