Microsoft Solutions Partner
Microsoft Cloud

Microsoft Solutions Partner
Infrastructure
Azure

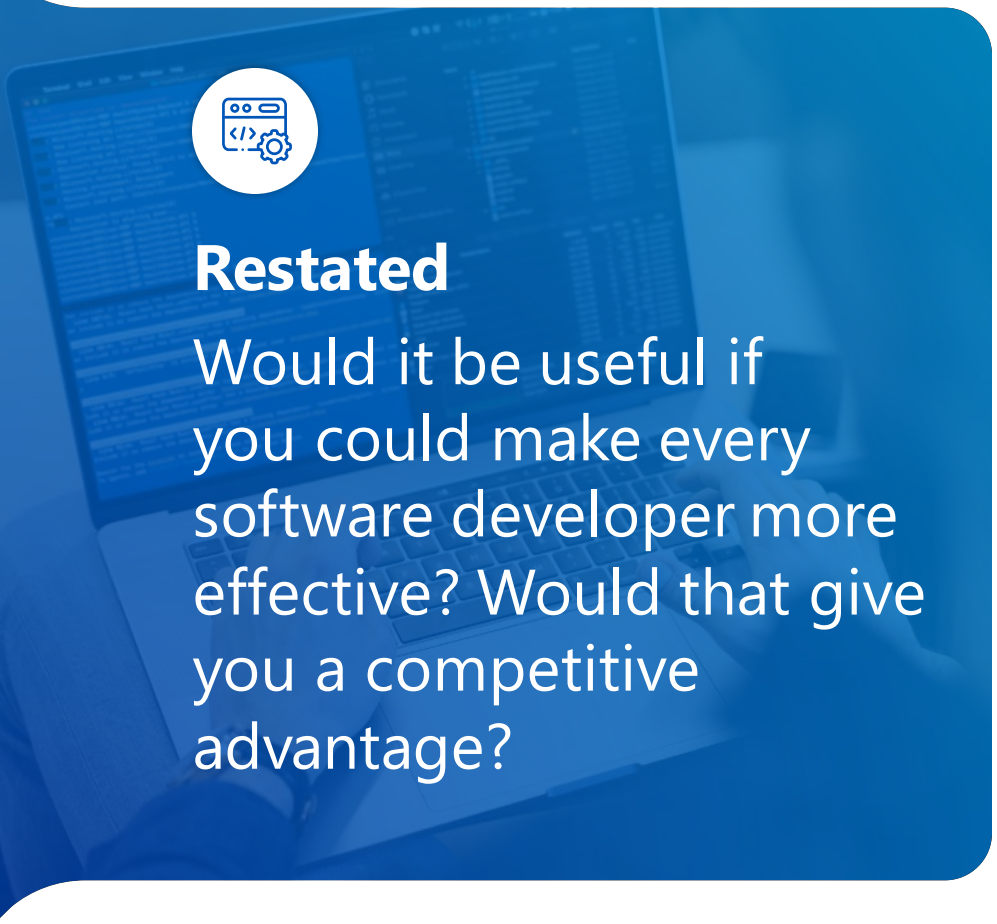
Specialist
Infra and Database Migration
Azure Virtual Desktop

DEVELOPER EXPERIENCE COMPASS

Upgrade Your Software Development Engine

REPLY VALOREM

HOW CAN YOU UPGRADE YOUR SOFTWARE DEVELOPMENT ENGINE?



Restated

Would it be useful if you could make every software developer more effective? Would that give you a competitive advantage?

Accelerating Developer Productivity

- Dev Box
- Deployment Environments
- Internal Developer Platform
- Azure Load Testing
- Test Data Management/Creation
- GitHub Copilot
- DevOps

DEVELOPER EXPERIENCE JOURNEY

The Developer Management Maturity Continuum

Retiring

- No Metrics on Deployment Frequency, Change Lead Time, Change Failure Rate, and MTTR
- Long Approval Processes for Testing Environments
- Little to No Test Data Management
- No Feature Flags
- No Ring Releases
- Little Visibility Into Pull Requests Deployed
- Time to Restore: 1- 6 mos.
- Change Failure Rate: 40-60%
- Lead Time: 1-6 mos.
- Deployment Freq: 1-6 x/mo.

