

**PROBLEMAS ASOCIADOS A LA CONVERSIÓN DE APLICACIONES DE
SOFTWARE A SAAS (SOFTWARE AS SERVICES)**

MONOGRAFIA

YULIETH PAOLA VELANDIA MARTÍNEZ

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA**

Enero de 2018

**PROBLEMAS ASOCIADOS A LA CONVERSIÓN DE APLICACIONES DE
SOFTWARE A SAAS (SOFTWARE AS SERVICES)**

MONOGRAFIA

Autor

YULIETH PAOLA VELANDIA MARTINEZ

Trabajo presentado como
requisito para optar al título de
Ingeniera de Sistemas y
Computación

Director: Ph.D Julio César Chavarro Porras

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PEREIRA**

Enero de 2018

Notas de Aceptación:

Presidente del Jurado

Jurado

Pereira, enero de 2018

AGRADECIMIENTOS

Agradezco a dios, a mis padres por su colaboración y dedicación durante el desarrollo de mi carrera, por estar siempre junto a mí brindándome su amor y por permitirme compartir con ellos este logro ya que es el resultado de su apoyo incondicional, gracias por estar presente no solo en esta etapa tan importante de mi vida sino en todo momento ofreciéndome lo mejor, gracias a mi novio porque siempre estuvo presente en cada paso de mi carrera por su permanente apoyo y comprensión, agradezco también a mi director de tesis por guiarme y compartir conmigo sus conocimientos y haberme tenido toda la paciencia del mundo para guiarme durante mi desarrollo educativo y profesional.

TABLA DE CONTENIDO

AGRADECIMIENTOS	4
RESUMEN.....	8
1. INTRODUCCIÓN	9
1.1 Justificación.....	9
1.2 Planteamiento del problema.....	9
1.3 Causas del problema objeto de estudio	10
1.4 Situación problema.....	12
1.5 Definición del problema.....	12
1.6 Objetivo general	12
1.7 Objetivos específicos	12
1.8 Beneficios esperados:.....	12
2. MARCO REFERENCIAL.....	13
2.1 Cloud Computing.	14
2.2 Servicios web	15
2.3 Arquitecturas para crear servicios y recursos web.....	16
2.4 La Computación Orientada A Servicios	18
3. ANÁLISIS DE PROBLEMAS BASADOS EN MML.....	24
3.1 Lista de Involucrados	24
3.2 Descripción de Involucrados.....	25
3.2.1 Personal directivo de la empresa.....	25
3.2.2. Clientes de las aplicaciones.....	26

3.2.3	Representante de los compradores	26
3.2.4	Usuarios finales de las aplicaciones	26
3.3	Personal de desarrollo de las aplicaciones	26
3-3-1	Analistas de Requisitos.....	26
3.3.2	Diseñadores	27
3.3.3	Desarrolladores	27
3.3.4	Implementadores	27
3.3.5	Pruebas de software.....	27
3.4	Capacitadores de las Aplicaciones	28
3.4.1	Capacitadores	28
3.4.2	Educadores virtuales	28
3.4.3	Personal de mesa de servicios	28
3.4	Árbol de problemas	29
3.5	Análisis de árbol de problemas	30
4	Árbol de objetivos	33
5	Análisis del árbol de objetivos	34
6	Análisis de alternativas	36
3.6	Gráfico Estructura analítica del proyecto – EAP	38
4	Análisis de la Estructura analítica del proyecto	39
4.	LA IMPORTANCIA DEL CAMBIO A LA NUBE.....	41
5.	RECOMENDACIONES EN LA MIGRACIÓN A LA NUBE	49
6.	BIBLIOGRAFIA.....	57

TABLA DE ILUSTRACIONES

Ilustración 1: SOA Extendida con las capas de servicio, la funcionalidad y los roles [1]	19
Ilustración 2: Gráfico de involucrados	25
Ilustración 3: Árbol de Problemas	29
Ilustración 4: Árbol de Objetivos	33
Ilustración 5: Estructura Analítica del proyecto	38

RESUMEN

Con la masificación del uso de internet, se han visto transformado los diferentes campos de la vida del hombre. La ingeniería de productos de software no ha sido ajena y hoy se prestan servicios de software sobre internet, los cuales pueden ser consumidos sin tener que haber pagado por un desarrollo o una licencia de producto.

Esta nueva condición en el software como servicio se ha conocido como SaaS por su acrónimo del inglés (Software as a Service). Es necesario reconocer que este modelo de comercialización de los productos de software se ha convertido en una constante tecnológica.

Las casas de software deben responder a esta nueva demanda y para ello necesitan convertir sus productos anteriores a este nuevo modelo. La presente monografía plantea algunos de los retos y problemas asociados a dicha conversión.

1. INTRODUCCIÓN

1.1 Justificación

El mercado de aplicaciones de software ha sido cubierto tradicionalmente, con productos en caja o con productos desarrollados a la medida. Ofrecer aplicaciones utilizando las bondades de *cloud computing* plantea nuevos problemas y retos a las casas de software.

El propósito del trabajo es identificar este conjunto de retos y problemas para una organización que tiene productos de software en las modalidades tradicionales y que quiere trasladarlos a SaaS, proponer mecanismos de gestión de los mismos y analizar estas propuestas de gestión a la luz del modelo de madurez propuesto en CMMI.

Para el desarrollo del trabajo, se plantea estudiar el caso de la empresa OSR S.A.S, quienes están en el proceso de conversión de algunas de sus aplicaciones, y adicionalmente se encuentran adelantando el proceso de certificación en CMMI categoría III en la constelación de servicios.

Con relación a la autorización de acceso a la información de la empresa OSR SAS, existe el permiso para realizar el trabajo, entre otras razones, debido a que actualmente una de las integrantes del grupo de proyecto de grado labora para esta empresa.

1.2 Planteamiento del problema

Las aplicaciones de software corren en plataformas tradicionales que se despliegan en ambientes alojados por el cliente y gestionados por sus propios empleados, estos ambientes son conocidos como On-Premises. El modelo opuesto a On-premises es el Software como un servicio SaaS (por su sigla en inglés Software as a service), también conocido como servicio en la nube o Cloud Computing.

La demanda de aplicaciones de software SaaS es creciente [1], y la tendencia es cada vez más fuerte no solo con el servicio de software en Cloud Computing, sino con los servicios de PaaS, IaaS, Colocation y Bare Metal.

La conversión de las aplicaciones a SaaS, es un proceso costoso, complejo y poco claro. Entre otras razones por:

1. Las tecnologías cambian constantemente generando la sensación de obsolescencia.
2. Se carece de la madurez organizacional adecuada que facilite la administración de proyectos y procesos.
3. La formación de los profesionales en ingeniería de Sistemas, no satisface los perfiles ocupacionales que se demanda por las casas de software
4. La ingeniería de software no ofrece metodologías o herramientas que guíen este proceso.

Todo lo anterior genera que un proyecto de conversión esté marcado por:

1. Atraso en los proyectos de software.
2. Problemas en la calidad del producto generado.
3. Gestión compleja del personal que participa en el proyecto.
4. Complejidad en las actividades relacionadas con la Gestión de la Configuración y con el Soporte.

Gestionar este conjunto de retos y problemas hace necesario el ajuste de metodologías que puedan guiar el proceso de desarrollo e implementación del nuevo servicio de software. Así como la administración de aquellos factores que solo podrán ser minimizados.

1.3 Causas del problema objeto de estudio

Un análisis de las diferentes causas que generan esta situación es la siguiente:

1. Las tecnologías cambian constantemente generando la sensación de obsolescencia. La tecnología en los últimos años ha evolucionado de manera acelerada y debemos estar siempre preparados para esos cambios constantes en las plataformas tecnológicas que se usan en el desarrollo de las aplicaciones, tanto las casas de desarrollo de software como los usuarios deben estar actualizados en las tecnologías actuales para que sus productos o servicios mejoren el rendimiento en sus empresas.

2. Se carece de madurez organizacional que facilite la gestión adecuada de proyectos y procesos. Se hace necesario implementar procedimientos gerenciales y administrativos que faciliten la ejecución de todas las etapas que tienen los proyectos utilizando el modelo de madurez en la administración de los proyectos.

3. Formación pre graduada no satisface los perfiles demandados en las casas de software La falta de personal capacitado en las nuevas tecnologías o competente para los perfiles que busca la empresa, genera la necesidad de que la empresa realice entrenamiento organizacional, esto lleva a que la empresa tenga unos costos altos en capacitación y entrenamiento.

4. La Ingeniería de Software no ofrece herramientas teóricas o metodológicas que faciliten y guíen este proceso

La ingeniería de software está orientada a cumplir unas fases de desarrollo e implementación de productos de software en general. La conversión de productos

existentes a software como un servicio rebasa los límites de la reingeniería de software e incluso de las técnicas conocidas como refactoring.

Se carece de una metodología que permita especificar la estandarización de los productos de software, haciendo que los procesos de estandarización se realicen con alto grado de informalidad.

Estas causas tienen efectos en la calidad y funcionamiento organizacional:

- 1. Atraso en los proyectos de reemplazo de software.** El atraso en los proyectos se debe a que no se define el alcance, se realiza una inadecuada toma de requisitos, los clientes no dan la información completa o cambian de idea de lo que desea en el proyecto durante la ejecución, las herramientas de desarrollo no están actualizadas acorde a las nuevas tecnologías que requieren implementar los usuarios, por esto se da el envejecimiento de los productos, otra causa en el atraso de los proyectos es el alto costo en la actualización de las plataformas tecnológicas de los productos.
- 2. Problemas en la calidad del producto** La Ingeniería del Software persigue como objetivo esencial el proporcionar las herramientas fundamentales para la producción de software de alta calidad, pero existen varias causas que originan los problemas en la calidad del producto como por ejemplo la mala planeación e implementación de los procesos, no se realiza el adecuado análisis en las fases de desarrollo, personal sin experiencia, utilización de plataformas incorrectas, esto causa Insatisfacción de clientes por entregas tardías, un desfase constante de agendas, recursos y costos.
- 3. Gestión compleja del personal** El departamento de recursos humanos es el encargado de suplir las necesidades del personal en la empresa de acuerdo al perfil requerido, pero existe una alta rotación del personal, ya que no son competentes para los cargos por falta de experiencia en las plataformas utilizadas en la empresa, no están preparados para el cambio, también al poco entrenamiento organizacional.
- 4. Complejidad del soporte y la gestión de la configuración.** La Gestión de Configuración identifica y define los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

La complejidad en el soporte de los aplicativos se da porque no hay un manejo adecuado de las versiones, actualizaciones de los productos, no hay documentación del código, o un manual donde diga que cambios se han realizado, las versiones se crean porque hay clientes que piden implementar otros módulos en el aplicativo, para corregir errores, o simplemente actualizar la aplicación.

No hay un control sistemático de los cambios en la Configuración y el mantenimiento de la integridad y trazabilidad de los proyectos, los cambios se reflejan en el modelo de ejecución del proyecto, pero no están documentados

Estos efectos deben ser mitigados para poder garantizar la sostenibilidad a largo plazo de la compañía

1.4 Situación problema

Se ha identificado que el proceso de conversión de aplicaciones a SaaS es complejo, costoso y poco claro, de la misma forma se ha señalado que existen múltiples factores que deben ser administrados y que dependen en gran medida del grado de madurez de la organización. Podemos por tanto concluir que el problema debe ser estudiado y administrado para cada organización que lo debe estudiar desde sus peculiaridades internas. Sin embargo, las características generales del problema pueden ser planteadas en la siguiente pregunta:

1.5 Definición del problema

¿Es posible identificar los problemas asociados a la conversión de aplicaciones a SaaS?

1.6 Objetivo general

Identificar el conjunto de retos y problemas asociados a la conversión de productos de software On-Premises a SaaS.

1.7 Objetivos específicos

1. Identificar los principales retos del proceso de conversión de productos On-Premises a SaaS
2. Proponer un conjunto de actividades, artefactos y mecanismos que faciliten la administración, seguimiento y control de los proyectos de conversión de software a SaaS.
3. Identificar los mecanismos administrativos de gestión de los nuevos productos de software como SaaS.

1.8 Beneficios esperados:

Entre los principales fines esperados se encuentran:

- 1. Mejorar la entrega de los proyectos a tiempo.** Para mejorar la entrega de los productos tiempo se actualiza las herramientas, plataformas de desarrollo que ayuden al mejoramiento de los productos, mejorar el uso herramientas virtuales para disminuir los costos, disminuir los costos de actualización en las plataformas tecnológicas de los productos, definir bien el alcance del proyecto para cumplir con los tiempos estimados en el contrato para los entregables de los

productos, mejorar el tiempo de obsolescencia de las herramientas tecnológicas según el entorno tecnológico en las nuevas plataformas a fines con los productos desarrollados.

- 2. Mejorar los atributos de calidad del producto.** Para mejorar los atributos de los productos de software se revisa los alcances que tiene los proyectos, se define cronogramas de actividades, se realiza un trabajo colaborativo entre el grupo de desarrolladores, se realiza pruebas funcionales y no funcionales para garantizar la calidad de los productos, se disminuye costos en el presupuesto del proyecto y se mejora las entregas de los productos a los clientes en los tiempos estipulados en el contrato cumpliendo con el cronograma en los tiempos de agenda.
- 3. Políticas adecuadas para la gestión de personal.** Implementar un modelo de desarrollo de personal teniendo en cuenta el reglamento y manual de procesos y procedimientos, el reglamento interno de trabajo, manuales de cargos y funciones de los empleados, mejorar el entrenamiento organizacional, construir un plan de incentivos ,reducir el impacto financiero y administrativo en el cambio de personal, contratar personal capacitado y preparado para el cambio, con experiencia en las nuevas tecnologías, también garantizar el crecimiento de los empleados mejorando las condiciones y habilidades, generar un clima ocupacional y ambiente laboral óptimo para el desarrollo personal de los empleados, promover las facilidades de estudio formal en maestrías y certificaciones especializadas de productos y tecnologías emergentes.
- 4. Alcanzar adecuados niveles de satisfacción del cliente en el servicio de soporte prestado** Implementar una metodología que proporcione el proceso adecuado a productos SaaS, con un método sistemático de desarrollo para controlar los procesos en la construcción del software, para gestionar este conjunto de retos y problemas hace necesario el ajuste de metodologías que puedan guiar el proceso de desarrollo e implementación del nuevo servicio de software, adicionalmente mejorar la competitividad de la empresa con productos renovados tecnológicamente proponiendo un modelo gestionado y definido para la conversión de aplicaciones On-premises a SaaS.

