



Transforming the
Developer Experience
Inside Dell's IT Organization

As Dell Digital, Dell Technologies' IT organization, continues to drive our companywide digital transformation, the role of our developers in that journey cannot be overstated. We rely on our developers to continuously deliver the innovative solutions our business users need fast and efficiently so we can create better experiences for our team members and customers. Developers remain at the center of our efforts to drive human progress through people, process and technology.

From eliminating manual and administrative tasks to standardizing and automating development processes to sharing reusable components, our work inside Dell Digital has made great strides in improving our developer experience. Here are some first-hand insights into how we have cracked the code on creating a world-class developer experience. We hope you'll find what we've learned useful as you consider new ways to innovate for your own teams, customers and business partners.

CONTENTS

- | | | | |
|----------|---|-----------|---|
| 3 | Cracking the Code for a World-class Developer Experience | 9 | A Shared Building Block Approach to Better User Experience |
| 5 | Finding Our DevOps North Star | 11 | Equipping Our Developers Inside Dell With Application Intelligence |
| 7 | Helping Developers Work Smarter With Our API Marketplace | 13 | How Dell Improved IT Developer Productivity by 275 Percent |



Cracking the Code for a World-class Developer Experience

Find out how Dell IT has freed developers from administrative tasks so they can do what they do best

By Jaynene Hapanowicz, CTO & SVP, Dell Digital – Technology Transformation and Services Development

Several years ago, Dell Digital, the internal IT team at Dell Technologies, did a study that showed us something that most IT organizations have long struggled with—a lack of developer productivity. We found that our developers were only spending an estimated 20% of their time writing functional code. The rest of the time they were dealing with administrative tasks like provisioning infrastructure, running quality and security scans and getting their code into production. This required coordinating with change and release management, operations and a variety of other dependencies across the organization.

Faced with that nagging statistic, we decided to double down on creating a developer experience that could support them so they could spend most of their time doing what developers do best — creating solutions for the business. This is what all companies are trying to do as they accelerate their digital transformation agendas.

I am happy to say that we have cracked the code on how to create great experiences for our developers — to let them run faster and do the work they like to do without being bogged down in administrative tasks and manual processes.

Our approach is multifaceted, including a fully automated cloud, with a catalog of robust services and a fully automated DevOps pipeline that has resulted in skyrocketing development productivity and quality.

Three years into our journey, our developers are now spending between 70% to 75% of their time writing functional code and using their innovative talents to introduce new solutions and add value to our company. They can provision the infrastructure they need in a matter of minutes instead

of days, weeks or months and quickly deploy the code in a standardized and automated manner using our CI/CD pipeline.

Moving beyond manual

When you go back several years at Dell, it could take 70 or more days to build an infrastructure before a developer really started writing the first line of code. This continues to plague many companies still using manual processes to build an environment.

Many times, you had to have a one-off design, order hardware, get approvals, receive it, get it racked and stacked, connect to the network, apply security settings, download the operating system, add configurations and then build your services on top, depending on what the workload was designed to do in that environment. Administering all these manual processes to build an environment is overwhelming and not an efficient use of time.

Along with the hugely time-consuming back and forth of manually building an infrastructure, the other burdens are the manual side of change management and of running code through quality and security scanning tools in disparate places. These activities have everything to do with the setup of the environment and nothing to do with writing functional code.

This is why developers turn to public cloud providers to get infrastructure services at the swipe of a credit card.

We knew the solution was clear, albeit challenging. We needed to adopt a maniacal focus on automating everything we do in our IT environment.

Automation, automation, automation

Making the transition to automation is a marathon, not a sprint. We began to build the automation of infrastructure services back in 2016 and pushed for a real change toward adopting DevOps across the organization the following year.

Uniquely, I had previously run the infrastructure team and pivoted to the DevOps team in 2019, I spearheaded a joint effort by both teams to drive automation. I understood both sides of this equation and how important it is to have empathy for the challenges that both infrastructure and development face. Only with both teams working together is it possible to transform your IT environment.

On the infrastructure team, we had to determine what automation capabilities we were going to use and how we were going to spin up those environments. This included creating capabilities that automated the provisioning of virtual machines (VMs), a network to build and segment containers, load balancing, security, database and caching services. We used, and continue to use, VMware technology, including VMware Tanzu for self-service cloud-native solutions and NSX for networking.

