



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

**PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO  
9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA  
HÍBRIDA, APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA**

**Pablo César Paniagua González**

Asesorado por el Ing. Mario Roberto Reyes Marroquín

Guatemala, octubre de 2010

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA, APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR:

**PABLO CÉSAR PANIAGUA GONZÁLEZ**

ASESORADO POR EL ING. MARIO ROBERTO REYES MARROQUÍN

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2010

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. Luis Pedro Ortíz de León
VOCAL V	Agr. José Alfredo Ortíz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Juan Alvaro Díaz Ardavín
EXAMINADOR	Ing. César Rolando Batz Saquimux
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**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:

**PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA, APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2009.

  
Pablo César Paniagua González

Guatemala, 13 de enero de 2010

Ingeniero  
Carlos Azurdia  
Revisor de Trabajo de Graduación  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería

Respetable Ing. Azurdia:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **PABLO CÉSAR PANIAGUA GONZÁLEZ**, titulado: **“PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Sin otro particular, me suscribo de usted.

Atentamente,



Mario Roberto Reyes Marroquín  
Ingeniero en Ciencias y Sistemas  
Colegiado No. 8900  
Asesor de Trabajo de Graduación



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

Guatemala, 03 de Febrero de 2010

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 **PABLO CÉSAR PANIAGUA GONZÁLEZ**, titulado: **“PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA APLICABLE A LA INDUSTRIA DE SOFTWARE 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 Privados  
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería 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 “PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA, APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA”, presentado por el estudiante PABLO CÉSAR PANIAGUA GONZÁLEZ, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

*Ing. Marlon Antonio Pérez Türk*  
*Director, Escuela de Ingeniería Ciencias y Sistemas*



Guatemala, 04 de octubre 2010

Universidad de San Carlos  
de Guatemala



Facultad de Ingeniería  
Decanato

Ref. DTG.289.2010

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: **PROGRAMACIÓN EXTREMA, LA NORMA ISO 9000:2000 Y LA NORMA ISO 9001:2000 COMO BASES PARA LA FORMACIÓN DE UNA METODOLOGÍA HÍBRIDA, APLICABLE A LA INDUSTRIA DEL SOFTWARE DE GUATEMALA**, presentado por el estudiante universitario Pablo César Paniagua González, autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Murphy Olympo Paiz Recinos  
DECANO

Guatemala, octubre de 2010

/cc  
c.c. archivo.

## **AGRADECIMIENTOS A:**

Dios...

Mis padres - Lidia y Juan Pablo -, hermanas - Lilian y Mireya -, tía Estela, padrino Leonel, doña Elena, tíos, primos, amigos, asesor Mario, catedráticos...

Mi abuelita Tina, tía Marta, tío Carlos, abuelos D.E.P...

...

# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES.....</b>	<b>V</b>
<b>GLOSARIO.....</b>	<b>VII</b>
<b>RESUMEN.....</b>	<b>IX</b>
<b>OBJETIVOS .....</b>	<b>XI</b>
<b>INTRODUCCIÓN .....</b>	<b>XIII</b>
<b>1. MARCO TEÓRICO.....</b>	<b>1</b>
<b>1.1. Introducción .....</b>	<b>1</b>
1.1.1. Desarrollo ágil .....	1
1.1.2. Manifiesto ágil .....	2
<b>1.2. Metodologías ágiles.....</b>	<b>5</b>
1.2.1. Scrum.....	6
1.2.2. Adaptive Software Development (ASD) .....	6
1.2.3. Metodologías Crystal.....	6
1.2.4. Feature Driven Development (FDD).....	7
1.2.5. Lean Software Development (LSD).....	7
1.2.6. Dynamic Systems Development Method (DSDM).....	7

1.2.7. Programación extrema (XP) .....	8
<b>1.3. Estándar ISO .....</b>	<b>19</b>
1.3.1. ISO 9000:2000.....	21
1.3.2. ISO 9001:2000.....	25
<b>2. RELACIÓN ENTRE LA PROGRAMACIÓN EXTREMA Y EL ESTÁNDAR ISO.....</b>	<b>33</b>
<b>2.1. Características distintivas de la programación extrema y el estándar ISO.....</b>	<b>33</b>
2.1.1. Características de aplicación del software .....	34
2.1.2. Características de gestión .....	34
2.1.3. Características técnicas.....	35
2.1.4. Características de las personas.....	35
<b>2.2. Niveles de métodos del software – Escala de Cockburn.....</b>	<b>36</b>
2.2.1. XP .....	36
2.2.2. ISO.....	37
<b>2.3. Comparación entre la programación extrema y el estándar ISO ...</b>	<b>37</b>
<b>3. PROGRAMACIÓN EXTREMA Y EL ESTÁNDAR ISO EN GUATEMALA</b>	<b>43</b>
3.1. Análisis preliminar .....	43
3.2. Programación extrema en Guatemala .....	46
3.3. Estándar ISO en Guatemala .....	48

3.4. Otras metodologías .....	49
<b>4. METODOLOGÍA HÍBRIDA XP-ISO.....</b>	<b>53</b>
4.1. Análisis preliminar.....	53
4.2. Factores de decisión crítica.....	55
4.2.1. Cultura.....	55
4.2.2. Personal.....	55
4.2.3. Criticidad .....	55
4.2.4. Tamaño .....	56
4.2.5. Dinamismo .....	56
4.3. Representación gráfica de los factores de decisión crítica.....	56
4.4. Evaluación de la situación actual de la organización.....	59
4.4.1. Paso 1 .....	59
4.4.2. Paso 2 .....	59
4.4.3. Paso 3 .....	60
4.4.4. Paso 4.....	61
4.4.5. Paso 5 .....	62
4.5. Combinación de la programación extrema y el estándar ISO .....	64
4.5.1. Realización del producto.....	64
4.5.2. Medición y seguimiento.....	69
<b>CONCLUSIONES .....</b>	<b>71</b>

**RECOMENDACIONES .....73**

**BIBLIOGRAFÍA.....75**

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1. Proyecto típico de programación extrema .....	17
2. Modelo de un sistema de gestión de calidad basado en procesos Norma ISO 9001:2000.....	31
3. Utilización de las metodologías de desarrollo en Guatemala .....	45
4. Gráfico polar – Factores de decisión crítica .....	58
5. Aplicación de un entorno híbrido XP-ISO .....	63

## TABLA

I. Organizaciones y personas entrevistadas .....	43
--	----



## GLOSARIO

<b><i>Ad-hoc</i></b>	Hecho a la medida. Adecuado, idóneo.
<b>Caos</b>	Cultura de desorden y desorganización.
<b>Explícito</b>	Dar un detalle exacto de alguna metodología, fundamentándolo en la teoría. Se cumple a cabalidad.
<b>Implícito</b>	No dar un detalle exacto de alguna metodología. No se cumple a cabalidad.
<b>Orden</b>	Cultura basada en la disciplina y procesos estrictos.
<b>Pseudo</b>	A medias. Supuesto.
<b>Refactorizar</b>	Reestructuración del código, sin cambiar su comportamiento. Busca que el código sea entendible.
<b><i>Sprints</i></b>	Fases o iteraciones del proyecto.
<b><i>Stakeholders</i></b>	Sinónimos: involucrados, interesados. Persona que tiene intereses directos sobre la funcionalidad adecuada del proyecto. Puede afectar o ser afectada por una decisión.

**Tester**

Encargado de pruebas. Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas periódicamente, da los resultados al equipo. Es responsable de las herramientas de soporte de las pruebas.

## RESUMEN

Al realizar un estudio de las organizaciones que desarrollan software en Guatemala, se encuentra que el proceso de desarrollo del software en algunas organizaciones, principalmente las pequeñas y medianas empresas, no utilizan una metodología de desarrollo específica para crear un software adecuado para el cliente, por lo que varias veces no se cumple con las expectativas del usuario y se reduda en el desarrollo de requerimientos sin saber exactamente que hacer.

En la metodología híbrida XP-ISO, estudiada en este trabajo de graduación, se indican los pasos clave para que una organización pueda identificar las características de su situación actual. Y a partir del punto obtenido, pueda determinar un plan para implementar dicha metodología.

Se presenta un análisis de la unificación de la XP con la ISO, demostrando de esta forma, que es posible garantizar por medio de un estándar internacional el desarrollo del software, con prácticas ágiles e idóneas para la creación del mismo - software -.



# OBJETIVOS

## **General:**

Realizar la combinación de la Programación Extrema junto a la norma ISO 9000:2000 y la norma ISO 9001:2000. Con dicha combinación proponer las bases de una metodología para contribuir y beneficiar a la industria del software de Guatemala.

## **Específicos:**

1. Hacer un análisis comparativo, determinando las ventajas y/o coincidencias de la Programación Extrema, la norma ISO 9000:2000 y la norma ISO 9001:2000, tratando de eliminar las desventajas de dichas metodologías en la mayor forma posible.
2. Estudiar las metodologías de software utilizadas actualmente en Guatemala, especialmente en las empresas o instituciones que hagan uso de la Programación Extrema y/o el estándar ISO 9000:2000 y 9001:2000.
3. Definir la metodología híbrida XP-ISO, principalmente las ideas clave que ayuden a su formación e implementación, por medio de un análisis comparativo.



# INTRODUCCIÓN

Al desarrollar software se necesita llevar cierta disciplina para que cumpla con el cometido deseado. Existen diversas metodologías que las empresas pueden utilizar para llevar a cabo el ciclo de desarrollo del software. Entre estas metodologías, se encuentra la metodología híbrida XP-ISO.

La metodología híbrida XP-ISO tiene como base la programación extrema - XP - y las normas ISO 9000:2000, 9001:2000 - ISO -. Dicha metodología propone un desarrollo flexible y dar un especial énfasis a los clientes, sin olvidar con ello el reconocimiento de estándares. Se busca tener una buena disciplina en el desarrollo del software y como consecuencia de lo anterior obtener un software de alta calidad.

Dado que en Guatemala existen pequeñas y medianas empresas, dedicadas al desarrollo de software, en las que no se utiliza explícitamente alguna metodología de software, la metodología propuesta, XP-ISO, resulta ser una opción importante que se podría considerar para implementar en un futuro.

En este trabajo de graduación se presenta una definición detallada de la metodología híbrida XP-ISO que es aplicable a la industria del software de Guatemala.



# 1. MARCO TEÓRICO

En las secciones 1.1 a la 1.3 se utilizaron las referencias [1], [2], [3].

## 1.1. Introducción

### 1.1.1. Desarrollo ágil

El desarrollo ágil no es una metodología de software, ni tiene predefinidos una serie de pasos para llevarse a cabo. Es más bien una filosofía que incluye formas de pensar y trabajar ágilmente.

Es un proceso de desarrollo incremental que minimiza los riesgos por medio de iteraciones cortas. Se disminuyen considerablemente los trámites burocráticos, creando de esta forma, una rapidez en la realización y entrega del software.

Cada iteración incluye planificación, requerimientos, análisis, diseño, codificación, pruebas y documentación; sin embargo, dicha iteración se aplica de una manera un poco más rápida que en los métodos tradicionales. Entregando el software completamente funcional y sin errores. Al final de la iteración se hace una reevaluación y se asignan prioridades para el desarrollo siguiente.

Existe una buena comunicación entre todas las partes involucradas, esto incluye a los analistas, codificadores, jefes de proyecto, *testers*, clientes. Se solucionan las dudas encontradas en el camino, con la retroalimentación constante y se eliminan los errores rápidamente. Logrando así, por una parte un buen grado de sinergia entre el equipo de desarrollo y por el otro una alta efectividad en el software.

### **1.1.2. Manifiesto ágil**

El término ágil aplicado al desarrollo del software surge en una convención, realizada del 11 al 13 de febrero del 2001, en donde se reunieron 17 expertos de la industria del software, en Utah EEUU.

El objetivo primordial de la reunión, era para tratar de las necesidades con que se encontraban las empresas en repetidas ocasiones, en las que había que desarrollar una determinada aplicación, relativamente pequeña, más rápido que de costumbre. Es decir, sin cumplir toda la ceremonia que se tendría que realizar con una metodología tradicional. Muchas veces, esto creaba un caos en las instituciones ya que no existían parámetros en los que se pudieran guiar para lograr una entrega rápida del software. Es así, como ven la necesidad de reunirse estos expertos del software, para acordar reglas claras a la hora de desarrollar software rápidamente. De esta forma, surgen los valores y principios para desarrollar software rápidamente con requerimientos cambiantes.

