



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**PROPUESTA DE UN MODELO PERSONALIZABLE DE
DESARROLLO, PARA MEJORAR LA CONSTRUCCIÓN Y
CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES
ORGANIZACIONES CON ÁREA DE TI EN GUATEMALA**

Helmerson David Oliva

Asesorado por: Inga. Victoria Isabel Girón Ruano

Guatemala, noviembre de 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROPUESTA DE UN MODELO PERSONALIZABLE DE
DESARROLLO, PARA MEJORAR LA CONSTRUCCIÓN Y
CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES
ORGANIZACIONES CON ÁREA DE TI EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

HELMERSON DAVID OLIVA

ASESORADO POR: INGA. VICTORIA ISABEL GIRÓN RUANO

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2008

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Angel Dávila Calderón
VOCAL IV	Br. Milton de León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Freiry Javier Gramajo López
EXAMINADOR	Ing. Alvaro Navarro Figueroa
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Ingá. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROPUESTA DE UN MODELO PERSONALIZABLE DE DESARROLLO, PARA MEJORAR LA CONSTRUCCIÓN Y CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES ORGANIZACIONES CON ÁREA DE TI EN GUATEMALA,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2007.

A handwritten signature in black ink, appearing to read 'Helmersen David Oliva', enclosed within a large, irregular oval scribble.

Helmersen David Oliva .



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS

Ref: ASESOR 02-02

Guatemala 29 de Julio de 2008

Señores
Comisión de Revisión de Tesis
Carrera de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Respetables Señores:

El motivo de la presente es informarles que como asesor del estudiante HELMERSON DAVID OLIVA he procedido a revisar el trabajo de tesis titulado *"Propuesta de un modelo personalizable de desarrollo para mejorar la construcción y calidad del software de medianas y grandes organizaciones con área de TI en Guatemala"* y que de acuerdo a mi criterio el mismo se encuentra concluido y cumple con los objetivos definidos al inicio.

He tenido reuniones periódicas con el estudiante y luego de haber revisado cuidadosamente el trabajo, considero que cumple con los requisitos de calidad y profesionalismo que deben caracterizar a un futuro profesional de la Informática.

Aprovecho para informarle que he leído detenidamente el documento Ref: ASESOR 01-02 y aplicando las recomendaciones que se dan en el mismo procedo a firmar de revisado el trabajo de tesis.

Sin otro particular me suscribo de ustedes,

Atentamente,

Inga. Victoria Isabel Girón Ruano

Victoria Isabel Girón Ruano
Ingeniera en Ciencias y Sistemas
Cc: 7215



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 15 de Octubre de 2008

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante HELMERSON DAVID OLIVA titulado: "PROPUESTA DE UN MODELO PERSONALIZABLE DE DESARROLLO PARA MEJORAR LA CONSTRUCCION Y CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES ORGANIZACIONES CON AREA DE TI EN GUATEMALA", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme.

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Prácticas
y Revisión de Trabajos de Graduación



E
S
C
U
L
A
D
E

C
I
E
N
C
I
A
S

Y
S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingenieria en Ciencias y Sistemas de la Facultad de Ingenieria de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **"PROPUESTA DE UN MODELO PERSONALIZABLE DE DESARROLLO PARA MEJORAR LA CONSTRUCCION Y CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES ORGANIZACIONES CON ÁREA DE TI EN GUATEMALA."**, presentado por el estudiante **HELMERSON DAVID OLIVA**, aprueba el presente trabajo y solicita la autorización del mismo.

"ID Y ENSEÑAD A TODOS"


Ing. Maximiliano Antonio Pérez Turk
Director, Escuela de Ingeniería Ciencias y Sistemas



Guatemala, 03 de octubre 2008

Universidad de San Carlos
de Guatemala



Facultad de Ingeniería
Decanato

Ref. DTG.374.08

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **PROPUESTA DE UN MODELO PERSONALIZABLE DE DESARROLLO, PARA MEJORAR LA CONSTRUCCIÓN Y CALIDAD DEL SOFTWARE DE MEDIANAS Y GRANDES ORGANIZACIONES CON ÁREA DE TI EN GUATEMALA**, presentado por el estudiante universitario **Helmerson David Oliva**, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Ocampo Paiz Recinos
DECANO



Guatemala, Noviembre de 2008

/cc
c.c. archivo.

ACTO QUE DEDICO A:

Dios, por ser la piedra angular de mi existir.

María Auxiliadora, por cubrirme siempre con su santo manto.

Marta Julia Oliva, por formar parte esencial de mi vida y ser fuente de mi aprendizaje.

Agradezco especialmente a:

Mi familia, por tener siempre confianza en mí.

Mis amigos, por acompañarme en las buenas y en las no tan buenas.

Victoria Girón, por su ayuda desinteresada en este proyecto de tesis.

“Nunca dejes de soñar, porque sin sueños no habrá metas que cumplir.”

ÍNDICE GENERAL

ÍNDICE ILUSTRACIONES	V
GLOSARIO	VII
RESUMEN	XI
OBJETIVOS	XIII
INTRODUCCIÓN	XV
1 MARCO TEÓRICO Y SITUACIÓN ACTUAL	1
1.1 TI en las organizaciones	1
1.1.1 Impacto actual	2
1.1.2 CIO como rol de apoyo en el negocio.....	3
1.1.3 Administración y gestión de proyectos de software	4
1.1.4 Resultados actuales	5
1.2 Metodología de desarrollo	6
1.3 Elementos de desarrollo	7
1.3.1 Programadores, desarrolladores y analistas.....	7
1.3.2 Equipo y recursos	9
1.3.3 Tecnología	9
1.4 Ambiente de desarrollo	10
2 MODELO DE DESARROLLO DE PROYECTOS DE SOFTWARE PARA ORGANIZACIONES	13
2.1 Definición del modelo.....	13
2.1.1 Elementos.....	14
2.1.2 Relaciones entre elementos	16
2.2 Gestión de procesos	18

2.3	Estandarización de elementos de desarrollo	21
2.3.1	Metodología de desarrollo	21
2.3.2	Estándares de arquitectura y de tecnologías	22
2.3.3	Estándares de nombrado y construcción de proyectos	23
2.3.4	Estándares de codificación.....	24
2.3.5	Estándares de control de versiones	25
2.3.6	Estándares de interfaces.....	26
2.3.7	Estándares de nombrado de base de datos.....	27
2.3.8	Estándares de documentación	27
2.4	Control de cambios.....	28
2.4.1	Repositorio de control de cambios para activos	33
2.4.2	Repositorio de componentes y releases.....	35
2.5	Gestión de conocimiento	37
3	PERSONALIZACIÓN DEL MODELO	41
3.1	Identificación de los atributos de la organización (Ej. SAT).....	41
3.2	Personalización de los elementos del modelo.....	42
3.2.1	Personalizar metodología de desarrollo	42
3.2.2	Personalizar estándares de arquitectura y componentes.....	45
3.2.3	Personalizar estándares de codificación	53
3.2.4	Personalizar estándares de nombrado y construcción de proyectos	54
3.2.5	Personalizar estándares de control de versiones	54
3.2.6	Personalizar estándares de interfaces	55
3.2.7	Personalizar estándares de base de datos	56
3.2.8	Personalizar estándares de documentación.....	57
4	INTEGRACIÓN Y UTILIZACIÓN DEL MODELO.....	59
4.1	Configuración e Implementación de los Elementos (SAT)	59
4.2	Publicación de estándares de elementos y plantillas del modelo	63

4.3	Integración con un proyecto real.....	64
4.4	Observaciones acerca de la integración y utilización del modelo	67
CONCLUSIONES		71
RECOMENDACIONES.....		75
BIBLIOGRAFÍA		79

ÍNDICE ILUSTRACIONES

FIGURAS

1. Elementos del modelo	15
2. Interacción de los elementos del modelo	17
3. Procedimiento de desarrollo	19
4. Administración del cambio	32
5. Control de activos	35
6. Proceso de creación de conocimiento	38
7. Personalización de la metodología de desarrollo en SAT	44
8. Arquitectura física de una organización	46
9. Arquitectura básica de aplicaciones web	47
10. Arquitectura de tecnologías y componentes	48
11. Dominio de tecnologías para las capas	49
12. Diagrama de componentes	51
13. Documentación acerca de una tecnología	52
14. Documentación acerca de una tecnología (II)	52
15. Estándares de codificación	53
16. Estándares de distribución de elementos en una interfaz	55
17. Estándares de interfaz para catálogos	56
18. Ejemplo de nombrado de objetos en un modelo de datos	57
19. Especificación de las guías o documentación	58
20. Flujo de integración y despliegue	60
21. Repositorio de proyectos	62
22. Repositorio de componentes	63
23. Publicaciones de proyectos	64

24. Matrícula de un proyecto 67
25. Softwareteca de desarrollo 69
26. Gestión de contenidos 70

GLOSARIO

Arquitectura de Software	Establece los lineamientos y fundamentos para que un equipo de desarrollo construya y trabaje en un camino común, que permita alcanzar los objetivos del sistema de información correcto.
Cluster	Conjunto de nodos de computadoras construidos mediante la utilización de componentes de hardware comunes y la aplicación de un software especializado y que se comportan como si fuesen una única computadora o bien vistos como una capa de software única.
DRA	Metodología de Desarrollo Rápido de Aplicaciones, que propone métodos para agilizar la construcción de componentes de software.
Framework	Conjunto de clases y componentes con diseño abstracto para solucionar un problema o implementar una herramienta que provee varias soluciones.
Hardware	Se denomina hardware o soporte físico al conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.
Hibernate	Framework java para el manejo de la persistencia de la data.
IDE	Aplicación o editor que provee una serie de facilidades para que un desarrollador realice código fuente y/o aplicaciones en un lenguaje determinado.

Interconectividad	Proceso de comunicación, el cual ocurre entre dos o más redes que están conectadas entre sí de alguna manera.
Interface	Representa una forma de comunicación correcta entre dos aplicaciones o contextos distintos.
Internet	Red de redes de grado mundial de millones de ordenadores interconectadas a través de un conjunto de protocolos TCP/IP. Dicha telaraña es parte fundamental para el funcionamiento de millones de aplicaciones, constituyendo uno de los activos y plataforma que nos provee de una gran ventaja para expandir, innovar y mejorar nuestras organizaciones.
Intranet	Red de Área Local (LAN) privada empresarial o bien educativa, que brinda una serie de herramientas a través de Internet, las cuales tienen como propósito principal suministrar lógica de negocios y servicios para proveer documentos, aplicaciones, consultas de información, reportes, etc. con el fin de auxiliar a los diferentes equipos de trabajo.
J2EE	Frameworks, patrones y métodos java para aplicaciones empresariales.
JSF	Framework para el manejo de la interfaz web bajo plataforma java.
Log4j	Librerías para el manejo de bitácoras para java.
Maven	Framework J2EE para el manejo y distribución de componentes.
Página Web	Documento situado en una red informática, al que se accede mediante enlaces de hipertexto.
Prototipo	Pequeño esquema que simboliza en cierta magnitud a un proyecto real y que representa alguna característica del sistema que se desea modelar.

Release	Conjunto de activos que componen un software como un todo.
RUP	Metodología de desarrollo de software predictiva propuesta por Rational.
Scrum	Metodología ágil de desarrollo orientada a equipos de trabajo.
Servidor de Aplicaciones	Servidor que provee de un contenedor y de un ambiente de ejecución de las aplicaciones de software realizadas bajo cierta tecnología.
Sharepoint	Software de Microsoft para manejo de contenidos y gestión documental.
Software	Software es la parte lógica del ordenador, es el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina.
Subversión	Framework open source para control de versiones.
UML	Estándar que provee de una serie de diagramas, elementos y relaciones con los cuales puede modelarse un sistema desde varias perspectivas.
Usabilidad	Atributo de calidad o requerimiento no funcional de las aplicaciones. Básicamente mide la forma en que una pantalla o interfaz es altamente usable.
Wiki	Sitio que contiene información y documentación acerca de cierto tópico o tema específico.
Workflow	Flujo de trabajo que define una serie de pasos y actividades que conllevan a un objetivo.
XP	Xtreme Programming, metodología ágil de desarrollo de software.

RESUMEN

Es interesante observar cómo, día con día, a través de la tecnología, crecemos, cambiamos e innovamos en nuestra forma de vivir. Vemos como también, gracias a ella, se puede mejorar nuestras actividades cotidianas, por lo que es un factor a tomar muy en cuenta a la hora de ofrecer un bien o servicio. Los servicios electrónicos, conforme crece la economía obtienen mayor demanda y necesitan ser más eficientes conforme el tiempo. Por tal razón, con el tiempo, son más los recursos y el interés que adquieren las áreas o departamentos de TI, además que crece la expectativa de poseer un sistema de información robusto y confiable.

Actualmente en nuestro país existen numerosas organizaciones que implementaron todo su núcleo en sistemas de información y en una plataforma que los soporta, como es el caso de instituciones gubernamentales como SAT, Organismo Judicial, IGSS, etc., además de otro tipo de instituciones tales como los bancos del sistema y otras organizaciones.

Es evidente que dichas organizaciones, por el tiempo y la razón de cambio tecnológica, han crecido de una forma personalizada y luchan cotidianamente con una serie de problemas en donde se obtienen soluciones que posiblemente reparan el inconveniente por ese momento, pero no atacan muchos problemas de raíz. Ya que muchas veces esos problemas son muy invisibles a los ojos de un programador o un administrador de proyectos que no tiene conciencia de todo lo que conlleva un proceso de TI.

Por tal necesidades e inconvenientes observados, se define un modelo que consta de cuatro elementos principales interrelacionados que juntos proveen de varios mecanismos, guías y proponen una mejor forma de trabajar, ajustándose a las necesidades y características de las organizaciones y áreas de TI, aplicándose principalmente en el área o departamento de desarrollo y mantenimiento de software.

Este modelo provee de una serie de lineamientos para su implantación, sugiriendo detalles para su implementación a través de etapas. Dicho modelo pretende principalmente la mejora continua de las actividades diarias, elevar la calidad del personal, madurar los procesos e interacciones con otros departamentos, elevar el capital intelectual y promover el aprendizaje en todo el personal involucrado.

Esto fue aplicado a la institución elegida para su estudio (SAT), consiguiendo un resultado favorable en varias iteraciones de desarrollo en proyectos designados para su implantación, consiguiendo una mejor organización, distribución del trabajo, mejoras en cuanto a tiempos, aprendizaje y distribución del conocimiento, madurez en cuanto a las aplicaciones y procesos de desarrollo y mantenimiento. Parte del éxito fue el apoyo de las unidades de mando y una implementación elaborada a través de varias etapas. Aunque es muy evidente que para alcanzar el nivel deseado (óptimo), es necesario pasar por una serie de iteraciones y contar con una diversidad de escenarios y tipos de sistemas de información. Por lo que la recomendación es implantarlo por fases y luego tener una tarea de crecimiento y maduración del modelo.

OBJETIVOS

General:

Proporcionar un método bien definido que sirva para la mejora, ayuda y orientación para los procesos de administración, configuración y desarrollo de los sistemas y componentes de software de las organizaciones guatemaltecas con recursos asignados a la Tecnología de Información y que a la vez tal modelo reduzca la necesidad y las dependencias que se acumulan a lo largo del proceso de construcción de software que ocurre en el ámbito de la gestión de proyectos, respecto de los recursos, tecnología y programadores.

Específicos:

1. Establecer los principales conceptos y problemas propios de las organizaciones que cuentan con servicios internos y externos de desarrollo de sistemas de software, que constituyen una de las líneas de actividades más prometedoras hoy en día, con respecto a las demás actividades administrativas y de gestión de cada organización.
2. Presentar un bosquejo de los diversos procedimientos y de los diferentes elementos que constituyen la base general de desarrollo de software en las organizaciones, tomando como referencia la organización elegida para el estudio (SAT).

3. Proponer un modelo que provea de la metodología y terminología apropiada que plantee los lineamientos a seguir para la elaboración y construcción de todo sistema de software que caracterice de manera única a la organización, al igual que los elementos que se deben definir y las relaciones entre ellos para mejorar la calidad y soporte del mismo.
4. Plantear un mecanismo de personalización de la metodología y modelo propuesto para que se construya la base de desarrollo para una organización determinada.
5. Evaluar la utilización del modelo y examinar su integración con los métodos de desarrollo de una organización (SAT), y exponer las implicaciones que conlleva su uso.

INTRODUCCIÓN

Las organizaciones actualmente tienen una creciente demanda de mejorar y optimizar los procesos de negocio y la infraestructura tecnológica que los soporta. Por tal razón, se ha concentrado mucho esfuerzo en dichas organizaciones hacia las áreas o departamentos de tecnología (TI) o informática, como comúnmente se le conoce. Dicho esfuerzo va encaminado a ofrecer una mayor eficiencia y calidad en los servicios y aplicativos que se tengan para soportar toda la gama de servicios prestados por las organizaciones.

Dichos departamentos, áreas o gerencias tienen una misión bien establecida y junto con ella, en las medianas y grandes organizaciones, comparten la función de producir y mantener los sistemas de software que soportan las necesidades de las diferentes unidades de negocio que componen una organización o empresa.

