



MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍAS DE
INFORMACIÓN

MARCO DE TRABAJO ÁGIL DE DESARROLLO DE SOFTWARE
COMBINANDO SCRUM Y XP. APLICACIÓN A UN CASO DE ESTUDIO.

Trabajo de Titulación presentado en conformidad a los requisitos establecidos
para optar por el título de Magister en Gerencia de Sistemas y Tecnologías de
Información

UNIVERSIDAD DE LAS AMÉRICAS
UNIVERSITY OF THE AMERICAS

Profesor guía

Magister en Ciencias de la Computación e Informática

Autor

Giovany Santiago León Ferigra

Año

2015

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante Giovany Santiago León Ferigra, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Carlos Estalesmit Montenegro Armas
Magister en Ciencias de la Computación e Informática
1704448818

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución las disposiciones legales que protegen los derechos de autor vigentes.”

Giovanly Santiago León Ferigra.

CC: 1713789764

AGRADECIMIENTOS

Agradezco a Dios por la fuerza en momentos duros en los que ya se desfallecía.

A mi familia, por su apoyo incondicional y por el soporte que día a día me han sabido mostrar en las buenas y en las malas.

Al Ingeniero Carlos Montenegro tutor de este trabajo, por su orientación, guía y apoyo constante.

A los profesores y compañeros, por su apoyo y colaboración a lo largo del estudio de esta maestría.

DEDICATORIA

Este trabajo está enteramente dedicado a Dios, mi madre y hermanas.

RESUMEN

Las prácticas ágiles de desarrollo de software son capaces de brindarnos la posibilidad de insertar actividades para la mejora de los productos intermedios. Para lograr este objetivo en la práctica, es necesario complementar actividades de gestión de proyectos con actividades técnicas propias del proceso de desarrollo.

En esta línea de trabajo, el propósito de esta tesis consiste en generar un marco de trabajo ágil híbrido, combinando las metodologías Scrum y Xp mismo que proveerá de las mejores prácticas de desarrollo de proyectos, integradas con las mejores prácticas de desarrollo de software.

El marco de trabajo generado es usado en un caso de aplicación en ambiente web y cloud, evidenciando su factibilidad técnica y operativa.

ABSTRACT

Agile software development standards push towards the improvement of intermediate software products. In order to reach this goal it is useful to complement project management technics with specialized software development processes.

This thesis propose to combine well knows Agile methodologies as Scrum and Extreme Programming (XP) to provide an integrated software framework.

The resulting framework is tested in a real work scenario showing it technical operational feasibility.

ÍNDICE

Introducción	1
Objetivo General	5
Objetivos Específicos	5
Justificación de la Investigación.....	5
Aspectos Metodológicos.....	6
Capítulo 1. Análisis de las metodologías Scrum y XP (Extreme Programming)	7
1.1 XP (Extreme Programming).....	7
1.1.1 Artefactos	8
1.1.1.1 Historias de usuario	8
1.1.1.2 Tarjetas de Tareas	8
1.1.1.3 Tarjetas CRC (Clase – Responsabilidad – Colaborador).....	9
1.1.1.4 Metáforas	10
1.1.2 Roles	11
1.1.2.1 Cliente.....	11
1.1.2.2 Programador	12
1.1.2.3 Encargado de Pruebas (Tester).....	14
1.1.2.4 Encargado de Seguimiento (Tracker)	14
1.1.3 Valores del Extreme Programing.....	15
1.1.4 Procesos de XP.....	15
1.1.5 Prácticas técnicas de Extreme Programming.....	18
1.1.6 Debilidades de la aplicación de XP en la práctica	21
1.2 Scrum.....	22
1.2.1 Roles	24
1.2.1.1 Product Owner	24

1.2.1.2	Scrum Master.....	24
1.2.1.3	Scrum Team.....	24
1.2.2	Actividades	25
1.2.2.1	Product Backlog – Pila del Producto	25
1.2.2.2	SprintBacklog – Pila del Sprint.....	26
1.2.2.3	Sprint.....	30
1.2.2.4	Incrementos del producto.....	30
1.2.2.5	Scrum Diario	30
1.2.2.6	Planificación y Revisión	31
1.2.3	Fortalezas de la aplicación de Scrum.....	33
1.3	Estrategia de Integración.....	34
Capítulo 2. Desarrollo del Marco de trabajo		
combinando las mejores prácticas de Scrum y XP		
		40
2.1	Desarrollo del Marco de Trabajo Ágil	40
2.1.1	Introducción	40
2.1.2	Alcance	40
2.1.3	Descripción General del Marco de Trabajo Ágil “Scrum + XP”	42
2.1.3.1	Roles.....	42
2.1.4	Actividades/Procesos	42
2.1.4.2	Sprint.....	43
2.1.4.3	Scrum diario	45
2.1.4.4	Sprint Review Meeting (Demo)	46
2.1.4.5	Retrospectiva	47
2.1.4.6	Planificación de siguiente iteración/Release Planning Meeting	47
2.1.5	Artefactos	47
2.1.5.1	Product Backlog.....	47

2.1.5.2	Historias de usuario	48
2.1.5.3	SprintBacklog	48
2.1.5.4	Tarjetas CRC (Clase Responsabilidad Colaboración)	49
2.1.5.5	Tarjetas de Tareas (Task Card)	49
2.1.5.6	BurnDown Chart.....	49
2.1.5.7	Burn Up Chart	50
2.1.6	Valores	50
2.1.6.1	Comunicación	50
2.1.6.2	Retroalimentación	51
2.1.6.3	Responsabilidad.....	51
2.1.7	Prácticas Técnicas.....	51
2.1.7.4	Iteraciones.....	51
2.1.7.5	Refactoring.....	52
2.1.7.6	Programación en Parejas.....	52
2.1.7.7	Liberaciones cortas	52
2.1.7.8	Código Estándar	52
2.1.7.9	Integración continua.....	53
2.1.7.10	Propiedad Colectiva	53
2.2	Recomendaciones de aplicación	54
2.2.1	Fase 1 Estudio de factibilidad	54
2.2.2	Fase 2 Definición o ajuste de políticas de Calidad	56
2.2.3	Fase 3 Conformación del equipo	57
2.2.3.1	Integración de los Equipos.....	58
2.2.3.2	Definición del Equipo	59
2.2.3.3	Capacitación	59
2.2.3.4	Velocidad, Trabajo y Tiempo (Burn Up)	60
2.2.3.5	Asignación de Responsabilidades	62

2.2.4	Fase 4 Definición del Proyecto	65
2.2.5	Fase 5 Kick Off	66
2.2.6	Fase 6 Sprint 0	66
2.2.7	Fase 7 Corrida del Modelo	67
2.2.7.1	Paso 1 Sprint Planning.....	67
2.2.7.2	Paso 2 Sprint.....	67
2.2.7.3	Paso 3 Scrum Diario	67
2.2.7.4	Paso 4 Sprint Review Meeting (Demo)	67
2.2.7.5	Paso 5 Retrospectiva	67
2.2.7.6	Paso 6 Planificación de la siguiente iteración	67
2.2.7.7	Paso 7 Lecciones aprendidas por iteración	67
2.2.8	Fase 8 Lecciones aprendidas generales y cierre del proyecto	68
Capítulo 3. Aplicación a un caso de estudio.....		70
3.1	Descripción del caso de estudio y Aplicación del Modelo	70
3.1.1	Fase 1 Aplicación de Encuesta para Estudio de Factibilidad	70
3.1.2	Fase 2 Definición o ajuste de las políticas de Calidad.....	71
3.1.3	Fase 3 Definición del Proyecto	75
3.1.4	Fase 4 Conformación del Equipo.....	75
3.1.5	Fase 5 Kick Off	80
3.1.6	Fase 6 Sprint 0	81
3.1.7	Fase 7 Corrida del Modelo	81
3.1.7.1	Paso 1 Sprint Planning Iteración o Sprint 1.....	81
3.1.7.2	Paso 2 Sprint.....	85
3.1.7.3	Paso 3 Scrum Diario	93
3.1.7.4	Paso 4 Sprint Review Meeting (Demo)	94
3.1.7.5	Paso 5 Retrospectiva	96

3.1.7.6	Paso 6 Planificación de la siguiente iteración	98
3.1.7.7	Paso 7 Lecciones aprendidas por iteración (Sprint 1).....	100
3.1.7.8	Paso 1 Sprint Planning Iteración o Sprint 2.....	103
3.1.7.9	Paso 2 Sprint 2.....	105
3.1.7.10	Paso 3 Scrum Diario Sprint 2.....	106
3.1.7.11	Paso 4 Sprint Review Meeting Sprint 2 (Demo).....	106
3.1.7.12	Paso 5 Retrospectiva Sprint 2.....	106
3.1.7.13	Paso 6 Planificación de la siguiente iteración	107
3.1.7.14	Paso 7 Lecciones Aprendidas Sprint 2	107
3.1.8	Fase 8 Lecciones aprendidas generales, cierre del proyecto.....	110
3.2	Discusión de Resultados	110
4.	Conclusiones y Recomendaciones.....	118
4.1	Conclusiones	118
4.2	Recomendaciones	122
	Referencias.....	125
	Glosario de términos	132
	ANEXOS	135

ÍNDICE DE FIGURAS

Figura 1. Modelo de test driven development (tdd) a ser usado en el marco de trabajo adaptado de: (test driven development (tdd): ventajas y desventajas , 2012).....	17
Figura 2. Definición del cliente y construcción del valor agregado por el cliente.....	18
Figura 3. Ejemplo de product backlog.....	26
Figura 4. Durn chart	27
Figura 5. Ejemplo de tablero de tareas	28
Figura 6. Ejemplo burn up chart	29
Figura 7. Durn up chart hitos.....	61
Figura 8. Durn up chart velocidad y evolución del producto.....	62
Figura 9. Proceso de reuniones del marco de trabajo hibrido	67
Figura 10. Ejecución de test.....	88
Figura 11. Resultado de test semáforo en rojo.....	88
Figura 12. Resultado de test semáforo en verde.....	89
Figura 13. Tablero de tareas proyecto ticket compras	94

ÍNDICE DE TABLAS

Tabla 1. Cuadro de referencia proyectos de éxito en el tiempo	2
Tabla 2. Ejemplo de plantilla de Tarjeta de tareas	9
Tabla 3. Ejemplo de Plantilla de tarjeta CRC	10
Tabla 4. Ejemplo de Plantilla de Metáfora.....	11
Tabla 5. Estimación de horas disponibles	30
Tabla 6. Matriz de cruce Scrum y Xp	37
Tabla 7. Matriz Nuevo Marco de trabajo “Scrum + XP”	41
Tabla 8. Respuestas de Encuesta y Puntajes	55
Tabla 9. Tabla de Cálculo de Respuestas Encuesta.....	55
Tabla 10. Tabla de Descripción de Cargos y Responsabilidades	59
Tabla 11. Matriz RACI	63
Tabla 12. Descripciones de roles y responsabilidades.....	65
Tabla 13. Cargos, Responsables Proyecto Creación Tickets Compras	76
Tabla 14. Matriz RACI aplicada al proyecto	78
Tabla 15. Descripciones de responsabilidades y roles.....	80
Tabla 16. Estimación de horas disponibles de miembros del equipo para Sprint 1	83
Tabla 17. Activación de Historias de usuario en el product Backlog	84
Tabla 18. Planificación de la siguiente iteración.....	99
Tabla 19. Lecciones Aprendidas Sprint 1	101
Tabla 20. Estimación de horas disponibles de miembros del equipo para Sprint 2.....	103
Tabla 21. Priorización y activación de historias de usuario en el Product Backlog Sprint 2.....	104
Tabla 22. Lecciones Aprendidas Sprint 2.....	108

ÍNDICE DE ANEXO

Anexo A. Plantilla de Identificador de Metáfora.....	136
Anexo B. Plantilla de Product Backlog	136
Anexo C. Plantilla de Historia de Usuario.....	137
Anexo D. Plantilla de Tarjeta CRC	137
Anexo E. Plantilla de Task Card.....	138
Anexo F. Encuesta de Factibilidad	139
Anexo G. Plantilla de Lecciones Aprendidas en un Proyecto.....	146
Anexo H. Acta de entrega recepción del proyecto	147
Anexo I. Encuesta aplicada a la empresa XYZ	150
Anexo J. Historias de usuario Caso de Estudio.....	158
Anexo K. Product Backlog aplicado al Caso de Estudio	164
Anexo L. Metáfora aplicado al Caso de Estudio.....	165
Anexo M. Tarjetas CRC aplicado al Caso de Estudio	165
Anexo N. Tarjetas de Tareas aplicados al Caso de Estudio	168
Anexo O. Casos de Pruebas.....	173
Anexo P. Plan de Pruebas de integración.....	175
Anexo Q. Acta de cierre del proyecto aplicado al caso de estudio.....	177

ÍNDICE DE ECUACIÓN

(Ecuación 1)	60
--------------------	----

INTRODUCCIÓN

En la actualidad, los procesos o metodologías tradicionales de desarrollo de software están asociados a un entorno de control de procesos, la mayoría de estos derivados de los conocidos modelos cascada y espiral introducidos por Bohem en el año de 1986 (Zarazaga Soria & Noguerras Iso, 2008, pág. 112). Si bien estos modelos han demostrado ser eficientes en la administración y planificación de tiempos y recursos en muchos proyectos, especialmente en aquellos de gran tamaño, resultan poco adecuados para aquellos proyectos en los que los requerimientos de los usuarios son muy cambiantes y en los que los tiempos de desarrollo deben reducirse de manera extrema y a su vez producir valor agregado y productos de muy alta calidad que es la exigencia del día a día de los clientes.

La realidad actual de los mercados exige altos estándares de calidad en los productos tecnológicos que generen valor, al tiempo que deben cumplir con restricciones relacionadas con el *time to market*. La calidad del software es por tanto un factor importante para el desarrollo de una empresa, por lo que los procesos y métodos que se usen para su desarrollo deben estar controlados y ajustándose todo el tiempo al entorno de sus clientes, además de permitir adecuadas y confiables planificaciones, estimaciones mucho más cercanas a la realidad, seguimientos y controles de los trabajos que se realiza en el orden de TI e indudablemente procesos de testeo de los productos. Es así que el desarrollo de proyectos de software informáticos es una disciplina joven, con menos de 50 años y como tal está en constante búsqueda de perfeccionamiento (Baskerville , 2006)

Según estadísticas de empresas especializadas, tales como Gartner, mencionan que cerca de un 20 % de los proyectos que se idealizan llegan a culminarse con los recursos y en el tiempo estimado. Según informes realizados por Standish Group, en el área de TI, fallan alrededor de un 71% de proyectos de tecnología, los presupuestos se exceden en un 56% y los tiempos

sobrepasan el 84% a lo estimado y las pérdidas económicas son exorbitantes (Anex, 2013).

El informe de CHAOS, que lo realiza la empresa Standish Group, es uno de los más famosos documentos que habla sobre el éxito o fracaso de los proyectos de software en el área de la tecnología, el mismo se realiza cada 2 años, a continuación se muestra una tabla con información de los últimos informes realizados sobre el tema:

Tabla 1. Cuadro de referencia proyectos de éxito en el tiempo

	1994	1996	1998	2000	2002	2004	2006	2008	2010	2012
Éxito	16%	27%	26%	28%	34%	29%	35%	32%	37%	39%
Deficientes	53%	33%	46%	49%	51%	53%	46%	44%	42%	43%
Fracasos	31%	40%	28%	23%	15%	18%	19%	24%	21%	18%

Adaptado de: (The Standish Group International Incorporated, 2013)

El mismo reporte para el año 2012 revela que:

Cerca del 18% de los proyectos de TI son cancelados antes de ser terminados. 43% de los proyectos de TI deben ser ajustados significativamente en cuanto a *tiempo*, presupuesto o alcance.

39% de los proyectos de TI son completados en el tiempo y en el presupuesto acordado.

Si bien es cierto que en los últimos 16 años ha habido avances, aproximadamente el 61% de los proyectos son cancelados o termina sobrepasados en tiempo o dinero (The Standish Group International Incorporated, 2013).

Existen gran cantidad de causas que determinan el fracaso de proyectos de software, tanto por qué no se terminaron o implementaron a tiempo, están fuera de presupuesto, o simplemente porque no cumplen con los requerimientos que

el usuario determinó, se debe entender que los proyectos no terminan mal, los proyectos inician mal.

Según investigadores como Adrián Amex los factores que influyen en el fracaso de los proyectos de software son:

Los Objetivos no están claros.- Se estima que esto es razonable al comienzo de todo proyecto pequeño o mediano, pero a medida que estos proyectos avanzan se deben ir clarificando los objetivos del mismo.

La no identificación de los interesados y el grado de su compromiso.- Se especifica que en esta etapa los interesados que muestran indiferencia al proyecto son los más peligrosos, todas las personas que tengan un grado de participación en el proyecto, tienen responsabilidad por el éxito o fracaso del mismo

Una planificación pobre o ausente.- La adecuada planificación de un proyecto comienza con la misión o propósito del mismo, porque existe este. Cuando se tienen totalmente claros los objetivos y metas del proyecto se pueden identificar los riesgos a los que este puede estar expuesto.

Un control débil del proyecto.- Por lo general esto se da cuando no existe una correcta retroalimentación del avance del proyecto a los patrocinadores y al gerente general, por una inadecuada utilización de herramientas.

Equipo de trabajo poco motivado.- Generalmente se encuentra con equipos desmotivados cuando no existe una clara evidencia de liderazgo.

Inadecuada administración de los riesgos.- un inadecuado manejo de la incertidumbre produce efectos negativos, estos deben ser identificados previamente por el gerente del proyecto.

No existe un procedimiento robusto para la gestión de cambios.- Se debe tener un adecuado manejo de los cambios que se van produciendo a lo largo del desarrollo del proyecto.

Mala comunicación.- Se debe crear un adecuado plan de transmisión y comunicación de la información, con frecuencias bien definidas, así como los destinatarios de esta información y los transmisores de la misma.

No saber decir 'NO'.- Este es el mayor de los problemas, cuando un gerente no sabe decir que no, es totalmente importante que se realicen análisis a fondo de los proyectos en los que se participa y mencionar siempre que cualquier cambio que se solicite va a tener un impacto en el proyecto sobre todo cuando este está ya muy avanzado.

Principalmente para esta encuesta participaron empresas del sector tecnológico ya que en el documento se hace referencia a empresas de Tecnología de información.

Desarrollar software es una actividad compleja, que depende principalmente del trabajo intelectual y creativo de personas, por lo que hay un alto grado de cometer errores en cualquiera de las etapas de creación de estos productos y servicios. (Anex, 2013).

En búsqueda de una mejor y aportar con calidad en los productos de software y mejoramiento en la productividad y eficiencia de la empresa, se propone un marco de trabajo ágil de desarrollo de software combinando metodologías como Scrum y XP, mismas que permitirán proveer de las mejores prácticas de desarrollo como claves fundamentales de éxito, brindando así competitividad a la empresa, ayudándole a obtener un mejor control sobre la variabilidad de las necesidades de sus cliente los que los proyectara como empresas con mayor dinamismo, evolutivas y flexibles.

Objetivos

Objetivo General

- Desarrollar un marco de trabajo ágil de desarrollo de software híbrido (en adelante “Scrum + XP”), combinando las mejores prácticas resultantes del análisis de métodos actuales como son **Scrum** y **eXtreme Programming** y su correspondiente validación aplicando a un caso de estudio.

Objetivos Específicos

- Analizar de forma estratégica los pros y contras que poseen las metodologías ágiles Scrum y Xp y como la combinación de estas, pueden ayudar a mejorar los procesos de desarrollo de productos de software de una empresa así como la calidad de sus productos
- Desarrollar un modelo en base a los artefactos, instrumentos y mejores prácticas de las metodologías mencionadas anteriormente para controlar los procesos de desarrollo y obtener mejor calidad de los mismos en base a características determinadas que estas metodologías otorgan como pruebas y test.
- Validar el modelo desarrollado ágil híbrido, aplicando un caso de estudio en una empresa de Software.

Justificación de la Investigación

Esta investigación es de importancia porque se está generando un aporte tanto para la gestión de los procesos de desarrollo con técnicas de Scrum y un apoyo totalmente técnico de los aportes de artefactos y experiencias de la metodología XP.

Además de ser un aporte metodológico este trabajo permitirá ayudar a resolver problemas de desarrollo de software y gestión de procesos de calidad de productos finales ya que con su combinación, se tiene un mejor control de los procesos y se apalancara en los test para tener productos de alta calidad.

Esto dará a los directores, técnicos y o responsables de los departamentos de desarrollo una mejor organización y dinamismo en sus procesos de trabajo.

Aspectos Metodológicos

El presente Trabajo de Titulación, muestra aspectos como el tipo de investigación, técnicas y procedimientos a ser utilizados para llevar a cabo el mismo, y son las siguientes:

- Recopilación y análisis de la información de las metodologías a ser usadas, casos de estudio y de éxito.
- Diseñar o elaborar el nuevo marco de trabajo ágil basado en la combinación de las mejores prácticas que se complementen entre si ayudando a mejorar el proceso de desarrollo y pruebas de los productos de software.
- Validar el marco de trabajo ágil, mediante su correspondiente aplicación a un caso de estudio.

Cabe mencionar que lo anterior es el contenido del estudio exploratorio y descriptivo del tema en mención.

Capítulo 1. Análisis de las metodologías Scrum y XP (Extreme Programming)

Para proponer un marco de trabajo basado en las mejores prácticas, es necesario conocer de forma independiente las metodologías o modelos que se van a usar, comenzando por las características, funcionamiento y procesos que están involucrados en cada una de ellas.

1.1 XP (Extreme Programming)

Programación Extrema o Extreme Programming por su traducción en inglés, es a diferencia de Scrum una metodología, misma que fue creada por Kent Beck, Ward Cunningham y Ron Jeffries en el año de 1996.

Se diferencia principalmente de Scrum ya que esta acumula un conjunto de prácticas y técnicas que aplicadas de manera simultánea, enfatizan los efectos positivos de un determinado proyecto. (Bahit, 2012, pág. 70).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como principio para el éxito en desarrollo de productos de software, poniendo énfasis en el trabajo en equipo, preocupándose por el aprendizaje de los miembros del equipo, y propiciando un buen clima de trabajo. La característica principal de XP es la realimentación continua entre el cliente y el equipo de trabajo, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. “XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico” (Letelier, Penadés, & Canós, s.f, pág. 3).

“Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP

organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.” (Letelier, Penadés, & Canós, s.f, pág. 3)

1.1.1 Artefactos

1.1.1.1 Historias de usuario

Estas son técnicas que se usan para especificar los requerimientos funcionales del cliente. Generalmente son tarjetas en las que el cliente describe de una forma breve las principales características que desea tanto funcionales como no funcionales, una característica en este tipo de documentos en XP es que son muy dinámicas y cambiantes, pueden romperse, reemplazarse, e inclusive pueden ser cambiadas por otras.

Estos documentos son generalmente muy comprensibles y delimitadas, con el fin de que los miembros de los equipos de desarrollo puedan convertirlos en producto en un tiempo delimitado. (Letelier, Penadés, & Canós, s.f).

A parte de esto, las historias de usuario describen la funcionalidad que debe poseer el software y que su implementación aporte el valor que el cliente necesita en el negocio, estas también describen la prioridad, tamaño y riesgo que presenta en el proyecto.

Existen variedad de plantillas de historias de usuario en la actualidad, sin que exista un formato estándar o normalizado, sin embargo muchos de los casos proponen la utilización genérica de un nombre y una descripción y de ser posible una estimación del esfuerzo (tiempo en días).

1.1.1.2 Tarjetas de Tareas

Representan la principal herramienta para los desarrolladores en cuanto a planeación, ya que permiten determinar cómo deberían funcionar las cosas o los requerimientos. Para el desarrollador representa la serie de pasos que debe

seguir para implementar una historia de usuario. Estas se asocian historias de usuario y están bajo la responsabilidad de cada uno de los programadores.

Además estas pueden ser muy útiles al momento de generar las tareas que van a realizar cada uno de los programadores ya que describen lo que se debe hacer.

Tabla 2. Ejemplo de plantilla de Tarjeta de tareas

Tarea de Ingeniería	
Número Tarea: 1	Historia de Usuario (Nro. y Nombre):12 Generación de Pagos
Nombre Tarea: Implementación de formas de pago	
Tiempo Estimado: 6 horas	
Tipo de Tarea :Desarrollo	Puntos Estimados: 4
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio: 10/02/2013	Fecha Fin:10/02/2013
Programador Responsable: Serrano Wilmer	
Descripción: Los pagos pueden realizarse de 3 distintas maneras. El usuario puede seleccionar de qué manera desea pagar, efectivo, cheque, tarjeta de crédito	

Tomado de: (Plantillas Tarjeta de Tareas, 2014)

1.1.1.3 Tarjetas CRC (Clase – Responsabilidad – Colaborador)

Estas contienen 3 secciones en las que se tiene el nombre de la clase, las responsabilidades y los colaboradores que van a intervenir. Estas tarjetas describen los objetos de la realidad con sus atributos y métodos.

Son usadas principalmente para realizar sesiones de diseño y la simulación de la interconexión del sistema a través de las relaciones que tendrán las clases u objetos, además permite la asignación de las responsabilidades que tendrán estas clases y su colaboración.

Permite llevar a cabo un contrato entre el cliente y el servidor, en el que se especifican sus requerimientos.

En estas tarjetas las clases representan una colección de objetos con similitudes.

En el siguiente ejemplo se puede apreciar cómo se genera una tarjeta Clase - Responsabilidad – Colaborador.

Tabla 3. Ejemplo de Plantilla de tarjeta CRC

Clase: Compra	
Responsabilidad 1 Mantenimiento de entidad Compra	Colaborador 1 Detalle Compra
. Ingreso	Colaborador 2 Proformas
. Actualización	
. Eliminación	Colaborador 3 Definición de nemónico
Responsabilidad 2 Parámetros de Despliegue .Recibir iteración .Relación Cabecera detalle compra	
Responsabilidad 3 Obtención de información de uno o varios tickets de compra .Consultar por Id de Compra .Consultar por parámetros Id, Rango de Fechas, Estados	
Comentarios	
Permite realizar el mantenimiento de una compra, creación, actualización, eliminación, búsqueda, y su asociación con el detalle de la compra, así como la generación de los nemónicos por cada una de las áreas donde se genera el ticket	

Tomado de: (Tarjetas CRC, 2014)

1.1.1.4 Metáforas

Las metáforas vienen a ser representaciones generales del sistema, estas son establecidas al comienzo de cada proyecto permitiendo ayudar a guiar el proceso de desarrollo y tener una visión completa y genérica para todos los miembros del equipo desde los clientes hasta los programadores. Trabaja con un estándar en cuanto a vocabulario que sea fácil de entender por los usuarios

y los clientes. Con esto se puede luego determinar tanto clases como métodos que se pueden usar en el sistema.

Tabla 4. Ejemplo de Plantilla de Metáfora

Identificación de la Metáfora	
Creación de Tickets de Compra	
Metáfora del Sistema	
<p>Se llena un documento físico, en el que se ingresan datos como:</p> <p>Nombre del solicitante fecha de solicitud oficina Área Número de Solicitud</p> <p>Luego de esto en un párrafo más abajo se llena la información del bien o servicio que se desea adquirir se describe la cantidad, la descripción del bien y si amerita una observación que ayude a tener más claro lo que se quiere comprar luego esta solicitud es entregada a la oficina de administración para que sea procesada, quedando el solicitante a la espera de una respuesta Positiva o negativa sobre la adquisición que se solicitó, el administrativo procede a sellar el documento como recibido.</p>	
Información de aprobación de la Metáfora	
Santiago León	María Elena Romero
Firma del Entrenador (Coach)	Firma del cliente

Tomado de: (Tarjetas de metáforas, 2014)

1.1.2 Roles

En xp es fundamental el tener bien definidos los roles, cada rol tiene sus responsabilidades, para así poder cumplir con las metas determinadas para el proyecto, algunos de estos roles varía dependiendo del tipo de proyecto que se ejecute. Sin embargo a continuación se procede a mencionar los roles más relevantes y usados en XP:

1.1.2.1 Cliente

Son quienes se van a encargar de dirigir y conocer los requerimientos y la finalidad que tendrá el proyecto, sabiendo que es lo que se busca del programa, que debe hacer, tiene que estar tan comprometido con el

conocimiento del proyecto, que es quien se encarga de la elaboración de historias de usuario.

