Agility Platform
Implementation Best Practices

An enterprise blueprint to transform IT using a new cloud-based IT operating model
Executive Summary

Enterprises are increasingly aware of the need for greater IT agility while controlling costs. Global market pressures and new competitive entrants continue to drive the imperative for change. ServiceMesh customers are implementing the Agility Platform to address these needs and provide greater IT agility, improve the pace and frequency of software releases, and improve cost visibility and control. This document will provide a best practices approach to achieve these results in any large enterprise.

Realizing these strategic benefits is dependent upon positioning the Agility Platform to effectively implement a new cloud-based IT operating model. Without focus, an IT organization can get delayed and sidetracked on cloud infrastructure automation projects without raising the bar to transform the software development lifecycle, drive IT resource standardization, broker hybrid cloud services, or enable continuous compliance and governance. This document highlights best practices to stay on track and ensure the strategic benefits of full IT transformation. The following topics are covered:

- **Agility Platform reference implementation architecture** – Successful implementations start with an understanding of the scope of the Agility Platform’s role in enabling that transformation, including integration with underlying cloud implementations and the existing IT ecosystem.
- **Enterprise adoption maturity model** – This reference adoption model define a best practice evolutionary path to achieve the end state goal of IT as a Service.
- **Phased implementation best practices** – A high level guide, with reference to supporting ServiceMesh materials, that describes how enterprises can rapidly progress toward their enterprise cloud objectives and avoid common pitfalls.

This document is intended to provide a strategic overview, and detailed field implementation documentation and supporting material from ServiceMesh is referenced throughout this paper. This includes Agility Platform whitepapers, product user and admin guides, ServiceMesh Agility Platform product training programs, ServiceMesh implementation training programs, ServiceMesh field implementation guides, and more. Additional resources are listed in the References section of this document and can be obtained through your ServiceMesh account executive.

Target Audience

This document is intended for business and IT leaders responsible for enterprise cloud strategy, ServiceMesh Agility Platform implementation teams, and IT architects involved in cloud management platform initiatives and projects.
The Enterprise IT Transformation Journey

Businesses today are under mounting pressures to be more responsive to customers, produce new and better products on shorter timelines, fend off aggressive global competitors, and all while maintaining tight cost controls. To meet these challenges, businesses are demanding that internal IT organizations make significant improvements to their agility and responsiveness to support their rapidly changing business needs. This requires an IT transformation. IT organizations that fail to adapt will face severe consequences as business units pursue external IT resources to address their needs, including a rapidly growing number of public cloud services. Of course, these external IT resources introduce their own set of risks, such as when ungoverned and unsecured corporate data and applications potentially expose the enterprise to costly regulatory penalties and/or loss of reputation. IT organizations that excel in the future will shift from manual, workflow, and infrastructure-based operating models within their data centers, to an on-demand, self-service IT delivery model spanning private and public clouds, also known as a “cloud-based IT operating model”.

Core Tenets of Cloud-based IT Operating Models

Cloud operating models enable IT organizations to be more flexible and responsive to business needs by providing self-service access to standardized IT service portfolios, improving the pace and frequency of software releases, providing real-time cost visibility and chargeback, and enabling cloud workload portability and vendor contestability. Moving to this model requires a strategic transformation that reshapes the IT organization and their role within the enterprise, ultimately positioning them as the cloud service broker and IT-as-a-Service provider to the enterprise.

Core tenets of cloud-based IT operating models include:

- Self-service, on-demand provisioning of cloud-based IT service portfolios, including the standardization of infrastructure and platforms to be used across the enterprise.
- Integration of on-demand IT resources directly into existing software development lifecycle toolchains and processes to support application release automation and DevOps initiatives.

What is “IT as a Service”?

IT as a Service is the delivery of IT infrastructure, platforms, and applications as on-demand services to business unit end users. IT as a Service requires IT to act as a service broker to combine internal and external services into the best and most competitive portfolio of IT services possible for business needs.
• Migration of existing application portfolios to the cloud
• Onboarding and governance of workloads deployed to public clouds without direct enterprise IT support (Shadow IT)
• Real-time service cost and capacity visibility including pay-as-you-go and chargeback models
• Cloud-native application architectures including multi-tier auto-scaling, and cloud-based high availability and disaster recovery capabilities.
• Implementation of hybrid cloud and vendor contestability models, including workload placement optimization to take advantage of cost, performance, and other parameters.

Each initiative along the IT transformation journey can deliver incremental value and ROI when best practices are followed.

Role of the Agility Platform

A prerequisite to discussion of best practices is an understanding of the scope of the Agility Platform and how it should be integrated within an existing enterprise IT ecosystem. Field implementation teams should be well versed in this Agility Platform foundational knowledge so that well-informed customer implementation decisions can be made.

Best practice use of the Agility Platform includes not simply managing cloud infrastructure, but also managing portfolios of cloud-based applications and platforms while meeting stringent requirements for governance, compliance, and security through flexible policy definition and enforcement. Used properly, the Agility Platform can greatly simplify the complexity of managing diverse workloads across diverse private, public, and hybrid clouds, and helps accelerate the adoption of these cloud services within an organization. It is important to understand the range of capabilities provided within the Agility Platform and their intended implementation within enterprise IT. The Agility Platform is often described in the context of its product modules (Planner, Designer, Store, Release Manager, Operations). Although useful from a feature and functionality level, these product descriptions are not as useful in describing the Agility Platform within the context of a larger enterprise IT implementation. For that reason, a review of an Agility Platform reference implementation architecture is provided.
Reference Implementation Architecture

The reference implementation architecture shows the logical components of the Agility Platform along with surrounding integration to other internal and external IT systems. The reference architecture includes three primary categories: the Agility Platform, the underlying Cloud Implementation layer, and the internal IT ecosystem. The surrounding IT systems and cloud services shown will be driven based on specific customer needs, and not all integrations are required. The following sections of this document will describe a maturity model that phases in the integration of third-party systems and clouds services in an incremental adoption model. An illustration of the overall reference implementation architecture is provided below:

Agility Platform

The Agility Platform is designed to provide a single control point for the orchestration, governance, and lifecycle management of an organization’s cloud-based offerings, including cloud-based applications and platforms deployed across hybrid clouds. A brief description of each logical component of the Agility Platform follows from the top down.

- **Consumer and Administration Interfaces:** Cloud-based IT operating models enable the consumption of self-service, on-demand IT services directly by end users. This can be facilitated
in a variety of ways, and is supported by a web-based self-service store for end users, a graphical user interface for Agility Platform administrators, a command line interface, a robust API, and a Software Developer Toolkit (SDK) to allow customers to create their own Agility Platform adaptors.

- **Identity and Access Management**: Agility Platform manages self-service access to resources not just based on identity/role; but also what organizational group they are apart of, what application project they belong to, and what lifecycle stage they are working in. Agility Platform interfaces with an organization’s identity products such as LDAP and Microsoft Active Directory and extend those systems with a fine-grained entitlements system and a policy engine capable of providing a high degree of control.

- **Orchestration and Automation**: Orchestration and automation extend across service design, lifecycle, and operations so that these components can support delivery and operation of cloud workloads. Agility Platform provisioning of complex workloads requires not only deployment of application components, but also orchestrating across a broad range of third party IT systems to support IP management, security, monitoring, chargeback, and also SDLC tool chains when promoting software builds and environments for application release automation. Agility Platform can address these orchestration needs internally, or in some cases drive other orchestration engines already embedded within the enterprise.

- **Policy Engine**: Giving end users on-demand, self-service access to IT resources means handing over much of the control that IT maintains today via manual processes. Agility Platform gives IT organizations the comprehensive policy controls they need to govern cloud-based applications and workloads. Agility Platform’s policy engine allows IT to empower business units without manual actions and/or approvals required by IT. This includes optimizing the deployment decisions for applications across private, public and hybrid clouds, enforcing governance and compliance throughout the lifecycle of cloud applications, and govern application release automation across the SDLC.