2. MARCO REFERENCIAL

El marco referencial, incluye elementos de ingeniería de software genéricos. Los elementos específicos son desarrollados en los capítulos 3 al 5.

2.1 Cloud Computing.

Es un paradigma de programación que permite el acceso a redes, servidores, almacenamiento, aplicaciones y servicios a través de la plataforma de internet con un esfuerzo mínimo de los usuarios¹.

El concepto de Cloud Computing se basa en la necesidad de los usuarios, de acceder a un servicio a través de internet que le permita disponer en todo tiempo y lugar de sus archivos y datos, como así también de las aplicaciones más utilizadas².

¿Cómo son las arquitecturas de Cloud Computing?

Los modelos de servicios de Cloud Computing son:

- **Software como servicio (SaaS)**³: Proporcionar el mantenimiento, soporte y operación del software (sitios web, aplicaciones de oficina-gmail, web integrable a otras aplicaciones) que utiliza el cliente.
- **Plataforma como servicio (PaaS)**⁴: Soluciones que permite construir y poner en marcha una aplicación (plataformas de desarrollo, bases de datos, cola de mensajes, servidores de aplicaciones) y servicio web completamente.
- **Infraestructura como servicio (IaaS)**⁵: Proporcionar al cliente infraestructura de computo (hosting, mantenimiento de redes, capacidad de computo. EJ: almacenamiento, administración de sistemas, procesamiento, etc.) como un servicio de manera virtual.

¿Qué tecnologías se usan en Cloud Computing?⁶

¹ La computación de nube(cloud computing), un modelo variable en la pequeña empresa-2011-Martin Alfonso Nieto Prada, Ingeniero de sistemas

² Cloud Computing: el objetivo de la nube-Graciela Marker-Informática hoy, 2010-<http://www.informatica-hoy.com.ar/la-nube/Cloud-Computing-el-objetivo-de-la-nube.php>

³ Las capas de servicios arquitectónicos del computo cloud-<http://www.itnews.ec/marco/000037.aspx>

⁴ Ibíd.

⁵ Ibíd.

⁶ La computación en la nube (Cloud Computing), un modelo viable en la pequeña empresa-Martin Alfonso Nieto-2011

- **Virtualización:** crea una versión virtual (abstracción de los recursos de una computadora) de un dispositivo o recurso, como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo, donde se divide el recurso en uno o más entornos de ejecución
- **Multitenancy:** multi-instancia donde cada cliente trabaja con una instancia de la aplicación virtual personalizada.
- **Conceptos de recursos y servicios web**

2.2 Servicios web⁷

Sistema de software diseñado para soportar interacciones interoperable máquina a máquina sobre una red. Se puede definir de manera más sencilla como un conjunto de tecnologías estándares de software para el intercambio de datos entre aplicaciones tales como SOAP, WSDL y UDDI. Estos pueden ser desarrollados en una gran variedad de lenguajes para ser implementados sobre muchos tipos de redes de computadores.

¿Cómo se construye un servicio web?

A continuación se describen los pasos empleados en .Net Framework para construir un servicio web⁸:

1. Abrir un editor de texto para crear un archivo, el cual se deberá guardar con la extensión .asmx.
2. Luego para que .Net Framework sepa que el archivo es un servicio web, se le indica mediante la directiva [ASP.NET@webservice](#), de la siguiente forma: `<%@webservice Language="" Class="" nombreClass"%>`.
3. En .Net Framework, definimos una clase que contendrá el servicio web a crear.

⁷ Estado del arte: Servicios web-
http://bpelproject.googlecode.com/svn/trunk/_doc/bibliograf%C3%Ada/Web%20service/articulo2.pdf

⁸ Crear un servicio web (con un editor de texto)-www.elguille.info/NET/ASPNET/crearservicioweb.aspx

4. Empezaremos utilizando la directiva `using System.web.service`, que contiene las clases que nos permiten crear servicios web y clientes de servicios web.
5. Luego de importar todos los nombres que necesitemos declaramos la clase con su respectivo código.
6. Publicar el código en sitio web.

¿Cómo se usan los servicios web?⁹

Cuando el *proxy* recibe del cliente una solicitud de llamada al servicio web, el *proxy* la traduce a un mensaje SOAP, que envía al servidor; el cual lo ejecuta, y devuelve un mensaje SOAP al *proxy*; éste entonces, traduce el mensaje a objetos Java, .NET, etc. y entrega el resultado al cliente que efectuó la petición.

¿Qué son recursos web?

Por lo general, un recurso es algo que puede ser almacenada en un ordenador y representada como una secuencia de bits: un documento, una fila de una base de datos, o el resultado de ejecutar un algoritmo¹⁰.

Herramientas o aplicaciones de las que podemos disponer en internet. Estos recursos son usados por toda persona que lo requiera.

2.3 Arquitecturas para crear servicios y recursos web

Arquitectura orientada a servicios (SOA)¹¹: Modelo de diseño que encapsula la lógica de aplicaciones independientes dentro de servicios que interactúan a través de un protocolo común de comunicaciones. Cuando migramos la arquitectura de una aplicación a una SOA, estamos aplicando los principios de diseño de servicios web y las tecnologías que los acompañan como partes centrales del entorno tecnológico.

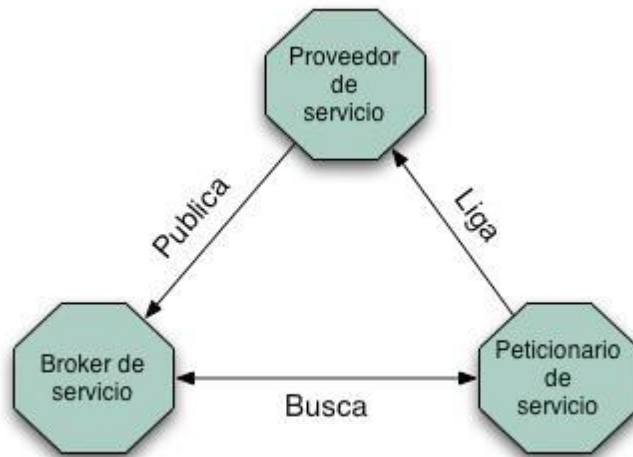
⁹ Introducción a los servicios web-<http://www.esi.uclm.es/www/mpolo/serviciosweb.pdf>

¹⁰ RESTful Web Services-Leonard Richardson E San Ruby Foreword by David Heinemeier Hasson

¹¹ La arquitectura orientada a servicios (AOS)-Microsoft corporation diciembre 2006-www.microsoft.com/soa

Comúnmente la arquitectura SOA es implementada mediante servicios web, tecnología basada en estándares e independiente de la plataforma, descompone aplicaciones monolíticas en un conjunto de servicios y los implementa de forma modular.

La SOA basada en servicio web XML, promueve mecanismos de descubrimiento de los servicios, a través de un bróker de servicios o agente descubridor.



Relación entre Peticionario, Proveedor y Bróker de servicio

El bróker de servicio devuelve un documento que permite al cliente localizar el proveedor y unirse a él. Todos los servicios están registrados en el bróker con la finalidad de brindar a los clientes la posibilidad de acceder a múltiples servicios con proveedores diferentes.

Beneficios de SOA

- **Permite el desarrollo de aplicaciones dinámicas.**
- **Mejorar la toma de decisiones:** Al integrar el acceso a los servicios e información de negocio dentro de un conjunto de aplicaciones dinámicas, los directivos cuentan con más información y de mejor calidad, permitiéndoles reaccionar de manera ágil y rápida cuando surgen problemas o cambios.
- **Mejora la productividad de los empleados:** un acceso óptimo a los servicios y a la información, hace que los empleados dediquen sus energías a los procesos que generan valor añadido y a las actividades de colaboración.

- **Potenciar las relaciones con clientes y proveedores:** los clientes y proveedores externos disponen de acceso a aplicaciones y servicios de negocio dinámicos, permitiendo una mejor colaboración y la satisfacción de clientes y proveedores.

Arquitectura orientada a recursos (ROA): Es un estilo de arquitectura de software para sistemas hipermedias distribuidos, tales como la web, refiriéndose específicamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen como los servicios son definidos y direccionados¹².

Aunque ROA no es un estándar, si está basado en estándares:

- HTTP
- URL
- Representación de los recursos: XML/HTML/GIF/JPEG/...
- Tipos MIME: text/xml, text/html,...

Los objetivos de ROA son¹³:

- **Escalabilidad:** crecimiento sin disminuir el rendimiento.
- **Generalidad de interfaces:** interacción del cliente con cualquier servidor http, sin realizar configuraciones especiales.
- **Puesta en funcionamiento independiente:** extensibilidad de los servidores antiguos.
- **Compatibilidad con componentes intermedios:** reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas

2.4 La Computación Orientada A Servicios

Las nuevas propuestas tecnológicas para la web que han surgido en los últimos años, tales como XML, Servicios Web, automatización de procesos de negocios, B2B, etc., han beneficiado altamente el surgimiento de un nuevo paradigma para el desarrollo de aplicaciones, conocido como Computación Orientada a Servicios (Service Oriented Computing, SOC), que propone la utilización de servicios como elementos básicos para

¹² REST Vs Web Services- Rafael Navarro Marset. ELP-DSIC-UPV Modelado, Diseño e Implementación de Servicios Web 2006-07

¹³ Ibíd

la construcción de aplicaciones. La importancia de SOC radica, entre otras cosas, en su capacidad para dar soporte a la manera en que las grandes empresas están conduciendo actualmente sus negocios, basados en redes de negocios en las que cada uno de los participantes provee a los demás servicios especializados.

En este nuevo paradigma, los Web Services o Servicios Web juegan un papel importante y proveen un marco para la aplicación basada en estándares de este paradigma. Estos servicios proporcionan un marco de trabajo sistemático y extensible, basado en XML y construido en base a protocolos Web existente, que permite la interacción de aplicaciones a través de la Web. Dichos servicios definen mecanismos estandarizados para describir, publicar, localizar e interactuar con aplicaciones en línea.

La Computación Orientada a Servicios abre nuevos mercados en el mundo del software, tanto para los proveedores a pequeña escala como para las grandes organizaciones que proveen servicios de carácter general. Ésta toma los servicios como elementos fundamentales para el desarrollo de aplicaciones. Y se basa en principios que se reflejan en las capas que establece la arquitectura orientada a servicios (SOA).

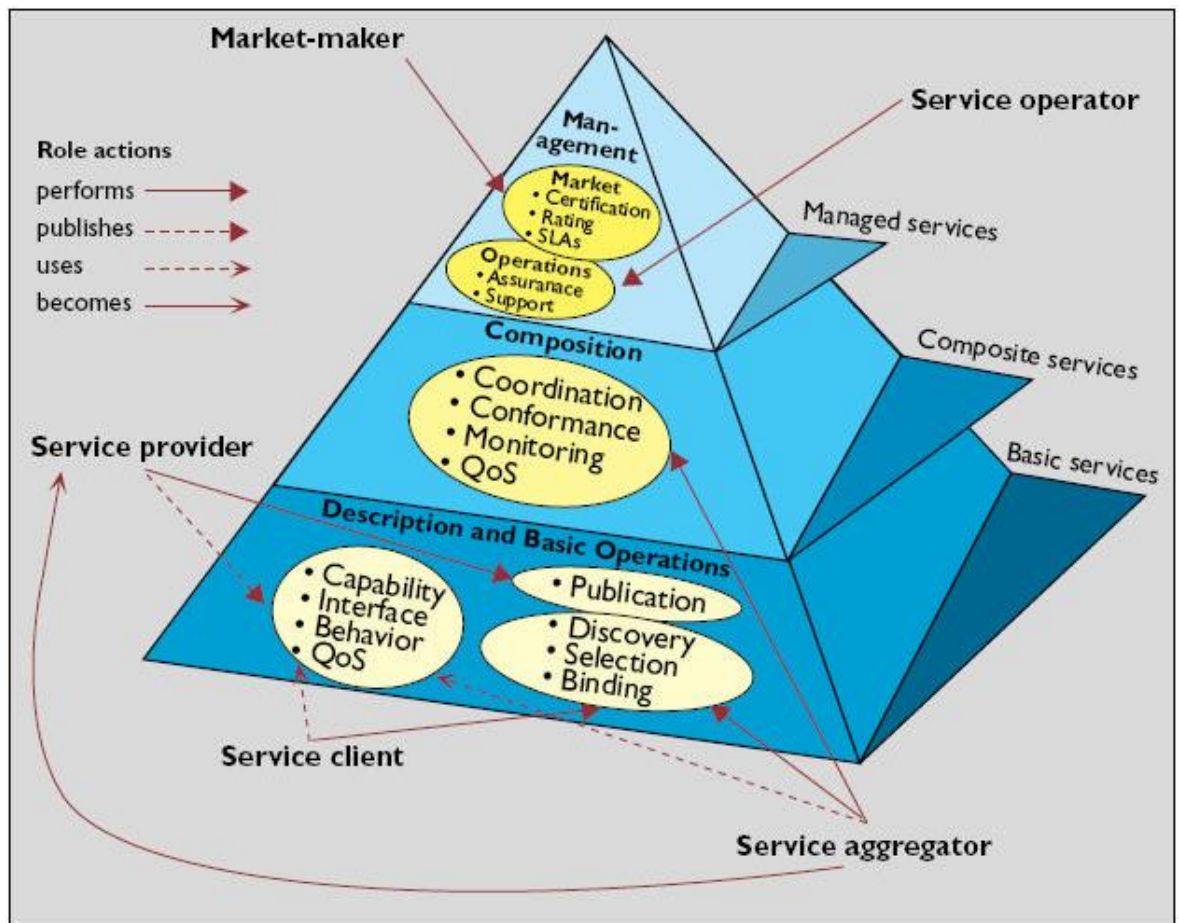


Ilustración 1: SOA Extendida con las capas de servicio, la funcionalidad y los roles [1]

En la figura anterior, se hace una descripción de los servicios básicos de la arquitectura SOA, que como dijimos anteriormente son los principios en los que se basa SOC, también encontramos las descripciones y operaciones básicas (publicación, descubrimiento, selección y vinculación) que producen o utilizan dichas descripciones. Las capas superiores de la pirámide, brindan el soporte adicional requerido para la composición y gestión de servicios. La composición de servicios, es uno de los principales conceptos de la Computación Orientada a Servicios, ya que se convierte en el núcleo del proceso de desarrollo de aplicaciones, abarcando los roles y la funcionalidad que nos permite generar un servicio mediante la unión de otros [1].

