HPE
POINTNEXT

Transform to rapid cycle application delivery

A blueprint for cloud-native IT

HPE POINTNEXT BLUEPRINTS: HOW TO GUIDES FROM SERVICES EXPERTS OF HEWLETT PACKARD ENTERPRISE

TABLE OF CONTENTS

EXECUTIVE SUMMARY 02
Application owners have a new, critical challenge

CLOUD-NATIVE IS DIFFERENT IN A BIG WAY 03
"Cloud-native" refers to using tools and methods to develop applications that operate in a cloud environment

THE JOURNEY TO OVERCOMING CLOUD-NATIVE CHALLENGES 06
Each organization navigates their own journey to improving application delivery cycle time

GUIDING PRINCIPLE 1: 07
Define the right application architecture based on current business needs

GUIDING PRINCIPLE 2: 09
Put in place tools and skills to transform and operate new business model

GUIDING PRINCIPLE 3: 12
Move to rapid release cycles using automation

GUIDING PRINCIPLE 4: 15
Design, recommend infrastructure for cloud, and microservices development

JUMP STARTING YOUR ADOPTION OF CLOUD-NATIVE 17
Recommendations for achieving the full benefits of cloud

HPE POINTNEXT 19
The expertise needed to help you make Hybrid IT simple

CONCLUSION 20

Executive summary

Application owners have a new, critical, challenge. Their business customers now expect continuous innovation through cloud-based applications with rapid development cycles.

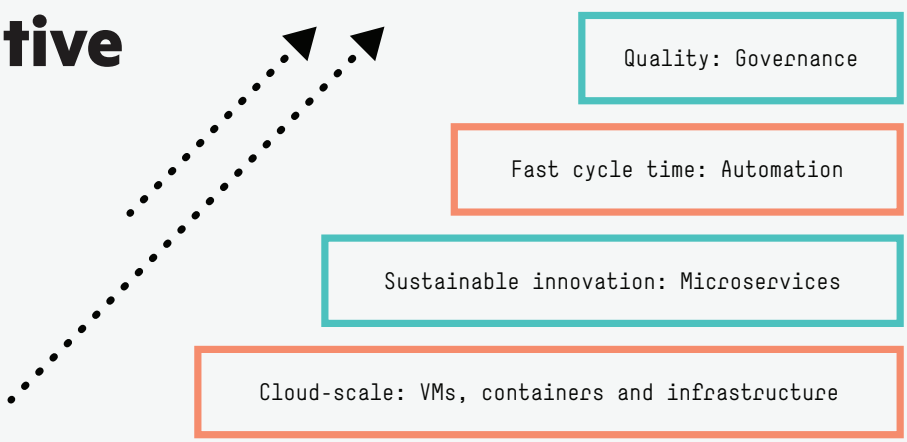
The answer cannot come from annual upgrades of legacy workhorse applications. Instead, IT leaders must quickly evaluate whether to re-architect their applications for cloud or develop new applications using cloud best practices such as DevOps, containers, automation and microservices.

The need to bridge the gap between traditional IT and cloud-native applications is vital. Upgrading applications or developing new ones to run as "cloud-native" requires a broad set of new components to work together and an understanding of workload requirements. It also demands an architecture that departs from traditional enterprise application design. HPE Pointnext, services for Hewlett Packard Enterprise, has enabled many of the world's largest companies to realize results such as these:

- Release preparation processes from 3 months of manual preparation down to 20 minutes
- Deployment of applications from 5 hours to 2 minutes
- Reduction in SaaS deployment time from 3 weeks to 90 minutes
- 75% productivity boost in IT operation, 40% cost saving in IT operation and TCO
- 38% reduction in operational expenditures
- 50% reduction in data center footprint

The cloud-native advantage

Flexible, fast, scalable



Adopting a cloud-native architecture reduces risk and minimizes time and money spent delivering business value. In some cases, organizations can reduce costs by nearly 90%, while improving software quality, security and compliance.

This guide examines the critical business and technology best practices for enabling cloud-native applications—from DevOps practices to underlying infrastructure. Based on HPE Pointnext cloud-native projects from around the globe, this HPE Pointnext blueprint helps IT teams accelerate the development and delivery of cloud-native applications. By adopting new technologies and processes, IT organizations can improve quality and streamline integration within development/testing and deliver predictable results in production.

Cloud-native is different in a big way

"Cloud-native" refers to using tools and methods to develop applications that operate in a cloud environment. These applications are found in the cloud, but can also be used in the data center. They are typically containerized, dynamically scheduled and micro services-oriented.

You can usually move legacy applications hosted on dedicated platforms to cloud-native environments with some re-engineering to support cloud computing infrastructure. Doing an up-front analysis of the architecture and examining destination options for your workloads can pay off quickly. These monolithic applications typically run on a single system and rely on hidden APIs. As such, they run as a complete unit with rigid upgrade and deployment processes that are hard to accelerate. In some cases, a "lift and shift" project to a cloud platform can provide immediate benefits. Eventually converting such applications to container-centric deployment sets the stage for conversion to a true cloud-native approach.

Conversion to micro-services
with containers

Implementing cloud-native applications in a microservices architecture allows each element of functionality to be put in a separate service container. This approach has many benefits such as:

- Cloud-native applications combine fail-fast error management with horizontal scaling
- Containers and platforms enable autonomous scaling and repair, expedite developer builds and improve delivery consistency
- Service-oriented applications and microservices address security and quality objectives
- Fully automated testing and deployment allows dozens of deployments per day
- Development cycles can be automated, API- driven, agile and accelerated

The reward is a controlled environment of cloud-centric, agile, application development and releases, accelerated application delivery and automation tools that improve results. With it, businesses can adapt to market changes quickly, seize new opportunities before competitors, improve reliability and scale rapidly.

But like all applications, cloud-native applications need a capable team to deliver them. Most IT staff, analysts and developers are still learning the steps, processes, tools and procedures to effectively deliver these applications. They need help and guidance to address the issues and challenges unique to the cloud-native application architecture.

> 57% of users are dissatisfied with the release cadence.
>
> **- gigaom**

Cloud-native adoption
A retail financial enterprise example

HPE Pointnext worked with a financial institution to transform their delivery strategy to retain customers with an innovative service experience, attract new customers with a redesigned sales experience, integrate social marketing to drive brand perception, and gain a 360-degree customer view to improve "always-available" customer satisfaction.