- **Security**: Agility Platform provides a consolidated control point to deploy and manage workloads across private, public, and hybrid clouds. As part of this role, the Agility Platform is also uniquely suited to enforce cloud security policies in a consistent and automated manner across the organization so that cloud workloads can operate as a trusted extension of the enterprise. Agility Platform facilitates this across a variety of security products with the policy-based configuration of physical and virtual firewalls.

**Additional Resources:**

**Enterprise Cloud Governance Requirements and Best Practices**
This whitepaper define enterprise requirements for cloud governance, along with a broad range of Agility Platform policy examples, use cases, and best practices.

**Cloud Security using the Agility Platform**
This whitepaper describes how the Agility Platform can enable corporate security policy enforcement, firewall policy enforcement, improve visibility and control, and provide timely vulnerability assessments.
encrypted persistent storage, encrypted and isolated network services, and more. Agility Platform also hardens workload security posture by embedding and orchestrating across a wide range of security tools including Host Intrusion Detection (HID) systems, anti-virus, Security Information and Event Management (SIEM) systems, encryption key stores, and others. In addition, the Agility Platform delivers logs and reports to track access, usage, and changes for auditing purposes, and creates thresholds to send real-time events/alerts to report security issues.

**Service Design:** Agility Platform enables IT organizations to plan, design, and publish service offerings into the Agility Platform Store, third party catalogs, or other systems for consumption by the business. These include not just virtual machine templates, but complete application environments, services, and packaged applications that can be ordered and consumed on demand. These multi-tier, orchestrated offerings are designed using Agility Platform’s application blueprints, which can include operating systems, middleware components, and other application services. These blueprints can be designed to be independent of the underlying infrastructure and cloud implementation, which is a critical best practice so that placement decisions are based on policies at provisioning time and not embedded into the design of the offering.

**Service Lifecycle:** Services and applications all exist within the context of a lifecycle that defines how they are planned, developed, tested, and released into production. Agility Platform was designed around a lifecycle management framework to ensure that services and applications get provisioned and configured correctly based on which stage of the lifecycle they are in. An example is leveraging the Agility Platform to accelerate the pace and frequency of software release and improve software development team productivity by integrating into existing development processes and tool-chains to help automate the application release process. Not only does this enable faster development cycle times but it helps alleviate many of the configuration defects encountered during Dev-to-Ops handoffs. Once applications have been provisioned and released, there are many reasons why an application could change: functionality enhancements, bug fixes, updates to the application platform such as application servers, databases, operating systems or associated packages and libraries. Applying upgrades or patches is an ongoing part of the application lifecycle and often the responsibility of multiple disparate organizations such as platform engineering, infrastructure and development. Agility Platform supports configuration management systems with policy-based governance to ensure properly orchestrated and automated application configuration updates to enable diverse IT organizations to revise their configurations or code and effectively roll out updates to the cloud. Without policy-based governance, orchestration of configuration updates involving different specialties becomes a risky manual process. Agility Platform is therefore critical to simplify the service lifecycle through application-level configuration management.

**Service Operations:** Agility Platform must meet the needs of multiple parties when workloads are operated in a hybrid cloud environment. For example, application teams need visibility into application performance and configuration, while IT Ops teams need to understand resource consumption across cloud environments. Both are concerned about security. Agility Platform collects and exposes this information via the Consumer and Administrative Interfaces layer. In
addition to providing performance, usage, configuration, and event management capabilities, Agility Platform also facilitates greater automation around the operation and support of applications. For example, Agility Platform can perform automated SLA management through active monitoring and auto-scaling. This becomes critical as cloud workloads grow from hundreds, to thousands, or tens of thousands of workloads, upon which automation is a necessity to meet business objectives.

- **Dynamic Resource Management**: Agility Platform leverages the underlying Cloud implementations to deploy services. Cloud implementations commonly expose programmatic access through an API or command line interface. The Agility Platform uses adaptors to bi-directionally communicate with supported CIs. This includes receiving information from the cloud implementation including current resources state, usage, price, capacity information, and more. This information is vital to Agility Platform’s threshold monitoring, autoscaling, and policy enforcement capabilities. A significant benefit of the cloud-based IT operating models is also vendor contestability. The Agility Platform integrates with many different cloud implementations and provides a SDK to rapidly extend support to new CIs as they emerge.

**Cloud Implementations**

A broad and growing range of Cloud Implementations (CI) are now available to enterprise customers. These CIs can be run and operated locally within an organization’s own datacenters, managed by third-party Managed Services Providers (MSPs), or provided by a long list of public Cloud Service Providers. The Agility Platform support over a dozen different private and public cloud implementations, and can enable cloud portable workloads between them. For cloud implementations that are not currently supported by ServiceMesh, an Agility Platform SDK has been created to rapidly create new cloud adaptors as needed.

**Private Cloud:**

- **Virtualized Resources**: The Agility Platform provides adaptors to popular virtualization resource managers such as VMware, Microsoft, OpenStack and others. The Agility Platform performs the governance, orchestration, and management of workloads, but drives the underlying deployment through the resource manager’s APIs. Enterprises start using the Agility Platform almost universally with their virtualized IT resources.

- **Physical Resources (Optional)**: There are many physical IT resources within an enterprise data center including servers, switches, storage, firewalls, load balancers, etc. The Agility Platform’s SDK can be leveraged to provision physical resources (bare metal provisioning) and govern usage, orchestration, and automation of physical resources with policy-based controls if required for certain services.

**Additional Resources:**

Agility Platform Adaptor List

This document provides the most current list of supported adaptors to third-party products and cloud providers.
Public Cloud:

- **Infrastructure-as-a-Service**: Public IaaS offerings can be based on a wide range of different technology stacks, and vary greatly in functionality from one to the next.

- **Platform-as-a-Service**: Public PaaS offerings are highly orchestrated application environments capable of abstracting and hiding the infrastructure from the developers and freeing developers to focus entirely on their application, sometimes at the risk of vendor lock-in.

Internal IT Ecosystem

Many existing IT investments can be leveraged to make the cloud an extension of the current IT organization. The reference architecture includes typical integration points that enterprises will encounter and need to evaluate over time as they adopt the Agility Platform, including the following:

- **Security Systems**: Ensure cloud workloads operate as a trusted extension of the enterprise by implementing the enterprise’s standard security tools such as firewalls, encryption key stores, Host Intrusion Detection (HID) systems, anti-virus, Security Information and Event Management (SIEM) systems, and others.

- **Identity Management Systems**: Extend existing identity and credentialing systems to cloud services with integration to systems such as LDAP and Microsoft Active Directory.

- **Network Services**: Automatically register cloud workloads with the enterprise’s DHCP and DNS systems and adhere to corporate standards for VLANs, subnets, hostnames, etc.

- **Configuration Management**: Automate registration of cloud workloads with CMDBs and other repositories to ensure consistent information on workload configuration. Integrate with other configuration management tools such as Puppet to ensure continuous compliance of cloud workload configurations as appropriate.

- **Monitoring/Reporting**: Agility Platform captures real-time IT resource consumption and performance monitoring information and provides visibility either through Agility Platform’s native reporting interface or exposed to third-party systems via Agility Platform’s API. In addition, agents for third-party monitoring tools can be embedded in workloads deployed by Agility Platform so that reporting information is sent directly to third-party dashboards and analytic tools, if desired.

- **SDLC Tool Chain**: Agility Platform integrates with tools across the software development lifecycle including IDEs, code repositories, and continuous build servers for application release automation and automated environment provisioning and governance to improve the pace and frequency of software releases.

- **Accounting/Chargeback**: Agility Platform allows IT organizations to track resource consumption and establish price models for IT services to provide cost transparency for different service offerings, real-time visibility into resource usage including show-back reports, and integration into accounting/cost tracking systems for chargeback of resource consumed.