Según Curbera y otros [2], cualquier arquitectura que implemente la filosofía SOC, deberá proveer definiciones basadas en estándares sobre el protocolo de comunicación que utilice para implementar la interoperabilidad, mecanismos para la descripción de los servicios, para el descubrimiento y la composición, así como un conjunto básico de protocolos para la calidad del servicio (QoS).

En SOC los servicios son autónomos, independientes de las plataformas, las entidades en estos pueden ser descritas, publicadas, descubiertas y ligeramente acopladas de forma novedosa, también, estos llevan a cabo funciones que van desde simples peticiones para contestar llamadas, a la ejecución de procesos de negocio que requieren sofisticadas relaciones peer-to-peer entre múltiples capas de los usuarios de los servicios y proveedores. Este enfoque es independiente de lenguajes de programación concretos o sistemas operativos. Los servicios Web, son actualmente la tecnología más prometedora basada en SOC, usan el Internet como medio de comunicación y estándares abiertos basados en Internet, incluyendo el *Simple Object Access Protocol (SOAP)* para transmitir datos, el *Web Service Description Languages (WSDL)* para definir servicios, y el *Business Process Execution Language for Web Service (BPEL4WS)* para la orquestación de servicios.

2.4.1 Concepto Computación en la nube “Cloud Computing”

Con el surgimiento en los últimos años de la disposición en todas partes de los recursos informáticos, el Cloud Computing se ha establecido como un tema importante de investigación. Algunos autores se refieren a esta como un nuevo paradigma de tecnologías emergentes, que de forma flexible ofrece recursos y servicios a través de Internet.

La computación en la nube desde la perspectiva tecnológica, es vista como uno de los más grandes avances en la historia de la computación, que se desarrollo desde las grandes máquinas de tabulación y arquitecturas mainframe centralizadas, a través de recursos de cálculo distribuido y descentralizado de arquitecturas cliente-servidor, hasta

los pequeños dispositivos personales que se pueden encontrar y utilizar casi en todas partes.

En la actualidad, el concepto de la computación en la nube para muchas personas sigue siendo poco claro e impreciso. Debido esto al uso indirecto que se hace de la misma, ya que aunque muchos utilizan las diferentes ventajas y funcionalidades que ofrece la computación en la nube, son pocos los que saben que se trata de ella. A continuación, se describen algunos conceptos de la computación en la nube que permitan llegar a una idea clara y concreta, acerca de lo que es realmente la computación en la nube “Cloud Computing”.

Según Graciela Marker¹⁴ *“el concepto de Cloud Computing se basa en la necesidad de los usuarios, de acceder a un servicio a través de internet que le permita disponer en todo tiempo y lugar de sus archivos y datos, como así también de las aplicaciones más utilizadas”*.

Por otro lado, Ambrust y otros [3] perciben la computación en la nube desde los siguientes aspectos: (1) la ilusión de la infinita capacidad de computación disponible bajo demanda, (2) la eliminación del compromiso inicial de recursos de lado del usuario de la nube, y (3) los precios de consumo con destino a los recursos informáticos a corto plazo.

Por último, los autores [4] realizan una recopilación de diversos conceptos, definen la computación en la nube “Cloud Computing”, como un modelo de implementación de TI basado en la Virtualización, donde los recursos en términos de infraestructura, aplicaciones y datos se despliegan a través de Internet como un servicio distribuido por uno o varios proveedores de servicios. Estos servicios son escalables en la demanda y pueden tener un precio sobre una base de pago por uso.

Este nuevo modelo llamado Cloud Computing, tiene como objetivo principal proporcionar interoperabilidad completa desde cualquier tipo de dispositivo que se utilice, ya sean terminales de telefonía móvil, PCs con cualquier sistema operativo instalado o inclusive PDAs, entre muchos otros, brindando la posibilidad de interactuar con la información y los programas desde cualquier lugar del mundo, mediante la simple utilización del Internet.

¹⁴ Cloud Computing: el objetivo de la nube-2010-<http://www.informatica-hoy.com.ar/la-nube/Cloud-Computing-el-objetivo-de-la-nube.php>

El modelo de computación en la nube promueve la disponibilidad, y consta de cinco características claves [5]: Auto-servicio por demanda, Acceso ubicuo a la red, Fondo común de recursos, Rápida elasticidad y servicio medido.

- **Auto-servicio por demanda:** El usuario o cliente, puede solicitar servicios a través de Internet, en donde pagará únicamente por el tiempo que hace uso del servicio.
- **Acceso ubicuo a la red:** Los servicios se encuentran en la nube, y a ellos se puede acceder a través de cualquier dispositivo que cuente con acceso a Internet (teléfonos móviles, portátiles y PDAs).
- **Fondo común de recursos:** Los servicios pueden ser utilizados por múltiples usuario, desde cualquier parte del mundo bajo un modelo multi-arrendatario. El cliente generalmente no tiene ningún control o conocimiento de la ubicación exacta de los recursos; ejemplos de recursos incluyen el almacenamiento, procesamiento, memoria, ancho de banda de la red y las máquinas virtuales.
- **Rápida elasticidad:** La calidad o cantidad o calidad de los servicios ofrecidos en la nube, puede aumentar o disminuir rápidamente dependiendo de las necesidades cambiantes de los usuarios.
- **Servicios medidos:** Los recursos consumidos por el usuario y que son facturables, deben ser medidos, con fines de tarificación y control

2.4.2 Componentes de Cloud Computing

La Computación en la Nube “Cloud Computing”, cuenta con 11 grandes categorías o tipos de tecnología: almacenamiento como un servicio (Storage-as-a-service), Bases de datos como un servicio (Database-as-a-service), Información como un servicio (Information-as-a-service), Procesos como un servicio (Process-as-a-service), Aplicación como un servicio (Application-as-a-service), Plataforma como un servicio (Platform-as-a-service), Integración como un servicio (Integration-as-a-service), Seguridad como un servicio (Security-as-a-service), Administración/Gobierno como un servicio (Management/governance-as-a-service), Pruebas como un servicio (Testing-as-a-service), Infraestructura como un servicio (Infrastructure-as-a-service). A pesar de existir todas estas tecnologías o arquitecturas en la Cloud Computing, frecuentemente se emplean solo tres de ellas, que son: la de Infraestructura, Aplicación y Plataformas como un servicio.

- **Aplicación como un servicio (Application-as-a-service):** En ocasiones se emplea el nombre Software como un servicio (Software-as-a-service), para referirse a esta arquitectura. Hace referencia a cualquier aplicación que se entrega a través de la plataforma web a un usuario final, normalmente aprovechando la aplicación a través de un navegador. Como Google Docs, Gmail y Google Calendar.
- **Plataforma como un servicio (Platform-as-a-service):** Incluye aplicaciones de desarrollo, desarrollo de interfaces, desarrollo de bases de datos, almacenamiento, pruebas, etc. Entregada a los suscriptores a través de una plataforma alojada remotamente. Se basa en el tradicional modelo de tiempo-compartido, y proporciona la capacidad de crear aplicaciones de clase empresarial para uso a nivel local o por demanda, con precios bajos de suscripción o gratuito.
- **Infraestructura como un servicio (Infrastructure-as-a-service):** Brinda la posibilidad de acceder de forma remota a recursos computacionales. El enfoque de este frente a los demás de Cloud Computing, es que en lugar de usar una interfaz y un servicio medido, se tiene acceso a toda la maquina, es decir, es menos envasado.

Este nuevo paradigma para el desarrollo de aplicaciones resume:

- Servicios web.
- Arquitectura Orientada a Servicios , y
- Arquitectura Orientas a Recursos.

3. ANÁLISIS DE PROBLEMAS BASADOS EN MML

Este capítulo presenta el análisis de los problemas que pueden ser presentados y que deben ser administrados en una organización que aborda un proceso de conversión de un producto que se ha construido para dar solución soportado en una arquitectura On Premises y que desean trasladar a Software como un Servicio – SaaS.

3.1 Lista de Involucrados

Personal directivo de la empresa S

Junta directiva

Clientes de las aplicaciones

Representantes de los compradores,

Usuarios finales de las aplicaciones

Personal de desarrollo

1. Analistas de requisitos

2. Diseñadores

3. Implementadores

4. Pruebas de software

Capacitadores de las aplicaciones

Capacitadores

Educadores virtuales

Personal de mesa de servicios

Gráfico de involucrados

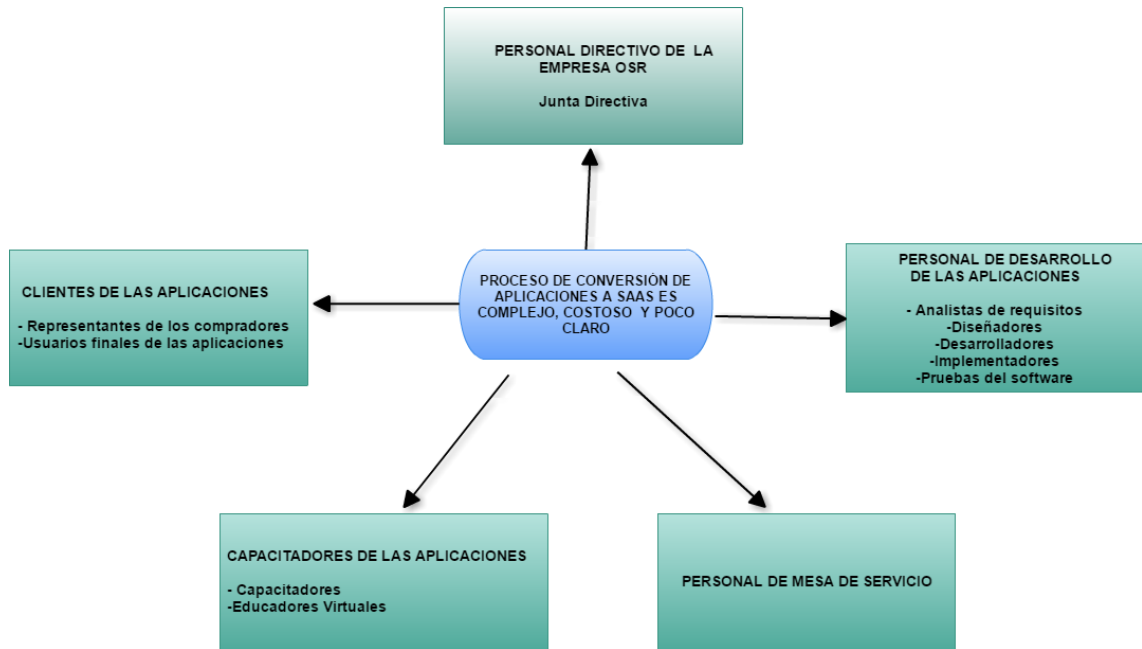


Ilustración 2: Gráfico de involucrados

Fuente: Elaboración propia

3.2 Descripción de Involucrados

3.2.1 Personal directivo de la empresa

El personal directivo de la empresa está conformado por la junta directiva y el equipo administrativo.

La junta directiva tiene entre sus funciones el direccionamiento estratégico y los proyectos de mayor envergadura de la organización.

Equipo administrativo se encuentra conformado por la gerencia, y los directores de las áreas de operaciones o ingeniería, calidad o innovación, comercial y administrativo.

3.2.2. Clientes de las aplicaciones

Son las personas interesadas en adquirir los productos/servicios de software dependiendo de la necesidad que tenga, el cliente puede adquirir un desarrollo de software a la medida, o un desarrollo de software estándar el cual *resuelve múltiples necesidades*.

3.2.3 Representante de los compradores

Es el que representa el interés de la organización en los productos/servicios adquiridos

3.2.4 Usuarios finales de las aplicaciones

Identifica las personas que van a interactuar con la aplicación y permiten conocer los requisitos de la aplicación.

3.3 Personal de desarrollo de las aplicaciones

Son todos los que intervienen en el desarrollo del proyecto, se tiene en cuenta el ciclo de vida del software donde se valida el desarrollo de la aplicación es decir, se garantiza que el software cumpla los requisitos, se asegura de que los métodos utilizados son apropiados, permite que los errores se detecten lo antes posible y que los desarrolladores los corrijan para así cumplir con un software de calidad en los plazos estipulados en la implementación y los costos asociados.

3-3-1 Analistas de Requisitos

Toma en cuenta las necesidades de los usuarios involucrados, manteniendo los estándares para el análisis y diseño de los sistemas de información. Son los encargados de recopilar y validar la información con los clientes por medio del levantamiento de la información con entrevistas, encuestas, etc., de acuerdo a ello se establecen los requisitos que debe cumplir la aplicación.

3.3.2 Diseñadores

Los diseñadores efectúan labores de análisis, modelamiento, diseño de las aplicaciones de software, implementa los estudios y cambios necesarios en la actualización del sistema corrigiendo deficiencias e implementando mejoras solicitadas. Después del análisis proceden a diseñar los modelos de las aplicaciones por ejemplo en [mockups](#), prototipos, modelos, etc, todo esto se valida con el cliente antes de pasar a la codificación de la aplicación.

3.3.3 Desarrolladores

El personal de desarrollo es el encargado de desarrollar la aplicación teniendo en cuenta las especificaciones del proyecto, cumpliendo con los estándares, también con niveles de seguridad, calidad y desempeño requerido.

