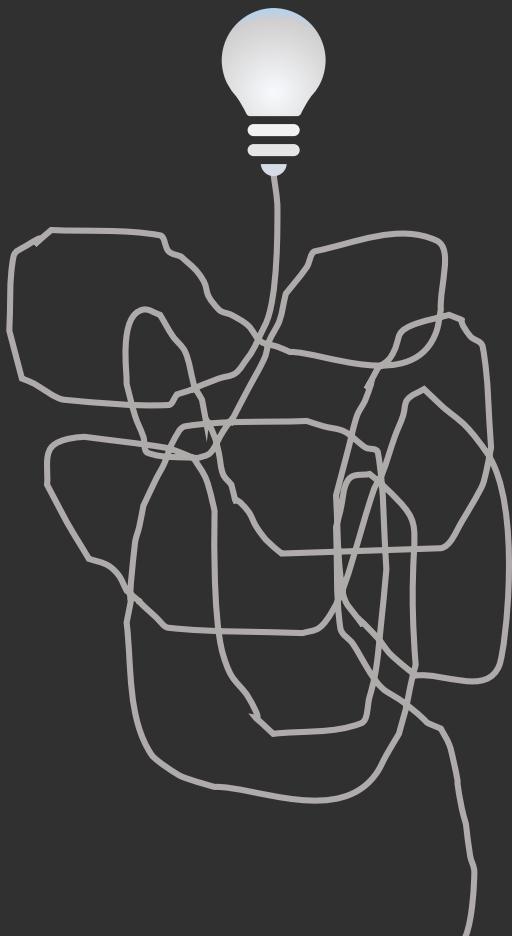


Helping **DevOps Transformation**



Keep it simple...

DevOps Maturity

	BASIC	INTERMEDIATE	STANDARD	ADVANCED
CULTURE SKILL	<ul style="list-style-type: none"> Prioritized Work Large Team Multiple Roles(Architect, Developer, Tester ,Release,....) Defined & Documented Process Quarterly Release 	<ul style="list-style-type: none"> Stable Teams Share Pain Remove boundary(Dev & Test) Agile Process Monthly Release Act On Metrics 	<ul style="list-style-type: none"> Team Responsible All the way Prod Dedicated Tools Team Remove boundary(Dev & Ops) Deploy disconnected from Release Continuous Improvement Component Ownership Decentralized Decisions 	<ul style="list-style-type: none"> All Full Stack Experts Cross Functional Small Teams Speedy Feedback Advanced Frameworks Driven RAD Excellent Training Framework
DESIGN ARCHITECTURE	<ul style="list-style-type: none"> Monolith Product Consolidate Platform System into Modules Library Management API Management Basic Automations 	<ul style="list-style-type: none"> Component Engineering Config As Code Event Architecture Service API interaction API Management Test / Release Automations 	<ul style="list-style-type: none"> Distributed Architecture Horizontal Scalability Cloud Centric, SAAS Model Deployment Automation Framework Operation Automation Framework 	<ul style="list-style-type: none"> Hybrid Cloud, Cloud Agnostic, Elastics Complete Software Defined.(IAAC) Multi cloud complete SOA Simplified Automated Framework Aggregated Hardware Container / Serverless Optimized Pooling
BUILD DEPLOY	<ul style="list-style-type: none"> Scripted Build Continuous Integration Dedicated Build & Release Server Multiplatform Packages Basic Test Automation Manual Staging Environments (Dev, Stage, Test, Prod) promotion 	<ul style="list-style-type: none"> Polled Build Manual Version & Tagging Automated Staging Automated Unit / Integration / UI / Regression / Performance / Code Security / Code Quality / Coverage / Acceptance Tests 	<ul style="list-style-type: none"> Quality Gates Realtime build metrics Deployment & Compliance Trend Test Results Consolidation TDD framework integration Dynamic Environment Provisioning 	<ul style="list-style-type: none"> Zero Touch Deployment Elastic Automation Cluster Parallel Continuous Delivery Container Based Build & Deploy Containerized Build environments for different language Simplified Build Orchestration Framework
TEST	<ul style="list-style-type: none"> Basic Test Automation Continuous Tests Integration Testcases maintained in Tool Risk Based Manual Tests 	<ul style="list-style-type: none"> Automated Unit / Integration / UI / Regression / Performance / Code Security / Code Quality / Coverage / Acceptance Tests Test Results Consolidation Component Wise Tests 	<ul style="list-style-type: none"> Well Defined TDD and BDD framework BDD Tests written in DSL Test Rule Based stage promotion Test Case Tool to Automation Tests Integration 	<ul style="list-style-type: none"> Service Wise Test Results Consolidation Tests Trend Analytics Continuous Tests in Production
OPERATION	<ul style="list-style-type: none"> ITSM Tools & Operations SLA's Performance Monitoring Log Monitoring Alerts Monitoring Manual resource provisioning Security Audit 	<ul style="list-style-type: none"> Diagnostics Advanced Monitoring through Scripts Log Analysis Automated ticket remediation & workflows 	<ul style="list-style-type: none"> Automated Hyperscale Cluster & Cloud Provisioning Elastic Auto Provisioning Resilience Scripts Dynamic VDMS / VDSS provisioning 	<ul style="list-style-type: none"> Hardware / OS / App server / License Agnostics Continuous Security / Audit / Billing Centralized Tools for Monitoring Different Environments, Products, Services, Clouds