Starting

- Some Images in Dev Box
- Some Load Testing at Scale
- DevOps Introduced but Not Followed, Siloed IT/Ops/Development
- Testing Separated from Development
- Time to Restore: 1 Day – 1 Week
- Change Failure Rate: 31-45%
- Lead Time: 1 Week – 1 Month
- Deployment Freq: 1x/wk. – 1x/mo.

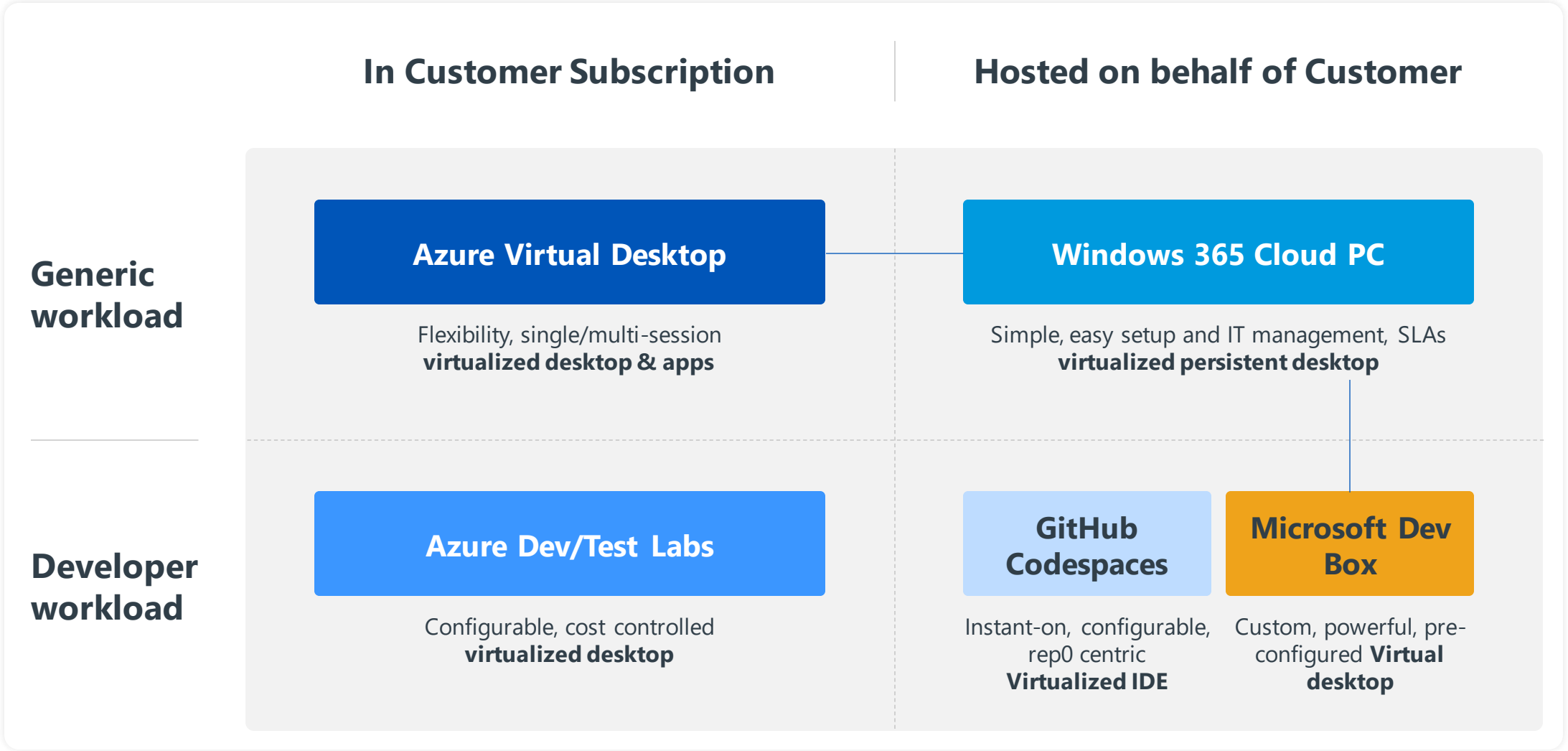
Slowing

- All Environments in Dev Box
- Test Data Management
- DevOps Used by Most Teams
- Some Targeted Load Testing at Scale
- Some Separation of Release and Deploy
- Time to Restore: < 1 day
- Change Failure Rate: 0-15%
- Lead Time: 1 wk. – 1 mo.
- Deployment Freq: 1x/wk. – 1x/mo.

Flowing

- Elastic Developer and Test Environments
- Developer Morale Boosts Productivity
- Every Pull Request Deployed to Production
- Automated Ring Testing and Release Gates
- Time to Restore: < 1 hr.
- Change Failure Rate: 0-15%
- Lead Time: < 1 day
- Deployment Freq: On Demand

TODAY'S CLOUD BOX SEGMENTATION



WHO IN YOUR ORGANIZATION DOES THIS IMPACT?

Developers – Self-Provision and use boxes that are pre-loaded and pre-configured on-demand

Team Management – Assign boxes and create effective team, assign the right tools and resources

IT – Manage with Microsoft endpoint manager, keeping everything secure and up-to-date

Leadership – Reporting enhanced, lead times reduced, product features are delivered quicker, promote remote team membership and remove supply-chain problems

WHAT PROBLEMS TO AGILITY ARE YOU FACING USING A TRADITIONAL INFRASTRUCTURE ORGANIZATION?

Complex Setup

Onboarding new developers is error-prone and time-consuming, and if configured incorrectly, issues take weeks to emerge.

Inefficient Environments

Mixing configurations from a variety of projects slows down workstations and can result in issues that are hard to diagnose.

Lack of Compute Scalability

Developers working on single, physical machines are limited to that device's compute resources and can't easily multitask between projects.