Después de realizada dicha reunión, se crea la conocida Alianza Ágil. Una organización sin ánimo de lucro, enfocada en dar ayuda a organizaciones en la implementación del desarrollo ágil.

Como punto de partida, la Alianza Ágil crea un documento, en el que se incorporan todos los principios y valores del desarrollo ágil. El documento, conocido por tener la filosofía ágil, es el llamado Manifiesto Ágil.

#### 1.1.2.1. Valores del manifiesto ágil

Los individuos e interacciones del equipo de desarrollo están sobre los procesos y herramientas: Las personas son la parte primordial para tener éxito en un proyecto de software. Así, se concluye que es mejor formar bien el equipo de personas involucradas en el proyecto y luego que estos creen el entorno de desarrollo conforme a sus necesidades, y no a la inversa como suele suceder equívocamente.

Desarrollo de software funcional es mejor que una buena documentación: Se deben generar documentos solo cuando sean estrictamente necesarios, de lo contrario, estos no se deben crear. Dichos documentos deben ser concisos.

Es mejor la colaboración del cliente que la negociación de contratos: Es muy importante que exista mucha interacción entre el cliente y el equipo de desarrollo, para garantizar un desarrollo adecuado del proyecto y el éxito del mismo.

Lo primordial es la respuesta al cambio, más no el seguimiento de un plan: Tiene que existir la habilidad para responder a los cambios que ocurran durante el desarrollo del proyecto. Puede ser que cambien, los requerimientos, la tecnología, el equipo, entre otras. Y la planificación debe ser flexible y abierta a cambios.

### 1.1.2.2. Principios del Manifiesto Ágil

Los valores descritos anteriormente fueron la base para proponer los principios del Manifiesto Ágil. Marcan la diferencia entre un proceso ágil y uno tradicional. Son doce principios. Los primeros dos, tienen un enfoque general mientras que los siguientes están más enfocados en el proceso de desarrollo.

1. Lo primordial es la satisfacción del cliente a través de entregas tempranas y continuas del software, para darle más valor.
2. Los requerimientos cambiantes son bienvenidos, incluso que sea a finales del desarrollo. Los procesos ágiles utilizan los cambios para dar al cliente una ventaja competitiva.
3. Entregar con frecuencia software funcional desde un par de semanas hasta un par de meses, tratando de tener un intervalo corto de tiempo entre cada entrega.
4. Personas del negocio y desarrolladoras deben trabajar juntos a lo largo del proyecto.
5. Desarrollar proyectos con individuos motivados. Crear el ambiente y el apoyo necesario y confiar en el equipo para alcanzar un buen trabajo.
6. El método más eficiente y eficaz para transmitir la información hacia el equipo de desarrollo y dentro del mismo es la conversación cara a cara.
7. Software funcional es el mejor indicador del avance del proyecto.

8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deben estar en una armonía permanente.
9. Soporte continuo a la calidad técnica y al buen diseño, mejora la agilidad.
10. La simplicidad es esencial.
11. De los equipos auto-organizados surgen las mejores arquitecturas, requerimientos y diseños.
12. A ciertos intervalos de tiempo, el equipo debe analizar respecto a cómo llegar a ser más efectivo, y de acuerdo a esto reubica su comportamiento.

## **1.2. Metodologías ágiles**

Existen varias metodologías que están enmarcadas como ágiles, todas ellas tienen sus bases en el desarrollo y Manifiesto Ágil. Entre estas, existen algunas características similares y hay otras en las que sus diferencias son notables.

A continuación se describen de una manera muy breve las metodologías ágiles más conocidas. La metodología programación extrema se describe detalladamente.

### **1.2.1. Scrum**

Sus creadores son Ken Schwaber, Jeff Sutherland y Mike Beedle. Es un marco para la gestión de proyectos, utilizado en proyectos de requerimientos cambiantes. Entre sus características principales están el desarrollo dividido en *sprints*, cada una de estas iteraciones con una duración de 30 días. También existen reuniones diarias de 15 minutos entre el equipo de desarrollo para la coordinación e integración del proyecto.

### **1.2.2. Adaptive Software Development (ASD)**

Su creador es Jim Highsmith. Es un proceso iterativo, acepta fácilmente los cambios y está orientado a los componentes de software más que a las tareas. Su ciclo de vida está compuesto por:

Especulación: Inicio del proyecto y planificación de las características del software.

Colaboración: Desarrollo de las características.

Aprendizaje: Revisión de la calidad y entrega al cliente. A partir de la revisión de los componentes se aprende de los errores y se vuelve a iniciar el ciclo de desarrollo.

### **1.2.3. Metodologías Crystal**

Desarrolladas por Alistair Cockburn. Este es un conjunto de metodologías que tiene a las personas como un factor clave y reduce en gran medida los artefactos producidos.

Existen políticas entre el equipo de desarrollo, por lo general utilizan una nomenclatura de colores de acuerdo al número de integrantes del equipo de desarrollo.

#### **1.2.4. Feature Driven Development (FDD)**

Sus desarrolladores son Jeff De Luca y Peter Coad. Tiene iteraciones cortas como máximo de dos semanas. Está enfocado principalmente en las fases de diseño e implementación del sistema.

#### **1.2.5. Lean Software Development (LSD)**

Uno de sus precursores fue Bob Charette's. Aquí se consideran los riesgos del proyecto, manejándolos oportunamente se convierten en mejoras al cliente. Es utilizada en numerosos proyectos de telecomunicaciones de Europa.

#### **1.2.6. Dynamic Systems Development Method (DSDM)**

Sus inicios se remontan a 1994, es un marco de desarrollo donde los procesos son iterativos e incrementales y donde el cliente y el equipo de desarrollo trabajan juntos. Las fases que contempla DSDM son estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción y la implementación.

En la subsección 1.2.7, es la siguiente, y las que la conforman se utilizaron las referencias [4], [5], [6], [7], [8].

### **1.2.7. Programación extrema (XP)**

La programación extrema también conocida por su nombre en inglés como eXtreme Programming (XP), es la más común de las metodologías ágiles. Iniciada por Kent Beck en 1990, y aunándose a él tiempo después Ward Caunnigham. Ambos logran concretar pruebas con esta metodología. La primera vez que se utilizó XP fue en el proyecto Chrysler Comprehensive Compesation por el equipo de Kent Beck, quienes utilizaron el lenguaje Smalltalk para poner en marcha dicho proyecto.

“Dejar que el desarrollo de software sea divertido, simple, flexible, predecible, con pocos riesgos, eficiente y más científico”. (Kent Beck, 2004)

Ron Jeffries da una mejora a la definición propuesta por Kent Beck, la cual dice:

“La programación extrema es una disciplina del desarrollo de software basado en los valores de simplicidad, comunicación, retroalimentación y coraje. Trabaja con todo el equipo involucrado en el proyecto junto con la presencia de prácticas simples y suficiente retroalimentación para que el equipo pueda ver en donde está y ajustar las practicas de acuerdo a su propia situación”.

La programación extrema es una disciplina de desarrollo de software enfocada en satisfacer las necesidades del cliente. Enfocándose principalmente en las necesidades que el cliente tenga.

Tiene la capacidad de cambiar en cualquier momento los requerimientos que él tenga, trabajando en equipo - administradores, desarrolladores y cliente - para lograr crear un software de alta calidad.

Una de las características importantes de esta metodología es el fundamento de las pruebas del desarrollo, esto significa que cada programador escribe sus pruebas junto al código de producción, agregándolas en el proceso de integración continua y construcción.

La programación extrema es un proceso evolutivo que utiliza iteraciones cortas para refactorizar un sistema con poca funcionalidad, esto permite a los desarrolladores entender de una mejor forma el proyecto, centrándose directamente en la iteración actual sin realizar nada con anticipación. Se obtiene así, una metodología disciplinada y adaptable.

#### 1.2.7.1. Valores de la programación extrema

- Simplicidad:

“Los programadores extremos hacen una cosa de la manera más simple de tal forma que esta pueda funcionar”. (Aderson, 2004)

Los programadores codifican lo que es necesario para el trabajo actual, no es necesario tener que trabajar para cosas futuras que quizá ni se lleguen a utilizar. Es mejor realizar algo con simplicidad y si es necesario después cambiarlo con un poco de trabajo extra. A diferencia de realizar algo muy complejo y nunca utilizarlo.

“No debe usarse la simplicidad como excusa para realizar un diseño pobre”. (Shore, 2008)

Dado que se involucra al cliente, es mejor tener conceptos simples, porque se entienden fácilmente, a utilizar algo muy complicado y que nadie lo entienda. Logrando así compartir a todos el conocimiento.

- Comunicación:

La programación extrema tiene un importante énfasis en la comunicación entre todos los involucrados del proyecto, desde clientes hasta jefes de proyecto. La programación extrema hace que la comunicación dentro del equipo de proyecto se mas fácil de fluir comparado con otras metodologías evitando retardos y gastos innecesarios.

Es inviable pensar que cualquier proyecto pueda desarrollarse sin comunicación. Es común encontrar que a la hora de tomar los requerimientos, estos no se captan de la mejor forma, generando confusiones entre todo el equipo.

La programación extrema da un fuerte impulso a la comunicación por medio de prácticas aplicables en corto tiempo, entre ellas están las pruebas unitarias, programación en parejas y la estimación de tareas, todas las actividades involucran la comunicación con el cliente.

- Retroalimentación:

Es la interacción entre los clientes y desarrolladores. Trabajan de la mano. Retroalimentación efectiva es necesaria en un equipo; esta constituye la confianza y ayuda a eliminar las confusiones creando conciencia de las cuestiones relacionadas con el proyecto. La retroalimentación de las pruebas ayuda a corregir este error.

La retroalimentación se da en cuestión de minutos y días. Los programadores prueban toda la lógica del sistema que podría fallar. La retroalimentación sirve en la identificación constante de las necesidades del cliente y ayuda a corroborar junto con el cliente si este sabe lo que quiere, si no se puede crear una interacción amena de tal forma que se hagan las correcciones necesarias para reflejar con exactitud los requerimientos del cliente.

- Coraje:

“Tener una acción efectiva ante los problemas”. (Kent Beck, 2004)

Los desarrolladores necesitan coraje para enfrentar situaciones de la vida real.

La simplicidad, comunicación y retroalimentación dan coraje. La comunicación da coraje porque se expone lo que se piensa y como se cree que se puede solucionar un problema. Retroalimenta porque el programador expondrá lo que no le parece y dirá lo que a su criterio es lo mejor. Simplifica porque cuando hay posibilidad de reducir algo lo intenta.

Hay que tener coraje para simplificar las partes del proyecto y respetar los tiempos de desarrollo establecidos. También cuando sea necesario, detener el avance del proyecto y organizar una reunión para la solución a algún problema.

- Respeto:

Kent Beck en el 2004 reconoció el respeto como otro de los valores que puede influir en el desarrollo del software.

Este enfatiza la necesidad del respeto entre todos los miembros del equipo. Esto es necesario para tener respeto por todas las ideas de cada miembro del equipo, las contribuciones de cada miembro serán valoradas y respetadas para dar coraje entre ellos.

- Otros valores:

Kent Beck agrega que son necesarios más valores como, seguridad, predictibilidad, seguridad, entre otros. También aconseja a las organizaciones pensar en valores positivos que podrían influenciar en el equipo para un desarrollo de software efectivo.

#### 1.2.7.2. Principios de la programación extrema

Los principios de la programación extrema están basados en sus valores.

Retroalimentación rápida: Trabajar de la mano con el cliente ayuda a entender mejor los requerimientos, tratando de cumplir todo lo que se desea, como por ejemplo la creación de una interfaz *ad-hoc*, entre otras, proporcionando la oportunidad de una retroalimentación rápida. Se debe retroalimentar en pequeños ciclos de tiempo para establecer si verdaderamente se está cumpliendo con los requerimientos del cliente.

Asumir simplicidad: Se debe asumir que cada problema en cuestión, se puede resolver de forma simple. Asumir la simplicidad implica tener el diseño solo para la iteración actual, sin considerar lo que pueda pasar para el futuro.

Realizar cambios incrementales: Los problemas se solucionan con una serie de cambios, tomando en cuenta la planificación, diseño, desarrollo y pruebas, obteniendo así, una solución mejorada.

Adopción del cambio: Los desarrolladores de la programación extrema deben estar dispuestos siempre a esperar cambios y estar listos para adoptarlos. XP sugiere adoptar una estrategia que permita establecer alternativas y preservarlas mientras se resuelven los problemas que están teniendo los usuarios.

Hacer trabajo de calidad: La calidad tanto en la codificación como en el sistema en general, tiene como consecuencia una gran satisfacción en el cliente. Es por ello, que la programación extrema tiene un énfasis importante en las pruebas antes del desarrollo.