ASSISTANCE INCLUDED:

Evaluation of needs:

- Assessed and proposed the best fit platform and architectures to integrate continuous delivery into their processes and define the supporting environment

Improve skills:

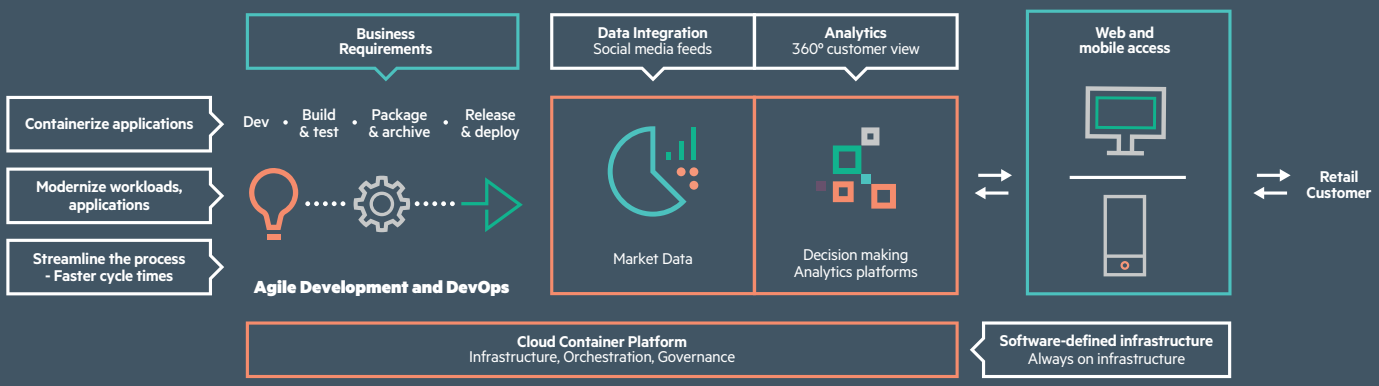
- Educate the client on cloud-native development practices
- Train teams on cloud-native application architecture

Support deployment:

- Accelerate apps development
- Deliver an online customer loan application
- Integrate development tools with continuous integration
- Design and build out the DevOps environment

Results: HPE Pointnext helped the customer meet their immediate goals and objectives as well as position them for future development and growth. Using this new business model gave them faster to market, quality, cloud-based applications, and fresh online banking services.

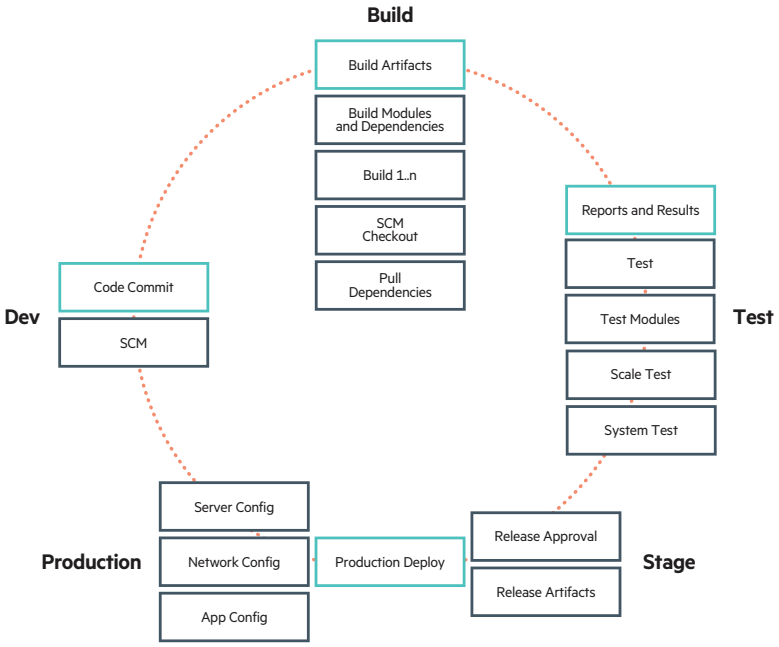
Retail financial enterprise - cloud-native adoption



The Journey to overcoming cloud-native challenges

Each organization navigates their own journey to improving application delivery cycle time. Recognizing the development, operational and infrastructure challenges that cloud-native architectures pose is the first step in addressing them. Below are the challenges we typically hear from our clients most relevant to cloud-native adoption.

- Need a way to incrementally evolve from monolithic applications—these applications can hold the business back
- Need the skills and tools for cloud-native development to eliminate risk and accelerate adoption
- The current application release cycle is much too slow to respond to business demands
- IT Infrastructure is too inflexible to enable rapid change and deployment



Addressing these four challenges is the purpose of this HPE Pointnext transformational blueprint "Enable Cloud-Native / Microservices Development," described below. Each challenge is addressed based on a "Guiding Principle" approach that includes an overview, a checklist of best practices and expected results. Note that the investment in each Guiding Principle varies by the needs of the organization. Tasks derived from any Guiding Principle can take priority.

Taken together, they enable organizations to recast the design–build–test–release–run cycle across people, processes and infrastructure to be consistent with cloud-native requirements.

Guiding principle 1: Define the right application architecture based on current business needs

Immediate architecture needs can vary from simply focusing on the performance of the application in production, to developing a cloud-native application from scratch. Cloud-native development is commonly based on containers and microservices or on virtual machines (VMs). Most customers use VMs today. However, containers offer efficiency and speed advantages over standard virtual machines.