Security of Remote Devices

Remote developer teams working on a variety of devices make it harder for IT technicians to control access to sensitive source code.

Increased Risk of Exposure

IT technicians can't manage the uptime of dev workstations running remotely, putting them at greater risk of exposure.

Varying Industry Standards

Compliance standards differ across industries, requiring a variety of approaches to the deployment and management of dev workstations.

HOW DO YOU START WITH DEV BOX?

- Create a compute image – Define the image that the developer will work with
- Send a message to a team member
- Let the team member Self-provision Developer Boxes – Select the resources for the type of development, use the right image for project, and scale testing infrastructure up and down as needed
- Schedule start and stop times for the resources
- Start development

SCENARIO

New user added to the team to help address a production issue in a SAAS ecommerce solutions company

WOULD YOUR CUSTOMERS FIND ANY OF THESE ITEMS IMPORTANT?

- Do you want an increase in developer productivity?
- Do you want to speed up your lead time? Do you want to get to market faster with the ideas you have?
- Would you be better off if you released fewer bugs into production?
- Do you have long wait times to shift resources to projects?
- Could you benefit from quicker offshoring or nearshoring your next project?

WHY ARE WE DISCUSSING DEVELOPER EXPERIENCE?

What is "developer experience"?

- It's everything you need to manage, procure, and administer that allows a developer to code, test, and deploy features and applications to the customer.
- This includes an IDE, cloud infrastructure, test data, and code repositories.
- It's an explicit recognition that when an enterprise relies on software development, we need to intentionally design and create internal developer platforms to promote that experience.

What are the key elements that drive this?