We also had to determine a reference architecture defining how our services were going to be built in a standardized yet flexible way. We took an API-first approach for everything that we built and incorporated that into our architecture. We worked closely with the development teams and created a self-service catalog that lets developers define and chose the capabilities they need—compute, network, database, etc. Today we have provided an environment that lets developers build what they need in five to 30 minutes.

A single DevOps pipeline

In parallel with infrastructure automation, the DevOps team worked to create all the automation services that a developer needs to optimize their job. We started by analyzing the more than ten pipeline solutions that had evolved around different development organizations over the years and defining the strategy of what we needed our single comprehensive pipeline to do.

Working with the development teams, we built a path with all the integrations and blueprints our developers needed to deploy their code. And then we migrated all teams to a single pipeline platform.

Developers write test cases, write their code, test the code against the test cases, deploy it into staging, perform code fixes and redeploy. They also need to run a variety of security checks before, during and after deploying into production. All of that is automated in our CI/CD pipeline.

We use third-party vendors and in-house capabilities to create service integration points to put everything that a developer needs at their fingertips — from scanning code

for security vulnerabilities, to checking for 12-factor coding best practices, to credential management.

Our API-first approach to developing code has yielded further development efficiencies. Developers can share APIs via our API marketplace. APIs are acting as gateways to unlock digital assets and data for use by IT and other internal business units and departments, as well as external businesses, customers and partners.

Our CI/CD pipeline has resolved what was a massive time suck for developers when they used to do all these steps manually.

Our pipeline, coupled with our catalog of automated infrastructure services, has created what we're calling an industry-leading developer experience.

Skyrocketing productivity, quality, efficiency

The benefits of our transformed developer experience are clear. Developer productivity is going through the roof and quality is significantly better. The time it takes us to address defects is dropping and the development feature cycle time is reducing, which means our efficiency is climbing.

We expect to finish our current fiscal year with 21 million pipeline job runs, doubling from last year. We have achieved more than a 35% increase in overall developer productivity in 2021 alone. The quality of the code has improved, the frequency of deployments and features have increased and the number of production incidents has sharply dropped.

Overall, cracking the code on creating an industry-leading developer experience means we have the velocity we need to shift more of our focus to innovation and driving business solutions forward. ■



Finding Our DevOps North Star

Learn how Dell IT is maximizing developer experience and efficiency with a streamlined, fully automated DevOps solution

By Cody Taylor, Senior Director, Product Management, Technology Transformation Services for Dell Digital

Being able to develop, deploy and maintain high-quality and secure software at the speed that business demands is essential to today's modern IT operations. As part of a three-year DevOps enablement push, Dell Digital, Dell's IT organization, has created a streamlined, fully automated solution that does just that.

Optimizing our DevOps approach to software development is a key pillar of Dell Digital's ongoing digital transformation, moving us away from fragmented and manual development processes to an agile, flexible, automated IT operation to meet fast-changing user needs.

Our focus was defining key process improvements to maximize developer experience and efficiency. This included creating a standardized CI/CD pipeline that has removed all manual intervention between code creation and production deployment and support. The pipeline gives developers a way to manage their application lifecycle autonomously. They can automatically provision infrastructure; perform functional, performance and security validations; quality check their code; and deploy to production with integrated monitoring and application tracing.

This means developers who were previously bogged down in operation tasks can now spend more time writing code and better serving our business users and customers.

This transformation takes time and commitment, but it's critical in today's fast-moving IT marketplace.

Mapping out a plan

So where did we start? After creating a DevOps Enablement team, consisting largely of engineers and architects, we began by auditing our current processes to identify the bottlenecks that were most slowing us down. We discovered major areas that required attention: the way we managed, tracked and deployed code; the fragmentation of our testing and manual reporting; an overall lack of integration with our security scanning tools; infrastructure provisioning and more. Our change and release process had gaps and lacked automation. Developers did not have the tools or access needed to be accountable for their applications after deployment to production.

From there, we defined our mission—our North Star for DevOps. We took a very methodical approach, first determining what metrics we wanted to target and then specifically defining our transformation goals.

We needed to standardize solutions and processes to address the fact that we had code in different repositories, across various solutions, each with their own set of processes, including CI/CD. Choosing the appropriate tools for code management and CI/CD was essential.

We landed on a specific tools stack, featuring an open-sourced Git-based approach to code management, leveraging and building upon industry standards. Over the next three months or so, with the help of some in-house developed utilities, we migrated all code from the various locations into a single repository.