Model

MAIN STEP OF PROCESS

Gap Analysis

STEP ONE

Using Maturity Model , establish
improvements in each capability



Mapping

STEP TWO



Map these to People / Process /
Technology



Execution

STEP THREE

Establish backlog and prioritize



Transition

STEP FOUR

Transition of knowledge and Artifacts to
internal team.



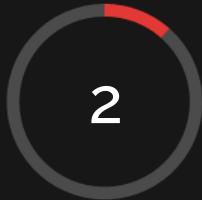
Gap Analysis

4 Phased – 5 Days



1 Day

Project Kick-off



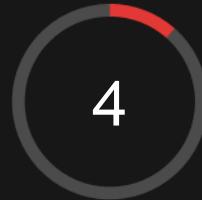
2 Days

Assessment



1 Day

Report Preparation



1 Day

Presentation

Initial Meetings

- Meeting with decision-makers to get an insight on client current pain-points and desired-future-state
- Kick-off meeting with DevOps assessment stakeholders

Questions Based Audit

- A DevOps briefing for all interviewees
- Audit on following key focus areas
 - Organization
 - Automation
 - As-is Analysis(People, Process, Tools, Technology)

Report Generation

- Preparation of Findings report, DevOps transformation Plan and Roadmap

Presentation

- Presentation of the report and plans to stakeholders, following plans are delivered:
 - Phase wise Maturity Development
 - Tools & Technology Adoption
 - Skill Development



Q's Culture & Skills

F/P/
N

1. What best describe your current state of team ? Please choose the following options:
 1. The team is producing higher quality products than before .The releases are more dependable than before.
 2. Team delivering functionality to users more quickly and/or more often than before.
 3. The team has higher morale than before. The business is recognizing greater economic value than before.
 4. The team is more responsive to our customer(s) than before. A speedy feedback driven process tool chain is established between customers requirements and engineering team.
2. Describe organization & Team structure for developing product ? Please choose the following options:
 1. Dev, Ops, Service Management, security QA teams are distinct and independent?
 2. QA is embedded with Dev Teams to ensure high quality in operations
 3. You have rightly adopted " You build it , You Operate " culture
 4. Security is interwoven into the development process
 5. Support Tickets are assigned immediately to dev teams without passing through Ops reviews
3. How would you describe how your dev and ops teams share tools?
 1. We are all one team, sharing responsibilities and using a common set of tools and sharing information.
 2. Dev and ops teams use a common set of tools but don't have visibility into each others' work.
 3. Dev and ops teams have different responsibilities and their own sets of tools, and they struggle to share data.
 4. Dev and ops teams use a common set of tools but share information manually.
 5. Dev and ops teams share some responsibilities but still use separate tools.
4. What is your organizational structure around service delivery?
 1. Development gets it to work and hands off to ops to finish the job
 2. Product management writes stories with operations as a top level concern
 3. Ops engineers are embedded in dev teams
 4. We have just one team, not two
 5. Our architecture is designed for operations and scale
5. What is your organizational state around technology delivery?
 1. The team do not have significant technical debt.
 2. The team incorporate refactoring into our daily work to reduce technical debt.
 3. Team can decide which tool team use .The tool choices are based on what is best for the people who use the tools.
 4. Team prioritizes keeping the software deployable over working on new features.
 5. developers can easily create environments that closely match production - even when physical resources are constrained for all dependent systems and without coding mocks and stubs?
6. How would you describe the project prioritization process in your company?
 1. Budgetary planning cycle (year, quarter)
 2. Small projects & iterative products
 3. Planning cadence (weeks)
 4. Continuous experimentation and releases
 5. Continuous planning
7. How would you describe the business requirement communication between teams?