- Developer Experience – Enhance the developer experience with self-provisioned environments.
- Hands-on Hardware – Eliminate the need for tracking, shipping, and maintaining a hands-on-hardware organization – Focus your team on differentiable value.
- Environment configuration – Adapt, configure, and provision at scale.
- Right-size specifications – The suitable environments and hardware for your needs – Not more or less than you need.
- Security – Manage and operate from the cloud.

TLDR

To enable 360-degree cloud operations and allow development to be fully agile using DevOps, centrally manage developer experience, and dynamically offer resources and environments to reduce MTTR, enhance time-to-market, and improve quality.

WHY? AND WHY NOW?

- Dev Box is a crucial way to start your organization toward cloud-native operations and rapid maturity
- Enhances already mature operations in adaptability
- Priced on operational usage
- Centrally upgraded and manageable
- The market rewards adaptability – Managing through Azure is gives you more maneuverability
- Manage budgets with cleaner cuts and more accurate traceability in assets
- Quickly shift to lower-cost resources – Move your development, increase the number of developers, and remove developers
- Innovation is Driven Through Experiment – Dev Box creates more opportunity for quick trials and iteration

ENABLE SELF-SERVICE DEVELOPMENT

Provide developers with self-service access to high-performance, cloud-based workstations preconfigured and ready-to-code for specific projects.

READY TO CODE

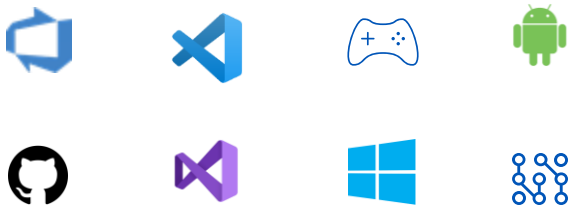
Maximize dev productivity with ready-to-code, self-service workstations preconfigured for specific projects and tasks.

FLEXIBLE BUT CONTROLLED

Customize workstations with everything developers need for their current projects while adhering to organizational standards.

MANAGED AND SECURE

Centralize governance of workstations running anywhere to maintain greater security, compliance, and cost efficiency.



Provision Any Workload

Build any app using any dev tool and repo.