- **Other ITSM**: The Agility Platform can integrate with other service management and service orchestration tools via adaptors, APIs, or SDK as required by specific enterprise use cases.
The transformation to a cloud-based IT operating model is an extended journey. Following the right adoption path can dramatically improve time-to-value and eliminate false starts and dead ends. It can also make the difference between providing significant incremental value and ROI, versus getting sidetracked on tactical initiatives that delay more strategic benefits and outcomes.

ServiceMesh has created a maturity model to better define a best practices implementation path to IT as a Service. The maturity model is an incremental adoption plan to build expertise, implement new product capabilities, revise processes, and manage organizational change. The model also provides a view of future-state goals with high-level initiatives and deliverables specified along the way.

Four stage model
The maturity model has been organized into four stages with organizational maturity increasing with each stage from left to right. There can be multiple entry points into this maturity model. In other words, maturity doesn’t have to evolve rigidly in waterfall style from beginning to end, but conversely you won’t be able to successfully jump from an early milestone to a much later one without building out core foundational capabilities along the way.

1. **Virtualize**: Consolidate data center workloads on virtualized IT infrastructure
2. **Private Cloud**: Improve agility by offering self-service IT resources, and improve efficiency by automating and standardizing around IT services
3. **Hybrid Cloud**: Accelerate time to market for new projects, extend governance models across all cloud usage, and begin implementing vendor contestability models
4. **IT as a Service**: Establish enterprise IT as the broker and portfolio manager of hybrid cloud services across the enterprise

The maturity model is represented below and provides a high level depiction of the transformation required. IT transformation requires changes in process, organization, and technology. In the following sections, we will delve into the prerequisites, best practices, and implementation guidance to successfully progress from one stage to the next in alignment with this maturity model:
Implementation Best Practices

The best practices that follow will be outlined in the context of the maturity model’s phased implementation approach. The intent of this section is to provide an essential overview, however detailed implementation documentation and guidance is beyond the scope of this paper and should be conducted in collaboration with ServiceMesh professional services staff and/or ServiceMesh partners.

Where appropriate, reference to other ServiceMesh documentation and resources are provided along the way. This includes whitepapers, product user and admin guides, product training programs, field implementation materials, and more. Additional resources may be available, and access to these and other resources can be obtained through your ServiceMesh account executive.

Additional Resources:
Agility Platform Training
ServiceMesh’s provides a range of training programs including computer-based and classroom-based courses.
“Virtualize” Phase

This phase is a starting point and foundation for future phases. The capabilities below are essentially prerequisites to implementing the Agility Platform, and any additional virtualization management capabilities beyond these will typically help facilitate the following phases:

- **Virtualization resource managers**: This can include VMware, Microsoft, or other vendor solutions for managing virtualized compute, storage, and network resources within the customer’s data center.

- **VM provisioning processes**: This can include fully manual or semi-automated business processes to request, approve, and provision a virtual machine by a user.

Typical benefits associated with this phase include improved IT infrastructure resource utilization and the ability to defer data center expansion by consolidating more workloads on existing resources.

“Private Cloud” Phase

The initiatives for this phase are described in a logical build sequence where the deliverables from the first project are aligned to the prerequisites for the following one. The scope of each initiative has been sized to deliver results and benefits within an approximate 3-6 month timeframe. Customer may of course elect to combine initiatives together or rearrange their sequence.

**Initiative 1: Self-service IaaS**

**Business and IT Objectives**

- Self-service IaaS provisioning to improve IT agility
- Improve IT infrastructure capacity utilization to reduce costs/defer additional investments
- Help mitigate ungoverned Shadow IT by introducing self-service infrastructure offerings
- Systematically reclaim unused/underutilized IT resources to reduce costs and sprawl
- Standardize on OS images and security/operating environments to mitigate non-compliance risks
• Standardize on OS images to reduce image sprawl and lower support costs

**Deliverables**

• Self-service store for end users
• Portfolio of standardized infrastructure offerings up to the OS level
• On-demand IaaS deployable to an internal private cloud
• Policy-based automation for RBAC, host naming standards, and other examples described in Implementation Best Practices below.

**Enabling Technologies and Prerequisites**

• VM resource manager used as a foundation for the internal private cloud
• Active Directory/LDAP for user credentials
• Network services for IP assignment
• CMDB integration for basic VM resource tracking
• IT resource monitoring (either existing third-party monitoring product or use Agility Platform native reporting capabilities)
• Encryption key store (either external key store or use Agility Platform internal key store)
• Security agents and utilities required for OSes and VMs
• Physical and virtual firewalls

**Process and Organizational Change**

• Maintain a self-service store to capture and automate IT resource requests
• Manage and administer an IaaS portfolio on an on-going basis
• Create and maintain RBAC policies for self-service IaaS consumption, host naming, and basic IaaS security/operating environment enforcement
• Create and maintain policies for lease expiration, instance limits, and VM sprawl reduction as applicable

**Implementation Best Practices**

• **Permission models:** Many organizations have not designed RBAC permission models to publish and manage self-service IT resources at scale across the enterprise. It is important to establish a permission model from the outset that accommodates future IT resources portfolio growth and management. ServiceMesh has created a best practice, customizable RBAC model to fulfill this need.

• **Establish host naming standards:** Organizations that do not have host naming conventions and standards need to implement this in order to create policies that assign hostnames during automated workload deployment. ServiceMesh can provide several example policies.

### Additional Resources:

**Best Practice RBAC models**

ServiceMesh has created a best practice RBAC model that can be customized as necessary.
• **Catalog design**: Organizations need to make infrastructure sizing decisions for their catalog offerings. Best practice approaches follow popular public cloud providers such as Amazon, with easy to understand compute resources labeled “Small”, “Medium”, “Large”, etc.

• **OS image standardization**: Most organizations have allowed OS images to sprawl over time to dozens or even hundreds. These OS variations increase support costs (patching, trouble tickets) and would result in a confusing and unmanageable self-service catalog. Best practice standardization efforts result in a reduction to 5 or less self-service OS offerings, and typically with a mix of Windows and Unix offerings to appeal to the broadest audience.

• **Security/operating environment enforcement**: All IaaS catalog offerings should be deployed with a default operating environment and security posture that meets IT standards. Typically, IT organizations have gold master VM images that contain their desired guest operating system and related configurations. These gold images should be imported directly into the Agility Platform when available. If not available, ServiceMesh can provide a selection of base OS images as an alternative. The Agility Platform enables additional software packages, configurations, and policy controls to be layered on top of these base images to create the desired security and operating posture for catalog offerings.

• **OS standardization enforcement**: Standardization enforcement initially focuses on new projects, rather than forced migration of existing or already deployed workloads. For new projects, the project teams are encouraged to use the standardized IaaS offerings primarily for the benefits of receiving reduced provisioning time down to minutes or hours, versus days or week. If these teams insist on a non-standard OS, they are usually told they can be accommodated but that provisioning of non-standard offerings will take “6 to 8 weeks” (enter your appropriate time frame) to account for exception-based, manual provisioning and deployment versus use of the automated deployment capabilities of the private cloud. This is the primary motivator for IaaS catalog adoption and other cloud services to follow.

• **Co-existence strategies for existing ITIL-based workflows**: For organizations where firmly established ITIL-based workflows are in place, implement the self-service model initially for non-production use cases (such as Dev and Test), leaving the ITIL workflows in place for production use cases until a later time.

• **Self-service store**: The Agility Platform comes with its own self-service store. In addition, the Agility Platform API can support third-party portals/catalogs for self-service provisioning, although this approach should be avoided if third-party catalog SKU management is too time consuming or restrictive, which can be the case for general purpose IT catalog offerings. Instead, use the Agility Platform’s Store module.