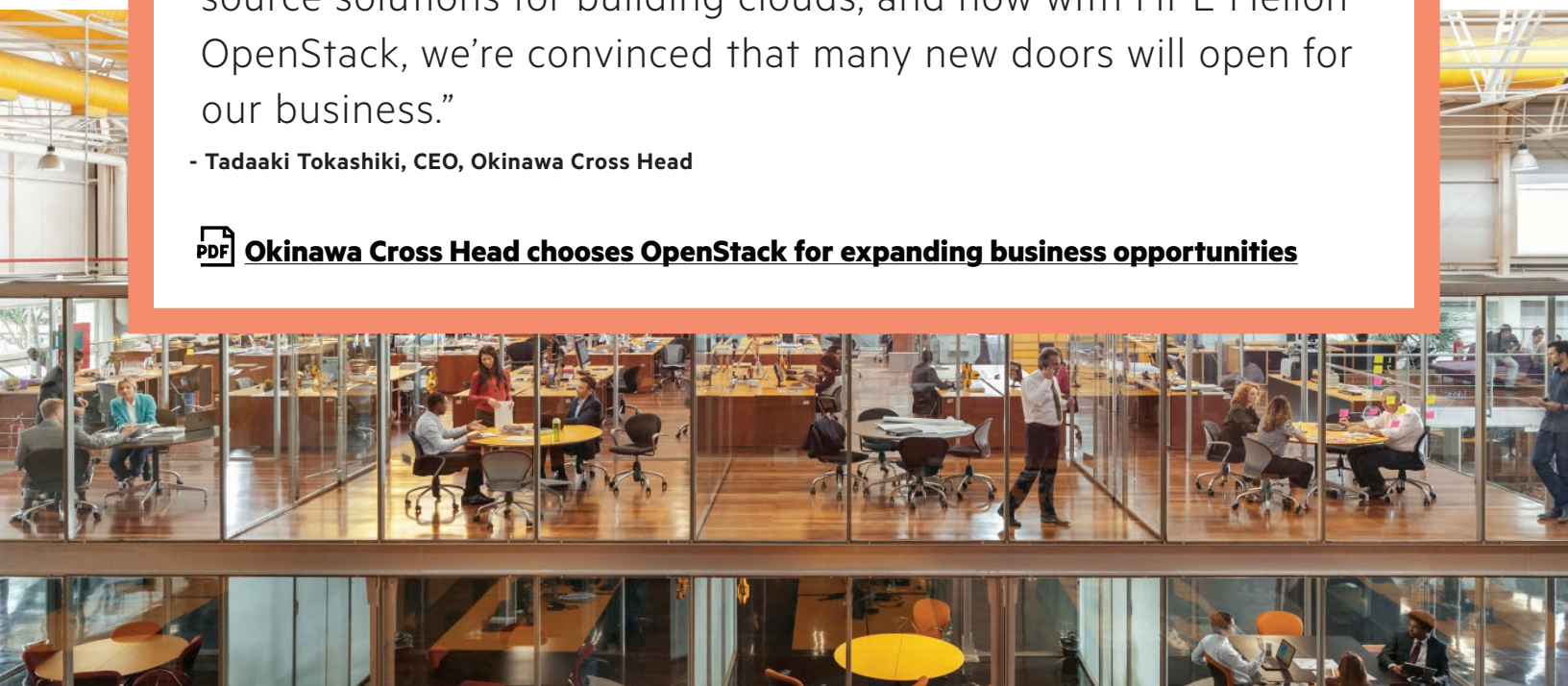
Microservices allow the architect to define an application as a collection of discrete services, with each service providing tightly defined business capabilities. Each microservice can be deployed, upgraded, scaled and restarted independently. When this architecture is applied across design, development and deployment, the utilization of the infrastructure resources can be as much as 70% greater, translating into infrastructure savings. However, microservices architecture must be planned carefully to achieve these results.

BUSINESS CHALLENGE: Need a way to incrementally evolve our monolithic applications to a cloud-native environment—these applications can hold the business back.

"Okinawa Cross Head is an active supporter and user of open source solutions for building clouds, and now with HPE Helion OpenStack, we're convinced that many new doors will open for our business."

- Tadaaki Tokashiki, CEO, Okinawa Cross Head

Okinawa Cross Head chooses OpenStack for expanding business opportunities



Checklist:

1. Capture the application requirements as the first critical step in cloud-native development. Applications require services, monitoring, logging and data persistence; and they need to be architected for the cloud with a service-oriented approach. This includes the ability to scale horizontally, fail well and provide a more seamless approach to SLA adherence utilizing cloud application techniques.

2. Assess the organization's current application workloads, development model, release and product cycles as well as their current infrastructure, capabilities and future vision to determine the best fit architecture and appropriate microservices options. Ensure they can align to the 12-factor approach. This analysis is often called "value stream mapping," which identifies where time is lost in complex processes.

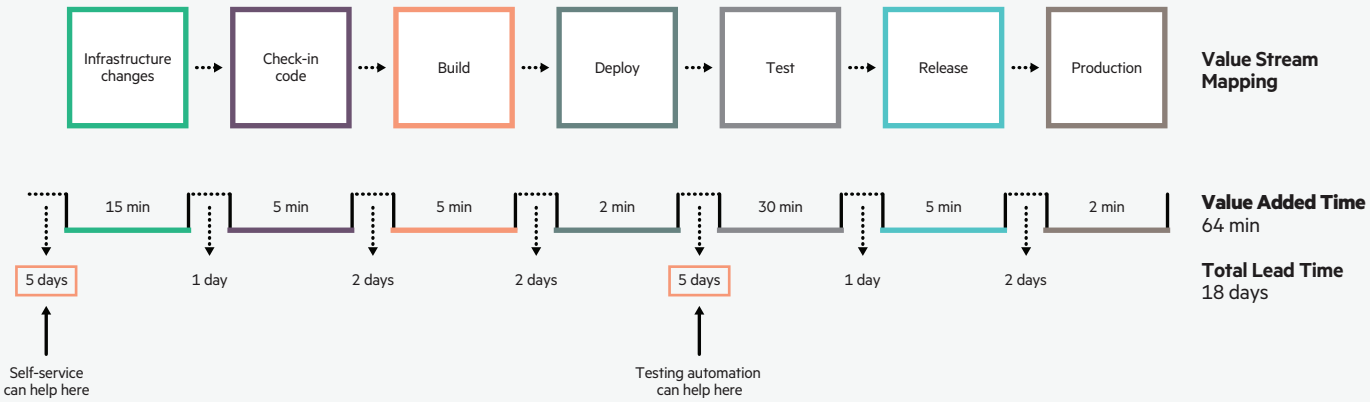
3. Evaluate the composition and support of the application to deconstruct it into microservices. Then identify the "seams" in the application from which to carve out the microservices. Map services to containers and select infrastructure platforms based on application lifecycle, update intervals and compliance requirements. Base architecture choices on the trade-off between developer productivity and the expected application scaling ramp.

4. Establish initial guidelines and standards for the application architecture and update them based on what you learned from each sprint. Decide which open-source tools or vendor based applications and toolsets are needed. Deliver service with the proper tools and integrated development processes.

Expected Results:

- Applying value stream analysis will reduce cycle time to deliver innovative products and services.
- Conversion to microservices will reinvigorate legacy app investments by simplifying integration with new applications such as mobile, consumer facing applications.
- Applying lessons learned from pilot programs can yield cost reductions approaching 90%, while improving software quality, security and compliance.