3.3.4 Implementadores

Efectúan las pruebas de desarrollador para la integración en subsistemas más grandes, de acuerdo con los estándares adoptados de proyecto. También se encargan de ir donde el cliente e instalar los aplicativos necesarios para el funcionamiento de la aplicación. Por ejemplo las bases de datos, aplicativos, etc.

3.3.5 Pruebas de software

Son los encargados de verificar que las aplicaciones desarrolladas obtengan un mínimo de errores antes de entregarlas al cliente, realizando pruebas funcionales y no funcionales, de producto, pruebas basadas en los casos de uso, pruebas de aceptación para determinar si se ha cumplido los requisitos.

3.4 Capacitadores de las Aplicaciones

3.4.1 Capacitadores

Son las personas que capacitan a los clientes o usuarios en el uso de las aplicaciones, y las nuevas tecnologías implementadas en el proyecto, para tener un mejor desempeño, con ello también contribuyen al mejoramiento de la productividad, calidad y competitividad de la empresa.

3.4.2 Educadores virtuales

Los educadores virtuales permiten la utilización de un entorno tecnológico y las comunicaciones, ya que por medio de este podemos dar soluciones a requerimientos solicitados, capacitaciones, servicio de soporte, etc.

3.4.3 Personal de mesa de servicios

El personal de mesa de servicios es la que está dispuesto a prestar el servicio de soporte para dar solución a problemas e incidencias presentadas en el sistema de información de manera presencial o por medio de las plataformas tecnológicas utilizadas por la empresa a los clientes cuando ellos lo requieran.

3.4 Árbol de problemas

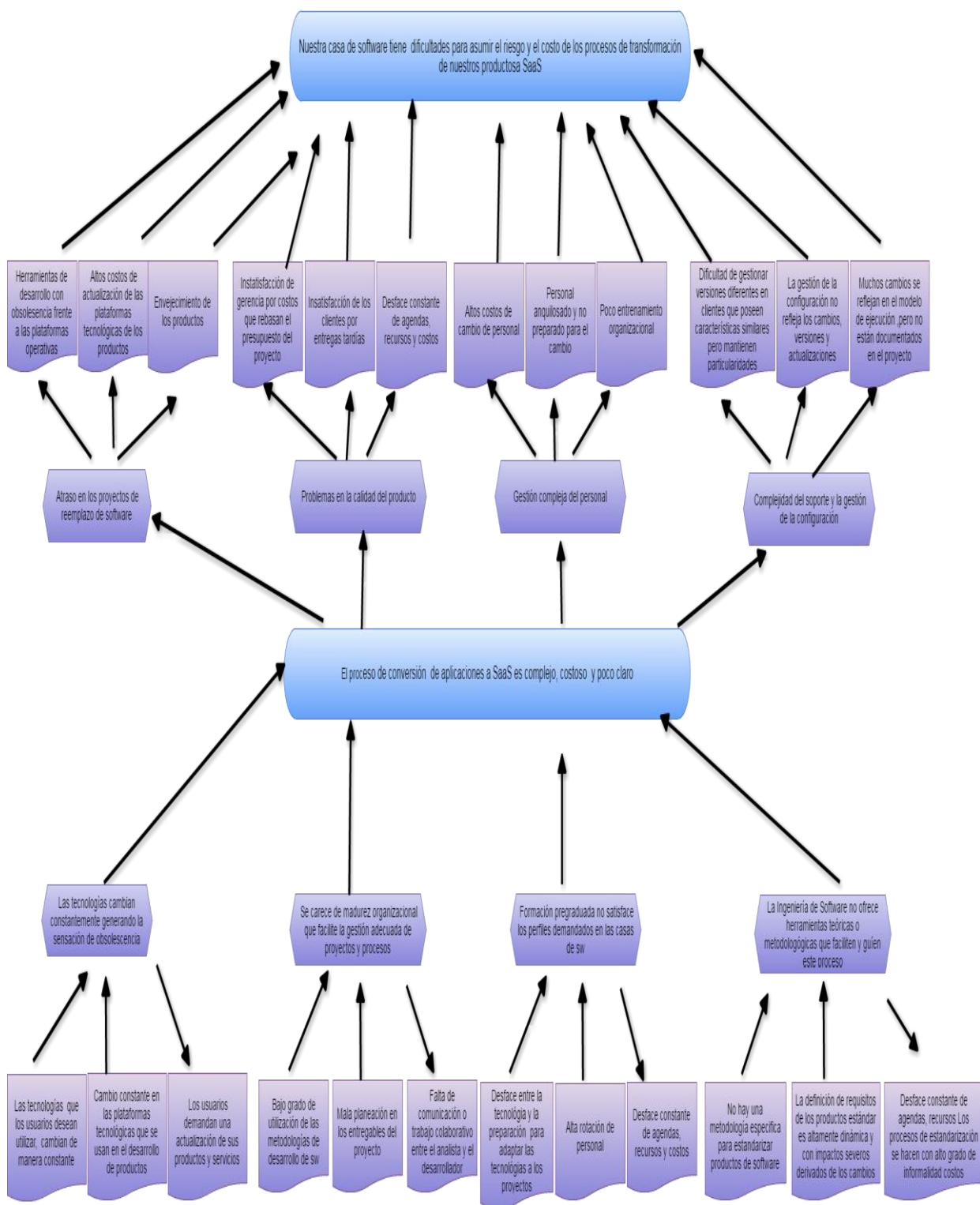


Ilustración 3: Árbol de Problemas

Fuente: Elaboración propia

3.5 Análisis de árbol de problemas

3.5.1 Problema

Se ha identificado que el proceso de conversión de aplicaciones a SaaS es complejo, costoso y poco claro, de la misma forma se ha señalado que existen múltiples factores que deben ser administrados y que dependen en gran medida del grado de madurez de la organización. Podemos por tanto concluir que el problema debe ser estudiado y administrado para cada organización que lo debe estudiar desde sus peculiaridades internas. Sin embargo, las características generales del problema pueden ser planteadas en la siguiente pregunta:

¿Es posible proponer un modelo gestionado y definido para la conversión de aplicaciones On-premises a SaaS, alineado con cmmi nivel 3 para la empresa OSR SAS?

3.6 Causas

3.6.1 Las tecnologías cambian constantemente generando la sensación de obsolescencia

La tecnología en los últimos años ha evolucionado de manera acelerada y debemos estar siempre preparados para esos cambios constantes en las plataformas tecnológicas que se usan en el desarrollo de las aplicaciones, tanto las casas de desarrollo de software como los usuarios deben estar actualizados en las tecnologías actuales para que sus productos o servicios mejoren el rendimiento en sus empresas.

3.6.2 Se carece de madurez organizacional que facilite la gestión adecuada de proyectos y procesos

Se hace necesario implementar procedimientos gerenciales y administrativos que faciliten la ejecución de todas las etapas que tienen los proyectos utilizando el modelo de madurez en la administración de los proyectos.

3.6.3 Formación pre graduada no satisface los perfiles demandados en las casas de software

La falta de personal capacitado en las nuevas tecnologías o competente para los perfiles que busca la empresa, genera la necesidad de que la empresa realice entrenamiento organizacional, esto lleva a que la empresa tenga unos costos altos en capacitación y entrenamiento.

3.6.4 La Ingeniería de Software no ofrece herramientas teóricas o metodológicas que faciliten y guíen este proceso

La ingeniería de software está orientada a cumplir unas fases de desarrollo e implementación de productos de software en general. La conversión de productos existentes a software como un servicio rebasa los límites de la reingeniería de software e incluso de las técnicas conocidas como refactoring.

Se carece de una metodología que permita especificar la estandarización de los productos de software, haciendo que los procesos de estandarización se realicen con alto grado de informalidad.

3.7 Efectos

3.7.1 Atraso en los proyectos de reemplazo de software

El atraso en los proyectos se debe a que no se define el alcance, se realiza una inadecuada toma de requisitos, los clientes no dan la información completa o cambian de idea de lo que desea en el proyecto durante la ejecución, las herramientas de desarrollo no están actualizadas acorde a las nuevas tecnologías que requieren implementar los usuarios, por esto se da el envejecimiento de los productos, otra causa en el atraso de los proyectos es el alto costo en la actualización de las plataformas tecnológicas de los productos.

3.7.2 Problemas en la calidad del producto

La Ingeniería del Software persigue como objetivo esencial el proporcionar las herramientas fundamentales para la producción de software de alta calidad, pero existen varias causas que originan los problemas en la calidad del producto como por ejemplo la mala planeación e implementación de los procesos, no se realiza el adecuado análisis en las fases de desarrollo, personal sin experiencia, utilización de plataformas incorrectas, esto causa Insatisfacción de clientes por entregas tardías, un desfase constante de agendas, recursos y costos.

3.7.3 Gestión compleja del personal

El departamento de recursos humanos es el encargado de suplir las necesidades del personal en la empresa de acuerdo al perfil requerido, pero existe una alta rotación del personal, ya que no son competentes para los cargos por falta de experiencia en las plataformas utilizadas en la empresa, no están preparados para el cambio, también al poco entrenamiento organizacional.

3.7.4 Complejidad del soporte y la gestión de la configuración

La Gestión de Configuración identifica y define los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

La complejidad en el soporte de los aplicativos se da porque no hay un manejo adecuado de las versiones, actualizaciones de los productos, no hay documentación del código, o un manual donde diga que cambios se han realizado, las versiones se crean porque hay clientes que piden implementar otros módulos en el aplicativo, para corregir errores, o simplemente actualizar la aplicación.

No hay un control sistemático de los cambios en la Configuración y el mantenimiento de la integridad y trazabilidad de los proyectos, los cambios se reflejan en el modelo de ejecución del proyecto, pero no están documentados

4 Árbol de objetivos

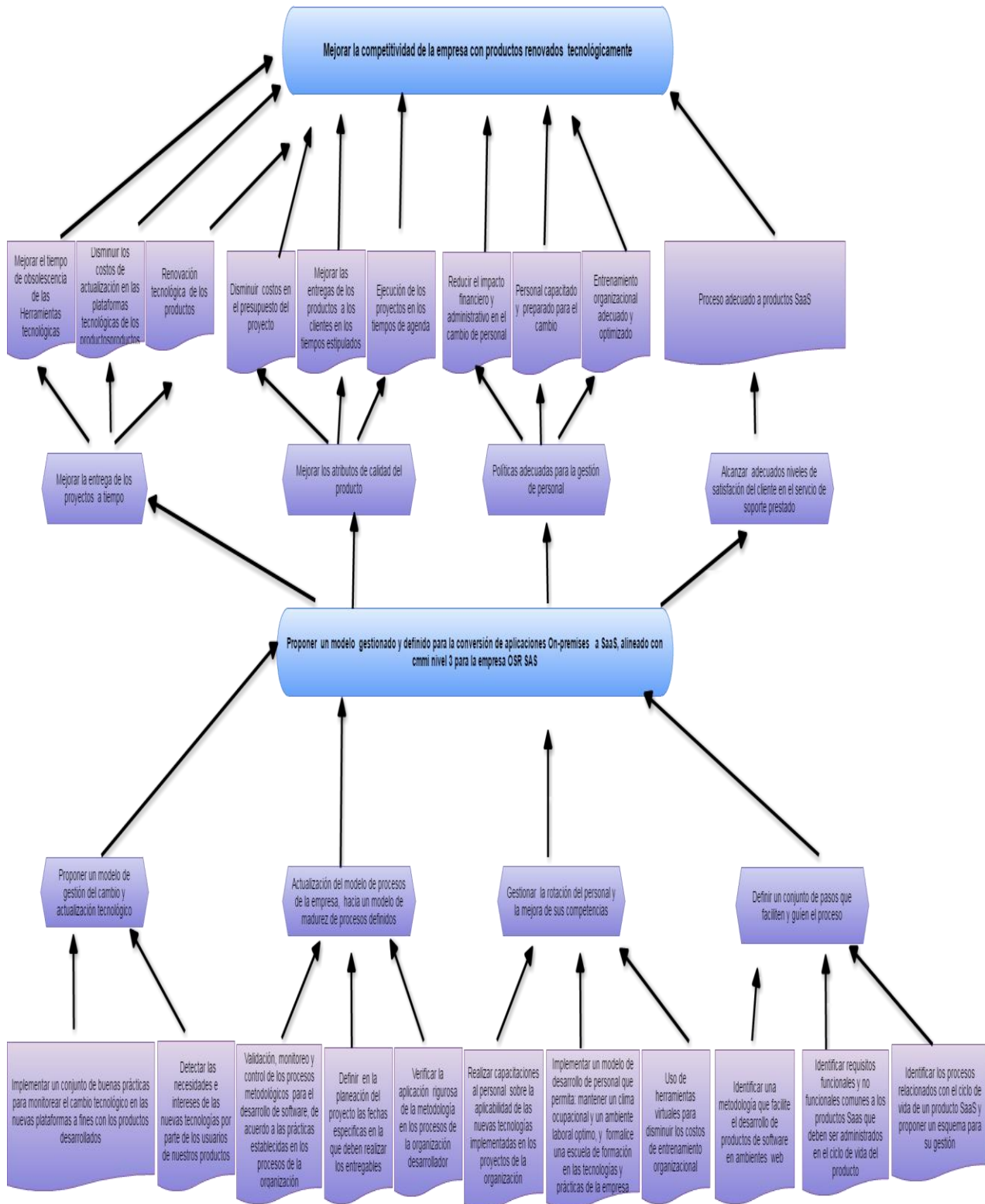


Ilustración 4: Árbol de Objetivos

Fuente: Elaboración propia

5 Análisis del árbol de objetivos

5.5 Medios

6.1.2. Proponer un modelo de gestión del cambio y actualización tecnológico

En la actualidad el crecimiento internacional y la globalización empresarial ha permitido el rápido desarrollo de las nuevas tecnologías de la información, provocando la exigencia de un proceso de cambio para la mejora continua en el modelo de gestión del cambio organizacional, los usuarios reclaman las nuevas tecnologías. El objetivo es implementar un conjunto de buenas prácticas para monitorear el cambio tecnológico en las nuevas plataformas a fines con los productos desarrollados, detectando las necesidades e intereses de los usuarios en las nuevas tecnologías.