Building the pipeline

The next step was building our pipeline. We decided that we



Advice on driving DevOps

If you are looking to transform your traditional IT operation to a DevOps and CI/CD model, there are several things you should keep in mind.

First, it needs to be supported and mandated by leadership, with a dedicated budget and CI/CD maturity targets. This was a vital foundation for us as we brought industry-leading tools and processes in house and integrated them within the Dell ecosystem. This also helped cover the cost of software licensing and the effort needed to establish DevOps in each of the product teams.

At the outset, you need to take a methodical approach to defining what DevOps is for your organization. Set priorities up-front, whether they are cycle-time reduction, incident reduction, increased deployment frequency, etc. Then track your progress in meeting those priorities with specific metrics.

We established a multi-factor maturity score based on metrics for software development standards, security, automation, and more. The CI/CD Maturity Score calculates a percentage representing how well each project is utilizing available automation. This scoring system is, by the way, also a fully automated process that scans the actual project logs looking for the existence of automation in the pipelines, with zero manual intervention and is free of subjectivity.

From a starting point of 35% in early 2021, we expect to reach a maturity score goal of 90% across our organization by the end of this year.

And finally, training is a necessary process that is critical to reaching your DevOps goals. Throughout our transformation, we provided constant training on every aspect of DevOps. The enablement team, made up of subject matter experts (SMEs), paired with many of the development teams to go through their specific use cases and technology stack.

We continue to provide ongoing training as we add new capabilities to our offering. To date, we've successfully trained more than 5,000 developers, product managers and engineering leaders on our DevOps offerings.

All this work is challenging but clearly a worthwhile investment where the return will be quickly realized if implemented correctly. We are now managing more than 30,000 application projects with approximately 17,000 users in our DevOps platform and are currently expanding to organizations outside of Dell Digital. More importantly, our developers appreciate having less manual work so they can focus on what they do best, and our internal customers appreciate that all our digital transformation work means we can meet their needs faster and more flexibly. ■

wanted to build reusable pipeline definitions, called blueprints, that were both flexible and easy to consume. These blueprints would be technology-specific definitions for pipelines that would include all necessary working stages for each automated job. Engineers could reuse existing pipelines, or they could create a new pipeline by pulling elements from existing ones, negating the need to create one from scratch each time.

When consuming a blueprint, the developer need only enter their application-specific information because the blueprints already have the integration jobs built in. For example, security scanning is a required step for all pipelines and as such, the blueprints contain the required security scanning job definition. This means developers only need to define parameters that are specific to their project, paths, names, endpoints and the like.

And to make this selection even easier, we created a CI/CD Marketplace from which the developer can search and discover the most appropriate pipeline jobs from a centralized location. This is a no-code solution that significantly simplifies the creation and maintenance of the pipelines and improves the developer experience.

The result is that developers can now fully manage their code end to end without dependencies on other teams to do it for them, resulting in greater ownership and support for their application throughout its lifetime.

Helping Developers Work Smarter With Our API Marketplace

Find out how our API Marketplace supports the needs of developers both inside and outside Dell.

By Ashok Muthukrishnan, Director, Software Engineer IT – Development Ops

A strong API strategy is essential for any organization that's going through either a business transformation or technology transformation.

A central part of our ongoing digital transformation is an API Marketplace that lets developers share these essential building blocks for connecting people, business and things. Developers don't have to start from scratch each time they used a common construct in a solution in order to access a particular business process or data set. They can just plug in the needed API.

Our API-first approach helps our developers work faster, smarter and with greater collaboration and consistency as they meet our business users' innovative needs.

And last month, we launched an external developer portal to allow our customers' application developers to access a growing list of APIs for Dell products. The [Dell Technologies Developer Portal](#) is a one-stop shop for application developers and DevOps teams to come and explore our capabilities and access to the latest Dell APIs.

Transforming customer connections with APIs

APIs are not new. They are actually many decades old. Over the years, however, they shifted from being standard pieces of software that provided a system-to-system integration to being gateways to connect apps to data and service. The rise of APIs as essential connections to customers began back in early 2000s when Amazon and other large internet companies began to promote APIs as a way to provide cus-

tomers with self-service access to their capabilities.

More traditional companies have come to realize that they can transform their legacy business operations by leveraging APIs to reach customers with new products and services. Fueling the expansion of API use is the fact that more and more of today's customers are demanding self-service access to companies' assets in order to get the personalized experience they want.