Identifying delays and bottlenecks in a process or pipeline through Value Stream Mapping



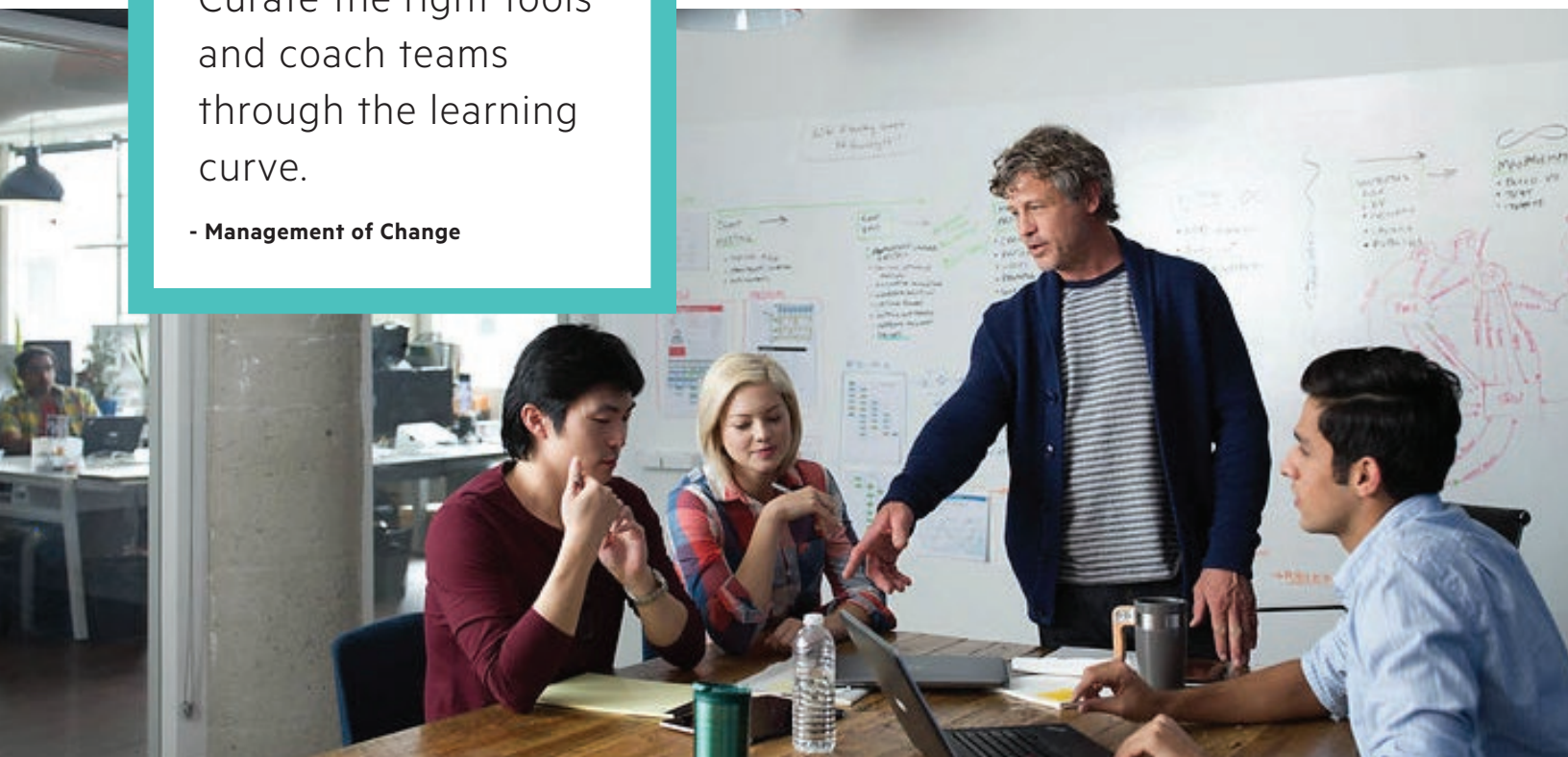
Guiding principle 2: Put in place tools and skills to transform and operate new business model

Business and IT personnel must work together to prioritize the legacy and cloud application workloads to determine the technical feasibility, strategic importance, direction and ROI of a cloud-native direction.

Development and operations teams must be taught to collaborate better, manage change, support new operational processes and effectively deploy new infrastructure and applications. Microservices are not only about the technical architecture, but also about the organization. For example, teams that work on each microservice are smaller compared to monolithic application models.

BUSINESS CHALLENGE: Need the skills and tools for cloudnative development and eliminate increasing risks that slow adoption of the current business applications in this environment.

> Curate the right tools and coach teams through the learning curve.
>
> **- Management of Change**