Q's Culture & Skills

8. What is the ratio to unplanned / bugfix work vs. new features / improving infrastructure?
 1. Predominately new features, rarely unplanned work
 2. Mostly unplanned work, some new feature development
 3. Mostly new features, some unplanned work
 4. A balanced mix of both unplanned work and new feature development
 5. Predominately unplanned work, rarely new feature development
9. What best describe your current state of engineering team traits? Please choose the following options:
 1. Team consistently handle complex, multi-tier deployments for thousands of servers and complex services without having to rely on the development of one-off customized scripts or manual
 2. Development and testing teams automatically replicating/reproducing more realistic conditions from production systems to create accurate virtual services and testing scenarios?
 3. Developers consistently spot code quality issues early in the development process, before they get near to production?
 4. You have feedback mechanisms in place to ensure continuous assessment and drive improvement and progress?
10. What best describe your current state of engineering team skill sets
 1. Distinct Team of Product Manager/ Architect/Dev/QA/OPS , with Dev/QA having some key expertise on some languages / Framework .
 2. Dev team does write some automation tests , but there are separate QA and Operation Team
 3. Product Management team aware of operation concerns, write stories with operation in mind, architects have adequate knowledge of infrastructure & applications both and rest all engineeri
8. What best describe your criterion for current state of resource hiring
 1. Mostly based on specific skill & framework, fit for specific area of development & QA.
 2. Development resources who has holistic view on both Infrastructure and Application
 3. Architects who have capability to draft multi cloud hybrid architectures
 4. Product Managers having thorough understanding of operation aspects.
 5. Developers skilled on development/deployments / operations
9. What best describe your training process
 1. No formal training in place , only happens through project level mentoring
 2. You have some defined training for language & framework in place i.e used in projects.
 3. You have training program available that guide your product managers to gain operation excellence and engineering team to gain DevOps excellence.
 4. You continuously improve your training based on latest framework and technologies and input from Architecture team.
10. How would you describe the project prioritization process in your company?
 8. Budgetary planning cycle (year, quarter)
 9. Small projects & iterative products
 10. Planning cadence (weeks)
 11. Continuous experimentation and releases
 12. Continuous planning

Q's Build & Deploy

F/P/
N

1. How would you describe your typical deployment process for key projects?
 1. Team develops code using continuous integration (CI)
 2. Team commits and pushes to a shared main branch/trunk at least once daily
 3. Each commit to main branch starts an automated build.
 4. Team members have an option to start a series of automated tests as they see fit.
 5. When the build fails, the errors are usually fixed and the new build is run as soon as possible.
 6. Automated tests are regularly run at least once per day and are part of the standard workflow.
 7. You keep the number of our branches to a minimum and not more than five active branches on the application's code repo.
 8. Your team never has code lock periods when no one can check in code or do pull requests due to merging conflicts.
 9. On my team, we can perform deployments during normal business hours with negligible downtime.
 10. Our software is in a deployable state early in the lifecycle.
 11. We don't fear code deployments.
 12. Code deployments are not disruptive, easy and pain-free.
 13. When the automated tests pass, I am confident the software is releasable.
 14. When people get feedback that the system is not deployable (such as failing builds or tests), they make fixing these issues their highest priority.
 15. Fast feedback on the quality and deploy ability of the system is available to anyone on the team.
 16. My organization deploys code to production more often than once per month.
 17. The majority or all of our code is deployed automatically to various environments.
 18. We can deploy our system to production - at any time - on demand.
 19. If there are deployment issue, the process is automatically aborted and the new features are rolled back without excessive friction.
 20. The lead time between a line of code has been written in development and available to live users in production has decreased over the last three months.
 21. Every change that passes the automated tests is deployed to production automatically.
 22. Are your teams fully proficient at handling any unforeseen issues that cause the rollback of an application release?
 23. Are your teams establishing continuous feedback loops to share important information before and after each release?
 24. Do your release deployment processes include integrating Continuous Integration and Build tools, automated Testing tools, Containerization (e.g. Docker) and/or Infrastructure Configuration and Cloud Provisioning tools?
 25. Are your application release deployments fully automated end-to-end across development, test and production environments?
 26. Can developers easily create a sandbox that matches production?
 27. Can you deploy new versions of your application to production without downtime? Do you support blue/green deployment?
 28. Are deployments automated end-to-end across development, test and production?
 29. Does your release process include comprehensive auditing and reporting capabilities?
 30. Do your release and deployment practices provide comprehensive auditing and reporting capabilities to ensure IT governance and compliance?