Otros principios: Kent Beck identificó otros principios que podrían ayudar para el desarrollo del software, utilizados en situaciones particulares, estos son: Humanidad, economía, beneficio mutuo, una misma semejanza, mejora continua, diversidad, reflexión, flujo, oportunidad, redundancia, fracaso, calidad, aceptar la responsabilidad.

### 1.2.7.3. Prácticas de la programación extrema

“Las prácticas son algo que el equipo de programación extrema debe realizar todos los días durante el desarrollo, esta debe ser su rutina o sus actividades diarias”. (Kent Beck, 2004)

Se deben utilizar las prácticas tomando en cuenta los valores, por ejemplo, al programar en parejas tiene que practicarse la buena comunicación, la retroalimentación y la simplicidad en el sistema.

“Las prácticas son un vector que indica dónde estás y hacia a dónde vas con XP”. (Kent Beck, 2004)

Existen doce prácticas dadas a conocer por High Richard en el 2004, estas se dan a continuación.

El juego de la planeación: Determina el ámbito de la iteración actual, y las prioridades del cliente para poder implementarlas. También establece la duración de cada tarea, la organización de la cultura de trabajo y la planificación de cada iteración.

Versiones pequeñas: Esto implica poner en marcha el sistema en poco tiempo, por lo regular entre un día y un mes; sin embargo, hay situaciones donde se puede requerir mas del tiempo indicado de acuerdo a la complejidad del proyecto, tomando en cuenta no excederse demasiado, como máximo podría ser un referente de 3 a 4 meses.

Diseño simple: El diseño simple ayuda con la entrega rápida, de una versión funcional del producto al cliente, para que este último pueda ver si es lo que realmente deseaba, a diferencia de si se realiza algo muy complejo que sería muy difícil de entregarlo a tiempo y quizás no sea lo que el cliente quería y luego lo cambie.

Pruebas: Las pruebas son imprescindibles en la programación extrema, ya que con ellas se garantiza el éxito del proyecto. A nivel de codificación por lo regular se crean las pruebas antes que el código a ser probado. Se hace la prueba asumiendo que existe lo que se desea obtener y luego se codifica para alcanzarlo.

Esto es de vital importancia para cuando se integra todo el proyecto, ayuda a optimizar el tiempo en la unificación de todo el trabajo.

Integración continua: Esto es algo muy importante ya que para no tener muchos problemas a parte de realizar las pruebas, se debe integrar el proyecto en lapsos cortos de tiempo, normalmente se hace a diario, ayudando así a eliminar rápidamente los problemas encontrados. No se tiene que invertir mucho tiempo a la hora de integrar ya que si se encuentran muchos fallos en el código cuando se integra y no se logra resolver rápidamente se debe descartar.

Refactorización: “Si el código apesta cámbialo” (Kent Beck, 2004). Si el código es difícil de entender o modificar, significa que se debe quitar ese código y reemplazarlo por otro mejor. Es mejor reescribir el código, a esto es a lo que se le llama refactorizar. La refactorización continua ayuda a crear cambios más rápidamente.

Metáfora: Los proyectos en programación extrema deben tener una metáfora y ser guiados por esta. Esta define como conceptualiza el equipo al sistema, escribiéndolo en un lenguaje relevante para el negocio.

Programación en parejas: El código se desarrolla en parejas, en la misma computadora. Esto ayuda ya que como es sabido dos cabezas piensan mejor que una, entonces se puede cooperar mutuamente para alcanzar la mejor solución al problema. Por lo regular se coloca a un programador con mucha experiencia y a otro con menos experiencia.

Propiedad colectiva del código: Es importante tener una estandarización dentro del código para que cualquier integrante del equipo que considere oportuno modificar parte del código, pueda hacerlo sin ningún problema.

40 horas semanales: Esto se centra principalmente en el tiempo que las personas trabajan en el proyecto, indicando que debe ser un tiempo adecuado siendo este alrededor de 40 horas semanales, ya que sería inconsistente que una persona trabajara 60 horas por semana, esto evidentemente reduciría el desempeño del equipo, y en tal situación difícilmente se alcanzaría un proyecto con buena calidad. Una buena planificación del tiempo ayuda mucho.

Cliente en el sitio: El cliente debe estar junto al equipo de desarrollo, para responder cualquier duda que los desarrolladores tengan, estableciendo las prioridades de acuerdo a sus requerimientos.

Estándares de codificación: Es vital para poder tener colectividad del código. Se debe buscar que cualquiera pueda entender el código escrito por cualquier miembro del proyecto. Ayudando a reducir el tiempo a la hora de tener que realizar modificaciones.

#### 1.2.7.4. Herramientas

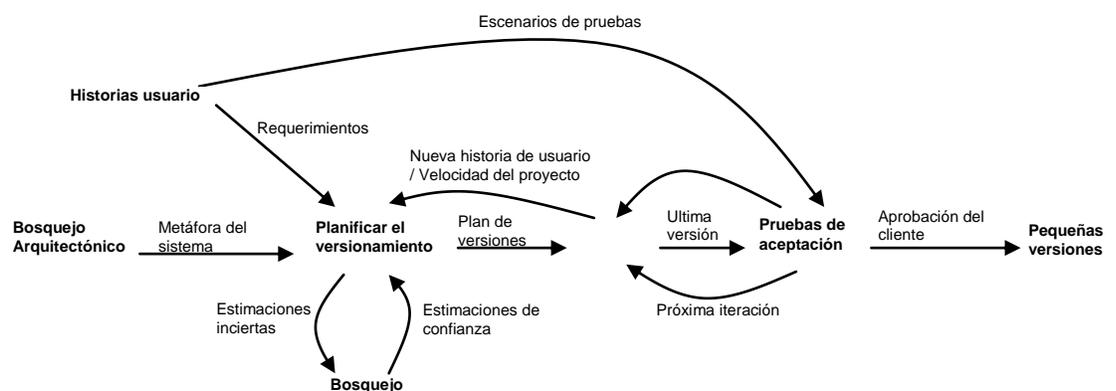
A continuación se describen algunas de las herramientas más utilizadas en la programación extrema.

Historias de usuario: “Las historias de usuario no son los detalles de implementación, tampoco son las especificaciones completas de los requerimientos. ...

Las historias no son requerimientos, tampoco casos de uso, ni narraciones, son más simples que eso. Son planes. Descripciones que indican el trabajo que el equipo debe producir” (Shore, 2008). Esta herramienta es de las más utilizadas en donde los clientes plasman de una manera simple las necesidades del sistema.

Tarjetas CRC (Clases, Responsabilidades y Colaboradores): CRC proporciona una forma simple de identificar y organizar las clases que son relevantes al sistema o los requerimientos del producto. Son una herramienta para el análisis de requerimientos, se explora el sistema determinando las clases necesarias y sus relaciones entre sí. Se utiliza CRC para el registro de los nombres de las clases, responsabilidades y colaboradores. Usualmente están integradas con herramientas de software que soportan el UML.

**Figura 1. Proyecto típico de programación extrema**



En esta imagen se utilizó la referencia [5].

### 1.2.7.5. Ventajas y desventajas de la programación extrema

#### ➤ Ventajas:

- Desarrollo basado en iteraciones.
- Respuesta rápida a requerimientos cambiantes.
- Entrega rápida de un prototipo.
- Permite la flexibilidad.
- Mayor fiabilidad del sistema.
- Mejora la calidad del software por medio de refactorización.
- Hay una taza mayor de código.
- Calidad de código, debido a que es en pareja se reducen las fallas.
- Propiedad colectiva del código.
- Acceso a usuarios involucrados en el proyecto.
- Poca administración o gestión.
- Adecuado para un equipo de mediano tamaño.
- Ambiente de desarrollo cohesivo.

#### ➤ Desventajas:

- Se enfatiza en la entrega de resultados parciales.
- No es escalable.
- Las pruebas continuas retardan el desarrollo.
- La programación en pareja es costosa.

- No hay predictibilidad.
- Requerimientos no definidos.
- Enfoque no estructurado para el desarrollo.
- Difícil de predecir costo y tiempo de implementación. Por lo regular es un costo elevado.
- No hay planificación.
- No hay documentación.
- Por lo regular la practican desarrolladores con experiencia.
- No es adecuado para equipos grandes.

En la sección 1.4 se utilizaron las referencias [6], [9], [10].

### **1.3. Estándar ISO**

La ISO (International Standardization Organization), también conocida por su traducción al español como Organización Internacional para la Estandarización. Es una organización que tiene su sede en Ginebra Suiza, fue fundada en 1946 para el desarrollo de términos comunes estandarizados de fabricación, comercio y comunicaciones.

En 1987, fueron publicadas las series ISO 9000, éstas incluyen estándares genéricos que brindan una guía para la gestión de calidad y los requerimientos de garantía de la misma. Son aplicables a cualquier compañía independiente de su tamaño y oficio, centrándose principalmente en empresas de fabricación y sectores de servicio.

Las series ISO 9000, son una familia de estándares de gestión y aseguramiento de calidad que proveen un marco de trabajo a la organización para la entrega continua de productos o servicios de acuerdo a los requerimientos del cliente.

La ISO 9000, está conformada por los 4 documentos siguientes:

ISO 9000:2000. Gestión de calidad y vocabulario. Es un reemplazo a la ISO 8402 y ISO 9000-1.

ISO 9001:2000. Requerimientos del sistema de gestión de calidad. Es un reemplazo a la ISO 9001:1994, ISO 9002 y ISO 9003.

ISO 9004:2000. Directrices de funcionalidad para los sistemas de gestión de calidad. Es un reemplazo a la ISO 9004-1.

ISO 19011. Directrices de auditoría para el entorno de los sistemas de gestión de calidad. Es un reemplazo a las partes 1 a 3 de la ISO 10011, 14010, 14011 y 14012.

Esta investigación hace uso principalmente de la norma ISO 9000:2000 y 9001:2000.

En la subsección 1.4.1 y de las que se compone se utilizaron las referencias [10], [11], [12], [13], [14].

### **1.3.1. ISO 9000:2000**

La norma ISO 9000 describe los fundamentos de los sistemas de gestión de calidad que contiene el vocabulario y terminología más utilizada en las normas de esta serie.

Sirve para conocer las bases filosóficas del estándar ISO y la terminología que se utiliza al aplicar dicho estándar. Hay ocasiones que podrían surgir dudas respecto a alguna interpretación que se está dando a algún concepto, posiblemente equivocada. Es en el documento de la ISO 9000:2000 en donde se puede consultar este tipo de dudas, logrando utilizar las normas de la mejor forma.

Está formada por dos secciones, en la primera se explican los doce fundamentos de los sistemas de gestión de calidad.

En la segunda, se describen los términos y definiciones de los sistemas de calidad, divididos en diez partes incluyendo los términos de calidad, gestión, organización, procesos y productos, características, entre otras.

Está orientada en procesos más que en productos. Es utilizada por las organizaciones para regular sistemas de calidad internos y garantizar así los sistemas de calidad de sus proveedores.

### 1.3.1.1. Principios de gestión de calidad 9000:2000

Los principios sirven como un apoyo a las normas ISO, con estos se determinan las bases para lograr una mejora continua en los procesos de la organización y obtener así una alta calidad en los productos o servicios de la empresa.

**Organización enfocada al cliente:** Los clientes son la razón de ser de una organización, es por ello la gran necesidad de satisfacer a cabalidad los requerimientos que desee.

**Liderazgo:** Los líderes son los responsables de dirigir hacia el camino correcto la creación de productos con calidad. Esto también implica establecer un ambiente de trabajo adecuado para obtener el máximo potencial de cada empleado. El liderazgo debe ser compartido, estableciendo una cadena de liderazgo en todos los niveles de la organización.

**Participación del personal:** Esto indica que todos los integrantes de la organización deben tener una actitud proactiva, contribuyendo de manera directa en procesos en los que se necesite a determinada persona.

**Enfoque basado en procesos:** Las actividades y recursos deben ser manejados como un proceso para lograr mayor efectividad en la empresa.

En el apartado 3.4.1 de la norma ISO 9000:2000 se define un proceso como:

“Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Nota1: Los elementos de entrada para un proceso son generalmente resultados de otros procesos.

Nota2: Los procesos de una organización (3.3.1) son generalmente planificados y puestos en práctica bajo condiciones controladas para aportar valor.”

Enfoque de sistema para la gestión: Aquí se visualiza a la organización como un conjunto de procesos relacionados entre sí, constituyendo un sistema formado por actividades, personal y recursos que deben administrarse como un proceso único.

