

Mejora Del Proceso De Desarrollo De Software Con Un Enfoque Al Cliente En Las Empresas De La Industria TI Colombiana.

[Process Improvement Software Development With A Customer Focus In The Companies Of The TI Colombian Industry.]

Angélica Rocío DULCEY SEPÚLVEDA, Hernán Andrés GALEANO ALFONSO, Vivian Elina GÓMEZ VEGA, Julián Ricardo ROMERO SARAY, Cindy Marcela SIERRA RODRÍGUEZ.

Artículo de investigación Científica y tecnológica.

Resumen

En la actualidad, debido a la globalización de la economía y al auge de la era tecnológica, las organizaciones se han visto obligadas a realizar cambios en sus esquemas estructurales y operacionales con el fin de mantener el nivel de competitividad de sus productos en el mercado. Hoy en día, las empresas que no cuenten con sistemas de información en sus procesos internos y externos están un eslabón abajo con respecto a sus pares que sí cuentan con todo tipo de herramientas tecnológicas que les permitan hacer más ágil sus operaciones. Con el incremento de estas necesidades, se dio origen a la generación de empresas desarrolladoras de software, las cuales en su afán por entrar a acaparar la demanda de este nuevo mercado, se centraron en crear empresas que pudieran desarrollar sistemas, perdiendo de vista su organización interna y no teniendo como prioridad un enfoque hacia el cliente.

En este artículo, se realiza un análisis de la estructura organizacional y de los procesos que maneja la empresa SCI Software Ltda., con el fin de identificar las principales fallas que cometen este tipo de empresas en la producción y prestación de sus servicios, para así poder brindar una solución transversal a una de las principales dificultades para estas organizaciones: el incumplimiento en la entrega del producto final al cliente debido a problemas de comunicación tanto internos como externos dentro de la organización. Para lograr este objetivo se utilizaron herramientas del Lean Management, el cual es un sistema de mejoramiento de procesos, mediante la identificación y eliminación de operaciones y actividades que no generan un valor agregado y que el cliente no está dispuesto a pagar. Finalmente, se obtuvo una metodología en la que se propone una estructura organizacional horizontal, la creación de equipos de trabajo multidisciplinarios, y la creación de plantillas dinámicas para la toma de requerimientos, todo esto encaminado a la comprensión gradual de las necesidades del cliente, desde lo mínimo hasta lo macro.

Palabras claves: Desarrollo de Software, Lean Management, Equipos Multidisciplinarios, Problemas de Comunicación.

Abstract

Nowadays, due to globalization of the economy and the rise of the era of technology and information, organizations have been forced to make changes in their structural and operational schemes in order to maintain the level of competitiveness of their products on the market. Today, companies that do not have information systems in their internal and external processes are a link below with respect to their peers who do have all kinds of technological tools that allow them to make their operations more agile. With the increase of these needs, it gave rise to the generation of software development companies, which in their eagerness to get to monopolize the demand for this new market, focused on creating businesses that could develop systems, losing sight of their organization internal and not having priority a focus on the customer.

In this article, An analysis is performed about the organizational structure and processes that manages the company SCI Software Ltda., in order to identify the main faults that make these businesses in the production and delivery of its services, and to provide a cross solution to one of the main challenges for these organizations: the failure to deliver the final product to the customer due to problems both internal and external communication within the organization. To achieve this goal, Lean Management tools were used, which is a system process improvement, by identifying and eliminating operations and activities that do not generate added value and the customer is not willing to pay. Finally, was obtained a methodology in which a horizontal organizational structure is proposed, creating multidisciplinary working groups, and the creation of dynamic templates for making requirements, all aimed at the gradual understanding of customer needs, from the least till the macro.

Keywords: Software Development, Lean Management, Multidisciplinary teams, Communication Problems.

0. Introducción

El siglo XXI se ha denominado como el siglo del conocimiento y la revolución tecnológica, en los últimos años, la economía mundial ha pasado de la dependencia de los recursos naturales al mercado tecnológico e informático, los países denominados potencias mundiales invierten un alto porcentaje de su PIB en los rubros de educación, innovación y desarrollo, buscando que las empresas sean altamente competitivas y que contribuyan al crecimiento del país. India ha realizado este tipo de inversiones desde la década de los ochenta, para el 2015 se posicionó como la novena potencia del planeta, además de contar con un crecimiento del 7,3% anual del PIB [1], y una tendencia creciente.

El crecimiento del sector de la tecnología y la información ha generado un aumento en la dinámica de la economía mundial. Este factor ha desencadenado cambios en el modelo estructural de las organizaciones, innovando en las formas de producción e interacción entre las partes internas y externas de las mismas. La elevada cantidad de datos que en la actualidad se manejan requieren de un procesamiento ágil y dinámico que permita la transmisión de información en tiempo real y contribuya en la toma de decisiones dentro de las organizaciones.

En Colombia, el sector de la tecnología ha tenido su mayor crecimiento en los últimos cuatro años, reportando para el año 2015 ventas por un monto cercano a los \$20.396.679.503. Una de las actividades que ha tenido un crecimiento positivo ha sido el desarrollo de software, contribuyendo con un 13,2% de las ventas a nivel nacional para el año 2015 [2], prestando sus servicios al sector de telecomunicaciones, finanzas, gobierno, consumo masivo y manufactura. El país se encuentra en el inicio de un proceso de transición desde una economía dependiente de la transformación de productos tangibles provenientes del sector primario, a la producción de servicios enfocados al sector de la tecnología. Sin embargo, lograr que el mercado colombiano trascienda de la producción basados en recursos naturales, al desarrollo de productos tecnológicos con altos estándares de calidad, requiere alinear las diferentes partes de las organizaciones bajo las necesidades y especificaciones de los clientes de software e innovar en este sector.

El presente artículo corresponde al diseño de una propuesta de mejoramiento al proceso de desarrollo de software, el cual impacta en la estructura organizacional y los métodos de interacción entre los procesos que se ejecutan en las empresas TI, esto, con el fin de garantizar la comunicación asertiva y el adecuado flujo de información dentro de este tipo de organizaciones.

La comunicación asertiva en las organizaciones se convierte en un mecanismo que permite la transferencia de información a través de canales específicos, el adecuado flujo contribuye a una pronta identificación de fallas y una toma de decisiones efectiva. Una empresa mantiene su nivel de competitividad en el mercado cuando desarrolla productos y servicios que superan las expectativas de sus clientes, debe estar al tanto de las innovaciones y actualizaciones de su sector y mejorando constantemente en la prestación del servicio a sus clientes, puesto que de ellos es de donde provienen los ingresos para su organización, consideramos que esto se puede lograr mejorando el flujo de comunicación en la organización.