Por lo tanto, es muy interesante denotar los problemas comunes que existen y que presentan un impedimento para la mejora continua de dichas áreas en las organizaciones, tomando en cuenta que un retardo u obstáculo en la construcción y mejora de la calidad de los sistemas, afectará e influenciará los servicios prestado por las organizaciones y del mismo modo, puede presentar una deficiencia e inconveniente para los objetivos y metas de la misma.

Además, existen varios tipos de problemas atados a un mal método de administración de un departamento o área de desarrollo de software, que son

un poco más difíciles de percibir a corto plazo, pero que son evidenciados con el paso del tiempo y cambios del personal o equipos de dichos departamentos. Creando varios inconvenientes y trabajo extra para la serie de tareas y actividades a realizarse cotidianamente en el área o departamento. Proveyendo de un deficiente servicio que afecta directamente el desempeño de la organización, ya que ésta, la mayoría de ocasiones, toma acciones correctivas a estos retrasos y problemas, en vez de atacar las raíces del problema: el modelo de desarrollo utilizado por el departamento.

Este modelo debe ser capaz de propiciar, con el tiempo, un alto desempeño en las actividades y una especie de mecanismo de auto aprendizaje y mejora continua de todo el personal y procesos que ejecuten los mismos, y además, que elimine con el tiempo, la serie de problemas que se acarrearán cotidianamente, debido a la forma de organizar y administrar los proyectos y sistemas de soporte a los servicios de una organización.

1 MARCO TEÓRICO Y SITUACIÓN ACTUAL

1.1 TI en las organizaciones

Hoy en día existe gran variedad de aplicaciones tecnológicas en las que la computadora juega un papel fundamental, tanto en las áreas científica, educativa, como en los negocios de servicios, producción y de entretenimiento.

La mayoría de organizaciones evalúan la necesidad de automatizar todos su procesos, llevando a la construcción de un área básica y fundamental especialista en tecnología en toda organización, como lo es el área de TI, o de informática como se conoce actualmente, es muy necesario contar con una especie de guía que nos indique que debemos tomar en cuenta para la construcción de dicho departamento o área en nuestra organización, bajo la premisa de querer tener los productos o sistemas de mejor calidad y correcta funcionalidad.

Los aspectos anteriormente mencionados han planteado nuevos retos a resolver, como es el desarrollo de sistemas de alta calidad y mantenibilidad, actualización y cambios de los sistemas legados y que todo esto desliguen de las restricciones atadas a los creadores (programadores/desarrolladores), recursos, tecnologías, etc.

Es necesario también mencionar que la gestión de todos estos recursos debe estar atada o fundamentada en los lineamientos o estrategias institucionales de la organización, es decir, que las políticas, normas y

procedimientos del área de TI deben ir orientadas a las estrategias institucionales planteadas y no que vayan por su propio camino.

1.1.1 Impacto actual

El cambio tecnológico tiene un impacto tan importante en la evolución de los modelos de negocio, estrategias institucionales, los mercados y la sociedad misma. Éste a su vez, debe gestionarse adecuadamente y estar consciente de la tecnología y los sistemas de información que la organización necesita convirtiéndose esto en la clave para la competitividad empresarial. Por lo que es evidente que para tratar este tema debemos tratar de entender y estructurar la relación entre estrategia de empresa y tecnología de la información (TI). La función de gestionar la tecnología y entender el valor de los Sistemas de información para la empresa se ha convertido en algo crucial y muy importante para mantener la competitividad en cualquier sector.

Todo directivo debe ser capaz de entender la influencia que la tecnología tiene para la situación competitiva de su negocio u organización y, al tiempo, tomar decisiones sobre su impacto actual y futuro en la cuenta de resultados. La situación de cambio permanente y de una economía global cada vez más competitiva donde asistimos a un continuo desarrollo de las nuevas tecnologías de la información, ha hecho que la Integración de Sistemas sea un importante activo a gestionar por las organizaciones.¹

¹. URL(<http://www.postgradum.com/cursos/visualizar.phtml?codigo=6247¢ro=P78765>)

1.1.2 CIO como rol de apoyo en el negocio

El rol del CIO en el negocio en general, debe representar una articulación entre las unidades de negocios de una organización en general y la tecnología.

El CIO debe trabajar con los líderes de las unidades de negocios de la organización, estableciendo junto con ellos las metas y estrategias a seguir para alcanzar los objetivos de la organización, jugando un papel muy importante, ya que deberá poner la tecnología como un medio para consolidar las estrategias planteadas, al igual que deberá proponer con base en la tecnología, prácticas o métodos nuevos que mejoren o innoven los procesos y estrategias de las distintas unidades, o bien, que propongan nuevas estrategias para alcanzar las metas propuestas, todo esto, siempre visto desde una perspectiva global, tomando muy en cuenta toda la arquitectura del negocio mapeada a la arquitectura tecnológica que la soporta, teniendo en cuenta toda la información que fluye a través de ella.

Además, el CIO está comprometido a responder a los cambios en los distintos ciclos de negocios que puedan darse a lo largo del tiempo (cambios de estrategia, por ejemplo la diversificación de productos), estableciendo los puntos críticos y los puntos más relevantes, al momento de realizar cualquier cambio en las actividades, estrategias o de arquitectura del negocio. Contando con un panorama general de la arquitectura tecnológica, a la vez, que gracias a ello, podrá notar en donde será necesario tener más precaución al realizar los cambios, o bien, realizarlos de manera más minuciosa, tomando muy en cuenta las repercusiones de dichos cambios en el cumplimiento de las metas establecidas.

En algunas organizaciones en donde existe una gran demanda de apoyo tecnológico, el CIO juega un papel muy importante ya que dependiendo de los recursos con los que se cuenta, en ocasiones es necesario priorizarlos, distribuir la infraestructura con la que se cuenta y proveer de soluciones estratégicas de TI, que soporte las diferentes estrategias planteadas por las unidades respectivas para que se acoplen y unifiquen de manera integra a la arquitectura global de la organización, teniendo un estricto control en el rendimiento de cada una de dichas soluciones.

1.1.3 Administración y gestión de proyectos de software

La gestión de proyectos es el proceso por el cual se planifica, dirige y controla el desarrollo de un sistema aceptable con un costo mínimo y dentro de un período de tiempo específico². Es importante que denotemos las características de todos los proyectos tienen en común:

- Todos los proyectos tienen alguna actividad que les da inicio y una entrega que marca un fin, el cual puede ser definitivo o generar una serie de cambios futuros (mantenimiento y cambios del proyecto).
- Todo proyecto es único. Son únicos en términos de sus requerimientos, desarrollo, arquitectura, componentes, despliegue, etc.
- Cada proyecto cuenta con recursos definidos, con disponibilidad de tiempo parcial o completo.
- La entrega de final del proyecto resulta siempre en varios entregables.

². URL(<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/gestioproyecto.htm>)

Existen también otras características, tales como: todas las organizaciones tienen proyectos, los proyectos pueden ser manejados usando un conjunto común de procesos. De hecho, este conjunto común puede ser usado incluso para gestionar proyectos sin importar el tipo de proyecto. Por ejemplo, todos los proyectos deben ser definidos y planeados y deben tener procesos para gestionar el alcance, los riesgos, la calidad, el estatus, etc. La administración de proyectos define los procesos generales de gestión y control para el proyecto. La administración del proyecto, de hecho no tiene como resultado la creación de los entregables del mismo. En algún punto, es todavía necesario definir las actividades que realmente son necesarias para construir los entregables específicos. Estas actividades colectivas asociadas con la creación de entregables del proyecto son conocidas como el ciclo de vida del proyecto.

De la misma forma en que hay procesos comunes para gestionar la mayoría de los proyectos, también existen modelos comunes que pueden ser guía para definir el ciclo de vida del proyecto. Estos modelos comunes son valiosos puesto que ahorran tiempo a los equipos de proyecto en la creación del plan de trabajo desde cero cada vez que están por iniciar un proyecto.³

1.1.4 Resultados actuales

En general, un producto de software es actualmente, dentro del contexto de cualquier sistema de información, el componente cuyo desarrollo presenta mayores inconvenientes, ya que es mucho más difícil de planificar, tiene mayor probabilidad de fracaso, y más riesgos de que no se cumplan las estimaciones de costos inicialmente propuestos. Y agregando que la demanda de productos

³ URL(<http://www.tenstep.com.mx/Paso0.0.1.2.asp>)

de software aumenta constantemente, nos hace evaluar la magnitud de la importancia de realizar una excelente gestión que incurre en el desarrollo de cualquier sistema o producto de software.

Hoy en día, la mayoría de proyectos posee una serie de problemas atados a su desarrollo, tales como falta de tiempo, mala asignación de recursos, elevación de costos, etc., que incurren en que solamente un 15% de los proyectos son exitosos, un 45% de los proyectos tienen problemas y extienden las fechas de entrega pero son terminados y un 40% de los proyectos fracasan, brindándonos una idea clara de la importancia que para la gestión y el desarrollo, es el elegir un buen modelo de desarrollo para nuestros proyectos o productos de software. Para corregir ello debemos adecuar o incluir dentro de los procesos de software de nuestra organización aquellas mejores prácticas que han demostrado en otros países la mejora de sus productos.

1.2 Metodología de desarrollo

La metodología desarrollo es una estrategia gerencial que orienta el desarrollo del proyecto hacia un conjunto de resultados o entregables (producto final) que mejor satisfagan los requerimientos del problema dado.

Una metodología de desarrollo brinda una aproximación metodológica que será definida según el enfoque de modelado, el cual es utilizado para describir la realidad del problema, valiéndose para ello de:

- Notación / técnica : conjunto de diagramas utilizados para describir el sistema
- Método : proceso disciplinado para generar un modelo, utilizando una notación bien definida

Entre las metodologías de desarrollo más utilizadas se encuentran:

- Cascada / Lineal
- Prototipado
- DRA (Rapid Application Development)
- Incremental
- Desarrollo basado en componentes
- Rational Unify Process - RUP (Rational)
- XP (Extreme Programming)
- Scrum

Es muy importante la elección o adopción de la metodología de desarrollo, ya que ésta debe contemplar dentro de sus funciones y proceso, el entorno en el cual se desenvuelve la institución, por ejemplo, si nuestra organización requiere de software funcional de alta calidad en tiempos cortos y con requerimientos cambiantes, o bien se requiere software complejo de alta calidad con lineamientos fuertes, etc.

1.3 Elementos de desarrollo

Un elemento de desarrollo es todo ente que contribuye moderadamente a la concepción, elaboración y construcción de un proyecto de software.

1.3.1 Programadores, desarrolladores y analistas

El desarrollo de software es una actividad que, dada su complejidad, debe desarrollarse en grupo. Además, esta actividad requiere de distintas

capacidades, las que no se encuentran todas en una sola persona. Por ello, se hace necesario formar el grupo de desarrollo con las personas que cubran todas las capacidades requeridas. Cada una de esas personas aportará al grupo parte del total de las capacidades necesarias para llevar a cabo con éxito el desarrollo.

En Guatemala, tradicionalmente en el área de informática, se identifican varios roles en lo que respecta al desarrollo de un proyecto, pero haciendo un sondeo y definición de esos roles y responsabilidades, solamente existen tres definiciones: programadores, desarrolladores y analistas de sistemas.

Los programadores y desarrolladores deben convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación.

Gran parte del éxito del desarrollo de software depende del conocimiento. Este conocimiento no sólo corresponde a habilidades de programación y de administración de proyectos, sino que a una percepción y entendimiento de los últimos desarrollos de la industria del software.

Respecto a los analistas, la palabra “análisis” se refiere a una característica típicamente relacionada con la inteligencia humana. Esta se refiere a la habilidad de estudiar un problema con una complejidad determinada, descomponiendo el problema en sub-problemas de menor complejidad.

De esa forma, la solución del problema completo se obtiene como la suma de las soluciones de los sub-problemas de menor complejidad. Como el experto en el problema es el cliente, se hace necesario trabajar junto a él para realizar la especificación correctamente. Los miembros del grupo que trabajan con el

cliente para realizar el análisis y especificación del sistema a construir son precisamente los analistas.

1.3.2 Equipo y recursos

Los recursos para el desarrollo de un software son todas aquellas necesidades identificadas a raíz de la solución definida para un problema o requerimiento dado. Por lo que se debe de tomar muy en cuenta todo aspecto, relacionado con las condiciones presupuestarias o limitaciones económicas y que englobe el desarrollo del proyecto. Estas pueden referirse, por ejemplo, a servidores, ordenadores, herramientas de software, costos atados al equipo de desarrollo, gastos varios, licencias, etc.

1.3.3 Tecnología

Es necesario tener conocimiento de todas las posibilidades tecnológicas que existen, para el desarrollo de todo proyecto, debido a la importancia de elección de la tecnología que mejor se acople a la solución de un problema, requerimiento o necesidad específica.

El desarrollo de sistemas de información y aplicaciones empresariales en general, la colaboración, tanto entre las diferentes unidades de negocios como entre organizaciones, y el desarrollo de aplicaciones y sistemas Web, han sufrido un apogeo muy importante en tanto los últimos años. Cara a este suceso existen dos grandes plataformas muy distintas para el desarrollo de este tipo de aplicaciones, con requerimientos de alto nivel: J2EE de Sun y .NET de Microsoft. J2EE y .NET en contraste con los demás lenguajes de

programación, ofrecen fundamentalmente las mismas funcionalidades aunque de diferente manera.

1.4 Ambiente de desarrollo

Es muy necesario antes de empezar todo desarrollo de un proyecto, especificar o determinar el ambiente de desarrollo o de trabajo, ya que es necesario dar soporte al proyecto con las herramientas, procesos y métodos adecuados, brindando una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

1.5 Problemas propios de la gestión de proyectos en las organizaciones

Los problemas que más incurren en el desarrollo de proyectos tienen como consecuencia final, los retrasos, que van relacionados con la mayoría de errores siguientes:

- Fechas límites poco realistas
- Cambios en los requisitos del cliente que no se reflejan en cambios de la planificación temporal
- Fallas de comunicación y trabajo en equipo
- Esfuerzos mal aplicados
- Proyectos que se vuelven mucho más complejos conforme el tiempo
- Subestimación de los recursos necesarios para el trabajo
- Riesgos predecibles y no predecibles no considerados al comenzar el proyecto
- Dificultades técnicas o humanas no predecibles

- Falta de medidas correctivas oportunas ante el reconocimiento de retraso.

2 MODELO DE DESARROLLO DE PROYECTOS DE SOFTWARE PARA ORGANIZACIONES

2.1 Definición del modelo

Es notorio pensar que cada organización posee distintas formas para poder administrar los recursos asignados al área de TI y desarrollo de cada una, y de igual forma no todas las organizaciones o instituciones tienen que solventar las mismas necesidades e implementar las mismas estrategias, tal como es el caso de instituciones como la SAT (Superintendencia de Administración Tributaria), y el IGSS (Instituto Guatemalteco de Seguridad Social) en nuestro país de Guatemala, ya que los fines de cada una son distintos y la demanda de mejoras en el ámbito tecnológico son muy variadas, como es el caso particular del monitoreo y desempeño de los sistemas de control Aduaneros (SAT) o como el control de las planillas por parte de cada empresa (IGSS), ya que la primera requerirá de la mejor tecnología para estar disponible y confiable los 365 días del año, mientras que la segunda requerirá que sea realmente fácil y muy funcional.

Pensando en todas esas cosas es necesario tratar de definir un modelo que cubra todas estas diferencias organizacionales, que posea todos los elementos para que pueda soportar toda una infraestructura sólida en cualquier organización para poder administrar y gestionar con un buen desempeño y minimizar la serie de diversos problemas atados hoy en día al desarrollo de sistemas para dichas organizaciones y además que brinde aprendizaje al mismo proceso con la experiencia que se adquiriera continuamente.

2.1.1 Elementos

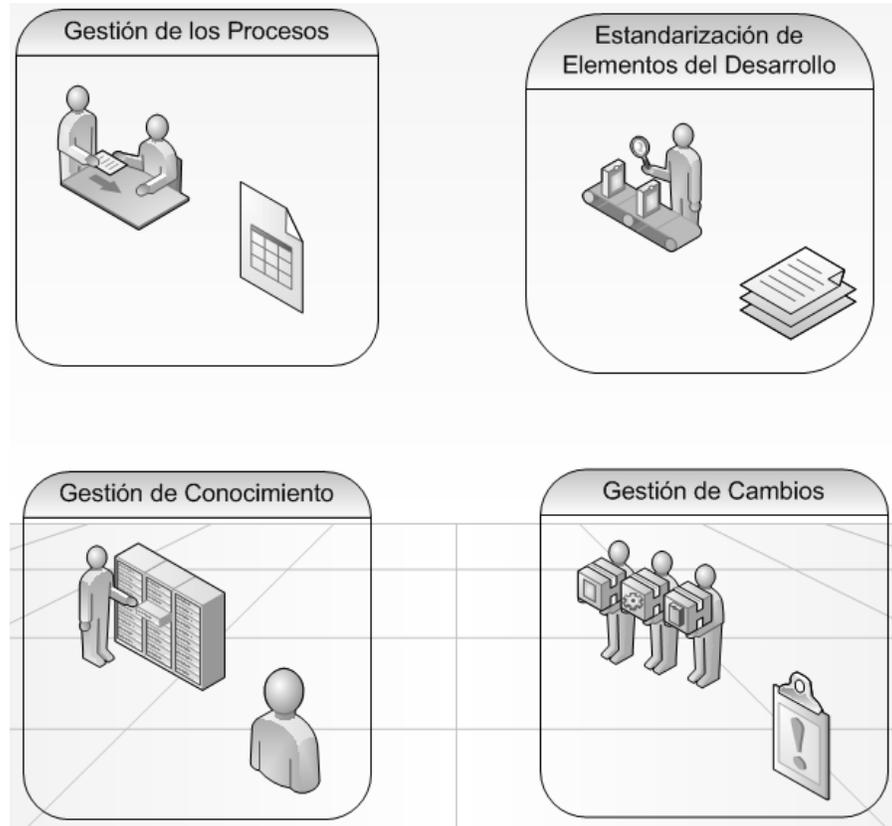
Para poder asimilar cualquier modelo, es necesario conocer la abstracción de más alto nivel para luego ir bajando poco a poco, hasta la especificación final de cada elemento.