1.1.2.2 Programador

Son los miembros del equipo de desarrollo, parte fundamental del proyecto, su principal tarea es la de transformar en código y funcionalidad, las historias de usuario elaboradas en conjunto con el Cliente.

El equipo de desarrolladores por lo general está compuesto por un número de entre 5 y nueve personas. Esto se debe a que la generación de equipos con un número menor a 5, implica caer en riesgos con los miembros del mismo en cambio un número mayor a 9, puede derivar en problemas de comunicación en el equipo y poca colaboración, sin embargo esto no es una norma de Scrum, se ha sabido de proyectos con un número menor a 5 y proyectos con miembros de más de 9 participantes.

De cualquier manera, se puede hacer Scrum con 3 personas y se ha utilizado en proyectos con 250 personas en varios equipos. Cuando es necesario que más de un equipo trabaje de manera ágil en un mismo proyecto, existen diferentes técnicas que permiten esta colaboración, desde el Scrum de Scrums hasta equipos de integración que dedican parte de su tiempo a trabajar con los equipos de desarrollo, siempre completando incrementos de producto de manera regular.

Es un equipo auto organizado, que comparte información y cuyos miembros confían entre ellos. Realiza de manera conjunta las siguientes actividades:

- Seleccionar los requisitos que se compromete a completar en una iteración, de forma que estén preparados para ser entregados al cliente.
- Estimar la complejidad de cada requisito en la lista de requisitos priorizada del producto o proyecto.

- En la reunión de planificación de la iteración decide cómo va a realizar su trabajo:
 - Seleccionar los requisitos que pueden completar en cada iteración, realizando al cliente las preguntas necesarias.
 - Identificar todas las tareas necesarias para completar cada requisito.
 - Estimar el esfuerzo necesario para realizar cada tarea.
 - Cada miembro del equipo se auto asigna a las tareas.
- Durante la iteración, trabajar de manera conjunta para conseguir los objetivos de la iteración. Cada especialista lidera el trabajo en su área y el resto colaboran si es necesario para poder completar un requisito.
- Al finalizar la iteración:
 - Demostrar al cliente los requisitos completados en cada iteración.
 - Hacer una retrospectiva la final de cada iteración para mejorar de forma continua su manera de trabajar.

El equipo es multidisciplinario:

- Los miembros del equipo tienen las habilidades necesarias para poder identificar y ejecutar todas las tareas que permiten proporcionar al cliente los requisitos comprometidos en la iteración.
- Tienen que depender lo mínimo de personas externas al equipo, de manera que el compromiso que adquieren en cada iteración no se ponga en peligro.

- Se crea una sinergia que permite que el resultado sea más rico al nutrirse de las diferentes experiencias, conocimientos y habilidades de todos. Colaboración creativa.

Los miembros del equipo dedicarse al proyecto a tiempo completo para evitar dañar su productividad por cambios de tareas en diferentes proyectos, para evitar interrupciones externas y así poder mantener el compromiso que adquieren en cada iteración.

Todos los miembros del equipo trabajan en la misma localización física, para poder maximizar la comunicación entre ellos mediante conversaciones cara a cara, diagramas en pizarras blancas, etc. De esta manera se minimizan otros canales de comunicación menos eficientes, que hacen que las tareas se transformen en un “pasa pelota” o que hacen perder el tiempo en el establecimiento de la comunicación (como cuando se llama repetidas veces por teléfono cuando la persona no está en su puesto).

El equipo debe ser estable durante el proyecto, sus miembros deben cambiar lo mínimo posible, para poder aprovechar el esfuerzo que les ha costado construir sus relaciones interpersonales, engranarse y establecer su organización del trabajo (Beneficios de Scrum, s.f).

1.1.2.3 Encargado de Pruebas (Tester)

Es la persona que se encarga en conjunto con el cliente escribir las pruebas funcionales, además de ejecutar las mismas de manera regular, retroalimenta al equipo de programadores sobre posibles errores y tiene a su cargo las herramientas de pruebas (Letelier, Penadés, & Canós, s.f).

1.1.2.4 Encargado de Seguimiento (Tracker)

Se encarga de conocer los alcances funcionales del equipo, proporciona retroalimentación al mismo, se encarga también de realizar el seguimiento de

cada una de las iteraciones del proyecto evaluando los objetivos (Pérez A., 2011).

1.1.3 Valores del Extreme Programing.

Extreme programming propone una serie de valores importantes que permiten afrontar los cambios de los requerimientos en los proyectos, estos se enfocan en el equipo de trabajo y son (Pérez A., 2011)

- **Comunicación.-** Procura mantener una comunicación fluida y continua sobre cualquier cambio que se de en el equipo, clientes o jefes.
- **Sencillez.-** Se debe comenzar desde lo más simple que pueda añadir funcionalidad y valor al sistema. En conclusión ver las cosas con el mayor grado de granularidad.
- **Retroalimentación.-** Este valor se pone de manifiesto cuando se realizan pruebas funcionales del software, lo que permite tener información confiable sobre la fiabilidad del mismo.
- **Valentía.-** Esto permite que el equipo de trabajo esté dispuesto a asumir retos y responsabilidades, estar listo ante problemas para así poder afrontarlos (Pérez A., 2011).

1.1.4 Procesos de XP

La metodología Extreme programming presenta una serie de fases, mismas que se basan en los valores mencionados con anterioridad y estas son (Pérez A., 2011):

- **Planeación.-**Comienza con la creación de las historias de usuario, en las que el usuario describe las funcionalidades y características del

producto. En esta instancia el cliente se encarga de priorizar las historias de usuario y asigna un costo a las mismas.

- Diseño.- Los diseños deben mantener un grado de sencillez de forma que faciliten a los desarrolladores su trabajo. Es recomendable la elaboración de glosario de términos.
- Codificación.- En esta fase se procede al diseño por parte de los miembros del equipo de desarrollo de las pruebas de unidad que se aplicaran a cada historia de usuario con el fin de ejercitarlas.
- Pruebas.- Estas deben realizarse dentro de un marco que permita que las mismas se puedan hacer de forma automática, con esto se pueden implementar pruebas de integración y validación que puede correr diariamente, proporcionando indicadores de progreso (Pérez A., 2011).

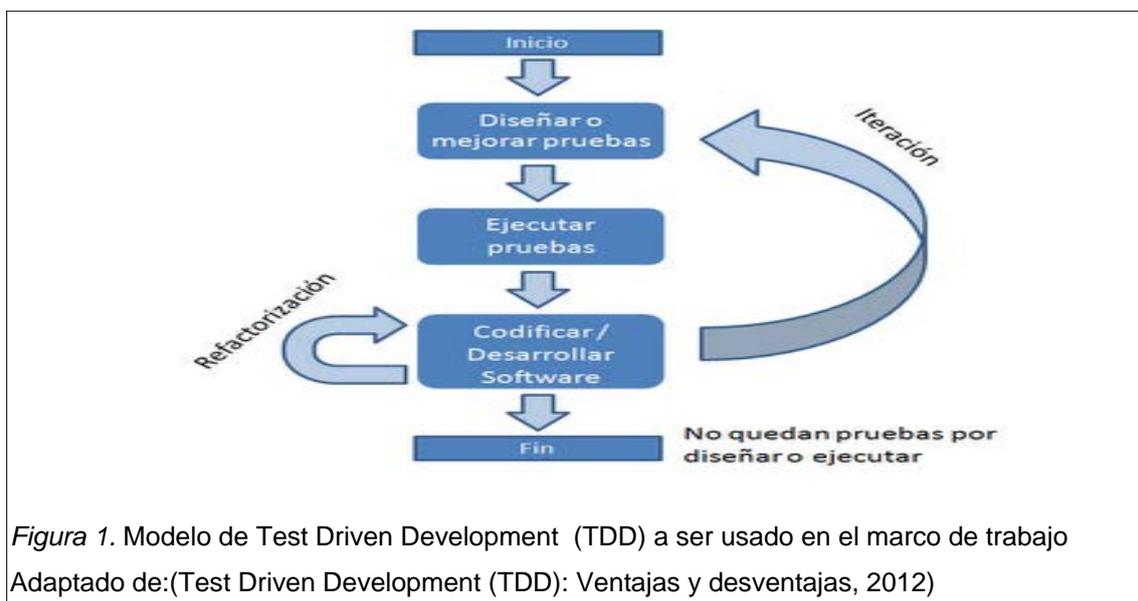
Se encontró que las pruebas en el proceso de XP tienen las siguientes características (Pérez A., 2011), citando a (Sommerville, 2005):

- Desarrollo Previamente aprobado: Esto quiere decir que en primer lugar se deben escribir las pruebas y luego el código que pase la misma. Las pruebas deben verificar que el resultado o salida cumpla con las especificaciones de la data de entrada.
- Desarrollo de pruebas incremental: Los requerimientos en un proyecto son expresados en historias de usuario, es tarea del equipo de desarrollo evaluar cada una de estas para subdividir las mismas en tareas, mismas que tienen características y funcionalidades distintas, por lo que se debe diseñar pruebas de unidad para cada una de estas tareas.
- Participación del usuario en el desarrollo de las pruebas: Al tener al usuario In Situ, característica del XP, este puede ayudar a la realización

de las pruebas de aceptación las cuales deberán ser realizados con data real del cliente, con lo que se verifican las necesidades del cliente.

- Uso de Bancos de datos Automatizados: Se debe usar sistemas que realicen ejecuciones periódicas de las pruebas de una forma automatizada.

Una buena herramienta para el desarrollo de software es el uso de pruebas unitarias de software con base en “TDD Test Driven Development” o Desarrollo conducido por pruebas, esta es una técnica valida tanto para Scrum como para Xp (Kniberg, Sutherland y Cohn, 2007).

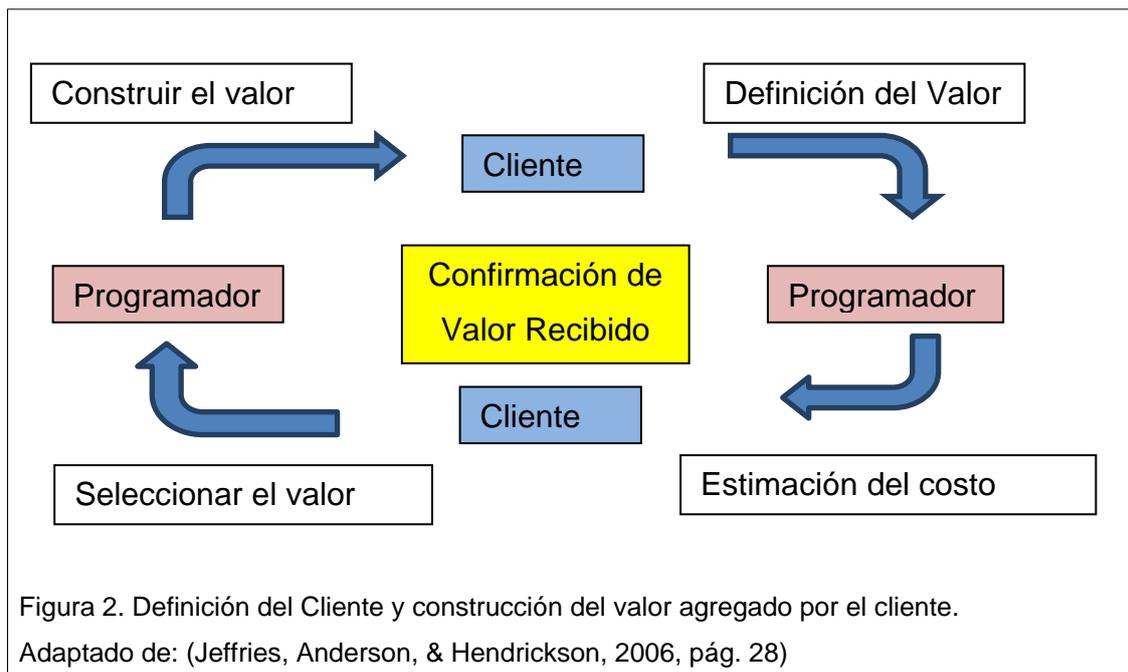


Se debe tener en cuenta que un proyecto manejado con XP llega a tener éxito cuando el cliente final “Selecciona el valor del negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo” (Letelier, Penadés, & Canós, s.f, pág. 10).

En resumen un ciclo de desarrollo con la metodología XP consiste en 5 pasos redundantes:

1. El cliente define el valor del negocio que se requiere implementar

2. El equipo de programadores estima el esfuerzo necesario para su implementación.
3. El cliente selecciona que es lo que se va a construir, priorizando y basado en las restricciones del tiempo que se planifica.
4. El equipo de programadores construye el valor del negocio
5. Se regresa al paso 1 (Letelier, Penadés, & Canós, s.f, pág. 10).



1.1.5 Prácticas técnicas de Extreme Programming

Esta metodología está basada en 12 prácticas, que aplicadas de manera conjunta son capaces de brindar a los proyectos excelentes resultados, a continuación una breve descripción de los mismos (Bahit, 2012, pág. 72):

1. Cliente in Situ.- En esta práctica es importante que el cliente esté dispuesto a participar activamente en el proyecto, su disponibilidad y participación es imprescindible con el equipo de trabajo a lo largo de las fases del proyecto.

En esta práctica la comunicación oral es factor importante en lugar de la comunicación escrita ya que esta última puede ser susceptible de malas interpretaciones (Bahit, 2012, pág. 72).

2. Semana de 40 horas.- En extreme programming se asegura la calidad del trabajo, considerando que el equipo “no debe asumir responsabilidades que le demanden mayor esfuerzo del que humanamente se puede disponer” (Bahit, 2012, pág. 73).

El trabajo extra llega a ser desmotivaste para los equipos de trabajo, si existe trabajo extra en un proyecto, esto quiere decir que existe un problema que debe ser corregido de manera inmediata.

3. Metáfora.- Esta práctica es útil para resolver problemas de comunicación entre el usuario y el equipo de trabajo, lo que busca es hallar una referencia que permita de una manera clara y simple representar un concepto de tipo técnico con una situación de la vida real y cotidiana que sea entendible por el cliente.

4. Diseño simple.- Las soluciones deben ser diseñadas de la manera más simple posible, de tal forma que estas puedan ser implementadas y funcionen en un momento determinado sin mayor complejidad. Todo lo que genere complejidad en el proyecto debe ser removido. Esta práctica se basa en un principio de desarrollo de software conocido como KISS “Keepit simple, stupid” (Mantenlo simple, estúpido). Kent Beck dice, “que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos” (Letelier, Penadés, & Canós, s.f, pág. 13).

5. Refactoring.- Esta práctica consiste en la modificación constante del código fuente sin que esto afecte el comportamiento del software.

Esta es una práctica de reestructuración constante del código con el objetivo de remover código innecesario, código comentado, mejorar la legibilidad del código, de forma que este pueda ser mucho más mantenible y flexible

6. Programación en parejas.- Al igual que la refactorización, la programación en parejas es otra de las prácticas fundamentales del XP. Consiste en que un par de programadores, trabajen en un mismo equipo de cómputo con roles diferentes, apoyándose y aportando ideas en el desarrollo de una tarea.

Se estila generalmente que uno de los miembros escriba el código mientras el otra se encarga de ir revisándolo, también es muy útil cuando se está capacitando a un programador con menor experiencia, de esta forma el programador senior puede ir desarrollando código mientras el desarrollador junior va aprendiendo.

7. Liberaciones cortas.- Esta práctica tiene como objetivo la producción rápida y liberación de versiones del sistema que sean totalmente operativas y funcionales para el cliente, si bien es cierto no son el total del proyecto planteado por el cliente, estas entregas deben contribuir con un resultado de valor para el cliente y su negocio.

8. Pruebas (Testing).- La producción del código como se ha mencionado anteriormente en este documento está dirigida por pruebas unitarias, las mismas que se establecen antes de escribir código. Extreme Programming propone 3 clases de pruebas:

- Test unitarios.-Estos se encargan de probar el código de manera individual, forma parte de la técnica conocida como TDD (Test Driven Development)
- Test de aceptación.- Prueban la funcionalidad del código, su comportamiento, estas son definidas por los clientes y se basan generalmente en casos de uso

- Test de integración.- Estos tiene la responsabilidad de integrar todos los otros test mencionados anteriormente, para así validar el correcto funcionamiento de la aplicación en su totalidad
9. Código Estándar.- No es más que la aplicación de estándares de programación al momento de generar las tareas, lo que hace que el código sea mucho más legible. Esto también ayuda a la comunicación de los programadores ya que el código es limpio
 10. Integración continua.- Esta práctica propone que el código desarrollado por todos los miembros de un equipo de trabajo esté centralizado en un solo repositorio para todos.
 11. Propiedad colectiva.- En base a la anterior práctica esta expone que nadie es dueño de un determinado código, con esto la idea es que todos los miembros del equipo conozcan el desarrollo del sistema, de esta forma todos conocen la funcionalidad, código y se evita que si hay un problema se tenga dependencia de una sola persona para la corrección de esta incidencia.
 12. Juego de la planificación.- Esta práctica permite que haya comunicación entre los programadores y los clientes, donde el equipo de desarrollo realiza estimaciones del esfuerzo requerido para el proyecto y los clientes se encargan de decidir sobre el ámbito y el tiempo de las entregas de cada iteración.

1.1.6 Debilidades de la aplicación de XP en la práctica

De la experiencia adquirida a lo largo del tiempo en el desarrollo de software se ha podido ver en muchas ocasiones con cualquier tipo de metodología que se llega a aplicar existen falencias, no por la metodología en sí, si no por su mala aplicación. Aquí algunos ejemplos de estas debilidades:

- Falta de disciplina derivada de la premura en la entrega de los productos
- No conocer de lleno cual es el alcance real del proyecto
- Retrasos y desviaciones
- Requerimientos mal levantados
- Cambios del negocio
- Desmotivación de los equipos de trabajo
- Alta rotación del personal
- Insatisfacción del cliente final
- Mala imagen de la empresa o departamento de TI

1.2 Scrum

A diferencia de lo que mucha gente cree Scrum no es una metodología, sino más bien es un Framework o marco de trabajo, el cual agrupa un conjunto de buenas prácticas que permiten a un equipo de trabajo mejorar su productividad en un entorno complejo.

Según (Deemer, Benefield, Larman, & Bas, 2009, pág. 5) “Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprints. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija terminan en una fecha específica aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos.”

Es característico de Scrum el realizar reuniones diarias breves en las que se procede a realizar informes de los progresos obtenidos por los miembros de los equipos, así como lo que se pretende realizar al día y si se tiene algún tipo de impedimento. Al terminar un Sprint se procede a realizar una reunión con los

interesados del proyecto para revisar lo que se ha logrado construir, así como también realizar una retroalimentación. En Scrum lo más importante es el hecho de al finalizar cada iteración liberar productos con funcionalidad, probados y potencialmente listos para su uso

En base a esto es importante saber que Scrum permite realizar un trabajo inspeccionado y adaptable, en el que se puede aprender e innovar.

La forma de trabajo de Scrum es realizada por bloques cortos y fijos, mismos que se conocen como iteraciones, generalmente estas vienen a ser de 2 a 4 semanas, en estas el objetivo principal como se ha dicho es el de proporcionar un resultado, una meta también conocido como un incremento sobre el producto final con equipos que van desde los 3 a 7 personas (Schwaber, 2004).

Para esto se procede a partir de una lista de requerimientos misma que debe ser priorizada con antelación por el cliente, este actúa como un plan de proyecto.

Scrum se basa en tres principios fundamentales (Cifuentes Lozano A. , 2012, pág. 28):

- **Transparencia:** Mediante este principio se pretende garantizar que los aspectos del proceso que afecten los resultados, sean visibles para aquellos que están encargados de su administración.
- **Inspección:** En este principio se determina que todos los aspectos del proceso deben ser verificados constantemente, para así encontrar variaciones en el producto.
- **Adaptación:** Si se llega a encontrar mediante la inspección que alguno de los aspectos del proceso está fuera de control, es necesario realizar un ajuste a la proceso para minimizar impactos mayores(Bahit, 2012).

Según (Manrubia Diez, 2009) quien cita a (Schawber, K., Beedle, M., y Sutherland, J., 1990), Scrum debe contar con un mínimo de actividades, roles, responsabilidades y artefactos para su normal desempeño, mismos que se procede a detallar a continuación:

1.2.1 Roles

Existen 3 roles importantes en Scrum:

1.2.1.1 Product Owner

Esta persona es quien se encarga de identificar todas las funcionalidades del producto a ser desarrollado y es quien se encarga de poner estas necesidades en la lista priorizada de funcionalidades, también es responsable de maximizar el retorno de la inversión, asumiendo así la responsabilidad de las pérdidas y ganancias que el producto pueda darle.

El dueño del producto representa los intereses de los interesados en el proyecto, resume los requerimientos del usuario y empuja al equipo en la dirección correcta desde la perspectiva del negocio (Hayata & Han, 2011).

1.2.1.2 Scrum Master

Es el encargado de brindar ayuda a los miembros de los equipos para que logren aprender y aplicar Scrum con el objetivo de darle valor al negocio.

1.2.1.3 Scrum Team

El equipo de trabajo o desarrollo tiene como responsabilidad la construcción del producto que va a ser usado por el cliente, este equipo llega a ser multifuncional y auto organizado. Son los miembros del equipo quienes deciden a que se van a comprometer al momento de comenzar una iteración.

El tamaño de un equipo de Scrum está dado entre 5 y 9 personas. Cuando el número de miembros está por debajo de 5 cualquier imprevisto o interrupción compromete el compromiso que el equipo adquirió con el cliente en la correspondiente iteración.

Cuando el número de miembros es mayor a 9 se pueden presentar problemas en la comunicación y a la colaboración entre los miembros del equipo (Proyectos ágiles, s.f.).

En Scrum los equipos de trabajo se centran en lo que se debe hacer y en cómo los problemas pueden ser resueltos (Hayata & Han, 2011)

Según encuestas realizadas en los años 2003-2005 en proyectos que tenían entre 1 y 20 miembros por equipo, lograron completar cerca de 35 mil a 95 mil líneas de código, se notó que la productividad se daba mejor en aquellos equipos conformados con un máximo de 5 personas, es decir equipos pequeños. La recomendación según Jeff Sutherland el equipo ideal debe estar conformado por un máximo de 9 personas, esto basado en otros proyectos exitosos de Scrum (Mundra, Misra, & Dhawale, 2013)

1.2.2 Actividades

1.2.2.1 Product Backlog – Pila del Producto

Es una lista en la que se debe incluir las necesidades de un proyecto, esta debe ser priorizada, en esta se tiene una visión general del producto, se especifican funciones generales, es decir aquellas que tienen mayor importancia y pueden ser realizadas en un periodo de tiempo fijo. Es usado como un plan del proyecto y una estimación inicial de los requerimientos. El product Back Log evoluciona como el producto y el entorno en el cual será usado es dinámico, realiza una gestión constante para identificar que productos necesitan ser apropiados, competitivos y útiles. (Schwaber, 2004, pág. 22)
Traducido por el autor, texto original en inglés

Product Backlog						Team Velocity	25
Priority	Estimate	Sprint	User Type	Story	Story Type		
1	1	1	Customer	I can see when the next show will begin for the show page I am on	Story		
2	2	1	Editor	I can select what I want to display for each "section" within the editorial content section of the page. My options include last episode, next episode, selected forum posts, selected editorial articles (tv generated), no selection and free form text	Story		
3	2	1	Editor	I can select what picture (if any) I want to display for the corresponding content section	Story		
4	5	1	Editor	I can select the default tab for the user to see upon visit to the page, for each show	Story		
5	5	1	Customer	I can roll over the fields in the media player and see the various tabs change	Story		
6	13	2	Editor	I can modify the existing headline for any show page	Story		
7	1	2	Customer	I can select another show page in the drop down list next to the countdown clock	Story		
8	1	2	Customer	I can click "remote record" and have the show for the show page I am on record on my tivo device	Story		
9	1	2	Customer	I can click "join the discussion" button (or link) on the show page which takes me to the appropriate forum page for that show	Story		
10	1	2	Customer	I can see how many recent posts have been posted in the forum for the show page I am on	Story		
11	3	2	Customer	I can see how many recent replies have been posted in the forum for the show page I am on	Story		
12	5	2	Customer	I can blog about the show for the show page I am on (I need to be signed in to see this)	Story		
13	13	3	Customer	If I am not signed in, I can see a link to sign in	Story		
14	13	3	Customer	If I am logged in, I can click "favorites" and have the show page added to my favorites menu on the site	Story		
15	13	4	Customer	If I have not contributed to the poll, I can see the poll questions and submit to the poll	Epic		
16	20	5	Editor	I can create a new poll for a specific show	Epic		
17	20	6	Editor	I can close an existing poll for a specific show	Epic		

Figura 3. Ejemplo de Product Backlog

Tomado de: (Lacey , 2014)

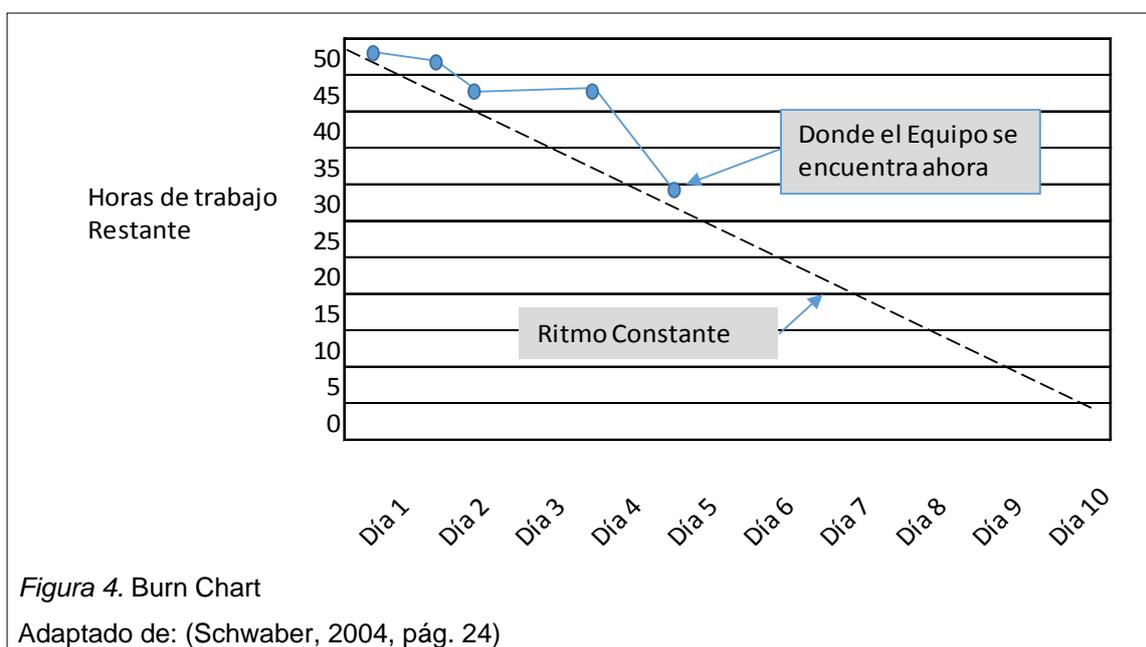
1.2.2.2 SprintBacklog – Pila del Sprint

Es una lista de tareas que elaborada con la participación del equipo de trabajo con el que se va realizar una determinada iteración, forma parte del plan que se determina para completar los requerimientos seleccionados de la iteración y en el que se comprometen con el cliente para su entrega creándose así un incremento en el producto. (Schwaber, 2004, pág. 24) Traducido por el autor, texto original en inglés.

