datalink

An Insight company

White Paper

Migration best practices: The benefit of making a clean break from legacy platforms

Introduction

Migrating from older to newer platforms is critical for IT organizations to ensure their enterprise continues to meet the availability, scalability, security, and efficiency standards set by the C-suite.

With the additional costs of supporting new infrastructure, any remaining workloads or data on old infrastructure effectively doubles the cost of infrastructure operations and eliminates the overall benefit of the new platform. Leaving workloads behind on old infrastructure after a transition completely defeats the ROI goals for a migration.

This paper examines the issues that lingering infrastructure creates for IT organizations and the various reasons these legacy platforms remain even after they have been "replaced." It also introduces a methodology for addressing the issues.

The problem with migrations today

Historically, migrations were about simply replicating a platform's technology stack on a new server and then copying content over from the old machine. Granted, it wasn't always that simple, but there was more involvement from the development staff and the business side, too, creating a smooth, focused effort.

Many things have changed. Hardware and software lifecycles have shortened, with systems coming and going much more quickly. The scope of IT has grown to involve every aspect of business. More efficiencies and higher expectations are squeezed from administrators and development teams as IT evolves.

Unfortunately, all these factors collide when new infrastructure and/or platform standards are emplaced. IT organizations struggle to complete migrations due to these issues, and often there are stragglers. After multiple migrations over the years, many organizations develop a trail of legacy platforms, servers, and applications.

This collection of old software and hardware poses a major threat to IT organizations on several levels:

No ROI for new platforms
Every day an old system remains in place, the cost savings and efficiencies that pay for the new systems go unrealized.

Resource cost
Old platforms typically require more power, maintenance, and administrative attention to maintain uptimes.

Licensing and support cost
Most old platforms must be supported by vendors' extended licensing programs. Platforms can be extremely expensive to support as they approach "end of life."

Availability risk
As IT employees come and go, so does the expertise required to keep a given platform/workload operational.

Security risk
Without a vendor keeping track of and patching vulnerabilities, and with hackers eager to exploit openings, old platforms are risky for any enterprise concerned with security.

Standards drift
Enforcing a standardized set of technologies becomes nearly impossible, as the trail of legacy platforms contains a myriad of old technologies that are not compliant with the standards.

Credibility risk
Old, mostly stable platforms can "disappear" in the buzz of everyday IT operations. Often the realization that these systems exist can be a surprise for leaders attempting to secure and standardize the enterprise.

Identifying the root cause

Currently, most migrations are done on a server-by-server basis, and either the servers are simply cloned or they are grouped by type and a migration process (or runbook) is created for each type. Typically, a combination of these methods can be successful.

However, upon close inspection, many infrastructure teams encounter issues with this approach, resulting in servers or workloads getting left behind.

Multiple target destinations

Many IT infrastructure teams have a variety of target platforms to choose from when migrating a workload. A blanket approach to a migration is usually dedicated to a single target, thus workloads are not always assessed to see if they can migrate to the best environment according to infrastructure priorities.

Developer availability/priority misalignment

Any dependency on development teams tends to delay the migration project as a whole as the IT infrastructure team waits for developers to send information or to help out.

Oftentimes, development teams want to help with migrations, but they are held accountable to the business for other priorities. Because of these conflicting priorities, migrations are prone to major delays.

Missing owners/inherited ownership

IT infrastructure teams tend to “inherit” server/application ownership. And, as an “owner of last resort,” they aren’t usually provided with the knowledge or documentation required to manage the workload. This creates a situation where the infrastructure team has no one to turn to for analysis or verification, culminating with gaps in information and delays in the migration project, or worse, a disruption in business due to failed migrations.

Missing expertise and resources

The concept of platform standardization attempts to prevent skill gaps resulting from employee attrition. However, as standards evolve and change, gaps can be created. These gaps hinder the forward progress of migration projects. This results in workloads getting left behind because the effort and cost to move them is prohibitive.

Leftover platforms are usually non-standard

In addition, non-standard or evolved standards can adversely impact migrations on an active platform. For example, in some instances a “template,” such as for web servers, changes over the years. The subtle differences can explode project timelines and result in delays, disruptions, and possibly additional leftover workloads on the platform.

No time for smoke or acceptance testing

Typically, migrating a workload requires extensive regression testing to ensure all functionality is intact prior to cutover. However, workloads with missing owners are difficult to test. Also, a typical bulk migration project doesn’t account for additional time requirements to create a test plan and execute it post-migration. As a result, verification gaps in bulk migration projects can lead to disruptions.

Business disruptions lead to fear of migrations

Skill gaps, template differences, missing business ownership, and verification gaps can create disruptions to the business side of an enterprise. If this occurs more than a few times during a project or just a few times across several migration projects, the business will become resistant to further moves. In fact, business and development teams may prefer the stability of the current platform over any benefits achieved by migrating to a new platform.