Para desarrollar la propuesta del presente artículo se analiza la situación actual de la empresa Sistemas Consultoría e Informática Ltda. (SCI Software), en la cual se analizaron las actividades que se ejecutan en el proceso de desarrollo de software a fin de identificar las fallas presentadas en el proceso.

Sistemas Consultoría e Informática Ltda. (SCI Software) es una empresa del sector informático fundada el 26 octubre del 2007, especializada en brindar acompañamiento a diferentes empresas, brinda servicios de consultoría informática, desarrollo software a medida, inteligencia de negocios, afinamiento de bases de datos y servicio de Outsourcing informático. El desarrollo de Software es uno de los productos de mayor demanda para la compañía. Estas aplicaciones van desde software para procesos de ventas, inventarios, compras, clientes, radicación y gestión de PQR's, entre otros. Los servicios ofrecidos por esta compañía hacen parte de los productos de tecnología e información que hoy en día contribuyen a la mejora de la productividad en las empresas; los entes económicos que cuentan con este tipo de servicios mejoran su productividad aproximadamente en un 600 por ciento [3].

El desarrollo de software a medida se caracteriza por ser un producto único, puesto que se construye para las necesidades de un ente económico específico. Para realizar un desarrollo es necesario comprender el mercado o negocio que desarrolla el cliente, las necesidades que tiene, las mejoras propuestas y las expectativas que motivan la producción del software. Dada la característica de este producto se incurren en dos variables que juegan con la satisfacción del cliente y la viabilidad del negocio, estas son: el grado de cumplimiento de las expectativas del cliente, y el tiempo de desarrollo.

En el análisis realizado en dicha empresa se identificaron fallas funcionales, de comunicación y de estructura a lo largo de todo el proceso, este inicia con la presentación de productos y servicios al cliente, pasando por las fases de difusión interna de la información, la generación de una propuesta comercial y finalizando en la producción del servicio requerido.

1. Fundamento Teórico

La evolución tecnológica impulsada desde comienzos del presente siglo por una era del conocimiento en constante crecimiento ha generado un cambio drástico en la forma en que las personas realizan sus labores cotidianas debido al auge de sistemas de computación que han facilitado y agilizado los quehaceres de la actividad diaria. Dentro de esos nuevos sistemas generados se encuentra el desarrollo de la ingeniería del software, esta disciplina nacida desde hace más de 40 años ha teniendo un progreso lento en su proceso de consolidación como una disciplina sólida. Parte de su retraso se debe a sus altos índices de fracaso en sus proyectos, fracasos en los que factores como el humano y los procesos que se desarrollan han tenido un nivel de participación bastante alto [4].

De acuerdo al estudio The Chaos Report elaborado por Standish Group International en 1996, concluyó que sólo un 16% de los proyectos de software son exitosos, es decir que cumplen con los tiempos, costos y requerimientos estimados, el 53% superan las condiciones iniciales y el resto no llegan al término [5]. Unas de las deficiencias que se presentan con mayor frecuencia en el desarrollo de software son:

- ❖ Estimaciones imprevistas de tiempo y costos.
- ❖ Cambios en las especificaciones iniciales.
- ❖ Ausencia de especificaciones técnicas.
- ❖ Insuficiencias en la documentación del proyecto.
- ❖ Ignorar lo obvio.
- ❖ Escasa validación con el cliente.
- ❖ Control deficiente en el progreso del proceso de desarrollo.

En Colombia la industria del software es considerada como una industria desarticulada, si bien es una industria que ha tenido un crecimiento importante, en su proceso de posicionamiento en el mercado a nivel regional e internacional ha presentado falencias de sincronía, de trabajo en equipo, y especialmente de comunicación entre sus partes interesadas, ocasionando un desaprovechamiento del potencial del sector [6].

Partiendo de estos registros negativos se han realizado estudios a nivel nacional y regional identificando claves a nivel organizacional de proceso para lograr el éxito en los proyectos de desarrollo de software, estos estudios mencionan que las principales fallas que se cometen en las empresas desarrolladoras de software se encuentran en tres factores: (1) interacción con el cliente, (2) falta de una estructura organizacional clara y flexible, (3) y desorganización en sus operaciones internas [7, 8, 9].

Actualmente, los procesos encaminados al desarrollo de software no son procesos que cuenten con una estandarización determinada. Por el contrario, los equipos que hacen parte de estas empresas eligen las actividades apropiadas

para dicho desarrollo. Sin embargo, en estos procesos se debe encontrar presente actividades enfocadas en:

- a. Comunicación: Tiene una importancia crítica, ya que por medio de ella se busca captar inicialmente los requerimientos del cliente, para luego ser transmitidos al equipo desarrollador.
- b. Planeación: Por medio de ésta, se crea un plan que puede brindar una guía al equipo; este plan busca definir tareas técnicas, recursos, cronograma, riesgos probables, etc.
- c. Modelado: Busca graficar el panorama general con el fin de entender mejor los requerimientos y el diseño.
- d. Construcción: Corresponde al desarrollo del software y las pruebas requeridas para encontrar fallas.
- e. Despliegue: Corresponde a la etapa donde el software es entregado al usuario para que sea evaluado y de esta forma pueda retroalimentar a la compañía.

No obstante, para que estas actividades permitan que los procesos se desarrollen de forma adecuada es necesario establecer vínculos entre cada proceso, convirtiéndose en un trabajo interrelacionado, donde cada vínculo tiene una causa-efecto. El propósito final de ello, es asegurar el desarrollo de los procesos de forma coordinada, mejorando efectividad, satisfacción, entre otras [10].

Los procesos relacionados con el desarrollo de software tienen como apertura el trabajo en conjunto con el usuario final, quién es el que define las necesidades del proyecto. En esta apertura se describen escenarios de uso, se definen las características, funciones y limitaciones del proyecto. Sin embargo, comprender estos requerimientos es uno de los problemas principales que enfrenta cualquier empresa desarrolladora de software, debido a que la mayoría de los desarrolladores pretenden llegar al diseño y a la construcción del software, sin antes haber entendido la verdadera necesidad. Esta comprensión abarca un conjunto de técnicas, en donde se relaciona la comunicación asertiva y continúa [10].