Para empezar a definir estos elementos, es necesario hacer énfasis en la definición descrita anteriormente, de donde se determinaban una serie de necesidades y lineamientos base para garantizar los objetivos del área de TI en lo que respecta a desarrollo de sistemas de software.

Sin importar el tipo de institución, la forma de trabajar, objetivos y estrategias organizacionales, recursos y personal asignado, un área de TI debe cumplir con el objetivo de solventar las necesidades tecnológicas de una institución, brindar productos de alta calidad y además contar con el valioso punto de vista tecnológico en la toma de decisiones en la institución, por lo que agruparemos las necesidades a gestionarse en cuatro bloques principales del modelo:

- Gestión de procesos
- Estandarización de elementos del desarrollo de sistemas
- Gestión de los cambios
- Gestión de conocimiento

Figura 1. Elementos del modelo



La Gestión de Procesos es la capa o elemento de alto nivel que organiza todas interacciones entre el departamento o área de desarrollo de sistemas con las demás unidades, es decir que norma y ordena todas las solicitudes, pedidos, alertas y cualquier comunicación que exista entre lo que es desarrollo de sistemas y las demás unidades de la institución.

La estandarización del desarrollo de sistemas abarca los ítems, guías o lineamientos que debe seguir toda persona en el ámbito de desarrollo y en todos los procesos del mismo, esto para establecer un orden y un patrón en el desarrollo de cualquier sistema, lo cual beneficiará todo mantenimiento y soporte posterior.

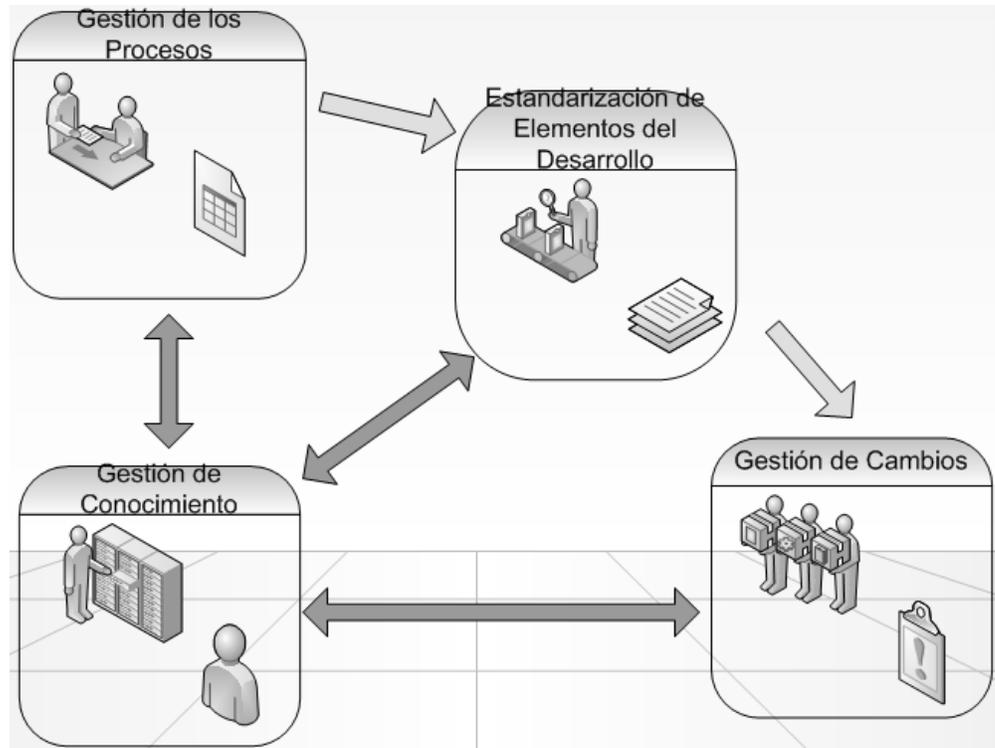
El control o gestión de los cambios en el área de desarrollo de sistemas debe facilitar y brindar toda introducción y seguimiento del software y hardware en un entorno de TI controlado.

La gestión de conocimiento administra y suministra el aprendizaje debido en todos los niveles del departamento o área de desarrollo de sistemas. Ésta es una necesidad latente hoy por hoy, por lo que esta parte del modelo planteado para el desarrollo debe ser extendido por todos los procesos de la organización, para lo que creará un ambiente propenso a mejoras e innovación, aumentando el valor de los activos en la institución (activos tangibles e intangibles).

2.1.2 Relaciones entre elementos

Como todo modelo, existen formas de aplicar cada elemento o es posible que cada elemento pueda tener una prioridad al momento de ser ejecutado para que su resultado sea exitoso, por lo que este modelo sugiere una jerarquía y ciclo ideal, con los cuales, se puede aplicar el modelo y adoptarse de mejor manera, tal y como se describe a continuación:

Figura 2. Interacción de los elementos del modelo

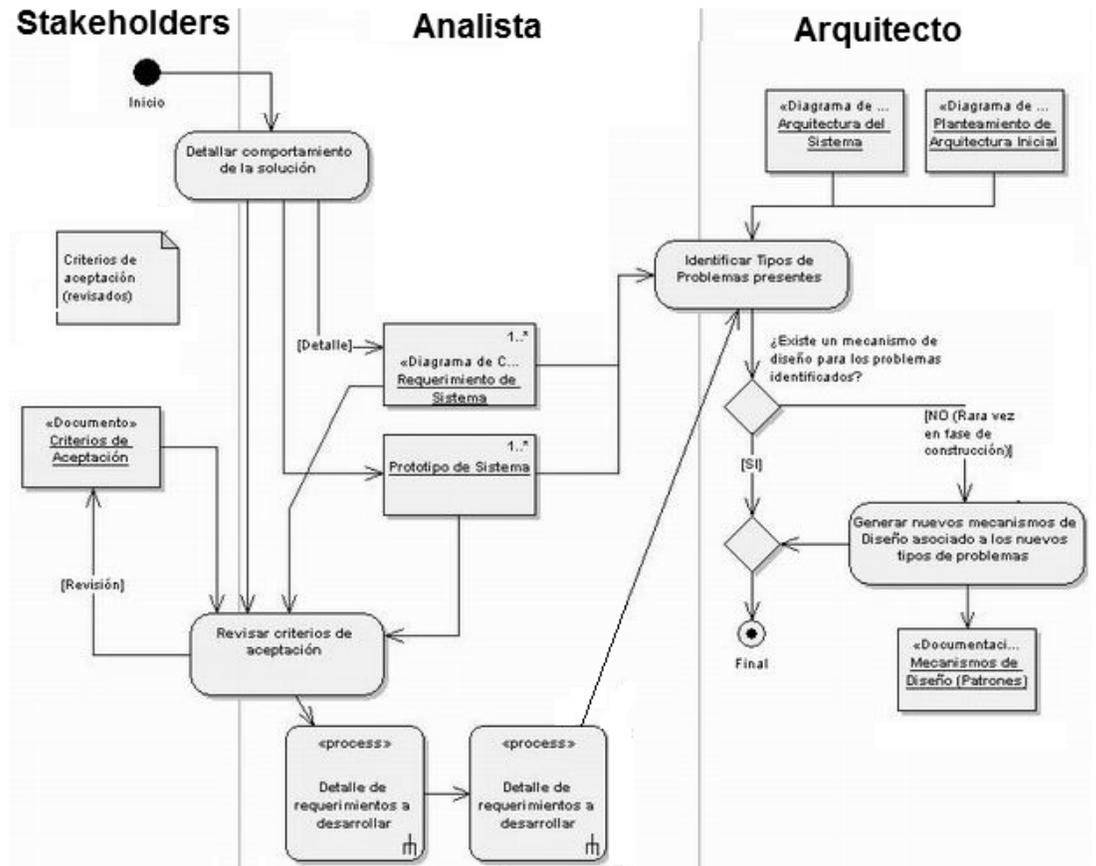


Es importante denotar que el elemento de Gestión de Conocimiento es un elemento transversal a los demás, es decir, que brinda retroalimentación para la mejora continua a los demás elementos. Éstos se presentan en forma cíclica partiendo de la Gestión de Procesos, ya que ésta propone una serie de guías y lineamientos en general de todos los procesos, seguidos por la estandarización de artefactos y metodologías, por lo que surge la necesidad de control de cambios en ambos y por último la innovación y mejora continua a través de la gestión del conocimiento. Esta secuencia con la que se deben instanciar los elementos del modelo es la aconsejada, pero ésta también puede ser modificada según las necesidades de la organización.

2.2 Gestión de procesos

La Gestión de Procesos tiene como objetivo minimizar el impacto de circunstancias poco previstas o de índole desconocida, previniendo la recurrencia de todo tipo de incidentes en el desarrollo y mantenimiento de los sistemas de información. Además busca tener, de alguna manera, todo realmente ordenado, es decir, que esté representado de manera de procesos y/o procedimientos todas las actividades e interacciones del departamento de desarrollo de software. Básicamente este elemento lo que busca es ordenar la forma a través de la cual se realizan las actividades del departamento y normar la forma de interacción con todas las demás unidades. Un ejemplo de un procedimiento se muestra en la siguiente figura:

Figura 3. Procedimiento de desarrollo



Éste puede tener una representación muy general o muy específica, dependiendo de las necesidades de la organización, lo importante es normar las interacciones con las demás unidades y especificar un proceso para el desarrollo de software y mantenimiento de sistemas.

A parte de minimizar el impacto en la organización respecto a los incidentes, problemas o inconvenientes provocados por errores en la infraestructura de TI en la organización, parte de su labor principal es organizar y estructurar la forma de trabajo de todo el departamento de modo que se pueda obtener un mapa general de las funciones, responsabilidades,

interacciones, normas y proveer una forma de medir lo que estamos haciendo y buscar la mejora continua a lo largo del tiempo.

Además se plantea un tipo de prevención referente a la recurrencia de incidentes o problemas, ya que es muy común ver en el área de TI de una organización a dos personas distintas tratar el mismo problema en diferentes tiempos y realizar todo el proceso para solucionarlo, uno independiente del otro.

Para determinar y proponer el uso de este elemento en nuestra organización es necesario seguir varias tareas y actividades:

1. Obtener un organigrama específico del departamento.
2. Identificar tendencias o patrones en el desarrollo y mantenimiento de software, además de las bases generales para la resolución de problemas o incidentes.
3. Realizar una narrativa de todos los procesos y/o procedimientos que se realizan actualmente, es decir, todas las actividades, responsabilidades y objetivos de cada actividad.
4. Definir el proceso macro y/o procedimientos específicos para el departamento.
5. Definir responsables y artefactos en cada actividad de los procedimientos.
6. Normar el proceso general y si hubiesen, los procedimientos específicos.
7. Educar a todos los involucrados en los procedimientos para que estén de acuerdo y que implementen de la mejor manera todos los procesos o procedimientos.
8. Retroalimentación con el tiempo, para mejora o innovación de los procedimientos.

Los procesos y procedimientos del departamento o área de desarrollo de software lo que buscan es realizar lo adecuado al fin u objetivo último y adaptarse a las necesidades específicas de los proyectos y requerimientos de nuestros clientes.

2.3 Estandarización de elementos de desarrollo

El cumplimiento de los estándares definidos para el desarrollo de un producto de software refleja la calidad que se tuvo para la elaboración del mismo, así como hace constar que el producto independientemente de la funcionalidad, presenta una serie de atributos que aseguran la calidad y mantenibilidad de tal producto.

La Ingeniería de software engloba un conjunto amplio de disciplinas, tales como administración de proyectos, definición y mejoramiento de procesos, factores humanos, diseño, programación y validación de software, las cuales conforman una columna básica de conocimientos que requiere ser consolidada para la elaboración de cualquier producto de software, pero además de ello, hoy en día es necesario agregar e implementar la disciplina de aseguramiento de calidad que actualmente tiene un alto impacto en el desenvolvimiento y resultado de un proyecto en la industria del desarrollo de software.

2.3.1 Metodología de desarrollo

Para que todo desarrollo resulte satisfactorio para ambos actores (Proveedor y el Cliente), es necesario seguir los lineamientos y guías especificadas en la metodología de desarrollo adquirida por la organización, ya

que esta guía provee de la especificación y resultados que se deben de construir para tener un entregable y darle solución a un problema, teniendo como resultado un software que cumpla con ciertas expectativas y características planteadas al inicio.

Un estándar que define la metodología de desarrollo es una descripción de la misma, definida para la realización o construcción de cualquier proyecto, que se presenta como una serie de lineamientos básicos y complementarios que constituyen un esquema genérico que además puede especializarse y personalizarse para una gran diversidad de proyectos de software y diferentes sistemas de información en cualquier organización y departamentos de TI de varios tamaños, presentando para cada disciplina mostrada una serie de actividades que se pueden cumplir para poder obtener varios artefactos, que en conjunto con los de las demás disciplinas, se obtenga un producto final entregable.

2.3.2 Estándares de arquitectura y de tecnologías

Es muy importante definir este tipo de estándares en una organización, en el área de TI, ya que presenta o describe los distintos elementos a utilizarse para lograr tener una presentación uniforme de las aplicaciones, facilitar posibles cambios, tanto de alto como de bajo nivel, al igual que asegurar un mantenimiento uniforme para todas y cada una de las aplicaciones desarrolladas en la organización.

El objetivo principal de estos estándares (arquitectura y de tecnologías) es presentar la arquitectura y estructura general para el desarrollo de todas las aplicaciones, definiendo el dominio de tecnologías a utilizarse y la forma de

acoplamiento entre ellas proponiendo así una estructura general que puede personalizarse en el desarrollo de aplicaciones por parte de la organización.

Todas estas políticas tienen como objetivo general, el garantizar el eficiente funcionamiento de las aplicaciones, asegurando que los sistemas desarrollados de esta forma optimicen las ventajas que la plataformas tienen, ya que existirá una especialización de un conjunto de ellas, y a la vez, un mejor conocimiento de las magnitudes de la utilización de ellas, garantizando un buen rendimiento de las aplicaciones.

2.3.3 Estándares de nombrado y construcción de proyectos

Los estándares de nombrado y construcción de proyectos son todas aquellas guías y lineamientos que definen la manera con la cual un desarrollador debe de empezar a definir y a trabajar un proyecto, considerando para ello:

- El nombre del proyecto, el cual servirá para especificar una especie de ficha de proyecto, con la que se deberá de seguir todo el desarrollo del mismo.
- Las herramientas de integración y ambientes de desarrollo de los proyectos (IDE(s), herramientas de compilación, integración y deploy), con lo que se obtiene una estandarización de la forma de empezar la construcción de un proyecto.
- El mecanismo de implementación y construcción del proyecto a través de un IDE establecido. Son todos aquellos pasos que deben de realizar los desarrolladores o programadores a través de un IDE y

otras herramientas de desarrollo, para establecer la estructura general que forme la base para empezar a realizar cualquier tipo de aplicación dentro del dominio de tecnologías y arquitecturas definidas previamente.

2.3.4 Estándares de codificación

Un estándar de codificación abarca todos los aspectos de la construcción y generación de código, repercutiendo claramente en la legibilidad y la extensibilidad de cualquier clase, componente o proyecto de software, fomentando que nuevos desarrolladores involucrados en la construcción de un determinado software, se acoplen prontamente al proceso de desarrollo.

Estos estándares plantean un esquema y una guía para nombrado de cualquier expresión de código, como por ejemplo:

- Paquetes
- Clases
 - Atributos
 - Métodos
- Procedimientos y Funciones
 - Variables Locales
 - Parámetros
- Constantes
- Librerías
- Componentes
- Ejecutables

Además de establecer estas guías, es necesario definir en los estándares de codificación, la manera con la cual se debe de crear el esquema de una aplicación, proyecto, librería o componente, es decir, la forma en la que un proyecto debe presentar la organización de los paquetes o directorios en donde se encuentren los fuentes, ya que es muy importante para el orden y entendimiento de una clara definición de responsabilidades.

Esta última definición debe de tomar muy en cuenta el patrón o estándar definido para la arquitectura y tecnologías, ya que esto influye directamente en la forma en que se organizará una aplicación o proyecto.

2.3.5 Estándares de control de versiones

La aplicación de este tipo de estándares se lleva a cabo con la gestión de versiones (revisiones) de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo.

Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado junto a las posibles especializaciones realizadas para algún cliente específico.

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos y no sólo para código fuente sino para documentos, imágenes, etcétera.

La misión principal de un sistema de control de versiones es permitir la edición colaborativa y la compartición de los datos. Sin embargo, existen diferentes sistemas que utilizan diferentes estrategias para alcanzar este

objetivo. Por lo que se debe de establecer un estándar o esquema específico que contenga como mínimo:

- Etiquetas
- Revisiones
- Ramas
- Repositorios
- Líneas base de los productos

2.3.6 Estándares de interfaces

El propósito de establecer los estándares de interfaces de usuario para la presentación de cualquier aplicación es definir las características, forma y estructura que debe caracterizar a la organización, llevando a una uniformidad de la apariencia en las aplicaciones desarrolladas, estableciendo para ello un conjunto de reglas acorde a la usabilidad y aspecto que deberá tener toda interfaz desarrollada.

Dentro de las características mínimas que se deberán definir para obtener un índice de calidad alto, respecto al seguimiento de los lineamientos para presentar la información, se pueden mencionar los siguientes:

- Disposición gráfica o plantillas
- Restricciones referentes a controles como marcos, tablas
- Ubicación de menús
- Bloques de aplicación
- Formularios
- Presentación y validación de campos de entrada
- Presentación y validación de campos y combos de selección

- Presentación de catálogos
- Bloque de botones de acción
- Nombrado de controles y botones de acción
- Validaciones, campos requeridos y campos opcionales
- Etiquetas y campos de salida de la información
- Zona de mensajes
- Mensajes y páginas de confirmación
- Páginas de error
- Textos explicativos
- Archivo CSS (hoja de estilo)
- Espacio efectivo de pantalla
- Tipografía
- Colores (presencia de los colores que identifican a la organización)

2.3.7 Estándares de nombrado de base de datos

Este tipo de estándar es el más común o el más utilizado en las organizaciones en la actualidad, y lo que define son todas aquellas especificaciones referentes a como establecer los nombres de los esquemas de base de datos, tablas, campos, disparadores, y cualquier objeto que forme parte de una base de datos, garantizando un entendimiento y dominio sobre las entidades y objetos manejados.

2.3.8 Estándares de documentación

La definición de este tipo de estándares establece los lineamientos para la preparación y administración de la documentación que se genera a lo largo del

desarrollo de un proyecto, además de definir las guías para la elaboración de documentación que se refiere a políticas, normas y estructuración.

Es muy importante que los documentos se estandaricen ya que presentan una forma tangible de representar el software y el proceso del mismo, aparte de brindar apariencia, estructura y calidad consistente, facilidad de leer y comprender.

La aplicación de estos estándares puede efectuarse de varias maneras, pero como mínimo debe definirse:

- Proceso de documentación
 - Proceso a seguir para la producción del documento
- Plantillas para todo tipo de artefactos
 - Estas deben tener un estilo y apariencia consistente, además de brindar una estructura consistente con los del proyecto y la organización
- Las herramientas definidas para elaborar los documentos
 - (Hojas de cálculo, procesadores de texto, herramientas de diagramación, etc.)

2.4 Control de cambios

Antes de empezar a describir las responsabilidades de este elemento, se debe de tomar en cuenta varios principios que ayudarán a conceptualizar y a valorar la importancia que conlleva el adaptar este elemento a los procesos internos y comprender la función del mismo en las organizaciones actuales. Estos principios a considerar son:

- El software debe de ser considerado como un activo de tecnología de la Información al igual que el hardware.
- El software debe permitir a las organizaciones alcanzar sus objetivos de negocio, por lo tanto debe tenerse el control de cuantos activos posee, los costos que conllevan tenerlo activo o inactivo, y la especificación a detalle de cada uno.
- En las organizaciones es necesario administrar el hardware y el software de manera correcta para obtener mayores beneficios.
- Es importante aclarar que el software es un activo de negocio que es esencial para un funcionamiento eficaz y eficiente, y está sometido a obligaciones legales y contractuales.

El control de cambios se encarga de la configuración de todos los Activos que constituyen un sistema como un todo y la manera de mantener esta configuración durante el tiempo. Este elemento administra toda la infraestructura y procesos necesarios para identificar, registrar, controlar, proteger de manera efectiva los activos de software de toda organización a lo largo de las diferentes etapas del ciclo de vida de los productos.

El control de cambios de elementos y activos también trata de un grupo de metodologías que permitirán adecuar y optimizar la gestión y el control de todos los activos digitales, el control de productos y versiones y la configuración necesaria de los mismos.

El desarrollo y la implementación del Plan para la Administración de la Configuración y Control de Cambios es un factor crítico de éxito para el control de los sistemas en producción, legados y desarrollos nuevos y la identificación de los beneficios financieros de la organización.

La incidencia de este elemento ocurre en la construcción y elaboración de los activos de un proyecto y luego que se concluye el proyecto y se expone el servicio o producto creado, ya que su función principal reside en mantener toda la infraestructura de los proyectos o sistemas de software bajo control, asegurando la debida aplicación de los procedimientos estándares y lineamientos definidos para la atención de los cambios, con el fin de minimizar el impacto de los servicios prestados por la organización.

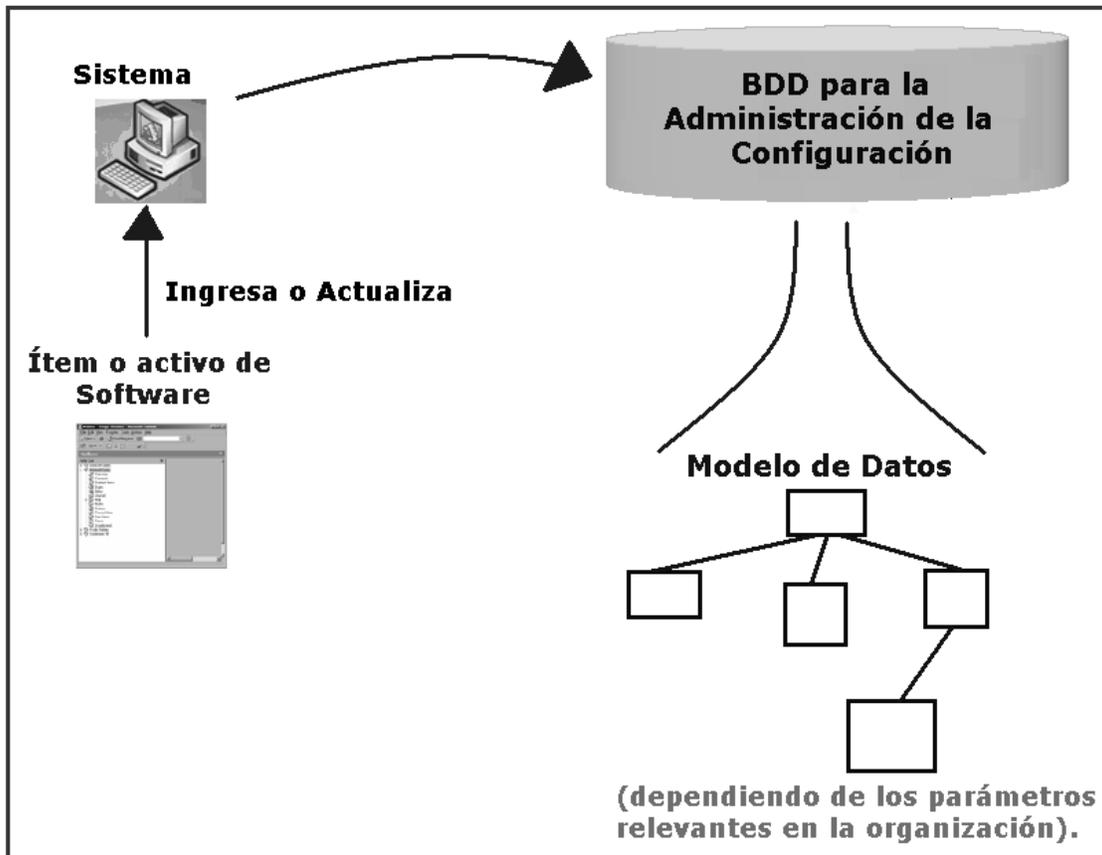
Para empezar con la aplicación del modelo y del elemento en cuestión, es necesario crear, definir o adquirir un software o sistema (que puede ser el utilizado para administrar cualquier inventario en la organización, siempre y cuando sea personalizable o configurable) que maneje toda la información que corresponda a cada ítem de hardware y de software, para lo cual se deben establecer primero: la Base de Datos o repositorio que contendrá la información y en algunos casos algunos ítems que forman parte del inventario del área de TI. Luego se deberán establecer los atributos o características que desean almacenarse para la administración y configuración de dichos ítems.

Al referirse a un ítem, lo que se quiere dar a entender, en el contexto de TI, es a identificar cualquier elemento, tangible o intangible, que contribuya a que la organización marche y opere de manera correcta y estable, además de ayudar a alcanzar los objetivos y fines de la organización y de servir de apoyo en todas las demás áreas.

Pudiéndose identificar fácilmente en cualquier organización cualquiera de los siguientes ítems correspondientes al departamento de desarrollo de software:

- Configuración de Servidores de Aplicaciones, S.O., etc. (ambiente)
- Componentes de software (reutilizables)
- Productos de software (sistemas en general)
 - Requerimientos Funcionales (casos de uso, historias de usuarios, listas de funcionalidades, etc.)
 - Modelo de datos
 - Modelo o documentos de arquitectura
 - Análisis
 - Diseños
 - Casos de prueba, pruebas funcionales, de rendimiento, etc.
 - Esquemas de despliegue
 - Código fuente

Figura 4. Administración del cambio



Es por eso que se distinguen dos tipos de inventario o repositorios para poder llevar el control de todos los activos y elementos que conforman un sistema como un todo, y que en su momento deben de aplicarse los dos para un mejor orden y organización del área de desarrollo de software.

- Repositorio de control de cambios (activos)
- Repositorio de componentes y releases (productos)

La aplicación de estos métodos para la administración de cada activo en una organización provee de varios beneficios, entre los que se pueden identificar la mejora en el rendimiento del negocio, tal como poder conocer

rápidamente los recursos que tiene que gestionar y ahorra tiempo, ya que el personal de informática no tendrá que dedicar tiempo a tareas administración y control de activos, ni a perder tiempo en la localización o búsquedas de los diferentes activos.

2.4.1 Repositorio de control de cambios para activos

Este repositorio tiene como función principal identificar, registrar, controlar y gestionar todos los ítems que conforman un sistema en un tiempo determinado y tener la capacidad e regresar a cualquier momento en el tiempo sin esfuerzo alguno.

Cada ítem registrado debe de contener las siguientes características, o bien, un subconjunto de ellas:

1. ID
 - Identificador del activo
2. Nombre
 - Nombre del activo
3. Atributos del elemento
 - Listado de atributos (dependiente del activo)
4. Relaciones con otros elementos
 - Listado de elementos con los cuales está relacionado, al igual que la razón de la relación
5. Versión o serie
 - Número de versión o serie del activo
6. Estado
 - Estado actual del activo (dependiente del activo)

7. Dependencias

- Listado de tipo y razones de dependencias

8. Responsable

- Persona responsable del control o actualización del activo

9. Nivel

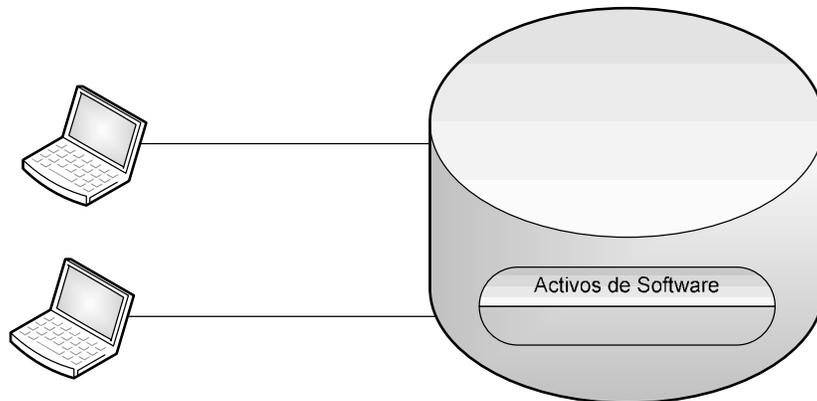
- Nivel de identificación del activo (control de detalle interno de la organización)

Todas estas características y algunos de los activos (software) deberán de estar almacenados en el repositorio desde el inicio.

Un repositorio de elementos adecuado proveerá a los usuarios (desarrolladores) de:

- Monitoreo de la calidad del desarrollo de un producto.
- Tener la imagen o foto de un sistema en cualquier parte del tiempo.
- Manejo de cambios eficiente con equipos grandes de trabajo.
- Si en la organización todo el software, está moderadamente ordenado, bajo el esquema propuesto.

Figura 5. Control de activos



2.4.2 Repositorio de componentes y releases

A diferencia del inventario general de activos o elementos, se ha identificado el inventario de sistemas y componentes de software, que del mismo modo, gestiona los ítems o activos del área de desarrollo en la organización, solo que todo enfocado a releases y componentes funcionales o que la organización los utiliza para alcanzar sus objetivos y que requieren de control, mantenimiento y soporte.

De la misma manera que en el inventario general de activos, este inventario lleva el control de los ítems de software a través de ciertas características, tales como:

1. ID
 - Identificador del activo
2. Nombre
 - Nombre del activo
3. Resumen general del componente

- Nombre del departamento o razón de su adquisición, creación o desarrollo.
4. Funcionalidad del activo
 - Listado de atributos que definen la funcionalidad del mismo.
 5. Relaciones y dependencias con otros elementos
 - Listado de elementos con los cuales posee alguna dependencia o está relacionado.
 6. Versión del activo
 - Número de versión
 7. Estado
 - Estado actual del activo
 8. Autor(es)
 - Listado de personas que construyeron el activo o que son responsables del control y mantenimiento del activo.
 9. Fecha de actualización
 - Fecha de creación o modificación del activo.
 10. Ubicación lógica
 - Ubicación lógica respecto de la organización, ésta depende de la forma en que se estandarizó la forma de desarrollar aplicaciones en la organización.
 11. Ubicación Física
 - Ubicación física del activo (en el repositorio o en el repositorio para la administración de la configuración.)

Este repositorio provee de muchos beneficios tales como los mencionados en el inventario general de activos anterior y otros tales como:

- Se observará si se está utilizando software que no se necesita.
- Se promueve la reutilización de software, ya que se tiene el historial de todos los componentes desarrollados, al igual que sus atributos.
- Se contará con una especie de biblioteca de software.

2.5 Gestión de conocimiento

La gestión de conocimiento es el elemento que nos proveerá del mecanismo enriquecedor de nuestro departamento y que a su vez nos brindará la posibilidad de aprender y mejorar la calidad de todo lo que hacemos conforme al tiempo.

Este elemento con el tiempo brindará la clave para la optimización de los recursos. Esta tarea es llevada hoy en día por muy pocas organizaciones derivado a que la gestión de conocimiento es una disciplina que implica mejorar el flujo del conocimiento hacia él mismo y hacia los departamentos involucrados de manera clara y concisa y necesita el uso de una serie de herramientas informáticas para su cometido.

Estas herramientas de soporte de la gestión de conocimiento deben ser aplicadas al objetivo de este elemento en el departamento de desarrollo de manera que promuevan y posibiliten el proceso de transferencia de conocimiento y con ello facilitar la generación, esquematización y transferencia del mismo, para poder mejorar la planificación, control y toma de decisiones en el desarrollo de proyectos de software. De tal forma que como principal meta, es necesario conseguir que la organización, empezando por desarrollo y mantenimiento de sistemas, se conceptualice lo que realmente se está

haciendo, se organice y por último se distribuya, convirtiéndose esto en un ciclo de mejora continua.

A partir de la proposición de Nonaka en su libro Organizaciones Inteligentes 1999, “La clave para la innovación reside en liberar el conocimiento tácito, personal, de los miembros de la organización”, se puede integrar varias características que deben tenerse en cuenta para que el sistema de gestión de conocimiento implantado marche y se desarrollen los activos intelectuales de los procesos, además de implementarlo con las tecnologías de información más acertadas.

Figura 6. Proceso de creación de conocimiento



Fuente: Ikujiro Nonaka, Hirotaka Takeuchis, **The Knowledge-Creating Company**, Universidad de Oxford, 1995.

Tomando como base el proceso de creación de conocimiento descrito por Nonaka, se puede elevar el capital intelectual del proceso general de desarrollo en la organización, implementando dicho ciclo con tecnologías de la información enfatizando el flujo de conocimiento a lo largo del proceso. Por consiguiente, se debe adherir al proceso de desarrollo las siguientes tecnologías.