Our developer portal is the first time we at Dell Digital have taken APIs from different business units across a business organization—our Infrastructure Service Group (ISG)—and provided a common place for our external developers to come and explore our capabilities. They don't have to engage with a support team within Dell to really understand the capabilities they can integrate as part of a product or a solution. Rather, they can come into the self-service portal to get such access.

Other business organizations are adding APIs to the new portal, which is growing rapidly. The portal is based on our internal marketplace platform, but with a new look and feel and expanded capabilities. We are transitioning our internal developer API ecosystem from our API Marketplace to the new portal as well, to provide them with improved search features and other experience improvements.

Our evolving API marketplace

The value of an API-first strategy is recognized throughout Dell and throughout the industry as essential to being competitive in a digitally transformed world. Over the past several years, our internal API Marketplace has grown rapidly with more than 1,200 production APIs in many categories, ranging from cloud and commerce to supply chain and support.

We began our API effort by creating an internal API ecosystem and then promoting an API-first strategy in IT. The idea was to encourage developers to write an API every time they wrote code for a microservice. The API would contain information on a process or data set that could be reused by other developers writing similar solutions—a sort of building block. The API is the connection that sends information back and forth between a website or application and a user.

We built a customized platform, incorporating some vendor tools, that gave developers a central place to catalogue and share those building blocks—the API Marketplace. Not only are developers able to create solutions more quickly and consistently using APIs, but the platform also provides developers with information on who is using their APIs.

Some 90% of our developers now use our API Marketplace. The number of APIs in our catalogue has grown an average of 24% percent per year.

The result is that our developers are able to create solutions faster, easier and with more consistency. And they regularly share new APIs from microservices, continually expanding the self-service marketplace.

While not every business group across Dell Technologies is API-first, we are making progress and our API strategy overall has matured based on a scoring process that considers multiple factors. We provide tools to help developers improve their API maturity as well as achieve more reusability of the APIs they create.

It was the recognized success of our internal API platform in making our development process more efficient that inspired the creation of the external-facing API developer portal for our Dell customers' developers to discover Dell's Digital asset in the form of API's.

Where do you start

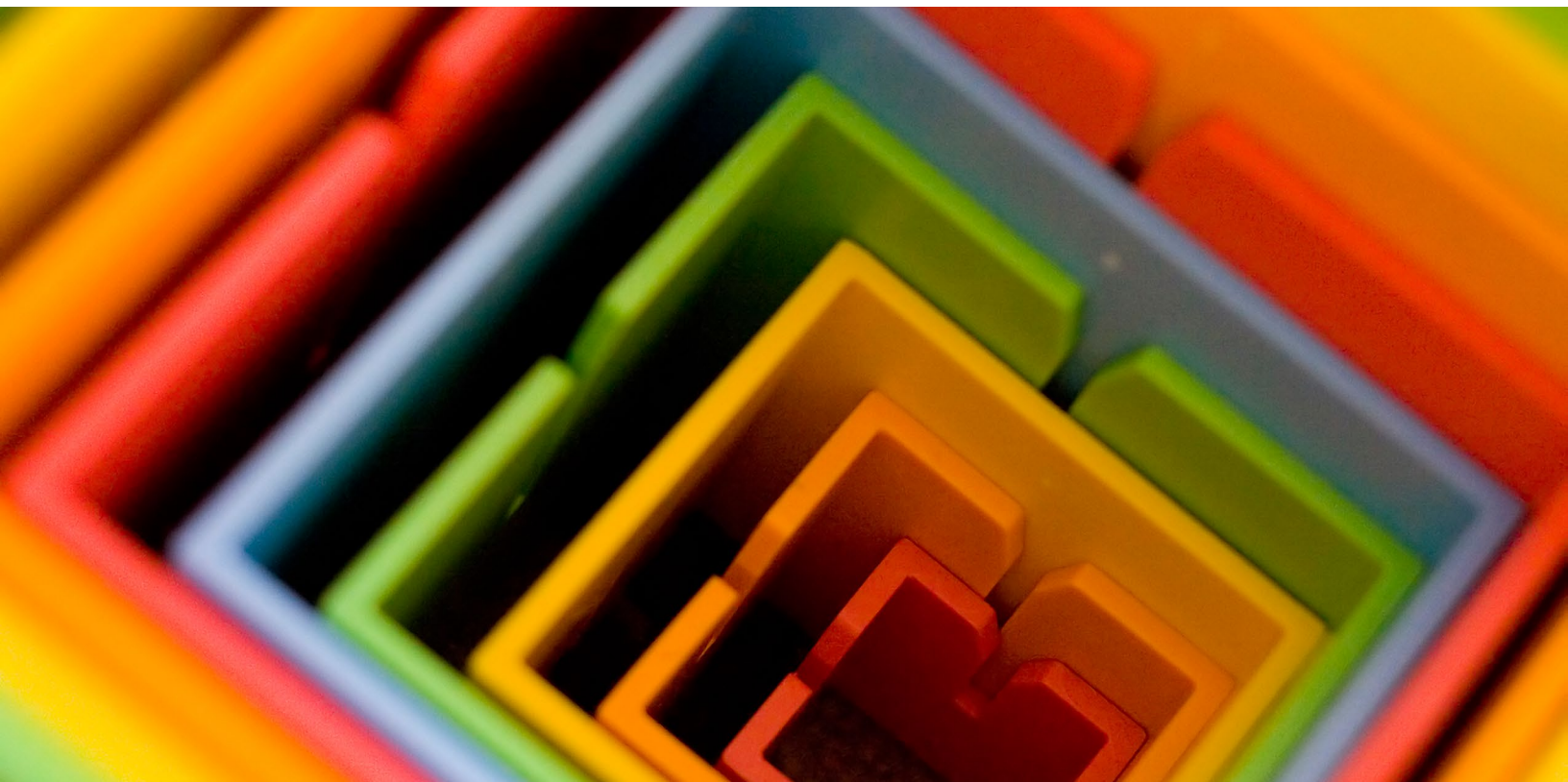
If your organization is seeking to pursue an API strategy, there are a few insights that may help.