En esta actividad se realizan las siguientes sub actividades(Lista de tareas de la iteración (Sprint Backlog), s.f.):

- Detallar las Tareas.- Es un desglose de la lista de requerimientos priorizados que tienen una duración de entre 4 y 16 horas.

- Determinar responsabilidades: Cada tarea debe tener un responsable que es un miembro del equipo.
- Identificar los riesgos.- Determinar aquellos requerimientos que pueden llegar a tener problemas.
- Figura de trabajo pendiente (Burn Chart).-Este gráfico permite ver la velocidad a la que se está trabajando y completando los requerimientos en un Sprint así como saber el trabajo realizado y lo pendiente por hacer.



- Tablero de tareas – Task Board.- Es un tablero en el que se puede distinguir 3 columnas la primera está determinada por lo que hay que hacer, lo que se está haciendo, y por último lo que está hecho, el propósito es el de tener una idea del estado en el que se encuentran las tareas asociadas a un determinado Sprint. Es de mucha importancia para los equipos, ya que este tablero permite mantener informado a todo el equipo de los avances en las labores, por lo que es importante que cada miembro del equipo reporte sus avances diariamente.

TASKBOARD Stay on target with up to the minute progress of work with virtual cards wall.

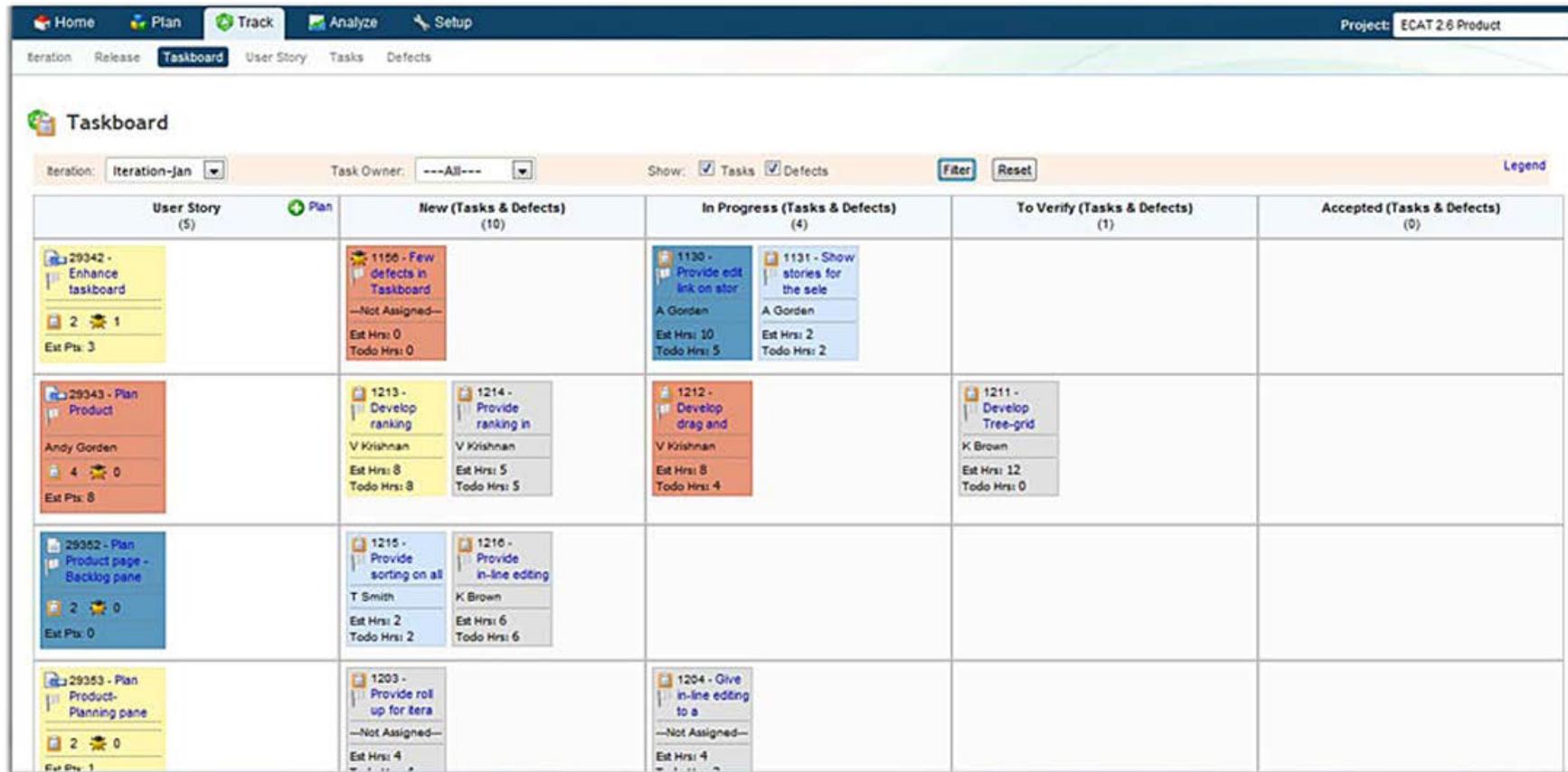
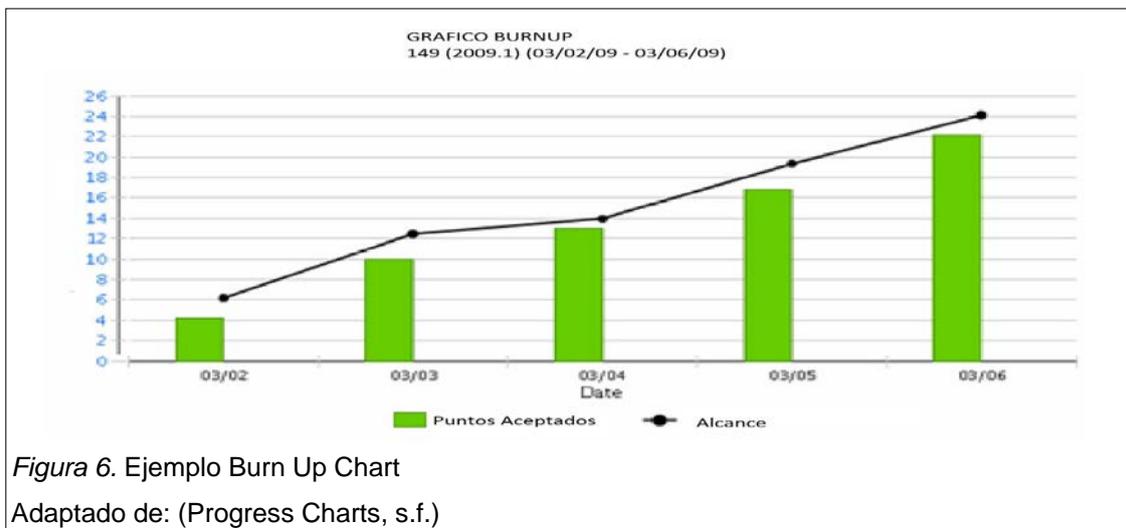


Figura 5. Ejemplo de Tablero de tareas

Tomado de: (Wood, 2010)

- Figura de Producto – Burn Up Es una de las herramientas de planificación que es más usada por el propietario del proyecto. En esta se visualizará el proceso evolutivo del proyecto o producto, ya que muestra en el tiempo como se va construyendo el producto, todo esto en base a la velocidad que mantenga el equipo de trabajo (SCRUMstudy, 2013).



- Estimación del tiempo de cada miembro del equipo. Aquí se determina el tiempo o día laboral que tiene cada uno de los miembros del equipo descontando el tiempo que pasa en reuniones, leyendo los correos, comiendo y otras posibles actividades laborales que no tengan que ver explícitamente con el desarrollo. Generalmente esto significa un tiempo promedio de entre 4 y 6 horas de tiempo disponible al día para el trabajo relacionado con el correspondiente Sprint y se calcula de la siguiente forma (Deemer, Benefield, Larman, & Bas, 2009):

Tabla 5. Estimación de horas disponibles

Longitud del Sprint			2 semanas
Días laborables durante el Sprint			8 días
Miembro del equipo	Días Disponibles durante el Sprint	Horas disponibles por día	Total de horas disponibles
Tracy	8	4	32
Sanjay	7	5	35
Phillip	8	4	32
Jing	6	5	30

Tomado de: (Deemer, Benefield, Larman, & Bas, 2009, pág. 10)

1.2.2.3 Sprint

Es la iteración o ciclo que está dado en el tiempo, puede ser determinado de 2 a 4 semanas, tiempo en el cual el equipo de trabajo convierte la lista de requerimientos en incrementos del producto. Este incremento debe ser funcional.

1.2.2.4 Incrementos del producto

También conocidos como definición de listo, hecho o Done (en inglés). En Scrum es necesario que los equipos construyan o productos en incrementos cuya funcionalidad sea potencialmente entregable, lo que significa que debe estar "Hecho".

1.2.2.5 Scrum Diario

Es una reunión que se realiza aproximadamente por un periodo de 15 minutos diariamente con el objetivo de hacer un resumen de las actualizaciones, y coordinaciones entre los miembros del equipo. En estas reuniones existen 3 preguntas fundamentales que son ¿Qué se hizo ayer?, ¿Qué se va a hacer hoy?, ¿Se tiene algún impedimento? Con lo que cada miembro del equipo está al tanto del avance del proyecto y se pueden sacar a conocimiento si existen impedimentos que interrumpan el normal desarrollo de las actividades.

1.2.2.6 Planificación y Revisión

1.2.2.6.1 Release Planning Meeting (Reunión de Entrega)

Se establece el plan y los objetivos de la entrega, además de decir la estrategia que el equipo Scrum y el resto de la organización van a seguir. Esta reunión se trata de contestar las siguientes preguntas:

- ¿Cómo podemos transformar el objetivo en un producto de calidad de la mejor manera posible?
- ¿Cómo podemos dejar al cliente satisfecho y conseguir un buen retorno de la inversión?

También se define el incremento específico del producto que se entregará, lo cual involucra la identificación y compromiso de la meta a lograr con la entrega, el conjunto de historias que serán desarrolladas y la estimación de cada una de ellas junto con su fecha de entrega. Esta reunión no es obligatoria (Cifuentes Lozano A. , 2012, pág. 34) .

1.2.2.6.2 Planificación del Sprint

“Al comienzo de cada Sprint se hace la Reunión de Planificación del Sprint. Se divide en dos reuniones”. (Deemer, Benefield, Larman, & Bas, 2009, pág. 9).

- Parte uno: ¿Qué vamos a hacer?

La meta de la primera parte de la reunión de planeación del Sprint es adquirir los compromisos frente a las listas de requerimientos priorizadas estimando los esfuerzos. El Propietario del producto presenta lo que le gustaría que el equipo completara en el Sprint. Los miembros del equipo discuten sobre ellos con el Propietario del producto y revisan los criterios de aceptación para asegurarse del común entendimiento de lo que esperan (Cifuentes Lozano A. , 2012, pág. 34) .

- Parte dos ¿Cómo lo Vamos a hacer?

Los equipos comienzan a descomponer la lista de requerimientos seleccionados en tareas, que para entregarlas deben estar completas. Las tareas pueden ser: conseguir una entrada adicional del usuario, diseñar una nueva pantalla, adicionar nuevas columnas a la base de datos, etc. En esta parte se definen las tareas necesarias para complementar cada objetivo o requisito creando la lista de tareas de la iteración (Sprint Backlog) basándose en la definición de Hecho (Cifuentes Lozano A. , 2012, pág. 34) .

1.2.2.6.3 Sprint Review Meeting

Cuando se finaliza un determinado Sprint los equipos muestran el trabajo realizado al dueño del producto, en este caso el dueño del producto puede tomar decisiones sobre la pila de requerimientos (productBacklog).

En esta reunión se realizan las siguientes actividades:

- Presentación del incremento comprometido (Demo)
- Revisión de las tareas que no se cumplieron en el Sprint
- Inclusión y revisión de cambios.

1.2.2.6.4 Retrospectiva

Esta Reunión se realiza al finalizar cada Sprint y una vez realizada el Sprintreview meeting, en esta, el equipo de trabajo trabaja en el análisis de cómo se realizó el trabajo durante la última iteración, se describe lo que se realizó bien, lo que fallo en el mismo y como pueden mejorar para la próxima iteración.

El enfoque de Scrum está dado para construir con calidad, los procedimientos que respetan la calidad son aquellos que son entendidos, mejorados y conocidos por el equipo de trabajo (Palacio, 2014).

Las pruebas en Scrum están dadas principalmente por los criterios de aceptación de, en las que se especifican las pruebas de aceptación que la liberación, debe superar para que se considere como cumplido el requerimiento.

1.2.3 Fortalezas de la aplicación de Scrum

Scrum aporta a la gestión de proyectos de software con las siguientes ventajas:

- Manejo en la gestión de expectativas del cliente final.- En este caso el cliente es quien indica lo que quiere recibir como valor al final de la iteración y comprueba de manera regular si sus expectativas sobre el producto se están o no cumpliendo levantando alertas tempranas.
- Ayuda a tener resultados anticipados de los productos con las entregas de productos en fases cortas.- Con esto el cliente está en capacidad de comenzar a usar su aplicación de manera temprana.
- Mitigación de riesgos.-A partir de la primera iteración los miembros de los equipos de trabajo son capaces de gestionar autónomamente cualquier tipo de riesgo que se presente manteniendo una alta comunicación con el cliente para detectar de forma temprana estos fallos y corregirlos a tiempo.
- Alineamiento entre los equipos de trabajo y el cliente final
- Mantiene al Equipo de trabajo motivado todo el tiempo.- Los miembros del equipo están mucho más motivados cuando sienten que ellos usan sus destrezas para realizar su trabajo y pueden escoger lo que ellos quieren realizar en la iteración así como los tiempos que ellos se plantean para determinada tarea, todo esto siempre en función del cumplimiento de lo comprometido en una iteración.

- Permite tener a los equipos de trabajo conscientes del alcance total del proyecto
- Maneja de mejor forma la flexibilidad y la adaptación.- El cliente es capaz de reestimar las funcionalidades en base a sus propias prioridades.

1.3 Estrategia de Integración

En el mundo del software existen gran cantidad de metodologías para desarrollo tanto tradicionales como ágiles. Según los autores (Jiménez y Orantes , 2012) de la revista académica digital unam.mx, quienes citan a Presman (2005), el término metodología en el área de software se refiere principalmente a un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas computacionales.

Actualmente la combinación de metodologías está marcando tendencia en la ingeniería de software, ya que permite considerar las mejores características de 2 y hasta 3 metodologías a la vez. La tendencia a las metodologías híbridas es atribuida principalmente a Ivar Jacobson quien fue uno de los creadores de UML (Lenguaje Unificado de Modelado) y que en el 2011 creó EssUp (Essential Unified Process) una metodología híbrida en la que se combina RUP y Scrum (Jiménez y Orantes , 2012).

Ejemplos de metodologías híbridas se pueden apreciar en casos de estudio como el realizado por los autores antes mencionados (Jiménez y Orantes , 2012) en el que basados en las experiencias de empresas de desarrollo de software en México acoplaron las mejores prácticas de metodologías tradicionales como RUP y la agilidad de Scrum y XP generando una metodología aplicable principalmente en proyectos de desarrollo de software para aplicaciones web, cuyo tiempo máximo de desarrollo era entre 2 y 6 meses, con equipos de desarrollo conformados por un máximo de 10 integrantes.

De igual forma se han podido revisar la combinación de metodologías ágiles con normas como la ISO 9001, tal es el caso de investigación realizado por (Qasaimeh & Abran, 2010) para ver la capacidad que tiene XP para soportar los requerimientos de la normativa ISO 9001 en los procesos de certificación de software, en este estudio se presenta un análisis de XP para las perspectivas de la normativa ISO 9001 y 9003, enfocándose principalmente en los requerimientos relacionados con el procesos de realización de los productos (desarrollo) y en determinar las debilidades que puede tener XP en el manejo de los mismos.

En algunas ocasiones el uso de metodologías híbridas con modelos ágiles ha sido llevado incluso a casos de estudio de proyectos de Inteligencia artificial como el que realizaron (Mukhtar, Motla, Riaz, & Khand, 2013), en el que explican los beneficios de la aplicación de metodologías ágiles en el desarrollo de aplicaciones de inteligencia artificial.

Del estudio de artículos académicos como “Un Modelo Híbrido para Proyectos de TI con Scrum”, publicado por (Hayata & Han, 2011), se puede deducir que gran cantidad de organizaciones están muy interesadas con la adopción de nuevas metodologías para el desarrollo de productos, especialmente con aquellas que les brinde agilidad. Sin embargo también es cierto que estas organizaciones ya tienen definidos su propio estilo de desarrollo y muchas veces sus culturas de resistencia hacen que los cambios sean duros de realizar.

En estudios realizados para la Industria Pakistaní por (Sultana, Motla, Asghar, Jamal, & Azad, s.f) se puede apreciar que los autores basaron el estudio de este artículo principalmente en la aplicación de XP, Scrum y DSDM (método de desarrollo de sistemas dinámicos) con el fin de obtener un manejo eficiente de la calidad, aseguramiento productividad y mantenimiento utilizando las mejores prácticas de desarrollo con calidad. Ayudando de esta manera a las empresas de Pakistán de desarrollo de software a lograr un estándar global acerca de la industria de software.

Para el caso de estudio de esta tesis en el que se está implementando una metodología híbrida entre Scrum y Xp, si bien es cierto que muchos autores mencionan que la combinación de estas 2 metodologías es totalmente factible, no muchos exponen casos prácticos de trabajo a excepción del conocido libro Scrum y XP desde las Trincheras de (Kniberg, Sutherland y Cohn, 2007), en el que en parte los autores tratan de explicar la aplicación de estas dos metodologías pero desde la óptica del significado de cada una de sus actividades, procesos y prácticas aunque no de una manera didáctica.

Con la finalidad de proporcionar un conjunto de buenas prácticas que permitan a las empresas gestionar de mejor forma sus procesos de desarrollo basados en la combinación de las metodologías mencionadas anteriormente, se plantea realizar un esquema de integración de las metodologías Xp y Scrum, basados en una matriz de cruce, en la que se especificaran las prácticas de cada una de las metodologías, con sus estándares, relaciones, etc. Para esto primero se procederá a seleccionar una de las metodologías como base o marco referencial para la construcción del nuevo modelo.

Puesto que Scrum está mucho más enfocado a las prácticas de gestión y organización, se va a adoptar esta como la base del nuevo marco de trabajo y sobre esta se procederá a consolidar las mejores prácticas de Xp que está más enfocado a las técnicas de programación y pruebas del software (Kniberg, Sutherland y Cohn, 2007).

Otras razones por las que se establece a Scrum como referente o base son las siguientes (Beneficios de Scrum, s.f.):

- Scrum realiza una gestión de las expectativas del cliente de una mejor forma, ya que este último es quien indica que es lo que espera y da el valor que aportará cada requerimiento que prioriza. Además de que el cliente periódicamente puede realizar comprobaciones del cumplimiento de sus requisitos.

- Scrum permite tener resultados anticipados (*"Time to market"*) "El cliente puede empezar a utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo.

Siguiendo la ley de Pareto (el 20% del esfuerzo proporciona el 80% del valor), el cliente puede empezar antes a recuperar su inversión (y/o autofinanciarse) comenzando a utilizar un producto al que sólo le faltan características poco relevantes, puede sacar al mercado un producto antes que su competidor, puede hacer frente a urgencias o nuevas peticiones de clientes, etc."

En base a estas premisas se procede a generar un modelo de integración de las metodologías Scrum y Xp expresadas en las siguientes matrices:

Tabla 6. Matriz de cruce Scrum y Xp

	Scrum	Extreme Programming	Observaciones	
Roles	Product Owner	Cliente	Similares en funcionalidad	
	Scrum máster	Encargado de Seguimiento	Similares en funcionalidad	
	Scrum Team(de 5 a 7 miembros)	Programador	Similares en funcionalidad	
		Encargado de Pruebas	No existe en Scrum, por lo que se adoptará en el nuevo marco	
Actividades / Procesos	Sprint Planning	Planeación	Similares en funcionalidad	
		Análisis	Se complementa con la etapa de sprint planning para realizar el análisis de los requerimientos	
		Diseño	Se adopta la actividad de diseño de XP, complementada con los artefactos de la misma metodología como son metáforas y diseño simple para tener representaciones de la realidad resolviendo problemas de comunicación entre el usuario y el programador.	
	Sprint	Codificación	Similares en funcionalidad	
		Pruebas	Pruebas unitarias	No existe en Scrum, por lo que se adoptará en el nuevo marco
			Pruebas de aceptación	
			Pruebas de integración	
			Uso de Bancos de	

		Datos Automatizados	
	Scrum diario		Adoptados por defecto por el marco base
	Sprint Review Meeting		Adoptados por defecto por el marco base
	Retrospectiva		Adoptados por defecto por el marco base
	re planificación del proyecto Release Planning Meeting		Adoptados por defecto por el marco base
Artefactos	Product BackLog	Historias de Usuario	Adopción del Product BackLog por defecto, pero apoyado en el uso de historias de usuario para tener una mejora en las especificaciones de los clientes y documentos que certifiquen la información levantada
	Sprint BackLog	Tarjetas CRC (Clase Responsabilidad Colaboración)	Adoptados por defecto por el marco base, apoyado en el uso de tarjetas CRC, con la finalidad de tener un grado de mayor granularidad y detalle de las responsabilidades que cada una de las clases tendrá y que cada uno de los miembros del equipo de desarrollo se compromete a realizar
		Task cards	Mejora el nivel de detalle de los requerimientos levantados por los usuarios. No existe en Scrum, por lo que se adoptará en el nuevo marco
	Burn down Chart		Adoptados por defecto por el marco base
	Burn up Chart		Adoptados por defecto por el marco base
Valores	Comunicación	Comunicación	Similares en funcionalidad
	Retroalimentación	Retroalimentación	Similares en funcionalidad
		Sencillez	N/A
	Responsabilidad	Valentía	Similares en funcionalidad
Prácticas Técnicas		Cliente in situ	N/A
	iteraciones de 2 a 4 semanas	Semana de 40 horas	Similares en funcionalidad
		Metáfora	Aplica y será usada en el proceso de Diseño arriba expuesto
		Diseño Simple	Aplica y será usada en el proceso de Diseño arriba expuesto
	Refactoring	Permite ir limpiando el código y mejorando funcionalidades. No existe en scrum, por lo que se adoptará en el nuevo marco	

	Programación en parejas	Permite mejorar la calidad del código, ayuda a aprender. No existe en scrum, por lo que se adoptará en el nuevo marco
Liberaciones cortas	Liberaciones cortas	Similares en funcionalidad
	Pruebas	Permite escribir pruebas automáticas y luego el código que pasa esas pruebas. No existe en scrum, por lo que se adoptará en el nuevo marco
	Código Estándar	Los Estándares ayudan a que el código sea más legible para el resto de los miembros del equipo. No existe en scrum, por lo que se adoptará en el nuevo marco
integración continua	integración continua	Similares en funcionalidad
	Propiedad Colectiva	Crean en los equipo mucha más robustes al compartir y conocer el código de la aplicación. No existe en scrum, por lo que se adoptará en el nuevo marco
	Juego de Planificación	N/A ya se realiza una planificación previa en el sprint planning

Generado por el (Autor)

Capítulo 2. Desarrollo del Marco de trabajo combinando las mejores prácticas de Scrum y XP

2.1 Desarrollo del Marco de Trabajo Ágil

2.1.1 Introducción

En este apartado, se procede al planteamiento de la propuesta de un marco de trabajo ágil para la gestión de desarrollo de proyectos de software, basado en las mejores prácticas de Scrum y cubriendo desventajas o mejorando las existentes con las mejores prácticas de XP, con lo cual se permite tener un modelo más completo y acoplado a las necesidades de desarrollo.

2.1.2 Alcance

El alcance de este modelo está determinado por el uso de las mejores prácticas de Scrum para lograr la gestión de los procesos de desarrollo, apalancados en las mejores prácticas y artefactos de XP para mejorar los procesos técnicos y pruebas del producto que se desarrolle.

De esta manera el nuevo marco de trabajo “Scrum + XP” quedaría de la siguiente manera como se puede apreciar en la siguiente tabla:

Tabla 7. Matriz Nuevo Marco de trabajo "Scrum + XP"

Nuevo Marco de Trabajo			
Roles	Product Owner		
	Scrum Master		
	Scrum Team(de 5 a 7 miembros)		
	Encargado de Pruebas		
Actividades / Procesos	Sprint Planning	Planeación	
		Análisis	
		Diseño	Metáfora
			Diseño Simple
	Sprint	Codificación	
		Pruebas	Pruebas unitarias
			Pruebas de aceptación
			Pruebas de integración
			Uso de Bancos de Datos Automatizados
	Scrum diario		
	Sprint Review Meeting		
	Retrospectiva		
	Replanificación del proyecto Release Planning Meeting		
Artefactos	Product BackLog	Historias de Usuario	
	Sprint BackLog	Tarjetas CRC (Clase Responsabilidad Colaboración)	
		Task cards	
	Burn down Chart		
	Burn up Chart		
Valores	Comunicación		
	Retroalimentación		
	Responsabilidad		
Prácticas Técnicas	iteraciones de 2 a 4 semanas		
	Refactoring		
	Programación en parejas		
	Liberaciones cortas		
	Código Estándar		
	integración continua		
	Propiedad Colectiva		

Generado por el (Autor)

2.1.3 Descripción General del Marco de Trabajo Ágil “Scrum + XP”

2.1.3.1 Roles

2.1.3.1.1 Product Owner

Encargado de la identificación de las funcionalidades del producto y de priorizar las mismas.

2.1.3.1.2 Scrum Master

Brinda ayuda a los equipos de desarrollo para resolver problemas que a estos se les presente en el proceso de desarrollo.

2.1.3.1.3 Scrum Team

Equipo con un mínimo de integrantes de 5 ± 3 , procurando un número par de miembros(Plantilla Pila Producto, 2014).

2.1.3.1.4 Encargado de Pruebas

Se encarga del desarrollo de los escenarios de pruebas en conjunto con el cliente y de la ejecución de las mismas

2.1.4 Actividades/Procesos

2.1.4.1.1 Sprint Planning

Propone que la planificación de las tareas a ser realizadas en una iteración se divida en 2 componentes que son (Sprint Planning, s.f):

1. Una reunión con el cliente o dueño del producto con la finalidad de que esta persona presente al equipo de trabajo la lista de requerimientos

priorizada y se negocie una meta para la iteración. Esto combinado con la Actividad o proceso de planeación de XP.

Luego los miembros del equipo de trabajo examinarán la lista de requerimientos y se realizarán preguntas al cliente, para satisfacer inquietudes que surjan, así como determinar las condiciones de satisfacción que el cliente solicite para los requerimientos. De igual forma en esta etapa se va a priorizar los requerimientos a ser entregados al final de la iteración. Esta parte se complementa con la Actividad de análisis de XP.

2. El equipo de trabajo planifica la iteración, determinando tácticas para conseguir el resultado esperado con la inversión del menor esfuerzo posible, adquiriendo compromisos a través de cada uno de los miembros del equipo.

Se definen las tareas que será necesario desarrollar para completar el requerimiento comprometido en la iteración que se crea en el SprintBacklog, todo esto teniendo en cuenta la definición de listo. En esta definición de tareas se utilizará la práctica de Xp Metáfora para resolver los problemas de comunicación entre el usuario y los miembros del equipo, manteniendo siempre la simplicidad de los procesos ante todo, un ejemplo de tarjetas para el levantamiento de metáforas se puede ver en el Anexo A.

De igual manera se procede a realizar las estimaciones de esfuerzos requeridos para cada tarea y se asignará a cada miembro del equipo las tareas que está en capacidad de realizar.

2.1.4.2 Sprint

Esta es una de las reuniones más críticas. En esta el propósito principal será de dar la información y conocimiento suficiente al equipo de trabajo para que este pueda comenzar a trabajar en la iteración que se plantea, además de ofrecer al dueño del producto la confianza de que tendrá un avance significativo de su producto.