- Foro electrónico de discusión para resolver dudas, inconformidades y plantear preguntas de manera que desvanezcan las incertidumbres entre los miembros de los departamentos respectivos y creando un historial de ayuda e inducción para muchos. (Socialización)
- Wikis y workflows de los diversos procesos y/o procedimientos de desarrollo de manera que se establezca y se norme el flujo para cada proyecto y se documente las guías para realizar varias actividades asociadas a los proyectos. (Exteriorización)
- Gestión documental de los procesos y activos de los proyectos de forma que se propague toda la experiencia de los integrantes en los diversos proyectos y desarrollo de los mismos, creando una base sólida de experiencia y de aprendizaje continuo. (Interiorización)
- Intranet y portal de conocimiento de modo que se ubique en un lugar la mayor cantidad de conocimiento para uso de todos los interesados. (Combinación)

3 PERSONALIZACIÓN DEL MODELO

3.1 Identificación de los atributos de la organización (Ej. SAT)

La Superintendencia de Administración Tributaria, es una institución que cuenta con una gerencia (Gerencia de Informática) dedicada a contribuir y apoyar el cumplimiento del deber y misión de la institución, a la vez que presta los servicios tecnológicos y de desarrollo de soluciones innovadoras y eficientes, para lo cual su departamento de desarrollo y mantenimiento de aplicaciones es lo suficientemente grande (alrededor de treinta y cinco analistas) para alcanzar los objetivos propuestos, y por tal motivo, se eligió dicha institución para poder verificar el nuevo paradigma o modelo propuesto.

Entre los atributos y características del equipo de desarrollo identificadas se pueden mencionar las siguientes:

- Personal de 35 analistas/desarrolladores.
- Equipo de punta para cada analista.
- Servidores suficientes para uso y despliegue de aplicaciones.
- Infraestructura de red bien cimentada.
- Ambiente de trabajo distribuido. (DBA's, analistas, equipo técnico, soporte especializado, etc.)
- Adecuadas instalaciones.
- Utilización de Herramientas y tecnologías de punta.
- Profundo interés por producir productos de alta calidad.

3.2 Personalización de los elementos del modelo

Es importante hacer notar que debemos de personalizar todos los estándares de desarrollo que sean de nuestra utilidad y que provean de algún tipo de significancia en la consecución de nuestros objetivos, y esto va a depender de los atributos que hallemos identificado en la organización en donde se desea aplicar el modelo propuesto.

Para nuestro caso, es necesario definir todos los tipos de estándares debido a que es una organización muy grande y que es preciso que todo el equipo se guíe acorde a los estándares y lineamientos de la organización, para una mejor estructuración y elaboración de los productos entregables.

3.2.1 Personalizar metodología de desarrollo

Para personalizar una metodología debemos preguntarnos ¿Qué metodología debo usar para el desarrollo de un determinado tipo de producto de software? Y de hecho esta pregunta se torna muy importante, pues como gestores o arquitectos de software, debemos tener un plano en que apoyarnos y poder aplicar las disciplinas, fases, asignación de roles y entregables que nos brinde la metodología elegida.

La gran mayoría de proyectos de desarrollo de software y sistemas de información lleva riesgos y son difíciles de gestionar, organizar, controlar y desde luego planificar.

Por no contar con una dicha metodología adecuada incurrimos en las siguientes implicaciones:

- Atrasos en los tiempos
- Altos costos
- Clientes insatisfechos
- Desarrolladores molestos e insatisfechos.
- Realizar diseños de proyectos de manera rígida.
- Caer en diseños de metodologías propios en distintos proyectos.

A menudo no se encuentra la más adecuada metodología y se termina diseñando o estableciendo una propia siempre y cuando cumpla con el objetivo. Esto por lo regular no está mal y es una de las alternativas más aconsejables.

Al realizar los diseños de los proyectos de software de manera rígida, se llega a tener un escenario en el cual un cambio en los requerimientos del sistema resulta muy complicado de realizarlo y en ocasiones peligran la planificación o cumplimiento del proyecto.

Actualmente, lo anterior mencionado puede repercutir y afectar en el hecho de que algunos errores el usuario solamente los vea en la parte final, causando una mala impresión del desarrollo del proyecto.

Cuando un proyecto no cumple con las expectativas del cliente suelen rebasar el presupuesto asignado y tener numerosos retrasos. Hoy en día hay diferentes metodologías aplicables a distintos tipos de proyectos, entre las más comunes y de las cuales se tienen buenos resultados podemos mencionar: RUP (Racional Unified Process), XP (Extreme Programming) y MSF (Microsoft Solution Framework).

Para el caso de la institución SAT se eligió utilizar la metodología RUP, personalizando gran parte de las disciplinas descritas, entregables, actividades y roles presentados, obteniendo una metodología que abarca un gran número de escenarios en los cuales se desarrolla proyectos en dicha institución. Además que por ser una institución muy grande y con un gran número de recursos, se publicó el resultado de la personalización de la metodología, para que cualquier persona pueda consultar los pasos o actividades que necesita efectuar, a parte de tener presente los entregables que se requieren.

Esta publicación puede ser realizada en el sitio de la intranet de la organización o bien en un servidor dedicado a la inducción del personal de desarrollo, tal como se muestra en la siguiente figura:

Figura 7. Personalización de la metodología de desarrollo en SAT



3.2.2 Personalizar estándares de arquitectura y componentes

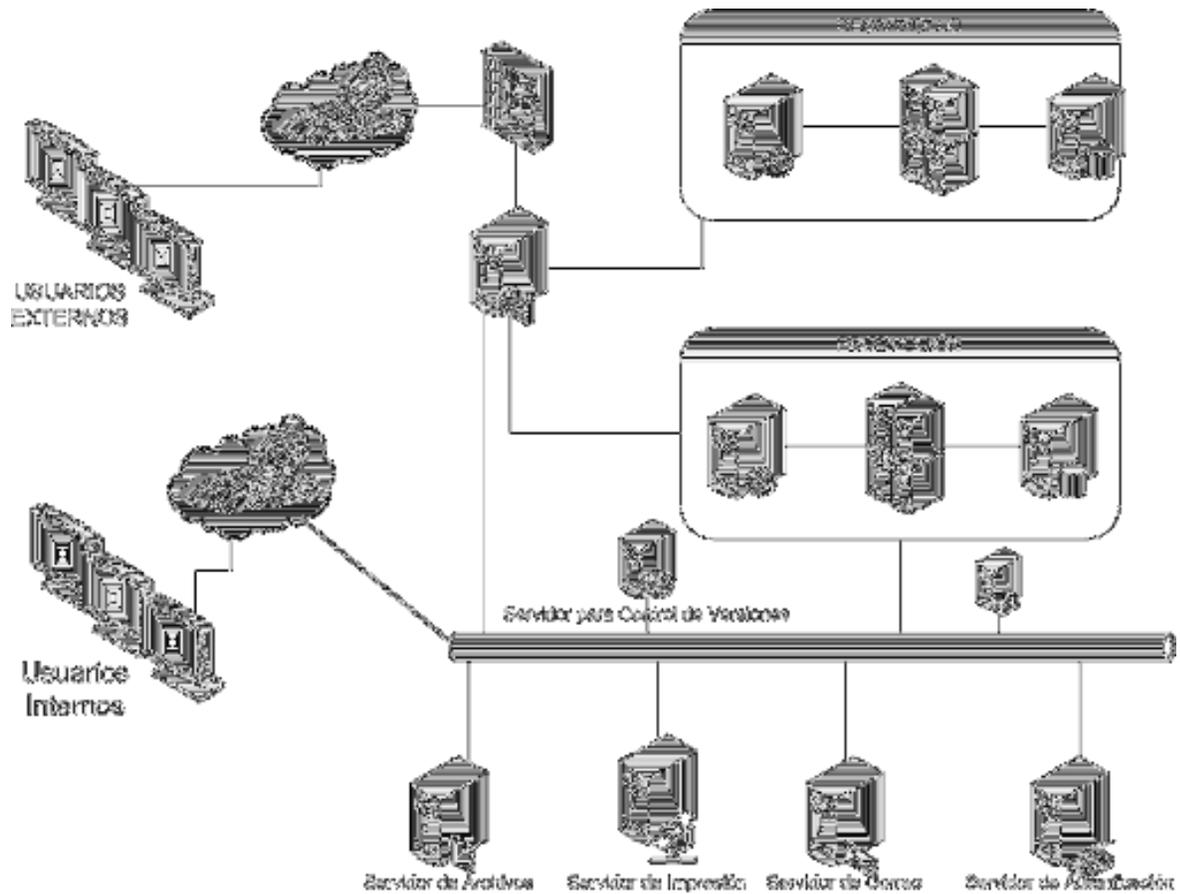
El éxito de toda aplicación en gran parte está dado por la arquitectura física y lógica definida al principio del desarrollo para las aplicaciones. Por esa razón es necesario que desde el inicio tengamos un dominio de arquitecturas candidatas para la elección acorde a las necesidades del proyecto y conocimiento de la forma en la que lo vamos a subdividir de tal forma que exista una división en elementos de alto nivel y una forma de desplegar o instalar dichos componentes.

Primero es necesario ver que tipo de aplicaciones o proyectos se desarrollan, como por ejemplo cliente-servidor, aplicaciones web, distribuidas, etc., para luego poder estandarizar varias arquitecturas en las que se puedan implementar un dominio de tecnologías y con esto llegar a una especialización de ellas y construir con el tiempo productos de mejor calidad.

También es necesario definir una arquitectura básica de servidores para el despliegue de nuestros componentes o elementos de alto nivel (capas de nuestras arquitecturas), para tener una mejor organización y estructuración de todos los sistemas desarrollados, por lo que antes se debe especificar, con cada tecnología a utilizar, que requerimientos de hardware y software necesita para su perfecto funcionamiento y desempeño, proponiendo una parte de nuestra arquitectura física a definir.

Como un ejemplo para la definición de una arquitectura física de servidores y de equipo necesario para una organización con departamento de desarrollo de sistemas, podemos apreciar la siguiente figura, que muestra un esquema básico y que se puede adoptar en un gran número de organizaciones de mediano tamaño.

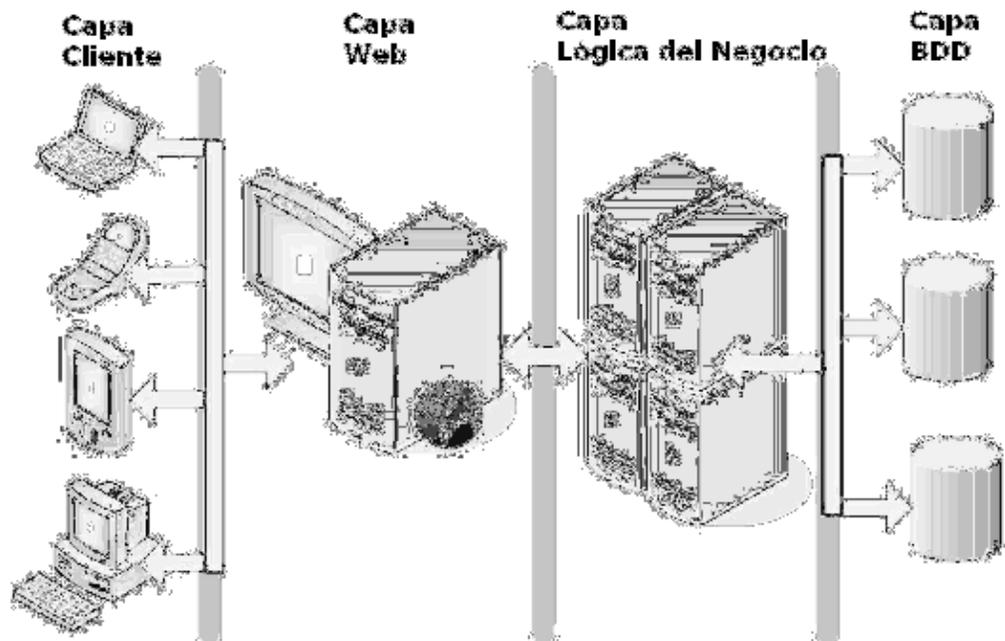
Figura 8. Arquitectura física de una organización



La elección de las arquitecturas candidatas con las que debe contar o adoptar una organización, tiene que ir acorde al tipo de problemas a los que regularmente se intenta dar solución con el desarrollo de un sistema informático. Como por ejemplo, hay arquitecturas para aplicaciones Cliente-Servidor, aplicaciones web, aplicaciones distribuidas, etc., para lo que se debe tener muy en cuenta, el dominio de tecnologías que se conocen o que se utilizan, ya que son éstas tecnologías las que nos van a brindar el tipo de estructura de servidores que debemos de tener, el tipo de hardware y software que sea necesario para que los componentes de las aplicaciones tengan un

buen funcionamiento y un buen desempeño, al igual que una mejor organización de todos los componentes realizados en dichas tecnologías. Un ejemplo de la definición de arquitectura para aplicaciones web podría ser como se observa en la siguiente figura:

Figura 9. Arquitectura básica de aplicaciones web

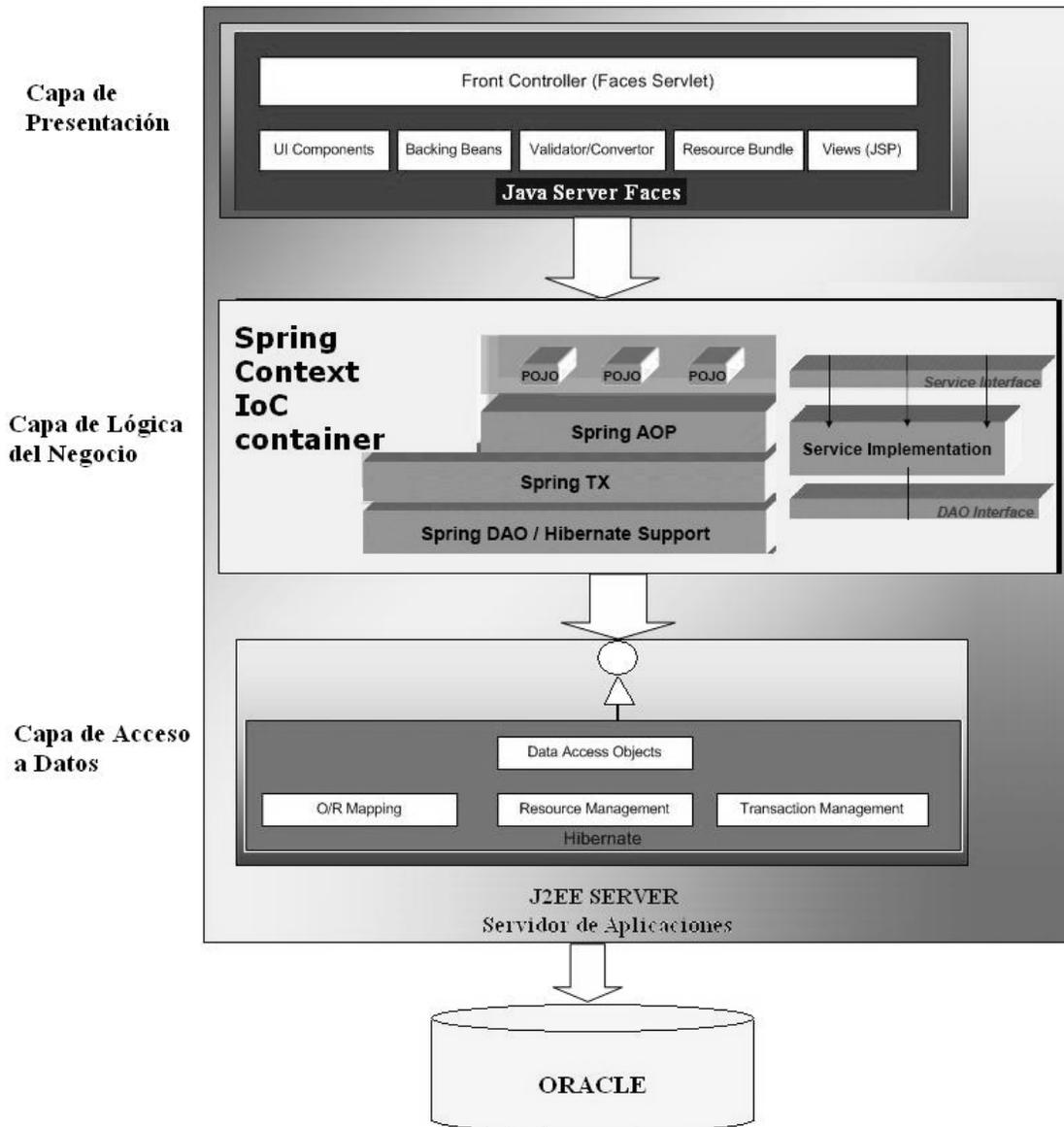


Que como vemos en la figura, dicha arquitectura, necesita la infraestructura de tres tipos de servidores para el acoplamiento y despliegue de las capas de la aplicación. Dichos servidores son: Servidor Web, Servidor de Aplicaciones y Servidor de Base de Datos.

