

Cloud Native Computing Organizational Readiness Review

Introduction

This document provides an organizational readiness self-assessment framework for companies wishing to migrate to Cloud Native Computing. This review ensures that the technical and business alignment of the organization reflects DevOps and Cloud-Native best practices. Where gaps exist, the reviewer should seek effective ways to remediate and increase organizational efficacy, trust, and manageability. If optimally organized, your company will yield improved retention, employee satisfaction, and agility.

The foundation for these practices comes from multiple sources including the experiences of the author, Jaice Singer DuMars, the annual State of DevOps reports, industry research and the practices of highly-successful technical organizations.

Managing the Technology Lifecycle

Migrating a company to cloud-native application delivery requires a very different approach to all phases of technology management. No part of the business or engineering organization is untouched by the changes, but every one of them benefits. A high-performing technology organization provides tremendous value. For example, employees at such companies are 2.2 times more likely to recommend their workplace to friends and colleagues. The ramifications of that one benefit extend to every corner of the business, no matter how large the enterprise. Additionally, high-performing organizations experience 3x lower failure rates for changes introduced into customer-facing environments, 24x faster recovery times and near exponential improvements in lead times and deployment frequency. These environments are also characterized by high levels of peer and organizational trust, as they are optimized for predictable and repeatable results, both technologically and organizationally.

In terms of the technology lifecycle, strong relationships between business units and technology silos becomes even more important than with traditionally-deployed software since the lag between ideation and delivery is significantly decreased. Many parts of the organization may have to change the way they operate in response. For example, traditional program and project management is typically focused on waterfall processes, even where Agile has a strong foothold in development. This is because handoff points between product management, operations,

security, compliance, architecture, marketing and legal are not smooth enough to allow for rapid, iterative deployment. For this reason alone, many enterprises shrug off cloud computing for core business lines, limiting it to new or emerging products to experiment with. As a result, benefits do not necessarily spread across the company, and a mature, cloud-focused suite of processes and skillsets never develops. Some enterprises have misguidedly implemented "DevOps Teams" or DevOps as a job description to address gaps between traditional engineering organizations and cloud-optimized ones. This is not the answer, as will be covered in more detail later.

Organizational Review Process

Depending on the size and complexity of the organization, meetings and interviews should be held with individuals and teams. Representation will be required from all parties who have material involvement in the line of business moving to cloud-native computing. Ideally, executive management will also participate, as top-level buyin has a significant impact on the overall success of the initiative.

The questions in the assessment must be asked of the primary team affected, as well as incidentally from other groups so a rounded view of perception can be assembled. For example, Information Security teams may feel very prepared for cloud computing, but software engineers may feel security teams are not. Where conflicting or polar opposite views exist, additional teams or individuals should be consulted to better refine where trouble spots may develop. The aim is not to impugn anyone's efforts, but to instead limit risks by identifying and mitigating them from the outset.

Another important, and less qualitative strategy is to run retrospectives across teams or groups to get more unfiltered feedback on the current state. It is not uncommon for whomever conducts the analysis to receive a large volume of negative feedback since they are considered a safe harbor for that information, and may help institute change. This is an absolutely critical part of the process and may provide much value above and beyond the cloud implementation.

Assessment Scoring

The assessment is based on a 5 point scale with 1 being least true and 5 being most true. The scoring is subjective and based on observations gathered during the organizational review. Assessment questions are stated as facts. If the fact is rated a 5, then it will need little or no work. If it is rated a 1, it will require the most work. Urgency and severity can be assigned at the discretion of the reviewer if particular areas are rated universally low.

Assessment Statements

1. Program and Project Management

Program and project management is the conduit between business strategy and execution. Because cloud-native technologies remove blockages from the value stream, it can be difficult for organizations to manage iterative project delivery. Quarterly "big room" Agile planning events may be necessary to untangle dependencies and align stakeholders, departments, and engineering teams. Also, a key success factor is participation of delivery teams in plan generation since a tightly-coupled feedback loop encourages trust and limits waste.