Para lograr esto se procede a plantear lo siguiente:

- Una meta bien definida
En este marco de trabajo la meta de cada Sprint se concentrará en hacer algo que no se haya hecho aún, algo que no se haya logrado de forma que no se pierda el rumbo del camino global que es la terminación y entrega de un proyecto final.
- La lista de miembros del equipo y su compromiso
Para cada uno de los proyectos se procede como ya se mencionó a tener un equipo bien definido de profesionales, con cada equipo se procede a revisar las historias de usuario y tareas que se pretende realizar para finalizar la iteración, se hace que cada uno de ellos decidan en que tarea trabajar y estimen el tiempo que les llevara la misma, con el fin de que cada uno de los miembros tenga en cuenta que se compromete a entregar una funcionalidad que él mismo seleccionó y un compromiso con el resto del equipo.
- La fecha y lugar para la demostración del avance.

Con el compromiso y entendimiento de cada uno de los miembros sobre el trabajo que se pretende realizar, se procede a la codificación de las tareas, es decir al desarrollo.

Al ser usado Scrum como base de este nuevo marco, los equipos de trabajo deben ser auto – organizados de forma que estos puedan tomar decisiones oportunas al momento que se genere incertidumbre sobre algún requerimiento.

El desarrollo de cada una de las tareas debe verse apalancado directamente con las pruebas unitarias de software como base se utilizará “TDD Test Driven Development” o Desarrollo conducido por pruebas, aquí lo que se realiza es un cambio en el orden con el que se desarrollarán las cosas en las iteraciones, es decir primero con la ayuda del Tester se procede a definir las pruebas (casos

de pruebas) y una vez estén listas se procederá a desarrollar código para que la funcionalidad pase estas pruebas.

La forma del procedimiento es la siguiente:

- Tomar un determinado requerimiento o tarea, bien definida.
- Escribir el código para las pruebas que este requerimiento necesite.
- Desarrollar el código o funcionalidad que pase o apruebe el caso de prueba.
- El Tester y el programador verifican el éxito o fracaso del caso de prueba y a continuación se los marca.
- De estas revisiones surgen nuevos casos de pruebas y afinamientos.
- Refactorización del código existente para pasar casos de pruebas adicionales.

De esta manera se puede asegurar que el código que se está generando es lo que se necesita para cubrir el requerimiento, además de que con esto se cumplen con las expectativas y criterios de aceptación del cliente final.

Una vez se tiene una cantidad de funcionalidad adecuada, se procede a realizar pruebas de integración para corrección y refactorización de código y por último con la ayuda del tester se trabajará en las pruebas de aceptación del Cliente (Test Driven Development, s.f).

2.1.4.3 Scrum diario

Se procede a realizar una reunión breve diariamente de preferencia en la mañana de no más de 15 minutos como lo dispone la metodología Scrum, en esta se realiza una sincronización para ponerse al día de los avances que ha tenido cada miembro del equipo, exponiendo lo que se hizo antes de la reunión, lo que se va a hacer en el nuevo día de trabajo y si acaso se tiene algún impedimento para continuar con el trabajo.

Se realiza el análisis del gráfico de Burn-Down, para lo cual todos los miembros del equipo anticipadamente deben haber actualizado sus tareas.

Como buena práctica esta reunión se realiza de pie de preferencia junto al tablero en el cual se encuentren todas las tareas de los miembros del equipo, este tablero puede ser una pizarra en la que se encuentren escritas o puestas las tareas con papeles, de preferencia se sugiere y sobre todo para mantener un orden y una actualización automática del gráfico de avance, que este tablero se lleve en una herramienta, en Internet existen variedad de herramientas gratis que pueden ser de gran ayuda tales como Kunagi, Scrum DoóSprinto Meter (Lista de herramientas Scrum, 2012).

2.1.4.4 Sprint Review Meeting (Demo)

De igual manera apalancada en las prácticas de Scrum, esta reunión tiene una duración máxima de 4 horas y presenta al Product Owner y Stakeholders el avance de la funcionalidad estimada en el inicio del Sprint. Basado principalmente en las definiciones de listo que se hayan estipulado en el inicio del Sprint.

Como buena práctica es importante para realizar la demostración, tener una agenda inicial, misma que deberá detallar lo que se va a revisar en la demostración, nada de presentaciones anteriores y mucho menos funcionalidades que no estén detalladas en la agenda.

En la demostración se puede seguir las siguientes reglas:

- Mostrar
- Ilustrar
- Aplicar
- Y por último dejar una sección para preguntas por parte de los participantes.

2.1.4.5 Retrospectiva

En esta reunión se hace un recuento de todo lo que se realizó en la iteración terminada, analizando por qué se está consiguiendo o no los objetivos planteados.

2.1.4.6 Planificación de siguiente iteración/Release Planning Meeting

En esta reunión se procede a revisar con el cliente y replantear las prioridades del proyecto.

2.1.5 Artefactos

2.1.5.1 Product Backlog

El ProductBacklog representa una lista priorizada de la visión y expectativas del cliente con respecto a los objetivos y entregas del producto.

El cliente será el responsable de crear y gestionar esta lista ayudado por el Facilitador, en la mayoría de las veces el Scrum master, con quien se procederá a realizar las estimaciones de tiempo y esfuerzo para completar los requisitos.

Esta lista Contiene los requisitos del cliente a nivel macro, mismos que suelen ser expresados en forma de Historias de Usuario en las que se indica el valor que cada requisito aportará al cliente (Proyectos ágiles, s.f.).

Las responsabilidades que tiene el Cliente en estas reuniones son las siguientes(Documentar Scrum, 2014):

- Registrar la lista de requerimientos o historias de usuario que sean necesarias para el sistema
- Mantener actualizado la pila de requerimientos del producto

Las responsabilidades del Scrum Master son las siguientes:

- Supervisar la pila de requerimientos
- Mantener reuniones con el Cliente para despejar posibles dudas.

Las responsabilidades del equipo de trabajo son:

- Conocimiento y comprensión de la lista de requerimientos
- Resolución de dudas y aclaraciones sobre cuestionamientos que tenga el cliente así como el correspondiente asesoramiento.

Existen varias herramientas que pueden ser usadas para llevar un registro apropiado de la pila del Sprint, en este caso se propone a registrar las mismas en un formato estándar que permitirá obtener la información necesaria de cada requerimiento, con su prioridad, importancia, estado entre otras cosas. El formato puede ser revisado en el Anexo B de este documento.

2.1.5.2 Historias de usuario

A través de estas se gestiona con el cliente la manera en que se realiza el proceso que se quiere implementar. Este artefacto permite que el cliente se responsabilice y gestione la lista de requerimientos con la ayuda del facilitador o ScrumMaster. En el Anexo C se encuentra el modelo de Historia de Usuario que se va a usar en este esquema de trabajo.

2.1.5.3 SprintBacklog

Documento donde se registra el detalle de los requerimientos a desarrollar en una iteración.

En este artefacto las responsabilidades que tiene el cliente se resumen de la siguiente manera:

- Estar presente en las reuniones en las que el equipo de trabajo realice la Planificación del Sprint para posibles resoluciones de dudas en las historias de usuario.

De igual forma en el SprintBacklog se proponen los siguientes artefactos para ayudar a planificar de mejor forma las iteraciones.

2.1.5.4 Tarjetas CRC (Clase Responsabilidad Colaboración)

Se procede a la creación de una tarjeta CRC por cada una de las clases que se presentan mencionando sus responsabilidades en la parte izquierda de la tarjeta y en la parte derecha la lista de los colaboradores de la clase. Esto ayuda a los miembros del equipo de desarrollo a tener un mayor grado de entendimiento al momento de desarrollar las tareas asignadas.

En el Anexo D se tiene una plantilla de las tarjetas CRC.

2.1.5.5 Tarjetas de Tareas (Task Card)

Usa tarjetas de tareas para determinar las tareas que cada uno de los miembros del equipo deben trabajar, así como para poder tener un control de que es lo que cada quien debe hacer, estas tarjetas se representan mediante una forma gráfica, misma que sirve también en las reuniones diarias en el tablero de tareas.

Más detalle se puede encontrar en el Anexo E (ejemplo de las tarjetas de tarea que se usará).

2.1.5.6 BurnDown Chart

Este gráfico es usado en las iteraciones para determinar la velocidad de trabajo del equipo, así como para conocer si existen retrasos en los proyectos, además

es usado en las reuniones con el cliente para explicaciones de avances en los trabajos.

Es responsabilidad del ScrumMaster para este artefacto:

- La constante supervisión de los avances o retrasos en el mismo.

El Equipo de trabajo es responsable de:

- La actualización diaria de las tareas para que se vea reflejado en gráfico el estado a la fecha del desarrollo de la iteración.

2.1.5.7 Burn Up Chart

Este Gráfico representa el plan del producto, es decir permite que el Dueño del Producto vea la evolución de su requerimiento.

Este gráfico es responsabilidad del Dueño del Producto, tanto en su confección como en su mantenimiento.

Tanto el equipo de trabajo como el ScrumMaster deben tener conocimiento de esta gráfica, para poder solicitar aclaraciones sobre dudas que puedan presentarse durante el desarrollo del mismo.

2.1.6 Valores

Los valores que deben ser practicados en los equipos de trabajo que van a formar parte de cualquier proyecto de desarrollo en el que intervenga este marco de trabajo ágil se proponen como:

2.1.6.1 Comunicación

Este valor es aplicable desde la creación de la lista priorizada de requisitos que se realiza con el cliente, así como en las reuniones de re planificación y en las estimaciones de esfuerzos en el trabajo.

Todo esto mediante:

- Reuniones de planificación
- Reuniones Diarias
- Retrospectivas

Como recomendación especial mantener a los equipos juntos en un mismo espacio físico para que se maximice la comunicación frente a frente.

2.1.6.2 Retroalimentación

Esta base de la metodología para mantener un feedback tanto con el cliente como con los equipos de trabajo involucrados, esto tiene enorme relación con el valor de comunicación.

2.1.6.3 Responsabilidad

Con este valor se pretende empoderar a cada miembro del equipo de su trabajo y terminar las tareas de forma adecuada con la calidad así como en el tiempo estimado por el mismo miembro.

2.1.7 Prácticas Técnicas

2.1.7.4 Iteraciones

Se toma esta práctica de Scrum con un tiempo de 2 semanas por Sprint para casos normales en los que las historias de usuario no presenten una complicación mayor y de 4 semanas en las que se requiera un mayor esfuerzo, luego de la correspondiente revisión y análisis.

2.1.7.5 Refactoring

Esta práctica permite tener un código mucho más limpio y optimizado ya que se remueve código innecesario, comentado y permite mejorar la legibilidad del mismo para otros programadores.

Una buena práctica para la refactorización es hacer una analogía con el béisbol, en el que dicen: tres strikes y Fuera, pues bien en programación puede traducirse como 3 strikes y re factoriza. Como ejemplo a esto podemos decir (Bahit, 2012)

“La primera vez que hagas algo, solo hazlo, la segunda vez que hagas algo similar, notarás que estás duplicando código, pero lo harás de todas formas. La tercera vez que te enfrentes al mismo caso, re factoriza”(Bahit, 2012, pág. 137).

2.1.7.6 Programación en Parejas

Con esta práctica se pretende la colaboración continua y el aprendizaje de los miembros del equipo que se integran en etapas posteriores del proyecto, así como colaboradores en la calidad de los productos.

2.1.7.7 Liberaciones cortas

De esta forma se pretende que el cliente pueda ver y tener versiones listas para trabajar de los productos que él requiera.

2.1.7.8 Código Estándar

Se utilizara los estándares siguientes:

- Nombres de variables mnemotécnicos con el fin de saber el tipo de cada variable solo con ver el mismo.

Por ejemplo:

- strUsuario (str = variable de tipo string + Nombre Descriptivo de la variable)
- intIdUsuario (int = variable de tipo entero + Nombre Descriptivo de la variable)
- Nombres de Variables Sugestivos, para saber de forma natural el uso y finalidad de las mismas

Por ejemplo:

- DecCalcular Saldo Cuenta (variable de tipo decimal que obtendrá el cálculo del saldo cuenta)
- Se usará el estilo de escritura “camelcase” para la codificación.

Por ejemplo:

- DecCalcular Saldo Cuenta (variable en la que la escritura de las primeras letras de cada palabra permite legibilidad de las mismas)
(Estilo de programación, 2008)

2.1.7.9 Integración continua

Para esta práctica se utilizará una herramienta tecnológica además de un servidor de repositorio.

Una herramienta útil para realizar esta práctica es el Team Foundation Server, permite tener control del código fuente, compilaciones automáticas, informes de gestión de tareas, así como la integración continua del código (Microsoft, 2015).

2.1.7.10 Propiedad Colectiva

De esta forma todos pueden conocer lo que se está realizando así como obtener ayuda de cosas que otra persona haya creado y que sirva como referencia.

Aquí es aconsejable tomar como buenas prácticas las siguientes heredadas de XP (Becerra Castro, s.f):

- En XP todos tienen responsabilidad sobre todo el código del sistema.
- El código creado es propiedad del equipo completo, no de uno de los desarrolladores específicos.
- Es necesario que los desarrolladores escriban código dentro de una arquitectura, estilo y propósito común que es la alta calidad.
- Promueve a que el equipo entero trabaje mucho más junto buscando producir diseños, códigos y casos de test de alta calidad.

2.2 Recomendaciones de aplicación

A partir de los elementos y de la información que se ha logrado obtener a lo largo de este documento y del cruce de las mejores prácticas de Scrum como marco metodológico base y XP como complemento del nuevo marco de trabajo se procede a realizar las siguientes recomendaciones:

2.2.1 Fase 1 Estudio de factibilidad

Para la implementación de este marco de trabajo híbrido de desarrollo de software, es importante hacer un análisis de factibilidad de su uso y aplicación, para lo cual primero se procede a hacer una encuesta, misma que permitirá obtener información básica de lo que se requiere en una empresa, en un equipo de trabajo para proceder a aplicar un nuevo marco de desarrollo ágil, esta encuesta ha sido basada en la encontrada en el blog Agile Iberoamérica (Buhler, 2013) y complementadas con algunas preguntas propias y otras adaptadas a los modelos de evaluación de situación de las empresas respecto al sistema de gestión según ISO 9000:2000 y también referentes al Manifiesto Ágil (Manifiesto Ágil, 2001) para conocer qué tan preparada esta una empresa para aplicar una metodología ágil.

Para evaluar el estado de adaptación de una empresa a una metodología ágil se ha procedido de la siguiente manera:

Se realizan preguntas referentes a diversos procesos como manejo de personal, métricas, trabajo con pruebas, jerarquías y estructuras de la empresa, se proponen con respuestas del tipo cerrado, bajo la siguiente tabla de puntajes:

Tabla 8. Respuestas de Encuesta y Puntajes

Respuesta	Valor
Prácticamente no se realiza	1
Se realiza parcialmente (en ocasiones puntuales)	2
Se realiza generalmente (en la mayoría de los casos)	3
Se realiza sistemáticamente y en casi todas las áreas	4
Se realiza siempre y de forma total, y somos un ejemplo para el sector.	5

Adaptado de: (Buhler, 2013) y (Gestión de Calidad ISO 9000:2000, s.f.)

La valoración se realizara contra la siguiente tabla:

Tabla 9. Tabla de Cálculo de Respuestas Encuesta

	1	2	3	4	5
Total Marcados					
Totales multiplicados por puntos	*1	*2	*3	*4	*5
De					

Adaptado de: (Gestión de Calidad ISO 9000:2000, s.f.)

Para la obtención de los totales, se procederá al conteo de las respuestas marcadas del mismo valor, se multiplicará por los puntos indicados para la pregunta.

Valoración de los resultados

Menos de 40 puntos: El sistema de gestión de desarrollo no se cumple o se cumple en aspectos parciales. Deben tomarse medidas correctoras urgentes para implantar un sistema de gestión de desarrollo eficaz tradicional o ágil.

Entre 40 y 60 puntos: El sistema de gestión de desarrollo se cumple, pero con deficiencias.

Entre 60 y 85 puntos: El sistema de gestión de desarrollo se cumple, con leves deficiencias.

Más de 85 puntos: Su empresa gestiona de manera correcta en cuanto a políticas de calidad y gestión de desarrollo de productos de software, es apta para aplicar metodologías de desarrollo ágil.

Se recomienda la aplicación de esta metodología a partir de aquellas empresas con puntuación sobre los 70 puntos, ya que es posible mediante disciplina lograr una adaptabilidad y mejora en el proceso de gestión de desarrollo.

La encuesta mencionada se puede ver en el Anexo F de este documento.

2.2.2 Fase 2 Definición o ajuste de políticas de Calidad

Es importante la definición de las políticas de gestión de calidad, mismas que son estipuladas por la empresa en la que se aplica la propuesta metodológica, estas van dirigidas principalmente por los altos directivos de la empresa, y vienen a ser una clara intención de dirigir a la Empresa en un camino de mejora continua, enfocándose principalmente en lo que a satisfacción del cliente se refiere en sus productos y servicios de software. Para lo cual se comprometen en la difusión de las mismas y brindar los recursos necesarios para su cumplimiento, apalancadas siempre en las políticas generales de la empresa.

Estas políticas deben ser institucionalizadas en la empresa a fin de que todos los miembros de la misma conozcan de qué se trata y estén comprometidos a trabajar para alcanzar los objetivos de calidad que se establezcan.

Cabe destacar que la determinación adecuada de las políticas de gestión de calidad viene a convertirse en un factor crítico de éxito de la empresa en cuanto al manejo de calidad por lo que es determinante su definición para continuar con el esquema de trabajo, sin políticas no se puede continuar con un proceso

de gestión de calidad. En esta fase se procede a trabajar en la definición de políticas en caso de no existir o a su correspondiente ajuste con las metodologías ágiles si estas ya están definidas.

2.2.3 Fase 3 Conformación del equipo

Es importante para un equipo de desarrollo ágil ser predecible en sus acciones, porque esto permite tener un alto grado de confianza ante el cliente final, además de un excelente ambiente de trabajo, motivado y liviano para cada uno de los miembros del equipo. Para el cliente esto es importante puesto que el equipo puede adaptarse a nuevos procesos, así como vincularse rápidamente a cambios inesperados (Técnicas de integración , s.f.).

Por otra parte también es importante entender a qué se refiere la predictibilidad. Principalmente se habla de que los miembros del equipo aprendan de sus errores y sean parte de un ciclo de mejora continua.

Los equipos de desarrollo en metodologías tradicionales poseen estructuras de mando o jerárquicas bien definidas en las que los roles están dedicados a realizar su trabajo o tarea especializada sin ir más allá, lo cual da como resultado una interacción muy restringida entre los miembros de un mismo equipo, no se diga con otros miembros de diferentes equipos.

Por esta razón los equipos de trabajo que se pretende tener en esta nueva metodología son equipos auto-organizados, con un alto nivel de alineamiento y empoderamiento en cuanto al proyecto en el que se trabaja, con lo que se logra tener alta flexibilidad, disciplina y responsabilidad por parte de cada miembro. Estos equipos ágiles son capaces de participar y tomar decisiones en igualdad de condiciones empoderándose del proyecto, si bien existe al igual que en otras metodologías una jerarquía se tratará de que todos se sientan dueños del proyecto.

Para trabajar estos proyectos con este nuevo marco de desarrollo ágil contaremos entonces con un equipo de trabajo conformado de la siguiente forma:

- Product Owner.- Dueño del proyecto
- Scrum Master.- 1 por cada proyecto que se esté implementando.
- Scrum Team.- Conformado como se ha mencionado. anteriormente por un número de integrantes de 5 ± 3 con el fin de tener equipos pares para así poder trabajar tareas en pares(Requisitos para utilizar Scrum, s.f).
- Encargado de Pruebas.- de 2 a 3 personas las cuales estarán dedicadas estrictamente a realizar las pruebas con el cliente.

El equipo para el nuevo marco de trabajo ágil estará conformado por miembros que se motiven y tengan un real interés por trabajar dentro de un grupo de profesionales que tienen en mente ayudarse mutuamente, mismos que deberán tener múltiples habilidades técnicas tanto de programación como habilidades para comunicarse con el resto de sus compañeros como con los mismos usuarios.

2.2.3.1 Integración de los Equipos

La integración de los equipos de trabajo es esencial para el buen funcionamiento de toda la organización. Un grupo de trabajo bien integrado funciona como una unidad en pro de la meta en común, por lo que es tarea impostergable de la empresa realizar actividades, jornadas y prácticas que fomenten el trabajo grupal, la coparticipación y la solidaridad entre los distintos sectores y miembros de la organización (Técnicas de integración , s.f.).

Las tareas de integración de los equipos se realizaran en jornadas cortas dentro de la organización en base a capacitaciones y siempre enfocados en la meta que se pretende buscar en cada proyecto (Técnicas de integración , s.f.).

La idea principal de estas integraciones antes de comenzar un nuevo proyecto es la de hacer que los miembros del equipo se fusionen para alcanzar los objetivos del proyecto (Beneficios de Scrum, s.f).

De igual manera esta integración permitirá que se comunique por parte del líder en que se va a trabajar, además de dar confianza y unión al equipo (Henrik & Reza, 2008).

2.2.3.2 Definición del Equipo

El equipo estará conformado de la siguiente forma:

Tabla 10. Tabla de Descripción de Cargos y Responsabilidades

Cargo	Descripción	Responsable
Product Owner	Encargado de Realizar y priorizar la lista de requerimientos	
Scrum Master	Encargado de la planificación, ejecución y control del proyecto	
Analista Negocio	Analista de los requerimientos de los usuarios y programación	
Programador Senior	Desarrollo y arquitectura de la aplicación	
Programador Junior	Desarrollo de aplicación	
Tester	Realizar pruebas funcionales de la aplicación y de aceptación	

Generado por el (Autor)

2.2.3.3 Capacitación

Aduciendo que se está realizando la implementación de un proyecto totalmente nuevo se procederá a realizar una capacitación total, es decir:

- Introducción breve a metodologías ágiles
- Resumen del Manifiesto Ágil
- Resumen de Metodología Scrum
- Resumen de Metodología XP
- Introducción al nuevo marco de trabajo ágil

- Roles
 - Actividades
 - Artefactos
 - Valores
 - Prácticas y Técnicas
- Técnicas de Estimación

Dependiendo del proyecto si se requiere algún tipo de capacitación especial en herramientas de desarrollo o gestión de proyectos se procederá a realizar las capacitaciones previa confirmación.

2.2.3.4 Velocidad, Trabajo y Tiempo (Burn Up)

Estas tres variables son importantes para medir la gestión de proyectos del tipo ágil (Scrum Manager BoK, 2014).

Para esto se aplica la fórmula de la velocidad:

$$\text{Velocidad} = \frac{\text{Trabajo}}{\text{tiempo}} \quad \text{(Ecuación 1)}$$

De esta manera se puede indicar por ejemplo que si se tiene un equipo de desarrollo de 4 miembros su velocidad de desarrollo es de 40 puntos por semana o de 80 puntos por Sprint (SCRUMstudy, 2013).

Con la velocidad se puede determinar cuál va a ser la forma en que evolucionará el proyecto mediante el gráfico Burn up, como se detalla a continuación:

El gráfico de Producto o Burn Up proyecta sobre un diagrama cartesiano, en el cual sobre el eje de las ordenanzas se va a colocar el esfuerzo estimado para construir las historias de usuarios que se obtengan del product Backlog, mientras que en el eje de las abscisas va el tiempo en Sprints (Scrum Manager BoK, 2013).

En este plano se procede a trazar la línea que representa el avance previsto según la velocidad media que se ha obtenido para el equipo, pero además es importante también trazar los ritmos de avances con una previsión pesimista basada generalmente en información de proyectos anteriores en los que haya habido problemas o retrasos, y de igual forma una velocidad optimista obtenida de igual forma de los proyectos anteriores pero esta vez con mejor resultado.

Algunos profesionales prefieren hacer esta estimación con el uso de un porcentaje de margen de error $\pm 10\%$... 20% ... etc.

Con esta información se procede a tomar del Product Backlog las posibles versiones a liberar, por ejemplo si se quiere entregar una primera versión luego de 4 Sprints se debe calcular el esfuerzo requerido para cada uno de esos Sprints, sumarlos y ese ser el primer hito y así con el resto de entregas. Para revisar esto en el gráfico se procede a situar cada hito en la gráfica en el eje vertical en la posición que corresponde al esfuerzo estimado.

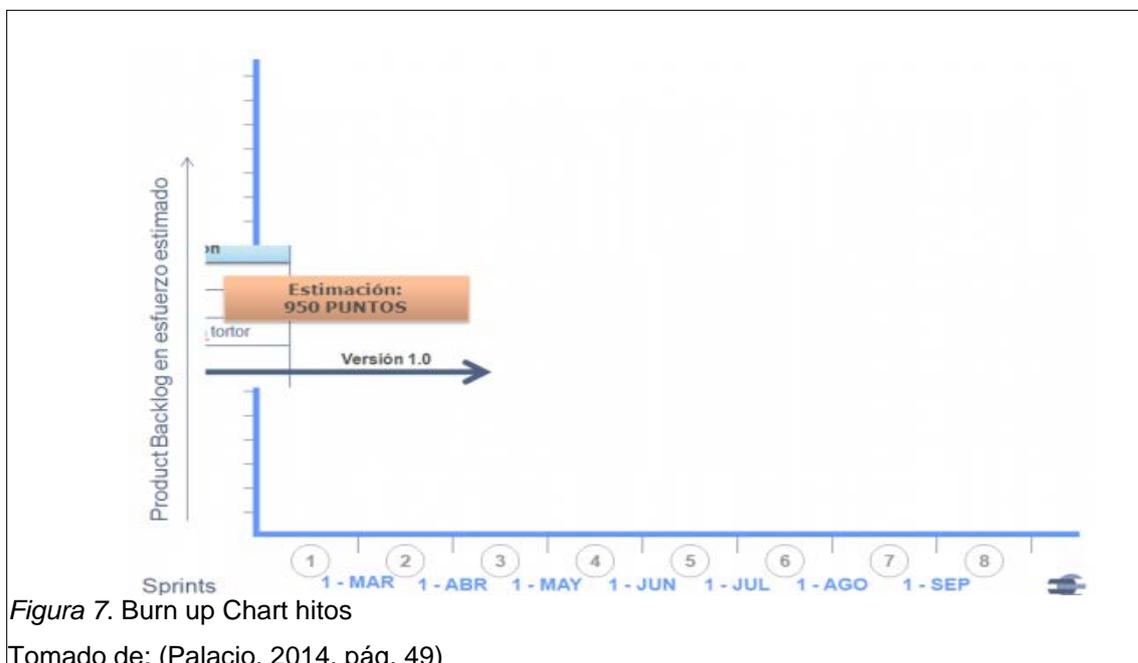


Figura 7. Burn up Chart hitos

Tomado de: (Palacio, 2014, pág. 49)

Con los puntos de corte que generan cada una de estos hitos con las líneas de velocidad del equipo, incluyendo las optimistas, pesimistas y realista, proyectan en el eje horizontal las posibles fechas o Sprints en el que se completará el proyecto como se puede apreciar en el ejemplo siguiente (SCRUMstudy, 2013).