Mejora continua: En un mundo cada vez más cambiante, es imperativo mejorar a través del tiempo, por medio de comparaciones hechas entre la misma empresa, haciendo un análisis directo sobre el desempeño de la misma, logrando mejorar significativamente los procesos actuales en los cuales no se logra la máxima eficiencia. Se puede estudiar a la competencia para tratar de mejorar al rival o rivales principales, ya que esto es un indicador de que tan bueno es el desempeño que tiene la organización.

Enfoque basado en hechos para la toma de decisión: Algo muy importante para tomar una decisión es la información que se tenga, para poder determinar un análisis eficaz. Es así como la organización, debería evaluar su desempeño basándose en planes, objetivos, resultados obtenidos en el tiempo, entre otras.

Relaciones mutuamente beneficiosas con el proveedor: Lo principal es que la organización junto a sus proveedores establezcan una relación cordial, creando reciprocidad en los beneficios que se puedan obtener al tener una participación mutua, de esta forma no se maneja tanto el concepto de proveedores sino más bien el de socios.

#### 1.3.1.2. Ventajas y desventajas de la norma ISO 9000

##### ➤ Ventajas:

- Sistema fiable y escalable.
- Hay escalabilidad.
- Existe predictibilidad.
- Planificación efectiva y manejo de riesgos.
- Enfoque estructurado para el desarrollo del software.
- Documento de especificación de requerimientos.
- Enfocado en el cliente.
- Es adecuado para un equipo grande.
- Mejora continua en los procesos de software.

##### ➤ Desventajas:

- No hay flexibilidad.
- Existe bastante burocracia.
- Hay que realizar mucha documentación.
- No es adecuado para un equipo pequeño.

- Da un enfoque muy general, no está basado directamente en industrias de software.
- Costo elevado de implementación.

En la subsección 1.4.2 y las que la conforman se utilizaron las referencias [15], [16], [17], [14], [13], [18].

### **1.3.2. ISO 9001:2000**

Describe los requerimientos que debe tener un sistema de gestión de calidad. Aplicables para que una organización demuestre su capacidad de proporcionar productos, que cumplan con las necesidades de sus clientes, teniendo como objetivo el aumento de satisfacción del cliente. Por ejemplo, el caso de una empresa que necesita asegurar calidad en los productos o servicios que provee a un cliente por medio de un contrato, se toma en cuenta la calidad en el diseño, producción, instalación y servicio postventa.

#### 1.3.2.1. Clausulas de la ISO 9001:2000

##### ➤ 4) Sistema de gestión de calidad

- 4.1) Requerimientos generales: Se deben identificar todas las actividades de la organización como un proceso, determinando la interacción de estos entre sí. También se deben determinar los criterios y métodos para asegurar eficacia en la operación y el control de procesos. Se debe buscar la sinergia entre todos los departamentos e individuos de la empresa, estableciendo un trabajo colectivo entre todos, teniendo una mejora continua en todos los procesos. Y lograr así cumplir a cabalidad las necesidades del cliente logrando una satisfacción óptima en él.

- 4.2) Requerimientos de documentación: Aquí se indican los documentos necesarios en un sistema de gestión de calidad, siendo estos, procedimientos e instrucciones, declaraciones de política de calidad y objetivos de calidad, manual de calidad, los procedimientos requeridos en esta norma, los documentos necesarios para asegurar la planificación, operación y control de procesos, y los registros requeridos por la norma ISO 9001:2000.

➤ 5) Responsabilidades de la dirección

- 5.1) Compromiso de la dirección: La alta gerencia debe estar permanentemente comprometida a cumplir todos los requerimientos y políticas necesarias para llevar a cabo la norma ISO, implementando y mejorando el sistema de gestión de calidad.

- 5.2) Enfoque al cliente: Es importante que la dirección conozca las necesidades del cliente, y estar comprometido a que se cumplan a cabalidad, guiando al equipo correctamente para que esto sea posible.

- 5.3) Política de calidad: La gerencia debe estar comprometida a tener una política de calidad acorde al negocio donde se puedan establecer los objetivos de calidad y específicamente guiar por el camino correcto el buen desempeño de la organización.

- 5.4) Planificación: La Dirección debe establecer adecuadamente los objetivos de calidad medibles, basados en la política de calidad, se deben planificar la implementación y los cambios del sistema de gestión de calidad.

- 5.5) Responsabilidad, autoridad y comunicación: La alta gerencia debe tener una buena comunicación con todos los líderes para hacer cumplir responsabilidades por medio de las autoridades asignadas a cada proceso.

- 5.6) Revisión desde la Dirección: La Dirección debe revisar continuamente el sistema en períodos de tiempo para asegurar que todo se este cumpliendo de la mejor forma.

➤ 6) Gestión de los recursos

- 6.1) Provisión de recursos: La organización debe proveer los recursos necesarios para la implementación y mantenimiento del sistema de gestión de calidad.

- 6.2) Recursos humanos: Las personas deben ser ubicadas en el área de su preferencia para lograr el mejor desempeño en determinado proceso, considerando por supuesto su conocimiento y experiencia sobre el área.

- 6.3) Infraestructura: La infraestructura debe ser la adecuada para lograr un buen desempeño en el proyecto, esto conlleva al hardware y software necesarios.

- 6.4) Ambiente de trabajo: Debe establecerse un buen entorno tanto en la infraestructura como con el equipo de trabajo para lograr crear un producto hecho a la medida del cliente.

➤ 7) Realización del producto

- 7.1) Planificación de la realización del producto: Se debe planificar una secuencia de procesos y subprocessos necesarios para la realización del producto.

- 7.2) Procesos relacionados con los clientes: Todos estos procesos implican una participación directa del cliente para poder generar un producto ad-hoc a sus necesidades.

- 7.3) Diseño y desarrollo: La ISO 9000:2000 define el diseño y desarrollo como un conjunto de procesos que transforma los requerimientos en características especificadas, o en la especificación de un producto, proceso o sistema. La organización debe definir bien los procesos para la creación y revisión continua del producto.

- 7.4) Compras: Se deben elegir a los proveedores adecuados, conociendo sus capacidades, principalmente que cumplan con los requerimientos de la organización.

- 7.5) Producción y prestación del servicio: Se debe llevar a cabo por medio de condiciones controladas, siendo estas los requerimientos que la organización debe cumplir. El control implica que los procesos realicen de la mejor forma las tareas que se desean, es así como se logra que obtener un producto o servicio de alta calidad.

- 7.6) Control de dispositivos de medición y seguimiento: La organización determina cuales dispositivos utilizar para monitorear y medir que el producto cumple con todos los requerimientos establecidos.

➤ 8) Mediciones, análisis y mejora

- 8.1) Generalidades:

Se deben planificar e implementar los procesos de seguimiento, medición, análisis y mejora necesarios para dar una buena calidad del producto, así también para garantizar una mejora continua en el sistema de gestión de calidad.

- 8.2) Supervisión y medición: Se debe verificar la satisfacción del cliente y a la vez, qué es lo que le disgusta, puede ser por medio de análisis estadístico, acerca de cómo el sistema funciona. Todos estos análisis realizados deben ser documentados para poder seguir mejorando el sistema y también ayudar a la creación de los demás productos de la empresa.

- 8.3) Control de servicio no conforme: Todos los productos que no cumplan con los requerimientos especificados por el cliente deben reevaluarse para tratar de redirigirlos, si es mucha la diferencia respecto a las necesidades del cliente entonces se debe volver a crear el producto antes de la entrega.

- 8.4) Análisis de datos: El análisis de datos es necesario para poder saber que tan efectivo es el sistema de gestión de calidad. Así también ayuda a mejorar continuamente el sistema.

- 8.5) Mejora: La organización debe estar comprometida a mejorar continuamente el sistema de gestión de calidad, por medio de planificación y diseño de procesos que puedan determinar que la mejora continua, incluyendo también las acciones correctivas y preventivas que puedan ser tomadas.

### 1.3.2.2. Ventajas y desventajas de la norma ISO 9001

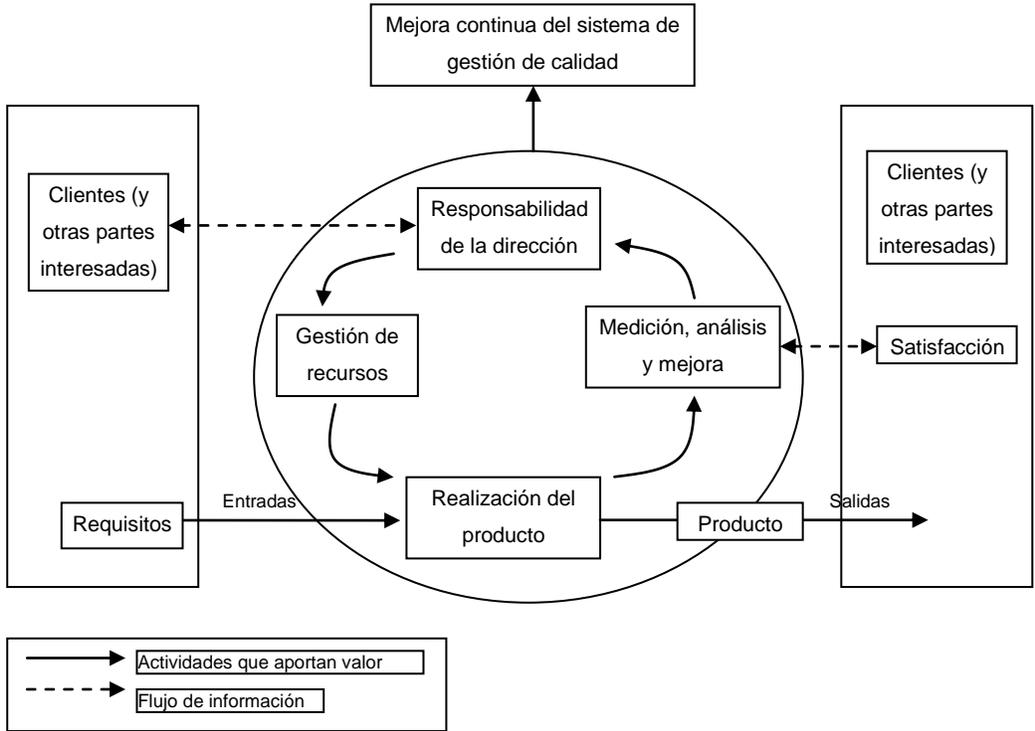
➤ Ventajas:

- Satisfacción del personal, tanto que no tratan de cambiarse a otra empresa.
- Mejora de la eficiencia de los procesos.
- Mejora en la calidad de los productos.
- Si existieran fallas en el producto se debe bajar el costo del mismo.

➤ Desventajas:

- Los objetivos y planes son muy ambiciosos.
- El costo de la certificación es alto.
- El esfuerzo y tiempo que se gasta en establecer y mejorar el sistema de gestión de calidad.
- Por lo general, las compañías no se interesan por la mejora continua.
- Si existe una mala dirección, los procesos de mejora fallan.
- Usualmente se genera mucha documentación, por lo que se vuelve un dolor de cabeza el tratar de hacer algún cambio.

**Figura 2. Modelo de un sistema de gestión de calidad basado en procesos  
Norma ISO 9001:2000**



En esta imagen se utilizó la referencia [16].



## **2. RELACIÓN ENTRE LA PROGRAMACIÓN EXTREMA Y EL ESTÁNDAR ISO**

En el presente capítulo se describen las relaciones encontradas entre la programación extrema y el estándar ISO, principalmente las normas ISO 9000:2000 y 9001:2000. Esta comparación se realiza con base a los conceptos de la XP (principios, practicas y valores) y los correspondientes a la norma ISO (clausulas y requerimientos).

La ISO 9000:2000 define el diseño y desarrollo como un conjunto de procesos que transforma los requerimientos en características especificadas, o en la especificación de un producto, proceso o sistema. Es así, que la organización debe definir bien los procesos para la creación y revisión continua del producto.

En la sección 2.1 se utilizaron las referencias [4], [7], [19], [15], [10].

### **2.1. Características distintivas de la programación extrema y el estándar ISO**

Las condiciones siguientes, son en las que se adaptarían mejor, tanto la ISO como la XP.

### **2.1.1. Características de aplicación del software**

ISO:

- Software predecible, estable, confiable y de alta calidad.
- Por lo regular hay grandes equipos y proyectos.
- Requerimientos y proyectos con estabilidad.

XP:

- Software que es capaz de responder rápidamente a requerimientos cambiantes.
- Pequeños y medianos equipos al igual que los proyectos.
- Requerimientos y proyectos inestables.

