Difference between IT Orchestration and IT Automation Explained

Automation is a buzzy topic in today’s IT world. Whether you work in IT or in another area of the business, you may hear about automating as a way of saving money, improving efficiencies, and removing inherent errors.

Often, automation is a term that may be used only partially correctly. In this article, we’re clearing up the differences between orchestration and automation and sharing how beneficial both are, when used in tandem, in a DevOps culture.

Defining Automation

When we hear about automation, it’s typically in reference to making entire processes or workflows run. This isn’t quite right.
The definition of automation is setting up a single task to run on its own – automating one thing. This single task can be anything from launching a web server, stopping a service, or integrating a web app.

Automating a task can apply to both desktop-only tasks and cloud tasks.

**Defining Orchestration**

More often, orchestration is the term we mean when we refer to automating a lot of things at once. After all, automating a process requires countless steps, often spanning app, mobile, and database – so orchestration is the perfect term for this larger, more complex technique.

The definition of orchestration includes the automated arrangement, coordination, and management of computer systems, middleware, and services. Orchestration takes advantage of multiple tasks that are automated in order to automatically execute a larger workflow or process. These could be comprised of multiple tasks that are automated and could involve multiple systems.

The goal of orchestration is to streamline and optimize frequent, repeatable processes to ensure accurate, speedier deployment of software – because companies know that the shorter the time-to-market, the more likely that success will follow. Anytime a process is repeatable and its tasks can be automated, orchestration can be used to optimize the process in order to eliminate redundancies.

Here’s an example: A developer needs to enter a line of code throughout the software. At first glance, it may seem like a quick line of code that the developer can manually insert. But there is a risk of human error when entering the code by hand. If the code is entered incorrectly, the developer has to spend time fixing the mistake, wasting valuable time.
Instead, the developer can opt to orchestrate this additional code by automating the individual tasks that comprise the process of inserting this code. That’s nice on its own and it may not seem necessary – but importantly, the developer has ensured that bad code isn’t entered, increasing efficiency by ensuring mistakes don’t occur.

Benefits of Automation and Orchestration

Automating one task may impress your users or make them aware that something has sped up a little bit. On its own, simple automations will not make a big perceivable difference. But when automation is built into a series of processes and workflows, which are then orchestrated to run automatically, there can be endless benefits.

Some commonly cited benefits include:

- Decreasing IT costs – or better using the budget towards innovation and new projects
- Decreasing friction among teams
- Increasing – and perhaps improving – productivity
- Standardizing processes and products across the spectrum so that they are more consistent and reliable

In an ideal DevOps culture, both concepts are used in tandem. Indeed, automation is the first step towards orchestration. Using both means they can be played off each other to streamline and accurately execute processes that lead to speedier and successful app deployments.

As automation takes care of technical tasks, orchestration takes care of IT workflows that are comprised of numerous technical tasks. The next step, then, is using orchestration to support a sophisticated DevOps culture. Well-orchestrated IT processes enable, support, and empower key DevOps theories...
of **continuous integration (CI) and continuous delivery (CD)**. The more processes that are orchestrated, the more DevOps engineers can work with other teams to create apps that meet a lot more (ideally all) requested requirements.

**Tips for Automating and Orchestrating**

At this point, you may be wondering why you aren’t automating or orchestrating all your IT needs. Some people say this is the next step of DevOps – when developers are freed from tasks to continue innovation.

Automation and orchestration can occur on an individual level or on a team- or company-wide level. Both non-IT and IT employees can take advantage of automation suites that can integrate many popular apps, like email, Microsoft and Google products, Slack, sales and marketing tools, and a lot more – requiring no programming or coding know-how.

Automating larger or more custom suites, like an in-house software solution, will likely require IT build out the workflows. In this scenario, you’ll need to prioritize workflows and processes.

Here are a few tips on choosing how and what to automate and orchestrate:

- Consider the business needs, not only what makes things better for the IT team
- Automate tasks that applications or machines can do with more speed and accuracy than a human can – even if it’s automating something as simple as a line of code, this can avoid human error mistakes
- Orchestrate workflows that free up the team to tackle more valuable projects that are not as easy to orchestrate, like creative and innovative output, or the
• Choose projects that create significant and measurable business value. If you’re using orchestration only to speed up how quickly tasks get completed, you’re not receiving the true business value that orchestration can provide.

As automation and orchestration continue to improve tasks and automation, more teams — both IT and otherwise — are able to work towards an ideal product that incorporates a larger amount, or eventually all, of the requested requirements.

Additional Resources

The Accelerator’s Guide to Digital Transformation from BMC Software