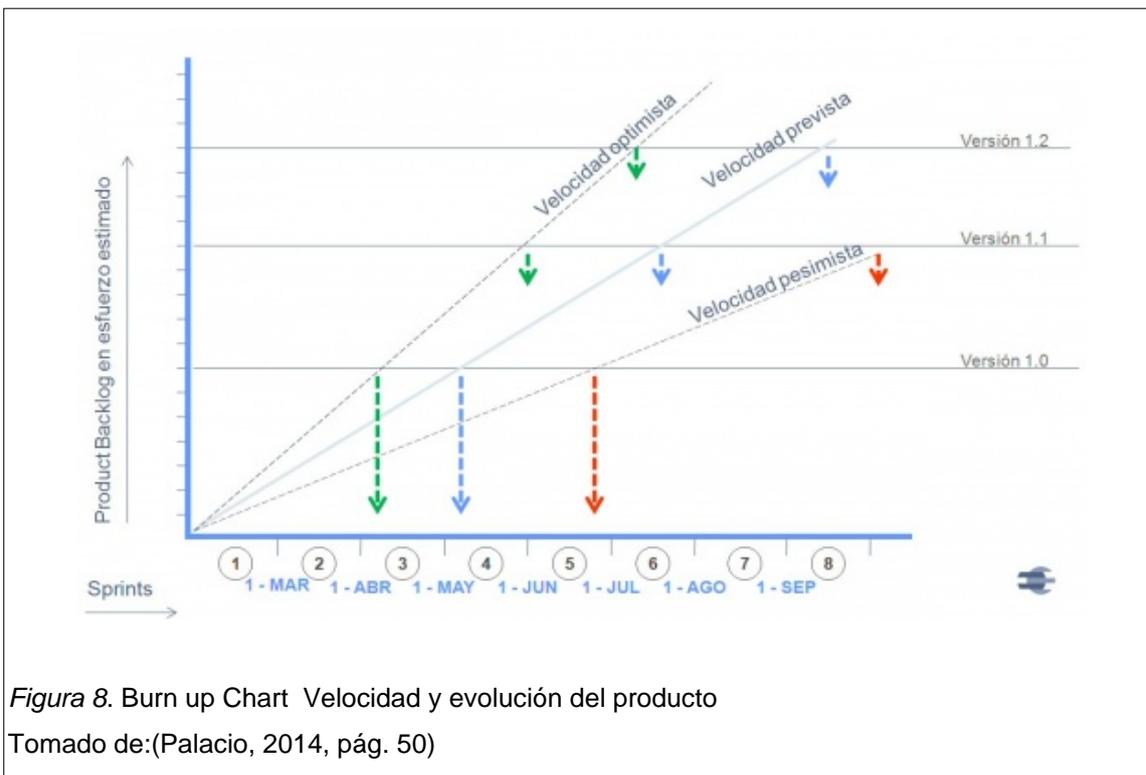


Figura 8. Burn up Chart Velocidad y evolución del producto

Tomado de:(Palacio, 2014, pág. 50)

2.2.3.5 Asignación de Responsabilidades

En la Matriz de Asignación de Responsabilidades se determina las tareas generales que se asignan a cada integrante en el proyecto(Plantilla Pila Producto, 2014).

Tabla 11. Matriz RACI

Elaborado por:									
Informado.									
Actividad		Roles/Responsabilidades							
ID	Actividad	Product Owner	Scrum Masters	Stakeholders	Analista de Negocio	Programador Senior	Programador 1	Programador 2	Programador n
1	Product Backlog	A, R	I	C, I	I	I	I		
2	Sprint Backlog	C	C, I	C, I	C, I	C, I	C, I		
3	Burndown charts	C, I	I	A, R	C, I	C, I	C, I	C, I	
4	Sprint Planning	C, I	A	C, I	C, I	C, I	C, I	C, I	
5	Sprint Review	C, I	A	R	C, I	C, I	C, I	C, I	
6	Sprint retrospective	C, I	C	C	C	A, R	C, I	C, I	
7	Daily Scrum Meeting	C, I	C, I	A, R	C, I	C, I	C, I	C, I	
8	Sprint	I	C	A, R	I	I	I	I	
9	SPRINT PLANNING	C, I	C	A, R	I	C, I	C, I	C, I	

10	Analizar y evaluar el Product Backlog	C, I	I	A, R	C, I						
11	Seleccionar el Objetivo del Sprint	C	C, I	C, I	A, R	C, I					
12	Definir las Historias de Usuario	C	C, I	C, I	A, R	C, I					
13	Definir las tareas	C	C, I	C, I	A, R	C, I					
14	Estimar la duración de las Tareas	C	C, I	C, I	C, I	A, R	C, I	C, I	C, I	C, I	
15	Investigación	C, I	I	I	A, I	R	C, I	C, I	C, I	C, I	
16	Desarrollo	C, I	I	I	A, I	R	C, I	C, I	C, I	C, I	
17	Verificación y Corrección de Bugs	C, I	I	I	A, I	R	C, I	C, I	C, I	C, I	
18	Pruebas Unitarias	C, I	I	I	A, I	R	C, I	C, I	C, I	C, I	
Tomado de: (Asignación Raci, 2014)											

Tabla 12. Descripciones de roles y responsabilidades

Roles y Responsabilidades	
Rol / Responsabilidad	Descripción
R	Responsable: Este rol es el que realiza (ejecuta) el trabajo asociado con la actividad, lo habitual es que cada actividad tenga una solo "R", si existe más de uno es recomendable subdividir la actividad.
A	Aprobador: Es el encargado de aprobar (firmar), el trabajo realizado, a partir de esa aprobación, este se vuelve responsable por la actividad. Como regla general debe existir una solo "A" por actividad. Este rol es quien asegura que se ejecutan las tareas, por ejemplo Líderes de área técnica, área de gestión de proyecto, entre otros.
C	Consultado: Posee alguna información o capacidad que se necesita para mantener el trabajo. Se le informa y consulta información, de manera bidireccional con el responsable y/o aprobador.
I	Informado: Rol que debe ser informado sobre el progreso y los resultados del trabajo. En este caso la comunicación es unidireccional (se le da información pero no se recibe información).

Tomado de: (Asignación Raci, 2014)

2.2.4 Fase 4 Definición del Proyecto

En esta fase se tiene un primer acercamiento para saber cuál es el objetivo del proyecto luego se procederá a la organización siguiendo las premisas de la definición de proyectos:

Planificación.- Determina la realización de acciones no repetitivas, únicas, con una determinada duración en el tiempo, mismas que van a estar organizadas y a las que se les asignaran recursos para su ejecución, estas actividades se dividen en tareas relacionadas entre sí.

Evidencias del desarrollo del proyecto.- Con lo que se procede a medir el progreso de proyecto se coordina con el dueño del proyecto en si la evidencia de desarrollo del proyecto vienen a ser las liberaciones cortas de producto.

Comunicación.- Pretende tener informados a todos los participantes sobre el progreso y evolución del proyecto, esto se realiza a través de informes.

2.2.5 Fase 5 Kick Off

Lo primero que se realizó en esta fase fue el levantamiento de información relacionada con el nuevo proyecto, tanto operacional, digital, de procesos, etc. Luego de recopilar esta información se procedió a analizarla con finalidad de poder plantear los objetivos que se van a tomar para el nuevo proyecto.

Con los pasos mencionados anteriormente se procede a generar el siguiente hito permite marcar el inicio o arranque formal de las actividades del proyecto, en esta se procederá a establecer los compromisos tanto del equipo de trabajo como de los clientes con referencia al proyecto, estableciendo un clima de confianza.

Se procede a la presentación del equipo de trabajo al cliente para confirmar así los roles y las responsabilidades que cada uno tendrá en el proyecto.

2.2.6 Fase 6 Sprint 0

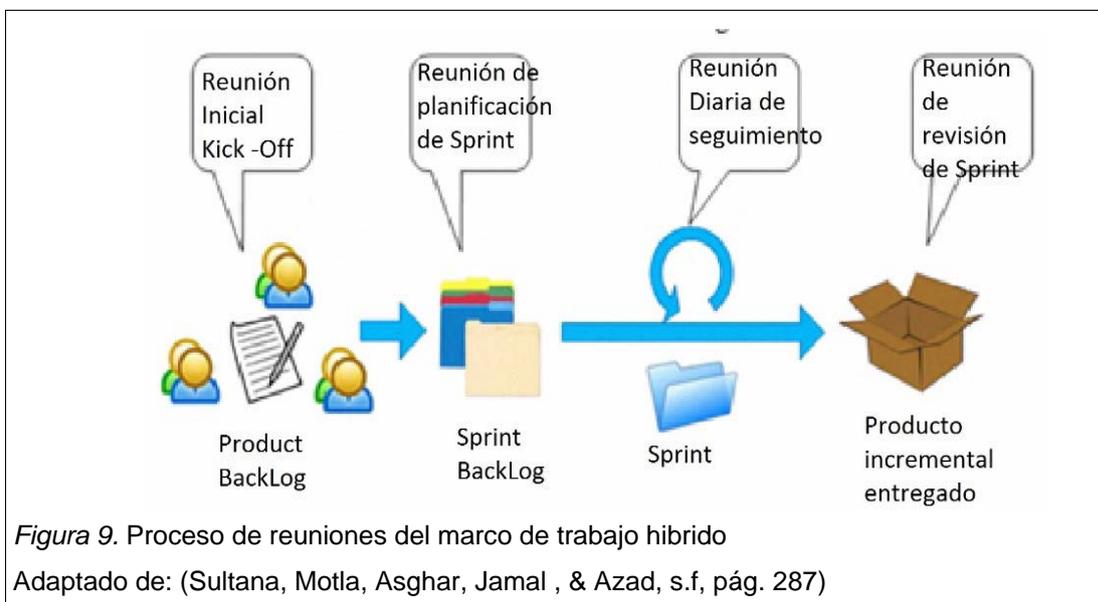
El objetivo de esta iteración es la de preparar el proyecto desde las siguientes perspectivas:

- Tecnológica
- Metodológica
- Organizativa

Con el fin de que el proyecto tenga un buen comienzo y una conclusión o finalización sin mayores contratiempos.

En esta iteración se procederá a realizar definiciones con el productowner tanto de las funcionalidades (historias de usuario) luego de esto con estas historias proceder a conformar el product Backlog.

Con esto se puede mantener ya una primera reunión con el equipo de trabajo para presentar las historias de usuario obtenidas para identificar posibles dudas que se deban resolver con el Product Owner y proponer mejoras o sugerencias a las historias.



2.2.7 Fase 7 Corrida del Modelo

En esta fase se comenzará ya el desarrollo de proyecto de software siguiendo las directrices señaladas en el nuevo marco de trabajo.

2.2.7.1 Paso 1 Sprint Planning

2.2.7.2 Paso 2 Sprint

2.2.7.3 Paso 3 Scrum Diario

2.2.7.4 Paso 4 Sprint Review Meeting (Demo)

2.2.7.5 Paso 5 Retrospectiva

2.2.7.6 Paso 6 Planificación de la siguiente iteración

2.2.7.7 Paso 7 Lecciones aprendidas por iteración

2.2.8 Fase 8 Lecciones aprendidas generales y cierre del proyecto

Esta fase se realizará constantemente durante todo el proceso a fin de revisar las acciones y estrategias que se necesitan tomar a futuro para evitar posibles amenazas o aprovechar oportunidades. Estas lecciones se obtendrán de las reuniones de retrospectiva que se realizarán al final de cada Sprint en las que se analizará el trabajo realizado en cuanto a alcance, costo, calidad y tiempo.

Para ayudar a tener un mejor manejo de las lecciones aprendidas se procederá a formular en cada una de las reuniones de retrospectiva un pequeño cuestionario, con el que se entenderá como avanza el proyecto y que es lo que se ha aprendido, estas preguntas son las siguientes (Gestiones sus proyectos, 2012):

- ¿Cuáles objetivos del proyecto se alcanzaron? (Fecha, Costo, Calidad, etc.).
- ¿Qué funcionó bien en nuestro proyecto?
- Para cada aspecto que salió bien u objetivo logrado: ¿Cuál es la causa raíz que desencadenó dicho resultado positivo?.
- ¿Cuáles objetivos no se alcanzaron? (Fecha, Costo, Calidad, etc.).
- ¿Qué no salió bien en nuestro proyecto?
- ¿Qué imprevistos (sorpresas) tuvo que manejar el equipo?
- ¿Qué circunstancias no fueron anticipadas?
- Para cada aspecto que no salió bien: ¿Cuál es la causa raíz que ocasionó la dificultad?

A continuación en el Anexo G se podrá encontrar un formato para la descripción de las lecciones aprendidas.

La etapa de cierre se inicia cuando se completa en su totalidad la ejecución del proyecto y el cliente está satisfecho y acepta el resultado final del producto a él entregado. Este se convierte en un escenario en el cual se verifican todos los criterios de cumplimiento solicitados. Por lo general en esta fase se acumulan o

realizan una serie de documentos que tienen que ver con recibir y hacer los pagos y entregas finales, así como las reuniones de evaluación.

En esta etapa se vinculan los grupos de procesos de gestión de proyectos para formalmente completar el ciclo. En esta fase se procede a generar una serie de documentos y entregables que son importantes para sellar la finalización del proyecto entre los que destacan los siguientes para el nuevo marco de trabajo.

Mandatorio: Sistema o producto solicitado: Módulo de Generación de tickets de compra

Documento de entrega del producto: Acta de entrega recepción

Opcional: Manuales de uso.

Manuales técnicos

A continuación en el anexo H se describe una plantilla para el documento de cierre del proyecto.

Capítulo 3. Aplicación a un caso de estudio

3.1 Descripción del caso de estudio y Aplicación del Modelo

A fin de validar la propuesta definida en capítulos anteriores sobre el conjunto de buenas prácticas que conforman el nuevo marco de trabajo ágil “Scrum + XP” de una manera más conceptual, se ha procedido a su aplicación a un caso de estudio en una empresa de desarrollo de software, misma que por temas de confidencialidad de denominará empresa XYZ con el proyecto de Creación de Tickets de Compra.

3.1.1 Fase 1 Aplicación de Encuesta para Estudio de Factibilidad

Se pretende realizar un proyecto sobre un módulo de Creación de Tickets de Compra en una empresa, para lo cual primero se aplicó una encuesta misma que ha permitido obtener un termómetro de cómo está el nivel para la aplicación de una metodología ágil. La encuesta aplicada se puede ver en el Anexo I de este documento.

De la encuesta realizada se ha obtenido el siguiente resultado:

La empresa XYZ es una empresa conformada por un número de 34 personas de las cuales 18 personas conforman el área de desarrollo de software, siendo así considerada dentro del espectro de las medianas empresas. Con este número de miembros de desarrollo es posible la conformación de equipos de trabajo en pares.

El puntaje obtenido en la encuesta fue de 70, lo que según la valoración presentada en capítulos anteriores quiere decir que la empresa cumple con un sistema de gestión de desarrollo, pero con algunas deficiencias, por lo tanto es apta para la aplicación de una metodología ágil ya que existen instancias que pueden ser mejoradas ya que como se afirma a partir de aquellas empresas

que tienen una puntuación sobre los 40 puntos es factible la aplicación de la metodología.

Una vez hecho el análisis de la encuesta se recomienda realizar mejoras en los siguientes puntos:

- Hacer que los equipos trabajen enfocados en un determinado proyecto y no en varios a la vez.
- Establecimiento de Políticas y objetivos de Calidad, ya que en la actualidad la empresa no cuenta con estas o no están del todo claras.
- Definición y limitación de roles.
- Entrega de proyectos a tiempo.
- Identificación de requerimientos del cliente y su correspondiente revisión antes de adquirir compromisos con el cliente.
- Tener un termómetro del nivel de satisfacción del cliente en cada entrega de incremento de producto.
- Hacer a los equipos auto organizado.
- Que los Equipos de trabajo tenga una visión completa del proyecto.
- Elección de trabajo en base a las aptitudes de cada miembro.
- Entrega de software de calidad.

3.1.2 Fase 2 Definición o ajuste de las políticas de Calidad

A continuación se presenta la política de calidad actual de la empresa:

Política de Calidad Empresa XYZ

Para orientar con el mayor éxito posible el sistema de gestión de calidad de productos y servicios de software, la empresa XYZ define como política general el potenciamiento y crecimiento de sus clientes, apoyados en el cumplimiento de sus objetivos estratégicos, mediante la provisión de soluciones tecnológicas financieras en “Cloud” de alto nivel. Además se pretende potenciar las actividades referentes a investigación, desarrollo e innovación con el propósito

de alcanzar mejores niveles competitivos en el sector con el fin de brindar la total satisfacción del cliente final.

Al ser una empresa pionera en el desarrollo de sistemas financieros en “Cloud” en el país, XYZ pretende ser un referente, por tal motivo es necesario manejarse bajo los más estrictos criterios de calidad, razón por la cual se han definido claramente sus objetivos y estrategias de Calidad Total:

Objetivos

- Generar una plataforma que soporte una rápida implementación de nuevos servicios tecnológicos financieros. (Velocidad = 70 Story Point / Iteración).
- Lograr eficiencia en el proceso de desarrollo ($\geq 80\%$).
- Lograr eficiencia en el proceso de operación en base a: Up-time, tiempos de respuesta, tiempo de resolución de incidentes y problemas, tiempo de implementación de RFC's (%).
- Proporcionar cumplimiento con leyes externas, regulaciones y contratos.
- Mantener un correcto manejo de # de issues de no cumplimiento.
- Disminuir el re trabajo en el proceso de desarrollo.

Estrategias

- Planes motivacionales para el personal.
- Alianzas estratégicas con otras empresas colaboradoras de interés mutuo.
- Planificación del trabajo a realizarse.

Compromiso

La empresa se ha comprometido a la difusión total de esta política con todo lo que ello conlleva a todo el personal y se compromete a destinar todos los recursos posibles para la formación de sus empleados, en la comprensión y participación en la misma.

En cuanto a la política, a partir de la información recopilada en la empresa, se sugirieron y aceptaron cambios y oportunidades de mejora, relacionadas con la

gestión de la calidad. A continuación se describen la política, objetivos y estrategias que a partir de este trabajo regirán en la empresa:

Política de Calidad Empresa XYZ (Actualizada)

Para orientar con el mayor éxito posible el sistema de gestión de calidad de productos y servicios de software, la empresa XYZ define como política general el potenciamiento y crecimiento de sus clientes, apoyados en el cumplimiento de sus objetivos estratégicos, mediante la provisión de soluciones tecnológicas financieras en “Cloud” de alto nivel. Además se pretende potenciar las actividades referentes a investigación, desarrollo e innovación con el propósito de alcanzar mejores niveles competitivos en el sector con el fin de brindar la total satisfacción del cliente final.

Al ser una empresa pionera en el desarrollo de sistemas financieros en “Cloud” en el país, XYZ pretende ser un referente, por tal motivo es necesario manejarse bajo los más estrictos criterios de calidad, razón por la cual se han definido claramente sus objetivos y estrategias de calidad total:

Objetivos

- Generar una plataforma que soporte una rápida implementación de nuevos servicios tecnológicos financieros. (Velocidad = 70 Story Point / Iteración).
- Lograr eficiencia en el proceso de desarrollo ($\geq 80\%$).
- Lograr eficiencia en el proceso de operación en base a: Up-time, tiempos de respuesta, tiempo de resolución de incidentes y problemas, tiempo de implementación de RFC's (%).
- Proporcionar cumplimiento con leyes externas, regulaciones y contratos.
- Mantener un correcto manejo de # de issues de no cumplimiento.
- Disminuir el re trabajo en el proceso de desarrollo.
- Creación de un ambiente adecuado para el desarrollo de los empleados tanto personal como profesional.
- Implementación y ejecución de planes anuales de capacitación para el personal de la compañía.

- Obtener una certificación internacional en calidad para la empresa.
- Reducción de Tiempos, costos y esfuerzos.
- Centralización de los procesos.
- Aseguramiento de Satisfacción del cliente.- Se debe asegurar que el producto o servicio de software cumpla a cabalidad con las expectativas del cliente.
- Conocer las necesidades y expectativas del cliente.- Conocer de manera profunda lo que el cliente las necesidades de nuestro cliente, de forma que se puedan establecer planes de acción y estrategias continuas para el cumplimiento de los objetivos.
- Optimización de procesos.- Mejora continua del funcionamiento de los procesos mediante la eficiencia.
- Uso de tecnología de vanguardia y probada para el desarrollo de nuestros productos y servicios.
- Uso de herramientas para pruebas de los productos.

Estrategias

- Planes motivacionales para el personal.
- Alianzas estratégicas con otras empresas colaboradoras de interés mutuo.
- Planificación del trabajo a realizarse.
- La Empresa es considerada como un referente en cuanto a productos y servicios de software, por lo que se debe aplicar una gestión calidad con bases en la mejora continua, excelencia y sobretodo en la innovación tanto en los productos y servicios ofrecidos, como en sus procesos de desarrollo, mismos que aportaran un valor agregado.
- Contar con consultores externos que aporten tanto en la calidad del negocio como en la calidad del desarrollo de las aplicaciones.
- Aplicaciones de ciclos permanentes y evaluativos de gestión de calidad.

Compromiso

La empresa se ha comprometido a la difusión total de esta política con todo lo que ello conlleva a todo el personal de la empresa y se compromete a destinar todos los recursos posibles para la formación de sus empleados, en la comprensión y participación en la misma.

A través de estas mejoras en cuanto a la política de calidad se pretende incrementar el desempeño a nivel de desarrollo de las aplicaciones en la organización, así como en conjunto áreas específicas de naturaleza técnica y decrecimiento personal que le aportan valor a la empresa.

Además de tener un mejor control de la calidad, lo que permite tener una mayor satisfacción de los clientes finales.

3.1.3 Fase 3 Definición del Proyecto

Se mantuvo una reunión entre el Cliente el Product Owner y el Scrum Master para realizar un acercamiento y primera revisión de lo que se quiere producir, en este caso se requiere la creación de un Módulo para la Creación de Tickets de Compras, entonces el Scrum Master tomó nota de todo lo que el Cliente y el Product Owner le platicaban sobre la funcionalidad de este proceso, produciéndose una lluvia de ideas. Esta charla se trasladó a Historias de Usuario, que luego fueron validadas por el cliente, y luego revisadas por el equipo completo. A continuación se describen las historias de usuario obtenidas en el formato recomendado. Las mismas pueden ser apreciadas en el Anexo J de este documento.

3.1.4 Fase 4 Conformación del Equipo

Para el proyecto se procedió a armar un equipo conformado por un total de 8 personas distribuidas de la siguiente manera:

Tabla 13. Cargos, Responsables Proyecto Creación Tickets Compras

Cargo	Descripción	Responsable
Product Owner	Encargado de Realizar y priorizar la lista de requerimientos	María Elena Romero
Scrum Master	Encargado de la planificación, ejecución y control del proyecto	Santiago León
Analista Negocio	Analista de los requerimientos de los usuarios y programación	Lucía Arias
Programador Senior	Desarrollo y arquitectura de la aplicación	David Quishpe
	Análisis y desarrollo	Eduardo Bustillos
Programador Junior	Desarrollo de aplicación	Luis Serrano
	Desarrollo de aplicación	Danilo Toapanta
Tester	Realizar pruebas funcionales de la aplicación y de aceptación	Franklin Tapia

Generado por el (Autor)

Al ser un equipo con experiencia en el trabajo con metodologías ágiles no es necesario realizar capacitaciones adicionales sobre metodologías o nuevas tendencias tecnológicas sin embargo se procedió a realizar una breve introducción al nuevo marco de trabajo poniendo énfasis en las premisas básicas como son:

- Roles
- Actividades
- Artefactos
- Valores
- Prácticas y Técnicas

La velocidad de Trabajo según las estimaciones realizadas en base a la experiencia del equipo en proyectos anteriores se ha determinado mediante la fórmula en:

$$Velocidad = \frac{Trabajo(númerodepuntosarealizar)}{tiempo(Sprint\ 2\ semanas)}$$

Lo que representado en valores da lo siguiente:

$$50 = \frac{100}{2}$$

De esta manera se puede estimar que en un equipo de desarrollo de estas características cada miembro puede realizar un esfuerzo de 25 puntos por iteración. Sin embargo esto no es del todo cierto, puede que por la experiencia de los miembros unos tengan que tomar más responsabilidades que otros.

La asignación de las responsabilidades será planteada de acuerdo a la matriz RACI descrita.

Tabla 14. Matriz RACI aplicada al proyecto

Elaborado por: Santiago León									
Actividad		Roles/Responsabilidades							
ID Actividad	Actividad	Product Owner	Scrum Master	Stakeholders	Analista de Negocio	Programador Senior	Programador Junior		
1	Product Backlog	A, R	I	C, I	I	I	I	I	
2	SprintBacklog	C	C, I	C, I	C, I	C, I	C, I	C, I	
3	Burndown charts	I	A, R	C, I	C, I	C, I	C, I	C, I	
4	Sprint Planning	I	A, R	C, I	C, I	C, I	C, I	C, I	
5	Sprint Review	I	A, R	C, I	C, I	C, I	C, I	C, I	
6	Sprintretrospective	C	A, R	C	C, I	C, I	C, I	C, I	
7	DailyScrum Meeting	C, I	A, R	C, I	C, I	C, I	I	I	
8	Sprint	I	A, R	I	I	I	I	I	
9	SPRINT PLANNING	I	A, R	I	I	I	I	I	
10	Product Backlog	I	A, R	I	I	I	I	I	
11	del Sprint	I	A, R	I	I	I	I	I	
12	Usuario	I	A, R	I	I	I	I	I	
13	Definir las tareas	I	A, R	I	I	I	I	I	
14	Estimar la duración de las Tareas	I	A, R	I	I	I	I	I	

15	Investigación	C, I	I	I	I	I	R	I
16	Desarrollo verificación y	C, I	I	I	I	I	R	I
17	Corrección de Bugs	C, I	I	I	I	I	R	I
18	Pruebas Unitarias	C, I	I	I	I	I	R	I

Tomado de: (Asignación Raci, 2014)

Tabla 15. Descripciones de responsabilidades y roles

Roles y Responsabilidades	
Rol / Responsabilidad	Descripción
R	Responsable: Este rol es el que realiza (ejecuta) el trabajo asociado con la actividad, lo habitual es que cada actividad tenga un solo "R", si existe más de uno es recomendable subdividir la actividad.
A	Aprobador: Es el encargado de aprobar (firmar), el trabajo realizado, a partir de esa aprobación, este se vuelve responsable por la actividad. Como regla general debe existir un solo "A" por actividad. Este rol es quien asegura que se ejecutan las tareas, por ejemplo Líderes de área técnica, área de gestión de proyecto, entre otros.
C	Consultado: Posee alguna información o capacidad que se necesita para mantener el trabajo. Se le informa y consulta información, de manera bidireccional con el responsable y/o aprobador.
I	Informado: Rol que debe ser informado sobre el progreso y los resultados del trabajo. En este caso la comunicación es unidireccional (se le da información pero no se recibe información).

Tomado de:(Asignación Raci, 2014)

3.1.5 Fase 5 Kick Off

En esta reunión se procedió a generar una presentación del equipo de trabajo al Cliente y Product Owner. Los objetivos que se dieron en esta reunión fueron:

- El establecimiento de los compromisos que deben asumir todos los involucrados en el proyecto.
- Por parte del equipo de trabajo encabezado por el Scrum Master el de generar un clima de confianza al cliente y ProductOwner para el proyecto.
- Aclaración de dudas de los involucrados.
- La confirmación de los roles y responsabilidades que se tendrá en el proyecto.

- Revisión de los riesgos del proyecto
- Formas como se van a realizar las comunicaciones de los avances.

3.1.6 Fase 6 Sprint 0

En esta fase se realizó la preparación de los ambientes de desarrollo, tanto en los servidores como en los equipos locales, se trabajó en la alineación del nuevo proyecto a la arquitectura actual que maneja la Empresa en otros proyectos.

Además de esto se trabajó en el Product Backlog, sobre todo en la priorización de las historias de usuario y las estimaciones necesarias lo que dio lugar al documento de Product Backlog que se encuentra en el Anexo K de este documento.

3.1.7 Fase 7 Corrida del Modelo

3.1.7.1 Paso 1 Sprint Planning Iteración o Sprint 1

Como primer paso se procedió a realizar una reunión en la que se conversó con el Cliente, en esta reunión el cliente presento al equipo de desarrollo la lista priorizada del Proyecto, así como los requerimientos que son importantes en la iteración que se está planeando. Una vez hecho esto el equipo examinó esta lista, y procedió a realizar cuestionamientos al cliente a fin de resolver dudas, todo esto en base al Product Backlog creado anteriormente con las historias de usuario.

Luego se planificó una segunda reunión, en la que el equipo de desarrollo en conjunto con el Scrum Master planificó la iteración, mediante la elaboración de varias tácticas para analizar los requerimientos, los tiempos y la utilización del menor esfuerzo.