Solution methodologies

Methodologies developed from experience with complex migrations ensure that the issues that prevent full conversion to new platforms are addressed. This includes:

- ✓ Supplement traditional methods that can handle the bulk of a large migration project. This process can be used to migrate workloads identified as too complex or out of scope for the reference architecture.
- ✓ "Mop up" operation following a large migration to completely remove workloads so the legacy platform can be retired.
- ✓ Process to effectively re-platform workloads. Moving workloads from one platform to another either to save licensing costs, upgrade over several versions of the same platform, or migrate to a new standard platform.
- ✓ Assessment and migration tool to determine the best target platform for a workload, either moving off of a legacy platform or an unsuitable platform given the workload requirements.

Goals of the process

The right methodologies can directly address the issues and concerns presented above. Because these workloads are either complex or old, the goal is to ensure every detail is addressed for a safe and smooth cutover on the best platform.

Focus on the workloads required by an application

The factory approach tends to focus on individual servers. Many applications have functionality that traverses multiple servers, which can lead to complexities, complications, and disruptions for traditional migration methodologies.
By focusing on application re-deployment to the new target, the new methodology actually simplifies the process.

Create a direct relationship with business and development teams

Otherwise, identify business users and establish communication with them instead. Disruptions and delays caused by bulk communications are typically due to a lack of engagement between the infrastructure team and the owners of the applications and/or data being moved – one of the main weaknesses of the traditional migration model.

Proactively research information about the application and its workload requirements

Going directly to the application and learning about its setup, configuration, and requirements can be the best way to get a good grasp of the real structure of an application. This type of detailed analysis is often what's missing in the current methods. Minor implementation differences that are not caught early in the bulk approach can lead to issues and delays.

Assess the workload to determine the best target platform

Traditional migration methods are dedicated to moving workloads to a single target environment that may not be the best – or even a good – destination for the workload. Cost and ROI efficiencies can be gained by considering the enterprise's priorities for each workload. So, assessing these opportunities for each application is a good idea.

Work with development and business teams to create a migration schedulethat coordinates with their release/business cycles

Rather than disrupt development teams or the business in general by overlapping a migration with development or business priorities, schedule changes in conjunction with these teams. Coordination is imperative.

Create effective verification plans to smoke-test migrations prior to cutover

Verification is key to preventing disruptions. Traditional methods don't anyways go deep enough to verify functionality before cutting over. A deeper verification is required to both prevent disruptions and identify specific issues for triage and quicker resolution.

Advanced application migration methodology

This methodology focuses on migrating individual applications off of legacy platforms by replicating the deployment process. This involves understanding the infrastructure goals and priorities for target platforms as well as developing an understanding of each application's deployment process.

Build a proactive relationship with the application team

For each application in scope for the project, begin with proactively investigating the current knowledge and user base of the application. Waiting for answers to questionnaires and emails can lead to apathy and delays in projects. Develop proactive relationships to engage the consumers and owners involved with these applications. This helps the flow of information and establishes the communication necessary to successfully migrate the application to the best workloads possible.

Determine the right workload fit

Perform a detailed assessment of each application to ensure that its related workloads are assigned to the best target environments. A cost-benefit analysis should be conducted in regard to coding gaps for higher priority platforms. A code change to move off of an expensive platform to a less expensive platform could yield a desirable ROI.

Create a runbook for each application

Essentially, an application migration is a form of re-deployment based on a new technology platform. The analysis and creation of the runbook should focus on how to deploy the application to the new set of target workloads.

Create a verification plan for post-migration smoke testing

During the analysis phase, work with application owners to create a solid acceptance test to apply to the application after the migration is complete. This will help prevent business disruptions and build credibility with users and owners. These tests can be reused to verify code changes, too.

Schedule in conjunction with the application team

Getting buy-in and coordinating migration scheduling with application development teams and users is very important for preventing business and delivery disruptions. For development teams, the scheduling is important because they might be in the midst of a release.

Execution = Deployment

Whether the migrations are done as a group/wave strategy or on a one-by-one basis, the execution of an application migration should be considered a deployment to a new set of workloads. Therefore, there is the potential that more than one server could be retired by the migration.

Add QA and development resources to the migration team

Having credible resources on the infrastructure team who can interact directly with the application teams, modify, and regression test the application will eliminate delays associated with waiting for these resources to respond or become available. Developers and QA professionals can also act as liaisons with application teams since they share a similar perspective.

Conclusion

Legacy platforms are a major drag on the overall performance and cost of IT infrastructure. Given that legacy platforms must be retired in order to effectively generate the ROI expected from upgrades and new hardware purchases, a complete migration is imperative. Following a defined and tested methodology cuts through the complexities and relieves the enterprise of stubborn legacy platforms.

A trusted platform migration resource

Datalink helps organizations transform technology, operations, and service delivery to meet business challenges. Experts in platform migration, we work with clients to deliver comprehensive transformations that enhance service levels, support growth, increase operational efficiency, and reduce risk. Visit datalink.com to learn about our full portfolio of services and solutions.

datalink

An Insight company

datalink.com | insight.com