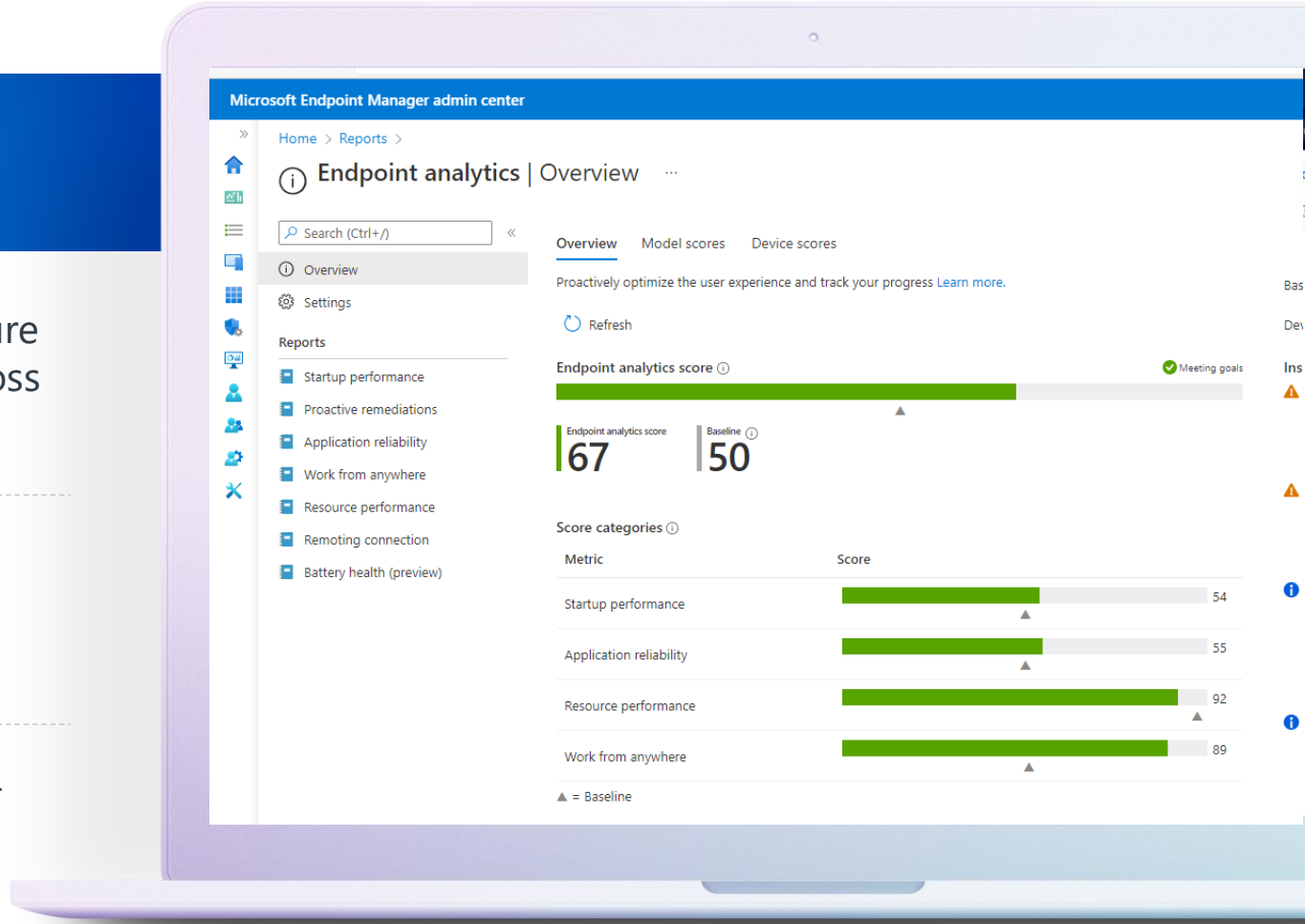
CENTRALLY MANAGED AND SECURED

Maintain greater control over security, compliance, and cost.

Integrate with existing enterprise infrastructure management systems to extend policies across remote devices.

Control cloud resource configurations on a project-by-project basis by enforcing Azure policies for each dev box.

Keep dev boxes up to date by applying zero-day patches to devices running across the organization.