En lo referente a estimación de tiempos, storypoints se realizó mediante la técnica de la experiencia, es decir se compararon las historias de usuario con historias de otros módulos o proyectos a fin de definir cuantos puntos cada una de estas historias iba a costar y tomar hacer estas estimaciones de realizaron en conjunto con el equipo de trabajo consultando y revisando las historias escritas en el Product Backlog este tipo de estimación también ayudo a descubrir historias de usuario demasiado grandes, mismas que fueron descompuestas en historias más pequeñas.

Para esto el equipo de desarrollo se apoyó en la utilización de diseño simple de tareas y en metáfora para crear las mismas como se aprecia un ejemplo en el Anexo L.

Una vez obtenidas las metáforas que se usaran para el proyecto se procedió al levantamiento de las tarjetas clase responsabilidad colaboración, en las que se reconocieron las principales clases a ser creadas para la primera iteración, estas podrán ser observadas en el Anexo M.

Antes de continuar con la planificación de tareas, es importante la estimación de horas disponibles por cada miembro del equipo por lo que se procedió a realizar este proceso para así determinar el número de horas que cada miembro del equipo tendrá disponible para trabajar en la iteración mencionada mediante la siguiente tabla:

Tabla 16. Estimación de horas disponibles de miembros del equipo para Sprint 1

Longitud del Sprint			2 semanas
Días laborables durante el Sprint			8 días
Miembro del equipo	Días Disponibles durante el Sprint	Horas disponibles por día	Total de horas disponibles
Santiago León	6	5	30
Lucía Arias	7	5	35
Luis Serrano	8	6	48
Eduardo Bustillos	8	6	48
David Quishpe	8	6	48
Danilo Toapanta	8	6	48

Tomado de:(Deemer, Benefield, Larman, & Bas, 2009, pág. 10)

Con esta información, el equipo de desarrollo en conjunto con el Scrum Master toman las historias de usuarios que se van a trabajar en el Sprint y las activan para asumir el compromiso de entrega de estas al finalizar la iteración pasando del estado de “Planificada” a “En Proceso” como se aprecia a continuación:

Tabla 17. Activación de Historias de usuario en el productBacklog

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Iteración (Sprint)	Prioridad
1	Como un usuario del sistema, necesito la generación de un ticket de compra, con la finalidad de ingresar y generar un ticket para solicitar la compra de un bien o servicio a el área de administrativo de la empresa y que este sea validado por las debidas autoridades de la institución, así como la asignación de los proveedores a los que se les puede solicitar los bienes o servicios	En Proceso	Sprint 1	1
2	Como Administrador de compras, necesito Realizar un análisis de los tickets de compra que emiten los usuarios, para que de acuerdo a valores y otros factores estos sean aprobados, o rechazados	En proceso	Sprint 1	2

Tomado de:(Plantilla Pila Producto, 2014)

Con esta información el equipo de desarrollo procede a la definición de las tareas que se van a realizar para completar cada historia de usuario comprometida en el Product Backlog para la iteración, todo esto basado siempre en una definición de listo.

Cabe mencionar que la planificación de la iteración fue determinada para un periodo de tiempo de 2 semanas (9 días hábiles) con lo que se cumple con la práctica del nuevo marco de trabajo de tener iteraciones cortas de trabajo.

Tomando como referencia la primera historia de usuario, se procedió a sacar las siguientes tareas, que se pueden apreciar en el Anexo J. Con estas cada miembro del equipo procedió a la auto – asignación de las mismas, en base a sus habilidades y experiencias y de igual manera en base a estas características cada uno de los miembros estimó el tiempo que le llevará a completar cada una de las tareas asignadas por ejemplo creación de estructuras de base de datos con una estimación de esfuerzo y tiempo de 4 horas, y así sucesivamente. De esta manera se cumple con el valor del nuevo marco de trabajo de responsabilidad.

Con esto se cierra la primera parte del nuevo marco de trabajo, dando inicio a la segunda parte que es el desarrollo en si del producto que se debe entregar a final del Sprint.

3.1.7.2 Paso 2 Sprint

En esta fase se procedió ya a realizar la codificación de las tareas creadas anteriormente y que ya se encuentran comprometidas para su ejecución, para esto se realizó trabajos con técnicas de TDD como se muestra a continuación:

1. Codificación

Las Tarea N° 7 y 8 descrita en el Anexo J, son tareas de generación de métodos para la manipulación de la información para preservar en el repositorio, pertenecientes a la cabecera y detalle de una orden de compra (Ticket), para esto se procedió al uso de Test Driven

Development que es una forma de construir software basado en pruebas.

Mediante un extracto de código se procede a mostrar un ejemplo de la aplicación de esta técnica basada en la tarea N° 7 y 8 Generando los métodos de inserción, actualización y eliminación de la cabecera y detalle de la compra.

2. Pruebas unitarias

Se procedió primero con la creación de un Proyecto de clase llamado Procesos Compras, mismo que contenía el código de inserción, actualización y eliminación tanto de la cabecera como del detalle de la compra, y para la realización de las pruebas a este código, se agregó un nuevo proyecto de tipo Test.

Luego se añadió una clase llamada Compra al Proyecto y sobre el proyecto de test se añadió una clase de tipo Test llamada Guardar Cabecera Compra Test, a la que se procedió a agregar la correspondiente referencia al proyecto Procesos Compras.

Para hacer trabajos con TDD se aplicó la técnica del semáforo en la que si la prueba da Rojo aún no ha pasado la misma, Verde significa refactoring, es decir primero escribimos la prueba, probamos, falla, luego escribimos el código necesario para que funciones y luego se procede a comprobar si existe algo que se pueda mejorar o refactorizar.

En base a esto, lo siguiente que se hizo fue realizar la prueba más simple, la inserción de los datos de la cabecera. De tal forma que la primera prueba unitaria se veía como se muestra a continuación:

```
[TestClass]
public class GuardarCabeceraCompraTest
{
    [TestMethod]
    public DataTransfer GuardarDocumentoCompletoAdquisicion(DataTransfer objetosDocumento)
    {
        ProcesosCompras objRegistroCompra = new ProcespsCompras(dao);
        DataTransfer retorno = new DataTransfer();

        AdquisicionCompleto documentoCompleto = new AdquisicionCompleto();
        int idCompra = 0;
        int idDetalle = 0;
```

```

try
{
    using (TransactionScope txnFacturacion =
        new TransactionScope(TransactionScopeOption.Required))
    {
        documentoCompleto = (AdquisicionCompleto)objetosDocumento[0];

        //Obtener
        idDocumento idCompra=objRegistroCompra.GuardarCompraCabecera
            (documentoCompleto);

        List<AdquisicionDetalles>listaDetalle = newList<AdquisicionDetalles>();
        listaDetalle = (List<AdquisicionDetalles>)objetosDocumento[1];

        foreach (var detalle in listaDetalle)
        {
            idDetalle = objRegistroCompra.GuardarCompraDetalle(idCompra,
                detalle.Cantidad, detalle.Descripcion, detalle.PasajeRemodelacion,
                detalle.Observacion, detalle.IdEstado, detalle.CantidadAprobada,
                detalle.Aprobado);
        }
        retorno.Add(true);
        retorno.Add(idCompra);
        txnFacturacion.Complete();
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return retorno;
}
}

```

Código generado por(Autor)

Se puede apreciar que automáticamente el compilador de la herramienta, en este caso Visual Estudio informó que no existen los métodos Guardar Compra Cabecera y Guardar Compra Detalle en la correspondiente clase Procesos Compras, por lo que este nos permite generar estos métodos.

Una vez generados se compilo y se procedió a la ejecución de la prueba haciendo click derecho en el método(Microsoft, 2015).

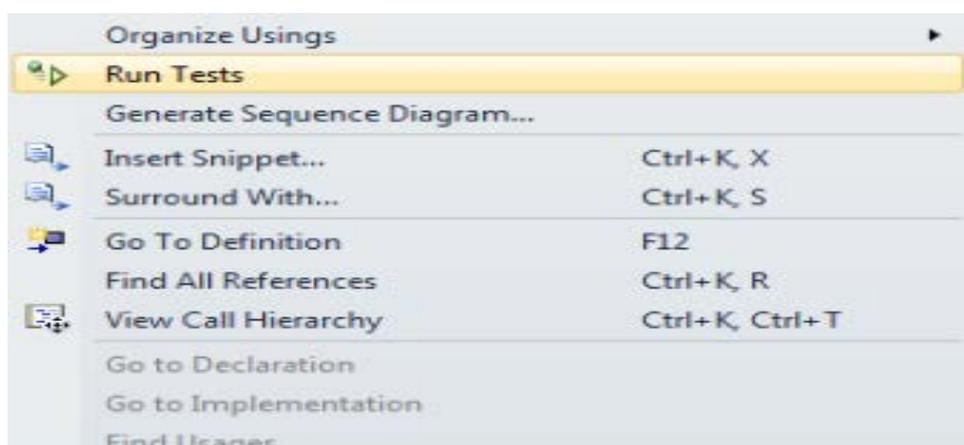


Figura 10. Ejecución de Test

Código Generado por el (Autor) en herramienta Visual Studio 2013

Al hacer esto se produjo el siguiente Fallo.

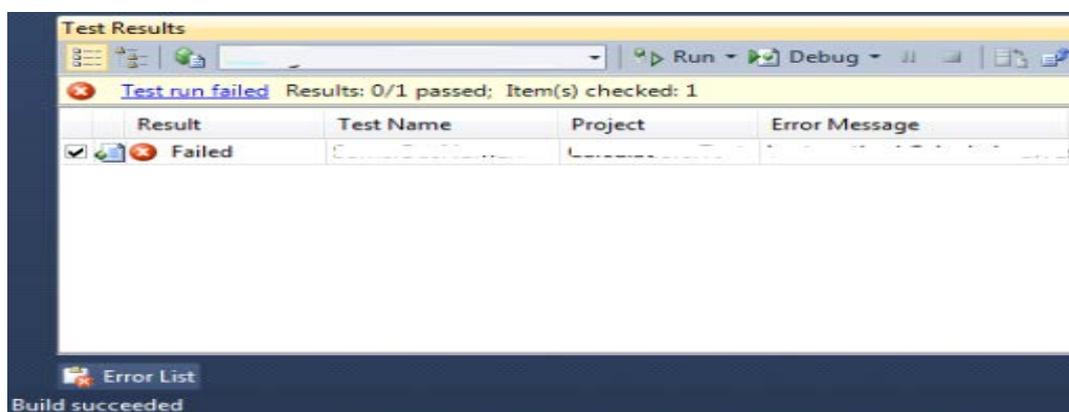


Figura 11. Resultado de Test Semáforo en Rojo

Código generado por (Autor) en herramienta Visual Studio 2013

Con esto se comprobó la primera parte de la programación basada en TDD, luego lo siguiente fue intentar poner la ejecución de la prueba en verde realizando los cambios necesarios para que el test pase(Microsoft, 2015).

```
public int GuardarCompraCabecera(AdquisicionCompleto cabecera)
{
    int retorno = 1;

    return retorno;
}

public int GuardarCompraDetalle(int idCompra, int Cantidad, string Descripcion,
string Observacion, int IdEstado, int CantidadAprobada, int Aprobado)
{
    int retorno = 1;

    return retorno;
}
```

Código generado por(Autor)

Con estos cambios se volvió a ejecutar la prueba y se comprobó su adecuado funcionamiento.

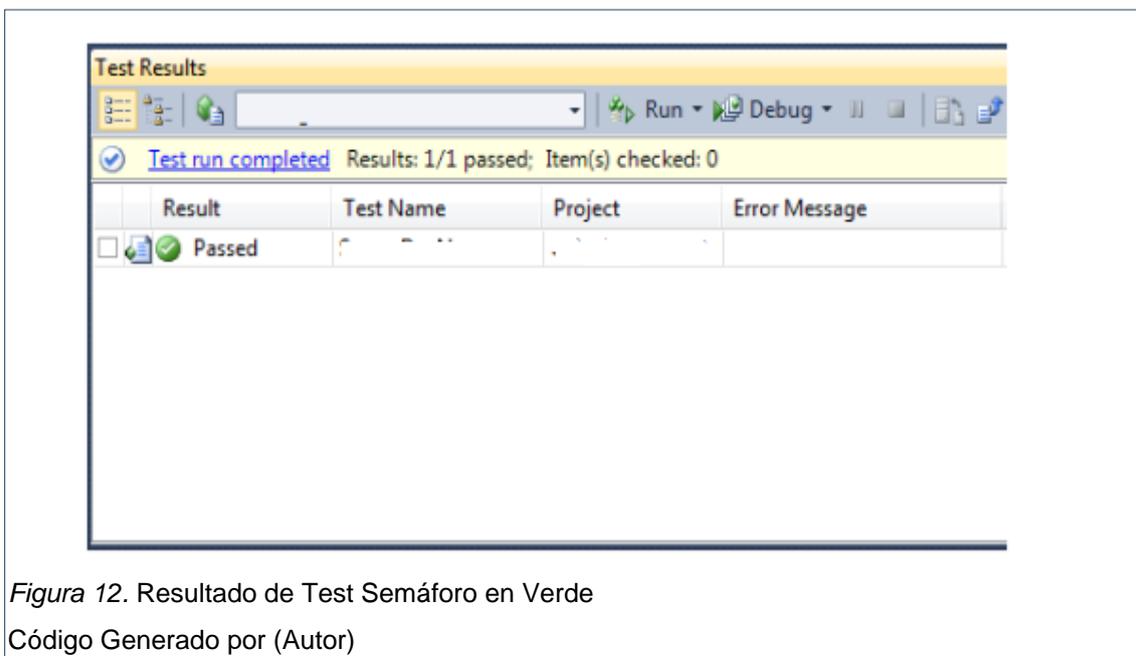


Figura 12. Resultado de Test Semáforo en Verde
Código Generado por (Autor)

En esta etapa de codificación participaron en tareas en pares los programadores Eduardo Bustillos y Luis Serrano, quienes usaron la técnica de ping-pong (Programación en parejas, 2009) lo que hizo la programación mucho más dinámica e interactiva realizando el trabajo de la siguiente manera:

- Luis Serrano escribió el código para la prueba y la dejó fallar como se apreció anteriormente.
- Eduardo Bustillos implementó el código para hacer pasar la prueba.
- Luego Eduardo Bustillos creó el código para otra prueba
- Y por último Luis Serrano escribió el código para que la anterior prueba pase y así sucesivamente.
- De igual manera luego la refactorización del código la realizaron los 2 programadores, el primero digitando y el segundo revisando y proponiendo ideas de mejora.

En esta fase fue esencial la realización de pausas para que los involucrados en tareas en pares puedan descansar y desestresarse, el hecho de realizar este trabajo en parejas no disminuyó la calidad ni la productividad.

3. Refactorización del Código

Lo siguiente que se hizo una vez comprobado el correcto funcionamiento del test es refactorizar con el objetivo de simplificar o de arreglar o eliminar código innecesario. A continuación se realizó la refactorización de los métodos de grabación para añadir código de validación (Refactorización, 2015).

```
public int GuardarCompraCabecera(AdquisicionCompleto cabecera)
{
    int retorno = 1;

    try
    {
        if (cabecera == null)
```

```
        {  
            Return100; // Código de error, No existen datos para grabar  
        }  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    return retorno;  
}
```

Código Generado por (Autor)

4. Código Estándar.

Como se pudo apreciar en el código ingresado como ejemplo anteriormente se usó estandarización en el mismo, con variables mnemotécnicas para saber el tipo de cada una de las variables como por ejemplo:

intCompras = variable que indica que el tipo de dato devuelto es un entero y sirve para un código de compra.

Se usaron nombres de variables sugestivos para conocer la naturaleza y uso de las mismas como por ejemplo:

Guardar Comprobante Detalle = Método que guarda los datos del detalle de un comprobante de compra.

De igual manera se usó el estilo Camelcase (Estilo de programación, 2008) Para la codificación del programa.

5. Pruebas de Aceptación.

Una vez terminadas las codificaciones y funcionalidades, se procedió a realizar pruebas de aceptación a través del personal asignado con la supervisión y operación del usuario experto, para esto se procedió a la utilización de casos de pruebas como se puede apreciar en el Anexo O,

en estos casos de prueba, se indicó las funcionalidades que debía cumplir la aplicación, áreas a las que pertenecen las pruebas y los resultados obtenidos. En este caso se encontraron varios errores y afinamientos que fueron reportados a los miembros del equipo de desarrollo a través de los casos de prueba y luego fueron subidos a la herramienta de tablero de tareas pero identificando a las tareas como bug, o afinamientos, luego en las reuniones diarias se procedieron a revisar estas y se discutió la prioridad para la terminación del Sprint, si estas eran demasiado críticas se procedía a su corrección caso contrario se las pasaba para ser atendidas en las próximas iteraciones.

6. Pruebas de Integración

Al momento en que se realizaron estas pruebas se tomó mucho en cuenta el nivel de complejidad que suponía cada uno de los componentes ya desarrollados y probados unitariamente. Para esto se procedió a realizar la integración de los componentes, en la que se fueron agregando una por una las funcionalidades que ya fueron probadas y aprobadas en las pruebas unitarias hasta integrar todo lo que se necesita para el proyecto.

En este caso se procedió a la integración y pruebas de las funcionalidades de las tareas 7 y 8 creación de cabecera y detalles de una orden de compra que ya se analizaron anteriormente con la tarea nº 9 generación de datos para proformas, cada una de ellas trabaja y fue probada de forma unitaria y luego se procedió a realizar su integración y nuevamente a probar de manera total, como se puede apreciar en el Anexo P de este documento, mediante un guion de pruebas de integración(Pruebas de Integración, s.f.).

7. Propiedad Colectiva.

Todo el código realizado anteriormente fue escrito dentro de un arquitectura y estilo en común con el objetivo de dar calidad al usuario, cabe mencionar que el código creado no pertenece al programador o

programadores que lo crearon, este forma parte del código total del equipo de trabajo por lo que si existen funciones que pueden ser reutilizadas por otros programadores serán usadas. Además el código desarrollado vino a formar parte de un proyecto mucho más grande por lo que es responsabilidad de todos los miembros de este y otros equipos, con lo que se cumple con la práctica de integración continua del código, para esto también se usó como repositorio del código la herramienta Team Foundation Server (Kniberg, Sutherland y Cohn, 2007).

3.1.7.3 Paso 3 Scrum Diario

Se planificó a partir de las 10 de la mañana cada día durante el primer Sprint realizar una reunión de no más de 15 minutos luego de que todos los miembros del equipo de desarrollo hayan actualizado sus tareas en las que estaban trabajando, esta reunión permitía la sincronización de todo el equipo además de establecer la planificación de actividades para las siguientes horas de trabajo. Los miembros del equipo están enterados del propósito de esta reunión y se pidió principalmente estar enfocados en 3 temas fundamentales:

1. ¿Qué hice ayer?
2. ¿Qué voy a hacer hoy?
3. ¿Qué impedimentos tengo?

Esto permitió estar en conocimiento de cada una de las actividades en las que se encontraba cada uno de los miembros del equipo así como saber si existían impedimentos que puedan ser analizados por todos de forma breve y luego resuelto por el ScrumMaster y la persona que levantó la alerta.

Las entradas principales que se manejaron fueron:

- La Pila del Sprint y el gráfico de avance (burn- down) actualizado a la fecha.

- Información del avance de cada miembro del equipo, reflejado en el tablero de tareas. En este caso se usó un tablero electrónico que se encuentra integrado en el Visual Studio y el Team Foundation Server. Como se puede ver a continuación.

Con la actividad de ScrumDiario se cumplen también los valores de comunicación interna y retroalimentación de la información.

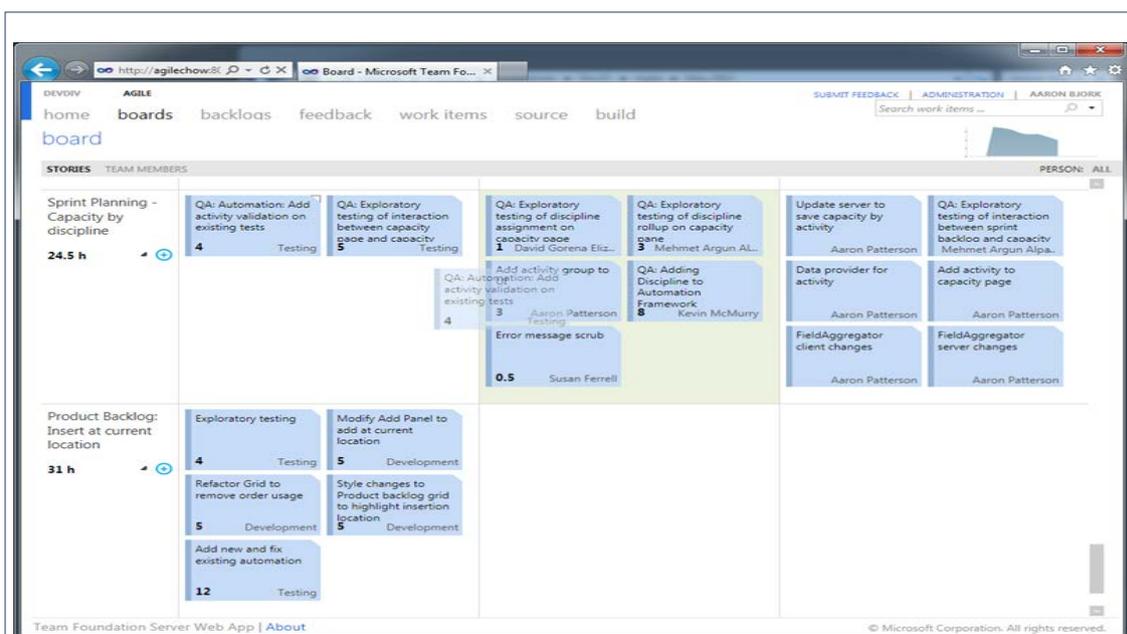


Figura 13. Tablero de Tareas Proyecto Ticket Compras

Generado por el (Autor)

3.1.7.4 Paso 4 Sprint Review Meeting (Demo)

Una vez finalizada la iteración de 2 semanas, con un tiempo adecuado el Scrum master procedió a convocar a una reunión de demostración de la funcionalidad terminada a los interesados.

En esta convocatoria el Scrum Master hizo saber mediante una agenda que es lo que se iba a ver en la correspondiente reunión como se muestra a continuación:

“Estimados, se solicita su asistencia para el Demo N° 1 del proyecto de Compras, mismo que se llevará a cabo el día Viernes 18 de octubre del 2014 a las 10:00.

La agenda a tratar en esta reunión será la siguiente:

1. Explicación de las metas del Sprint.
2. Presentación de Funcionalidades para la creación de Tickets de compra vía Administrativo 1.
3. Presentación de Funcionalidades para la creación de Tickets de compra vía Administrativo 2.
4. Presentación de funcionalidades de pantallas de análisis de compras y aprobación de tickets.
5. Presentación de Funcionalidades para la creación Proformas
6. Presentación de Funcionalidades para la creación y mantenimiento de Proveedores.
7. Preguntas y observaciones de los asistentes.”

Generado por el (Autor)

El objetivo de esta reunión fue la de presentar a los stakeholders y al Product Owner la funcionalidad que fue completada. Esta primera reunión de Demo tuvo una duración de 3 horas.

Toda la funcionalidad presentada se ejecutó desde un servidor con todas las características y parecidos a los servidores y ambientes de producción, pero en un ambiente intermedio de pruebas.

En esta reunión los stakeholders y el Product Owner realizaron comentarios de mejoras, observaciones y críticas sobre el incremento, las cuales fueron tomadas en cuenta por el personal de trabajo para ver su prioridad y darles tratamiento.

En esta actividad también se puede apreciar los valores de comunicación y retroalimentación de la información pero a nivel de equipo y stakeholders.

Al finalizar esta reunión el Cliente cuenta ya con una liberación de código probado y listo para usarse. De esta manera se cumple otra de las prácticas del nuevo marco de trabajo que viene a ser las liberaciones cortas de producto funcional y utilizable.

3.1.7.5 Paso 5 Retrospectiva

Esta reunión se realizó al finalizar la demostración del incremento en la funcionalidad con el fin de mejorar de manera continua la productividad del equipo de trabajo y del producto, se comenzó analizando la forma en la que se trabajó en la correspondiente iteración (Sprint N°1), se analizaron el por qué se consiguieron los objetivos comprometidos y que fue mal para no completar los que no se dieron. La base de esta reunión estuvo dada por las siguientes preguntas que el Scrum master realizó a cada uno de los miembros del equipo:

1. ¿Qué se hizo bien en esta iteración?
2. ¿Qué se hizo mal?
3. ¿Qué cosas debemos mejorar para la próxima iteración?
4. ¿Qué se ha aprendido?

De forma que todos los involucrados estén al tanto de que es lo que está pasando en el proyecto también el Scrum master procede a informar el estatus del mismo, cuantos Sprints faltan, la velocidad de desarrollo actual, entre otros indicadores.

A continuación un ejemplo de la retrospectiva practicada en la primera iteración del proyecto:

Resumen de Retrospectiva

1. ¿Qué se hizo bien en esta iteración?
 - Terminar la creación de la historia de usuario Creación de Tickets Compra.
 - Afinamientos de creación de proveedores.

- Creación de proformas.
 - Pruebas en ambiente de testing.
2. ¿Qué se hizo mal?
- No se ha mantenido la velocidad de desarrollo esperada (90 puntos).
 - Gran cantidad de bugs de prioridad baja.
 - No actualizar tareas.
 - Problemas en comunicación interna y externa.
 - A pesar de haber realizado un buen trabajo a lo largo del Sprint existió mucha dependencia del Scrum master por parte del equipo de trabajo.
 - Compromisos no fueron totalmente asumidos.
3. ¿Qué cosas debemos mejorar para la próxima iteración?
- Actualización de tareas diariamente.
 - Mejorar las pruebas de desarrollo para evitar bugs prioridad baja.
 - Revisión de ambientes testing.
 - Comunicación interna y comunicación externa con el cliente.
 - Afinamientos en tareas en pares.
4. ¿Qué se ha aprendido?
- Procesos de análisis de compras.
 - Comunicación con el cliente.
 - Configuración de ambientes intermedios.

“Informe del Scrum Master al equipo

Nº de Sprints terminados: 1 de 9

Nº de Sprints restantes: 8

Velocidad de desarrollo actual: 90 /100

Fecha de terminación del proyecto: 31/03/2015

3.1.7.6 Paso 6 Planificación de la siguiente iteración

Esta reunión se realizó con los Stakeholders y con el equipo de trabajo y se procedió a analizar el cambio potencial del Product Backlog en base a las observaciones que los involucrados tuvieron en la reunión de demo, para este caso solamente hubieron observaciones de forma más que de fondo y el cambio en la forma de generación de los tickets, el Product Backlog cambio en la prioridad de aprobación de compras bienes y servicios , por una observación realizada por el Product Owner en la reunión de demostración, por la historia de usuario, como se puede observar en el Anexo J la primera historia de usuario para el Sprint 2 era la de aprobación proveedor y compra del bien, pero de acuerdo a las observaciones se requiere que primero se realice la historia de usuario aprobación por montos por lo que el Product Backlog quedaría de la siguiente manera para el Sprint número 2:"

Tabla 18. Planificación de las siguientes iteraciones

Identificador (ID) de la Historia	Enunciado de la Historia	Alias	Estado	Dimensión / Esfuerzo	Iteración (Sprint)	Comentarios
5	Como Administrador de compras deseo que exista un nivel de aprobación de compras intermedio validado por montos de acuerdo a los mismos se realicen aprobaciones por niveles de gerencias y juntas.	Aprobaciones por montos	Nueva	30	Sprint2	
3	Como Administrador de compras, necesito validar la y aprobar los proveedores y productos o servicios que los usuarios están solicitados comprar en base a diferentes validaciones de los manuales de compras que maneja la institución.	Aprobación proveedor y compra del bien o servicio	Nueva	100	Sprint 2	
4	poder validar y realizar el proceso de compra de un bien o servicio solicitado por un usuario y validado por las distintas autoridades de la institución	Compra del bien o servicio	Nueva	100	Sprint 2	

Tomado de: (Plantilla Pila Producto, 2014)

3.1.7.7 Paso 7 Lecciones aprendidas por iteración (Sprint 1)

En esta etapa se procede a levantar todas las observaciones positivas y negativas que se fueron detectando a lo largo del sprint 1 y se procede a detallar la manera en la que estos impedimentos de tipo técnico, táctico o de gestión fueron solucionados, de tal manera que puedan servir como ayuda para futuros eventos similares que se presenten

Las lecciones aprendidas para el sprint 1 se resumen en la siguiente plantilla usada para su gestión, se manejará una por iteración:

Tabla 19. Lecciones Aprendidas Sprint 1

Proyecto		Generación de Tickets de compra						
Fecha Inicio:		06/10/2014		Fecha Fin:		18/10/2014		
Entidad Ejecutora:		Todas la Áreas						
Líder de Proyecto:		Santiago León						
Financiador del Proyecto:								
Miembros del Equipo:		Luis Serrano, Eduardo Bustillos, David Quishpe, Danilo Toapanta						
Cliente Final:		Maria Elena Romero						
#	Tema	Descripción	Fase del Proyecto	Categoría	Acciones Implementadas	Resultados Obtenidos	Recomendaciones	
1	Descripción de requerimientos	Se describen en detalle. Lo que ocasionó interpretaciones varias de las tareas no cumplían con la definición de listo	Sprint 1	Gestión de Requerimientos	Se realizaron reuniones con los usuarios para describir el comportamiento de las	Retraso en el arranque de algunas historias de usuario(análisis de compras)	Implementar un control de aprobación de los documentos funcionales.	
2	Definición de Listo		Sprint 1	Gestión Técnica	coordinar la definición de listo terminado las mismas y pasando las respectivas	que volver a revisar y en algunos casos refactorizadas y	Implementación de controles de pruebas.	