Q's Build & Deploy

F/P/
N

1. How would you describe your typical release process for key projects?
 1. Continuous deployment to production augmented with experimental releases
 2. Continuous deployment to production
 3. Traditional waterfall release process
 4. Continuous delivery to pre-production
 5. Continuous integration, automating code into the master branch
2. In general, how do applications perform once released into production?
 1. New application releases have major issues in production forcing rollbacks
 2. Continuous integration reveals potential problems prior to release
 3. New applications occasionally have minor issues requiring fixes once in production
 4. Continuous delivery and continuous improvement prevent the need for rollbacks
 5. Continuous deployment enables quick rollback to remediate operational issues
3. How quickly can developers see the results of the integration / unit / etc. testing?
 3. We do not have any testing in place
 4. 10 minutes to 1 hour
 5. 1 hour to 1 day
 6. 1 day or greater
 7. 10 minutes or less
4. How would you describe your build deployment technology adoption and implementation
 3. A elastic deployment automation framework is available central to organization
 4. We use container technologies heavily to provision build and deployment environment dynamically
 5. The organization use modern framework to unify the deployment of various different components of the system
 6. The deployment framework can be extended quickly to meet the business SLA
 7. The Development team is capable of creating integration tests environment in their local machine rapidly
 8. The build & deployment trend analytics is available for system quality improvement
 9. The team has integrated unit , integration, code quality , coverage , code security , ui , regression , performance , acceptance tests as a part of continuous delivery process
 10. Continuous delivery statistics are visible across the components and services from a single window
 11. Can your team provide comprehensive visibility into your application release activities and timing to all major stakeholders?

Q's Tests



Team Roles

F/P/
N

1. What best describes your testing environment?
 1. Hostile environment testing
 2. Load testing / redirecting transactions from production into a test environment
 3. Unit testing
 4. Automated integrated testing
 5. Manual testing with testing environment separated from development
2. What best describes your test efficiency?
 1. your teams quickly generate, discover and mask sensitive data/messages from existing systems, and/or generate synthetic data sets to increase test coverage and reduce search times?
 2. Is testing (including unit, functional, load, regression and performance) fully automated across the SDLC, with validation mechanisms and strategies shared across teams?
 3. As requirements are updated/changed, can systems auto-generate test-cases along with access to the required test data?
 4. Automated tests are regularly run at least once per day and are part of the standard workflow.
 5. Teams develop and test against systems that behave like the production system.
 6. All team members are responsible for creating and maintaining automated acceptance tests.
 7. Test failures are likely to indicate a real defect.
 8. Teams can easily access a common pool of tests so they can reproduce failures.
 9. Every change that passes the automated tests is deployed to production automatically.
3. What best describes your test framework adoption?
 1. You maintain all your test cases inside a tool.
 2. You have a test framework in place that can integrate all your unit, integration, UI , Performance , Acceptance tests and generate test results in central location
 3. Your team is skilled at adopting a matured TDD and BDD framework where they can easily hook the acceptance tests
 4. The test framework can be easily integrated with the continuous delivery system to show consolidated test results for each tests on each services
 5. Your team has adequate knowledge and understanding of writing DSL based BDD driven tests
4. What best describes your test automation capability?
 1. You have automated unit/integration/UI/Regression/Performance/ Code Security/Code Quality / Coverage/ Acceptance Tests
 2. Your automated tests are written per component wise and you maintain a finer level of separation of your actual code and automation test code
 3. You have a right integration of test case tool to automation tests
 4. You have a visibility of component wise different test results in a single place
 5. You have real time tests trend analytics present to analyze tests effectiveness
 6. You have some availability and functional tests that continuously run in production to check system reliability



Q's Design & Architecture

State (Full , Partial, Not)
<p>1. How would you describe the state of design processes in your company</p> <ol style="list-style-type: none">1. We regularly evaluate the design of our system and make incremental changes to it.2. We regularly set aside time to work on improving the design of our system and reduce technical debt, even if it means delaying feature delivery.3. We can make large-scale changes to the design of our system without the permission of somebody outside the team.4. In general, architects collaborate regularly with teams to evolve the design of our systems.
<p>2. How would you describe the state of architectural adoption in your company</p> <ol style="list-style-type: none">1. We have mostly large products managed by large teams, having effective library and API management in place2. We apply component engineering effectively and our products mostly composed of components interacting through api and service buses.3. Every component of our product is distributed and Horizontally scalable4. We are developing microservices capable of being deployed to any cloud , each having a right blend of auxiliary services and databases (SQL/NOSQL/CACHE/GRAFH/BIG)
<p>3. How would you describe the state of cloud architecture in your company</p> <ol style="list-style-type: none">1. Our workloads are mostly residing in our private clouds and pure physical environments2. We have tools and framework available to distribute our workloads in our private cloud as well as specific public cloud provider.3. We have adopted hybrid cloud wherein public clouds are connected in private address space4. We are creating optimized resource pooling in hybrid cloud and deploy our services across the clouds
<p>4. How would you describe the state of application delivery model</p> <ol style="list-style-type: none">1. Applications are delivered as packages for multiple OS environment2. Applications are delivered as host templates capable of running to any cloud on the fly3. Application services are delivered as container with build once run anywhere model4. Application services are delivered with right blend of container and faas model
<p>5. What best describe the state of infrastructure provisioning capability in your organization</p> <ol style="list-style-type: none">1. We provision server , network , storage primarily through automation scripts and some are manual2. We provision entire infrastructure in software defined manner in our private cloud through the cloud orchestrator3. We provision the infrastructure and pool resources across the cloud in complete software defined manner