6.1.3. Actualización del modelo de procesos de la empresa, hacia un modelo de madurez de procesos definidos

Se reestructura la planeación estratégica de la empresa creando un nuevo modelo de procesos orientados a servicios teniendo en el mapa de procesos, los procesos de apoyo, los procesos misionales y los procesos estratégicos. Se define en la planeación del proyecto las fechas específicas en la que se deben realizar los entregables. Se valida, monitorea y controla los procesos metodológicos para el desarrollo de software, de acuerdo a las prácticas establecidas en los procesos de la organización. El modelo de madurez como CMMI, está orientado a fortalecer los procesos ejecutados por las empresas y facilitan la gestión de proyectos.

6.1.4. Gestionar la rotación del personal y la mejora de sus competencias

El principal desafío que enfrentan las organizaciones es el mejoramiento continuo del personal que lo integra, por eso OSR pretende gestionar la rotación del personal y la mejora de sus competencias realizando capacitaciones al personal sobre la aplicabilidad de las nuevas tecnologías implementadas en los proyectos de la organización, adicionalmente Implementar un modelo de desarrollo de personal que permita: mantener un clima ocupacional y un ambiente laboral optimo, y formalice una escuela de formación en las tecnologías y prácticas de la empresa, con el uso de herramientas virtuales para disminuir los costos de entrenamiento organizacional

6.1.5. Definir un conjunto de pasos que faciliten y guíen el proceso

En la implementación de la metodología para mejorar los procesos de la empresa y obtener un mejor producto se identifica una metodología que facilite el desarrollo de productos de software en ambientes web, también se identifica los requisitos funcionales y no funcionales comunes a los productos Saas que deben ser administrados en el ciclo de vida del producto, otro medio que se identifico fue los procesos relacionados con el ciclo de vida de un producto SaaS y proponer un esquema para su gestión.

5.6 Fines

5.6.1 Mejorar la entrega de los proyectos a tiempo

Para mejorar la entrega de los productos a tiempo se actualiza las herramientas, plataformas de desarrollo que ayuden al mejoramiento de los productos, mejorar el uso de herramientas virtuales para disminuir los costos, disminuir los costos de actualización en las plataformas tecnológicas de los productos, definir bien el alcance del proyecto para cumplir con los tiempos estimados en el contrato para los entregables de los productos, mejorar el tiempo de obsolescencia de las herramientas tecnológicas según el entorno tecnológico en las nuevas plataformas a fines con los productos desarrollados.

5.6.2 Mejorar los atributos de calidad del producto

Para mejorar los atributos de los productos de software se revisa los alcances que tiene los proyectos, se define cronogramas de actividades, se realiza un trabajo colaborativo entre el grupo de desarrolladores, se realiza pruebas funcionales y no funcionales para garantizar la calidad de los productos, se disminuye costos en el presupuesto del proyecto y se mejora las entregas de los productos a los clientes en los tiempos estipulados en el contrato cumpliendo con el cronograma en los tiempos de agenda.

5.6.3 Políticas adecuadas para la gestión de personal

Implementar un modelo de desarrollo de personal teniendo en cuenta el reglamento y manual de procesos y procedimientos, el reglamento interno de trabajo, manuales de cargos y funciones de los empleados, mejorar el entrenamiento organizacional, construir un plan de incentivos, reducir el impacto financiero y administrativo en el cambio de personal, contratar personal capacitado y preparado para el cambio, con experiencia en las nuevas tecnologías, también garantizar el crecimiento de los empleados mejorando las condiciones y habilidades, generar un clima ocupacional y ambiente laboral óptimo para el desarrollo personal de los empleados, promover las facilidades de estudio formal en maestrías y certificaciones especializadas de productos y tecnologías emergentes.

5.6.4 Alcanzar adecuados niveles de satisfacción del cliente en el servicio de soporte prestado

Implementar una metodología que proporcione el proceso adecuado a productos SaaS, con un método sistemático de desarrollo para controlar los procesos en la construcción del software, para gestionar este conjunto de retos y problemas hace necesario el ajuste de metodologías que puedan guiar el proceso de desarrollo e implementación del nuevo servicio de software, adicionalmente mejorar la competitividad de la empresa con productos renovados tecnológicamente proponiendo un modelo gestionado y definido para la conversión de aplicaciones On-premises a SaaS, alineado con cmmi nivel 3

6 Análisis de alternativas

Las alternativas evaluadas para este proyecto son:

1. Desarrollar el proyecto en busca de un modelo de adecuación de las aplicaciones existentes a las nuevas plataformas de software como un servicio.
2. Mantener el actual modelo de desarrollo y priorizar el crecimiento horizontal y vertical de las aplicaciones On Premises, sin hacer cambios a las nuevas plataformas Cloud.

Para realizar el análisis de las alternativas se evaluó el impacto organizacional de las dos alternativas. Este impacto se midió con base en dos grandes ejes:

Ítem	Variable	SW Personalizado	SaaS
1	Crecimiento Ingresos	200%	600%
	Ventas máximas /año	\$ 1.500 MM	\$4.000 MM
	Máx. Nro. Clientes /año	20	200
2	Incremento del Costo Operacional	300%	200%
	Capacidad Operacional		
	Personas / Producto	60	20
	Mesa de Ayuda /prod.	2	10
	Capacitación	Presencial	Virtual
	Migración de datos	Inherente al desarrollo	Servicio adicional
	Tiempo de ejecución proyecto	inmediato	18 meses

1. Crecimiento de ingresos: las expectativas de ventas generan el crecimiento probable de ingresos; las ventas dependen de la capacidad que se tiene de vender múltiples copias, o instancias del mismo producto, minimizando los cambios y posibles personalizaciones. Por esa razón se observa un mayor valor de recuperación de inversión y el margen de utilidad es mayor con productos y servicios estandarizados que sobre productos altamente personalizados. Los cálculos se han tomado de la proyección de ingresos de la empresa por Proyectos de nuevos desarrollos frente a ventas de servicios.
2. Incrementos del costo operacional. Esta variable se mide respecto al costo que debe ser asumido para incrementar la capacidad operacional de la empresa, frente a la línea base actual, es decir, en cuanto debe crecer la actual infraestructura para alcanzar los niveles de ventas proyectados para generar los

ingresos esperado máximos. La capacidad operacional de la empresa engloba las características relacionadas con: Grupo de personas trabajando en una versión de producto, Mesa de ayuda por producto y por versión, Esfuerzo en capacitación virtual o personalizada, Capacidad de Migración de datos. Aunque existen otras variables solo se han evaluado estas y se presentan en la tabla de análisis.

Del análisis del cuadro se desprende que en el primer año de funcionamiento se justifica plenamente la inversión económica para la conversión a productos SaaS. Es necesario esperar un periodo contable para observar el efecto de la transformación. Pero en el primer año posterior al lanzamiento del producto ya se ha recogido la inversión de la transformación.

Por las anteriores razones, la alternativa seleccionada es la conversión a plataformas SaaS.

3.6 Gráfico Estructura analítica del proyecto – EAP

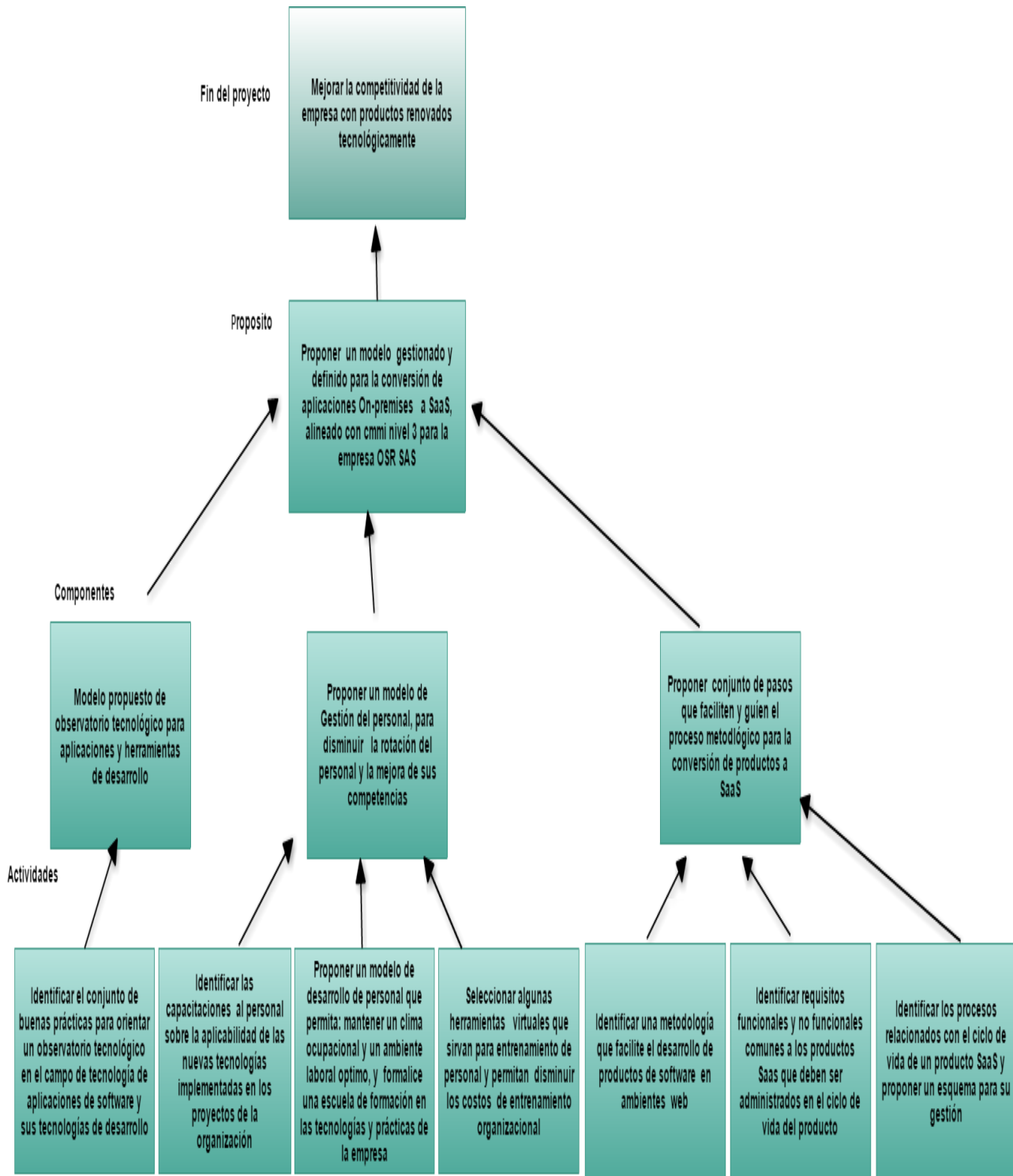


Ilustración 5: Estructura Analítica del proyecto

Fuente: Elaboración propia

4 Análisis de la Estructura analítica del proyecto

Fin del proyecto

Mejorar la competitividad de la empresa con productos renovados tecnológicamente

Se busca llegar este fin, implementando una metodología que proporcione el proceso adecuado a productos SaaS, también gestionar el conjunto de retos y problemas en OSR que hace necesario el ajuste de metodologías que puedan guiar el proceso de desarrollo e implementación del nuevo servicio de software, adicionalmente mejorar la competitividad de la empresa con productos renovados tecnológicamente proponiendo un modelo gestionado y definido para la conversión de aplicaciones On-premises a SaaS, alineado con cmmi nivel 3

Propósito

Proponer un modelo gestionado y definido para la conversión de aplicaciones On-premises a SaaS, alineado con cmmi nivel 3 para la empresa OSR SAS

Componentes

C1: Modelo propuesto de observatorio tecnológico para aplicaciones y herramientas de desarrollo

C2: Actualización del modelo de procesos de la empresa, hacia un modelo de madurez de procesos definidos. Este componente está fuera del alcance del proyecto.

Este componente no se ejecuta en el proyecto

C3: Proponer un modelo de Gestión del personal, para disminuir la rotación del personal y la mejora de sus competencias

C4: Proponer conjunto de pasos que faciliten y guíen el proceso metodológico para la conversión de productos a SaaS

Actividades

Actividades componente 1

Identificar el conjunto de buenas prácticas para orientar un observatorio tecnológico en el campo de tecnología de aplicaciones de software y sus tecnologías de desarrollo

Actividades componente 2 no se realizan porque no se ejecuta en el proyecto

Actividades componente 3

Identificar las capacitaciones al personal sobre la aplicabilidad de las nuevas tecnologías implementadas en los proyectos de la organización

Proponer un modelo de desarrollo de personal que permita: mantener un clima ocupacional y un ambiente laboral óptimo, y formalice una escuela de formación en las tecnologías y prácticas de la empresa

Seleccionar algunas herramientas virtuales que sirvan para entrenamiento de personal y permitan disminuir los costos de entrenamiento organizacional

Actividades componente 4

Identificar una metodología que facilite el desarrollo de productos de software en ambientes web

Identificar requisitos funcionales y no funcionales comunes a los productos SaaS que deben ser administrados en el ciclo de vida del producto

Identificar los procesos relacionados con el ciclo de vida de un producto SaaS y proponer un esquema para su gestión

4. LA IMPORTANCIA DEL CAMBIO A LA NUBE

A medida que el mundo se mueve hacia el software basado en la nube, muchos desarrollos de software ejecutivos luchan con la transición del empaquetado para nublado productos. Los indicadores de los proveedores de software exitosos pueden facilitar la decisión y, finalmente, el movimiento.

Crecimiento reciente en software basado en la nube como servicio (SaaS) se espera que continúe al 20 por ciento cada año hasta 2018, cuando el mercado global podría alcanzar casi \$ 85 mil millones. Cambio desde empaquetado o el software "on premise" para SaaS tiene un número de beneficios que incluyen una mejor experiencia de usuario y menores costos de entrega y soporte. También permite a las empresas acceder a nuevos mercados e incorporar software de nube innovador de terceros.

