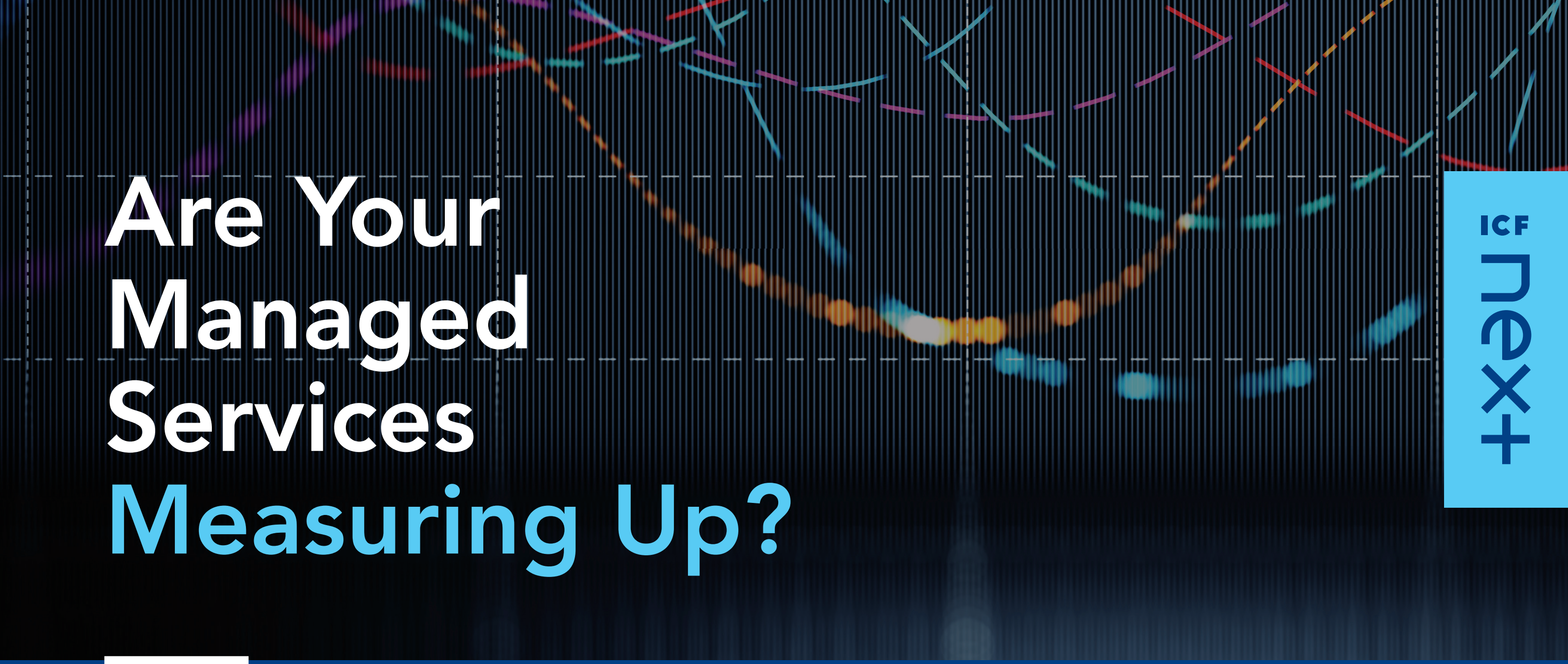

Are Your Managed Services Measuring Up?

Most Hosting and Managed Services providers view the infrastructure team as simply building a server, answering a ticket, or completing monthly patching. ICF Next approaches this differently with clients as we work with them to navigate the challenges around their complete web hosting solution, DevOps, continuous integration, and continuous delivery.

It is often difficult for technology and business leaders to stay ahead of new products, methodologies and technologies entering the market. We often see this result in spending too much money, taking on too much risk and ultimately unsuccessful solutions.

In addition to the fast-changing market, clients often struggle to balance the goals of the various parties involved in delivering a website or service. Marketing and Business Teams are focused on communicating information as quickly and cheaply as possible. Software Developers want to work on new and emerging technologies. Infrastructure teams, as we historically have thought about them, want to build something and not worry about supporting it. Security teams want to make sure that they maintain control and protect the environments.

In order to help guide our clients through all of these challenges, we have developed a Web Hosting Maturity Scale.

This Scale provides a single purpose, and more importantly, a single destination for our clients and our technology teams. It helps align the various teams and facilitates open conversation and collaboration. We use this with new clients to help assess where they are today, where they want to be, and what the value is to all teams (cost reduction, speed of delivery, uptime, etc.)

Our Web Hosting Maturity Scale focuses on six different areas that need to be matured over time to achieve the ideal state in the following areas: Software Development, Portability, Operational Capabilities, Management Capabilities, Validation/Verification, and Security.

Capabilities Overview

ICF Next's Platform Solutions group offers a comprehensive solution to developing, deploying and managing your cloud infrastructure. Our team can support the entire process from evaluation to ongoing support. We will start by guiding you through evaluating cloud platforms and offerings that best align with your application requirements and business objectives. From there, we will develop a migration plan, including a fully automated approach to deploying and managing resources. In addition, our dedicated Cyber Security team will help evaluate security regulations and ensure we are meeting compliance requirements. Once the application has been successfully deployed, we can turn over the ongoing maintenance to your team, or we can take over the 24x7 support; allowing your team to focus on higher-value business.