Q's Operations

F/P/
N

1. What best describes your operation monitoring capability?

1. Monitoring Tool in place which can monitor the following from single window

- Service cluster performance
- Application services performance
- Application availability
- Database performance
- Load balancer statistics
- Network traffic
- Physical server performance
- Virtual Server performance
- Storage performance

2. Monitoring practices scale to manage more modern distributed and microservices-style architectures with many new languages and data stores and potentially thousands of services to support?

3. Teams can visually depict changes in application environments to quickly identify when and where changes occurred, together with the resulting impact on performance?

4. Is your monitoring able to distinguish the most critical and recurrent problems and how they impact business services, without lengthy configuration and baselining?

5. Has your organization been effective in moving monitoring into development and test - making it an established element of the process in those areas to prevent defects from getting into production?

6. Do you have complete end-to-end visibility into all business services and transactions - from customer engagement via mobile devices and apps, through middleware and databases to back-end systems-of-record.

7. How effective are your teams in monitoring information and analytics on mobile app behavior and usage to not only resolve problems, but also drive development improvements?

8. Can your teams monitor complete business services that link end-user experience, end-to-end transactions and applications to infrastructure and pinpoint root cause and potential impact?

9. Are your teams able to quickly correlate and analyze multiple application performance metrics to solve complex emerging problems before they impact end-user experience?

10. Do teams have visibility into system capacity in order to understand the impact of increased system workloads?

11. Are your teams aware of third-party dependencies to ensure reliable service delivery?

12. Do teams have a unified view that enables them to quickly pinpoint and triage problems?

2. What best describes your operation log analysis capability?

1. Central Log collection and search tool is present and team is trained on that.

2. You have intelligent log analysis tools present that continuously gives insights to application services reliability & performance.

3. Your log analysis tools are integrated with ITSM tool to raise alerts depending on log analysis results.



Q's Operations

F/P/
N

3. What best describes your ITSM process capability?

1. You have a central ITSM tool and Operation SLA's defined and different teams in operation address different tickets
2. You do resource provisioning manually through ITSM tickets
3. You have intelligent tools that does automated ticket remediation
4. Your resource provisioning is entirely automatics , the architecture guarantee automated elastic provisioning of cluster , services , and associated resources.

4. What best describes your operation log analysis capability?

1. Central Log collection and search tool is present and team is trained on that.
2. You have intelligent log analysis tools present that continuously gives insights to application services reliability & performance.
3. Your log analysis tools are integrated with ITSM tool to raise alerts depending on log analysis results.

5. What best describes your audit capability?

1. You have tools and scripts in place to do regular application and infrastructure security audits
2. You have audit trails tools present to perform audit trailing
3. You have continuous security and continuous compliance system in place that gives you real time visibility on security and compliance
4. You have centralized tools for auditing different environments (dev/stage/qa/tests) , products deployed across clouds in real time

