ICS DevOps Deployment Assessment

**Solution Architecture**

|  |  |
| --- | --- |
|  | Document control |

**Change history**

|  |  |  |
| --- | --- | --- |
| Version | Date | Comments |
| 0.1 | 29 – Dec -2016 | Initial version |
|  |  |  |
|  |  |  |

**Related documents**

|  |  |  |
| --- | --- | --- |
| Reference | Title | Version / Date |
|  |  |  |
|  |  |  |

**Document stakeholders**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Stakeholder | Action |  | Stakeholder | Action |  | Stakeholder | Action |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

P= producer, C= contributor, R= reviewer, A= authoriser, I= for information only

The copyright in this document is owned by . All material, concepts and ideas detailed in this document are confidential to . This document shall not be used, disclosed or copied in whole or in part for any purposes without the express permission of .

© 2008. All rights reserved.

[1 Introduction 2](#_Toc470703074)

[2 1. DevOps Solution Plan 3](#_Toc470703075)

[3 1. Executive Summary 4](#_Toc470703076)

[3.1 Background 5](#_Toc470703077)

[3.2 Constraints 6](#_Toc470703078)

[3.3 Existing Best Practices 7](#_Toc470703079)

[3.4 Existing Hindering Practices 7](#_Toc470703080)

[3.5 Key Areas for Improvement 7](#_Toc470703081)

[3.5.1 Current State – Urgent Issues 7](#_Toc470703082)

[4 Roadmap to DevOps Maturity 8](#_Toc470703083)

[5 DevOps progression Framework/Readiness Model 9](#_Toc470703084)

[6 Devops Adaptation from present model 10](#_Toc470703085)

[7 Roadmap 11](#_Toc470703086)

[8 Conclusion 13](#_Toc470703087)

1. Introduction

Businesses who can establish software delivery as their core competency will be able to exploit the emerging technology trends

* Unprecedented demands from cloud, mobile and social, and big data and analytics, instrumented products etc. are causing significant changes in how companies deliver software, requiring new levels of collaboration and innovation.

Business as usual in IT is not going to survive. Traditional software delivery methods evolved over the years across many IT industries now pose a bigger risk than ever, resulting in loss of competitive advantage where business survival can be at stake

1. 1. DevOps Solution Plan

Recommendations for optimizing the DevOps practices and tools in ICS are detailed within this document.

1. 1. Executive Summary

At the request of existing deployment team, this document is developed with the following objectives:

* Understand the current, complete end-to-end software lifecycle from development to post-production
* Elicit the desired objectives for a DevOps implementation
* Document existing Application Lifecycle Management (ALM) topology
* Create a baseline measurement of the current DevOps tools and capability
* Uncover opportunities for improvement
* Identify the most impactful areas to the business
* Document ideal end-state for teams
* Generate and present a roadmap to implement DevOps process using Open source DevOps solution tools: Ansible, python, Jenkins, Nexus and others, including third-party, as appropriate

The DevOps progression Framework model was used as a framework to develop a vision and sustainable approach by which SEVI or any other Vocalink projects can prioritize business investments that fuel business growth. The engagement focused on understanding existing development and best DevOps processes and recommending improvements. Technology, practices, and people/knowledge requirements were then identified to support the process.

* 1. Background

The following are the issues with the current delivery capability.

* Limited human and/or technical resources for testing
* Manual, ad-hoc, or inconsistent deployments
* Resolving production issues is slow and/or highly disruptive
* Deployments are large and take a substantial periods of time to deliver.
* Challenges in testing
* Lack of collaboration between IT and Development organizations

The following business priorities will be achieved thru devops.

* Team Autonomy and Enterprise alignment
* Rigorous management of technical debt
* Focus on flow of customer value
* Hypothesis drive development
* Evidence gathered in production
* Live site culture
* Manage infrastructure as a flexible resource

The following tools and processes (strategy) were articulated at the start of the assessment

* Usage of current DevOps tools
* Challenges that exist in the DevOps tools
* Knowledge of the available practices
  + Configuration management
  + Infrastructure as code
  + Release management
  + Continuous integration
  + Continuous deployment
  + Application performance monitoring
  + Test automation
  + Source Control Management
  1. Constraints

A few constraints arose that would impact a DevOps adoption. Some of these are due to policies and external requirements, while others are due to team or organizational issues that can be addressed in the following ways:

* SEVI needs to have a security model and workflow as the ready code moves through the environment. This requirement however does slow the delivery and testing practices as teams have a challenge accessing the environment.
  1. Existing Best Practices
* Use of Continuous Integration
* Use of SCM and Artefact repository

These practices continue to be employed and are continuously evaluated and improved in order to promote process optimization.