### **2.1.2. Características de gestión**

ISO:

- Los clientes se enfocan en negociaciones.
- Compartimiento y transferencia del conocimiento explícita, con documentación.

XP:

- Usuarios enfocados en prioridades.
- Compartimiento y transferencia del conocimiento tácita, sin mucha documentación.

### **2.1.3. Características técnicas**

ISO:

- Definición de requerimientos formales y documentación formal de los mismos.
- Diseños grandes, iteraciones largas y refactorización tardía.
- Plan de pruebas documentado con procedimientos formales.

XP:

- Definición informal de requerimientos, por medio de las historias de usuario.
- Diseño simple que responde a las necesidades del cliente, con iteraciones pequeñas y constante refactorización.
- Pruebas frecuentes, plan de pruebas antes de la codificación por último la ejecución de las pruebas.

### **2.1.4. Características de las personas**

ISO:

- Los clientes no están con los desarrolladores en la mayoría de los casos.
- La experiencia de los desarrolladores es importante solamente al inicio del proyecto.
- Libertad limitada, enfocada en procesos. Caracterizada por el orden.

XP:

- Los clientes están comprometidos con los desarrolladores.
- Desarrolladores con experiencia desde el inicio hasta el final del proyecto.
- Cultura de empoderamiento, libertad, enfocado en las personas, caracterizada por el caos.

En la sección 2.2 se utiliza las referencias [7], [20].

## **2.2. Niveles de métodos del software – Escala de Cockburn**

### **2.2.1. XP**

Nivel -1: El personal posee conocimientos técnicos. Es difícil que se comparta dicho conocimiento.

Nivel 1B: Se pueden utilizar métodos procedurales con el respectivo entrenamiento. Conforme el perfeccionamiento se convierte en el nivel 1A.

Nivel 1A: Se da prioridad a las historias para realizar el desarrollo incremental. Conforme se perfeccione este nivel se pasa al nivel 2.

Nivel 2: Personal capaz de modificar los procesos y adaptarse a una nueva situación.

Nivel 3: Capaz de cambiar por completo un método o proceso, para la creación de un nuevo entorno.

### **2.2.2. ISO**

Nivel -1: El personal debe tener asignada una responsabilidad en lugar de trabajar como un equipo ágil.

Nivel 1: Las personas son muy efectivas en un entorno de desarrollo estable. Por el contrario son inefectivas en un entorno inestable. Estas personas pueden trabajar muy lento en un equipo ágil, debido a su falta de adaptación rápida a los cambios continuos. Se acoplan mejor en un entorno disciplinado y organizado.

Nivel 1A: Se pueden adaptar a una metodología ágil, con el respectivo entrenamiento, con mayor facilidad si existen miembros que tengan conocimientos del nivel 2.

Nivel 2: Se necesita de la ayuda del personal del nivel 3. Es efectivo en equipos pequeños que tengan conocimientos ágiles.

Nivel 3: Existe un buen desempeño, por parte del personal, en proyectos grandes. Se requiere experiencia y la ayuda de equipos de nivel 2.

En esta sección se utilizaron las referencias [15], [5], [4].

### **2.3. Comparación entre la programación extrema y el estándar ISO**

ISO: En la determinación de los requerimientos relacionados con el producto se identifican los requerimientos del cliente, el punto hasta el cual el cliente ha especificado los requerimientos del producto, incluyendo los legales.

XP: Los usuarios definen sus requerimientos en las historias de usuario y de cómo desean que sea el sistema y las ordenan de acuerdo a sus prioridades. Los desarrolladores crean escenarios en base a las historias de usuario.

ISO: La revisión de los requerimientos relacionados con el producto se refiere a la definición y documentación de los requerimientos y el registro de los pedidos verbales.

XP: Se tiene la práctica de entregar pequeñas fases (sprints), que van desde un par de semanas hasta un mes de acuerdo al tamaño del proyecto. Conforme la creación de cada prototipo se revisa continuamente los requerimientos del cliente.

ISO: La comunicación con el cliente son los requerimientos respecto a la manipulación de las consultas y pedidos, información del producto/servicio, voz del cliente y procesos de anulación.

XP: Existe una comunicación bastante fluida entre el equipo desarrollador y el cliente o usuario final, en caso de que el usuario no pueda estar personalmente, tiene alta disponibilidad por cualquier otro medio de comunicación. En caso contrario el cliente está físicamente con el equipo desarrollador indicando sus prioridades y haciendo las pruebas que considera necesarias.

ISO: Las entradas al diseño y desarrollo se refieren a los requerimientos del cliente, requerimientos legales y medioambientales.

XP: Los requerimientos del cliente, respecto al producto que desea, se incluyen en las historias de usuario.

ISO: Los resultados del diseño y desarrollo implican cumplir con requerimientos de entrada, criterios de aceptación y características especiales, para su utilización segura y apropiada, y la aplicación del producto o servicio. Los documentos de salida deben ser analizados antes de su publicación.

XP: Se realizan pruebas después del diseño y la implementación del producto.

ISO: En la revisión del diseño y desarrollo se realiza una revisión formal, se identifican problemas y se evalúa la capacidad de cumplir con los requerimientos.

XP: La programación en parejas ayuda a revisar el código realizado. La naturaleza iterativa de la programación extrema ayuda a revisar continuamente la funcionalidad del sistema durante el desarrollo.

ISO: La verificación del diseño y desarrollo debe ser en etapas planificadas, para asegurar que los resultados del diseño cumplen con los requerimientos de entrada del mismo.

XP: El test de aceptación ayuda a probar la especificación de los requerimientos del cliente con las características del producto. El usuario es quien verifica si las características del sistema coinciden con lo descrito en las historias de usuario.

ISO: En el control de cambios del diseño y desarrollo, antes de realizar el cambio se debe determinar el efecto en el diseño y en las partes del producto o servicio.

XP: Los requerimientos son cambiantes durante el ciclo de vida del proyecto. La programación extrema tiene la flexibilidad necesaria para adoptar los cambios.

ISO: La organización debe hacer un seguimiento y medición de los procesos para asegurar su capacidad, buscando la mejora interna.

XP: La programación en parejas con programación de 40 hrs semanales no se cumple algunas veces, debido a que los integrantes de la pareja no disponen del mismo tiempo, por lo que se dividen el trabajo en 20 hrs trabajando individualmente. Así también, la disponibilidad del usuario puede ser de acuerdo al horario que disponga, o cuando se estrictamente necesaria su presencia. Algunas veces se hace necesario trabajar más de las 40 hrs semanales.

ISO: En el seguimiento y medición del producto se debe verificar el cumplimiento de los requerimientos especificados para el producto/servicio.

XP: El usuario se mantiene retroalimentando constantemente la creación de cada prototipo, es aquí donde el usuario también le da seguimiento y medición al producto que está siendo creado a lo largo de la realización de la fase.

ISO: En el control del servicio/producto no conforme todos los productos que no cumplan con los requerimientos especificados por el cliente deben reevaluarse para tratar de redirigirlos, si es mucha la diferencia respecto a las necesidades del cliente, entonces se debe volver a crear el producto antes de la entrega.

XP: El plan de pruebas es creado antes de la codificación del proyecto y las pruebas siempre son requeridas para realizar la nueva iteración. Las pruebas funcionales son las que se ejecutan después de codificar y la aceptación se da hasta la integración final.

ISO: En la acción correctiva se debe eliminar y reducir las causas de la no conformidad, utilizando las quejas de los clientes y el sistema de gestión de calidad.

XP: Existen actualizaciones al plan de pruebas.



### 3. PROGRAMACIÓN EXTREMA Y EL ESTÁNDAR ISO EN GUATEMALA

#### 3.1. Análisis preliminar

El trabajo de entrevistar a más de 30 personas del campo de la ingeniería del software, entre ellas ingenieros en sistemas y analistas-desarrolladores, quienes son empleados de 27 empresas ubicadas en la ciudad capital de Guatemala (ver tabla 1), se realizó en las fechas siguientes: 17/09/09, 18/09/09, del 23/09/09 al 26/09/09, del 12/11/09 al 15/11/09, 28/11/09 y 07/01/10. Las entrevistas fueron realizadas por medios electrónicos (correo-e, mensajería instantánea) y otras de forma presencial.

**Tabla I. Organizaciones y personas entrevistadas**

No.	Empresa o Institución	Persona entrevistada
XP		
1	Fundación de Antropología Forense - FAFG -	Gianni Leiva
2	IGSS	Marco García y otros.
3	Ministerio Público	Otto Santizo
4	Universidad Francisco Marroquín - UFM -	Ing. Gustavo Alvarado
ISO		
5	ACS-BPS Guatemala	Ing. Julio Rodas y otros.
6	Latin American Byte, Inc.	Jorge Díaz
7	Ministerio de Educación - MINEDUC -	Ing. Bayron López y otros.

XP+Tradicional		
8	AGEXPORT	Jorge Say y otros.
9	BANRURAL	Julio Chicas
10	Business Development Group S.A. - BDG -	Erick Icute
11	Sega Outsourcing Services -SEGA-	Pedro Santos
Cascada+Espiral		
12	ACE Technologies	Felipe Roche
13	Banco GyT Continental	David Guerra
14	DIPCOMI	David García
15	E-Solutions	Ing. Daniel López y otros.
16	Superintendencia de Administración Tributaria (SAT)	Ing. Ricardo Morales y otros.
Scrum		
17	Aldea Systems, S.A.	Gerson Pérez
18	Open Training	
No definida		
19	CANELLA	Carlos Valdez
20	Consolidados 807	Jennifer Martínez
21	INACIF	Mario Pecorelli
22	ISUZU Autocom	Steve de la Cruz
23	KTFSoluciones	Carlos Recopachi
24	NetGameServices - NGS -	Hesler Solares
25	Secretaría Ejecutiva de la ICMSJ/AECID	Edwin Nájera
26	Universidad del Istmo - UNIS -	Julián Figueroa
27	USAC – Fac. de Ingeniería – Centro de Cálculo	Wendy Juárez y otros

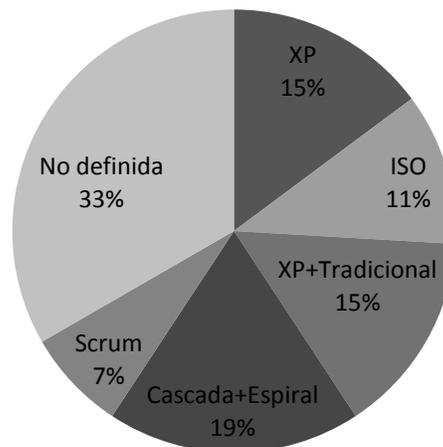
## Script de la entrevista

Para la realización de la entrevista, a cada persona, fue necesaria la utilización de las siguientes preguntas.

1. ¿Qué metodología de desarrollo utilizan en el departamento de desarrollo de su empresa?
2. ¿Cuál es la duración aproximada de cada proyecto que realizan?
3. ¿Cuántos desarrolladores realizan un proyecto?
4. ¿Qué nivel considera que tienen los desarrolladores?
5. ¿Cuántas personas hay aproximadamente en el departamento de desarrollo?

A continuación se presenta un análisis de la situación en que se encuentra un sector de la industria del software en Guatemala respecto a la utilización de las metodologías de desarrollo. El total de la muestra es de 27 empresas.

**Figura 3. Utilización de las metodologías de desarrollo en Guatemala**



### **3.2. Programación extrema en Guatemala**

Se ha comprobado que la utilización, de forma explícita, de la programación extrema en Guatemala es baja. Esto se fundamenta en los resultados obtenidos de las entrevistas realizadas a ingenieros y personas del campo de la ingeniería del software. Se determina que la programación extrema no es utilizada en Guatemala probablemente por la cultura del equipo desarrollador, o del jefe de proyecto, quienes no conocen o entienden a cabalidad los conceptos que implica esta metodología.

En esta investigación, al presentar los resultados, cuando se menciona a la XP se hace referencia principalmente a la pseudo-utilización de la misma y no así a la implementación completa de dicha metodología.

El 15% de las organizaciones, de un total de 27, utilizan en su departamento de desarrollo de forma implícita la programación extrema, esto significa que no aplican totalmente esta metodología, sino solamente algunos de sus principios.

Estas organizaciones cuentan con un departamento de desarrollo pequeño y otras de mediano tamaño. El equipo se compone de un promedio de 15 personas. De acuerdo al tamaño del proyecto que se realiza, dividen al equipo en grupos de 4 personas por proyecto. Estos tienen una duración aproximada de 6 meses y lo van entregando en fases mensualmente y en algunos casos cada 3 semanas. Un 70% de los integrantes de la empresa tiene bastante conocimiento técnico, se pueden caracterizar como de los niveles 2 y 3 de la escala de Cockburn, el resto entra en los niveles 1B y 1A.