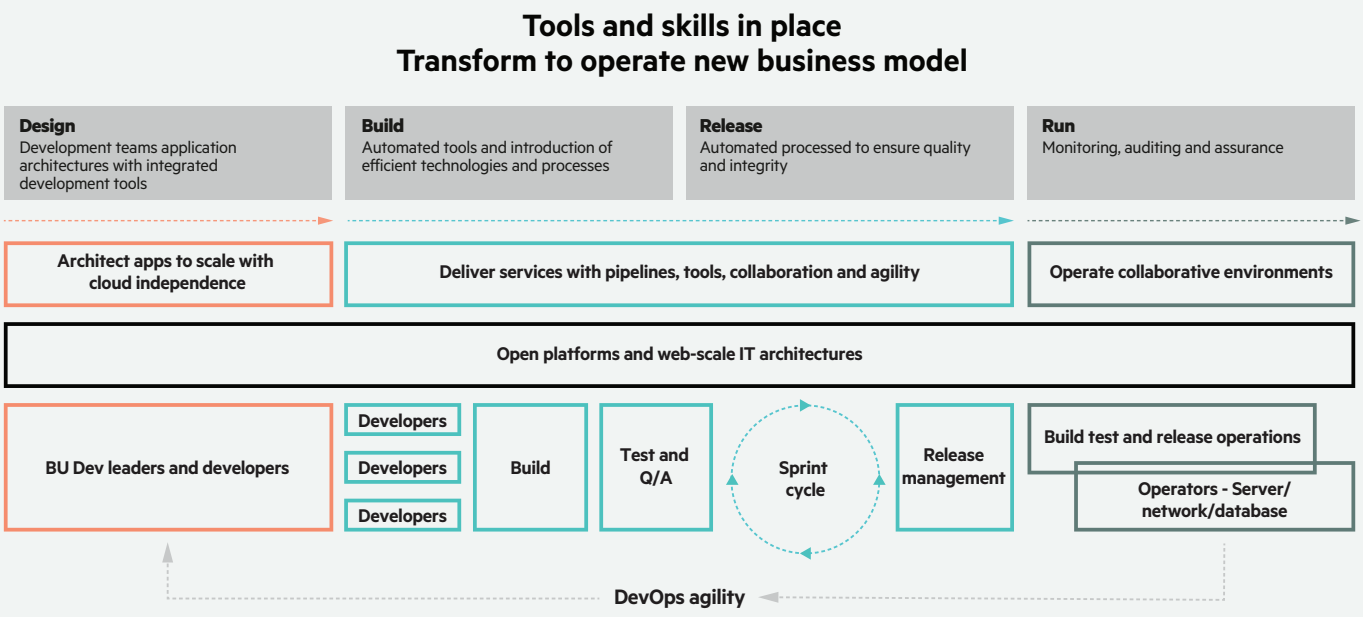


Each team for each microservice is typically composed of developers, testers, product managers and related personnel. HPE Pointnext recommends focusing on developer productivity with a curated PaaS platform, and then building the skills and teamwork in place to make best use of it.

Having the right skills, tools and teams is a vital step that must be met for success. Improving and maintaining custom development operations, along with continuous integration (CI) and continuous delivery (CD) to improve performance (the toolchain approach) must be a priority. Growing and building the business while improving architectural governance, operational continuity and risk avoidance may require using outside resources who are properly training and skilled.

With an integrated toolchain and IT operations automation, teams can focus on driving business priorities. They can lead process improvement and automation, delivering value to the business. This includes monitoring and managing all the cloud-native operational platforms and applications. Many open-source tools are available to streamline development operations making it easier to support and manage the platforms. In this way, infrastructure automation can eliminate downtime due to human error.

> Moving to a new business model drives process change at every step in Design, Build, Release, Run. Management of change investments are required to mitigate organizational friction and improve collaboration.



Checklist:

1. Assess organizational needs and put in place the tools and skills necessary to transform to and operate within the new application development, release and business model. Make sure the knowledge base, integration, automation steps, and builds processes are all working together seamlessly.

2. Curate best fit tools and technologies including open source. Add more agile development tools and CI / CD processes. Provide training on these tools and coaching on best practices around items such as sprints, QA, build, test and release processes.

3. Invest in management of change to lock in these processes, skillsets, tools and development to ensure buy-in and success. Building out DevOps skills and organizational effectiveness training can help here.

4. Deploy a trial with a key business application and a small team. Evaluate lessons learned and have those team members lead new teams to propagate change organically.

Expected Results:

- An integrated toolchain, supported by training, simplifies delivery of new applications and services.
- Automation of operations ensures applications and code work as planned with each release and generate instant feedback.
- Pilot approach generates buy-in and enables changes to meet organizational requirements.

Guiding principle 3: Move to rapid release cycles using automation

Automation can take friction out of the application lifecycle and improve release governance.

Models such as DevOps with well-defined responsibilities and common incentives, tools and processes enable collaboration between development and operations functions, to speed and transfer finished app code into production. HPE Pointnext notes that this is a challenging and difficult step, as it requires collaboration across the entire development and production release cycle.

Shortening the application development and release cycles can reduce time-to-market and investments in market testing. For example, a typical waterfall release is every 6-18 months, which is too slow for today. This leads to higher TCO producing between 200% - 400% more monthly spending than necessary for development and production.

BUSINESS CHALLENGE: The current application release cycle is much too slow to respond to business demands. There needs to be something faster and better.

> Automate the release process, reduce scope and impact of changes, and increase frequency.
>
> **- Management of Change**

Checklist:

1. Structure the development environment to optimize developer efforts and reduce time spent producing code. Reducing manual steps and the cycle times between builds is essential to developer productivity and integrated development environments. Testing is also key in decreasing development cycles, failure injection and integration times. This can be automated and tailored to CI/CD practices to improve performance, reduce bad code, and speed release cycles via instant feedback.

2. Evaluate current release processes and adapt to a better, more effective and controlled cloud-native continuous driven release cycle with more reliable quality code. Accelerate the application delivery process and automate it.

3. Evaluate current compliance and audit requirements and adjust accordingly with the cloud-native process. Automate this using tools identified through Guiding Principle #2.

4. Work with the organization and guide the automation procedures and steps based on their current processes and future requirements. Evaluate and implement orchestration and management engines and monitor their success. Position investments for the next wave of new application requirements to meet the business needs.

5. Promote collaboration between development and operations teams. They must work together to be successful. To do this, gather a small team to work together on a pilot project and have them list lessons learned on what happened and didn't and steps to amend any issues or challenges.

6. Evaluate release history for significant error / failure events to prevent reoccurrence and apply this to the app release cycle and processes. Eliminate or correct any processes that introduced bad code.

7. Implement robust automation steps and procedures to handle the application release cycle. For example, allocating and reallocating containers, upgrades and services. Evaluate performance, enhancing architecture and infrastructure based on application needs as they happen. Automate repetitive processes to improve quality, auditability and speed. Automation can take friction out of the application lifecycle and improve release governance. Models such as DevOps with well-defined responsibilities and common incentives, tools and processes enable collaboration between development and operations functions, to speed and transfer finished app code into production. HPE Pointnext notes that this is a challenging and difficult step, as it requires collaboration across the entire development and production release cycle.