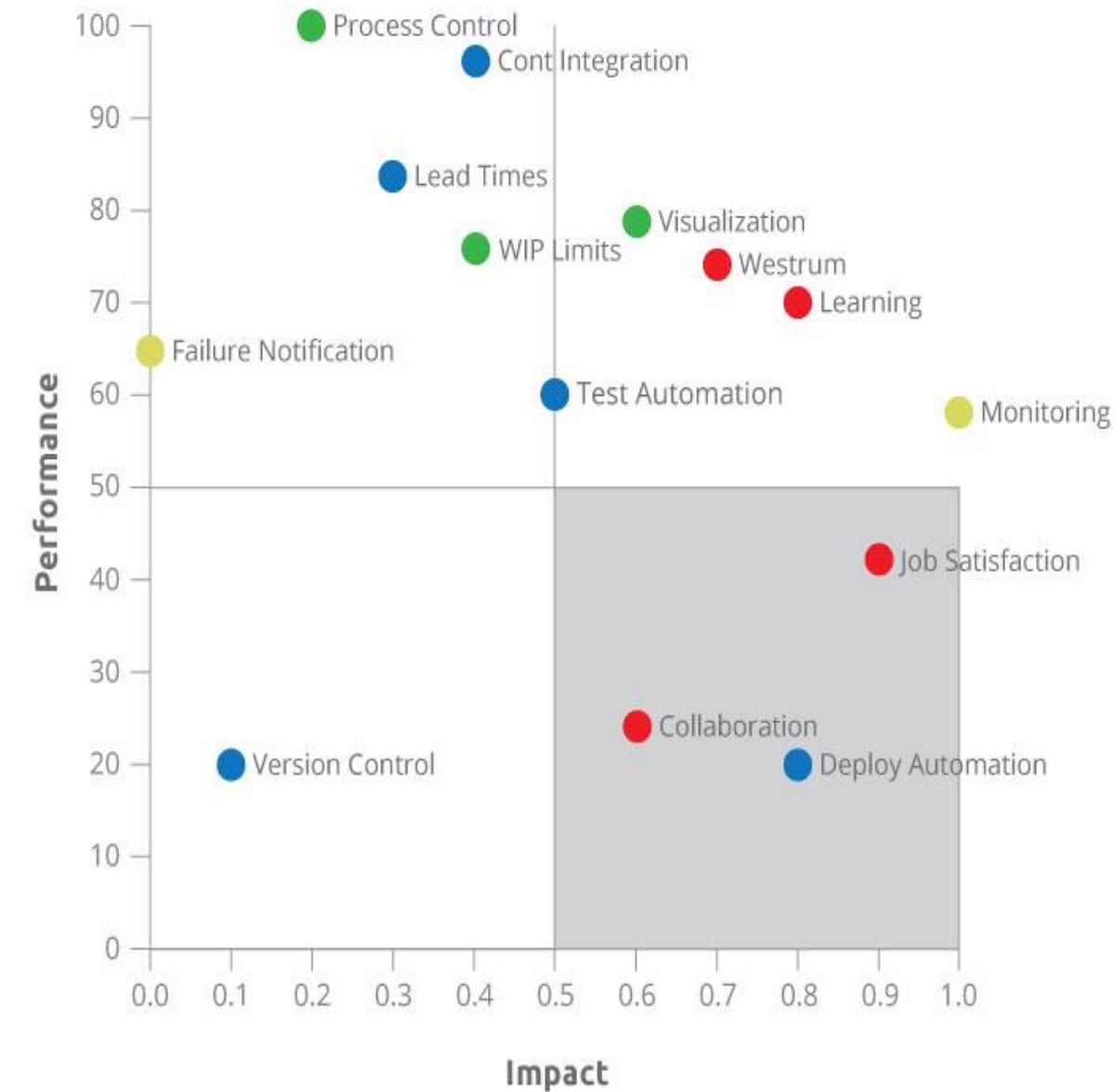
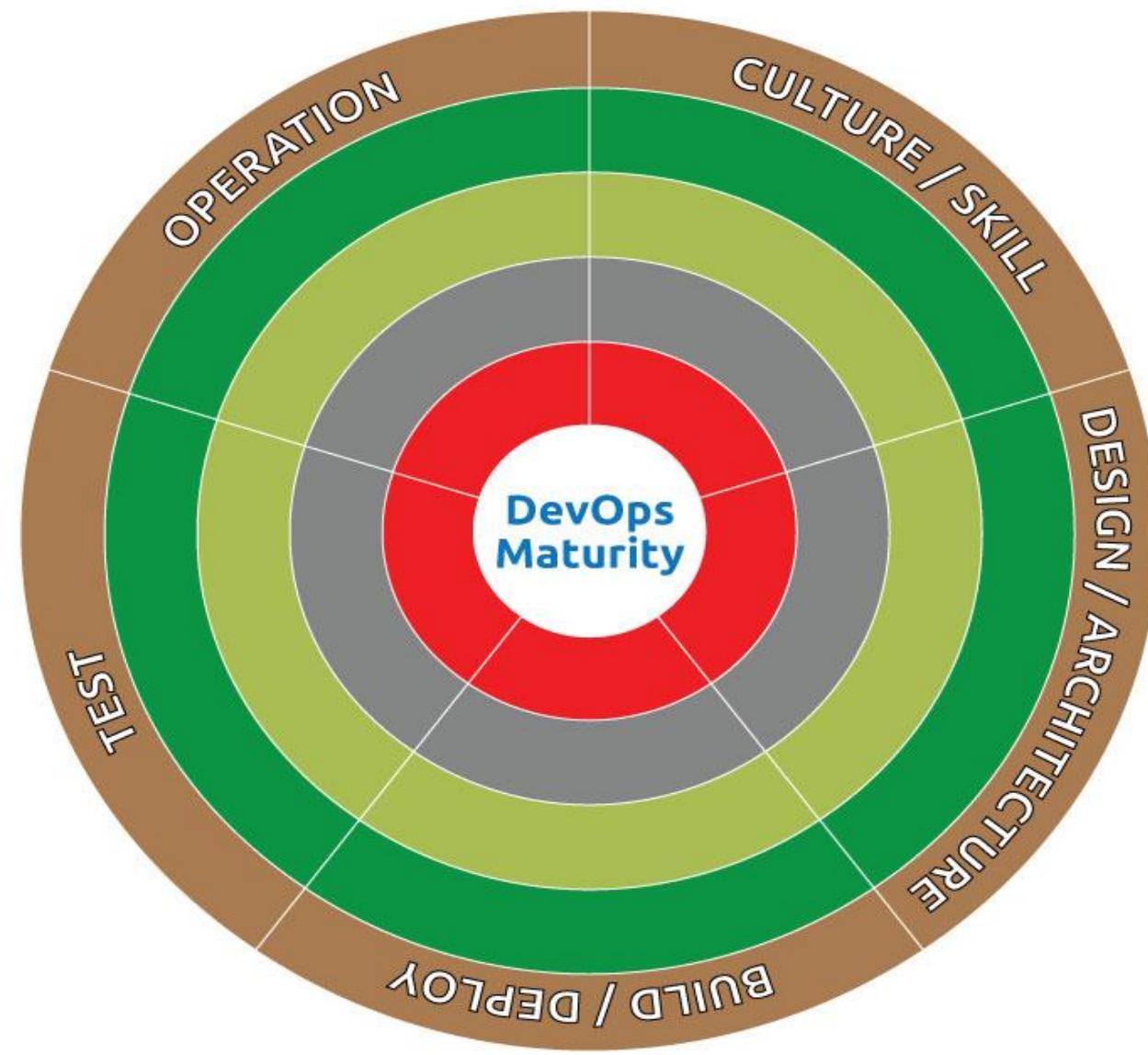
6. What best describe your resilience capability?