Esa infraestructura mencionada puede estar atada a la demanda de las tecnologías que se utiliza para el desarrollo de aplicaciones. Un ejemplo de esta explicación puede verse en la siguiente figura, la cual nos brinda un panorama

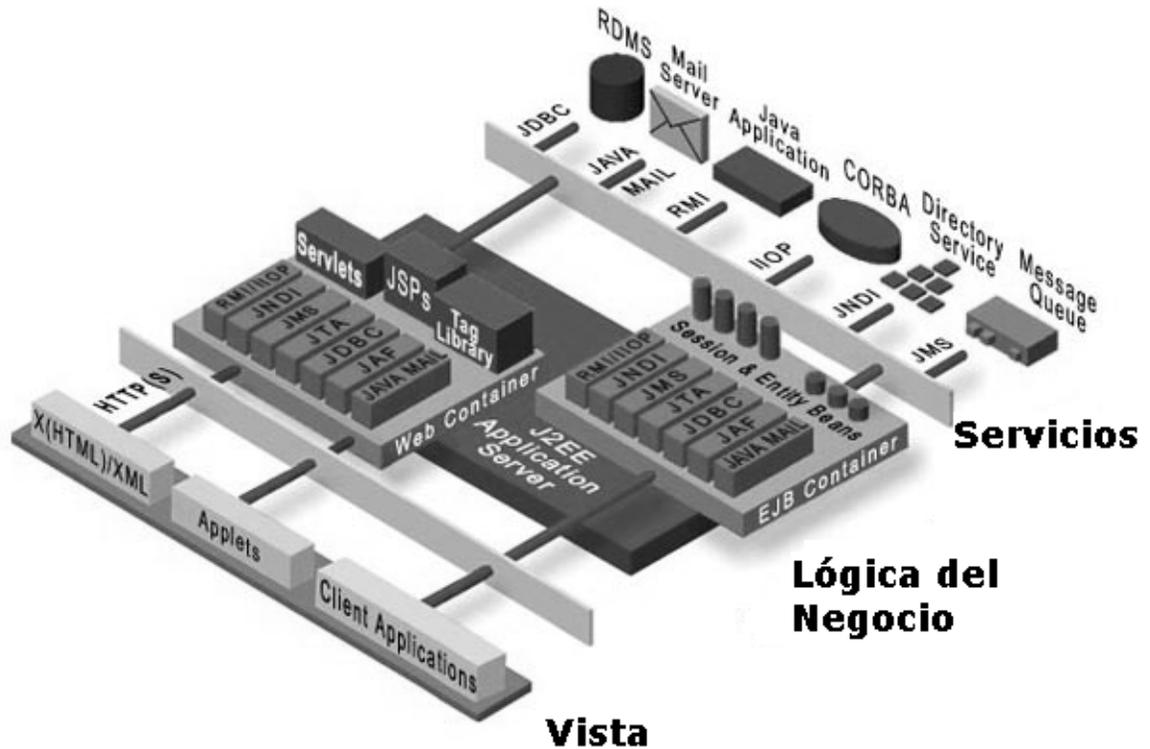
de arquitectura y de las capas que utilizaremos, al igual que los requerimientos físicos para su correcta ejecución.

Figura 10. Arquitectura de tecnologías y componentes



La figura anterior muestra un dominio específico de tecnologías para las capas, es decir el diseño de la arquitectura para una aplicación, pero para una organización de gran tamaño es necesario y conveniente tomar muy en cuenta una serie de problemas y escenarios en los que vamos a tener obligadamente, contar con diversas tecnología para solucionar e implementar esas necesidades y para lo cual hay que estar preparados y tener un mapa en el cual basarnos y tener el apoyo y soporte necesario. Un mapa de varias tecnologías puede asemejarse al siguiente:

Figura 11. Dominio de tecnologías para las capas

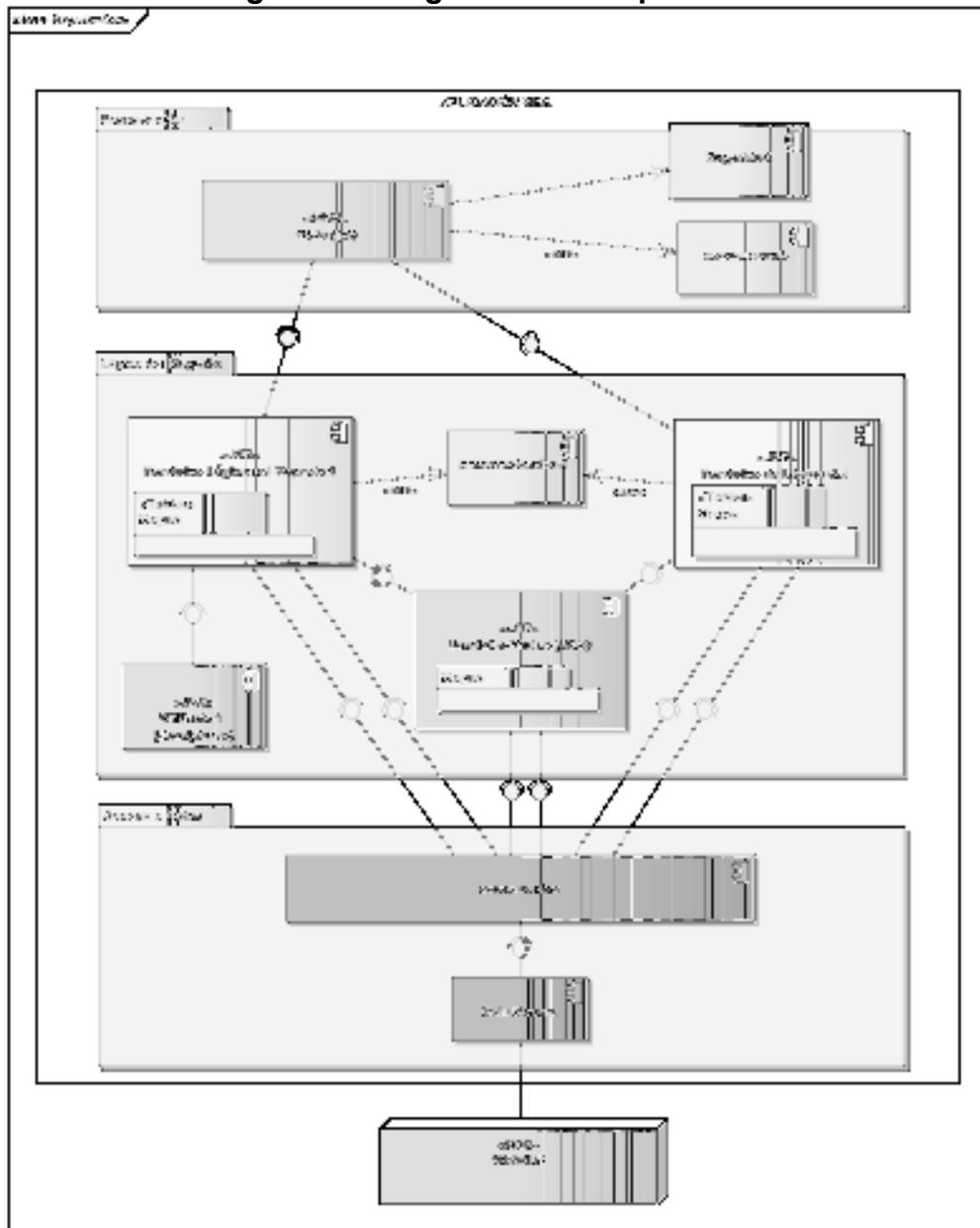


La personalización de todos estos argumentos mencionados acorde a arquitecturas y tecnologías utilizadas, tiene que ser proporcionado por el

personal dedicado a aspectos arquitectónicos y de apoyo general a las aplicaciones

En la Superintendencia de Administración Tributaria (SAT), este personal es el encargado de brindar el apoyo necesario a cada desarrollador y analistas para el análisis, diseño implementación, pruebas y puesta en producción de las aplicaciones, ya que plantea los lineamientos y estándares a través de los cuales los desarrolladores y analistas pueden construir las aplicaciones. Para ello publica toda información, documentación o artículos referentes a estos temas, además que toma el rol de soporte y apoyo para la construcción de aplicaciones, investigando y construyendo componentes de ayuda para todo el departamento de desarrollo. Un ejemplo de apoyo es brindar una arquitectura candidata, como se muestra en la siguiente figura:

Figura 12. Diagrama de componentes

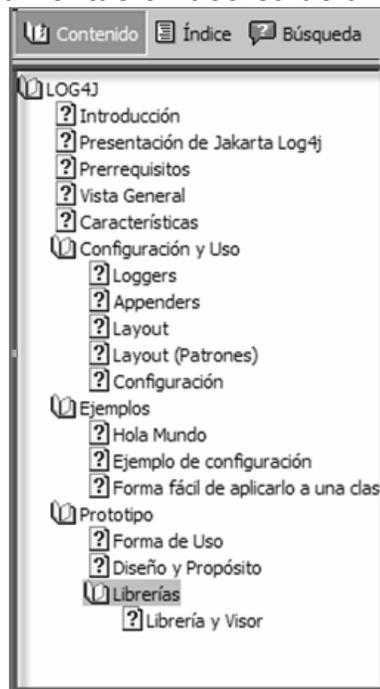


O bien, proporcionar la información y documentación necesaria para que un analista o desarrollador pueda implementar una tecnología en la aplicación designada para su desarrollo, y que esta información esté publicada para uso general de todo el departamento o área, como bien podemos observar en las siguientes figuras un ejemplo de esta publicación, en la organización SAT:

Figura 13. Documentación acerca de una tecnología

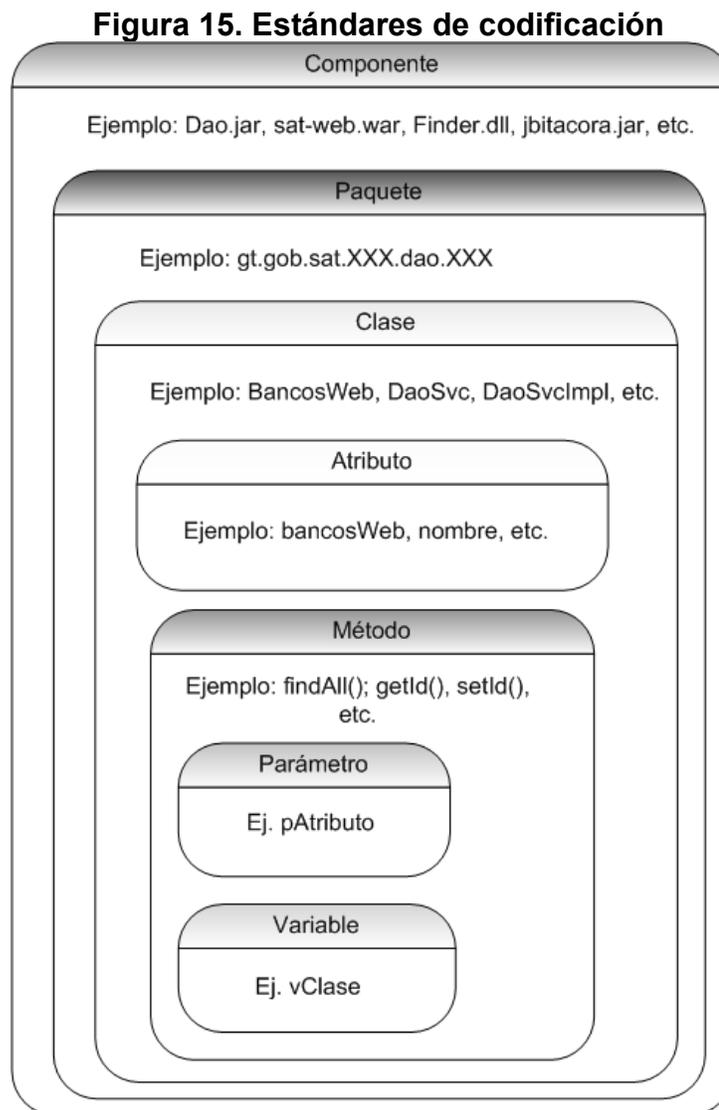


Figura 14. Documentación acerca de una tecnología (II)



3.2.3 Personalizar estándares de codificación

La personalización de este tipo de estándares se efectúa con base a la estructura de proyectos que se tenga en un organización, como por ejemplo, si en la organización se realizan proyectos orientados a capas se puede disponer de sufijos o prefijos como Web, Dao, Svc, Impl, que indican la funcionalidad y objetivo de algún componente o elemento del software. La figura siguiente ejemplifica varios elementos que se pueden personalizar:



3.2.4 Personalizar estándares de nombrado y construcción de proyectos

La personalización de este tipo de estándares en una organización tiene mucho que ver con el dominio de tecnologías que se utilicen, ya que están estrictamente relacionados con las herramientas de edición y construcción de los proyectos (IDE's).

Por ejemplo, si la organización desarrolla en la plataforma J2EE puede estandarizar la forma de construir proyectos a través de varios IDE's, como por ejemplo: Eclipse, JDeveloper, NetBeans, JBuilder, etc., mientras que en la plataforma DotNet se debe usar el IDE por default.

Con todas estas decisiones de alto nivel, se puede elaborar un ambiente de desarrollo para cada analista o desarrollador con el cual se organizará de mejor forma la manera de construir aplicaciones empresariales.

3.2.5 Personalizar estándares de control de versiones

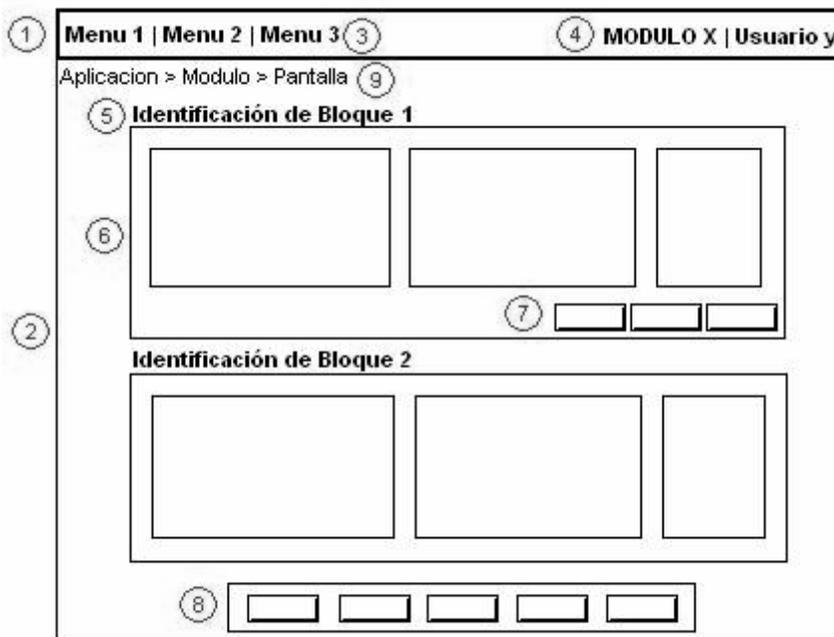
Estos lineamientos se refieren principalmente a la manera en la cual se van a administrar los fuentes de cada proyecto, de manera que muchas personas puedan estar desarrollando una aplicación, con lo que para ello, además del servidor en donde se encuentre el repositorio de información, es necesario tener en cuenta la ubicación en la cual se deben manejar las copias de trabajo y revisiones en los directorios en el repositorio.

3.2.6 Personalizar estándares de interfaces

Los estándares de interfaz son la manera más peculiar con la cual podemos caracterizar a nuestra organización, es decir, la forma con la cual se pueden identificar un conjunto de aplicaciones debido a que la parte de presentación o vista de ellas, presentan la información de la misma manera, colores, estilos, estructura, etc.

Un comienzo puede ser elaborar la estructura de toda interfaz de usuario para el ingreso y consulta de datos, como se puede apreciar en la siguiente figura:

Figura 16. Estándares de distribución de elementos en una interfaz



La forma más viable de empezar este tipo de caracterización es con una especificación de componentes de interfaz, la cual se puede traducir a una guía de usabilidad con varias hojas de estilo.

La siguiente figura muestra un estilo propio para el manejo de catálogos de las aplicaciones Web en SAT:

Figura 17. Estándares de interfaz para catálogos

No	Código	Nombre	Fecha de creación	
1	110	IMPUESTO EXTRAORDINARIO Y TEMPORAL DE APOYO A LOS ACUERDOS DE PAZ	24/08/2006	  
2	2014	Formulario de Test creado por Fritello	24/08/2006	  
3	100	RECIBO DE INGRESOS VARIOS	23/08/2006	  
4	204	REGIMEN SIMPLIFICADO 5%	23/08/2006	  
5	201	DECLARACION Y RECIBO DE PAGO MENSUAL DEL IMPUESTO AL VALOR AGREGADO	22/08/2006	  

Catálogo con iconos

3.2.7 Personalizar estándares de base de datos

Como se mencionó anteriormente, este tipo de estándar uno de los más utilizados y básicamente puede personalizarse utilizando prefijos que identifiquen las aplicaciones para búsquedas más eficientes y un mejor entendimiento de las tablas pertenecientes a los esquemas. Ejemplo:

Figura 19. Especificación de las guías o documentación

The image shows a screenshot of a web page from a virtual library. At the top left is the SAT logo with the text 'SAT' and 'Sistemas de Administración Tributaria' below it. At the top right is the text 'Biblioteca Virtual'. A horizontal line separates the header from the main content area. The main content is enclosed in a large rectangular frame with a dark background. At the top of this frame is a title box with the text: 'Instalación de un Servidor con Subversion, para Control de Versiones, con acceso vía http a través de Apache, en Windows 2003 Server'. Below the title is a dark rounded rectangle containing the text 'Autor: David Oliva' and 'J2EE'. To the left of the main content area are two logos: the Apache HTTP Server logo (a feather) and the Subversion logo (a stylized 'S'). To the right of these logos are four stacked, rounded rectangular buttons with the following text from top to bottom: 'Versión 1.0', 'Biblioteca Virtual', 'Gerencia de Informática', and 'SAT'. A horizontal line is located below the main content area. At the bottom center of the page, below the main content area, is the number '1'.