Expected Results:

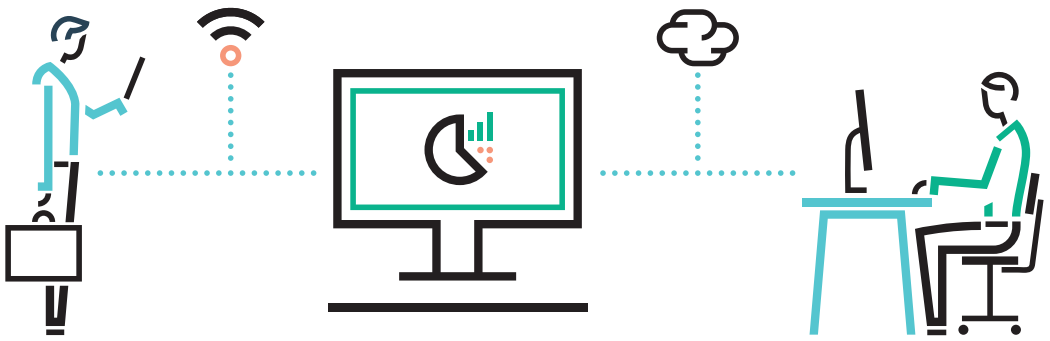
- Deploying the CI/CD process yields faster time-to-market while maintaining compliance.
- Task orchestration engines will improve performance and developer productivity, and enable more efficient collaboration across teams.
- Automation and integration of the development and test environment will reduce human error, risk and cost.

Creating an agile supply-chain platform

The HPE portfolio of private, public, and hybrid cloud products and services is core to Fox media and content operations. Innovating with these technologies, Fox now has an agile, scalable, cost-effective, digital supply-chain platform. Fox accomplishments include:

- Automated over 70 digital media scenarios across our global Theatrical, Home Entertainment and Television businesses.
- Added over 3 million assets in the content repository, distributing over 150,000/month.
- Passed the 1.3 Exabyte per year mark for content sent, received, downloaded, and uploaded.
- Rebuilt and expanded our global network footprint, covering 149 offices, 95 cities, and 44 countries across six continents.

20th Century Fox finds their right mix of cloud



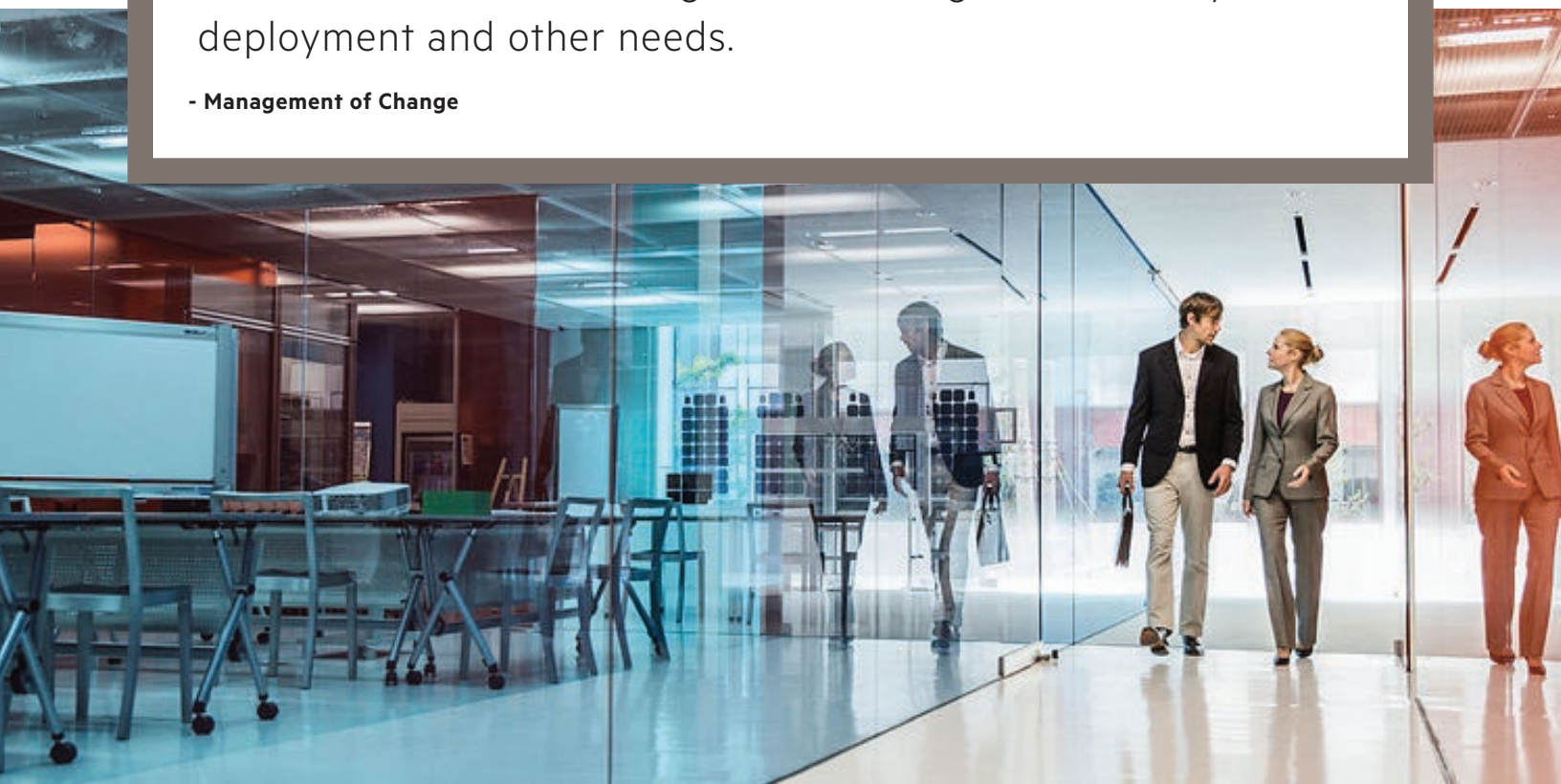
Guiding principle 4: Design, recommend infrastructure for cloud, and microservices development

The cloud-native approach for new applications forces the business to reshape processes and procedures to catch up.

For example, a software-defined architecture enables rapid deployment to the underlying infrastructure to accommodate application needs and production deployment processes. This approach is API-based to rapidly adjust to new demands.

BUSINESS CHALLENGE: IT Infrastructure is too inflexible to enable this rapid change and deployment.

> Adopt an appropriate IT infrastructure to match the cloud-native architecture with regards to scaling, data security, deployment and other needs.
>
> **- Management of Change**



This architecture supports cloud-native applications through the dynamic, centralized, automated control of the infrastructure. This satisfies the needs of applications, IT and business users. Instead of monolithic high-availability systems historically favored in data centers, hyperconverged architecture spreads resources across hundreds or thousands of nodes to support simple scaling and recovery.

Checklist:

1. Create and communicate changes to processes to minimize risk and improve the infrastructure. This approach will ensure platform performance, software availability, security, business continuity and always-on capabilities are priorities. To achieve this and meet demand, make sure the development cloud has customized capabilities, features and service to provide minimize risk in change and to aid in service delivery throughout the software lifecycle and platform performance requirements.