Un informe reciente estima que solo 8 por ciento de los ingresos generados por el Los 100 principales vendedores de software se originan de SaaS modelos, y siete de las diez empresas más grandes dibujar menos del 5 por ciento de sus ingresos de software de SaaS. Otra investigación muestra la Penetración de SaaS en la mayoría de las categorías de aplicaciones de software.

Penetración de SaaS se duplicará para 2018-y llegar a más de 20 por ciento sigue siendo bajo hoy, que van desde 1 hasta 36 por ciento. Por 2018, sin embargo, es compartir debería aumentar materialmente, logrando hasta 72 por ciento de penetración con algunas aplicaciones.

Si bien muchos proveedores todavía tienen que saltar al SaaS wagon, algunos que han estado entregando las experiencias de SaaS durante años están ocupadas mejorando sus arquitecturas técnicas para implementar la última generación de tecnologías en la nube. Estas incluye nuevos modelos de persistencia y base de datos (bases de datos en memoria o NoSQL, por ejemplo), plataformas analíticas más rápidas, interfaces de usuario adaptables, y computación elástica, entre muchos otros.

A medida que las empresas intentan hacer la transición del software empaquetado a SaaS o actualice las soluciones SaaS existentes para aprovechar las nuevas arquitecturas de la nube, a menudo enfrenta una serie de desafíos. Conversaciones con ejecutivos senior de desarrollo de software aparecieron una serie de inquietudes y preguntas sobre esta transición

¿A qué tipo de arquitectura de nubes deberían dirigirse?

¿Deben los desarrolladores usar infraestructura pública como servicio?

(IaaS) o plataforma como servicio (PaaS)

soluciones o elegir la nube privada? Ellos

¿Necesita volver a escribir su código completo?

Cómo gestiona la organización la transición a su estado objetivo, desde la arquitectura heredada hasta arquitectura orientada a servicios basada en la nube? Cómo de largo

debe tomar la transición y cuáles son los pasos clave? Debería la compañía esperar hasta la nube y los productos en las instalaciones alcanzan la paridad?

Qué cambios deberían hacerse al desarrollo y modelos operativos? En caso de desarrollo métodos pueden ser cambiados? ¿Cómo podría afectar este cambio? ciclos de lanzamiento de software? ¿La empresa tendrá que cambiar la forma en que se relaciona con los clientes?

¿Qué capacidad y cambios culturales tiene la organización? ¿necesitar? ¿Cómo debería una empresa construir el talento y capacidades necesarias, qué mentalidad y cambios de comportamiento que necesitan, y cómo seleccionan el desarrollo, la TI y la infraestructura correctos ¿enfoques?

Un proceso deliberativo comienza con una consideración cuidadosa de qué tipo de base de código, arquitectura modificación, y la infraestructura de la nube es más apropiado. Entonces, para garantizar una ejecución exitosa sobre la elección hecha, los ejecutivos del software, necesidad de hacer varios compromisos con respecto al alcance del producto, el enfoque del desarrollo, y la asignación de recursos.

Elegir el enfoque correcto

Empresas de software que están considerando hacer el cambio al software en la nube enfrenta tres decisiones críticas, y su forma óptima de avanzar dependerá en sus principales objetivos y puntos de partida.

La primera decisión se refiere a si hace más sentido con el tiempo para elegir una base de código unificada para software empaquetado y en la nube o tener un separate uno para cada uno. En definitiva, esta decisión se reduce a algunos factores clave. Primero, la organización la visión a largo plazo es importante al determinar el objetivo final de la aplicación. Es el equipo que intenta construir una aplicación optimizada para la nube o está tratando de aprovechar específica beneficios de la nube, mientras proporciona opciones para los clientes? El segundo problema se refiere los costos de mantenimiento para dos bases de código. En esto caso, cuánto tiempo piensa la compañía continuar con productos de software empaquetados y en la nube, y se requiere paridad de características? Para muchos programas vendedores, parece probable que el software empaquetado las suites no desaparecerán pronto. El último factor involucra talento y cultura. ¿El equipo tiene el deseo y actitud requeridos para aprender nuevas tecnologías y desaprender las prácticas de codificación del pasado?

Cuando una base de código unificada tiene sentido. Una base de código unificada podría ser preferible si los clientes actuales ver la nube como solo otro canal. Es decir, la compañía no espera a todos sus clientes para alejarse del software local en el corto a mediano plazo (ver cuadro de texto, "Cuando menos es suficiente"). O la empresa podría no necesitar la mejor de las variedades arquitectura en la nube para aprovechar los beneficios básicos de la nube (incluyendo elasticidad, escalabilidad, y bajo costo). Una base de código unificada funciona cuando la compañía tiene que mantener y administrar múltiples versiones del producto. Desde un punto de vista práctico, otra razón para elegir una solución unificada es que una compañía tiene evidencia de que su

desarrollo los equipos están dispuestos no solo a aprender nuevas tecnologías pero para desaprender las prácticas de codificación del pasado también.

Cuándo elegir bases de código separadas. Mantenimiento de bases de código separadas para software empaquetado y en la nube puede ser ideal cuando los gerentes ven la nube como el canal clave para el crecimiento futuro y esperamos eliminar el producto en las instalaciones. Si los clientes esperar que el producto basado en la nube sea diferente en términos de apariencia en comparación con el escritorio versión y también esperan que incluya características proporcionado por otras ofertas basadas en la nube (semanal lanzamientos, mejor escalabilidad y soporte para redes sociales herramientas, por ejemplo), luego las bases de código separadas pueden también sea la elección correcta. Otras razones para optar para bases de código separadas puede ser el hecho de que el la compañía no tiene que administrar la paridad de características entre los productos basados en la nube y en las instalaciones, ya que pronto eliminará este último o el El equipo de software debe reconsiderar por completo al usuario experiencia y tiene las habilidades necesarias para ejecutar.

La segunda decisión crítica: las empresas pueden elija refactorizar y "rediseñar" sobre la marcha o construir una arquitectura completamente nueva. Cuando están haciendo esta decisión, los líderes deben considerar dos factores: la viabilidad de la arquitectura actual dado el hoja de ruta proyectada del software de la compañía productos y el requisito de time-to-market.

Cuándo refactorizar

Refactorizar es típicamente mucho más rápido y preferible si la arquitectura actual podría no ser ideal para la arquitectura de la nube, pero tiene elementos estructurales básicos tales como identificables capas. También tiene sentido si los desarrolladores pueden portar aplicaciones de varios niveles a la arquitectura de la nube sin realizar una reescritura completa. Otro punto a favor de la refactorización es cuando la compañía necesita lanzar la primera basada en la nube versión tan pronto como sea posible. Completamente refactorizado, la arquitectura basada en servicios puede ayudar a conducir con frecuencia y lanzamientos pequeños, pero no es una necesidad para Comience con el producto basado en la nube.

Cuándo diseñar una nueva arquitectura

Desarrollar una nueva arquitectura tiene sentido si la corriente el diseño simplemente no es adecuado para la nube. Por ejemplo, podría ser una arquitectura monolítica eso demuestra síntomas del código "spaghetti". Otra consideración es la escalabilidad de la arquitectura.

A veces una arquitectura construida originalmente para las instalaciones no está diseñado para escalar hasta un mayor número de usuarios o no es compatible multi Alquiler. Las empresas a menudo construyen relacionadas "Deuda técnica" debido a la arquitectura previa decisiones. Por ejemplo, una empresa de procesamiento de nómina decidieron revisar su arquitectura actual para poder mover a la pila abierta ya que la portabilidad es un requisito clave para ellos. La compañía construyó parte del nuevo sistema desde cero, mientras aprovechando tácticamente los motores de cálculos estables y otros componentes. Incluso mientras se vuelve a diseñar gran parte de la pila,

la compañía no actualizó una pocos componentes maduros, incluidos algunos en mainframe, ya que los riesgos de actualizar esos componentes superado los beneficios potenciales.

Otra decisión crítica es la elección de público versus infraestructura de nube privada. Las empresas pueden construir sus productos en la parte superior, ya sea de forma privada plataformas alojadas o IaaS públicas o PaaS que confía en un proveedor de servicios. Esta decisión principalmente se refiere a las economías de escala, ya que la escala de infraestructura despliegue, el tolerancia de la empresa para el riesgo (seguridad de los datos) o rendimiento problemas, por ejemplo), y regulatorio requisitos finalmente conducirlo. Plataformas IaaS

Seguridad de datos y escala de despliegue son factores al hacer el público versus nube privada decisión

Proporcionar flexibilidad y control pero implica la compensación de complejidad adicional y el esfuerzo inicial requerido para construir un servicio listo para el usuario para ellos. Por el contrario, las plataformas PaaS a menudo ofrecen muchas capacidades que puede ayudar a las empresas a acelerar el transición a la nube, pero estas plataformas en general incluir capacidades propias o específicas del vendedor. Como tal, requieren software creado para un plataforma y pila de vendedores específicos, bloqueando en esos proveedores. Mientras que un pequeño grado de vendedor lock-in existe con los sistemas IaaS, es relativamente fácil de planear alrededor de esas áreas.

Cuándo ir por una nube privada.

Las nubes privadas funcionan cuando el desarrollador tiene suficiente interna escalar para lograr un costo total de propiedad comparable a las elecciones públicas. Eso típicamente significa que emplea decenas de miles de máquinas virtuales (VMs) También es la elección correcta si al menos uno de las siguientes cuatro consideraciones son fundamentales para sistema o aplicación específica y, por lo tanto, excluye el uso de la nube pública: seguridad de los datos, problemas de rendimiento, control en caso de una interrupción, o tecnología de bloqueo. Un último factor involucra requisitos regulatorios que podrían restringir los esfuerzos para almacenar datos de clientes fuera del conjunto geográfico límites o prevenir el almacenamiento de datos en multitenant ambientes

Cuándo elegir la nube pública

Los desarrolladores deben considerar el enfoque de la nube pública si el proyecto carece de escala suficiente (no involucrará decenas de miles de máquinas virtuales, por ejemplo) o un alto grado de incertidumbre existe con respecto a la demanda probable. El uso de una nube pública es más eficiente en términos de capital enfoque, ya que la creación de una nube privada requiere recursos significativos que la compañía podría Probablemente invertir más eficazmente en la corriente principal negocio. Otra razón para hacerlo público: el sistema o la aplicación es tolerante a la latencia. Experiencia muestra que los niveles de latencia en nubes públicas pueden varían hasta 20 veces dependiendo de la hora del día. También tiene sentido si no hay requisitos regulatorios específicos que las aplicaciones almacenar los datos en un lugar particular más allá del rendimiento necesariamente. Incluso si las compañías deciden usar una nube privada

para sus aplicaciones más críticas, muchos deciden usar la nube pública para obtener más casos de uso básicos (cargas de trabajo de desarrollo / prueba, adicionales capacidad temporal, por ejemplo).

Seis principios de diseño de salto a la nube

Una vez que los ejecutivos hayan hecho su base de código, arquitectura, y las decisiones de infraestructura, comienzan desarrollando su software basado en la nube. Para mejorar entender cómo los jugadores de software con éxito hacer la transición, McKinsey revisó un número de casos externos y entrevistas en profundidad realizadas con los principales jugadores de software. Aquellos que triunfar en el viaje desde las instalaciones a la nube el desarrollo de software comparte seis compromisos.

Dispara para el producto viable mínimo en su lugar de paridad de características

Organizaciones que mueven productos a la nube a menudo descubren que lograr la plena paridad podría tomar varios años. En lugar, los vendedores exitosos a menudo deciden lanzar un producto viable mínimo (MVP) a los clientes en de seis a nueve meses. Esta estrategia les permite prueba su arquitectura y funcionalidad rápidamente. El enfoque también los obliga a pensar profundamente sobre qué tipos de funcionalidades del producto ofrecen experiencias de cliente central buscadas y que tienen que enfatizar para obtener esa funcionalidad derecho. Al poner un MVP viable con el mayor características importantes en las manos del usuario tan rápido como posible, el equipo puede reunir elementos cruciales comentarios iniciales de los clientes y mejorar rápidamente sus habilidades de desarrollo basadas en la nube.

Tratar a los usuarios como parte del desarrollo diario equipo

Los desarrolladores necesitan interactuar con sus clientes temprano y con frecuencia, y cambiar a un modelo de nube abre nuevas formas de interactuar con ellos. Los equipos pueden recibir comentarios de los clientes en casi en tiempo real tan pronto como antes o incluso antes liberar una característica. El compromiso del usuario también permite desarrolladores y gerentes de producto para preguntar a los clientes para priorizar sus necesidades a través de blogs cuando el producto está en la fase de concepto y proporciona una producto básico para primeros usuarios específicos. Ellos pueden luego codifique la versión completa con ellos. La experiencia muestra que se recogen los primeros comentarios puede ayudar a los equipos a dar forma a cómo priorizan las características que todavía están en desarrollo.

Ejecución centralizada de la aplicación para todos los clientes también abre nuevas capacidades. Los desarrolladores pueden, por ejemplo, emplea el registro y el análisis para comprender las acciones de los clientes, tomando una gran cantidad de datos enfoque para rastrear sus patrones de uso. Del mismo modo, la realización de funciones y funciones "A / B" las pruebas dan a los equipos un enfoque basado en datos a la toma de decisiones. El 2012 Obama presidencial campaña en los Estados Unidos, por ejemplo, utilizada alrededor de 500 pruebas A / B en la interacción, copia e imágenes utilizadas en su página web. El enfoque aumento de las donaciones en casi un 50 por ciento e inscripciones por más del 160 por ciento.

La nube también permite a los equipos implementar la funcionalidad de forma controlada (el primer 1 por ciento de clientes, luego un 5 por ciento si todo va bien, luego 10 por ciento, etc.).