La Ingeniería de Requerimientos es un mecanismo que “Ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”. [10]

Este mecanismo incluye siete etapas fundamentales que se adaptan a las necesidades puntuales de los proyectos:

- a. Concepción: Los proyectos de las compañías de Software inician con la identificación de una necesidad, el equipo realiza un análisis de las necesidades, define el alcance del proyecto y determina la descripción funcional del alcance. Sin embargo la información que se entienda está sujeta al cambio; es por eso que la Concepción del Proyecto va ligada con

el entendimiento del problema. El éxito de la solución que el cliente desea está en la eficacia de la comunicación entre los participantes.

- b. Indagación: la búsqueda de la información necesaria para determinar qué es lo que el cliente desea y como se soluciona va atada a la forma organizada para obtener los requerimientos.
- c. Elaboración: la información interpretada en la Concepción e Indagación se desarrolla en una propuesta. Allí se especifica cada requerimiento, así como los aspectos de funcionalidad.
- d. Negociación: de acuerdo al orden y a la priorización de los requerimientos, se evalúan los costos y los riesgos. En este punto se pueden eliminar, combinar o modificar requerimientos a fin de que cada parte obtenga su grado de satisfacción.
- e. Especificación: puede corresponder a un documento escrito, un modelo gráfico, matemático o un prototipo.
- f. Validación: es esencial que la calidad del producto se refleje. Por ello el análisis de la especificación es vital para garantizar que no existan inconsistencias, omisiones y/o errores y que el producto sea conforme a los requerimientos del usuario.
- g. Administración de los Requerimientos: durante el proceso de desarrollo, la información está sujeta a cambios, es por esto que el equipo a través de una adecuada administración de los requerimientos identifican, controlan y dan seguimiento a los requerimientos y a los cambios durante el desarrollo del proyecto.

Así pues, es de vital importancia que las empresas desarrolladoras de software adquieran conocimiento que les permita comprender de una forma completa las necesidades los clientes y/o usuarios. De igual forma, hacen énfasis en la adopción de herramientas de comunicación y coordinación que permitan a los participantes colaborar e interactuar de manera efectiva. [10]

Los equipos multidisciplinarios logran tener una comunicación más asertiva con el cliente, los integrantes al tener diversas capacidades técnicas, habilidades y conocimientos pueden llegar a generar variedad de ideas y soluciones para gestionar los requisitos de los clientes. Así como también, aportará en la adecuada comunicación y comprensión de los requerimientos.

Para gestionar de una forma adecuada los equipos multidisciplinarios es necesario definir qué es un equipo de trabajo y qué diferencia hay con los grupos de trabajo.

Un grupo de trabajo es entendido como un conjunto de personas que tienen una formación similar y desempeñan una labor similar dentro de la compañía. Mientras que, un equipo de trabajo es un conjunto de personas que

participan en un mismo proyecto y cuentan con una responsabilidad compartida ante la labor desempeñada. Los equipos de trabajo son conformados por un líder y una serie de colaboradores con conocimientos, habilidades y capacidades diferentes entre sí. Cada integrante es complementario y dependen uno del otro, por lo que se requiere coordinación, sincronía, etc. [12]

1.1 Antecedentes

La mayoría de las empresas desarrolladoras de software cuentan con una estructura organizacional vertical, donde se elige a una persona como responsable de un área funcional, con uno o varios subordinados que cumplen cada uno determinadas funciones. En esta compañía la cadena de mando se despliega desde los cargos superiores a los cargos inferiores.

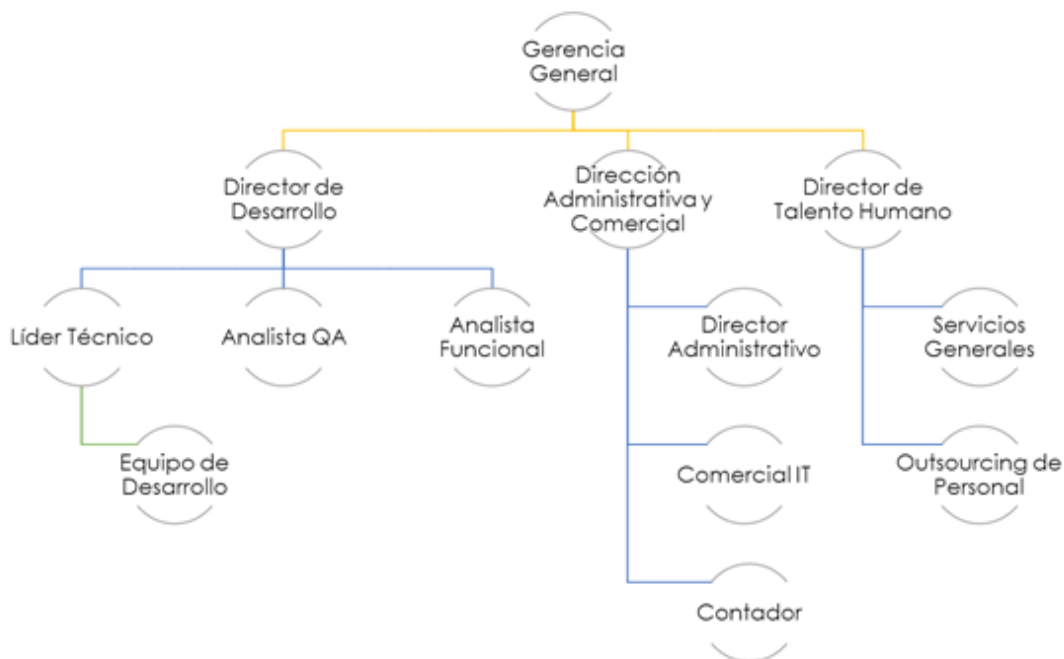


Figura 1. Estructura organizacional de la empresa SCI Software. Fuente: SCI Software.

En el Proceso Misional de la compañía intervienen los siguientes roles:

- ❖ **Gerente Comercial IT:** Busca oportunidades de negocio, realiza la planificación operativa comercial y participa con las diferentes áreas que están involucradas en la generación de propuestas, con el objeto de desarrollarlas con una visión integral y de calidad a clientes.
- ❖ **Líder de Desarrollo:** Garantiza la ejecución del proyecto a través del seguimiento constante del el equipo de trabajo hasta llevarlo a su terminación.
- ❖ **Analista Desarrollador:** Desarrolla Sistemas de Información, haciendo uso de las herramientas y metodologías de desarrollo adecuadas.