4 INTEGRACIÓN Y UTILIZACIÓN DEL MODELO

4.1 Configuración e implementación de los elementos (SAT)

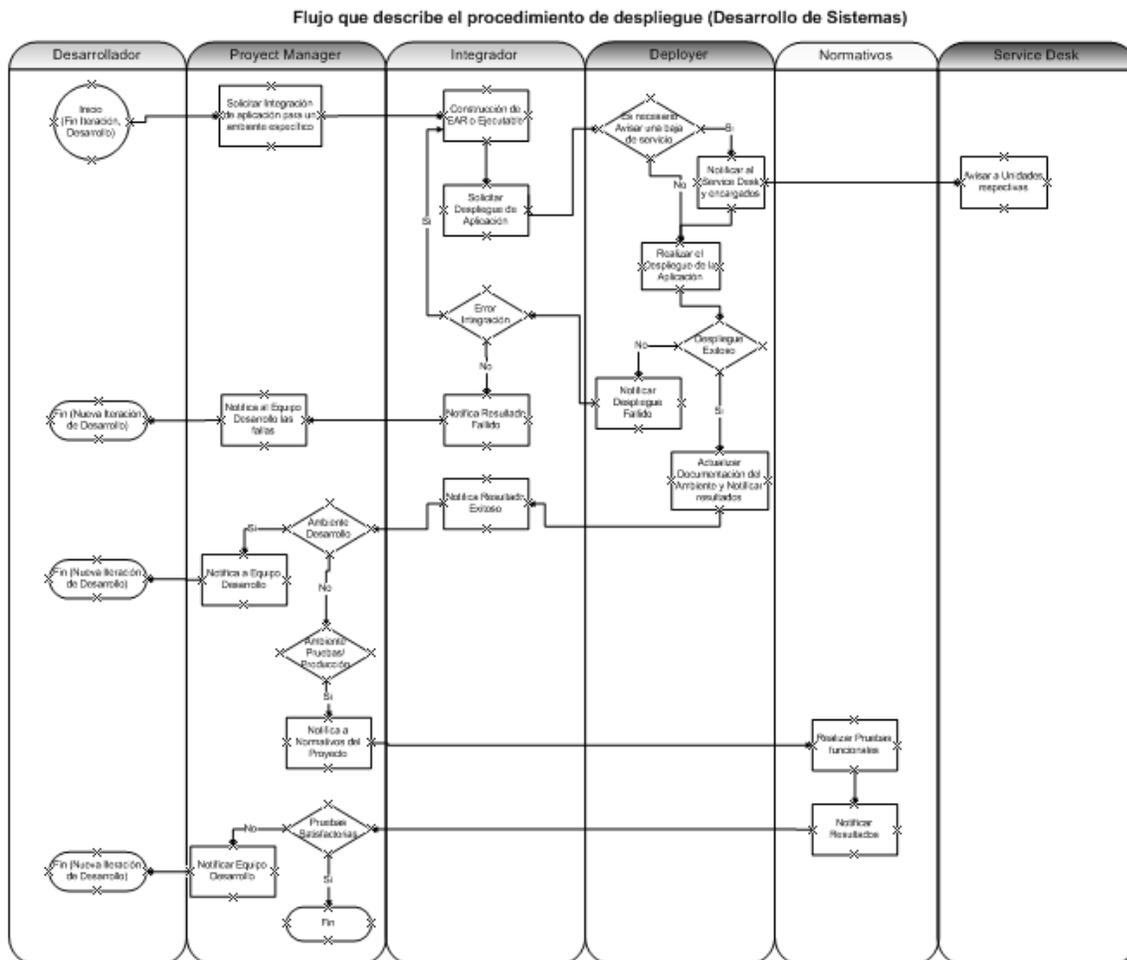
Como se explicaba en el capítulo dos, es necesario tener normado ya sea desde una perspectiva muy general o muy a detalle, todo lo que se realiza en el departamento y normalizar todas las interacciones del mismo con otras áreas relacionadas. En otras palabras, es ordenar lo que estamos haciendo y la forma en la que lo estamos haciendo, tomando en cuenta todas las variantes que pueden existir entre los diferentes escenarios de los diversos proyectos de software que se manejan en el departamento o área, basándose en la experiencia adquirida.

Es necesario que se normen los procesos o procedimientos en donde se vea involucrado el departamento o parte de él, ya que ésta es una manera de saber que alcance tienen todas las actividades del mismo y de que forma se tiene conciencia de éste alcance, del mismo modo que se organiza el estado actual, para poder ver hacia un estado de mejora, a través de que se vaya aplicando el o los procedimientos respectivos.

Siguiendo con el modelo, se realizó la publicación e implementación de un procedimiento entre el departamento de desarrollo y el departamento de operaciones, derivado que es necesario una fuerte interacción entre estos dos departamentos ya que un problema anterior a este cambio era el orden y control de las aplicaciones al pasar al ambiente de producción todos los sistemas y productos de software construidos o modificados por el departamento de desarrollo, constituyendo un mejor orden para realizar dichos cambios y

organizar de mejor manera la trabajar de estos dos departamentos, pudiendo observar un mejor cuidado al momento de realizar cualquier cambio al ambiente y a los sistemas en ejecución.

Figura 20. Flujo de integración y despliegue



Es importante mencionar que éstos procedimientos deben ser de dominio público, al igual que éste mismo se puede medir, tomando en cuenta tiempos y artefactos, además del resultado. Por lo que se hace necesario que dichos procesos o procedimientos hayan sido aceptados por la unidad jerárquica más

grande del departamento o área y expuestos tales procedimientos con el personal involucrado.

Posteriormente a la definición de los procesos y procedimientos, es necesario adjuntar a dichos elementos, los artefactos que contienen o implican. Aquí es donde varían de acuerdo a las necesidades de la organización y éstos deben ser estandarizados para todo proyecto o sistema tal y como se vio en el capítulo anterior.

Siguiendo con la utilización del modelo, luego de tener listos los procedimientos y artefactos, estos deben contar, para ser efectivos a través del tiempo, con un tipo de control de versiones y de cambios. Es decir que yo trabaje sobre una misma base y que pueda ver todos los cambios realizados sobre cualquier activo o artefacto a través del tiempo. Ésta disciplina debe estar ligada a todo artefacto de todo proyecto o sistema que esté involucrado en la organización. Y la herramienta utilizada debe proveer la capacidad de manejo de usuarios y roles para poder acceder a los artefactos e información oportuna.

Para la implementación del repositorio de control de activos se utilizó el Framework de Subversion, que es una herramienta para el control de versiones, la cual debe ser configurada para acceso (total o parcial) de todos los miembros del departamento de desarrollo. Como bien se describió en la definición del modelo es necesario que no existan propietarios de ningún código fuente, de manera que éste sea consultado y enriquecido en cualquier momento por cualquier miembro de un equipo de desarrollo, de modo que el repositorio de activos de los proyectos debe organizarse según las necesidades de los grupos o equipos en el departamento de forma que se organicen pequeños repositorios o bien, alguna jerarquía de los activos respectivos, tal como se observa en la siguiente figura:

Figura 21. Repositorio de proyectos

Revision 131: /trunk

- ..
- E-Servicios/
- SatSeguridad/
- VisaNet-web/
- cb-datasource/
- common-backend/
- jautenticacionWS/
- logging-backend/
- manifiestos-utils/
- mntu-backend/
- mntu-portal/
- mntu-usuarios/
- persistencia-base/
- pki-utils/
- ps-backend/
- ps-datasource/
- rtu-utils/
- vn-backend/

Powered by Subversion version 1.4.0 (r21228).

Para la implementación del repositorio de componentes y releases se utilizó el framework de Maven, que es una herramienta para el control de y distribución de componentes java, la cual, del mismo modo que el repositorio anterior, debe ser configurado para acceso (total o parcial) de todos los miembros del departamento de desarrollo. Es necesario definir el nombrado de éstos componentes, así como la forma de versionamiento y distinción de cambios, que dependerá de la naturaleza de la organización y de los cambios realizados.

Figura 22. Repositorio de componentes

maven-proxy Unversioned				
CONFIG SEARCH REPOSITORY ADMIN				
Browsing /gt/gob/sat/common-backend/1.1.1/				
	Size (bytes)	Last Modified	Name	Repository
	16686	7/1/08 4:15 PM	common-backend-1.1.1.jar	Global Repository
	16686	7/1/08 4:15 PM	common-backend-1.1.1.jar	local-repo
	32	7/1/08 4:15 PM	common-backend-1.1.1.jar.md5	Global Repository
	32	7/1/08 4:15 PM	common-backend-1.1.1.jar.md5	local-repo
	40	7/1/08 4:15 PM	common-backend-1.1.1.jar.sha1	Global Repository
	40	7/1/08 4:15 PM	common-backend-1.1.1.jar.sha1	local-repo
	1378	7/1/08 4:15 PM	common-backend-1.1.1.pom	Global Repository
	1378	7/1/08 4:15 PM	common-backend-1.1.1.pom	local-repo
	32	7/1/08 4:15 PM	common-backend-1.1.1.pom.md5	Global Repository
	32	7/1/08 4:15 PM	common-backend-1.1.1.pom.md5	local-repo
	40	7/1/08 4:15 PM	common-backend-1.1.1.pom.sha1	Global Repository
	40	7/1/08 4:15 PM	common-backend-1.1.1.pom.sha1	local-repo

4.2 Publicación de estándares de elementos y plantillas del modelo

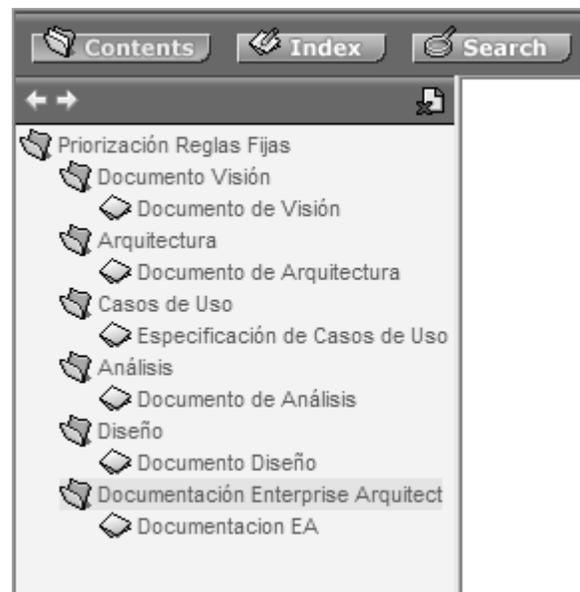
La publicación de estándares y plantillas, proporciona la ventaja de que cualquier desarrollador o analista puede personalizar la aplicación identificando todos aquellos aspectos relevantes en los artefactos durante el desarrollo o implementación de la solución estableciendo una mejor práctica en la organización, sumando a ello que lo más aconsejable es que exista en la BDD de la administración de la configuración un directorio en donde se almacenen todos los proyectos realizados y entregados, como tal es el caso de la Gerencia de Informática en la SAT.

Ya teniendo registrada una aplicación, la Gerencia de Informática de la Superintendencia de Administración Tributaria (SAT) exige a los desarrolladores que expongan la aplicación elaborada por varias razones: primero para que

sirva de retroalimentación para la construcción de una aplicación similar o que vaya a funcionar con la misma información o recursos y segundo para aplicarle control de calidad, y poder mejorar una serie de aspectos relacionados con la arquitectura, tecnologías, componentes, estándares y documentación.

La figura siguiente muestra como es que un proyecto asignado a un desarrollador es publicado con sus artefactos y documentación determinada, sirviendo muchas veces de guía o para retroalimentación de todo el personal de desarrollo.

Figura 23. Publicaciones de proyectos



4.3 Integración con un proyecto real

Para la validación del modelo, se desarrolló una aplicación utilizando los elementos mencionados anteriormente y aplicando los estándares requeridos, personalizando gran parte de ellos y tratando de formar la base con la cual los

desarrolladores de la Gerencia de Informática de la SAT, puedan trabajar y con el tiempo mejorar los productos o aplicaciones pedidas.

El proyecto a desarrollar fue un módulo para la unidad de fiscalización y se trabajó con un equipo de dos analistas, con lo que se obtuvo mucha retroalimentación para la aplicación de los estándares, ya que gracias a ese primer intento, se pudieron observar muchas cosas que hacían falta para que el modelo cubriera la gran mayoría de las necesidades de aplicaciones empresariales grandes.

Los resultados obtenidos referentes a la aplicación del modelo fueron los siguientes:

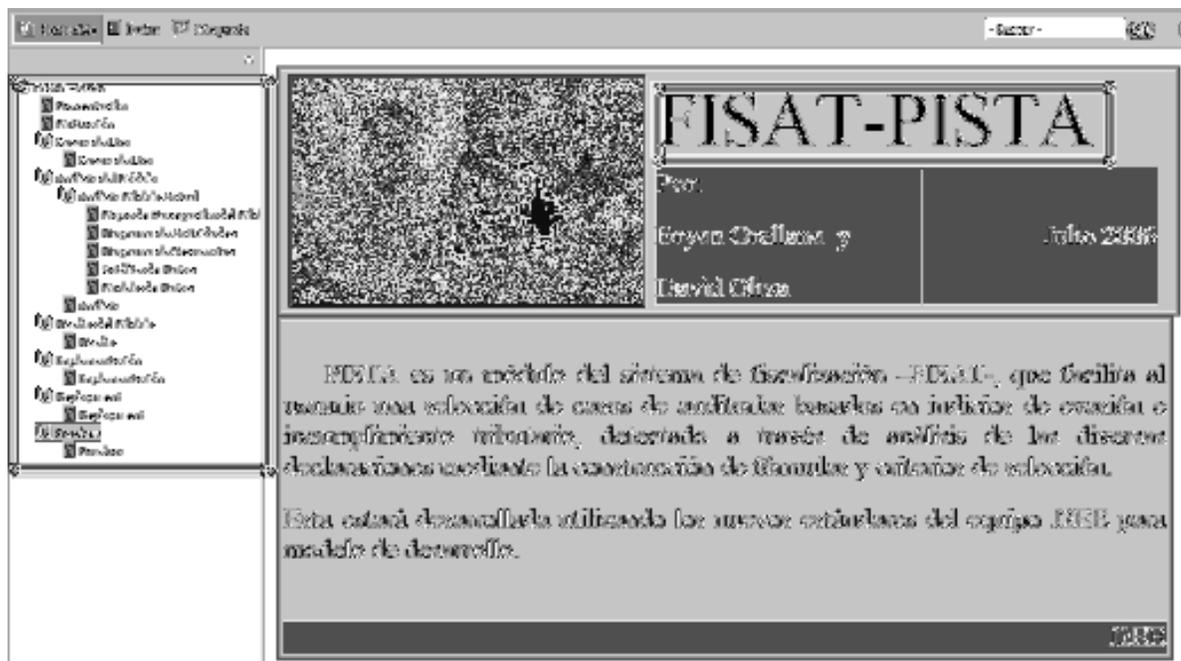
- Mejor organización y orden de las funciones
- Mejor calidad del producto
- Información publicada constantemente para mejora de la aplicación
- Entendimiento de la codificación por parte del equipo
- Desarrollo claro y ordenado
- Desarrollo orientado a componentes (reutilización)
- Desarrollo encaminado a la especialización del analista o desarrollador en cuanto a las tecnologías y procesos de desarrollo.
- Una imagen mejorada del producto.
- Documentación de guías y base de su construcción para efectuar mantenimiento por otros desarrolladores
- Especificación de componentes realizados para despliegue en otros servidores

- Publicación de los archivos fuentes para la reutilización de algún patrón de diseño, o bien, reutilización de código.
- Mejor entendimiento de la aplicación por parte de los normativos y usuarios clientes.
- Aumento en la eficiencia de la captura de requerimientos, debido a que la información del proyecto es pública y se actualiza constantemente según el avance.
- Retroalimentación para todo el equipo y exteriorización del conocimiento tácito constituido en dicho equipo.
- En general, se concluyó que todo el modelo constituye una serie de mejores prácticas las cuales pueden personalizarse o caracterizarse según el estado y características de los proyectos y de la organización, con el fin de obtener una constante mejora de los productos elaborados (software), a la vez que se incrementa la posibilidad de aplicarle mantenimiento o realizar cambios por parte de un desarrollador que no necesariamente estuvo en el equipo de construcción del proyecto.

La experiencia con el desarrollo de esta aplicación ha servido como base para poder transmitir este mecanismo de trabajo a los demás desarrolladores del departamento o área de desarrollo de sistemas en la Gerencia de Informática en la Superintendencia de Administración Tributaria (SAT), con el objetivo de ser cada día más competentes y poder cumplir con los requerimientos y necesidades de dicha institución, realizando para ello productos de buena calidad y sin que las aplicaciones se aten a una o varias personas, ya que la estandarización rompe por completo ese viejo paradigma y provee de un mecanismo mucho más estructurado, ordenado y organizado para el desarrollo.