First, you should start the process by creating a vision for how APIs can help your company. What are the benefits of providing a platform where you can share your digital assets in the form of APIs internally, externally or both? Clearly define the value, using feedback from developers and other stakeholders.

Once the vision has been agreed upon, build an API platform and create a prototype showing how you can build and share APIs. There are many API management tools available, or you can customize your ecosystem. From there, reach out to developers and business stakeholders to reinforce the merits of an API strategy.

The benefits of an API marketplace are clear. It enables developers to build solutions faster without reinventing the wheel. And organizations in general can readily see the benefits of providing customers' developers with these easy tool sets that they can access themselves rather than sitting with them and explaining.

APIs just make sense to enable developers internally or externally. It is a strategy you can't afford to ignore if your organization is going to compete in a modern digital marketplace. ■



A Shared Building Block Approach to Better User Experience

Learn about Dell IT's UX Marketplace, where developers can share components for consistent webpage experiences

By Nitya Seth, Director-Cross Product Lead- Developer Experience, Dell Digital

Helping developers work smarter and faster with a sharable, building-block approach to creating consistent webpage experiences is central to Dell Digital's modern, customer-centered design focus. And now developers have a central place to access those building blocks. It's called the User Experience (UX) Marketplace, a repository of **micro frontends** (MFEs) and webpage components that can be leveraged by frontend developers to build their user interface experiences.

Dell Digital, Dell Technologies IT organization, launched the UX Marketplace seven months ago to further our efforts to improve our developer experience as well to create a unified customer experience across our digital ecosystem. We now have more than 100 developers utilizing the platform, which has hundreds of components and MFEs so far.

Rather than having to design the same webpage features over and over from scratch, UX developers can now leverage marketplace components and MFEs to gain speed and efficiency. Sharing common design components also ensures webpage design will have a more consistent look and feel.

A platform for sharing

The idea that developers should share UX code began with one business group's move to change its approach to building on-line customer experiences. Dell Digital's Global eCommerce adopted an MFE approach to building webpage features to address the fact that its developers were having to build the same page elements over and over again. It also wanted to create a unified look and feel to its experiences by sharing uniform UX components, smaller building blocks

that could be plugged into a page.

Around the same time that architecture approach was gaining ground, Dell Digital decided to create a central repository where all UX developers could easily catalog and access MFEs and smaller UX components used to create MFEs. The broader focus was to create a centralized clearinghouse to help deliver a consistent experience leveraging Dell Design System standards across the lifecycle of our customer interaction for all of Dell Technologies.

About a year ago, I headed up a small team charged with realizing that goal. The first step in creating the marketplace was to explore options to set up a platform that would meet our needs. We found that a third-party Software-as-a-Service (SaaS) tool offered the features we needed without going through the time and effort to build our own platform tools. We did a proof-of-concept using the vendor platform last spring, working with developers and user communities to get their feedback and add needed features.