- ❖ **Analista QA:** Ejecuta las actividades de tésting, set de pruebas y soporte de aplicaciones desarrolladas.
- ❖ **Analista Funcional:** Elabora el análisis funcional de nuevas aplicaciones a desarrollar. Controla, analiza y supervisa el desarrollo funcional de las aplicaciones informáticas, asegurando su correcta explotación y su óptimo rendimiento, de acuerdo a las especificaciones del cliente.

1.2 Proceso

El proceso que se realiza en SCI Software con sus respectivos documentos de salida son:

a. Presentación de productos y servicios al cliente

El representante comercial contacta a los clientes, presenta el portafolio de producto y servicios de la compañía con sus respectivos precios. Finalizado el proceso se agenda un encuentro entre el cliente y el analista funcional para el levantamiento de requerimientos.

Documentos: Solicitud de levantamiento de requerimientos.

b. Especificación:

Se desarrolla una reunión entre el cliente y el analista funcional de la compañía, en este encuentro se detalla la funcionalidad de la herramienta y las restricciones o validaciones identificadas por el cliente. Como resultado de esta actividad, se obtiene una propuesta comercial que contiene el tiempo de desarrollo, costos del producto a desarrollar, la forma de pago y los recursos que se utilizarán en el desarrollo del proyecto.

Documentos: Propuesta Comercial, Product Back Log (Lista de todas las tareas del proyecto con la priorización de las mismas).

c. Presentación del proyecto al equipo desarrollador.

Posterior a la aprobación de la propuesta comercial (por parte del cliente) se generan las historias de usuario, en éstas el analista detalla de forma funcional el producto, generando como salida un documento el cual se socializa con el equipo desarrollador, el cual realiza un cronograma priorizando las actividades a desarrollar (Sprint Back Log.).

Documentos: Sprint Back Log (Lista de tareas a realizar para la primera entrega del producto al cliente): Un proyecto puede tener uno o varios Sprint, los cuales no deben ser superiores a una semana, con el fin de ir realizando entregas periódicamente al usuario final.

d. Desarrollo

El proceso de desarrollo parte con la creación del cronograma de actividades por Sprint, en donde se especifican los responsables y tiempos de desarrollo, además se da inicio al desarrollo del primer Sprint. Las publicaciones del

software para pruebas deben ser contempladas en el cronograma, contarán con una periodicidad de por lo menos una semana con el objetivo de contar con una detección temprana de incidencias y fallas, esto mitigará los costos y tiempos extra por la detección tardía de funcionalidades erróneas o especificaciones no cumplidas.

e. Pruebas de concepto y calidad SCI

Las pruebas del software se deben realizar cada vez que se genere una nueva publicación por parte del equipo de desarrollo y se notifica al analista QA mediante un correo electrónico en donde se listan las funcionalidades que se liberan. El analista QA realiza pruebas a las funcionalidades liberadas detectando incidencias o fallas que serán reportadas mediante la plataforma Team Foundation Server (TFS), al desarrollador encargado. Esta es una plataforma interactiva que permite realizar seguimiento a las incidencias reportadas. El desarrollador informa las modificaciones ejecutadas al analista QA quien realiza la verificación y da por cerrada la incidencia. Todas las pruebas realizadas por el analista QA deben quedar documentadas en la plataforma.

Documentos: Set de Pruebas (soporte de las pruebas realizadas por cada publicación).

f. Pruebas de aceptación cliente

Posterior a la certificación del sprint por parte del analista QA, esta publicación es congelada y liberada para que el usuario final realice las pruebas correspondientes. Las incidencias que reporte el cliente se registran en TFS y serán gestionadas en el menor tiempo posible.

g. Puesta en producción

Desarrollados todos los sprint planeados para el proyecto, y realizadas las pruebas respectivas, el aplicativo es trasladado a un servidor de producción para su utilización. Una vez entregado el software el cliente cuenta con dos meses de garantía, dentro de los cuales el cliente puede reportar fallas, estas fallas deben ser clasificadas como nuevas funcionalidades o incidencias. Si son incidentes SCI Software los debe resolver en el menor tiempo posible, y cuando son nuevos requerimientos se debe generar un control de cambios para su posterior aprobación por el cliente.

2. Metodología

El proceso de identificación de problemas y diseño de soluciones inicia con la diagramación del flujo del proceso de la empresa SCI Software (figura 2) en el cual se describe el proceso de desarrollo de software y los actores que intervienen en el mismo.

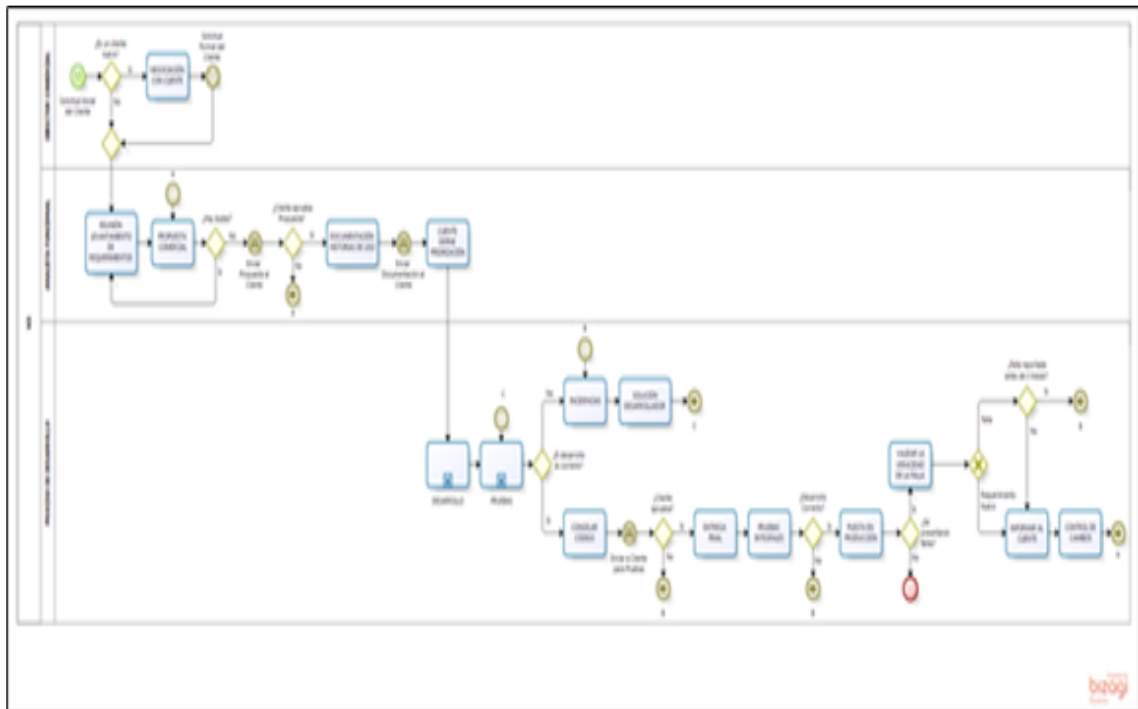


Figura 2. Proceso de desarrollo de software. Fuente: Autores.