2. Foster collaboration between applications and operations teams to jointly design the new infrastructure to accommodate the cloud-native and microservices development. For example, software-defined or hyperconverged container as a server infrastructure that can support VM "vending machines", containers as servers and microservices as well as bare metal is ideal. Be sure to consider migration issues.

3. List performance requirements and use these with containers and platforms. Define the performance requirements and areas of improvement for taking traditional applications and converting to cloud. Also include infrastructure agility requirements—rapid provisioning, re-provisioning and scaling.

4. Evaluate any migration plans to move legacy applications to a cloud-based application. What is coming up? What did not work and why? What is planned for future migration? Traditional monolithic to cloud-native?

5. Provide suitable applications, tools and processes to support and monitor the production environment. Apply rapid recovery processes and synergies.

6. Define the right mix of infrastructure to accelerate time-to-value from applications and data. Typically this means transforming to a hybrid IT environment. HPE Pointnext notes that the right mix of infrastructure, defined to address core workload types is key. The right mix approach opens the door for simplifying operations, speeding up workloads, lowering costs and heightening efficiency. These contribute to rapid creation of new business models and revenue streams.

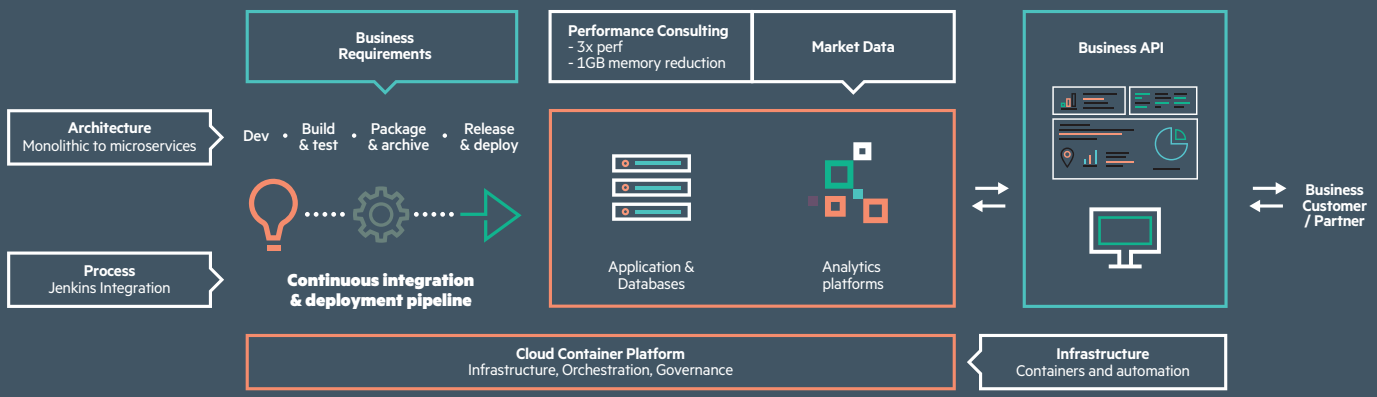
Expected Results:

- Focusing on risk management ensures optimized infrastructure and processes that align with core workloads.
- Coordination between supporting teams delivers an efficient, timely migration to the new environment.
- Planning for the right mix and key workloads will reduce infrastructure costs with faster deployment cycles.

A B2B E-commerce solution
Architecture and Infrastructure Services

HPE Pointnext worked with a customer via an assessment workshop to discover the current application processes around the requirements, architecture, infrastructure, application development, release cycles, tools and operation procedures.

Results: HPE Pointnext helped the customer adopt a set of process improvements, put in place a high-level roadmap and begin automation. Taken together, they provided quicker, higher quality releases, fewer compute errors, and rapid recovery, yielding higher end-user satisfaction.



Jump starting your adoption of cloud-native

As we have discussed, adopting the cloud-native process means a major shift in the Design, Build, Release, and Run software application development paradigms. To achieve a full benefit of the cloud, adopt a Software Defined Infrastructure (SDI), a Cloud-Native Application Architecture, and a Continuous Deployment paradigm supported by composable infrastructure. HPE Pointnext recommends:

Get the application architecture right

- Evolve existing monolithic applications and adopt cloud-native methodologies to better scale applications and capabilities with elasticity and availability. Define and adopt a set of application architectures that meet the principles of 12-factor development.

Reboot skillsets and tools

- Improve developer and development productivity by curating the right toolsets and the skills to best apply them
- Provide services to adopt and thrive with containers and cloud-native development, allowing developers to define their strategy and integrate at their own pace

Speed up application release cycles

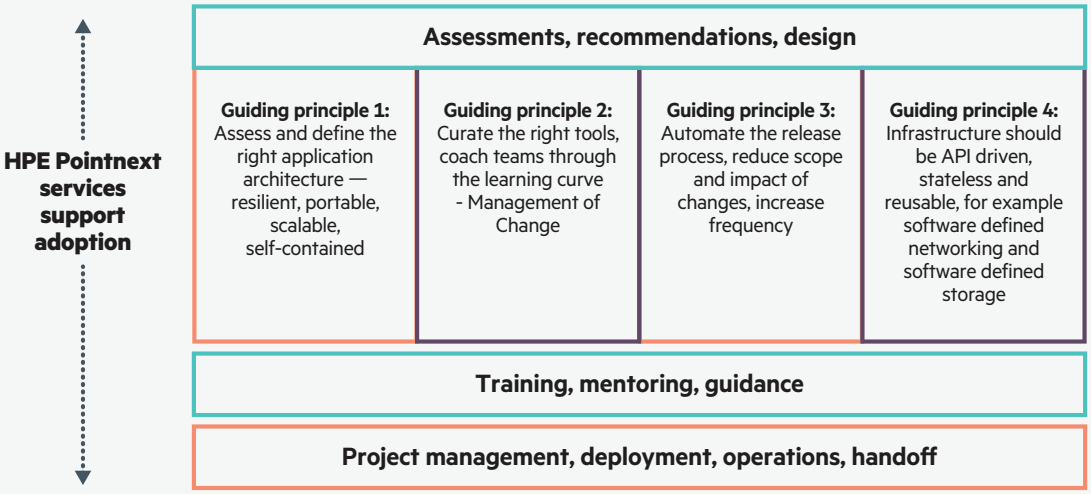
- Provide capabilities to orchestrate and provision service more quickly; enable workload portability and virtual provisioning of services in minutes (< 1 day for physical)
- Integrate a complete PaaS across multiple IaaS platforms, accelerating time-to-production. The goal is to deliver applications and new services at the speed of the market
- Accelerate deployment of applications and new services (cloud-native) and deploy and run applications in the new environment 50% faster

Adopt flexible infrastructure designed for cloud-native deployments

- Control, optimize and manage services across multivendor environments
- Optimize cost building and managing private cloud services
- Provide capabilities to design, build, operate private IaaS, PaaS and CaaS

Recognizing the capability of cloud-native applications and microservices architecture, HPE Pointnext has consolidated our experiences working with enterprises globally into a comprehensive methodology. The diagram shows how HPE Pointnext services help businesses set priorities and move ahead quickly.