Cada vez que se entrega una fase al cliente, este puede sugerir algún cambio si es que lo considera necesario. La cultura del equipo se caracteriza por el caos.

El caso más cercano que se pudo encontrar en donde utilizan programación extrema, en un 80%, es en el departamento de desarrollo de la Fundación de Antropología Forense de Guatemala - FAFG -, en la que se entrevistó a parte del personal. Esta es una organización no gubernamental, autónoma, técnico-científica, sin fines de lucro que contribuye al fortalecimiento del sistema de justicia y al respeto de los derechos humanos; a través de la investigación, la documentación, la divulgación, la formación y la sensibilización de los hechos históricos de violaciones al derecho a la vida y de casos de muerte no esclarecidos.

Los proyectos son realizados en parejas, son aplicaciones pequeñas, estas tienen un tiempo de vida de aproximadamente un año, luego de este tiempo solo se utiliza la información proporcionada por estas aplicaciones para hacer reportes y estadísticas. La duración del desarrollo de cada proyecto es de aproximadamente 3 semanas, para ello los usuarios, en este caso los digitadores, indican a los programadores que necesitan ingresar cierta información, muestran una formulación en papel como debería de verse la aplicación, los interesados en la aplicación están en contacto permanente con el equipo desarrollador, haciendo los cambios que consideran necesarios. Se le muestra un prototipo al digitador y si la acepta se instala o se hacen los cambios requeridos.

## **XP+Tradicional**

Un 15% de las organizaciones utilizan implícitamente la XP combinándola con una metodología tradicional como cascada. El tamaño de los proyectos que manejan va desde 6 meses hasta 2 años. La comunicación con el cliente no es tan frecuente, él manifiesta su punto de vista cuando le entregan la fase del proyecto, si considera necesario realizar algún cambio, lo indica y el equipo de desarrollo lo hace como parte de una actualización en la aplicación.

El equipo de desarrollo tiene un promedio de 25 personas, y de acuerdo al proyecto se divide en grupos de 6 ó 5 personas la realización de cada proyecto. La documentación que utilizan no es tan extensa. El equipo se caracteriza por el caos. Tienen un 60% de personas de los niveles 2 y 3 y el resto del nivel 1A y 1B.

### **3.3. Estándar ISO en Guatemala**

De acuerdo a la investigación realizada, actualmente existe un 11% de 27 instituciones, que tienen un certificado con Normas ISO 9000:2000 y 9001:2000 en Guatemala.

Se observa que la mayoría de las instituciones certificadas con el estándar ISO tienen un tamaño grande. Por ejemplo, en el departamento de informática de ACS-BPS Guatemala, hay aproximadamente 400 personas, mientras que las otras organizaciones, MINEDUC ( [22], [23]) y Byte, cuentan con un promedio de 100 personas. Se determinó que se necesita la inversión de una buena cantidad de recursos y a la vez de una cultura organizacional adecuada para lograr mantener la certificación. Aquí la cultura del equipo de desarrollo se caracteriza por el orden.

Estas empresas tienen personas certificadas por parte del ente certificador que entrenan a todas las personas de la organización. Y de esta forma también ayuda a mantener el certificado ISO.

Los proyectos tienen una duración de 6 meses hasta 2 años, y pueden existir proyectos que necesitan más de este tiempo.

Las entregas se realizan cada mes o cada 2 meses de acuerdo al proyecto en cuestión y existe poco dinamismo, se observa que están regidas por el orden. Los proyectos de acuerdo a su tamaño se asignan a grupos de 3 ó 4 personas, y en algunos casos a menos personas. El tamaño del departamento de desarrollo considera a personas con un alto grado de conocimiento los cuales se pueden categorizar como 75% del nivel 2 y 3 de la escala de Cockburn y el resto de los niveles 1B. La cultura del equipo se rige por el orden.

### **3.4. Otras metodologías**

#### **Scrum**

Al igual que en el caso de la XP, las empresas que dicen utilizar Scrum no lo aplican en un 100%. En este caso solamente un 7% de las organizaciones encuestadas dijeron que utilizaban Scrum. Se identificó que dichas organizaciones indican que usaban esta metodología por el grado de agilidad que ellos emplean para realizar los proyectos y el tamaño de los proyectos que manejan. Realizan fases de 1 mes para cada proyecto, la mayoría de los proyectos que tratan, tardan aproximadamente de 6 meses a un año.

## **Cascada + Espiral**

Existe un 19% de organizaciones que utiliza las metodologías tradicionales cascada junto con espiral. Al igual que en los casos anteriores no lo aplican en un 100% como lo indica la teoría sino que más bien lo adaptan de acuerdo a las necesidades de los proyectos que realizan. Son empresas que cuentan con un departamento de desarrollo de 50 personas en promedio. Los proyectos tienen una duración de 6 meses hasta 2 años o más y realizan entregas en algunos casos cada 2 ó 3 meses. Si hay necesidad de realizar algún cambio lo manejan como una actualización al sistema. Para la realización de cada proyecto asignan equipos de un promedio de 10 personas.

Manejan cierto grado de formalidad en la documentación que realizan. La comunicación con el cliente no es muy frecuente. El grado de dinamismo es bajo. El nivel de las personas se considera entre el 1B y el 3 de la escala de Cockburn.

## **No definida**

Se determina que un 33% de las organizaciones, principalmente aquellas que son pequeñas y medianas, no cuentan con una metodología de desarrollo claramente definida. En ellas existe una cultura de caos. Debido a la carencia de la utilización de una metodología se encuentran muchas veces con problemas de entrega a tiempo del software o de cumplir a cabalidad con los requerimiento de los clientes. Existen varias ocasiones en las que necesitan hacer reajustes al software creado, de acuerdo a las necesidades del cliente. Cuentan con un equipo de desarrollo de 10 personas aproximadamente, dividen a las personas en grupos de 3 o en parejas, y a cada equipo se le asigna más de un proyecto a realizar.

La duración de cada proyecto es de 6 meses a 1 año y realizan entregas cada 2 meses directamente al cliente. El 70% del personal se considera dentro del nivel 1B e inferiores, mientras que el resto está en los niveles 2 y 3 de la escala de Cockburn.



## **4. METODOLOGÍA HÍBRIDA XP-ISO**

### **4.1. Análisis preliminar**

En Guatemala, la mayoría de las empresas que aplican el estándar ISO son grandes, mientras que las que aplican la programación extrema son medianas y en su mayoría pequeñas.

Al considerar la combinación de la programación extrema junto a la norma ISO 9000:2000 y 9001:2000, se encuentra necesario saber la situación en la que se ubica actualmente la empresa. Esto debido a que, existen empresas que cuentan con la certificación del estándar ISO y otras que tienen alguna experiencia en la pseudo-aplicación de la programación extrema. También están las empresas que implementan en cierto grado, tanto la norma ISO como la XP. Y por último las que no poseen ninguna de las dos metodologías, por lo que en cierta forma deben iniciar desde cero la aplicación de la metodología híbrida.

Es así, como se ve la necesidad de crear los esquemas necesarios para los diferentes casos de la situación en que la empresa se encuentra. Para ello es necesario utilizar 5 factores tanto para la XP como para la ISO - cultura, personal, tamaño, criticidad, dinamismo -. Más adelante se encuentra la forma de evaluar la situación actual de la organización.

La programación extrema está basada en los principios y valores del manifiesto ágil.

Por aparte, el estándar ISO se fundamenta en procesos disciplinados y la utilización de planes explícitos para el desarrollo del producto, en este caso el software.

Al combinar la XP con el estándar ISO se obtienen procesos del desarrollo de software disciplinados y ágiles. En un proyecto grande es necesario tener tanto la disciplina como la agilidad ya que es necesario terminar cuanto antes dicho proyecto. Existen casos en Guatemala en los que hay proyectos grandes, pero por no tener metodologías adecuadas se obtienen pérdidas económicas debido a la lentitud del desarrollo y también cuando hay necesidad de realizar cambios a los requerimientos del cliente.

Para combinar la programación extrema con el estándar ISO se adaptarán, a conveniencia para el desarrollo en Guatemala, los factores de decisión crítica de Boehm y Turner. Con estos factores se puede determinar en el punto se encuentra la organización o el proyecto que se realiza. Si se encuentran más cercanos a la aplicación ya sea de la XP, la ISO o está en medio entre ambas.

Los factores de decisión crítica referidos son cultura, personal, criticidad, tamaño, dinamismo.

En la subsección 4.2 y 4.3 se utilizaron las referencias [7], [4] , [20].

## **4.2. Factores de decisión crítica**

### **4.2.1. Cultura**

ISO: Es más efectivo en un entorno en donde las personas se sienten cómodas y con empoderamiento. Tienen sus roles definidos claramente en la organización por medio de documentos.

XP: Es más adecuado en un entorno en donde los miembros del equipo se sienten cómodos utilizando sus propias iniciativas. Y da bastante libertad a los miembros del equipo.

### **4.2.2. Personal**

ISO: Necesita varios expertos de los niveles de Cockburn 2 y 3, durante el inicio y estabilización del proyecto. Conforme el avance del proyecto es suficiente con tener miembros del nivel 1B y una pequeña cantidad de expertos del nivel 2 y 3.

XP: Requiere experiencia de los miembros del equipo. Y expertos del nivel 2 y 3 de Cockburn.

### **4.2.3. Criticidad**

ISO: Evoluciona para adaptarse a proyectos críticos. Es difícil adaptarse a proyectos de baja criticidad.

XP: Carece de diseños grandes y documentación detallada. No se han realizado pruebas con proyectos altamente críticos.

#### **4.2.4. Tamaño**

ISO: Es adecuado para proyectos y equipos grandes. Es difícil tener una escala hacia abajo, con pequeños equipos y proyectos.

XP: Es adecuado para equipos y proyectos pequeños. Carece de escalabilidad, debido a que el conocimiento tácito es el único medio de la transferencia de dicho conocimiento.

#### **4.2.5. Dinamismo**

ISO: Planes detallados, diseños grandes, la documentación específica y comprensiva es ideal para un entorno altamente estable. Es muy costoso realizar una reingeniería hacia un ambiente dinámico.

XP: Diseños simples, iteraciones pequeñas, refactorización e integración continua, la hacen más apta para entornos altamente dinámicos. Es muy costoso realizar una reingeniería hacia un ambiente estable.

### **4.3. Representación gráfica de los factores de decisión crítica**

Determinado proyecto se puede medir dentro de los ejes de la gráfica 3, de acuerdo a que tanto se aproxima en la gráfica. Si está demasiado cerca de la ISO o la XP lo más aconsejable sería utilizar la metodología pura.

Sin embargo, si el proyecto se encuentra distante de los extremos de los ejes, es decir en medio de ambos extremos, sería más difícil y arriesgado aplicar la metodología pura, por lo que en estos casos sería más recomendable utilizar la metodología híbrida XP-ISO.