Initially, the platform supported only one technology stack commonly used by UX developers. After talking to the Dell development community, however, we found there were multiple technologies they were using and arranged for the vendor to expand support to be more inclusive. Ultimately, we have evolved the platform to support four technology stacks and five testing tools.

We launched the full UX Marketplace for Dell Digital in early August and continue to gain users as well as positive feedback.



Delivers results for developers and users

In terms of developer experience, sharing components via the marketplace empowers them to develop and release features faster and in a consistent way. They can publish and access components such as buttons, product cards, product lists, sign-on elements, check out processes and other UX features applicable across many digital experiences.

Developers had been really frustrated with having to write seemingly a million lines of code over and over again, figuring out details and doing the testing. With readily available components in a single catalog, they can now develop and deliver UX pages in about one quarter the time it took previously. They can put their efforts into more innovative feature development.

The marketplace is part of Dell Digital's effort to expand the adoption of MFE's in creating and maintaining webpages, which allows developers to deploy changes to pages without having to take the page offline for a substantial amount of time, as was previously required. The MFE architecture means that whole page is not just owned by one team anymore, you have four or five experiences sitting on the same page.

When a developer is going to make a change to an experience, they can make the change while the page is still running. Page changes are not tied to the whole page and can be deployed in seconds. And it's not visible to the customer, making webpage changes a very smooth process.

Another feature of the marketplace is that it provides data to developers and designers on who is consuming their components, creating a more collaborative communication channel between them on how best to serve UX component needs.

And finally, our marketplace content has the consistent look, feel and standards set by the [Dell Design System 2.0](#), Dell's evolving design language system which contains the core design and code components that our company uses to build our digital experiences.

Evolving the marketplace

We are working to further the adoption of the marketplace across Dell Digital and beyond, offering weekly education session on various aspects of its use and reaching out to business organizations. We currently have more than a dozen organizations outside of Dell Digital that have expressed interest in the marketplace, which offers value to any group that provides digital experiences.

We are also in the process of building a governance framework, looking into the industry best practices and standards we need to leverage to ensure good quality components hosted in the marketplace. All of this work means an improved developer experience and, if we do it right, leads to a unified customer experience across our digital ecosystem. ■

Equipping Our Developers Inside Dell With Application Intelligence

Dell IT is providing development teams with observability tools to better understand and own the products they build

By Marcio Lena, Director of Application Intelligence Experience, Dell Digital Technology Transformation Services

As **Dell Digital**, Dell's IT organization, has continued our **DevOps journey** and **embraced the Product Model approach**, the role of our application developers has shifted significantly. Developers no longer hand off applications they create to someone else to operate, monitor and manage. They now own their applications top to bottom, throughout their lifecycle. To support this, Dell Digital provides developers with the data, tools and methodologies they need to help them better understand their applications.

Over the past two years, our Observability product team has been evolving our traditional application monitoring process into what we now call Application Intelligence, a much more extensive set of tools and processes to gather data and provide it to developers to analyze, track and manage the applications they build. It helps them gain insights about application performance, behavior, security and most importantly, user experience. It is data that developers had limited access to before, since they had no reason to monitor the operations of their applications in the old operating model.

As developers began to use more data to evolve their **DevOps operating model**, a key challenge we had to overcome was the high level of industry fragmentation. Multiple and overlapping tools were forcing developers to change context frequently, ultimately impacting productivity. To reduce fragmentation, we have created a simplified and streamlined experience for developers to easily access application intelligence via automated processes in our DevOps pipeline, as well as self-service capabilities for them to instrument their own applications.

By alleviating the friction previously caused by using multiple tools, different navigation flows and different data and

organizational structures, we are enabling developers to spend less time trying to find where performance issues are and more time on adding value.

Making a cultural change

Transforming our observability portfolio to meet developers' needs required both a cultural and technical strategy.

We started with the cultural change, educating developers on where they could find the data they need to accommodate the fact they now own their apps end-to-end. Under the product model, where our IT software and services are organized as products defined by the business problems they solve, developers are part of product teams responsible for their solutions and services from the time they are built throughout their entire lifecycle. Giving them the data to better understand their products helps developers deliver higher-quality code in less time to meet the rapid delivery that the product model demands.