**Success Metrics**

- Improved IT infrastructure utilization to reduce costs
- Lower ratio of IT staff to VMs under management
- Mitigate VM sprawl by automatically reclaiming underutilized IaaS to reduce costs

**Additional Resources:**

Agility Platform ROI model

ServiceMesh’s ROI model can help build a business case for multiple initiatives and stages of maturity.
Standardized IaaS to reduce support costs for patching, updating, and trouble ticket resolution

Initiative 2: Initial self-service platforms for dev teams

**Business and IT Objectives**
- Measure significant improvement in dev team productivity and project cycle time compression
- Seek out dev team early adopters to use private cloud-based, self-service application development platforms
- Provide customized platforms that foster greater dev team productivity and innovation, while also demonstrating appropriate governance controls to prevent platform standard deviation and sprawl
- Eliminate ungoverned Shadow IT in selected teams by offering platforms customized to dev team needs deployed within the security of their own private cloud
- Obtain new dev team leader sponsors and champions

**Deliverables**
- Multi-VM, multi-tier platforms that are built upon initial IaaS assets
- Dev platforms customized for internal dev team needs
- Self-service store containing on-demand platforms deployable to an internal private cloud
- Policy-based controls for entitlements/provisioning of dev platforms for specific teams

**Enabling Technologies and Prerequisites**
- Expanded network services to support multi-tiered workload deployments across networks as appropriate
- Expanded storage options to support database offerings

**Process and Organizational Change**
- Expand the self-service store to capture and automate dev team platform requests, and manage/administer platform offerings
- Develop in-house expertise orchestrating multi-VM, multi-tier platforms using packages, scripts, and blueprints
- Create a role responsible for dev platform modeling, design, and support to address evolving dev team needs. This includes application components such as databases, app servers, and other application services.
- Create and maintain expanded RBAC policies to encompass platform designers and dev teams
- Create and maintain policies for platform security/operating environment enforcement
- Create and maintain policies for dev team leases and instance limits, as applicable

**Implementation Best Practices**
• **Dev team candidates:** Some development teams have direct control over their own IT resources, including the virtualized environments they use to deploy their work. Alternatively, central IT Ops organization may have already assigned a discreet set of resources to a dev team, and also assigned an IT manager to oversee these resources and take their direction primarily from a dev manager to meet dev needs. In both cases, the dev team operates their IT resources fairly autonomously from central IT operational oversight. These Dev teams are generally good candidates for initial self-service platform adoption. The screening criteria for these autonomous dev teams must also include several technical criteria including the specific VM resource managers, network services, and storage services being utilized to ensure they can be readily managed by the Agility Platform.

• **Initial self-service platform offerings:** To facilitate adoption, initial self-service platforms should target the specific needs of the selected dev teams and their projects. Common examples include multi-tier Java platforms consisting of both open source and/or proprietary application components, and Windows-based platforms. Depending upon the organization, open source platforms may be the preferred choice due to simpler software license management.

• **Dev platform ownership and support:** Ultimately, cloud-based dev platforms need to fall within the cloud service portfolio that enterprise IT must manage and support in their evolving role as a hybrid cloud service broker. However, this is dependent upon enterprise IT’s timeline to build the internal product management skills to understand and respond to end customer needs (including dev teams and their platform needs). Until that capability is in place, the best interim approach is to engage senior software developer resources with deep expertise in dev platforms to create initial platform offerings alongside the Agility Platform administrators with support from IT Ops whenever possible.

• **Agility Platform Blueprint best practices:** Because the Agility Platform blueprinting tool is very versatile, it is important to establish best practices early regarding dependency modeling and stateless architectures to ensure the reusability and cloud portability of platform offerings in the future. Your Agility Platform account executive can direct you to the appropriate ServiceMesh professional services and training programs to facilitate this.

• **Initial platform governance model:** Due to the controlled and limited scope of this implementation stage to a few select dev teams, a “just enough” governance approach can be used to encourage adoption and use. Governance is typically limited to basic RBAC and resource consumption constraints (leases, max limit on instances, etc) for development environments. The intent is to provide fully configured platforms so dev teams can be immediately productive, while also restricting the excessive platform customization and sprawl.

• **Capturing provisioning metrics:** Capturing platform provisioning metrics both before and after implementation will assist with justification of future on-boarding efforts. Be sure to capture the time it take to start using a fully configured platform, ready to use by developers, rather than provisioning bare VMs or individual application components. Be sure to differentiate and capture metrics from the perspective of the end user developers, versus the provisioning time calculated
from the perspective of an IT administrator to spin up the bare VMs. If possible, also attempt to capture metrics associated with compressing of overall dev project cycle times.

**Success Metrics**
- Reduced provisioning time for development platforms
- Reduced/eliminated time for developers to manually configure development platforms
- Improved developer productivity
- Compression of overall project cycle times

### Initiative 3: Multi-Vendor Private Cloud

**Business and IT Objectives**
- Lower cost private cloud deployment options
- Demonstrate cloud workload portability and vendor contestability, and with it, the ability to drive down overall IT costs and prevent costly vendor lock-in in the future
- Increase the private cloud program’s strategic importance/visibility and accelerate adoption

**Deliverables**
- Automated workload deployment to multiple heterogeneous private cloud implementations
- Workloads that are cloud portable and leverage stateless architectures
- Initial policies to automate workload deployment decisions across multiple clouds

**Enabling Technologies and Prerequisites**
- Additional private cloud implementations (e.g. OpenStack, Microsoft System Center/Hyper-V)
- Agility Platform adaptors to the new private clouds

**Process and Organizational Change**
- Expand policies and governance model to address multi-cloud use cases

**Implementation Best Practices**
- **Rapid pilots and rollouts**: Agility Platform cloud adaptors enable you to rapidly onboard and configure new clouds. Pilot programs and initial rollouts can be done for existing teams that are already using Agility Platform by rapidly exposing the new cloud deployment option for cloud portable workloads.
- **Cloud workload portability**: By adhering to Agility Platform blueprint best practices in the previous initiative, a range of cloud portable workloads with stateless architectures should already be available. It is important to note that not every existing cloud workload will be “cloud portable”, unless it has already adhered to best practices regarding stateless architectures. By adhering to these best practices, you can avoid the unnecessary rework of workloads later to support additional heterogeneous private and/or public clouds.
- **Initial multi-cloud governance model**: Because both clouds are private and cloud workloads will originate from the same self-service store, a “just enough” governance approach is appropriate to facilitate adoption. Initial multi-cloud policies are usually focused on provisioning constraints to ensure workloads are compatible with target cloud environments along with aligning roles and teams to the most appropriate cloud operating environment and security posture.

- **Increasing project visibility**: Leverage your multi-vendor private cloud as a working model of vendor contestability to increase the project’s strategic importance as a vehicle to drive down IT costs and prevent vendor lock-in. Leverage this visibility to recruit new teams and accelerate the migration of other workloads to the private cloud.

Success Metrics
- Reduced licensing cost by using open source and/or lower cost resource managers and private cloud implementations

### Initiative 4: Incubate app release automation for dev team early adopters

**Business and IT Objectives**
- Prove out the business agility and time to market value of application release automation for an actual production application
- Break down organizational silos across the SDLC within selected teams to automate release processes and define common deployment environments
- Reduce environment configuration defects and improve software quality
- Gain additional dev and QA sponsors and champions

**Deliverables**
- SDLC tool chain integration with the Agility Platform
- Software release approval workflows implemented with Agility Platform Release Manager
- Policy-based automation and control of deployment environments across different SDLC stages

**Enabling Technologies and Prerequisites**
- Agility Platform adaptors for SDLC tool chain products (IDEs, continuous integration servers, build servers, artifact repositories, source control systems)

**Process and Organizational Change**
- Bring Dev, QA, and Operations together (just within the scope of this team or pilot) to define and govern Common Application Platforms
- Establish software release approval processes, typically through pre-production stages, using Agility Platform’s Release Manager module
- Create and maintain expanded RBAC policies to include Release Management roles
- Create and maintain policies to govern deployment environment for different SDLC stages

**Implementation Best Practices**

- **Pilots and early adopters:** Pilot programs and early adopters of application release automation will make more organizational impact and gain more visibility if the implementation is based on a business application that is currently important/strategic to the business. Good application candidates are typically customer-facing and/or subject to frequent change. In addition to business context, candidate applications must also be evaluated on important technical screening criteria, such as the ability to use existing blueprints, templates, and packages in the current IT portfolio, and the complexity of the application’s SDLC tool chain.