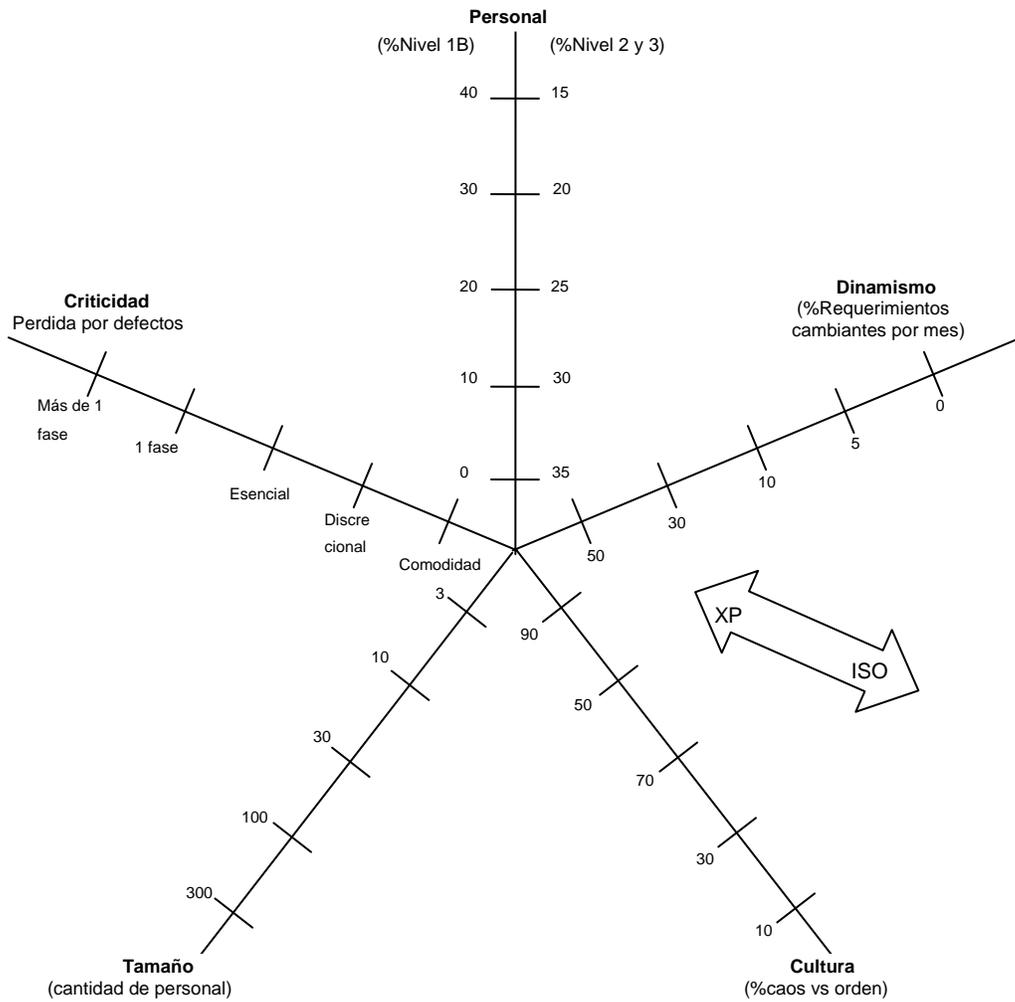
Los ejes del tamaño y la criticidad son los más utilizados para la diferenciación de métodos ágiles y los estrictos. Los ágiles están cerca del centro, en el gráfico, mientras que los estrictos se acercan a los extremos.

La cultura indica que la XP es efectiva en procesos caracterizados por el caos, a diferencia de la ISO que se caracteriza por el orden.

Los ejes del dinamismo y el personal son asimétricos, esto significa que tanto lo ágil y lo disciplinado son bastante efectivos, independientemente de la dirección que tomen. El dinamismo indica que la XP funciona, tanto con algo rápido, como con algo de velocidad normal.

El eje del personal, que se basa en la escala de Cockburn, se enfoca en la adaptación del equipo, a los cambios, conforme lo requiera el proyecto. Una persona que desarrolla proyectos simples puede ser considerada de nivel 2 mientras que alguien que realiza algo más complejo se le da el nivel 1A. La ISO puede trabajar en organizaciones con personal de niveles altos y bajos, mientras que la XP trabaja con organizaciones en las que su personal es principalmente de los niveles altos.

**Figura 4. Gráfico polar – Factores de decisión crítica**



En esta imagen se utilizó la referencia [20].

En la sección 4.4 y subsecciones se utilizan las siguientes referencias [7], [24], [20].

## **4.4. Evaluación de la situación actual de la organización**

Una organización de Guatemala debe seguir los siguientes pasos en la auto-evaluación de su situación actual. Se debe tomar en cuenta a todos los involucrados en el proyecto (equipo de desarrollo, usuarios, entre otros), para obtener un mejor resultado.

### **4.4.1. Paso 1**

Realizar una evaluación de los proyectos de la organización haciendo una medición en la figura 3, para ver en donde se encuentra el proyecto respecto a los 5 ejes. Es necesario verificar los resultados esperados, como máximo, en los próximos 12 meses a partir de la implementación.

### **4.4.2. Paso 2**

Si al realizar la medición se observa que el proyecto tiende hacia el centro de la figura 3, significa que se está utilizando principalmente las practicas de la XP. Por el contrario si la posición del proyecto tiende hacia los extremos o la periferia, indica que la organización está bastante cerca de las prácticas del estándar ISO.

Si después de haber realizado la evaluación, se obtiene cualquiera de los dos casos mencionados anteriormente, la aplicación pura de la XP o la ISO independientemente, se debe continuar utilizándolas así, y utilizar procesos de mejora continua para utilizar lo mejor de ambas, la XP y la ISO.

Para realizar los procesos de mejora continua se debe tomar en cuenta lo siguiente:

- Crear un comité que incluya a todos los involucrados en el proyecto (*stakeholders*) para evaluar, analizar y recomendar técnicas de mejora continua que mejor se adapten a la situación actual.
- Para actualizar el proyecto con procesos de mejora continua, se debe tener un equipo entusiasta, comprometido y capacitados para llevar a cabo el proyecto.
- Conforme se consolide la aplicación de la mejora continua, se debe extender hacia toda la organización.

#### **4.4.3. Paso 3**

Si después de realizar la evaluación se encuentra que se están aplicando la XP o la ISO de manera individual pero con las siguientes características:

- Bastante comodidad, que es caracteriza de la ISO, sin embargo el entorno es muy dinámico.
- La comodidad es efectiva con desarrolladores ágiles, pero los usuarios son empresas integradas y aplicaciones confiables.
- El equipo técnico se encuentra en un ambiente dinámico. Sin embargo, el proyecto no está obteniendo los mejores resultados.

Estas características son consideradas como factores de riesgo, en estos casos se debe también implementar procesos de mejora continua, tomando en

cuenta los riesgos con un análisis detallado y gestionándolos con buenas técnicas.

Otra alternativa es adoptar la estrategia de combinar la XP con la ISO, en los casos donde sea aplicable. Por ejemplo, en donde se utiliza la ISO, modificarse para que pueda combinarse con un ambiente dinámico. Y en la XP modificar sus prácticas para que pueda ser aceptada por el estándar ISO.

#### **4.4.4. Paso 4**

Si con la evaluación se obtiene que la organización y el proyecto, están en un entorno híbrido de la XP y la ISO, es mejor utilizar una estrategia híbrida incremental.

Por ejemplo consideremos la siguiente situación. Una organización que realiza un proyecto con un equipo de 50 personas aproximadamente. Se encuentra en el centro del eje crítico, es decir se categoriza como esencial. Tiene un personal con el 20% de personas de los niveles 2 y 3 y un 30% de personas del nivel 1B. La cultura está en un 30 % de caos. Un dinamismo que crece de 5% mensual al 10 % mensual. Ver figura 5.

En este caso se debe entrenar al equipo para adaptarse correctamente al dinamismo y realizar un plan donde se cambie la cultura del orden al caos, dado que se necesita mayor agilidad ya que solo se posee el 30% de dinamismo. Para convertir al equipo con mayor agilidad se necesitan varias personas del nivel 2 y 3. El equipo que se tiene actualmente se puede entrenar para subirlo del nivel 1B a los siguientes niveles.

Se puede contratar un consultor experto en XP, sin embargo, sería de realizar un análisis de factibilidad para ver si es adecuado o no contratarlo, esto para ayudar a convertir los procesos de la empresa.

#### **4.4.5. Paso 5**

Al elegir alguna de las opciones descritas anteriormente, se debe tratar de sostener la realización completa, sin estar cambiando de estrategia. No olvidar que lo principal a tomar en cuenta son: las personas, los valores, la comunicación, la retroalimentación continua y los resultados que se esperan obtener. Se debe dar un seguimiento al avance del proyecto, aplicando las medidas necesarias para cumplir con las metas trazadas al inicio.



## **4.5. Combinación de la programación extrema y el estándar ISO**

En esta sección se presenta una guía de la forma en que una empresa que aplica la XP pueda obtener una certificación ISO 9000. Se estudia la forma de combinar la agilidad y la disciplina en los proyectos.

En tal caso se modifican las prácticas de la programación extrema para adaptarlas a la ISO 9001:2000, principalmente en las áreas de realización del producto, medición y seguimiento, contenidas en los capítulos 7 y 8 de la ISO 9000, y mencionadas en la sección 1.3.2 de este trabajo de graduación, que contienen los requerimientos para la metodología a ser empleada por la organización.

### **4.5.1. Realización del producto**

Para modificar las prácticas de la XP con un enfoque hacia la ISO 9001:2000 es necesario utilizar el modelo de madurez XP - XPMM -. Dicho modelo es utilizado en la iniciación de la utilización de esta metodología en determinada organización. Este modelo garantiza las prácticas de programación de la XP, ya que muchas veces existen lugares en las que al no utilizar buenas prácticas de programación y documentación utilizan como excusa a la programación extrema, debido a que no entienden a cabalidad que es la XP. Es necesario utilizar el modelo de madurez para la XP, que ayuda a indicar los riesgos del proyecto y garantizar la utilización exacta de esta metodología. Se proponen 4 niveles para las prácticas de la XP.

Nivel 1 - No cumple

Nivel 2 - Relación con el cliente

Nivel 3 - Calidad del producto, Programación en parejas

Nivel 4 - Desempeño del proyecto

### **Relación con el cliente:**

Una característica importante de la programación extrema, que se incluye en el nivel 2, del grado de madurez de la XP, es el énfasis en la satisfacción al cliente.

Las prácticas de la XP, que se enfatizan en la satisfacción del cliente, se tratan a continuación.

Los usuarios o clientes de la aplicación a crear escriben sus requerimientos y lo que esperan obtener del sistema en las historias de usuario. Estas se crean de una manera simple sin tanto detalle, a diferencia de lo que se realiza en un sistema de desarrollo tradicional.

El proceso de desarrollo de la XP se divide en fases entregables que duran de 6 a 9 semanas. Estas fases a su vez se dividen en 2 iteraciones realizadas en un tiempo de 2 a 3 semanas en las que se desarrolla de acuerdo a las historias de usuario. Con este proceso se obtiene una retroalimentación continua entre los desarrolladores y el cliente. De esta forma se ayuda a crear software a la medida del cliente y con calidad.

Al practicar el juego de la planeación se permite participar activamente al cliente en el proyecto, dando sus puntos de vista, trabajando de esta forma en conjunto con el equipo de desarrollo. Este proceso implica también un análisis y

gestión de riesgos, esto ayuda a priorizar las historias de usuario e implementarlas después de realizar un análisis de factibilidad de costos y riesgos.

Es necesario utilizar la metáfora, para que el cliente se sienta cómodo. Cuando el equipo de desarrollo necesita utilizar palabras técnicas, es necesario que los desarrolladores utilicen una metáfora para que el cliente pueda entenderlo claramente y de esta forma se involucre en el proyecto. Así, se logra una adecuada comunicación entre el cliente y los desarrolladores.

Las prácticas extremas ayudan a que el cliente indique a cabalidad las funcionalidades que el sistema en cuestión debe tener, evitando que los desarrolladores creen funcionalidades que el cliente no requería.

Es necesario ubicar al cliente junto a los desarrolladores para una planeación efectiva, una buena comunicación y retroalimentación continua.

Es importante tomar en cuenta la velocidad del proyecto para la toma de decisiones y la planeación.

Se puede ver que en la XP a diferencia de la ISO 9001:2000 en las cláusulas 7.1, 7.2.2 y 7.3.2 se especifica la necesidad de tener los requerimientos documentados. De esta forma habría una contradicción con las prácticas de la XP en la que los requerimientos no se documentan tan detalladamente. Sin embargo, es claro que la programación extrema no omite la documentación sino que simplemente no se utiliza mucha documentación. El mayor reto es la manera de incluir la documentación en la programación extrema, que no es una práctica importante dentro de esta metodología, ya que

es ágil y se enfoca en que el programador codifique y no se distraiga en otras cosas.

Para unificar la documentación de la ISO con la XP se debe trasladar la tarea de documentar, lo más importante, ya sea a los *testers*, encargados de realizar las pruebas, o los jefes del proyecto, esto con el fin de que los programadores se dediquen específicamente a codificar. Puede ser que los *testers* trabajen en conjunto con el jefe de proyecto para ponerse de acuerdo que es lo que van a documentar y realizar ellos la documentación en conjunto, dependiendo de las características del proyecto realizado.

El hecho de responsabilizar al *tester* de documentar, es debido a que es quien trabaja más de cerca con el cliente, él es quien define las pruebas funcionales del cliente. También realiza las pruebas de aceptación de la aplicación y es quién guía al cliente en la forma de realizar alguna prueba. El *tester* es quien se relaciona con todos los involucrados en el proyecto - clientes desarrolladores y jefes de proyecto -.