La siguiente figura ilustra el modo en que se publica la información y documentación del proyecto proponiendo el mismo esquema (links en la parte izquierda para navegación y el contenido del link en la parte derecha.):

Figura 24. Matrícula de un proyecto



4.4 Observaciones acerca de la integración y utilización del modelo

Se pudieron identificar una serie de características y una serie de tópicos a tomar en cuenta en una implantación de similares magnitudes. Para empezar es necesario tener muy bien delimitado el alcance que se espera tener al implementar este modelo, con lo que se podrá delimitar las personas involucradas, actividades e interacciones del departamento o área de desarrollo y mantenimiento de sistemas de información. Para luego poder efectuar una

planificación de toda la implementación tomando en cuenta que éste será solamente un bosquejo que represente el camino desde una perspectiva inicial, ya que esta planificación puede ir cambiando de acuerdo a los obstáculos encontrados. Es recomendable tomar como básico 4 etapas, correspondientes a los 4 elementos descritos. Esto puede tener una mejor aceptación por las personas involucradas y además no se sobrecarga de trabajo al personal tomando en cuenta que ellos aplicarán por partes el modelo y no todo junto.

Primordialmente, si fuese posible es recomendable una especie de moderador o equipo moderador, dependiendo de las magnitudes de las organizaciones, para que pueda responder a la mayoría de preguntas o inquietudes que puedan darse lugar en el transcurso de la implantación y que ésta no sufra atrasos debido a la mala documentación que se pueda tener de algún paso en un momento determinado.

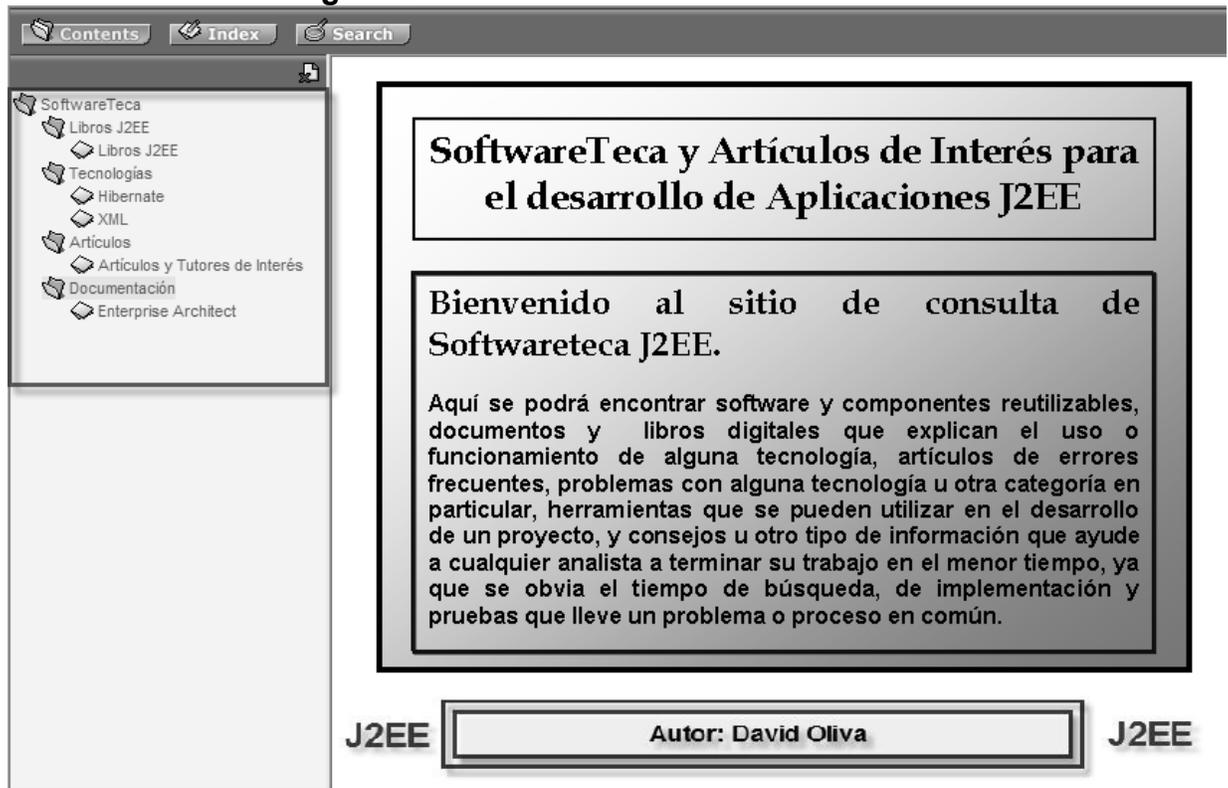
En el transcurso del desarrollo de las aplicaciones, se notó que regularmente, al montar el ambiente de desarrollo o la estructuración de las capas e inclusión de las nuevas tecnologías, los analistas presentaban un patrón referente a los problemas que tenían acorde a la aplicación de ciertos lineamientos o con las tecnologías que estaban utilizando, con lo que se planteó la idea de crear un sitio en donde estuvieran almacenadas todas las preguntas o problemas referentes a la migración de las tecnologías y a la aplicación de los estándares del modelo, con lo que se aprovechaba esa retroalimentación con los analistas y se crecía en nuestra experiencia en la utilización del modelo.

Tal sitio debería de tener las características de un foro y con el apoyo para brindar la documentación necesaria para cada tipo de problema, con el fin de servir de apoyo en la construcción de cualquier proyecto con los lineamientos del modelo y sin tener la experiencia en la elaboración de alguno. Con lo que se

optó de igual forma que todos los estándares, a publicar el sitio cuyo nombre fue 'Biblioteca Virtual'.

La figura siguiente ilustra la portada de dicho sitio y presenta el mismo esquema de publicación que todos los anteriores:

Figura 25. Softwareteca de desarrollo



Todo lo anterior propuesto es necesario estar expuesto a todo el equipo y contar con las herramientas necesarias para aplicar la gestión de conocimiento (portal, foro, flujos, referencias, etc.), de forma que se promueva y se extienda el conocimiento y la mejora continua a través del departamento respectivo y demás unidades de la organización. En SAT, se utiliza un manejador de contenidos como lo es SharePoint para la exposición y herramientas respectivas, tal como se puede observar en la siguiente figura:

CONCLUSIONES

1. Se elaboró un modelo que provee de una serie de mecanismos por cada elemento que lo compone, los cuales propician a una mejor administración, configuración, construcción y mantenimientos de los sistemas de software. Constituyendo un método por el cual, las organizaciones amplían la capacidad y conocimiento en lo que respecta a los sistemas de información, ya que ellos actualmente representan altos costos y variabilidad en tanto a calidad y tiempo de construcción. Los elementos del modelo presentarán, con el tiempo de implantación del mismo, un crecimiento y madurez de las personas y equipos asociados al departamento de desarrollo, además de todas las interacciones con los departamentos asociados, suministrando un ambiente propicio de trabajo con alta calidad.
2. La definición del modelo fue elaborada a través de un análisis de un conjunto de problemas propios de las organizaciones que cuentan con áreas o departamentos dedicados a proveer y mantener servicios de desarrollo y mantenimiento de sistemas de software, ya que actualmente en el país esta área toma una gran importancia en la prestación de una serie de servicios de las organizaciones, constituyendo una línea de crecimiento muy prometedora. Dichos problemas tienen una correlación entre los paradigmas actuales de gestión de proyectos de software y la manera clásica de administración de las organizaciones. El Modelo planteado corrige ciertos tópicos de la administración clásica y propone una modalidad basada en el conocimiento y aprendizaje de toda la estructura y personal involucrado en éstas áreas. Del mismo modo que

se hace evidente, con el tiempo, la búsqueda de una mejor calidad y organización de la estructura y equipos asociados.

3. El modelo propuesto propone a través de sus cuatro elementos, un mapa y una guía para poder iniciar a organizar la forma en que se administra, construye y se mantienen los sistemas de información. Planteando para ello una manera de implementar dicho modelo, comenzando con la gestión de procesos para poder seguir luego con la estandarización de todos los elementos involucrados en los procesos definidos y posteriormente definir la manera de controlar todo tipo de cambios sobre éstos artefactos, teniendo la gestión de conocimiento como un elemento transversal a los tres anteriores.
4. Es muy importante enfatizar que el modelo propuesto, capta la mayoría de necesidades encontradas en las deficiencias y problemas actuales, las cuales pueden ser o no, que afecten a una organización en el presente tiempo o que sean una herramienta para eliminar o minimizar alguna deficiencia en el futuro, conforme se utilice el modelo. Cada una de estas necesidades, se ven reflejadas en algún método de algún elemento o en varios artefactos de apoyo a los procesos, los cuales pueden ser personalizados o modificados de acuerdo a las necesidades de la organización. Esta personalización puede ir desde algo muy sencillo, como los colores de la organización plasmados en una hoja de estilo, hasta algo más complejo, como el seguimiento de un procedimiento legal para adquirir software y darle mantenimiento en la institución.

5. A partir de la definición del modelo se procedió a la identificación de los atributos más importantes, normas generales y aspectos relevantes de la institución elegida para el estudio, en lo que concierne a desarrollo y mantenimiento de sistemas de software, con los que se definieron la mayoría de procedimientos y estándares a usar.
6. La implantación del modelo en la organización SAT fue realizada parcialmente en varios proyectos y tuvo un resultado observable luego de una serie de iteraciones, tomando en cuenta aspectos tales como la magnitud, importancia, prioridad y naturaleza de los proyectos en cada iteración con lo que resultaba un conjunto de ajustes al modelo, presentando un nivel de madurez mayor para el siguiente proyecto, observándose como principales avances, el tiempo de desarrollo, los tiempos de respuesta ante cambios o problemas y el aprendizaje dichos proyectos o sistemas de información.
7. Para lograr la implantación de este modelo, es aconsejable que se realice de forma gradual, elemento por elemento, fijando plazos para su integración y utilización, ya que institucionalizar todo el modelo en un instante en el tiempo puede causar más desorden del que se pretende corregir y eventualmente baja productividad y mala calidad en los sistemas de software.
8. Para que este modelo sea exitoso, lo primero que debe tomarse en cuenta para la implantación, es contar con el entero apoyo de la unidad jerárquica superior o los altos mandos, ya que la curva de aprendizaje puede variar dependiendo de la magnitud de los departamentos y de los sistemas de información que se manejen. Por lo que éstos deben estar al tanto y de igual forma apoyar enteramente los cambios y nueva forma de

trabajo, aunque esto al inicio cause una serie de contratiempos o atrasos a diferencia de lo cotidiano.

9. Es importante resaltar que ante la gran variedad de herramientas y utilidades que se encargan del manejo y difusión de la gestión de conocimiento, debe establecerse un marco general que permita su clasificación indicada e integración dentro del proceso general de desarrollo y mantenimiento de sistemas de información, es decir, plantear un esquema de aprendizaje general y varios objetivos que conlleven al elevamiento de los activos intangibles y el capital intelectual de todo el equipo involucrado.

RECOMENDACIONES

1. El aspecto de calidad presenta un área de oportunidad para cualquier estudio de investigación, debido a que es un concepto que debería estar presente en toda actividad y proceso para estar evolucionando continuamente a favor de la optimización, por lo que es necesario mantener siempre una o varias personas en constante investigación e implementación de nuevos aspectos para el control de calidad de los proyectos.
2. Para todo departamento o área de una mediana o grande organización que tenga recursos en el ámbito de Tecnología de la Información, se debe establecer un modelo de desarrollo de software. La solución recomendada por el autor es seguir los lineamientos dados en este trabajo y personalizarlo, según los atributos y características de la organización propia con el fin de ir en el camino de la optimización y mejora en cuestiones de calidad en los proyectos o sistemas desarrollados.
3. Para el seguimiento y aplicación de esta guía, podría ser necesario la utilización de cierto personal capacitado para llevar con éxito la transición durante la etapa de inducción y asimilación del modelo, ya que entre mayor es el número de personas en el área o departamento de desarrollo de sistemas, mayor aún es el esfuerzo para lograr obtener un cambio en el paradigma de desarrollo, tal y como fue el caso en la organización que sirvió de referencia para este trabajo (SAT).

4. El proceso para el mejoramiento de calidad es continuo y nunca se debe de estar satisfecho con el resultado, ya que constantemente surgen herramientas o aspectos con los cuales se pueden obtener mejores resultados.

5. Para la realización de ciertos elementos como el de gestión de procesos, es posible que surja la necesidad que se asigne a un grupo de expertos en las diversas actividades del departamento o área de desarrollo para identificar los aspectos relevantes y los diversos caminos producto de una serie de escenarios previstos, para definir los procesos, lo más cercano a la realidad y que no exista una serie de cambios sobre éstos ya en el tiempo de aplicación. Es recomendable que las primeras versiones sean lo más parecidas a la realidad, pero también pueden aplicar ciertas prácticas que se quisieran aplicar en la primera iteración de éstos, tomando en cuenta que el cambio debe ser gradual y no aplicar un cambio de estrategia total en las actividades, ya que esto provocaría seguramente una resistencia al cambio o bien una mala aplicación de los mismos.

6. Una sugerencia en la definición de todos estos elementos y de la metodología a seguir, es que debe tenerse muy en cuenta el tamaño, magnitud y alcance de los proyectos y sistemas de información que se manejan, ya que es imprescindible que ésta definición vaya acorde a ello, debido a que para proyectos pequeños y medianos es recomendable utilizar un tipo de metodología ágil y procedimientos más orientados a los objetivos, ya que agilizarán los resultados; y para proyectos grandes utilizar una metodología predictiva y procedimientos en donde resalten los artefactos y un fuerte control sobre ellos, ya que

estos proyectos por lo regular son muy extensos y puede que el personal o conocimiento del mismo vaya esparciéndose a lo largo del tiempo.

7. Sería aconsejable luego de tener dominado el modelo, que se indague, investigue y analice sobre la posibilidad de poder llevar el control de métricas de los proyectos, debido a que se va a llevar un control estricto en cuanto a planificación, recursos e implementación de las tecnologías y proyectos, bien puede obtenerse información que sea de utilidad para la planificación y que ayude a tomar decisiones en cuanto al número de personas que debe trabajar en conjunto para el desarrollo de una aplicación en un tiempo determinado, con el fin de ir en una actividad constante de mejora del modelo.

8. Sería aconsejable luego de tener dominado el modelo, que se indague, investigue y analice sobre la posibilidad de poder llevar el control de métricas de los proyectos, debido a que se va a llevar un control estricto en cuanto a planificación, recursos e implementación de las tecnologías y proyectos, bien puede obtenerse información que sea de utilidad para la planificación y que ayude a tomar decisiones en cuanto al número de personas que debe trabajar en conjunto para el desarrollo de una aplicación en un tiempo determinado, con el fin de ir en una actividad constante de mejora del modelo.

9. El aprendizaje y la gestión de conocimiento en las organizaciones es una necesidad latente hoy por hoy, por lo que el modelo planteado para la administración del conocimiento debe ser extendido por todos los procesos de la organización, para lo que creará un ambiente propenso a

mejoras e innovación, aumentando el valor de los activos en la institución (activos tangibles e intangibles).

10. La mejor recomendación para empezar a organizar los proyectos, es llevar un ciclo de integración sobre una plataforma estandarizada y unificada, en la que las aplicaciones estén totalmente desacopladas, el desarrollo pueda o no ser distribuido, y que se cuente con un proceso bien identificado y documentado (workflow), para el desarrollo, mantenimiento, integración y despliegue de aplicaciones.

BIBLIOGRAFÍA

1. Varas C. Marcela. **Gestión de proyectos de desarrollo de software**. Chile, Universidad de Concepción, Facultad de Ingeniería, 2000. 81 páginas.
Disponible en la World Wide Web:
<<www.inf.udec.cl/~mvaras/gpis/apunteGPDS.pdf>>
2. Peña Rodrigo. **Gestión de proyectos**. Chile., Gestiópolis, 2002.
Disponible en la World Wide Web:
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/gestio_proyecto.htm>
3. Nonaka, Ikujiro, and Hirotaka Takeuchi. 1995. **The knowledge-creating company: how japanese companies create the dynamics of innovation**. Oxford University Press. 284 páginas.
4. Barreiro Enrique Alonso. **La calidad del software**. España., Universidad de Vigo, Departamento de Informática, 2006. 31 páginas
Disponible en la World Wide Web:
<<http://trevinca.ei.uvigo.es/~ebalonso/asignaturas/esx/transparencias/esxTransp7.pdf>>
5. Wikipedia. **Control de versiones**. España., Wikipedia 2008.
Disponible en la World Wide Web:
<http://es.wikipedia.org/wiki/Control_de_versiones>

6. Sena Medellín. **Cluster**. Chile., Sena Medellín, 2008.
Disponible en la World Wide Web:
<<http://slcolombia.org/SenaMedellin/941575/Aprendices/AndresRestrepo/CLUSTER>>

7. X-DT. **Administración del proyecto vs ciclo de vida del proyecto.**,
Project Management Institute, 2006.
Disponible en la World Wide Web:
<<http://www.tenstep.com.mx/Paso0.0.1.2.asp>>

8. Mendoza Sanchez, María A. **Metodologías de desarrollo de software.**
Perú., Team Soft, 2004.
Disponible en la World Wide Web:
<http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html>