En el análisis del proceso se identificaron debilidades en la comunicación entre los actores del proceso de diseño y desarrollo, desencadenando en el aumento de costos y tiempo de producción, y disminuyendo la satisfacción del cliente.

Dentro del levantamiento del proceso se realizó el análisis de la información disponible en la herramienta *Team Foundation Server (TFS)* en la cual se consolida toda la información y los controles de cambios realizados a los proyectos desarrollados y las propuestas comerciales enviadas. Dicha información se toma como base para enumerar los 5 focos que se catalogan como raíz del problema principal. A continuación se muestran los datos estadísticos de en los cuales se basa el análisis del proceso:

Tabla 1: Valor promedio hora de desarrollo.

Principales Clientes	Valor hora desarrollo
Banco GNB Sudameris	\$ 80.000,00
El tiempo Casa Editorial	\$ 65.000,00
Andicall	\$ 37.500,00
Américas BPS	\$ 35.000,00
Valor promedio de la hora de desarrollo	\$ 54.375,00

TOTAL DE PROYECTOS EJECUTADOS 2015 II		
Proyectos Con Controles De Cambios	60	69,77%
proyectos Sin Novedad	26	30,23%
Tiempo No Programado	444	días
Tiempo Promedio No Programado	7,4	días
Total Horas No Estimadas	3.552	horas
Diseñador	345	horas
Desarrollador	1.858	horas
Arquitecto de Base de Datos	1.350	horas
Costos No Planificados	\$ 193.140.000,00	

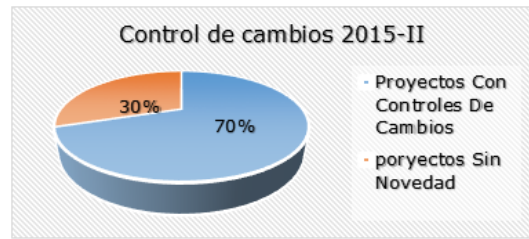


Figura 3. Informe TFS 2015 II. Fuente: SCI Software.

TOTAL DE PROYECTOS EJECUTADOS 2016 I		
Proyectos Con Controles De Cambios	74	73,27%
proyectos Sin Novedad	27	26,73%
Tiempo No Programado	547,6	días
Tiempo Promedio No Programado	7,4	días
Total Horas No Estimadas	4.381	horas
Diseñador	1.021	horas
Desarrollador	2.720	horas
Arquitecto de Base de Datos	640	horas
Costos No Planificados	\$ 238.206.000,00	

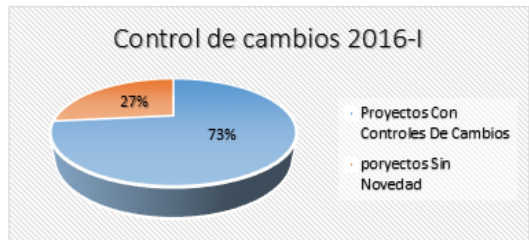


Figura 4. Informe TFS 2016 I. Fuente: SCI Software.

- ❖ El representante comercial omite información respecto a las especificaciones del cliente establecidas en el proceso de negociación, siendo relevantes para la consideración de recursos y repercutiendo en la estimación del tiempo de desarrollo y el costo del proyecto.

De acuerdo al estadístico sobre el control de cambios para el segundo semestre del año 2015 se ejecutaron modificaciones en un 69,27% de los proyectos adjudicados, y para el primer semestre del 2016 se ejecutaron modificaciones al 73,27%.

Se entiende como control de cambios los eventos que surgen del mismo proyecto que influyen en un aumento directo de recursos, los controles de cambio se ejecutan por falencias en el entendimiento de requerimientos u omisión de los mismos. Las modificaciones que los clientes solicitan y no se acuerdan en la negociación o levantamiento de requerimientos son tratados como control de cambio.

En los informes se estima que el 54% de los controles de cambios son ocasionados por la omisión de información en el proceso de negociación, esto ha significado 4.283 horas de trabajo no estimadas equivalentes a \$81.567.886, cifra que no se tomó en cuenta durante la negociación.

- ❖ En el proceso de levantamiento de requerimientos, el usuario no comunica todas las especificaciones de forma y funcionalidad con las cuales debe contar el desarrollo, generando una visión diferente por parte del equipo desarrollador y desencadenando en la insatisfacción del cliente y reprocesos en la cadena de producción.

Según el informe para el primer semestre del presente año, las dos principales razones de que el 54% de los controles de cambios provienen de la omisión de especificaciones iniciales son: (1) el representante comercial no transmite la totalidad de las características del desarrollo ofertadas, y (2) durante el levantamiento de requerimientos el tiempo estimado es de menos 2 horas, periodo en el cual no se incursiona en profundidad sobre los requerimientos del aplicativo, además que las especificaciones del cliente no otorga la información suficientes para tener una propuesta totalmente clara para ambas partes.

- ❖ La diferencia entre el lenguaje del desarrollador y el analista genera múltiples interpretaciones de los requerimientos del cliente, debido a que la estructura de la información no es técnica.

Para los dos últimos semestres evaluados se estima que el 46% de los controles de cambios se generaron por falencias en la comunicación interna, adjudicando 128 de los 134 totales a errores de comprensión entre el analista funcional y la parte técnica. Esto ha significado 3.649 horas de trabajo no estimadas equivalentes a \$69.483.755, cifra que no se tomó en cuenta durante la negociación.

- ❖ El tester no es específico en los reportes de incidencias durante las pruebas de concepto y calidad, ocasionando que el equipo de desarrollo consuma mayor cantidad de tiempo buscando el error, o que este no sea tomado en cuenta.