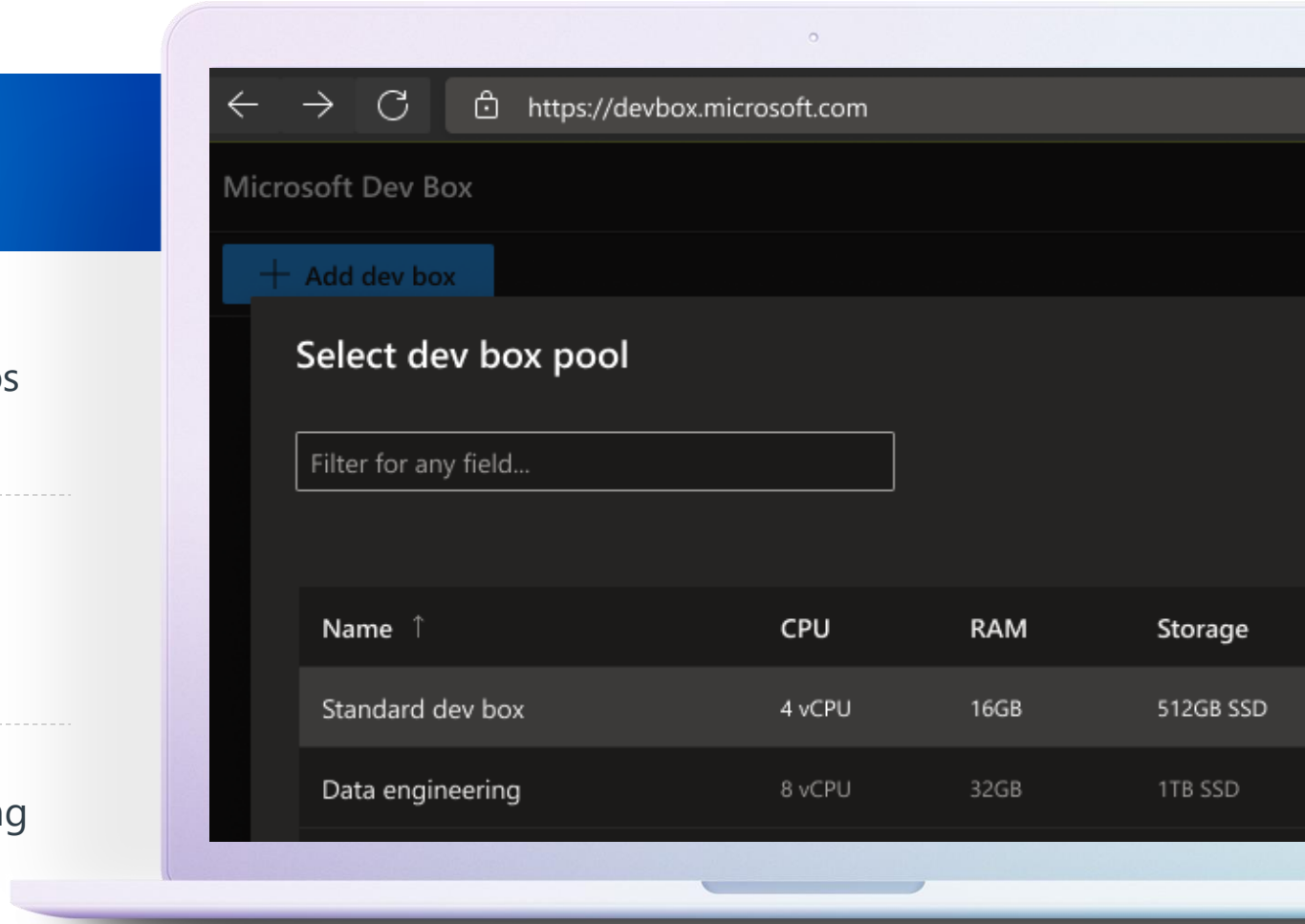
ON-DEMAND & SELF-SERVICE DEVELOPMENT ENVIRONMENTS

Quickly set up developers for virtually any development workload.

Configure project-specific, task-based dev boxes with the right dev tools and code repos to support any workload.

Create role-based dev boxes to support everyone on the team – from site reliability engineers to contractors.

Keep costs under control with stop scheduling and a consumption-based pricing model.



ENHANCE THE METRICS THAT MATTER

Lead Time for Changes

Keep all images that a developer could need and available on-demand – Reduce the time to make a developer productive.

- Low: 1-6 mos.
- Medium: 1 wk. – 1 mo.
- High: 1 day – 1 wk.

Deployment Frequency

Reduce configuration time and employ more robust testing with the right environments to move through your sprint more quickly.

- Low: 1 mo. – 2 yr.
- Medium: 1 wk. – 1 mo.
- High: On Demand

Mean Time To Recover

Rapid testing at scale can anticipate and resolve problems more quickly.

- Low: 1 wk. – 1 mo.
- Medium: 1 day – 1 wk.
- High: < 1 day

Change Failure Rate

Testing in the environments that you will deploy to reduces the need to rollback.

- Low: 46-60%
- Medium: 16-30%
- High: 0-15%

OUR APPROACH FOR *DEVELOPER EXPERIENCE*

Seamless and Accelerated Developer Experience Through DevOps

Change management, training, enablement and adoption are key to cementing organization change in developer experience.