Esperar y tolerar fallas

Infraestructura de la nube trae muchos beneficios, incluida la capacidad de crecer o reducir recursos para una aplicación en tiempo real. Sin embargo, la naturaleza compartida de la infraestructura en la nube puede plantear desafíos debido a factores más allá del control del desarrollador, como hardware o fallas o ralentizaciones de la red. Y, como con todos los datos del cliente centralizados en el entorno de la nube, los desarrolladores necesitan diseñar una arquitectura para la aplicación que puede acomodar estas fallas y trabajar alrededor de ellas.

Para tener éxito, las empresas necesitan desarrollar una mentalidad que acepte fallas que están más allá del control del desarrollador y diseñar una arquitectura para trabajar alrededor de ellas.

Para tener éxito, las empresas necesitarán desarrollar una mentalidad que acepte fallas. Sin ella, los desarrolladores dudarán en hacer cambios, haciendo que los ciclos de publicación se detengan. Un proveedor de contenido de Internet aprendió esta lección de la manera difícil después de experimentar el servicio de interrupciones debidas a servicios web de terceros falla del proveedor. En respuesta, la compañía hizo sus aplicaciones más robustas frente a tales interrupciones. Ahora, si ocurren problemas similares, sus aplicaciones están diseñadas para proporcionar un poco disminuido experiencia del cliente en lugar de una completa choque. Además de esto, para simular fallas aleatorias, la compañía creó una herramienta especial en forma de un script que matará indiscriminadamente la infraestructura de servicios. Este enfoque le permite probar la aplicación respuestas contra fallas que pueden eventualmente ocurrir. También ayuda a los equipos a aprender sobre los desafíos específicos para el desarrollo basado en la nube e incorporar entradas del cliente temprano.

Adoptar enfoques ágiles y DevOps

Las empresas deberían adoptar un pensamiento ágil y DevOps, un software enfoque de desarrollo que se centra en entrega del producto, aseguramiento de la calidad (QA), función desarrollo y lanzamientos de mantenimiento. DevOps se basa en muchos conceptos "ágiles", como trabajar en equipos multifuncionales y en iteraciones cortas todo el camino a la implementación. Ejecutivos de software también necesitan integrar su QA, operaciones y organizaciones de seguridad con sus equipos de I + D y programar al menos un lanzamiento por mes. Continuo integración, incluida la integración en el rama de software, se debe implementar con la frecuencia mínima diaria. Las versiones deben ser programadas con la mayor frecuencia posible para garantizar comentarios del usuario. El ciclo de lanzamiento puede variar desde varios lanzamientos por día a uno por mes. Una vez la base de código se re fabrica en servicios granulares, es posible lograr ciclos de liberación muy cortos sin desestabilizando toda la base del producto.

La necesidad de este cambio va al corazón de las diferencias entre el software empaquetado y en la nube. Con el software empaquetado, las versiones son caras porque los equipos tienen solo una verdadera oportunidad de lanzar un producto. En consecuencia, ocurren las liberaciones una o varias veces cada 24 meses. En una nube medio ambiente, en cambio, la mayoría de los vendedores encuentran que las versiones incrementales reducen la complejidad de despliegue y la magnitud de las fallas potenciales en el momento del lanzamiento. La versión incremental enfoque conduce a docenas de pequeñas versiones de un producto individual.

De acuerdo con un estudio reciente de Puppet Labs-a los principales proveedores de software de DevOps: equipos que abrazar DevOps y prácticas de lanzamiento continuo despliega código 30 veces más frecuentemente, tiene medio el número de fallas de producción, y puede restaurar servicios 12 veces más rápido después de un problema de producción. La investigación de McKinsey muestra que abrazar ágil el pensamiento aumenta la productividad del equipo en un promedio del 27 por ciento, aumenta la puntualidad de la función liberaciones en un 30 por ciento, y disminuye la tasa de defectos residuales en un 70 por ciento.

Ofrecer control de calidad a los desarrolladores y probar la responsabilidad

Otro sello distintivo de mudarse exitosamente del paquete Cloud es la elección de las empresas para hacer que sus desarrolladores de software -no los que evalúan el código- rindan cuentas por su calidad. Estas empresas parecen desdibujar los límites entre el desarrollo y roles de QA. La idea es permitir el software desarrolladores para resolver problemas críticos de inmediato como se vuelven aparentes Este enfoque requiere ellos para implementar soluciones críticas de forma continua, además para ejecutar ciclos cortos de lanzamiento. Hace sentido: a pesar de la naturaleza iterativa del software ágil desarrollo, los usuarios de la nube no aceptarán ni usarán aplicaciones con problemas importantes sin resolver. Y es también eficiente: un desarrollador puede corregir un error introducido Hace apenas dos semanas mucho más rápido que uno que fue introducido hace seis meses o dos años.

Otro factor muy importante es que los desarrolladores entiendo que la solución de problemas de SaaS es fundamentalmente diferente de solucionar problemas con paquetes software, que les exige adoptar nuevas prácticas. Es normal, por ejemplo, esperar clientes para poner sus servidores fuera de línea para depurar un problema para el software local. Esta no es una opción para software basado en servicios, ya que los clientes de múltiples zonas horarias están utilizando el servicio. Edificio capacidades avanzadas de diagnóstico y rastreo en el software es mucho más imprescindible para cloudbased software. Otro ejemplo similar es NoSQL adopción de la base de datos, que requiere una importante desaprendizaje de cómo los desarrolladores trabajaron con tradicional, bases de datos relacionales.

Invertir en capacidades de vanguardia y automatizadas entornos de prueba.

Las observaciones de McKinsey de desarrolladores de software exitosos sugieren que la contratación de los mejores talentos de desarrollo que pueden inyectar nueva

experiencia externa en la organización a nivel operacional y administrativo es fundamental para cambiar al software en la nube. Otro habilitador crucial es invertir en herramientas e infraestructura para impulsar el modelo de desarrollo enfocado en la nube. La organización debería cambiar todas las operaciones de creación, integración y prueba a un modelo continuo y automatizado que admita ciclos de liberación rápidos. Los principales proveedores de software en la nube, como algunas de las empresas de búsqueda y redes sociales más grandes del mundo, compilan y prueban regularmente la base de código completo varias veces al día. Las empresas pueden reducir estos intervalos de tan solo 15 minutos, pero hacerlo requiere un entorno de TI muy avanzado. Los principales jugadores de la nube proporcionan un entorno donde los desarrolladores pueden probar el cambio de código en contra de cualquiera de los productos de la cartera que podrían verse potencialmente afectados. Esto permite a los diseñadores realizar pruebas de integración sólidas por su cuenta antes de enviar el código para la integración real.

Éxito de la caja a la nube requiere continuo, pruebas automatizadas e integración y una TI avanzada ambiente

Incluso si genera mucho ruido, basado en la nube SaaS sigue siendo una parte relativamente pequeña de la mayoría de los principales carteras de productos de desarrolladores de software. Como el parte del software basado en la nube crece, los desarrolladores tendrá que centrarse cada vez más en la transición lejos del software en el local empaquetado. Alcanzar decisiones de tecnología cuidadosamente consideradas y comprometerse con varias organizaciones y enfoques operativos para desarrollar software como un servicio puede ayudar a los desarrolladores a hacer una transición exitosa del software empaquetado a la nube.

5. RECOMENDACIONES EN LA MIGRACIÓN A LA NUBE

La transición a Software como servicio (SaaS) no siempre es fácil. Conocer los pasos a seguir y no hacer de la migración e implementación de SaaS es esencial cuando se utiliza SaaS

Computación en la nube y servicios alojados

Cuando una empresa migra de un modelo tradicional de aplicación de software local, a un modelo de entrega de software como servicio, hay algunos cambios que las empresas tendrán que hacer en su entorno de software existente, junto con algunas tareas y procesos que deberán llevar a cabo, para garantizar que tengan una transición sin problemas. Sin embargo, estos procesos serán de beneficio para ellos a largo plazo, ya que eliminarán la cargas innecesarias que las aplicaciones de software locales colocan en sus centros de datos y reducirán en gran medida los problemas relacionados con la administración de software.

Migración de un entorno de software a una versión SaaS (antes y después)

Lo mejor de migrar de un modelo de aplicación de software local tradicional a un modelo de entrega de aplicaciones SaaS alojado es que la mayoría del trabajo ya está hecho para ti. En otras palabras, las aplicaciones de software ya están instaladas, protegidas y en funcionamiento, listas para ser utilizadas. Los clientes lo hacen no tiene que comprar hardware adicional o servidores nuevos, no necesitan instalar sistemas operativos en estos servidores y no tienen que perder el tiempo hora de configurar estos servidores.

Sin embargo, esto no significa que una migración desde un entorno de software estándar a un entorno de software basado en SaaS no implique ningún trabajo. Negocios todavía tienen que definir claramente cuáles son sus procesos comerciales y poner a su fuerza de trabajo detrás del nuevo entorno de software, entre muchas otras cosas.

Cosas que hacer y considerar antes de la migración

Lo primero que una empresa querrá hacer, incluso antes de sopesar si su infraestructura de hardware existente cumplirá con los requisitos de sus nuevos servicios de aplicaciones SaaS definir claramente cuáles son sus procesos comerciales reales.

Usar cualquier tipo de nueva tecnología o software puede confundir fácilmente a una empresa y a sus empleados sobre los procesos comerciales a los que se aplica esta nueva aplicación de software y qué empleados abordarán con este nuevo software. Al definir claramente cada uno de sus procesos y asignar roles a cada uno de sus empleados para su nueva infraestructura de SaaS antes de que realmente se implemente, una empresa puede evitar confusiones en una etapa posterior.

Además de definir claramente los roles de sus nuevas aplicaciones y empleados de SaaS en el esquema general de las cosas, una empresa también debería ser capaz de responder

una serie de preguntas importantes antes de implementar SaaS en su entorno empresarial. Estas preguntas se detallan brevemente a continuación.

¿Por qué están implementando SaaS?

Una empresa debe tener claro por qué están implementando SaaS y deberían poder justificar su implementación enumerando los beneficios que proporcionará (llevando a cabo un análisis de costo-beneficio, por ejemplo). Si los beneficios de implementar SaaS superan los costos de implementación (incluidos los costos iniciales, los costos de suscripción, los costos relacionados con los cambios en su propia infraestructura, etc.), una empresa puede implementar SaaS dentro de su entorno empresarial, si así lo desean.

¿Saben qué quieren obtener de la implementación de SaaS?

No es bueno implementar SaaS, pero luego no saber qué vas a sacar de él. Las empresas deben tener una serie de objetivos que desean lograr mediante el uso de sus nuevos servicios de software. Por ejemplo, una empresa puede querer que sus aplicaciones de software SaaS reduzcan sus costos de administración de software o mejoren la colaboración de datos entre varios sitios comerciales diferentes que puedan tener. Una empresa debe tener claro qué quiere obtener del uso de sus aplicaciones de software, independientemente de lo que pueda ser realmente.

¿Han considerado el verdadero costo de implementar SaaS?

Aunque inicialmente no parezca así, cuando se implementa SaaS, las empresas tienen más costos que solo sus costos iniciales de suscripción. Por ejemplo, es posible que necesiten considerar los problemas de licencia, dependiendo de cuántas instancias de un servicio de software en particular requieran y según cuántos usuarios permita un proveedor de servicios SaaS, accediendo a una sola instancia de ese servicio de software.

Deben considerar los costos internos, como los costos laborales de implementar SaaS o cualquier costo debido a un cambio en su infraestructura (por ejemplo, pueden requerir conexiones de red más rápidas para respaldar sus servicios SaaS). Necesitan pensar en los costos de capacitación y, finalmente, cómo los usuarios van a acceder a estos servicios de software (por ejemplo, ¿necesitarán computadoras portátiles nuevas para acceder a estos servicios de forma remota?), Así como los costos relacionados con el almacenamiento. Es evidente que hay varios costos diferentes que enfrenta una empresa al implementar SaaS y deben hacer todo lo posible para no pasar por alto ninguno de estos costos.

¿Sus proveedores de servicios de software SaaS les proporcionan un SLA sólido?

Sin un acuerdo de nivel de servicio (SLA) sólido implementado, los proveedores de servicios de software de SaaS pueden abusar de sus clientes al no proporcionarles el servicio que pagaron. Sí, un proveedor de software seguirá ofreciendo sus servicios de software a sus clientes, pero la calidad y disponibilidad de estos servicios de software no está garantizada. Las empresas deben elegir un proveedor de servicios SaaS que

tenga un sólido acuerdo de nivel de servicio, que defina la calidad exacta y la disponibilidad de los servicios de software que entregarán a sus clientes, junto con cualquier repercusión que enfrentarán si no cumplen estos requisitos.

Está claro que la mayoría de las tareas que realiza una empresa, antes de implementar SaaS como parte de su infraestructura, están planificando tareas basadas. Una empresa necesita planificar cuidadosamente y descubrir por qué están implementando SaaS. Solo cuando sus razones justifiquen los costos de implementación de SaaS y resalten los beneficios que una empresa recibirá cuando se implemente SaaS, deberían incorporar SaaS como parte de su infraestructura.

Cosas que hacer después de la migración

Al implementar aplicaciones de software locales tradicionales, primero una empresa tiene que configurar su hardware, luego tienen que instalar estas aplicaciones de software en sus plataformas de hardware, luego tienen que configurar sus aplicaciones de software y, finalmente, tienen que protegerlas, así como prueba ejecutarlos.

Implementar SaaS es mucho más simple que esto y es muy rápido. La única desventaja real es que las aplicaciones de software SaaS actualmente no ofrecen el mismo nivel de personalización que las aplicaciones de software tradicionales locales. Sin embargo, la mayoría de los proveedores de software pueden vivir con este hecho.

La mayoría de los proveedores de servicios de SaaS ofrecen a sus clientes opciones de personalización de 'apuntar y hacer clic'. La personalización de sus servicios de aplicaciones de software SaaS es lo principal que debe hacer una empresa, una vez que se hayan implementado sus servicios de software SaaS. La razón principal para que una empresa personalice sus servicios de aplicaciones de software SaaS es que estos servicios de aplicaciones de software pueden ayudar de manera más eficiente a una empresa a cumplir con sus propias necesidades específicas.