1.1	Quarterly planning (or rolling planning) events occur to align technology implementation with business strategy.	
1.2	Project managers use Agile, Lean or comparable practices to ensure early feedback and adjust requirements accordingly.	
1.3	Programs with software components or technology focus have active, regular input from all stakeholders including operations, software engineering, compliance, architecture and security.	
1.4	Requirements are vetted with subject matter experts before being finalized and are adaptable to information gathered while the project is underway.	
1.5	A value stream map has been created for the delivery of program or project deliverables.	
1.6	Flow between project handoff points and silos is smooth and does not delay projects.	
1.7	Funding is allocated as a budget, not an estimate with program and project decisions optimized to best utilize the budget.	
1.8	Software projects are structured so delivery can happen iteratively, not necessarily at the end of the project.	
1.9	Strategic themes behind projects and programs are clearly communicated to not only stakeholders but also all project participants.	
1.10	Project prioritization is accomplished through a formal exercise such as "weighted, shortest job first".	
1.11	Corporate programs and projects use Kanban or other easy-to-understand way to radiate status to both senior leadership and stakeholders.	

1.12	Decision-making is decentralized wherever possible.	
1.13	Project value flow is closely monitored against customer demand.	
1.14	Business cases are only as detailed as they need to be to deliver customer value.	
1.15	Work breakdown structure is focused on estimating complexity and Agile estimation.	
1.16	Decisions are made with varying levels of certainty, acknowledging that the further out delivery is, the less certain the outcome.	
1.17	Projects use objective, fact-based measurements to determine progress, not arbitrary milestones.	
1.18	Project governance is focused on identified key performance indicators.	
1.19	Qualitative data is gathered throughout the life of the project to help inform and improve future projects.	
1.20	There is a high level of trust between the business and program/project managers.	
1.21	Business decisions around technology are primarily based on value delivery.	
1.22	Customers (not just stakeholders) of programs and projects are clearly identified and consulted regularly during the progression of work.	
1.23	Program and project-level retrospectives are held regularly.	
1.24	The organization handles change smoothly as part of normal operations.	
1.25	The governance model around cloud computing balances agility with risk management.	

2. Product Management

The product organization is the "why" behind everything that engineering teams do. If there is not a steel thread running between corporate strategy, product, and engineering, it can be a serious inhibitor to success. Product is the voice of the customer in every important conversation. Cloud computing presents the product organization with both an opportunity and a challenge: rapid delivery of customer value. Their job is to balance agility with non-disruptive delivery, as well as ensure that what got delivered is what customers actually want. This includes both internal and external customers. Additionally, operations and DevOps must become client-facing engineering organizations as well, complete with product representation.

2.1	Customers of software projects are well understood.	
2.2	Requirements are limited as much as possible, instead focusing on value delivery.	
2.3	The "voice of the customer" is always present in design, implementation, and delivery decisions.	
2.4	Product management has product staff (product owners) embedded directly in development and delivery teams to ensure their backlog is organized in accordance with strategy and commitments.	
2.5	Product backlog and prioritization is highly visible to stakeholders, participants, and executive management.	
2.6	Customers are part of planning events.	
2.7	Product managers and product owners attend engineering retrospectives.	
2.8	The feature release process is well-understood and flexible enough to accommodate continual product updates.	
2.9	Product managers leverage rapid delivery capabilities for business advantage.	
2.10	Product roadmaps are in a state of continual refinement in response to customer feedback.	
2.11	Product scope is regularly compared to the understood velocity of delivery teams.	
2.12	Features can be isolated and presented selectively to customers programmatically.	
2.13	Change is expected and adjusted for at regular planning events.	
2.14	The organization handles change smoothly as part of normal operations.	
2.15	Program vision and roadmaps are clearly understood by Product Management, as is the role of product efforts in the greater strategy.	
2.16	Product management effectively communicates the program vision and roadmap to delivery teams.	
2.17	Product value is clearly understood by participants, stakeholders and customers.	
2.18	Product Management drives the overall release cadence and increments.	
2.19	Product Managers and Owners attend engineering demos.	

2.20	Product teams have retrospectives after every significant customer delivery.	
2.21	Product teams work directly with release management to coordinate stakeholder progress on budget, release strategy and readiness of their specific features.	
2.22	Delivery teams trust product management to drive good technical decisions.	
2.23	Product teams trust delivery teams to accurately estimate their efforts and consistently deliver.	
2.24	At the software engineering level, features are broken down into small-point user stories that can be completed in a single sprint or less.	
2.25	Work in Progress (WIP) is limited on all delivery teams, and no team is required to exceed the WIP limit except under extraordinary circumstances, and only for very short periods of time.	
2.26	Product owners are fully versed in and practice Agile/Lean methodologies.	