- **SDLC tool chain integration:** DevOps pilots will implement faster when the client uses tool chain products recommended by ServiceMesh. In addition to speeding implementation, this enables more time and analysis to be focused on collaboration areas between Dev and Ops teams, approval workflows, Common Application Platform definition, and other high value areas that can drive significant benefits beyond product integration details. Of course, in cases where the project must use an existing toolchain that is not currently supported by Agility Platform, standard best practices such as adequate upfront project scoping and early access to tools is necessary. This is important because many legacy tool chains were not designed for DevOps automated workflows, and the custom integration work required can be significant.

- **Initial collaboration efforts across silos:** Often, IT Ops groups have limited to no visibility into the architectural decisions and design of the business application they are asked to enable for application release automation. To facilitate rapid modeling of a new business application for release automation using Agility Platform, it is important for application owners (architects, development leads, etc) to share information upfront on application design and architecture. This will eliminate reverse engineering an existing application into an Agility Platform blueprint, and also foster collaboration to improve an application design to take advantage of cloud-native architectures (such as autoscaling).

- **Application release automation through pre-production:** Continuous Delivery is the ability to implement DevOps principles from development all the way to production deployments. This is often the eventual end state goal, but almost all initial projects and pilots do not proceed down the entire SDLC path to production, but instead stop short at a pre-production stage such as UAT or Staging. This approach allows organizations to address the initial technical, workflow, and organizational barriers between Dev, QA, and UAT/Staging first before tackling the more stringent barriers going into final production.

- **Benchmarking:** Before starting a pilot or initial program with an early adopter, you need to define what success looks like. Is it more frequent releases? Higher-quality releases? Fewer
failed deployments? Once you have decided the appropriate goals, document current benchmarks prior to implementing the Agility Platform so you can measure against them later.

Success Metrics
- Improved pace/frequency of software release to get business functionality out to market faster
- Time/cost savings through continuous build
- Time/cost savings through simplified application launch, updates, patching, and roll-backs
- Time/cost savings through automated/streamlined release approval processes
- Time/cost savings through automated/streamlined application change requests
- Improved the quality of software releases

Initiative 5: Standardized Platform Portfolios

Business and IT Objectives
- Improve business agility by providing a standardized portfolio of self-service platforms to support a broad range of enterprise projects and initiatives
- Reduce one-off, customized platform sprawl
- Lower IT support costs
- Implement standardized platform security/operating environments to mitigate non-compliance risks
- Reduce ungoverned Shadow IT by introducing a broader portfolio of platform offerings to address common use cases
- Significantly expand footprint of private cloud services across the organization

Deliverables
- Standardized platform portfolio (includes databases, app servers, webservers, etc)
- Platform security/operating standards (above OS level)
- Policy-based automation and control for platform-specific regulatory compliance use cases

Enabling Technologies and Prerequisites
- Standardized application components and services (e.g. databases, app servers, etc)
- CMDB integration to support platform/app configuration and tracking
- Regulatory compliant security and operating environments (e.g. PCI-compliant clouds) for different platform use cases

Process and Organizational Change
- Dedicated Onboarding and Support Team to bring new end user teams onboard, upkeep standards, provide Agility Platform administrative support, and update/maintain policies for RBAC, leases, instance limits, and regulatory compliance
• Dedicated Product Management role/team to select and manage new platform offerings and drive new cloud services into the organization
• Dedicated Internal Marketing role to promote cloud services across the broader organization and gain further understanding of the end user’s IT service needs
• Create and maintain policies for more complex regulatory compliance scenarios in collaboration with business application owners and the Chief Compliance Officer’s office
• Expand traditional IT demand and capacity planning functions to encompass this initiative’s new cloud platform portfolio

Implementation Best Practices

• Selecting standardized platform: Work closely with end users to identify current platform services that will benefit the business due to their from greater agility and on-demand availability. Involve your top development teams in determining which application development platforms to standardize moving forward. Failure to include these teams and their feedback in the platform standardization process means you may be build a portfolio they will not use.

• Enforcing platform standards: Large enterprises can have hundreds or thousands of different platforms permutations and one-off configurations deployed in their data centers at any one time. Platform standardization initiatives driving that number down to 5 or 8 platforms (for example) will encounter resistance, and teams losing their best-of-breed platforms in favor of “good enough” alternatives will complain. Help these teams understand that the “old way” led to ever-increasing complexity, fragility, and costs in enterprise IT. Moving forward, enterprise agility will come at the expense of some individual choice. To encourage new platform portfolio adoption, IT organizations should continue to entertain requests for “custom” platform provisioning, but point out that standardized platform offerings can be made available on a self-service (i.e. real-time or near real-time) basis, whereas custom platform would take multiple weeks (or months) using exception-based, manual processes. This is the primary motivator for standardized platform adoption although support cost differences should also be highlighted.

• Consolidate orchestration and automation scripts and tools into Agility Platform: Existing IT automation and orchestration scripts and tools are often fragmented across the enterprise and are typically used to address a specific project or problem. As a result, there is little coordination or consolidation effort between them. This has the effect of raising costs, complexity, and impairing IT agility over time. Agility Platform can provide a single control point to consolidate this disparate activity into a more manageable, scalable, and agile solution. An inventory of scripts and automation tools currently used for your platforms should be cataloged and evaluated/prioritized for either migration into Agility Platform blueprint orchestrations and policies, or to be driven by Agility Platform policies via the tool’s API or CLI, if available.

• Dedicated product management: Dedicated product management resources become required when standardized platforms are published across an enterprise. This role will gather platform requirements, align them to platform offerings, drive/facilitate broad platform adoption across the organization, and recommend future platform additions or removals including evaluation of
public cloud services. This role will grow in importance over time, and is responsible for driving continual cloud service innovations into the organization and assisting with demand and capacity planning.

**Internal marketing:** Dedicated marketing resources are required to promote, evangelize, and help adopt new portfolio offerings across the organization. This extends beyond passive corporate communication vehicles (such as monthly newsletters) and executive sponsorship in departmental management meetings. Detailed “how-to” guides, tutorials, interactive lunch-and-learn sessions, and other face-to-face communication activities are required. These sessions should also be used to gather ongoing end user requirements and IT resource needs for future cloud portfolio upgrades.

**Cloud onboarding and support:** A dedicated team is required to provide operationally focused support and perform day-to-day administrative tasks to onboard new end user teams, update and maintain standards, orchestrate platforms and blueprints, and codify policies for RBAC, leases, and instance limits, and regulatory compliance.

**Governance and security collaboration:** Many multi-tier platform and application usage scenarios will require more detailed governance and security models. This can include industry regulatory constraints that affect security posture, provisioning within geographic regions, and other parameters. In addition, multiple security zones may be needed for the deployment of different tiers of a platform or application (e.g. the database tier in a secure private zone, web server tier in a DMZ zone, etc). For these reasons, the Dedicated Onboarding and Support team will need to sustain a high level of collaboration with other stakeholders and remain diligent to codify policies for governance and security posture rather than fall back into manual workflow patterns.