Lo último que debe hacer una empresa después de implementar SaaS con éxito es probar estos servicios de aplicaciones de software. Por ejemplo, podrían probar la compatibilidad predeterminada de sus navegadores de Internet con estos servicios de aplicaciones de software. Podrían probar qué tan eficientemente funcionarían sus servicios de aplicaciones de software en dispositivos móviles, como computadoras portátiles, por ejemplo, o simplemente podrían probar qué tan eficientes eran sus servicios de aplicaciones de software, probando si era probable que chocaran con cargas pesadas o no.

Un simple cómo guiar

La implementación y despliegue real de las aplicaciones de software SaaS es un proceso relativamente rápido y fácil. De hecho, las empresas dedicarán más tiempo a los procesos relacionados con la planificación y determinar por qué SaaS se está implementando y si vale la pena implementarlo, en lugar de la etapa de implementación real de estos servicios de software. Hay varias maneras diferentes en que una empresa

puede implementar SaaS, pero la mayoría de los métodos incluirán las etapas que se detallan a continuación (o una variación de estas etapas).

1. Justificar la implementación de SaaS utilizando una amplia gama de criterios

Existen dos modelos principales de implementación de software para empresas, que consisten en un modelo de implementación de software local y un modelo de implementación de software SaaS alojado. Cuando una empresa decide que desea implementar SaaS, debe llevar a cabo una investigación exhaustiva, determinando si SaaS será beneficioso para ellos o no.

Los servicios de software SaaS no solo deben coincidir con sus criterios de investigación, sino que también deben coincidir con las aplicaciones de software tradicionales locales en función de sus criterios de investigación. Al hacer esto, una empresa puede comparar y contrastar ambos modelos de implementación de software, lo que los colocará en una mejor posición para justificar qué modelo de implementación de software les será de mayor beneficio.

El costo es un criterio de investigación obvio, por ejemplo, el modelo de implementación de software con los costos iniciales más baratos será favorecido por la mayoría de las empresas. Sin embargo, las empresas no deberían basar su selección final solo en el costo. También se deben considerar otros criterios de investigación, que incluyen:

■ La flexibilidad que ofrece cada modelo de implementación de software. Las aplicaciones de software locales generalmente ofrecen a las empresas más flexibilidad pero, por supuesto, generalmente tienen costos iniciales más altos que las aplicaciones SaaS.

➤ Riesgos potenciales que puede presentar cada modelo de implementación de software. Por ejemplo, un modelo de implementación de SaaS puede amenazar la privacidad de una empresa, ya que sus datos comerciales se transfieren a través de Internet y, en la mayoría de los casos, se almacenan en los centros de datos de los proveedores de software.

➤ Problemas relacionados con la licencia. Por ejemplo, cuántos usuarios pueden acceder a una única instancia de una aplicación de software particular si se usa SaaS y un proveedor de servicios SaaS cobrará extra si más usuarios de los permitidos usan una sola instancia de un servicio de software en particular.

➤ Cuánta capacidad de almacenamiento proporcionarán los proveedores de servicios SaaS a sus clientes y cuáles son los costos de almacenamiento adicional.

➤ ¿Qué tipo de soporte es ofrecido por el proveedor de software y es este soporte gratuito?

➤ Costos adicionales que pueden no ser tan obvios como los costos iniciales que enfrenta una empresa. Por ejemplo, el costo de los profesionales de TI para administrar

las aplicaciones de software de las instalaciones o los costos de proporcionar capacitación con el fin de enseñar a los empleados cómo conectarse y utilizar los servicios de aplicaciones de software SaaS.

Una empresa debe comparar ambos modelos de implementación de software (cuando corresponda) con los criterios de investigación anteriores. Una vez hecho esto, una empresa podrá ver más claramente qué modelo de implementación de software es el mejor para ellos. Otros dos criterios de investigación que una empresa puede utilizar (que son aplicables a SaaS), es si podrán migrar fácilmente a un modelo de implementación de software local en el futuro, una vez que se haya implementado SaaS y qué a largo plazo, perspectivas de su proveedor de servicios SaaS.

2. Negocie un contrato que sea adecuado para ellos

Entonces, una empresa ahora justificó la razón para implementar SaaS utilizando los criterios anteriores (así como otros criterios). El siguiente paso para ellos implica negociar un contrato adecuado con su proveedor de servicio SaaS.

Entonces, una empresa ahora justificó la razón para implementar SaaS utilizando los criterios. La mayoría de los proveedores de servicios SaaS a menudo no brindan Acuerdos formales de nivel de servicio (SLA) a sus clientes. La razón de esto es porque no quieren asumir la responsabilidad adicional que les brinda un Acuerdo de Nivel de Servicio. En cambio, los proveedores de servicios SaaS a menudo tienen cláusulas de entrega de mejor esfuerzo en sus contratos, pero esto no garantiza nada a un cliente y los proveedores de software pueden abusar fácilmente de un contrato que contiene cláusulas de entrega de mejor esfuerzo, debido a este hecho. Para protegerse y asegurarse de obtener exactamente lo que pagan, una empresa debe solicitar un contrato razonable de acuerdo de nivel de servicio de su proveedor de servicio SaaS.

Las empresas también querrán asegurarse de que no haya costos ocultos en su contrato. Por ejemplo, hay una cláusula que establece que su suscripción solo incluye una cierta cantidad de transacciones por mes y que las transacciones adicionales generarán cargos (o algo similar a esta cláusula). ¿El servicio de atención al cliente seguirá siendo gratuito o se cobrará a una empresa después de que se haya agotado un cierto número de tickets de soporte (si se utiliza un sistema de soporte basado en tickets)?

Finalmente, una empresa querrá confirmar qué cláusulas relacionadas con la recuperación y la seguridad figuran en su contrato. Por ejemplo, ¿qué procedimientos de respaldo tiene un proveedor de servicio SaaS, si corresponde? ¿El proveedor del servicio será responsable por la violación o corrupción de los datos de sus clientes cuando se almacenan dentro de sus propios centros de datos (después de todo, debe ser responsabilidad del proveedor del servicio asegurar sus propios centros de datos).

Todas estas son consideraciones importantes para que una empresa realice cuando están negociando su contrato de SaaS y si no están contentos con algo, entonces ellos deben solicitar los cambios que requieren.

3. Use los métodos de implementación de mejores prácticas

Una vez que una empresa ha justificado la razón para implementar SaaS y ha negociado un contrato adecuado, puede implementar SaaS como parte de su infraestructura de software. La implementación real de los servicios de software SaaS es un proceso relativamente sencillo. Sin embargo, las empresas aún deben seguir los procedimientos de mejores prácticas generalizadas que seguirían cuando implementen el software local (o cualquier otra tecnología nueva para ese caso), a fin de ayudar a minimizar las posibilidades de que surja algún problema.

Confirmando nuevamente los objetivos que la nueva solución SaaS les ayudará a lograr

Una empresa ya debería tener una buena idea desde el principio, sobre lo que sus nuevos servicios de software SaaS les ayudarán a lograr y lograr. Sin embargo, después de negociar un contrato adecuado, una empresa debería confirmar una vez más que los servicios de SaaS que van a implementar seguirán ayudándolos a alcanzar sus metas y objetivos. Si este sigue siendo el caso, entonces una empresa puede seguir adelante e implementar SaaS como parte de su infraestructura de software.

Construyendo un equipo adecuado

Una vez que se implemente SaaS, una empresa requerirá una cantidad de profesionales de TI para administrar y mantener sus nuevos servicios de software. Una empresa debe definir roles dentro de su organización y una empresa también debe crear un equipo (o grupo de equipos) que se encargue de sus nuevas ofertas de SaaS, antes de que se implementen realmente sus servicios de software SaaS.

Algunos de los roles dentro de una empresa que tiene una infraestructura de software basada en SaaS incluyen un administrador de software SaaS (que se encargará de los servicios de software SaaS). Un comité de grupo de usuarios (que sugerirá cambios para afinar los servicios de software para una empresa), así como un desarrollador de software SaaS (que podrá realizar los cambios en los servicios de software de una empresa, que el comité del grupo de usuarios sugiera).

Definir una línea de tiempo clara

Aunque la implementación de SaaS es un procedimiento relativamente rápido y directo, una empresa aún debe definir una línea de tiempo clara para su procedimiento de implementación real (podrían usar una estructura de desglose del trabajo para hacer esto, por ejemplo). Una empresa debe definir claramente las fechas de los hitos, así como las fechas en las que las tareas importantes deben completarse, de modo que puedan evitar ser desviadas al implementar SaaS y realizar los cambios necesarios en su infraestructura existente.

Configurar la solución SaaS para satisfacer sus necesidades específicas

Una vez que una empresa ha implementado sus servicios de software SaaS, querrán configurar estos servicios de software para satisfacer sus propias necesidades específicas (del mismo modo que configurarían cualquier otra aplicación de software o tecnología). Una empresa también puede necesitar realizar configuraciones en su infraestructura existente antes de implementar realmente sus nuevos servicios de software SaaS.

Algunas de estas configuraciones y cambios pueden consistir en cambios basados en hardware, cambios basados en la red o incluso cambios basados en software. Independientemente de los cambios y las configuraciones que una empresa deba realizar en su infraestructura, tanto antes como después de la implementación de SaaS, siempre deben planificar estos cambios de antemano mediante la realización de investigaciones para evitar problemas imprevistos.

En una nota final, una vez más, no es la implementación real de SaaS lo que es un desafío para las empresas (los servicios de software SaaS que van a utilizar ya están en su lugar y listos para usar). En cambio, los desafíos a los que se enfrenta una empresa consisten en asegurarse de que los servicios de software SaaS que van a implementar sean los servicios de software adecuados para ellos y se integren fácilmente en su infraestructura comercial existente. Si este no es el caso, entonces como se detalló anteriormente, una empresa debe hacer los cambios necesarios, ya sea a su propia infraestructura existente o a sus nuevos servicios de software SaaS (utilizando el método más práctico y el más rentable).

4. Adoptando prácticas sólidas de seguridad de datos

Una vez que se implementó SaaS, no significa que el trabajo de una empresa se detenga allí. La seguridad es lo siguiente en lo que una empresa debería pensar.

Aunque la seguridad de los servicios de software de una empresa y los datos que utilizan estos servicios de software están en manos de un proveedor de servicios de SaaS empresarial, esto no significa que una empresa no tenga ningún poder en lo que respecta a la seguridad.

Algunas de las cosas que una empresa puede hacer para garantizar que sus servicios de software y datos sean seguros incluyen garantizar que su proveedor de servicios SaaS les brinde una garantía de protección de datos y garantizar que su proveedor de servicios SaaS cumpla con los estándares de la industria. Las empresas también deben analizar los antecedentes de sus proveedores de servicios SaaS y averiguar qué niveles de seguridad tienen para proteger sus centros de datos, qué procedimientos de respaldo y recuperación de desastres tiene, si es que los hay y cómo se aseguran cada uno de ellos. Los datos de su cliente.

Finalmente, una empresa debe definir claramente los roles y los derechos de acceso dentro de su propia organización (de nuevo, este es un proceso que también se llevaría a

cabo en las instalaciones soluciones de software). Por ejemplo, ningún usuario sin las credenciales correctas debería poder acceder a más de lo que tiene acceso permitido. Los usuarios de estaciones de trabajo solo pueden tener acceso a versiones restringidas de un servicio de software, mientras que los administradores tendrían acceso a las versiones completas de estos servicios de software.

5. Adoptando una estructura de soporte sólida

La última tarea que una empresa desea hacer al implementar SaaS es adoptar una sólida estructura de soporte para sus servicios SaaS. Por ejemplo, una empresa querrá decidir cómo van a admitir sus servicios de software SaaS, qué empleados soportarán estos servicios de software SaaS (es decir, empleados existentes o tendrá que incorporar profesionales de TI adicionales) y cómo se integrarán estos servicios de SaaS con software de proveedores de software de terceros, por ejemplo

Un error que las empresas pueden cometer fácilmente al adoptar SaaS es pensar que ya no tienen que llevar a cabo el soporte de software y los procedimientos relacionados con el mantenimiento. Sin embargo, esto no es del todo cierto. Aunque los proveedores de servicios de SaaS llevarán a cabo procedimientos de actualización de software, procedimientos de gestión de datos y tareas relacionadas con la asistencia, el usuario final de estos servicios de software aún debe asegurarse de que brinden soporte adicional para estos servicios de software, para asegurarse que estos servicios de software satisfacen sus necesidades exactas (en lugar de cumplirlos vagamente).

Finalmente, las empresas que implementan SaaS siempre deben tener en mente un segundo o tercer proveedor de servicios SaaS. La razón de esto es simplemente porque si se producen cambios en su proveedor de servicio SaaS, también se verán afectados por estos cambios. El mercado de software SaaS aún está madurando y es muy probable que un proveedor de servicios SaaS pueda fusionarse con otra empresa o incluso ser asumido completamente. Teniendo en cuenta un segundo o tercer proveedor de servicios SaaS, las empresas están dispuestas a cambiar de proveedores si se producen cambios en su proveedor de servicios existente (hasta el punto en que su proveedor de servicios SaaS ya no puede ofrecer sus servicios de software como antes).

6. BIBLIOGRAFIA

1. Papazoglou, M.P. y Georgakopoulos, D., Service – Oriented Computing, in Communications of the ACM. 2003. p. 25-28-
<http://infolab.uvt.nl/pub/papazogloump-2003-52.pdf>.
2. Curbera, F., Khalaf, R., Mukhi, N., Tai, S. y Weerawarana, S., The Next Step in Web Services, in Communications of the ACM. 2003. pp. 29-34.
3. Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia (2009). Above the Clouds: A Berkeley View of Cloud Computing. EECS Department, University of California, Berkeley.
4. Leimeister, Stefanie, fortiss, Riedl, Christoph, Böhm, Markus, Krcmar, Helmut- Technische Universität München (TUM), the business perspective of cloud computing: actors, roles, and value networks-2010-Scient Direct.
5. David S. Linthicum. Cloud Computing and SOA Convergence in Your Enterprise. 2009. p. 9-10.
- 6.