Según el reporte extraído de la plataforma *Team Foundation Server (TFS)* en lo corrido del año 2016 se ha generado 2.130 incidencias, de estas 1.123 se solucionaron después de transcurrir 15 días. Además, 1.813 pasaron por más de tres integrantes del equipo desarrollador antes de darse por solucionada.

- ❖ El cliente da por aprobada la propuesta de trabajo sin revisar lo que en esta se enuncia, trayendo como consecuencia el desarrollo de funcionalidades que no cumplen con lo que el cliente espera.

El reporte de nuevos desarrollos a medida muestra que el 10% de los proyectos cortos (50 en el último semestre) son nuevas especificaciones de aplicativos que se entregaron cumpliendo lo pactado en los acuerdos, sin embargo, no satisface las expectativas del cliente. Según el informe, para el primer semestre del 2016, la principal razón de esta situación es que los clientes no leen el acuerdo y no tienen claro que es lo que van a adquirir.

Identificada la problemática del proceso, se procedió a la generación de ideas por parte del grupo de trabajo, elaborando una lista de posibles soluciones a los problemas. Se seleccionaron las mejores alternativas y se integraron, bajo los dos principales conceptos de la teoría Lean Management: eliminar los procesos que no generan valor y aumentar la satisfacción del cliente; de acuerdo

a esto se desarrolla una propuesta general que da solución a los cinco focos identificados a lo largo del proceso.

Una vez desarrollada la solución general se especifica cómo integrar y adaptar cada componente de esta a la compañía, construyendo una solución integral y aplicable a las problemáticas identificadas.

2.1 Propuesta

El desarrollo de esta propuesta apunta a generar en las empresas desarrolladoras de software con características similares a la empresa SCI Software una cultura Lean en la que todos los roles que participan en el proceso de desarrollo de software, iniciando desde las directivas hasta el último eslabón del proceso sean parte activa en la generación del pensamiento Lean, a fin de generar una propuesta de mejoramiento que garantice la comunicación asertiva y el adecuado flujo de comunicación basados en una metodología de mejora de procesos con un enfoque hacia el cliente.

La propuesta de este trabajo inicia con el cambio de la estructura organizacional de la compañía. Las empresas desarrolladoras de software en Colombia se caracterizan por tener una estructura organizacional vertical, comprendida principalmente por las áreas de operaciones y comercial [2]. En el modelo descrito, las organizaciones restan importancia a las necesidades del cliente y al trabajo en equipo, teniendo como consecuencia pérdida de la comunicación entre departamentos o áreas y falta de organización en los procesos internos.

Uno de las fallas identificadas en el proceso actual, es el pago de comisiones a los representantes comerciales por el número de ventas y no por el cumplimiento de las entregas por proyecto. Parte de la propuesta organizacional, es la eliminación de la figura comercial del proceso de desarrollo, cambiando el método en el que se comisionan sobre los proyectos. En esta propuesta, las comisiones serán pagadas sobre el cumplimiento de fechas de entrega y requerimientos de los clientes, motivando a los equipos de trabajo a la consecución de las metas propuestas.

La creación de grupos multidisciplinarios que contribuyan en la búsqueda de soluciones a las necesidades y requerimientos de los clientes es primordial para mejorar la productividad de todo el proceso. Con este cambio se busca que los actores del proceso productivo trabajen mancomunadamente, permitiendo caracterizar la operación de las empresas desarrolladoras de software como un proceso flexible capaz de adaptarse a todo tipo de solicitudes por parte de sus clientes.

La toma de requerimientos al cliente generalmente es un foco de desinformación para todo el proceso de desarrollo, debido a que el cliente no tiene los medios ni el apoyo necesario para tener presente todas las características que necesita el sistema a desarrollar. Por tal motivo, es necesario diseñar una plataforma dinámica en la que el cliente encuentre todas las posibles

alternativas que la empresa ofrece para la producción de sus productos. Con esto se busca que el cliente se sienta parte del diseño y producción del desarrollo y a su vez, tenga una visión clara de la forma y funcionamiento del producto final.

Actualmente las devoluciones por parte del cliente son acusados a deficiencias en el diseño del desarrollo. La falta de claridad en las especificaciones enunciadas en los contratos ocasiona que se omita, o se dé por hecho los requerimientos de diseño del aplicativo o programa. Como respuesta a esta falla identificada, se propone la creación de pantallas que permitan visualizar los requerimientos de diseño y así mitigar las devoluciones por esta causa.

Otra fuente de devoluciones, en este caso en el proceso interno, se presentan en las pruebas funcionales, allí la información suministrada por el tester no permite identificar de forma clara la falla detectada en el desarrollo, siendo este una fuente importante de reprocesos. Para ello es necesario implementar un sistema estandarizado de control de incidencias, en el que el tester notifique de forma clara y precisa las inconsistencias encontradas en las pruebas con el fin de darle celeridad al proceso. En dicha estandarización se hace necesario un formato web con las siguientes características: Título del error, descripción del error, campo para seleccionar el desarrollador responsable, tiempo estimado de solución, un semáforo para que de forma gráfica el desarrollador pueda gestionar las incidencias que se van a vencer y una opción que permita el tester adjuntar un pantallazo del error.

Los cambios propuestos, apuntan a la mejora de la comunicación interna entre los diferentes participantes de la cadena, los requerimientos dinámicos (aplicativo y pantallas de diseño) realizados con el usuario serán distribuidas al equipo de trabajo, con el fin de que todos tengan un mismo objetivo, y sobre este se identifiquen y se propongan mejoras para futuros desarrollos.

2.2 Desarrollo de la propuesta

El plan estratégico se desarrolla bajo el principio Lean: “Eliminar todo aquello (estructuras, procesos, actividades, etc.) que no generen un valor agregado al cliente”, por esta razón, se busca maximizar su satisfacción por medio de la atención exclusiva de un equipo multidisciplinario de profesionales.

La estructura del equipo multidisciplinario tiene origen en identificar los procesos que aportan valor al cliente, o que son estrictamente necesarios para la ejecución de la actividad comercial de la organización. Posterior, se identifica la estructura organizacional a implantar en la empresa; estructura de tipo horizontal, desencadenando en canales de comunicación externos e internos cortos, además de desarrollar flexibilidad para el trabajo en equipo y superación de dificultades.