Applying the cloud-native blueprint

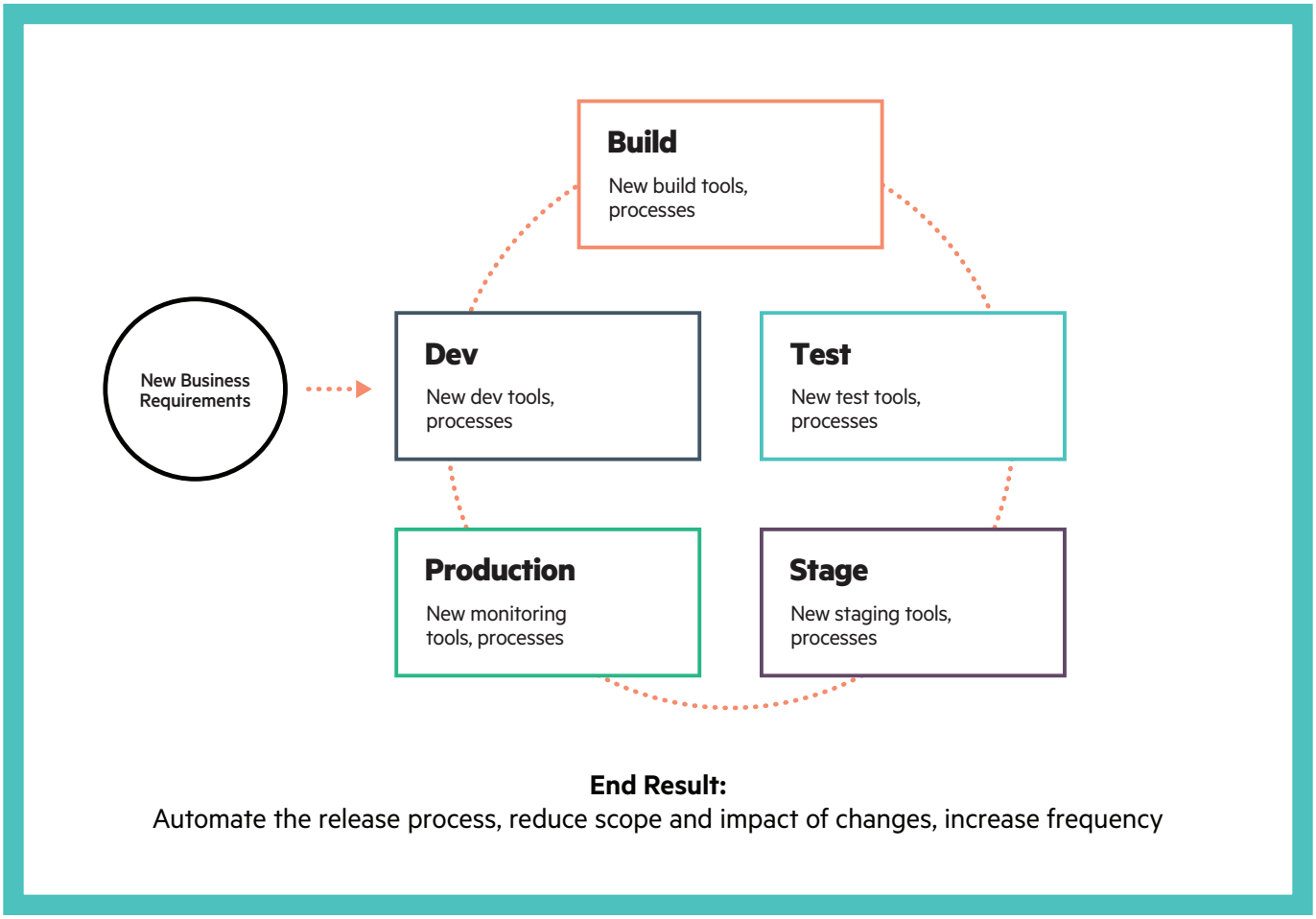


HPE Pointnext

HPE Pointnext, the services organization of Hewlett Packard Enterprise, brings you the expertise needed to help you make Hybrid IT simple and deliver on the promise of DevOps, open source and cloud technologies.

Combined with HPE heritage, strength in infrastructure and partner ecosystems, HPE Pointnext professionals can help you quantify the cost savings and the revenue streams from cloud-native and open source technology adoption.

HPE Pointnext recommends an initial discovery session, whether your focus is on planning, build, release process, or operations. The following services can assist with your cloud-native journey: HPE Cloud Native Agile Development Service to deploy a pilot application in a cloud-native environment, HPE Cloud Native Container Service to build your cloud-native container or HPE Cloud Native Operation Service for software upgrades, patching, capacity planning or ongoing automation, custom code support.



Conclusion

Enterprises, service providers and governments are looking to cloud-native technologies to cut costs, increase agility, and enhance productivity, while enabling new revenue streams and providing competitive advantages.

This is challenging. Cloud-native changes the way you deploy and manage applications and software platforms. Application development efforts must leverage existing assets where possible, provide cloud portability and accelerate the speed of delivery. Toolsets must balance the benefits of open source while providing the quality the enterprise requires.

The importance of building collaborative operations and development teams in order to deliver accelerated application deployment and delivery should not be underestimated. This means a focus on the applications, platforms and container management, orchestration, and how to scale the new and existing applications to the cloud.

Additional resources

HPE Pointnext
Cloud services
Cloud-native container services
Cloud-native agile development service

Hewlett Packard
Enterprise

September 2017



SHARE THIS WITH YOUR NETWORK