Maturity of Software Development is the most critical item in the maturity scale. Eventually, infrastructure, security, and control all become a software development activity. A defined SDLC and code storage strategy are needed to convert everything to code. Your branching and merging strategy have to be well defined and followed because you eventually merge your infrastructure and security code in with your core code base and then deliver it through the same CI/CD chain as your solution code. Maturing software development enables you to turn every manual function in infrastructure, short of putting servers in a datacenter, to a 100% repeatable, testable piece of code.

Security Maturity is often not thought about until after the fact. In an immature environment, it is something that is done to a project. Embedding your security settings, tools, reporting and scanning into the SDLC and deployment capabilities removes a significant roadblock or requirement for go-live activities and makes your compliance teams happy.

Furthermore, leveraging the orchestration and monitoring capabilities of your environment can help you detect and respond to security incidents in an automated manner vs. having someone manually correlate and determine what happened.

Unlike the other areas, your security team does not need to become middleware developers. They just need to be able to tell that team what is desirable and undesirable from a security standpoint, and they will code it in.

Portability is about separating your applications relationship with the infrastructure. As you mature, you go from a manual server build to a scripted build, to an automated build of your environment. Portability is almost holistically focused on the ability to build and rebuild your infrastructure on the fly anywhere. Adding container technology further enhances portability. The net result as you mature in this space is the foundation of being able to change the way you think about recovery and troubleshooting servers. As patterns are defined and matured, systems no longer require dedicated attention and human error is removed from your build process. Your target business savings are in labor, disaster recovery and backup costs.

Operational Capabilities are really focused around the ability to deploy services to servers, configure them and prevent them from changing to an unknown state. This is where desired configuration management takes place. CI and CD deployment chains are not just used for deploying application code, but also the deployment and configuration of platforms, like Tomcat, that the application uses. Entire chains of dependency are built to deploy full blown applications such as AEM and Sitecore, fully configured.

Many clients choose to have us add in APM tools like AppDynamics to improve their visibility of application performance. Like Portability maturity, Operational Capabilities add to the ability to build your application anywhere.

Virtually the same code is reusable on any platform as long as it's got Windows or Linux running on it. If done correctly, this capability will eliminate the need for detailed manual documentation of your environments and the need to maintain that documentation.

Maturity in Management Capability is centered on the ability to maintain something that is already there. This includes patching, hotfixes and versions of software and components as well as the ability to monitor everything at the component level. When starting out, you have tons of labor go into this space. Planning for upgrades and patching is done manually, but as you mature and begin to progress these items through your SDLC model, you eliminate the need for things like patching separately or having change control meetings. Servers begin creating their own monitors in the system, registering themselves the CMDB tool, while the orchestration engine can begin making decisions on conditions within your environment to scale or repair failed states. Labor is a key reduction in this space overall.

Validation/Verification is focused on ensuring the quality of the product and reducing the labor needed to validate it. While many of the areas above reduce labor and the need for a human to manually do something, this area's maturity builds the capability to validate that there are no issues. In the early phases of this, you have very manual testing of your application code. As this area evolves, your monitoring system and QA tools not only auto test the application, but the infrastructure code as it progresses through the deployment pipeline. Security scanning and load testing are sometimes also pulled into this area to completely test a solution in an automated fashion. Combined with your monitoring and orchestration systems, maturity in this area can remove significant labor from multiple areas in your organization.

Web Hosting Maturity Scale

	Software Development	Security	Portability	Operational Capabilities	Management Capabilities	Validation + Verification
5	Development team can self- configure infrastructure and environments	Pattern and behavior analysis with automated or alerting triggers	Platform agnostic	Automated blue/green methodology with triggered rollbacks and self documentation	Self-healing and auto-scaling	Development team can self-configure real-time monitoring and automated QA processes
4	Infrastructure is converted to code. It is versioned and follows the SDLC process	Security is actively a part of architecture and the SDLC, code scanning, site scanning part of QA	Hypervisor agnostic	Development teams are able to progress infrastructure code with app code	Centralized operational tasks as code. Manual interactions happen in development environments	Automated baseline monitoring and configuration generation
3	Code is versioned and stored in a central repo	Security Best Practices are embedded in infrastructure automation. Security done by all groups	Scripted installations and deployments for specific use cases	CI/CD deployment pipeline is used and delegated to the software groups	Structured release of updates through a change control process that follows SDLC	Automated QA processes and monitoring gap analysis begins
2	An SDLC is defined and in use	Role separation and 3rd party verification of environment. Central logging, scanning with triggers	Documented, reusable setup procedures	Scripts are used to build code artifacts	Manual tracking of systems, versions, use and configurations	Manual controlled monitoring system and QA processes
1	No SDLC is defined	Following best practice ACL based protections	Non-reusable setup procedures	Manual builds	Manual administration by a Systems Administrator or Developer	Manually checking code as it goes live, or waiting for customer complaint

Lower Support Costs, Higher Availability, Increased Security