1. You mostly troubleshoot from you log analysis or monitoring tool and restore the services manually
2. You have diagnostics scripts available that you run to troubleshoot failure and restore services manually
3. Application support resiliency through resilience architecture (circuit breaker sort of)
4. Automation scripts are present which does regular and preventive maintenance in response to system nature

7. What best describe your operation maturity in adopting advanced technologies?

1. You have automated ticket remediation and workflows that caters your 60% tickets
2. You provision your infrastructure with automated tools that seamlessly does hyper scaling cluster and cloud provisioning
3. You have centralized tool to monitor different product , environments , services and clouds
4. You have continuous security , billing , compliance mechanism in place

Positioning





Key Outcome

F/P/
N

- Technical teams to baseline their performance and understand strengths as well as areas of improvement
- Recruitment team to develop capabilities and recruit related talent
- Management team to target the investment areas
- Operations team to understand the most critical operations

Different Plans :

1. Phase wise Maturity Development
2. Tools & Technology Adoption
3. Skill Development

Why Us?

SOME KEY DIFFERENTIATOR



Expertise

As IT changes so the resource requirement. We understand the minimal requirement and designed only 2 Types of resources 1) Cloud Architect 2) DevOps

Specialized
Resource

Experience

We have done large transformation projects with Toyota, Reliance and addressed the technology and cultural challenges effectively and brought the changes, resulting significant improvement and savings