3. Architecture

Enterprise architecture is vested with the challenge of maintaining long-term vision while actively exploring innovation. Because architects are typically seasoned technology and process veterans, they are able to temper desire for change with a need for consistency. Their role in cloud computing is critical because they must fully embrace the paradigm in order to reap the most value from it. If they are ambivalent about the cloud, then they represent a risk to viability in the organization. There may be serious resistance in architecture, specifically, because it upends so many traditional paradigms. The common resistances are cost, reliability and portability, which are not reflective of the current maturity of cloud providers.

3.1	Enterprise architecture is established as the governing body for strategic technical decisions.	
3.2	Architectural standards, guidelines and requirements comply with the best practices of 12-factor application development.	
3.3	Service-oriented architecture is the expected pattern for application development.	
3.4	When evaluating architectural decisions, preference is weighted toward cloud-based solutions.	
3.5	Container orchestration solutions are designed with partnership between architecture and engineering.	

3.6	Architects have significant, current experience with containerization.	
3.7	Architects have significant, current experience with cloud computing.	
3.8	Enterprise architecture is an active stakeholder in product decisions.	
3.9	Technology strategy is aligned at the architecture level with business strategy.	
3.10	Continuous integration and delivery is a key part of the architecture roadmap.	
3.11	Architectural decisions and projects are rolled out incrementally in accordance with Agile, Lean or comparable best practices.	
3.12	Enterprise architects regularly observe day-to-day engineering activities to get a better understanding of real-world impacts of architectural decisions.	
3.13	Service catalogs exist such that SOA consumers and providers are easily determined and documented.	
3.14	Operational data is fed back into architectural planning to address long-term scale issues.	
3.15	Total cost of ownership (TCO) is understood for cloud computing.	
3.16	There is a strong and regularly reinforced line of communication between architects, software engineering, and technical operations.	

4. Software Engineering

Software engineering is the engine behind the idea. These groups are heavily affected by cloud computing because it drives decisions all the way down to how they write and maintain code. Writing containerized microservices is a completely different mindset, skillset and experience than constructing traditional monolithic applications. The pivot to cloud-based compute resources may represent a major challenge from a hiring perspective.

4.1	Applications conform to 12-factor best practices.	
4.2	Software source control is used for every application.	
4.3	Deployment artifacts are immutable.	
4.4	Application scaling is strictly horizontal.	
4.5	There is an expectation that applications can be delivered to production environments programmatically.	

4.6	Application testing and delivery is automated.	
4.7	Engineering teams practice test-driven development.	
4.8	Testing teams leverage Agile, Lean or comparable best practices.	
4.9	Development environments provide an accurate reflection of target/production environments.	
4.10	Application-dependent services are differentiated via configuration value.	
4.11	Service discovery is ubiquitous.	
4.12	A service catalog exists.	
4.13	Rollback and more ideally, rollforward, are expected in all application deployments.	
4.14	Semantic versioning is used on all projects and deployments.	
4.15	There is a strong and regularly reinforced line of communication between software engineering and technical operations.	
4.16	Software engineers have a clear understanding of business, product and release strategy.	
4.17	Product owners are embedded in engineering teams.	

5. Quality Assurance and Software Testing

Testing and validation in cloud environments is relatively different than in traditional monolithic, datacenter environments. In many ways, testing is one of the highest beneficiaries of cloud-native applications since they are built with testing in mind, and can be spun up and down quickly and easily. Dependency management is also much better since 12-factor applications should be stateless. In full CI/CD environments, challenges may shift to maintaining build systems, and this can become a serious burden if not carefully managed. Test flakes can be the bane of quality engineers.

5.1	Software testing is accomplished completely through automation.	
5.2	Once all tests pass on code, it can be deployed without human intervention.	
5.3	Testing teams leverage Agile, Lean or comparable best practices.	
5.4	Test environments are as closely analogous to production as possible.	

5.5	Cloud-specific failure domains are regularly tested.	
5.6	QA and testing engineers are experienced with cloud-based application deployments.	
5.7	QA and testing engineers participate in up-front project/product planning.	
5.8	Global application test coverage exceeds 90%	
5.9	Continuous integration testing provides real-time status of application code branch deployability.	

6. Deployment/Release Management

In scaled Agile implementations, the release management process is a critical component of seamless delivery because it ties all of the actors together to ensure unblocked flow of value to the customer. This should not be a heavyweight process. It should be optimized through automation wherever possible. Human oversight should be focused on what value is being delivered, not how.