* 1. Existing Hindering Practices

Few issues that are interfering with growth in DevOps maturity. Adopting a new mindset around these issues and prioritizing improvements in these areas will yield significant results. These issues are:

* Lack of Devops tools knowledge.
  1. Key Areas for Improvement
     1. Current State – Urgent Issues

Lack of framework for CI/CD

1. Roadmap to DevOps Maturity

The level of DevOps maturity varies greatly between companies and the type of software being delivered. The goals outlined in the maturity level are not “one size fits all” solutions for companies and teams; rather, they provide a direction for maturing your complete software lifecycle to meet the needs of the business and your customers.

The growth of a mature DevOps practice is not a direct one. As the team becomes more efficient, requirements become refined and testing results become more transparent, and there will be new areas of concern that will emerge.

As DevOps maturity levels move through the next stages (for example, Repeatable to Reliable), some common “growing pains” will begin to emerge. These common “growing pains” will be the best choices for additional assessments and guidance to help assist with the transition for the business, teams, and individuals.

1. DevOps progression Framework/Readiness Model

The model below shows the different practice areas and maturity levels of DevOps at a broad scale. Levels of DevOps maturity may vary across teams, even within a common department or division, and the same organization may have significantly more advanced practices in some categories than others. Enhancing the entire workflow to achieve the best possible DevOps process throughout the enterprise should be the end goal of the team and individual effort.

**Backlog**

**Schedule & Team**

**Technical Debt**

**Flow**

**Evidence**

**Production**

**Foundational**

Releases: ad-hoc

Decisions: brute force

Teams: soiled

Goals: unknown

Limited or no public cloud

**Repeatable**

Releases: periodic

Decisions: anecdotal

Teams: communicative

Goals: inconsistent

Individual teams working with VM’s

**Reliable**

Releases: frequent

Decisions: corroborated

Teams: collaborative

Goals: aligned

Corporate wide with no team autonomy

**Aspirational**

Releases: continuous

Decisions: evidence-based

Teams: unified

Goals: shared

Public Cloud fully embraced

The above model diagram shown to give understanding of different approaches. We at IPS intend to take Reliable approach.

1. Devops Adaptation from present model

Manual

**Applications**

**Data**

**Midware Config**

CI/FT/SIT..

Automation



**Networking**

**Storage**

**Servers**

**Virtualization**

**O/S**

**Middleware**

**Mid Config**

**Data**

**Applications**

**Storage**

**Networking**

**Virtualization**

**Servers**

**O/S**

**Middleware**

**Proposed**

**Infrastructure**

**as a Service**

**Current   
On-Premises**

1. Roadmap

The following iterative roadmap be implemented in order to understand and instill DevOps best practices within the project.

CI/CD Pipeline

The deployment pipeline in essence represents a flow chart of production process and enables to automate and visualize the deployment process from source repo to production. Each step of the build and deployment process can be modeled into the pipeline to provide high resolution visibility into deployment workflow.

Each commit goes through a series of stages that lead all the way to production. With each passing stage, one can get higher confidence with that revision of the code. If something fails, the pipeline stops and it has to be fixed for the build OR revert the commit that caused the failure. If the deploy to production fails, we can rollback by triggering the last successful “Deploy to Production” stage.

With this level of visibility, we can not only determine whether the application is releasable any point, but also identify the bottlenecks in your process and set ourselves in a position to continuously improve the process over time.

The pipeline abstraction can also support complex build and deploy configurations including component dependencies.

Env&App Config

Sourcecode

GIT

Comp1

Build 1

Test Env

Comp22

Build 2

Devlop

Package

Integration

Prod

Pre Prod

Test Env

v

sourcecode

Comp32

Build 3

Test Env

---------------------- BUILD ------------------ -------------- TEST AND RELEASE -----------------------

Above is the value stream map of all the dependencies involved on a deployment workflow down to

production from the Development. It displays the state of each commit, and the dependencies of the application that need to be packaged. It provides the end-to-end visualization to production which indicates the confidence whether the application is releasable at any point. It can be seen that the deployment pipeline can also easily support requirements like integration testing microservice dependencies. Thus, the deployment pipeline is the backbone that provides the right primitives for an effective implementation.

Building pipeline

Implementing the above pipeline benefits the business following way

* Automated QA
* Agile in the form of testing and commit code frequently
* Production like Environments
* Automated deployments
* One click promotion

1. Conclusion

The implementation of the practices outlined in this document be validated during the initial deployment and as projects and teams are brought on board the system. Teams constantly have to adapt and change their processes as the business and technical environment changes around them.