That was the first step to opening doors for developers with access to data that was highly restricted before. My team always had extensive observability data but fiercely controlled it because that was the operational process.

But now, we gave developers the freedom to see what was going on for themselves by ensuring the data was available for everyone in a shift to data socialization.

We spent a lot of time talking to developers about what they were looking for, what was missing and what technologies worked for them. We asked them things like, "when you wake up at three a.m. because someone called you with a priority one issue and it's raining fire on your app, how do we make your life easier during that moment?"



Self-service was the first thing they asked for. They don't want to have to call us anymore to get the information they need. So, in a crucial first step, we gave developers self-service use of the tools that they need to get their job done without asking us for permission.

We also changed the focus of how we look at application data to meet developers' needs by making the application the center of our observations. In our traditional monitoring process, our focus was a bottoms-up view of the environment, which assumed if the infrastructure—the servers, CPUs and hard drives—was working, the app was working. However, that is not always the case.

Two years ago, we began a top-down data view, looking first at whether the app is working and drilling down to infrastructure functionality from there. The application is now the most important thing in our observability process.

Streamlining and simplifying tools

In the second year of our transformation, we focused on creating a centralized app intelligence platform, consolidating the number of tools we use and radically simplifying the application data process. We chose to utilize three major vendor tools that complement each other to stitch together the data collections and let users see what's happening across the stack, from traditional to cloud.

One key tool we use offers out-of-the-box, real-time observability of all applications across our ecosystem in detail, including which apps are talking to each other. It automatically gets the data out of the applications for users. Developers just need to put a few lines of code in the DevOps pipeline for cloud native applications and it automatically maps everything out. Because it spans production and non-production environments, developers can even understand how well their applications are working while they're coding them.

By enabling access in the DevOps pipeline, developers get continuous feedback on every aspect of their application, including performance, how their customers are using their application, how fast the pages are loading, how their servers are working, which applications are talking to their applications and more.

With this tool, they can actually go top to bottom, on every problem that is happening in the environment, for any application.

This detailed platform works alongside a second tool that provides a longer-term executive, birds-eye view of the environment. This tool provides developers and **our Site Reliability Engineering (SRE) teams** the ability to customize their views to exactly match how the applications are architected. We also deliver real-time data to our business units enabling them to leverage the information that we regularly collect.

If a dashboard sends an alert about a problem, developers can use the more detailed tool to see exactly what the issue is. The solutions are used in combination to provide visibility and quick access to information from an ecosystem view all the way down to the transaction level.

The third tool we offer is an open-source-friendly stack that provides a virtualization layer that reads data from anywhere and provides users with sophisticated dashboard capabilities. Developers access this tool from Dell's internal cloud portal rather than the DevOps pipeline, since it requires users to design how data is presented.

As we transitioned to our streamlined portfolio in the summer of 2021, we actually re-instrumented all the critical Dell systems in 13 weeks, with no production impact. The transition, which would normally have taken a year and a half, was completed just in time for Black Friday.

In addition to the efficiencies gained through automation and self-service access to these platforms, developers are also able to instrument their applications by themselves, giving them the freedom to define and implement what works best for their application.

Overall, we have given developers a toolbox they can use to get the best results in their specific roles in the DevOps process without having to get permission or wait for IT. We are also continuing to improve our offerings based on ongoing feedback from developers. Or like we always say: Build with developers for developers. ■

How Dell Improved IT Developer Productivity by 275 Percent

Learn about Dell IT's use of standardization, automation and self-service to create a productive development framework

By Jaynene Hapanowicz, CTO & SVP, Dell Digital – Technology Transformation and Services Development

If you have any doubts about how vital it is to improve your IT developer experience by freeing developers from the administrative and manual tasks that drag down their ability to create the software solutions you need, let me ask you a seemingly off-topic question. If you had a manufacturing facility that was running at only 16 percent productivity, what would you do? The most common response I hear is, “Shut it down.”

But this is exactly what's happening with developer productivity. A recent IDC survey* of 2,500 developers showed that they are only spending 16 percent of their time writing the functional code that is so essential to their organizations' digital transformation. The other 84 percent is taken up with manual and fragmented development processes.

Technology organizations are the lifeblood of all digital transformations. It's not an option to “shut it down,” as a well-performing organization is the future of your company. But you can take a page from the manufacturing industry, which revolutionized its productivity and efficiency in the 1990s by using new technologies to standardize and automate processes vital to survival. You can also draw inspiration from the sales industry, which gained efficiencies in the 2000s from advancements in CRM technologies, eliminating tedious and repetitive tasks.