**Success Metrics**

- Compress project cycle times for teams that leverage standardized platforms, versus customized platforms with long lead times
- Lower platform support costs by automating the update/patching/trouble ticket resolution effort for standardized platforms
- Reduce security team manual costs by automating security posture in standardized platforms
- Lower audit costs by automating corporate and regulatory compliance into standardized platform entitlement, security, and deployment policies
- Percent of new projects leveraging new platform standards
- Number of existing one-off platforms migrated to the new platform standards
Initiative 6: Hybrid Cloud Contestability

Business and IT Objectives

- Lower operating costs via workload portability across multiple private and public clouds
- Vendor contestability including price and negotiating leverage over software and cloud providers
- Leverage the hybrid model to support new business use cases, such as deploying apps to clouds in new geographies for business/market expansion
- Implement automated policy controls to optimize deployment decisions across clouds based on cost, performance, or other criteria
- Deploy new cloud native HA/DR application reference architectures to improve application performance, reliability, and costs

Deliverables

- Policy controls that entitle teams to public cloud services, and policies to optimize deployment decisions across public or private clouds
- Public cloud services assigned SLA, security, and other compliance/governance parameters as defined by central IT
- Basic pricing to compare cloud services when applicable (basic pricing could be fixed per month costs)
- Cloud-portal platform and application blueprint models, including reference architectures for HA/DR applications

Enabling Technologies and Prerequisites

- Additional Agility Platform cloud adaptors
- Enterprise SLAs/contracts with one or more public cloud service providers
- Standardized security/operating environments (including packages, agents, utilities, etc) compatible in new public clouds
- CMDB integration to support resource tracking across supported public clouds

Process and Organizational Change

- Dedicated Cloud Account/Relationship Manager(s) to provide single point of contact for the enterprise for account services, support escalations, SLA monitoring and enforcement issues, etc across one or more public cloud providers
- Dedicated Cloud Application Architect to create new cloud application reference architectures, promote reference architecture standards, and facilitate architectural reviews for select applications/workloads
- Expand the Cloud Onboarding and Support Team role to update/maintain policies for public cloud service entitlements, leases, instance limits, deployment optimization, etc
- Expand the Product Management role to select new public cloud service offerings and manage them in the cloud service portfolio

Implementation Best Practices
• **Selecting Public Cloud providers and services:** The Dedicated Product Management team should be responsible for driving new cloud services, including public cloud services, into the organization. Other strategic objectives should also be considered beyond the cloud provider’s service offerings. This can include cloud geographic regions to help support future business expansion, vendor contestability issues (see below), the cloud provider’s regulatory constraints (Patriot Act), among others.

• **Maximizing vendor contestability and negotiating leverage:** Agility Platform establishes IT as a hybrid cloud service broker, putting the enterprise in a position of negotiating strength with providers and vendors. Agility Platform capabilities reduce enterprise switching costs, such as cloud portable blueprints, a dozen different private and public clouds supported, workload onboarding wizards, and deployment optimization policies. Describe these to providers to help drive negotiate favorable terms. Also note the high volume of current private cloud workloads and your ongoing expansion of cloud workloads into the future.

• **Leveraging Agility Platform’s cloud benchmarking capabilities:** Agility Platform allows you to benchmark the performance of your private and public cloud providers, which can be leveraged when negotiating/renewing with providers. You can compare cloud provider performance across different operational parameters and limits, such as average VM provisioning time, number of failed or degraded instances, max number of concurrent provisioning requests executed, and others metrics.

• **Initial public cloud service roll-out:** The Dedicated Product Management team should be aware of specific customer needs/requirements for new public cloud services. Good candidates include teams that do not work with sensitive/proprietary workload and/or data, or have IT resource demand spikes that can be best addressed with elastic public cloud resources. Assuming these teams are already using private cloud services, Agility Platform can facilitate a very rapid roll out of the public cloud services directly to these existing teams.

• **Creating and Enforcing Reference Architectures:** Hybrid cloud can offer cost/performance advantages for HA/DR applications relative to more traditional data center dual site and offsite backup solutions. These trade-offs should be evaluated by the dedicated Cloud Architect role, who should also publish and maintain application reference architectures that are tailored for specific company and industry requirements.

**Success Metrics**

- Lower cost ELAs from enterprise software vendors
- Lower operating costs for commodity IaaS cloud workloads
- Lower operating costs for HA/DR application architectures
- Defer private clouds/data center build out by leveraging public cloud elastic IT resource pools

**Initiative 7: Enterprise DevOps and Continuous Compliance**

**Business and IT Objectives**
- Extend DevOps benefits of increased business agility and time to market more broadly across the organization’s dev teams
- Provide pre-packaged tool chain integration and release automation approval workflows to help break down organizational silos and accelerate broader adoption
- Ensure standards compliance for IT resource portfolios after deployment to mitigate risks, reduce environment configuration defects, and improve software quality
- Extend release automation across the SDLC to achieve continuous delivery for applications that differentiate heavily based on new features for competitive advantage
- Simplify audits and security compliance by implementing standardized platform configurations, governance models, and release approval workflows directly into the SDLC so that compliance is largely achieved through the deployment and release automation processes themselves.

**Deliverables**

- Agility Platform integration with a wide range of SDLC tools used within the organization
- Best practice software release approval workflows implemented with Agility Platform Release Manager
- Expanded governance model to encompass multiple BUs, departments, and teams for automation and control of application deployment environments across SDLC stages
- Definition and enforcement of standardized configurations of IT resources via Agility Platform configuration management policies and other configuration management tools

**Enabling Technologies and Prerequisites**

- Integration with a range of supported SDLC toolchain products used within the organization (IDEs, continuous build servers, code/build repositories)
- Configuration management tools such as Puppet, Chef, or CF Engine
- CMDB integration including updating of both workload commissioning and decommissioning

**Process and Organizational Change**

- Modification of traditional roles, responsibilities, and incentives to break down existing organizational silos and develop a shared understanding and personal responsibility for successful deployments, including:
  - Bring Dev and Operations together to define the configuration and governance of Common Application Platforms at the beginning of the project
  - Expand Dev team responsibilities to include some level of exposure to frontline support. Developers may resist, but they will quickly learn about the realities of running their code in the production environment, and they are also in a position to produce the fastest fix if the code has defects or is running poorly.
- Establish software release approval processes leveraging Agility Platform’s Release Manager
- Create and maintain expanded RBAC policies to include Release Management roles
- Create and maintain policies to govern application environment for different SDLC stages
- Create and maintain configuration management standards, default remediation actions if configuration drift occurs, and codify these into Agility Platform policies
Implementation Best Practices

- **Tiering your applications by priority for DevOps:** Application release automation and DevOps should be prioritized for those applications where a faster pace of innovation and new business features will provide the best business value. Typically, good application candidates are categorized as “systems of innovation”, which are often customer-facing and/or targeting fast moving markets subject to frequent change. Gartner’s report on "Pace-Layered Application Strategy to Guide Your DevOps Strategy" offers a good guide to begin segmenting your application portfolio and their suitability for DevOps investments.

- **Pilots and early adopters:** Pilot programs and early adopters of application release automation will make more organizational impact and gain more visibility if the implementation is based on a business application that is currently important/strategic to the business. Good application candidates are typically customer-facing and/or subject to frequent change. In addition to business context, candidate applications must also be evaluated on important technical screening criteria, such as the ability to use existing blueprints, templates, and packages in the current IT portfolio, and the complexity of the application’s SDLC tool chain.

- **Using single, rather than multiple, Agility Platform implementation across the SDLC:** When implementing DevOps, existing team silos may initially request separate installations of the Agility Platform, rather than sharing one common installation across all SDLC teams and lifecycle stages. However, a single Agility Platform instance is required to provide seamless release automation workflows across teams. A single instance is capable of providing the desired environment isolation and RBAC/entitlements to address the needs of multiple teams via policy control.

- **Integration with third-party Configuration Management tools:** Configuration management tools such as Puppet, Chef, and CF Engine are taking a more prominent role within enterprise IT. The Agility Platform supports these third-party tools and uses configuration management policies to drive them and enforce continuous compliance on workloads. The recommended best practice is to keep configuration management policies within the Agility Platform as a single source of truth, rather than distribute manifests and other configuration management artifacts across multiple tools. In addition, the Agility Platform contains much more context and detail on cloud workloads, and can perform more sophisticated remediation actions for configuration drift than can be achieved with Puppet, Chef, CF Engine, and other tools in isolation.