El Equipo Multidisciplinario va a estar integrado por Colaboradores que adopten los roles definidos por Kent Beck (creador de metodologías ágiles):

- ❖ **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema.
- ❖ **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- ❖ **Encargado de pruebas (Tester):** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- ❖ **Encargado de seguimiento (Tracker):** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- ❖ **Entrenador (Coach):** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas y se siga el proceso correctamente.
- ❖ **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- ❖ **Gestor (Big Boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. [11]

Estos equipos van a estar involucrados desde la *Concepción* de los Requerimientos hasta la *Administración* de los mismos, por ello el Proceso de Producción del Desarrollo de Software se reestructurará como se indica en la Figura 3.



Figura 3. Metodología de trabajo propuesta. Fuente: Autores

Los cargos Gerente Comercial IT y Analista Funcional desaparecen del proceso de desarrollo a causa del nuevo enfoque estratégico. Bajo el esquema que maneja la compañía, el gerente oferta los productos y servicios, desarrolla la negociación, y culmina su participación en el proyecto. De ahí que la metodología de trabajo actual no genera un compromiso directo con el cumplimiento del cronograma y expectativas del cliente por cada miembro de la compañía. Cada integrante ejecuta su trabajo sin dar importancia al avance en la producción del software y la satisfacción del Cliente.

El Analista Funcional actualmente es quien enlaza los diferentes procesos, éste se encarga de que el trabajo en ejecución se encuentre alineado con los requerimientos solicitados, ya que es el intermediario entre el Cliente y Líder del proyecto. Sin embargo, en el modelo propuesto este cargo se cataloga como uno de los que no generan valor al cliente, en cuanto a que alarga los canales de comunicación y absorbe responsabilidades de otras áreas.

En la nueva estructura se define un Líder de Proyectos, quién adopta el rol de Gestor (*Big Boss*). Como líder se va a desenvolver como representante de la empresa ante el cliente, sus compromisos inician desde la representación de la empresa en el levantamiento del requerimiento, hasta dar por finalizado y entregado el proyecto concluido el periodo de garantía. Los demás cargos de la estructura actual van a presentar un cambio en la forma en que desarrollan sus actividades y su alcance dentro del proyecto.

La estructura organizacional es de tipo horizontal, debido a que, se desarrolla en función al proceso principal de la compañía: el desarrollo software a medida.

Los roles de la nueva estructura organizacional se encaminan a tres conceptos fundamentales en la cultura Lean: El cliente es lo más importante, deben existir solo procesos que generen valor al cliente y debe ser un sistema flexible. A continuación se exponen los roles que se sugiere, deben integrar el equipo multidisciplinario, algunas funciones generales y la interacción que existe entre estos.

a. Gestor de proyectos

El gestor de proyecto debe asumir el rol de facilitador de los procesos, acompañar al equipo y al cliente desde el levantamiento de requerimientos, el desarrollo del proyecto y la entrega, eliminando impedimentos y realizar un proceso gradual de cambio y mejora continua.

b. Equipo de Desarrollo

El equipo de desarrollo debe estar conformado por personas que compartan las responsabilidad y el objetivo del proyecto que tienen a cargo. Deben tener la capacidad de trabajar de manera ágil y flexible y contar con las habilidades necesarias para poder ejecutar diferentes tareas (de desarrollo, soporte, pruebas o diseño) que proporcionen al cliente la confianza y tranquilidad necesaria para el cumplimiento de cada uno de los requerimientos.

La comunicación entre los miembros del equipo debe ser fluida, utilizando canales de comunicación eficientes, evitando así teléfonos rotos. Debe ser un equipo estable y sus miembros deben rotar lo menos posible por proyecto, todos deben compartir la visión del cliente y trabajar enfocados a lograr la satisfacción del mismo.

Por cada proyecto debe nombrarse un líder técnico diferente, que junto con el gestor del proyecto definen la arquitectura, lenguaje de programación, herramientas complementarias, recursos y entregables que se deben tener en cuenta en el software. Se recomienda que líder varíe para que todos adquieran experiencia de estructurar diferentes tipos de proyecto.

2.3 Herramientas

2.3.1 Plataforma Dinámica

La comprensión y toma de requerimientos se van a detallar en una Plataforma Dinámica, ésta va a proporcionar a las partes (Cliente y SCI Software) una visión clara del proyecto. La producción de esta herramienta tiene como base una Plantilla para el Levantamiento de Requerimientos y el Portafolio de la Compañía.

Ya que el objetivo es diseñar con el cliente, se va a desarrollar una plantilla para especificar descripciones detalladas de los aspectos del software que tienen alto impacto en la producción del aplicativo, pero que son tenidos en cuenta por los clientes y no son evidentes en la plataforma dinámica. Con esto, el líder técnico lo modela en la aplicación, de tal forma que los requerimientos sean claros. La plataforma no va a contar con funcionalidades altamente desarrolladas, simplemente será un apoyo visual para dar claridad a las especificaciones.

La utilización de esta plataforma dinámica va vinculada con una sensibilización en el tema de *Indagación y Obtención de Requerimientos*. Por medio de esta sensibilización se busca que los equipos multidisciplinarios estén en la capacidad de captar la información necesaria, detallada y exacta, ajustándose a las necesidades del cliente.

El detalle de la Plantilla se va a basar en el criterio que Karl Wieggers expresa; allí detalla el contenido con el que debe contar una ERS (Especificación de Requerimientos de Software) [10]. Para SCI, la plantilla debe contener:

- ❖ **Propósito del Software:** propósito y alcance del proyecto.
- ❖ **Descripción general:** perspectiva del producto, características del sistema, tipos y características del usuario, ambiente de operación, restricciones de diseño e implementación, documentación para el usuario, suposiciones y dependencias.
- ❖ **Requerimientos de la interfaz externa:** interfaces de usuario, interfaces del software e interfaces de las comunicaciones.
- ❖ **Otros requerimientos no funcionales:** requerimientos de desempeño, requerimientos de seguridad, requerimientos de estabilidad y atributos de calidad del software”. [10]

2.3.2 Feedback

La retroalimentación del equipo es una metodología de trabajo altamente productiva; siendo éste un equipo multidisciplinario, esta forma de trabajo va a permitir ver los problemas desde diversas perspectivas, buscando la solución que genere mayor valor para el cliente y facilite el trabajo. Además, la adecuada interacción entre los integrantes generar la cohesión del equipo y una visión clara de los objetivos planteados.