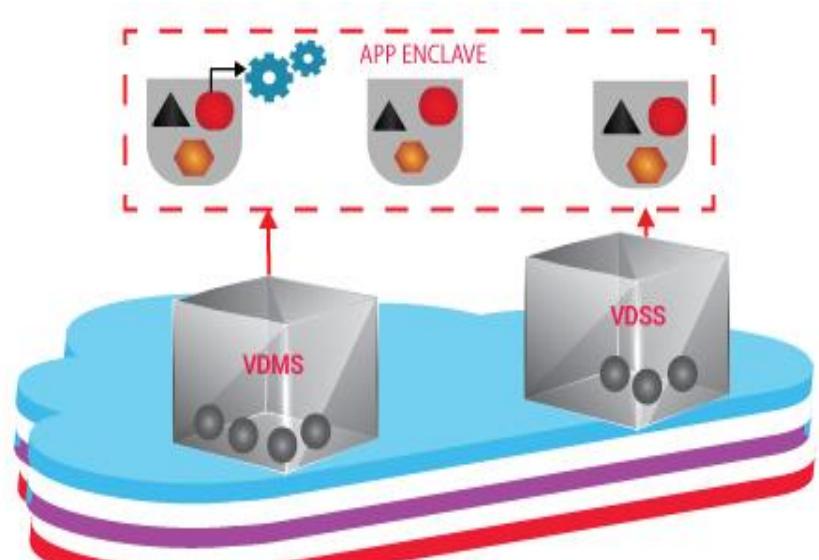
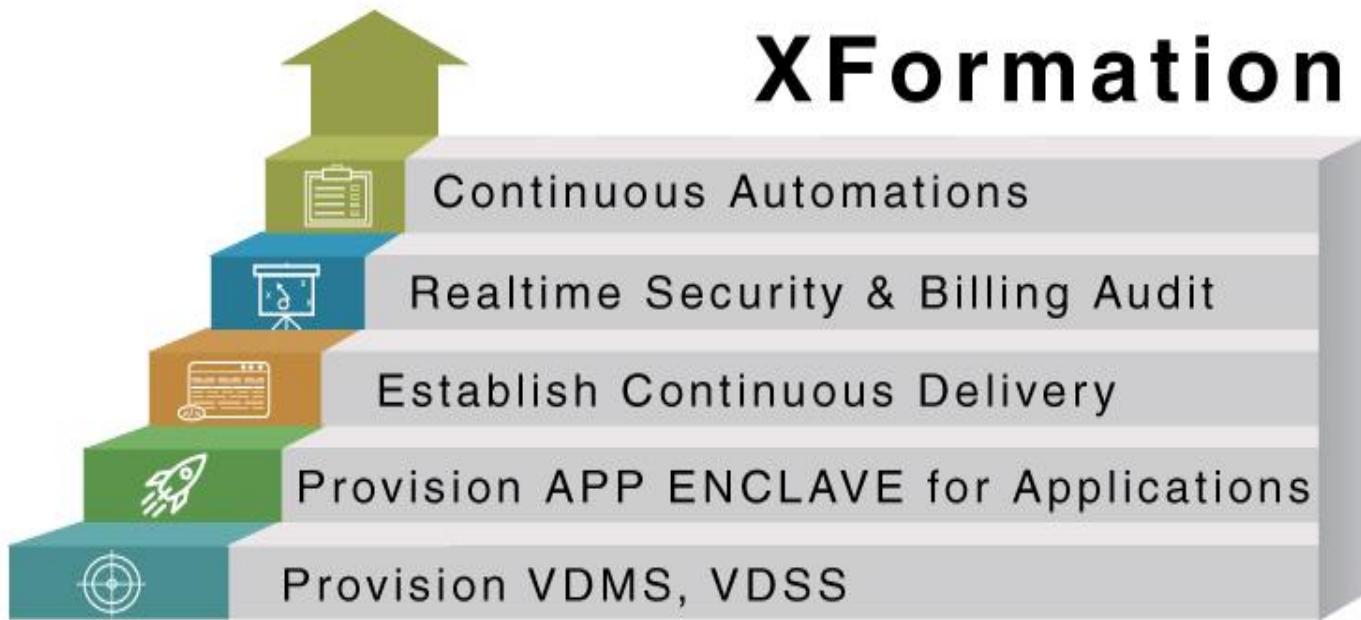
Large
Transformations

Contribution

We are continuously contributing open XFormation platform that is accelerating the transformation @50% Time & Cost.

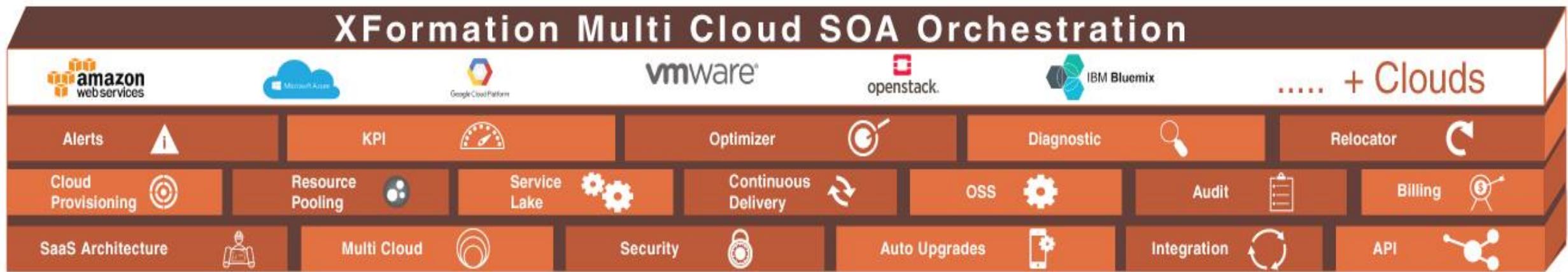
Open Source
Contribution

AWS Transformation @50% Time & Cost



VDMS: AD, LDAP, SSO, OCSP, DHCP, DNS, NTP, Patch, Log Monitoring, ACAS

VDSS: Firewall, IDP/IPS, Network Flow logs



/The clients

SOME OF OUR CLIENTS

HITACHI
Inspire the Next


TRILOGY