- **Cascading Roll-outs, Updates, and Roll-backs:** A best practice for organizations that use continuous delivery or deployment is to implement a cascading roll-out for production launch. This entails initial A/B testing, so that an initial subset of servers receive the new release and undergo live customer use and detailed performance monitoring to minimize risk before launching broadly across all servers. Agility Platform supports cascading roll-outs with a REST API so that external scripts can specify the desired server cascading sequence, while Agility Platform automates the orchestration and code deployment. For updates and roll-backs, the Agility Platform provides the option to decommission the entire instance and redeploy from scratch, or keep the environment running and simply redeploy the code. In addition, the Agility Platform
helps automate patching of underlying software packages with the option to automatically update all templates and blueprints that have a dependency on that package the next time they are redeployed.

- **Collaborative Development of “Common Application Platforms”:** Traditionally siloed SDLC teams are typically engaged for discreet portions of a project, and rarely collaborate upfront or implement effective feedback loops. Agility Platform’s blueprint definition of application deployment environments, a.k.a. “Common Application Platforms”, provides an important catalyst for IT Operations to get involved early to define and implement deployment environment best practices and standards, and simultaneously allow Development teams to explain how an application is expected to operate including dependency mapping and more. This upfront collaboration is key to reducing configuration mismatches and defects downstream.

- **Visibility into the Release Pipeline:** You need visibility into your software release pipeline and approval gates to identify bottlenecks and continually improve processes further. End-to-end visibility across SDLC stages will surface problems earlier and ultimately build confidence to move the overall pipeline faster. Agility Platform’s Release Manager module provides a graphical display of the release pipeline to support these objectives, and should be combined with other tools to trace defects and monitor the heath of applications as they flow through various stages (dev, QA, UAT, prod, etc).

- **Fully Leverage Standardization:** Reducing the complexity and fragility of application deployment environments is a key objective of DevOps initiatives. This includes building applications based on resilient, standardized application components and services that make it easier to deploy and maintain. Earlier initiatives were focused on driving adoption of standardized infrastructure and platform services, and those efforts should be fully leveraged in enterprise DevOps. The same rules apply to encourage end user teams to adopt standardized services, although these tactics can now be used to obtain the time-to-market benefits of enterprise DevOps.

- **“Automate Everything” Mindset:** Automation is the key to both compressing cycle times and establishing processes that contain less human error and improve software quality. The Agility Platform delivers important aspects of toolchain and process automation, but other automated systems are required. This includes the build process and automated test suites, among others. In particular, automated testing should continually strive to be expanded to enable faster release approvals. This goes beyond automated units tests to include integration, functional, and load/performance testing. The more robust the end-to-end automation framework, the better the release cycle compression and software quality. It’s easy for manual steps to creep into a process over time and so constant evaluation is required to ensure optimal automation is in place.

- **Continuous Delivery vs. Continuous Deployment:** Continuous delivery requires release automation across SDLC stages, but the final decision to deploy into production is based on business need rather than simply deploying every commit to production. Continuous deployment extends continuous delivery and results in every build that passes quality gates to be automatically deployed to production. For most large enterprise customers, continuous delivery
is sufficient and continuous deployment is not necessary. The level of automation and rigor associated with continuous deployment can be substantially higher, and should be reserved only for rare applications that are highly dependent upon innovative features and first mover advantage.

- **SDLC tool chain integration:** In cases where dev teams must use an existing toolchain that is not currently supported by Agility Platform, standard best practices such as adequate upfront project scoping and early access to tools is necessary. This is important because many legacy tool chains were not designed for DevOps automated workflows, and the custom integration work required can be significant.

- **Benchmarking:** For each dev team implementing DevOps, you should define what success looks like to them. Is it more frequent releases? Higher-quality releases? Fewer failed deployments? Once you have decided the appropriate goals, document current benchmarks prior to implementing the Agility Platform so you can measure against them later.

**Success Metrics**
- Improved pace/frequency of software release to get business functionality out to market faster
- Time/cost savings through simplified application launch, updates, patching, and roll-backs
- Improved the quality of software releases
- Time/cost savings through continuous build
- Time/cost savings through automated/streamlined release approval processes
- Time/cost savings through automated/streamlined application change requests
- Time/cost savings through simplified audits, as governance and security controls are embedded and documented in the continuous delivery process
- Time/cost savings by automating what was previously manual configuration efforts

---

**Initiative 8: Onboard Shadow IT**

**Business and IT Objectives**
- Mitigate a wide range of regulatory, legal, and security risks associated with cloud workloads purchased and operated outside of direct central IT control
- Enforce enterprise IT’s emerging role as cloud service broker across public and private clouds
- Define and enforce minimum security/operating environment standards applicable for any public cloud workload deployed and operated from within the organization
- Systematically onboard Shadow IT workloads by either migrating to other securely managed and governed clouds, or by enforcing security/operating environment standards on the current cloud

**Deliverables**
- Onboarding processes to bring Shadow IT workloads under Agility Platform governance control
- Migration processes to move Shadow IT workloads to other securely managed and governed clouds
- Minimum security/operating environment definition for all public cloud workloads
- Support, monitoring, and backup service options provided by enterprise IT for on-boarded Shadow IT workloads

**Enabling Technologies and Prerequisites**
- Agility Platform cloud adaptors to enforce security/operating environment standards for workloads in the target clouds
- Configure Agility Platform’s “VM onboarding wizard” including creation of onboarding policies to enforce the desired security/operating posture of on-boarded workloads

**Process and Organizational Change**
- Confirm/clarify Exec IT ownership and responsible for the effective use of IT services across the entire organization (not just to run the central IT organization)
- Revise IT procurement processes and policies to reduce/eliminate and report on Shadow IT resource requests
- Revise IT support processes to effectively report on Shadow IT support requests
- Expand the responsibilities of IT’s Dedicated Internal Marketing role/team to expand awareness of the risks of Shadow IT and the existence of suitable central IT alternatives
- Expand the responsibilities of IT’s Dedicated Product Management role/team to investigate the root cause of Shadow IT usage and gain further understanding of the end user’s IT service needs
- Expand the responsibilities of IT’s Cloud Onboarding and Support Team role to include Shadow IT workload evaluation, onboarding, and support processes

**Implementation Best Practices**
- **Assessing Shadow IT usage:** By definition, Shadow IT is difficult to see. Start by accessing procurement records, financial accounting, and IT support requests. Have the CIO communicate the issue of Shadow IT to other executive leaders and solicit help from their teams to assess the current state of Shadow IT and the risks and vulnerability from it.
- **Selecting Shadow IT workloads for onboarding:** Depending upon results of the assessment, several Shadow IT workloads may need to be on-boarded due to performance, security, or reputation threats. The onboarding process should be defined as a constructive effort that aligns the organization to a common governance model and security posture as defined by discreet policy controls, rather than simply policing any IT that is outside of the central IT department.
- **Onboarding and enforcing security/operating environment standards:** The Agility Platform provides a single control point for governance and security across hybrid clouds, and existing policy controls can be modified/reused to establish a minimum security and operating environment standard for workloads deployed to a support public cloud. Public cloud workloads that were previously deployed by another management tool can be on-boarded to the Agility Platform including the automatic enforcement of selected governance policies using Agility’s VM onboarding wizard. Agility Platform’s VM on-boarding wizard provides the means to rapidly bring dispersed Shadow IT workloads under centralized management and governance.
— **Internal marketing**: Taking a draconian stance to rein in Shadow IT will likely move it further into the shadows, and damage the IT organization’s reputation in the process. Instead, take the approach of acknowledging the existence of Shadow IT, promote awareness of the associated risks, and then offer programs to better manage these risks without threatening to shut Shadow IT down. Regularly encourage end users to the necessary actions to make sure their Shadow IT usage is in compliance.

— **Understanding the source of Shadow IT**: Shadow IT usage should be studied to understand what underlying issues led to its adoption and whether other alternatives are available in the enterprise IT service portfolio. Use Shadow IT as important input to the evolution of IT’s portfolio management, Product Management, and hybrid cloud service broker functions.