2.3.3 Reporte de incidencias

La detección tardía de problemas en las funcionalidades desencadena en mayor cantidad de trabajo y tiempo en desarrollo, actualmente la compañía hace uso de la plataforma *Team Foundation Server (TFS)* para el reporte de incidencias. Sin embargo, no detallar los problemas detectados finaliza en que el equipo técnico no dé importancia a lo que se reportó, o que se destine mayor cantidad de tiempo en la búsqueda del problema. Por tal motivo, se hace necesario que el equipo técnico que reporte dichas incidencias esté sensibilizado y capacitado en temas de técnicas de revisión, aseguramiento de la calidad del

software, estrategias de pruebas de software y en el manejo de la plataforma. De esta forma el equipo técnico va a tener la capacidad necesaria para dar una solución rápida a las incidencias y continuar con la liberación del aplicativo hacia el cliente.

2.3.4 Variables económicas por cumplimiento

En busca de cohesionar el equipo y mantenerlo alineado para dar cumplimiento al cronograma establecido y satisfacción al cliente, se plantea segmentar la retribución económica en una parte fija y otra variable que se encuentra ligada a la productividad, esta va a ser comisionada según el cumplimiento de los indicadores de control de gestión: entregas satisfactorias al cliente (fechas y negociación acordada), porcentajes de fallas en los meses pactados e incidencias en el lapso de pruebas, éste va a ser otorgado por el desempeño del equipo, puesto que siendo un conjunto multidisciplinario debe existir trabajo en equipo y apoyo mutuo para alcanzar las metas planteadas.

Una vez cerrado el proyecto se realiza la evaluación de cumplimiento, en ésta se analiza cómo trabajó el equipo y cómo afrontó cada meta para entregar resultados en los tiempos establecidos, además de cuantas incidencias fueron reportados, tanto interna como externamente. Este análisis y el cumplimiento de los indicadores frente a cada proyecto determinará el pago de comisiones a los diferentes integrantes del equipo, el porcentaje asignado va a depender del proyecto y su dificultad.

Indicadores:

$$\% \text{ de cumplimiento} = \frac{n^{\circ} \text{ de entregas satisfactorias}}{n^{\circ} \text{ entregas totales}} * 100$$

$$\% \text{ de fallas} = \frac{n^{\circ} \text{ de funcionalidades con fallas reportadas por el tester}}{n^{\circ} \text{ de funcionalidades totales}} * 100$$

2.4 Resultados esperados

Garantizar la comunicación asertiva a lo largo de las actividades que componen el proceso de producción y comercialización en las empresas de desarrollo de software, generando un ambiente laboral con sentido colaborativo y competitivo.

Eliminar los reprocesos implantando la plataforma dinámica que asegura captar los requerimientos necesarios en la construcción del software, garantizando un excelente producto y la satisfacción del cliente.

Buscar la interacción del cliente en el proceso de diseño y desarrollo del software, mediante el uso de la plataforma descrita con anterioridad, la cual constituye una herramienta efectiva para el equipo técnico.

3. Conclusiones

Implantar la estructura organizacional horizontal para generar que el cliente cuente con la asesoría y acompañamiento de profesionales altamente capacitados en el desarrollo de software a medida.

Los grupos multidisciplinarios van a permitir que se estimulen formas de trabajo en equipo mediante canales de comunicación ágiles y dinámicos, con el fin de lograr el cumplimiento de los objetivos planteados.

Las plantillas propuestas para la toma de requerimientos van a mitigar las omisiones de información por parte del cliente, dándole a éste la oportunidad de tener una percepción aproximada de cómo será el diseño del producto solicitado y así mismo generando ideas sobre su funcionamiento.

Las herramientas propuestas pueden ser desarrolladas por el mismo personal de la organización, no se requiere de una inversión alta para implantar.

4. Recomendaciones

Antes de su implantación se recomienda que todos los colaboradores involucrados cuenten con una capacitación previa acerca de la cultura Lean y de la importancia de su implementación en la compañía.

Una vez implementada la propuesta, se sugiere realizar campañas internas y externas, en las que se promueva el Lean como método de desarrollo organizacional.

5. Referencias Bibliográficas

- [1] Robbins, S. (2004). *Comportamiento Organizacional*. Pearson Educación.
- [2] Watzlawick, P.; Beavin, J.; Jackson, D. (2002) Teoría de la comunicación humana, Tiempo Contemporáneo, Buenos Aires.
- [3] Orue-Echevarría, J. (2004). Manual de valoración de puestos y calificación de méritos. Deusto.
- [4] Tuya, J. Ramos Roman, I & Dolado Colsin, J. (2007) Técnicas cuantitativas para la gestión en la Ingeniería de Software. Netbiblo.
- [5] Arias Chaves, M (2005) La ingeniería del requerimiento y su importancia en el desarrollo de Software. Revista Electrónica.
- [6] Bastos Tigre, P & Silvera F. (2009). Desafíos y oportunidades de la industria del software en América Latina. Cepal en coedición con Mayol Ediciones S.A.
- [7] Gallego & Cáceres. (2015). Identificación de factores que permitan potencializar el éxito de proyectos de desarrollo de software. Scienti et Technica.
- [8] Cerpa, N. (2011). Factores de éxito en proyectos de desarrollo de software: análisis de la industria chilena de software.
- [9] Machuca, M. (2008). Análisis del factor estratégico para alcanzar el éxito de un proyecto de gestión del conocimiento. Dirección y Organización.
- [10] Pressman, R. (2006). *Ingeniería del software, un enfoque práctico*. McGRAW-HILL.
- [11] Beck, K. (2000). *Una explicación de la programación extrema. Aceptar el cambio*.
- [12] <http://www.wentels.com/es/course/trabajo-en-equipo/diferencias-entre-grupo-de-trabajo-y-equipo-de-trabajo>
- [13] Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Educación.
- [14] McCall, Richards y Walters (1977). *Factors in Software Quality*.
- [15] Romero Bermúdez, E.; González Villareal, A. (2011) Elementos a considerar en un programa de comunicación efectiva, afectiva y asertiva: propuesta para dinamizadores del enfoque RBD. Umbral Científico.
- [16] Frenda Vega, M. (2012) Habilidad para una comunicación eficaz. Universidad de Cádiz.

[17] Galindo, J.; Karam, T.; Rizo, M. (2005) Cien libros hacia una comunicología posible. Universidad Autónoma de la Ciudad de México, México. Marc.