Lo importante a documentar son las historias de usuario, las pruebas planeadas, el resultado de las pruebas realizadas, resultado de pruebas de aceptación, retroalimentación del cliente y aquellas que el jefe de proyecto considere importantes a incluir. No olvidar, que la documentación a crear, no debe ser excesiva, sino más bien incluir los puntos importantes siendo breves y concisos en la redacción de la documentación. Para efectos de aceptación, por parte del estándar ISO, también será necesario llevar un control más estricto de las historias de usuario, almacenándolas digitalmente, y colocando un detalle del avance del proyecto en cada historia de usuario y las historias de usuario que se han concluido y aceptado por los clientes. No está demás decir que debe llevarse un control estricto en las pruebas y almacenar todo lo

concerniente a ellas digitalmente para tener una mejor documentación posteriormente.

Lo ideal sería que el *tester* documentará cada vez que se realiza una fase del proyecto, claro está después de la aceptación total de las partes involucradas, para que al final del proyecto sólo se unifique la documentación de cada fase, y sea una integración más simple.

### **Aseguramiento de calidad del producto:**

La programación extrema garantiza la calidad por medio de las pruebas antes de la codificación, siendo esta una forma fácil la de codificar después de haber creado la prueba.

Esto también ayuda al desarrollador a prever y en otros casos a entender qué es lo que se debe realizar, codificando así, de manera más precisa los requerimientos.

Otra práctica es la utilización de pruebas unitarias, importantes en las pruebas de regresión. Las pruebas de regresión se realizan repetidamente con el objetivo de que al modificar el código no se introdujo ningún error.

Existe una integración continua, esto ayuda a retroalimentar al equipo y detectar cualquier error rápidamente, evitando los errores que surgen al tener una integración tardía.

Las practicas de la programación extrema coinciden con los requerimientos indicados en la clausula 7.3.5 de la ISO 9000. Es aquí en donde se indica que debe realizarse una verificación o revisión en la que se

compruebe que el diseño y los resultados del desarrollo satisfacen los requerimientos correspondientes al diseño inicial.

### **Programación en parejas:**

La codificación debe realizarse de acuerdo a un estándar organizacional en el que sea fácilmente modificable y entendible por cualquier integrante de la empresa.

El hecho de programar en parejas da un mejor resultado en la codificación y de esta forma se comparte de mejor manera el conocimiento, evitando que el conocimiento lo posea solamente un individuo, siendo más fácil de modificar por cualquiera, aquí nadie es propietario del código.

Para adaptar esta práctica con la ISO 9000, se debe realizar la programación en parejas en un área de trabajo compartida tal como lo indica las cláusulas 6.3 Infraestructura y 6.4 Ambiente de trabajo. Con esto se garantiza una comunicación eficaz entre todo el equipo del proyecto, compartiendo el conocimiento entre sí.

### **4.5.2. Medición y seguimiento**

La cláusula 8.2.3 de la ISO 9001:2000 indica que la organización necesita comprobar la forma de que un proceso pueda alcanzar sus objetivos. Esta cláusula se puede unificar con la XP por medio de las prácticas descritas a continuación.

Disponibilidad del cliente para resolver dudas que surjan al equipo desarrollador.

Estimación de un tiempo extra para la realización del proyecto.

Se debe verificar la velocidad del proyecto por medio de métricas que indiquen el tiempo que el equipo del proyecto debe invertir en cada tarea. Normalmente esta métrica son las horas, días o meses. También se debe establecer con qué frecuencia se debe realizar la integración del proyecto, esto es recomendable realizarlo diariamente. Se debe establecer que fragmentos de código serán creados en parejas y cuales se crearán individualmente.

También se debe medir que tan rápido codifica la persona, una recomendación sería por medio de las líneas de código, aunque lo principal en este caso sería la funcionalidad de la aplicación y no tanto el tamaño del código.

La clausula 8.2.4, de la ISO 9001:2000, seguimiento y medición del producto indica que debe crearse un reporte de las pruebas de las unidades creadas y de las pruebas de aceptación. Esta clausula se adapta fácilmente con la XP por medio de un registro de las pruebas de aceptación y pruebas unitarias.

La clausula 8.3 Control de servicio no conforme indica que el producto que no cumpla con los requerimientos especificados por el cliente debe reevaluarse para tratar de redirigirlo. Por medio de la realización de pruebas unitarias antes de la entrega al cliente, la XP cumple con esta clausula de la ISO 9001:2000.

Con el análisis previo, se comprueba que la programación extrema garantiza la calidad del producto, coincidiendo con lo indicado en la ISO 9000.

## CONCLUSIONES

1. Las organizaciones de Guatemala que están certificadas con la ISO cumplen con las prácticas y cláusulas indicadas en los documentos de la ISO 9000. A diferencia de las que dicen utilizar la programación extrema, en la mayoría de ellas no se cumple en un 100% con sus prácticas y/o principios.
2. Las pequeñas empresas y algunas medianas organizaciones de Guatemala, no utilizan una metodología de desarrollo de software específica, sino que más bien hacen uso de varias prácticas de desarrollo, en ocasiones ambiguas, que no definen con claridad un proceso de seguimiento en la creación del software.
3. La satisfacción del cliente y la producción de un software de calidad con procesos flexibles son algunas de las ventajas obtenidas con la combinación de la Programación Extrema junto a la norma ISO 9000:2000 y la norma ISO 9001:2000.
4. La implementación de la metodología híbrida XP-ISO implica que cada organización, por medio de todos los involucrados, realice una evaluación a sí misma respecto al grado de criticidad, dinamismo, tamaño y cultura. De acuerdo a los resultados de dicha evaluación se debe decidir si es factible aplicar en su organización la metodología híbrida XP-ISO.



## RECOMENDACIONES

1. Buscar organizaciones que estén dispuestas a practicar la metodología híbrida. Se debe tomar en cuenta que esto implica riesgos, quizá uno de los principales es el económico. Sin embargo, de dar el resultado esperado, se pueden obtener ganancias significativas en la empresa. Lo importante sería aplicar esta metodología en organizaciones que no empleen una metodología específica en su desarrollo actual, o en empresas que estén iniciando sus gestiones como desarrolladoras del software.
2. Dado que la metodología propuesta es nueva y aún no se ha escrito mayor información sobre ella, resulta adecuado que se realicen más investigaciones sobre la misma, incluyendo prácticas claramente detalladas, y de esta forma ayudar a la formalización de esta metodología híbrida XP-ISO.



## BIBLIOGRAFÍA

- [18] Aldana Acajalon, Jorge Mario. Diseño de un manual de calidad basado en la norma ISO 9001:2000, aplicado a una planta de confección. Trabajo de graduación Ing. Industrial. Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2004. 88 pp.
- [30] Boehm, Barry y Richard Turner. "Agile Project Leadership Network". [Online]. <http://www.agileprojectmgt.com/docs/Boehm20.pdf>
- [4] Beck, Kent. **Extreme Programming explained: Embrace Change**, Primera edición. USA: Editorial Addison-Wesley, 2000. 190 pp.
- [3] Canós, José. "Metodologías Ágiles en el Desarrollo de Software". 8 pp. [Doc. Online]. <http://www.willydev.net/descargas/prev/TodoAgil.pdf>
- [9] Cumes Rodríguez, Edgar Oswaldo. Implementación de un sistema de calidad en el departamento de informática de la Dirección de contabilidad del Estado, según la norma ISO 9004-2. Trabajo de graduación Ing. Ciencias y Sistemas Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 1999. 121 pp.
- [28] Cunningham, Ward. "Manifiesto for Agile Software Development". 2001. [Online]. <http://agilemanifesto.org/>
- [16] Díaz Castillo, Claudia Larissa. Implementación de un sistema de gestión de calidad ISO 9001:2000 en Centro Educativo de Ingenio La Unión (CEDILU). Trabajo de graduación Msc. Ing. Gestión Industrial Guatemala, universidad de San Carlos de Guatemala, Escuela de Postgrado, 2008. 43 pp.
- [19] Fernandez Pereda, Héctor. "Norma ISO 9001:2000". 2008. [Online]. [http://www.buscarportal.com/articulos/iso\\_9001\\_2000\\_gestion\\_calidad.html](http://www.buscarportal.com/articulos/iso_9001_2000_gestion_calidad.html)

- [17] Grimes, Kevin R. **ISO 9001:2000 A practical quality manual explained**. USA: Editorial ASQ, 2003. 140 pp.
- [6] Hernández Ramírez, Pedro Pablo. Metodologías para el desarrollo de software: un análisis comparativo. Trabajo de graduación Ing. Ciencias y Sistemas Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2004. 153 pp.
- [14] ISO. "Norma ISO 9001:2000. 2009". [Online]. <http://www.homoqualitas.com/castella/infos/iso90002000/portada.htm>
- [13] ISO. "Orientación acerca del enfoque basado en procesos para los sistemas de gestión de calidad". ISO/TC 176/SC 2001. 12 pp. [Online]. [http://www.iram.org.ar/Documentos/Certificacion/Sistemas/ISO9000\\_2000/requisitos.pdf](http://www.iram.org.ar/Documentos/Certificacion/Sistemas/ISO9000_2000/requisitos.pdf)
- [25] Jedlitschka, Andreas y Outi Salo. **Product-Focused software process improvement**. Primera edición. Alemania: Editorial Springer, 2008. 448 pp.
- [29] Kendall, Keneth y Julie Kendall. **Análisis y diseño de sistemas**. Sexta edición. México: Editorial Pearson Education, 2005. 752 pp.
- [26] López, Carlos. Uch Portal de estudiantes de recursos humanos Administración y gerencia. "ISO 9001". 2008. [Online]. <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/iso9001uch.htm>
- [20] Maurer, Frank y Don Wells, **Extreme Programming and Agile Methods - XP/ Agile Universe 2003**. Primera edición. Alemania: Springer, 2003. 213 pp.
- [27] Mendía Arriola, Héctor Alberto Heber. Metodologías ágiles incorporadas a las necesidades de las empresas que desarrollan software en Guatemala. Trabajo de graduación Ing. Ciencias y Sistemas Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2006. 114 pp.

- [22] MINEDUC, "Guía para generación de documentación administrativa de expedientes". Documento institucional. 2009.
- [23] MINEDUC. "Instructivo de desarrollo de aplicaciones". Documento institucional. 2009.
- [12] Nava y Jiménez. **ISO 9000:2000 Estrategias para Implantar la norma de calidad para mejora continua**. Primera edición. México: Noriega Editores LIMUSA, n.a. 40 pp.
- [11] Pérez Siliezar, Walfred Ibán. Estudio de norma ISO 9000-3 y su aplicación en el desarrollo del sistema: Control de donaciones a la universidad Rafael Landívar. Trabajo de graduación Ing. Informática y Sistemas Guatemala, universidad Rafael Landívar, Facultad de Ingeniería, 2000. 147 pp.
- [24] Shafazand, Hassan y A Min Tjoa. **EurAsia-ICT 2002: Information and communication technology**. Primera edición. Alemania: Editorial Springer, 2002. 1020 pp.
- [2] Solís Ubago, Jean Carlo. La programación extrema como garantía de satisfacción en los sistemas. Trabajo de graduación Ing. Informática y Sistemas Guatemala, universidad Rafael Landívar, Facultad de Ingeniería, 2009. 183 pp.
- [15] Tricker, Ray, Bruce Sherring y otros. **ISO 9001:2000 In Brief**. Segunda edición. Inglaterra: Editorial FISEVIER, 2005. 204 pp.
- [1] Valdés Matta, Alfredo Eduardo. Evaluación de la metodología de desarrollo de software programación extrema. Trabajo de graduación Ing. Ciencias y Sistemas Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2004. 83 pp.
- [21] Wade, Mike. "Yorku University". 2008. [Online]. [http://www.fsc.yorku.ca/york/istheory/wiki/index.php/Stakeholder\\_theory](http://www.fsc.yorku.ca/york/istheory/wiki/index.php/Stakeholder_theory)

- [8] Weitzenfeld, Alfredo. **Ingeniería de software orientada a objetos**. México: Editorial Thomson, 2005. 704 pp.
- [5] Wells, Don. "Extreme Programming". 2009. [Online]. <http://www.extremeprogramming.org/>
- [10] Xitumul Ruiz, Delmi Marilú. Normas ISO 9000 vs CMMI-SW como estándar de calidad en el desarrollo del software y el proceso de obtención de la certificación en cada estándar. Trabajo de graduación Ing. Ciencias y Sistemas Guatemala, universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2007. 106 pp.
- [7] Zannier, Carmen y otros. **Extreme programming and agile methods - XP/ Agile Universe 2004**. Primera edición. Alemania: Editorial Springer, 2004. 225 pp.