3	Organización de equipo multidisciplinario	Acostumbrar al equipo a trabajar en varias áreas del mismo proyecto	Sprint 1	Gestión Técnica	Disciplina en las reuniones y organización con el Scrum master para acoplarse y organizarse a los miembros de	Mejoras en la organización de los equipos	Mayor disciplina en los equipos de trabajo
4	Organización de tareas en pares	Organización de tareas en pares dentro del equipo	Sprint 1	Gestión Técnica	Organización y acoplamiento de los programadores para la programación en pares	Mejor organización del código, menos problemas de	Acostumbrar a los miembros del equipo a realizar subidas y bajadas de código con pruebas del mismo
5	Integración continua	Acoplamiento con la herramienta de teamfundation server	Sprint 1	Gestión técnica	Organizar al equipo para que se acostumbren a realizar la subida del código que crean y prueban, y también a bajar las últimas	Mejor organización del código, menos problemas de versionamie	Acostumbrar a los miembros del equipo a realizar subidas y bajadas de código con pruebas del mismo

Tomado de: (Plantilla Lecciones Aprendidas, 2014)

3.1.7.8 Paso 1 Sprint Planning Iteración o Sprint 2

En esta ocasión se procedió nuevamente con la planificación de la iteración para el segundo Sprint siguiendo el mismo patrón del Sprint 1, pero basado en el Backlog priorizado por el cliente con los cambios solicitados, además de las observaciones que se obtuvieron del Sprint 1 como el mejoramiento de la comunicación interna entre otras cosas.

Con la experiencia obtenida en la primera iteración en cuanto a planificación de tareas, tiempos, aptitudes y comunicación la planificación de esta segunda iteración se realizó con mayor rapidez y con un mejor entendimiento por parte de los miembros del equipo de trabajo.

La velocidad para este Sprint se planificó en 100 storypoints para todo el equipo con el compromiso de lograr esta meta puesto que en la primera iteración el objetivo no se logró, debido a la curva de aprendizaje y adaptación al nuevo marco de trabajo, a pesar de que el equipo cuenta ya con experiencia en metodologías ágiles.

Se realizó la estimación de horas para cada uno de los miembros del equipo de trabajo de igual manera que en la iteración anterior.

Tabla 20. Estimación de horas disponibles de miembros del equipo para Sprint 2

Longitud del Sprint			2 semanas
Días laborables durante el Sprint			8 días
Miembro del equipo	Días Disponibles durante el Sprint	Horas disponibles por día	Total de horas disponibles
Santiago León	6	5	30
Lucía Arias	7	5	35
Luis Serrano	8	6	48
Eduardo Bustillos	8	6	48
David Quishpe	8	6	48
Danilo Toapanta	8	6	48

Tomado de: (Deemer, Benefield, Larman, & Bas, 2009, pág. 10)

Luego se procedió a generar el nuevo Product Backlog para la iteración 2 y activar las tareas como se muestra a continuación:

Tabla 21. Priorización y activación de historias de usuario en el Product Backlog Sprint 2

Identificador (ID) de la Historia	Enunciado de la Historia	Estado	Iteración (Sprint)	Prioridad
5	Como Administrador de compras deseo que exista un nivel de aprobación de compras intermedio validado por montos de acuerdo a los mismos se realicen aprobaciones por niveles de gerencias y juntas.	En proceso	Sprint2	1
3	Como Administrador de compras, necesito validar la y aprobar los proveedores y productos o servicios que los usuarios están solicitando comprar en base a diferentes validaciones de los manuales de compras que maneja la institución.	Nueva	Sprint 2	2
4	Como administrador del sistema requiero poder validar y realizar el proceso de compra de un bien o servicio solicitado por un usuario y validado por las distintas autoridades de la institución.	Nueva	Sprint 2	3
6	Tareas de ingeniería y Corrección de Bugs	Nuevo	Sprint 2	4

Tomado de: (Plantilla Pila Producto, 2014)

En base a esto y a lo aprendido en la iteración anterior las historias de usuario y las tareas se planificaron de mejor forma y con tiempos mucho más reales de acuerdo a cada una de las tareas y a la dificultad que estas podrían presentar.

En esta nueva iteración se procedió a realizar un trabajo de reorganización del personal debido a la existencia de algunos bugs de alta prioridad encontrados en la iteración anterior para lo cual se procedió de la siguiente manera:

Se tomó a una parte del equipo para que se dedique a la corrección de los bugs más importantes encontrados en la iteración anterior, en conjunto con los miembros del equipo de pruebas y con la ayuda de los usuarios, a este equipo se le conoce como equipo de soporte, el cual va a estar concentrado en proteger al primer equipo de desarrollo (Henrik & Reza, 2008).

3.1.7.9 Paso 2 Sprint 2

La codificación en la iteración 2 de igual forma se realizó utilizando la técnica de TDD de igual manera que en la anterior, se pudo detectar mejoras basadas ya en las lecciones aprendidas en el Sprint anterior tal es el caso de los estándares de programación.

En lo que respecta a las pruebas también se pudo detectar mejoras al tener control sobre los guiones de pruebas con lo que se trabajó.

La refactorización y la propiedad colectiva de código siguen manteniendo el mismo formato de trabajo debido a que ya se ha logrado que los miembros del equipo se acoplen a la misma. Sin embargo la programación en pares si bien en la primera iteración fue buena, tuvo sus problemas de acoplamiento que en esta iteración se han visto reflejados pero en menor medida.

3.1.7.10 Paso 3 Scrum Diario Sprint 2

Las reuniones se mantienen en el mismo estándar y formato que en el Sprint anterior con las revisiones y preguntas frecuentes y las correspondientes revisiones en los tableros y gráficos de avances.

En esta iteración se pudo verificar y corregir el hecho de que los programadores no actualizaban sus tareas diariamente, con esto se llegó a tener mejor control en los cuadros, gráficos e indicadores de avances del proyecto.

De igual manera se aplicó el uso de las preguntas en cada una de las reuniones:

1. ¿Qué hice ayer?
2. ¿Qué voy a hacer hoy?
3. ¿Qué impedimentos tengo?

Con la observación de que en esta iteración la ayuda del Scrum Master fue mejor enfocada, sobre todo en asuntos que tenían que ver con preguntas al cliente.

3.1.7.11 Paso 4 Sprint Review Meeting Sprint 2 (Demo)

Por último en la iteración 2 se realizó la convocatoria a la demostración del producto terminado y comprometido para esta iteración siguiendo los mismos preceptos que en la primera iteración.

3.1.7.12 Paso 5 Retrospectiva Sprint 2

En la retrospectiva del Sprint 2 se pudieron revisar en resumen que las expectativas que se tenía para este Sprint en comparación al anterior se llegaron a cumplir de mejor forma, por lo que se puede sacar como resumen que se cumplió la meta de mejora, cumpliendo los 100 puntos en la iteración y mejorando el rendimiento del equipo.

3.1.7.13 Paso 6 Planificación de la siguiente iteración

De igual manera que en la iteración anterior con las observaciones que se han obtenido de las reuniones de con los dueños del proyecto se realizaron los correspondientes análisis para ver potenciales cambios de las priorizaciones, pero en esta ocasión para la iteración 3 no se obtuvieron cambios en el orden de desarrollo de las tareas, pero si se encontraron varios ajustes que se procederá a realizar en algunas de las funcionalidades para lo cual se realizaran tareas de ingeniería.

3.1.7.14 Paso 7 Lecciones Aprendidas Sprint 2

A continuación se exponen las lecciones aprendidas en la aplicación del nuevo modelo "Scrum + XP" para el Sprint 2 a través de las plantillas mencionada:

Tabla 22. Lecciones Aprendidas Sprint 2

Proyecto	Generación de Tickets de compra					
Fecha Inicio:	06/10/2014	Fecha Fin:	06/11/2014			
Entidad Ejecutora:	Todas la Áreas					
Líder de Proyecto:	Santiago León					
Financiador del Proyecto:						
Equipo:	Luis Serrano, Eduardo Bustillos, David Quishpe, Danilo Toapanta					
Cliente Final:	Maria Elena Romero					
#	Tema	Descripción	Fase del Proyecto		Recomendaciones	
			Proyecto	Implementadas		Obtenidos
1	Organización y priorización	Priorización de tareas	Sprint 2	Reuniones con Stakeholders para determinar la prioridad de las historias dependiendo de la importancia que los usuarios le dan.	Reorganización de los procesos de trabajo con los equipos y revisiones periódicas del trabajo	Realizar este tipo de reuniones de forma periódica para seguir con las mejoras

2	Reorganización de equipo	Reestructuración del equipo de trabajo para funciones de soporte	Sprint 2	Gestión Técnica	Dividir al equipo de desarrollo en 2 partes para trabajar en corrección de bugs y protección del equipo encargado de nuevos desarrollos y	Mejoramiento en el equipo de desarrollo de funcionalidad es mayor	Rotar al equipo de mantenimiento para refrescar conocimientos y no cansar a todos los miembros
---	--------------------------	--	----------	-----------------	---	---	--

Tomado de: (Plantilla Lecciones Aprendidas, 2014)

De la misma se realizó la ejecución de los pasos, fases y etapas mencionados en los sprints 1 y 2 para el resto de Sprints del proyecto que en total sumaron 9 dando como resultado la finalización del mismo en el tiempo estimado que fue de 5 meses más un mes de estabilización que también fue contemplado en la planificación inicial del proyecto, con lo cual se procedió al cierre del proyecto pasando a la última fase del macro de trabajo ágil “**Scrum + XP**” que es la fase de lecciones aprendidas del proyecto y cierre del mismo.

3.1.8 Fase 8 Lecciones aprendidas generales, cierre del proyecto

Para esta fase se procederá a hacer un resumen general de las lecciones aprendidas durante todo el sprint usando las plantillas de cada uno de los sprints, se tomarán las lecciones más importantes que tenga que ver con mejoras en cuanto a la gestión y aquellas técnicas que puedan ser más importantes para ser aplicadas en nuevos proyectos.

El cierre formal del proyecto una vez terminados todos los sprints y realizadas las validaciones de los usuarios con su entera satisfacción se procede a la firma del acta de entrega recepción del producto a los usuarios finales así como también la validación de los manuales de usuarios que son generados en gran parte con las pruebas que los mismos usuarios realizan. Un ejemplo de la plantilla de este documento de cierre se puede ver en el anexo Q de este documento.

3.2 Discusión de Resultados

Una vez aplicado el nuevo marco de trabajo ágil (“Scrum + XP”) a lo largo de todos los sprints del proyecto, la estrategia planteada mostró los siguientes resultados tanto positivos como negativos:

Si bien se describe en este Capítulo el funcionamiento esperado del nuevo marco de trabajo se pudo evidenciar también:

- Con la aplicación de la encuesta se dio conocer que es mucho más fácil la aplicación de este nuevo marco de trabajo en empresas que tienen ya una formación o conocimiento en metodologías ágiles.
Las empresas con un puntaje menor a 60 necesitaran de un mayor trabajo para aplicar este marco u cualquier metodología.
- En un inicio se pudo comprobar que existía mucha dependencia del equipo de trabajo hacia el Scrum Master, para saber qué hacer.
Esto fue un impedimento al inicio de la iteración 1 que se fue solucionando poco a poco y mostro mejorías en la iteración 2.
- A pesar de tener planificadas las tarea, costó al principio a los miembros del equipo comprometerse y responsabilizarse de la culminación de las mismas.
- Se especuló mucho sobre la forma de trabajar en pares entre los miembros del equipo, por lo que fue necesario al comienzo realizar una demostración con experimentación con un par de miembros del equipo quienes cada día reportaban sus experiencias favorables así como las limitaciones. Pero demostrando que es una buena práctica a la final.

Los puntos de mejora que se han visto beneficiados con este marco de trabajo ágil pasan a ser descritos a continuación:

- La encuesta realizada dio como resultado que los miembros de los equipos de trabajo están trabajando en varios proyectos lo que hacer que estén desenfocados y poco concentrados. Se ha logrado que el equipo de trabajo esté enfocado en un determinado proyecto y no en varios a la vez, esto permite que el equipo reaccione y se ajuste a las necesidades de un solo proyecto y esté totalmente concentrado en este trabajo a tiempo completo.

- El establecimiento de políticas y objetivos de calidad son parte fundamental de toda empresa, en este caso se realizaron mejoras a las políticas actuales de la empresa, mismas que ayudarán a mantener un mejor nivel de calidad en el desarrollo de productos de software, además que permitirán crecer a la empresa como tal. Los objetivos de estas políticas se adaptan al mejoramiento señalado en el Manifiesto Ágil y promovido por el nuevo marco de trabajo tal como el aseguramiento de la satisfacción del cliente, el uso de tecnología de vanguardia entre otras.
- La definición clara de roles y responsabilidades en equipos de trabajo, aseguró que todos los miembros entiendan y respeten las normas establecidas, haciendo que el trabajo en equipo se convierta en una herramienta eficiente en este caso, además de permitir que estos equipos se vuelvan multi disciplinarios y auto organizados esto se apreció ya desde el Sprint 1, con algunos problemas pero que se han ido solucionando y se han visto superados en los Sprints finales.
- Al trabajar con el esquema de iteraciones cortas y entregas de código funcional en cada una de estas, se ayuda a que el usuario tenga funcionalidades del proyecto deseado, esto permite en gran medida que los proyectos sean entregados a tiempo y con calidad.
- Si bien es cierto que existen muchas técnicas para el levantamiento de especificaciones, las historias de usuario usadas en el nuevo marco de trabajo ágil permiten tener una forma documentada de lo que se va a ir creando además de darle al usuario la posibilidad de priorizar la importancia de las mismas, así como también ayudaron a que los miembros del equipo se comprometieran en la entrega de resultados finales. En el caso de estudio específicamente el uso de las plantillas que se encuentran en los anexos proporcionaron gran ayuda, sin embargo, estas pueden ser mejoradas, tal es el caso de las platillas de historias de usuario con una región para describir las posibles pruebas

de aceptación y una región para diagramar una posible pantalla o flujo de requerimiento. Así mismo la organización en los board cards en donde se organizó las historias de usuario por tareas ha permitido en las iteraciones 1 y 2 ir cerrando los procesos y verse reflejado en este documento.

- De igual manera con la entrega de software funcional en cada iteración y con la revisión y pruebas realizadas con los usuarios correspondientes se cierra la brecha y se puede medir la satisfacción del cliente. De igual manera esto se pudo probar en los Sprints 3 y 4 trabajados.
- Al tener reuniones periódicas con el equipo de trabajo y los usuarios se logró que los miembros del equipo de desarrollo tengan una visión mucho más amplia del proyecto esto pudo ser comprobado en las reuniones diarias mantenidas en las que una vez que se discutían los estados de las tareas los miembros se ayudaban entre sí.
- Otro de los puntos que se ha visto mejorado es la elección del trabajo por parte de los miembros del equipo de acuerdo a sus aptitudes. A la hora de planificar las tareas, personal del equipo de trabajo especializado en base de datos puede seleccionar las tareas de este tipo para resolverlas con mayor rapidez y agilidad, aparte de mayor satisfacción para cada uno de ellos ya que estaban trabajando en su elemento.
- El hecho de llevar una plantilla de lecciones aprendidas es de gran ayuda puesto que permite registrar errores y soluciones que bien documentadas puede ayudar a otros equipos a no caer en las mismas problemáticas.
- El uso de la reunión de retrospectiva es una de las mejores experiencias, permitieron ayudar al equipo a ver las deficiencias que se dieron en el trabajo de la primera iteración y ver cómo solucionarlos

para las siguientes aparte de que permite medir internamente al ScrumMaster el ambiente de trabajo al finalizar la iteración.

Las ventajas de la aplicación de este marco de trabajo híbrido entre Scrum y XP han sido muchas, por ejemplo el uso de las mejores prácticas de Scrum ha dado como resultado:

- Las expectativas del cliente en el primer Sprint han sido resueltas en la mayoría de los casos, ya que como se describe en este Capítulo se han mantenido en constante comunicación con el mismo y se lo ha involucrado en el proyecto de una forma muy activa tanto en el levantamiento de los requerimientos como en las pruebas del mismo. De modo que cuando el cliente llegó a la reunión de demostración su conocimiento de la primera parte del trabajo realizado era amplio.
- La entrega de resultados anticipados, puesto que de esta manera el cliente está en posición de comenzar a usar partes de su producto antes de que el proyecto se encuentre finalizado. Esto depende mucho de que el cliente este consiente de la prioridad de sus requerimientos y el valor que cada una de las cosas que solicita. Como se tiene entendido esto se realiza al inicio de cada iteración. De esta forma se puede ir midiendo el progreso que vaya teniendo el proyecto. Al finalizar la primera iteración el cliente de este caso de estudio además de tener ya una parte de su software funcionando, conoce cómo trabajar con él debido a su participación en las pruebas.
- Mediante la aplicación de Scrum se ha conseguido la coexistencia y alineamiento entre el equipo de trabajo y los clientes, de esta forma todos los integrantes del proyecto están informados de cuál es la meta que se requiere, hacia donde todos están remando, es decir, no es conocimiento únicamente del Product Owner el objetivo del producto y del equipo de trabajo únicamente conocer cómo hacerlo.

- Al usar las mejores prácticas de Scrum se ha logrado obtener una gestión de desarrollo de proyectos de software mucho más eficaz debido al uso de elementos como reuniones diarias, el uso y revisión de tableros de control de tareas, cuadros de control y toma de decisiones como los Burn Dow, entre otros como se apreció en el caso de estudio de este Capítulo.

El uso de las mejores prácticas de XP por su parte ha ayudado a estemarco de trabajo a conseguir lo siguiente:

- Creación de software con calidad mediante la aplicación de las correspondientes pruebas unitarias, de concepto y de aceptación se ha logrado que en la primera iteración el producto entregable final sea de un alto grado de calidad y adaptado totalmente a las necesidades del cliente y probado por el mismo cliente.
- Permite atender las necesidades del cliente con mayor exactitud. Aunque el uso de tarjetas CRC y plantillas de tareas y metáforas, no son propias y exclusivas de XP se adoptaron en esta combinación y han dado una gran ventaja y ayuda al momento de levantar las historias de usuario de cada uno de los requerimientos.
- Los productos que se obtuvieron con el uso de las mejores prácticas de XP expuestas en este documento han permitido la consecución de productos mucho más robustos debido al uso y diseño de las pruebas realizadas de forma previa en la codificación.
- Definitivamente un gran aporte ha sido el tener una codificación mucho más simple y entendible con lo que se reduce el número de errores gracias al uso de refactoring y como se ha podido ver en este Capítulo también ha sido de mucha ayuda el manejar código estándar.
- El uso de la práctica de programación en pares, si bien es cierto que para muchas personas puede resultar tediosa, ha sido de gran aporte en

este proyecto, ya que ha permitido a los programadores aplicar buenas prácticas al momento de codificar, probar y controlar el código.

El uso por separado de estas metodologías hubieran dado buenos resultados pero, no hubieran complementado cosas muy importantes tal es el caso por ejemplo:

Si se hubiera usado únicamente Scrum en el proyecto, se hubiera podido realizar el trabajo de levantamiento de requerimientos, desarrollo y entrega de producto en iteraciones pero de una forma tradicional, el seguimiento y gestión de desarrollo se llevaría a cabo de una forma controlada, pero la calidad del software tendría sus cuestionamientos ya que no se hubieran tomado las consideraciones de pruebas como se hizo en este caso de estudio.

Si el proyecto se hubiera realizado estrictamente con XP, la calidad de las aplicaciones tendría un mayor grado, la propiedad colectiva del código se manejaría de mejor manera, la programación en pares hubiera permitido trabajar de mejor forma a los programadores, pero la gestión de desarrollo y con el cliente pudiera ser menos eficaz, lo que ocasionaría que existan muchos reprocesos.

La metodología Extreme Programming Clásica (XP) sola, es poco aplicable para proyectos de mediano y gran tamaño, por lo que es importante realizar cambios en la planificación de los proyectos, así como en las fases de análisis y administración de riesgos. Al realizar estos cambios se puede apreciar una leve mejora en la construcción de los productos (Jameel Qureshi, 2011).

El uso combinado de las características expuestas en los capítulos 2 y 3 de este documento referentes a Scrum y XPhan permitido que este marco de trabajo logre tener a parte de una adecuada gestión de desarrollo de productos de software al llevar un control y mejor comunicación tanto internamente como externamente, una gestión de calidad en los productos que se van desarrollando como se puede apreciar a través de estrictos controles mediante

el uso de pruebas del software, refactorización de código propiedad colectiva del mismo, entre otras cosas. Lo que por separado estas 2 metodologías no podrían conseguir, ya que Scrum está más enfocada a la gestión de desarrollo y XP está más orientado a la parte técnica del desarrollo.

4. Conclusiones y Recomendaciones

4.1 Conclusiones

Del Capítulo 1 de este documento se puede concluir lo siguiente:

- Tanto la Metodología XP como Scrum muestran formas paralelas durante cada iteración, mismas que luego se repiten pero agregando valor mediante los cambios que se dan cuando el cliente realiza nuevas priorizaciones a las entregas.
- Por su naturaleza, parecido y complementación la integración de las metodologías XP y Scrum es totalmente viable como se puede ver en el Capítulo 2.
- Se pudo apreciar que Scrum es un marco de trabajo muy fuerte en cuanto al manejo de roles y gestión de procesos pero es algo débil ya en la práctica por algunas de las actividades y herramientas tales como las reuniones diarias de seguimiento, entre otros factores.

Las conclusiones que se pudieron obtener al realizar la investigación en el capítulo 2 son las siguientes:

- El uso de un nuevo marco de trabajo híbrido no resolverá todos los problemas que se presenten durante un proyecto, más bien resulta una herramienta que al ser usada de una manera adecuada puede lograr maximizar los procesos de desarrollo y control de calidad de un determinado producto.
- Durante la elaboración e integración del nuevo marco de trabajo, se logró la identificación de diferentes riesgos que presentan en un proyecto, tales como la mala aplicación de pruebas o la poca

importancia que se le da en algunos proyectos, esto produce que la calidad de los productos se vea totalmente afectada. Otro de los riesgos que se pudo detectar fue la poca claridad en la definición de roles en las empresas, tal es el caso de personas que tienen el rol de tester y también realizan tareas de programación.

- Al momento de realizar la integración se pudieron encontrar muchas tareas, actividades y prácticas similares, al darse esos casos se procedió a adoptar las características, prácticas o actividades de la metodología base.
- En base a lo revisado se tomó como base del nuevo marco de trabajo la metodología Scrum por ser una de las metodologías más usadas en la actualidad, sobre esta base se fueron incluyendo las mejores prácticas, valores y técnicas de XP que aportaron valor al nuevo marco.
- La integración de documentos de Scrum tales como Product Backlog y documentos de XP como metáforas y tarjetas CRC van a aportar mayor detalle al momento de generar historias de usuario y la información necesaria para los requerimientos.
- Partiendo de la descripción lograda entre las 2 metodologías, se procedió a la integración de las fases, actividades, y demás artefactos que se unieron para la formación del nuevo marco de trabajo híbrido.
- El realizar una encuesta para obtener un medidor del estado de la empresa en cuanto a disposición de trabajo y conocimiento de metodologías ágiles fue fundamental para arrancar el caso de estudio.
- El apalancamiento del nuevo marco de trabajo en las políticas internas de la empresa permite que este trabaje en base a lo que la empresa propone como filosofía, esto podrá ser apreciado más adelante.

- La integración de las metodologías antes mencionadas permite contar con procesos livianos y simples, con lo que se puede tener a su vez prácticas y técnicas mucho más adecuadas y completas, esto se puede evidenciar en el complemento de actividades de XP como integración continua, refactoring entre otras en la actividad de Sprint específicamente en la codificación.

Las conclusiones del Capítulo 3 son las siguientes:

- La utilización de documentos tales como historias de usuario, tarjetas CRC, plantillas de lecciones aprendidas, ProductBacklog entre otras han permitido tener un mejor control del proyecto y respaldos a la hora de mantener informado tanto al equipo como a los clientes sobre los avances de los proyectos como ejemplo se puede hablar de las reuniones de retrospectiva mantenidas en el Sprint 1 en donde se detalla lo que se debe hacer y la forma de llevar un demo adecuadamente.
- La revisión de las políticas de calidad de la empresa permitieron adaptar de mejor manera el marco de trabajo a las necesidades de la misma.
- En cuanto a la comunicación, se puede decir con seguridad que esta se convirtió en uno de los pilares fundamentales para que la primera iteración del proyecto se desarrolle y culmine con éxito, a través de las reuniones diarias entre los equipos de trabajo y por su puesto las reuniones de demostración y comunicación con el cliente quien tiene la oportunidad de dar una retroalimentación de sus percepciones.
- De acuerdo a lo descrito a lo largo de este trabajo de tesis, queda comprobado que existe compatibilidad entre las 2 metodologías usadas, y que cada una de las mismas ha logrado aportar una parte fundamental en el nuevo marco de trabajo ágil.

- Con el uso de este nuevo marco de trabajo se puede ayudar en el desarrollo de productos de software de manera mucho más ágil, teniendo un mejor control en la gestión tanto de desarrollo como de calidad de los productos, esto se puede apreciar en el desarrollo del Capítulo 3.
- El uso de este nuevo marco de trabajo, sumado con conceptos de Gestión Ágil, ha permitido un aumento tanto en la calidad del proceso de desarrollo como en el producto en sí si bien en la primera iteración de este trabajo se pudieron dar varios problemas de adaptabilidad, estos poco a poco fueron corregidos en la siguiente iteración.
- El prototipo desarrollado en este caso de estudio deja como resultado y evidencia que el marco de trabajo propuesto cumple a cabalidad con los objetivos y lineamientos propuestos tanto desde la etapa de levantamiento de requerimiento hasta la entrega de la última iteración (9).

Conclusiones Generales del estudio

- La mayoría de las metodologías ágiles a pesar de poseer grandes diferencias, se basan en el uso de características muy parecidas entre ellas, tal es el caso de las usadas para este trabajo Scrum Y XP en las que encontramos similitudes tales como: iteraciones y entregas cortas, comunicación y diseño fuerte, equipos auto organizados entre otras.
- Las metodologías de tipo ágil brindan a los proyectos de desarrollo una mejor integración de todos sus elementos, permitiendo que todo se desarrolle como un conjunto a pesar de que se entregan en cada iteración entidades independientes. Lo que hace que estas se acoplen perfectamente al cambiante mundo actual, tomando así los cambios que se dan como una ventaja competitiva y como una oportunidad de mejora constante para los miembros que aplican este tipo de metodologías.