**Success Metrics**
- Monitor and track periodic assessments of Shadow IT usage (as described earlier)
- Measure the number of Shadow IT workloads on-boarded to governed IT environments
- Monitor and track the number and costs of audits associated with Shadow IT workloads

---

**“IT-as-a-Service” Phase**

**Initiative 9: Chargeback and New Enterprise IT Economic Models**

**Business and IT Objectives**
- Rather than run IT on a preset budget, establish the economic framework to “run IT like a business” as a self-sufficient enterprise cloud broker with internal P&L responsibility
- Change perception of IT as a cost center to a value added contributor to the business
- Provide cost transparency from applications down to infrastructure
- Enable workload deployment decisions to be optimized based on real-time cost models

**Deliverables**
- Pricing models for cloud services with appropriate margins for both internal and external cloud service offerings
- Departmental chargeback model based on IT resource consumption, rather than a fixed central IT budget
- Provide pay-as-you-go IT resource consumption
- Workload deployment decisions based on cost optimization policies, where applicable

**Enabling Technologies and Prerequisites**
- Accounting, cost control, and financial reporting system integration with the Agility Platform
- Real-time cloud service cost/price visibility to support cost optimization policies
• Real-time resource capacity information to support provisioning policies

Process and Organizational Change
• Change IT management mindset from running IT as a cost center occasionally at the expense of customer requirements, to giving customers a choice wherever possible and requiring them to pay for their choices
• Created a Dedicated Financial Analyst role/team to analyze internal and market costs/prices for comparable IT services, calculate price margins for IT service portfolios and various SLA levels, and work with resource capacity planners and others to adjust pricing to drive desired end user behaviors.
• The existing Dedicated Product Management role/team at this point should now understand core customer needs to the point where they can shift focus to offering the best service portfolio at the best combination of price and performance
• The existing Dedicated Internal Marketing role/team should position IT as having one major advantage over any external IT provider, which is the opportunity to become a true strategic partner with the business and drive value for both.
• Accounting processes for consumption-based billing must be created, including integration with existing chargeback processes and accounting systems
• Generate policies that govern consumption (quotas, leases, etc) from pre-allocated IT resource budgets with options for soft- and hard-limited over consumption
• Create consolidated billing for both internal and external cloud services, either from Agility Platform’s consolidated reporting or through accounting systems integrated with Agility Platform’s API
• More flexible and frequent resource capacity planning

Implementation Best Practices
• Creating Cost optimization policies: The Agility Platform allows organizations to begin centrally managing and driving systematic cost optimization activities from the bottom up across the broader enterprise. These can include policies to automatically recommend or select the best pricing and performance for comparable IT services. Policies can also enable you to “do the same with less” by reducing the size or capacity of previously selected IT services if workload demands can just as easily be accommodated with smaller size resources. These Agility Platform cost optimization policies can be enforced at a high level by default across the enterprise while still enabling ample opportunity to create more detailed policies lower in the organization or to override them where appropriate.
• Cloud service pricing models: The Dedicated Financial Analyst role/team is responsible to determining end user pricing models with the intent to run enterprise IT like a business with their own P&L responsibility. In addition, at this point in organizational maturity, the existing Dedicated Product Management role/team should already have a matured IT service portfolio and a reasonable understanding of existing IT service costs. However, based on prior levels of IT financial transparency provided to end users and existing budget cycles, many IT organizations decide to ease into chargeback pricing in stages, starting with showback consumption reports
and simple “cost plus” margins on common services to improve visibility and assist with future planning. From there, actual chargeback consumption and pricing typically begins on a small portion of the private cloud service portfolio and eventually expands to include multiple SLA options and select public cloud offerings.

- **Chargeback/Showback models:** The recommended best practice is to provide chargeback and showback reports based on the actual consumption of IT services used, rather than allocating costs based on abstract metrics such as percent of revenues or employees. The Agility Platform is designed to support this model, which fosters better transparency; more detailed cost optimization analysis, and answers to the question “why does IT cost so much?”

- **Consolidated IT service billing:** End users should be provided with consolidated IT service billing to foster greater management oversight, visibility into the alignment between business projects and IT. Agility Platform supports consolidated IT service billing for all IT services under management across both private or public clouds.

- **Tracking Actual to Forecasted IT resource consumption and costs:** The Agility Platform allows projects, departments, and other organizational units to track actual IT resource consumption and costs versus forecast amounts to support financial management and reporting, more frequent IT resource capacity planning, and to help track on-time project delivery.

- **Benefits Realization for IT projects:** (See "Benefits Realization: The Gift That Keeps On Giving" for best practices on measuring, tracking and realizing benefits on new initiatives.) IT needs to communicate the life cycle cost of a project at the time of project approval so that more than just implementation costs are funded and tracked. These new operational expenses are also required as part of the financial section of a business case to calculate financial metrics such as ROI, net present value (NPV) and payback period. Accountability for ensuring that business benefits are actually delivered is often missing in many enterprises, driving some IT organizations to lead the benefits realization effort in order to demonstrate the value of IT.

- **IT cost industry benchmarking:** With the increased financial transparency, you will be in a better position to compare and track your organization’s IT financial costs and performance over time while your peers.

**Success Metrics**

- Consumption of IT resources through self-service, chargeback versus existing consumption or purchase models
- Comparison of IT cost-performance model to external providers of similar services

---

**Initiative 10: App Migration and Portfolio Optimization**

© Copyright 2013 ServiceMesh, Inc.
Business and IT Objectives
- Gain the benefits of cloud-based IT operating models (agility, cost, maintenance, etc) for broad application portfolios
- Centralize governance and control over application portfolios through an extensible policy engine.

Deliverables
- Ranked list of application candidates for prioritized migration to the cloud
- Migration plans for applications that include categorizing the necessary modifications to the application to be compliant with common application platforms and standard operating environments identified in previous Initiatives

Enabling Technologies and Prerequisites
- Automated virtual machine discovery and onboarding
- Model and optimize applications for the cloud using Agility Platform’s blueprint designer
- Application-centric governance and security

Process and Organizational Change
- Review committee for application candidates
- Application re-platforming or re-architecture teams consisting of platform engineers and developers familiar with the legacy applications and platforms. This team will be responsible for analyzing the application to be migrated and implementing the necessary modifications to the platform and application itself.

Implementation Best Practices
- **Categorize the Applications to be migrated** – Categorize applications based on their requirements to migrate to cloud environments:
  - “Lift-and-Shift” – Snapshot application images and deploy to different clouds
  - **Re-platform** – Modify application to run on company standardized cloud platforms
  - **Re-architect** – Optimize application performance with cloud-native architectures
- **Utilize Agility Platform’s monitoring to analyze migrated applications** – Agility Platform’s monitoring tools can be applied to legacy applications to gather application metrics for use in policy controls and redesign efforts.
- **Apply policy-driven governance to legacy applications** – Legacy applications can benefit from Agility Platform’s policy-driven governance without complete re-platforming or re-architecture to reduce operating and maintenance costs, improve security, etc.
- **Re-platform and re-architect with Agility Designer** – Utilize Agility Platform Designer to redesign platforms that apply standard platforms consisting of operating systems, packages and scripts to provide compliance and improved security posture.

Success Metrics
- Reduced cost to support legacy applications
- Reduced infrastructure costs

Glossary of Terms

- API – Application Programming Interface
- CLI – Command Line Interface
- CMDB – Configuration Management Database
- CMP – Cloud Management Platform
- Dev – Software Development Organization
- ELA – Enterprise License Agreement
- HA/DR – High availability and Disaster Recovery
- IaaS – Infrastructure-as-a-Service
- IT Ops – IT Operations Organization
- OS – Operating Systems (Windows, Linux, etc)
- PaaS – Platform-as-a-Service
- QA – Quality Assurance Organization
- RBAC – Role-Based Access Control
- SDK – Software Development Kit
- SDLC – Software Development Lifecycle
- SOE – Standard Operating Environment
- UI – User Interface
- VM – Virtual Machine