6.1	Applications can be continuously deployed programmatically.	
6.2	Deployment management is included in program, project and product planning.	
6.3	Non-delivery stakeholders such as marketing and legal participate in the release management process.	
6.4	The deployment and release management team is aware of and follows the release governance model.	
6.5	After major releases, deployment and release retrospectives occur with participation of the full release team.	
6.6	Business stakeholders have an easy and effective way of viewing release status at all times.	
6.7	Release cadence is not limited by the release management process, but rather governed by it.	

7. Operations

Traditional technical operations may offer some resistance to cloud computing since it represents a completely new paradigm. While many skills overlap between datacenter

operations and cloud operations, many tasks such as provisioning are obviated. A risk among traditional operations adopting cloud-native computing is the tendency to port datacenter practices over. For example, there may be attempts to use traditional configuration management practices inside of containers instead of relying on 12-factor practices. Also, network operations can be challenged by the use of overlays and policy agents. Because operations is arguably the most impactful to customer satisfaction (downed systems are the number one cause of customer exodus), having a product owner and running operations as a delivery team is critical. Cloud system agility can be completely blocked by a "waterfall" operations group.

7.1	Operations staff responsible for supporting production and other customer-facing environments are included as stakeholders in every phase of the release process.	
7.2	New infrastructure is deployed iteratively in the same manner as code.	
7.3	Infrastructure spin up and spin down is code-driven.	
7.4	All environments are under the auspices of version control.	
7.5	Operations follows Agile/Lean/comparable best practices.	
7.6	Monitoring and alerting is seamless between container, orchestration, cloud and non-cloud systems.	
7.7	Cloud utilization and efficiency is actively monitored and reported on to financial stakeholders.	
7.8	Production change management is enforced programmatically.	
7.9	Cloud quota management is reviewed on a regular cadence to ensure limits do not prevent deployments.	
7.10	Production environments leverage cloud best practices for redundancy and high availability.	
7.11	There is a strong and regularly reinforced line of communication between operations and software engineering.	
7.12	Manual operations on production assets is strictly limited or completely obviated by automation.	
7.13	All routine tasks are automated.	
7.14	Operations engineers have significant, current, relevant experience managing infrastructure, applications and system dependencies in the cloud.	
7.15	Operations leadership advocates for and promotes use of containerization.	

7.16	The operations organization leverages DevOps best practices.	
7.17	Operations staff are included in project and program inceptions.	
7.18	Technical operations is fully represented in Enterprise Architecture.	
7.19	Operational requirements are clearly communicated to all project and program stakeholders.	
7.20	Lead time for cloud infrastructure delivery is minimal or ideally trivial.	
7.21	Operations staff hold regular retrospectives.	
7.22	Operations has a product owner.	

8. Site-Reliability Engineering (SRE) or comparable

Groups responsible for maintaining infrastructure uptime are faced with one of the most challenging aspects of cloud computing since they must rely almost completely on self-service tools, automation, log aggregation, and documentation provided by engineering teams. Making sense out of service-oriented architectures is not straightforward. Also, direct access to systems may not be available, so empirical data can be hard or impossible to obtain. From the organizational perspective, SREs are a radically different group of people than traditional support teams. They need the engineering, troubleshooting, and testing skill sets. They must have a thorough understanding of the entire technical landscape, as well as how all the pieces fit together to provide any given application full functionality.

8.1	SREs receive actionable support documentation from software engineering and operations.	
8.2	SRE teams have clear supportability requirements for newly-crafted software or architecturally-changed existing applications.	
8.3	SREs are fluent in supporting and troubleshooting cloud environments.	
8.4	SRE is represented in project and program inceptions.	
8.5	SREs have a proven track record of appropriate issue and outage escalation.	

9. Incident Response and Escalation

Troubleshooting microservices, containerized applications and massive orchestration environments is a completely new world for many engineers. Even developers who construct

the applications and services may not understand how to properly diagnose and troubleshoot errors. As such, it's critical path for everyone involved in the process to be adequately trained, informed on changes, and part of the full deployment lifecycle. It's not good enough to simply hand off a "run book" to operations or reliability engineers. There is a tremendous learning curve, and it is measured in minutes of downtime.