You could say IT is entering its own era of standardization and automation, driven by the fact that digital transformation is dependent on developers being able to fully optimize the use of their time.

Dell Digital, Dell's IT organization, has taken on that challenge over the past several years, after determining that

*IDC PaaSView and the Developer Experience 2021

our own developers were bogged down with administrative tasks, spending only 20 percent of their time creating the software our digital transformation depended upon. Since then, we have cracked the code for a world-class developer experience, freeing our developers to spend 70 to 75 percent of the time writing functional code.

Our success centers on a layered approach to improving the developer experience, including a fully automated cloud, with a catalog of robust services and a fully automated DevOps pipeline.

Building the layers

We built our world-class developer experience with a relentless focus on standardization and automation, driven by our infrastructure and DevOps teams working in tandem.

At its foundation is a centralized self-service marketplace within Dell's private cloud for developers to consume automated infrastructure services. We created the private cloud marketplace by putting a layer on top of Dell Technologies private cloud infrastructure platform which runs on Dell's hyperconverged infrastructure (HCI), namely Vx-Rail and PowerFlex, at the heart of our modern data center.

Working closely with development teams, we built a self-service catalog that lets developers choose and deploy the capabilities they need—compute, network, database, etc.—in minutes rather than days.

The private cloud marketplace was a starting point, not an end game to maximizing developer experience. From there we built a standardized DevOps CI/CD pipeline that eliminates manual intervention between code creation and product deployment and support. Developers have everything at

their fingertips to manage their application lifecycle autonomously. They can use the pipeline to automatically provision infrastructure; perform functional, performance and security validations; run code quality checks and deploy to production with integrated monitoring and application tracing.

For the next two layers of our approach, we created an API Marketplace and a UX Marketplace where developers can “shop” for shared APIs to fit development needs and for micro front ends (MFEs) and webpage components to build user interface experiences. The API Marketplace is part of our API-first approach in which developers can consume and repurpose assets via this self-service portal. Similarly, the UX Marketplace lets developers leverage sharable, building blocks to create consistent webpage experiences.



And finally, we measure our CI/CD progress, application maturity, application performance, resiliency and more via our DevOps dashboard. Through aligned KPIs we track our progress with granular analytics and reporting to provide feedback to our developers.

The benefits of our layer cake approach to improving our developer experience are clear. Developer productivity has increased by 275 percent. Quality is up and the time it takes to address defects is dropping.

Start simple with little wins

If you are taking on the challenge of improving your developer experience, there are a few basic points you should consider.

First, keep it simple. Your problem statement to solve is that your developer productivity needs to go from 16 percent to 80 percent. It's not about the number of automations you build. It's not about anything other than fingers on keyboards and expanding productivity time.

Next, realize that much of what you need to transform is focused on people more than technology. For us, that meant shifting away from the traditional way IT was organized in gigantic teams with multiple handoffs to a product model where dedicated teams could focus on building independence and capabilities within product areas.

And lastly, realize that forging a new developer experience is an iterative progress, not a big bang one. Don't underestimate the importance of making little steps toward a broader goal of improving developer experience. You break the inertia by getting little wins, and then before you know it, you look back and you have enough critical mass and you've got enough of the organization behind you that it starts to drive the organizational change.

A seat at the table

Dell Digital has a longstanding tradition of using our own innovative resources and technology to provide first-hand experience and insights in addressing the same problems our customers are facing. In this case, the framework and tools we have created to improve our developer experience is helping to shape the solutions and services that Dell offers to our customers to enable their digital transformation.

In a broader sense, the changes happening within Dell Digital are part of a more extensive shift in this age of technology. Rather than an organization that is often the focus of complaints, IT has become a catalyst for driving change within the company in a world where technology-led organizations are thriving.

At Dell and elsewhere, IT now has a seat at the table and a role in leading company changes via software innovation. ■

Our Developer Focus Continues

Dell Digital is continuing to evolve our DevOps processes and technology to free up and empower our developers to do what they do best—create the innovative solutions our fast-moving digital world requires. We will also continue to share our experiences applying automation, leveraging APIs, and adopting technologies like AI and machine learning to build a more agile, resilient, and innovative IT organization.

We invite you to stay up to date on the latest inside Dell Digital by reading the [Dell Technologies blog](#) or by visiting [Our Digital Transformation](#).



 **DELL**Technologies