Assess Developer Environments & Experience

Design> Implement> Refine

Developer Experience Workshops

Prioritized Roadmaps

Initial Deployment
Targeted releases within organization.

Expansion
- Expansion along easiest/most beneficial opportunities.
- Expansion of existing capabilities.

Optimization
- Expanded Usage
- Sticky Features

Training, Enablement, Adoption, and Change Management

How can I use AI to improve developer efficiency and effectiveness?

GitHub Copilot **MASTERCLASS**

1 Day – 1 Week

Curated learning journey that offers hands-on workshops and deep technical insights for a comprehensive understanding of AI and GitHub Copilot.

- OpenAI and GitHub Copilot Basics
- Advanced Techniques for Efficiency & Productivity
- Getting Started with GitHub Copilot

Ask about our GitHub Copilot Masterclass

My dev team lives in silos, releases infrequently, and struggles with release quality. How can we modernize our DevOps environment?

DEVELOPER EXPERIENCE **WORKSHOP**

1 – 2 Weeks

Map current setup, discover the current procurement process, and analyze the current developer environment's needs. Includes thought leadership, strategies, and mentoring on the entire Developer Experience including:

- Internal Dev Platform
- Load Testing
- Test Data Management
- DevOps
- Developer Environment Provisioning

Ask about our Developer Experience Workshop.

We are ready to invest in our developer experience but want to modernize and execute changes more quickly.

DEVELOPER EXPERIENCE **IMPLEMENTATION**

8 – 12 Weeks

Deploy and configure the right developer experience platform for your team and appropriate Azure services such as:

- Microsoft Dev Box Implementation
- AI-generated Synthetic Test Data
- DORA (DevOps Research Assessment) Metrics
- Feature Flags
- Ring Releases

We invested in DevOps but developer morale is still low and turnover is high. How can we improve our developer experience and increase work in production?

DEVELOPER EXPERIENCE **OPTIMIZATION**

8 – 12 Weeks

Optimize existing DevOps and developer platforms using Azure technologies to target roadblocks or concerns in your current developer experience.

- Advance DORA (DevOps Research Assessment) Metrics
- Prepare for and Adapt to Economic Downturns
- Improve Developer Adoption of Modern DevOps Tools and Process
- Anticipate Future DevOps Maturity & Disruption

Valorem Reply's **DEVELOPER EXPERIENCE COMPASS**

Accelerator Solutions

SYNTHETIC DATA GENERATOR
AI data generation tool for test production systems that automatically creates large test datasets that accurately mimic the characteristics, patterns, and statistical distributions of your production data.

COPILOT AI KICKSTARTER
Accelerate Copilot integration with shoulder-to-shoulder expert guidance getting your team to integrate Copilot into an existing application or proof-of-concept.

AI KUBERNETES SIDECAR
Scale AI modernization on a pod-by-pod basis with an add-on approach to your existing AKS application(s).

DEVELOPER EXPERIENCE IMPACT

Branching Strategy

Payment Services

Branching strategy crafted to fit the diverse experience of a newly-formed team to deliver and track feature releases consistently.

Production Lead Time

Professional Services, Engineering

Reduced lead time.

Integrated Infrastructure as Code in existing pipelines.

Helped accelerate and work with moving to DevOps.

Created test-data-management workflow.

Ring Releases

Financial Services

Worked with architecture and leadership to introduce ring environment and ring releases as a production environment.

Started organization on ring maturity journey of testing in production, releasing faster, and automated feedback from production.

Cloud Services

Technology Services

Built and maintain customer-focused migration platform with fully automated customer experiences.

Reduced human and manual efforts that cut cost per unit by 65%, eliminated data leakage and reduced migration team resource demands from hundreds to tens.

Microsoft Solutions Partner
Microsoft Cloud

LEARN MORE ABOUT VALOREM REPLY'S DEVELOPER EXPERIENCE COMPASS

Learn more: https://go.valorem.com/Developer-Experience-Compass