9.1	The incident response and escalation procedure for cloud applications is documented and fully understood by all actors in the process.	
9.2	Service Level Agreements and Operational Level Agreements underlie the incident management process such that uptime is bound to the lowest uptime dependency -1.	
9.3	Production support is shared across the entire engineering organization.	
9.4	Self-service and automation are key tools in the incident resolution process.	
9.5	System uptime is calculated against the 90th percentile or better.	
9.6	Disaster recovery and business continuity programs are documented and fully tested on a regular cadence.	
9.7	Failover strategies include multi-cloud implementations.	
9.8	Product teams and customer relationship/account managers are included as part of the incident resolution process.	
9.9	Retrospectives are held after every top-severity service disruption.	

10. Security

Security should never be an afterthought in a cloud-native environment. With the agility of rapid iteration also comes the near constant introduction of vulnerabilities. Every system, process and design decision must put security front and center. Also, the working relationship between the security and engineering organizations must be extremely strong and collegial. When tension exists between these groups, it is very disruptive. It's also crucial for security to be involved in product decisions since most meaningful applications are data-driven, and data is the dominion of security engineering.

10.1	Cloud hosts are hardened according to applicable regulatory best practices, guidelines and requirements.	
10.2	Access controls are delegated and maintained through a centralized identity management system.	

10.3	Role-based access control defines a least-privilege model for individuals, systems, orchestration and applications.	
10.4	Securing systems in cloud environments is well understood by software engineers.	
10.5	Securing systems in cloud environments is well understood by operations engineers.	
10.6	Containerized applications are scanned for vulnerabilities as part of the release process.	
10.7	Security scanning is automated and run at a regular cadence.	
10.8	Patching of systems, container base images and orchestration components is automated as much as possible within the constraints of service level agreements.	
10.9	Orchestration environments are regularly assessed for vulnerabilities independent of release cadences.	
10.10	Security engineers are part of the initial project/program inception.	
10.11	Security leadership is a key stakeholder in the release process.	
10.12	CI/CD pipelines are considered secure by the security organization.	
10.13	The production infrastructure mirrors applicable design recommendations from the cloud provider.	
10.14	Container hosts have locked down the container runtime to least privilege.	

11. Compliance

No matter the industry, regulatory compliance will play a part in how systems are designed and maintained, as well as how data flows through. Having a strong partnership between auditors and delivery teams will make success in the cloud much easier. Successful organizations do not fight compliance, but instead embrace it.

11.1	The cloud implementation is considered regulatorily compliant by the appropriate auditing agency, department or third party.	
11.2	The regulatory framework applied to production infrastructure has explicit accommodation for containerized and cloud environments.	
11.3	Production cloud environments are fully compliant with corresponding	

	regulations.	
11.4	All cloud environments provide full audit trails for regulatorily-covered actions.	
11.5	Compliance monitoring is programmatic and accounts for ephemeral workloads and systems.	
11.6	There is a strong partnership between engineering and auditors.	
11.7	Compliance is not a burden for daily operations.	
11.8	Compliance overhead for new applications and services is minimal due to automated controls.	
11.9	There is an efficient and well-understood remediation process for findings.	
11.10	New controls needed are not instituted on a one-off basis, but instead programmatically and systemically.	
11.11	Engineering is proactive in determining compliance for new applications, services and systems.	
11.12	Container orchestration as a strategy is well-understood by auditors.	

12. Other parts of the business

A cloud-native organization extends to every corner of the business. Talent acquisition and retention is a major concern, as is the landscape around open source software in general. Sales and marketing organizations can use the cloud as a key differentiator.

12.1	Recruiters understand the specific skill sets necessary to staff a cloud-centric engineering organization.	
12.2	The business understands the value proposition of Agile/Lean/comparable practices and is seeking ways to expand them.	
12.3	Compensation for engineering staff is in line with the market.	
12.4	Incentives for engineers include remote work, skills enrichment and flex time.	
12.5	Engineers are encouraged to work on open source and community projects.	
12.6	Decision-making is as distributed as possible.	
12.7	All retrospectives are conducted with a discovery mindset not for blame or admonishment.	

12.8	Kaizen (the spirit of continuous improvement) is part of every process.	
12.9	Sales teams understand the competitive advantage of cloud-native application delivery and use it in pitches.	
12.10	Marketing teams understand the competitive advantage of cloud-native application delivery and use it in marketing campaigns.	
12.11	Legal staff understand the challenges and opportunities associated with creating and maintaining open source software.	