- En la actualidad el uso de Scrum es muy alto a nivel mundial para el desarrollo de proyectos de software, sin fijarse en el tamaño o alcance del proyecto, esto se debe a que Scrum es una de las metodologías más flexibles y adaptables a los proyectos.
- El uso de Scrum permite tener una adecuada Gestión de Riesgos, debido a que no es un proceso independiente, sino que está inmerso durante todo el proyecto en cada etapa que se va planificando y ejecutando.
- En la mayoría de las metodologías de tipo ágil los equipos de trabajo son capaces de tomar decisiones de forma rápida y oportuna lo que permite que se trabaje de mejor forma. Además en estas metodologías prima la comunicación tanto a la interna como a la externa del proyecto.
- Las prácticas de Scrum, usadas en este proyecto han permitido tener mejor control del tiempo y entrega de valor al cliente, por ejemplo en las entregas cortas de productos funcionales.
- Si bien el uso de Programación en Pares resultó ser algo nuevo e incómodo para algunos de los programadores, su adaptación es rápida y con excelentes resultados.

4.2 Recomendaciones

- Es recomendable la aplicación de este marco de trabajo con más proyectos para así tener un mayor grado de comparación en cuanto a sus beneficios o falencias.
- Con el nuevo marco de trabajo híbrido se aprecia que existe mayor control en los roles y funciones que cada uno de los miembros de los equipos realizan se recomienda la especialización de los mismos en tareas o áreas más específicas tales como arquitectura, pruebas y QA, gestión con el

personal, puesto que ayudaría a tener mayor grado de profesionalismo y atención a los clientes.

- Se recomienda la aplicación y uso de pruebas automáticas de software, mismas que podría ayudar a tener mayor calidad en los productos ayudando al personal de pruebas a tener mayor visión de errores y posibles fallos.
- En este marco de trabajo están inmersas la Gestión de Riesgos y de Calidad pero a breves rasgos, por lo que es recomendable tomar actividades referentes a estas áreas que sirvan de apoyo al nuevo marco y así le den mayor robustez.
- La propuesta que se ha dado en este trabajo es recomendable para uso en proyectos de cualquier tamaño, siempre y cuando su desarrollo esté dado bajo los preceptos de metodologías ágiles.
- La motivación de los equipos de trabajo es fundamental para el éxito de estas metodologías, por lo que es totalmente recomendable mantener elevada la moral de los equipos, con actualizaciones, cursos, promociones que ayuden a su desarrollo profesional y personal, lo que da como resultado mejoras para la Empresa.
- El conocimiento y participación constante de la Gerencia en los proyectos es imprescindible, por lo que es recomendable que la Dirección este en constante seguimiento del desarrollo que tenga cada uno de los proyectos en los que se vea involucrado este nuevo marco de trabajo, y de igual forma participar de forma constante, tal como lo realizan los clientes en mejoras y priorizaciones para ayudar a mejorar el rendimiento de los equipos ágiles.
- La adopción del nuevo marco de trabajo al principio puede mostrar gran cantidad de falencias y no es raro que haya renuencia al cambio y se quiera volver a las metodologías antiguas, sin embargo, es aconsejable no cambiar

y continuar con el proceso, ya que con la introducción de estas nuevas prácticas se apoya a la Organización, además de brindar una mejor experiencia documentada para futuras adopciones en otros proyectos y equipos.

Referencias

- Anex, A. (03 de 05 de 2013). *Gerencia*. Recuperado el 21 de 11 de 2014, de Gerencia: <http://www.emb.cl/gerencia/articulo.mvc?xid=1275>
- Autor. (s.f.).
- Bahit, E. (2012). *Scrum y eXtreme Programming para Programadores*. Buenos Aires , Argentina.
- Baskerville , R. (2006). Business agility and diffusion of information. *European Journal of Information Systems*(15). Recuperado el 20 de 11 de 2014, de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.296.5817&rep=rep1&type=pdf>
- Becerra Castro, C. R. (s.f). *Propiedad Colectiva del Código*. Recuperado el 13 de 01 de 2015, de Propiedad Colectiva del Código: <http://www.inf.utfsm.cl/~visconti/inf321/documentos/trab6.pdf>
- Buhler, E. (2013). *Agile Iberoamérica*. Recuperado el 13 de Diciembre de 2014, de 15 preguntas antes de implementar agile/scrum en la empresa: <http://agilib.org/2013/02/08/las-15-preguntas-antes-de-implementar-agilescrum-en-la-empresa/>
- Cifuentes Lozano, A. (2012). *Integración de Buenas Prácticas*. Recuperado el 24 de Noviembre de 2014, de Integración de Buenas Prácticas: http://bibliotecadigital.icesi.edu.co/biblioteca_digital/bitstream/10906/68430/5/modelo_integracion_practicas.pdf.txt
- Código En Línea*. (2008). Recuperado el 08 de Enero de 2015, de Código En Línea: <http://codigolinea.com/2008/05/25/estilo-de-programacion-y-convencion-de-nombres-ii/>
- Cuestionario para evaluar la situación de la empresa*. (s.f.). Recuperado el 30 de Noviembre de 2014, de <http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&sqi=2&ved=0CBwQFjAA&url=http%3A%2F%2Fplanificacionygestiondeportiva.es%2Fresources%2FCuestionario%2Bpara%2Bevaluar%2Bla%2Bsituaci%24C3%24B3n%2Bde%2Bla%2Bempresa.doc&ei=fJlwVc>

Cuestionario para evaluar la situación de la empresa respecto al Sistema de Gestión de la Calidad ISO 9000:2000. (s.f.). Recuperado el 20 de Noviembre de 2014, de <http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CBwQFjAA&url=http%3A%2F%2Fplanificacionygestiondeportiva.es%2Fresources%2FCuestionario%2Bpara%2Bevaluar%2Bla%2Bsituaci%24C3%24B3n%2Bde%2Bla%2Bempresa.doc&ei=XXYsVYLwFKvj>

Deemer, P., Benefield, G., Larman, C., & Bas, V. (2009). *The Scrum Primer Versión en Español*. San Francisco, California, Estados Unidos.

Formato ejemplo para documentar el uso de scrum. (20 de 12 de 2014).

Recuperado el 05 de Febrero de 2015, de

http://www.navegapolis.com/files/formato_ejemplo_para_documentar_el_uso_de_scrum_en_un_proyecto.doc

Gestión de Proyectos. (2012). Obtenido de Gestión de Proyectos:

<http://es.slideshare.net/rcelemin/administracin-de-proyectos-40697565>

Hayata, T., & Han, J. (2011). *A Hybrid Model for IT Project with Scrum.*

Recuperado el 30 de Enero de 2015, de

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5986572&queryText%3Dscrum>

Henrik, K., & Reza, F. (2008). Bootstrapping Scrum and XP under crisis A story from the trenches. Recuperado el 30 de Enero de 2015, de

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4599519&pageNumber%3D6%26queryText%3Dscrum>, Consultada el 30 de enero del 2015

Jameel Qureshi, M. R. (2011). Agile software development methodology for medium and large projects. *ietdl*. Recuperado el 30 de Enero de 2015, de

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6322858&queryText%3DExtreme+Programming+.LB.XP.RB>

javiergarzas.com. (2 de 04 de 2012). Recuperado el 20 de Diciembre de 2014, de <http://www.javiergarzas.com/2012/04/herramientas-scrum.html>

- Jeffries, R., Anderson, A., & Hendrickson, C. (2006). *Extreme Programming Installed* (Vol. 1). Addison-Wesley Professional. Recuperado el 16 de 01 de 2015, de <http://homes.di.unimi.it/~belletc/corsi/xpinstall.pdf>
- Jiménez y Orantes . (2012). Metodología Híbrida para Desarrollo de Software en México. *Revista Unam.MX*. Recuperado el 25 de Febrero de 2015, de <http://www.revista.unam.mx/vol.13/num1/art16/>
- Kick Off Meeting*. (2012). Recuperado el 2 de Enero de 2015, de Qué es Kick Off Meeting: <http://www.ilunik.com/kick-off-meeting>
- Kniberg, Sutherland y Cohn. (2007). InfoQ.com.
- La Oficina de Proyectos de Informática*. (2014). Recuperado el 09 de Diciembre de 2014, de La Oficina de Proyectos de Informática: <https://sites.google.com/site/oficinaproyectosinformatica/home/archivo/TarjetasTareas>
- Lacey , M. (01 de 01 de 2014). *Mitch Lacey & Associates, INC*. Recuperado el 20 de Noviembre de 2014, de <https://www.mitchlacey.com/intro-to-agile/scrum/the-product-backlog>
- Letelier, P., Penadés, M. C., & Canós, J. (s.f). Metodologías ágiles para el desarrollo de Software Extreme Programming (XP). Valencia, España. Recuperado el 21 de Noviembre de 2014
- Manifiesto Agil*. (2001). Recuperado el 01 de Noviembre de 2014, de <http://agilemanifesto.org/iso/es>
- Manrubia Diez, J. (01 de 06 de 2009). Metodologías Ágiles: Scrum y Técnicas de Estimación ágil. *Gerencia de Informática de la Seguridad Social*. Madrid, España. Recuperado el 24 de 11 de 2014, de <http://jorgemanrubia.net/blog/wp-content/uploads/2009/06/2009-06-CharlaPreparaticAgil.pdf>
- Microsoft*. (2015). Recuperado el 4 de Enero de 2015, de Sobre TDD con Visual Studio: <http://msdn.microsoft.com/es-es/library/jj573876.aspx>
- Microsoft*. (2015). Recuperado el 17 de Enero de 2015, de Refactorización: <http://msdn.microsoft.com/es-es/library/719exd8s.aspx>
- Mukhtar, M., Motla, Y. H., Riaz, M., & Khand, M. (2013). A Hybrid Model for Agile Practices Using Case Based Reasoning. Recuperado el 30 de Enero de 2015, de

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6615431&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6615431

Mundra, A., Misra, S., & Dhawale, C. A. (2013). *Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software*. Recuperado el 30 de Enero de 2015, de Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software:

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6681108&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6681108

Oficina y Bienestar. (s.f.). Recuperado el 14 de Diciembre de 2014, de Oficina y Bienestar: <http://oficinaybienestar.com/n/5081/tecnicas-de-integracion-de-equipos-de-trabajo.html>

Palacio, J. (2014). *Gestión de proyectos Scrum Manager*. Recuperado el 29 de Noviembre de 2014

Pérez A., O. A. (1 de Abril de 2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP- MSF- XP- Scrum.

PMOinformatica.com. (03 de Diciembre de 2012). Recuperado el 26 de Noviembre de 2014, de <http://www.pmoinformatica.com/2012/12/test-driven-development-ventajas-y.html>

PMOInformatica.com. (2014). Recuperado el 06 de Diciembre de 2014, de <http://www.pmoinformatica.com/2013/07/plantilla-matriz-raci-asignacion.html>

PMOInformatica.com. (2014). Recuperado el 27 de Diciembre de 2014, de Plantilla Pila Producto: <http://www.pmoinformatica.com/2013/11/plantillas-scrum-pila-producto-product.html>

PMOInformatica.com. (2014). Recuperado el 07 de Diciembre de 2014, de Gestione sus proyectos Plantilla de Lecciones Aprendidas: <https://sites.google.com/site/oficinaproyectosinformatica/home/archivo/Plantilla%20Registro%20de%20Lecciones%20Aprendidas.xls?attredirects=0&d=1>

- PMOInformatica.com.* (2014). Recuperado el 26 de Diciembre de 2014, de Plantillas de Historias de Usuario:
<https://sites.google.com/site/oficinaproyectosinformatica/home/archivo/Plantilla%20Historias%20Usuario%20-%20Oficina%20de%20Proyectos%20de%20Informatica.xls?attredirects=0&d=1>, 2014
- PMOinformatica.com.* (s.f). Recuperado el 29 de Noviembre de 2014, de PMOinformatica.com: <http://www.pmoinformatica.com/2012/09/test-driven-development-scrum.html>
- Programación en parejas.* (05 de 01 de 2009). Recuperado el 18 de Noviembre de 2014, de <http://www.dosideas.com/noticias/metodologias/371-programacion-en-parejas-sp-1687791370.html>
- Proyectos ágiles.* (s.f.). Recuperado el 04 de Diciembre de 2014, de Proyectos ágiles: <http://www.proyectosagiles.org/beneficios-de-scrum>
- Proyectos ágiles Como Gestionar proyectos con Scrum.* (s.f). Recuperado el 21 de Noviembre de 2014, de Proyectos ágiles Como Gestionar proyectos con Scrum: <http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning>
- Proyectos ágiles.* (s.f). Recuperado el 16 de Noviembre de 2014, de Proyectos ágiles: <http://www.proyectosagiles.org/beneficios-de-scrum>
- Proyectos Ágiles.* (s.f). Recuperado el 02 de Febrero de 2015, de <http://www.proyectosagiles.org/requisitos-de-scrum#tamano-equipo>
- Proyectos ágiles.* (s.f.). Recuperado el 18 de Noviembre de 2014, de Proyectos ágiles: <http://www.proyectosagiles.org/requisitos-de-scrum>
- Proyectos ágiles.* (s.f.). (L. d.-S. Backlog, Productor) Recuperado el 25 de Noviembre de 2014, de Proyectos ágiles:
<http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog>
- Pruebas de Integración.* (s.f.). Recuperado el 16 de Noviembre de 2014, de Academica: <http://www.academica.mx/blogs/las-pruebas-integraci%C3%B3n-software>
- Qasaimeh, M., & Abran, A. (2010). Investigation of the Capability of XP to Support the Requirements of ISO 9001 Software Process Certification. Recuperado el 30 de Enero de 2015, de

<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5489840&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5489041%2F5489073%2F05489840.pdf%3Farnumber%3D5489840>

Schwaber, K. (2004). *Agile Project Management with Scrum*.

Scrum Manager BoK. (13 de 08 de 2013). *Scrum Manager Body of Knowledge*.

Recuperado el 07 de Febrero de 2015, de

http://www.scrummanager.net/bok/index.php?title=Gr%C3%A1fico_de_producto

Scrum Manager BoK. (10 de 08 de 2014). *Scrum Manager Body of Knowledge*.

Recuperado el 04 de Diciembre de 2014, de

http://www.scrummanager.net/bok/index.php?title=Velocidad,_trabajo_y_tiempo

SCRUMstudy. (2013). Scrum Manager Body of Knowledge. Recuperado el 21 de Diciembre de 2014, de Gráfico de Producto:

http://www.scrummanager.net/bok/index.php?title=Gr%C3%A1fico_de_producto

Sultana, S., Motla, Y. H., Asghar, S., Jamal, M., & Azad, R. (s.f). A Hybrid Model by Integrating Agile Practices for Pakistani Software Industry.

Recuperado el 30 de Enero de 2015, de

<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6808600&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel7%2F6802612%2F6808546%2F06808600.pdf%3Farnumber%3D6808600>

Tarjetas CRC. (2014). Recuperado el 26 de Diciembre de 2014, de Tarjetas CRC:

<http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=0CDAQFjAF&url=http%3A%2F%2Fuagrm-components-design.googlecode.com%2Fsvn%2Ftrunk%2Fmodule7%2Fsistema%2520documentacion%2Fplantillas%2Fplantilla%2520tarjeta%2520CRC%2520-%2>

Tarjetas de metáforas. (2014). Recuperado el 30 de Noviembre de 2014, de

Tarjetas de metáforas:

www.inf.utfsm.cl/~visconti/xp/Identificacion_Metafora_2.doc

The Standish Group International Incorporated. (2013). *THE CHAOS MANIFESTO*. Boston.

Wood, L. (2010). *Sokets and lightbulbs*. Recuperado el 01 de Diciembre de 2014, de Sokets and lightbulbs:

<http://blog.socketsandlightbulbs.com/2012/02/19/the-best-scrum-ban-plan-board-ever/>

YOUR CENTRAL LOCATION FOR ALL RALLY AND AGILE HELP

RESOURCES. (s.f.). Recuperado el 02 de 12 de 2014, de YOUR

CENTRAL LOCATION FOR ALL RALLY AND AGILE HELP

RESOURCES: <https://help.rallydev.com/progress-reports>

Zarazaga Soria, F., & Nogueras Iso, J. (2008). *Introducción a la Ingeniería de Software*. Informático Académico, Universidad de Zaragoza, Zaragoza.

Glosario de términos

Agile Testing: Enfoque de pruebas de software en los procesos ágiles de desarrollo, basados en el cuadrante de Brian Marick.

Buena práctica: una forma de hacer las cosas que ha demostrado ser correcta en la práctica.

CMMI: Capability Maturity Integration

CPI, Cost Performance Indicator: Es el índice de rendimiento del costo en la teoría del valor ganado. Indica la eficiencia en el uso del presupuesto de un proyecto. Un valor mayor a 1 indica que el proyecto se ha salido del presupuesto, un valor menor a 1 indica que el proyecto está dentro del presupuesto.

CSI, Continuous Service Improvement: Proceso de mejora continua del servicio de 7 pasos propuesto por ITIL.

Dashboard: El nombre en inglés del tablero de instrumentos localizado debajo del parabrisas en los automóviles. Para nuestro caso se hace referencia al siguiente concepto: Herramienta de BI para la toma de decisiones empresariales en tiempo real.

Extreme Programming (Xp): Es un conjunto de prácticas de ingeniería destinadas al mantenimiento de un alto nivel de calidad interna lo que da como resultado mantener una velocidad de desarrollo de productos de software sostenible.

Feedback: Devolución de una señal modificada a su emisor

ISO: Siglas de la expresión inglesa International for Standardization, 'Organización Internacional de Estandarización', sistema de normalización internacional para productos de áreas diversas.

Iteración: Es un espacio de tiempo en el cual el equipo implementa un incremento al producto con la calidad necesaria para ser liberado al cliente.

Kick Off.- El kick off es un encuentro entre los responsables de la empresa y el cliente (responsables del nuevo proyecto) en el que se habla sobre todo lo que tenga que ver con el nuevo proyecto. Este término también está asociado a reuniones internas de empresa, asociadas a nuevas etapas, donde todos los trabajadores hablan y dan su opinión. Ambos encuentros deben resolver aspectos relacionados con el nuevo proyecto o nueva etapa, tanto conocidos como futuros (Kick Off Meeting, 2012).

Mejora continua: El proceso de mejora continua es un concepto del siglo XX que pretende mejorar los productos y procesos

POO: Programación Orientada a Objetos

PYMES: Las pequeñas y medianas empresas son entidades independientes, con una alta predominancia en el mercado de comercio, quedando prácticamente excluidas del mercado industrial por las grandes inversiones

ProductOwner: Ser el representante de todas las personas interesadas en los resultados del proyecto (internas o externas a la organización, promotores del proyecto y usuarios finales)

Quality Assurance (QA): se refiere a un conjunto de reglas, prácticas y preceptos que se realizan para asegurar que el producto o servicio que se entrega al usuario final o cliente, tenga la calidad adecuada y la mejor posible

RTF: Revisiones Técnicas Formales

Refactoring - Refactorizar: La refactorización es el proceso que consiste en mejorar el código una vez escrito cambiando su estructura interna sin modificar su comportamiento externo (Microsoft, 2015).

SEI: Software Engineering Institute

Scrum: Es un pseudo proceso ágil de desarrollo de software que sigue los enunciados del manifiesto ágil y sus principios derivados.

Sprint: Ver Iteración en este glosario.

Standish Group: El informe CHAOS, que realiza el Standish Group, es el informe más famoso sobre el fracaso de los proyectos en el sector de las tecnologías.

Storypoint: Es una unidad de medida del esfuerzo, incertidumbre y riesgo que tomará desarrollar una historia de usuario.

Tester: Rol técnico dedicado a la ejecución de pruebas de software para detección y reporte de fallas.

Time to market: la sincronización del plazo de comercialización con las diferentes fuerzas impulsoras.

TI: Tecnologías de la información

ANEXOS

Anexo A. Plantilla de Identificador de Metáfora

Identificación de la Metáfora	
[[Nombre del Proyecto]]	
Metáfora del Sistema	
Información de aprobación de la Metáfora	
Firma del Entrenador (Coach)	Firma del cliente

Tomado de: (Tarjetas de metáforas, 2014)

Anexo B Plantilla de Product Backlog

Desarrollo ágil: Pila de Producto

(Product Backlog)

Elaborado por:

Identificador (ID) de la Historia	Enunciado de la Historia	Alias	Estado	Dimensión / Esfuerzo	Iteración (Sprint)	Prioridad	Comentarios
XX-XXXX-XXXX	Como un [Rol], necesito [descripción de la funcionalidad], con la finalidad de [Razón o Resultado]						
XX-XXXX-XXXX							
XX-XXXX-XXXX							

Tomado de: (Plantilla Pila Producto, 2014)

Anexo C. Plantilla de Historia de Usuario

Historia de Usuario	
Número: 1	Nombre Historia de Usuario:
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Encargados de las bodegas	
Prioridad en Negocio: Alta	
Riesgo en Desarrollo: Alta	
Descripción: Permite realizar el pedido de surtido de las bodegas de las zapaterías "La ribera" .	
Observaciones: Actualmente dentro de las bodegas la información referente a esta es manejada a mano con un formato de surtido de calzado .	

Tomado de: (Plantillas Historias Usuario, 2014)

Anexo D. Plantilla de Tarjeta CRC

Clase	
Responsabilidad 1	Colaborador 1
.	.
.	.
Responsabilidad n	Colaborador n
Comentario	

Tomado de: (Tarjetas CRC, 2014)

Anexo E. Plantilla de Task Card

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario (Nro. y Nombre):
Nombre Tarea:	
Tiempo Estimado	
Tipo de Tarea :	Puntos Estimados:
Desarrollo / Corrección / Mejora / Otra (especificar)	
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

Tomado de: (Plantillas Tarjeta de Tareas, 2014)

Anexo F. Encuesta de Factibilidad

UNIVERSIDAD DE LAS AMERICAS

MAESTRIA EN GERENCIA DE SISTEMAS

La presente encuesta es parte del trabajo de titulación de la carrera de Maestría en Gerencia de Sistemas de la Universidad de las Américas. La información consignada en este documento será mantenida confidencialmente y únicamente se utilizarán los datos estadísticos totales con fines académicos.

Datos de la Empresa Encuestada

Nombre de la Empresa: _____ **Ciudad:** _____

Persona que llena la encuesta: _____ **Cargo:** _____

Email de contacto: _____ (opcional) **Fecha:** _____

Número de empleados de su empresa?

1 – 5	
6 – 10	
11 – 20	
21 – 49	
50 o más	

Cuestionario

1. ¿Los Ingenieros/Programadores, trabajan en varios proyectos en forma simultánea a tiempo exclusivo?

	Nunca
	Casi Nunca
	A veces
	Casi Siempre
	Siempre

2. ¿Cuál considera que es el nivel de aplicación de metodologías de gestión de desarrollo en su empresa?

	Muy Bajo
	Bajo
	Regular
	Aceptable
	Muy Alto

3. ¿Está establecida la política de la calidad y los objetivos de la calidad?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

4. ¿Están definidas las responsabilidades y autoridad entre ellas la función de calidad?

	Prácticamente no están
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

5. ¿Los roles están bien definidos y limitan o aumentan las posibilidades de las personas?

	Prácticamente no están
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

6. ¿Los proyectos anteriores han sido entregados en tiempo y forma o han sufrido retrasos?

	Prácticamente no se entregan
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

7. ¿Asegura la dirección la disponibilidad de los recursos necesarios: Humanos, instalaciones y equipos?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

8. ¿Se tienen identificados los requisitos de los clientes tanto los especificados por ellos como los no especificados, así como los requisitos legales y reglamentarios?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

9. ¿Se revisan los requisitos del producto o servicio antes de adquirir un compromiso con el cliente?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

10. ¿El cliente final está contento con el producto?

	Prácticamente no
	Parcialmente (en ocasiones puntuales)
	Generalmente (en la mayoría de los casos)
	Sistemáticamente y en casi todas los caso
	Siempre y de forma total, y somos un ejemplo para el sector.

11. ¿Se realizan planes para el personal (admisión, formación, desarrollo, etc.) evaluando el rendimiento y las necesidades de desarrollo de todas las personas?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

12. ¿Es prioridad satisfacer al cliente mediante la entrega temprana y continua de software con valor?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

13. ¿Se entrega software funcional frecuentemente?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

14. ¿Se acepta que los requerimientos cambien incluso en etapas tardías del desarrollo?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

15. ¿Los proyectos se desarrollan en torno a individuos motivados?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

16. ¿El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

17. ¿El software funcionando es la medida principal de progreso?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

18. ¿Los promotores, desarrolladores y usuarios son capaces de mantener un ritmo constante de forma indefinida?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

19. ¿Se analizan e interpretan las métricas y datos obtenidos?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

20. ¿El equipo de desarrollo y jefes de producto están físicamente en la misma área?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

21. ¿Los desarrolladores conocen la visión completa del producto?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

22. ¿Las decisiones de cambio son discutidas con el equipo de desarrollo o tomadas por la parte gerencial?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

23. ¿Existe rotación de las personas y roles o cada individuo trabaja en su especialidad?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

24. ¿Las personas pueden elegir su trabajo o tareas basadas en sus conocimientos o les son asignadas?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

25. ¿Su empresa Realiza pruebas de software?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

26. ¿Existe un programa de mejora continua que afecta a todas las actividades de la empresa empleando herramientas adecuadas y estableciendo objetivos de mejora?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

27. ¿Cuáles considera que son las principales dificultades en la aplicación de estándares de control de calidad?

	Falta de entrenamiento del personal
	Falta de apoyo a nivel de consultoría externa
	Falta de tiempo
	Falta de presupuesto
	No se considera necesario

28. ¿Se controlan las no conformidades y se asegura que el producto no conforme es identificado y controlado para prevenir una utilización o entrega no intencionada?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

	1	2	3	4	5
Total Marcados					
Totales multiplicados por puntos	*1	*2	*3	*4	*5
Suma total de Puntos					

Adaptado de: (Cuestionario para evaluar la situación de la empresa)

Anexo G. Plantilla de Lecciones Aprendidas en un Proyecto

Lecciones Aprendidas						
Proyecto						
Fecha Inicio:		Fecha Fin:				
Ejecutora:						
Proyecto:						
del Proyecto:						
Equipo:						
Cliente Final:						
#	Tema	Descripción	Fase del Proyecto		Acciones Realizadas	
			Implementadas	Obtenidos	Recomendaciones	

Tomado de (Plantilla Lecciones Aprendidas, 2014)

Anexo H. Acta de entrega recepción del proyecto

**Acta de Cierre de
Proyecto o Fase**
[Nombre del Proyecto]
[Fase del Proyecto]
Fecha:[dd/mm/aaa]

Información del Proyecto

Datos

Empresa / Organización	
Proyecto	
Fecha de preparación	
Cliente	
Patrocinador principal	
Gerente de Proyecto	

Patrocinador / Patrocinadores

Nombre	Cargo	Departamento / División	Rama ejecutiva (Vicepresidencia)

Razón de cierre

<En la siguiente lista se certifica las razones del cierre del proyecto o fase, específicamente si se entregó todos los componentes del producto, si algunos componentes fueron entregados y otros cancelados, o si se cancelaron todos los entregables>

Por medio de la presente, se da cierre formal al proyecto, por las razones especificadas en la siguiente ficha:

Marcar con una "X" la razón de cierre:

Entrega de todos los productos de conformidad con los requerimientos del cliente.	
Entrega parcial de productos y cancelación de otros de conformidad con los requerimientos del cliente.	
Cancelación de todos los productos asociados con el proyecto.	

Aceptación de los productos o entregables

A continuación se establece cuales entregables de proyecto han sido aceptados:

Entregable	Aceptación (Si o No)	Observaciones

<El cuadro se completa haciendo referencia a las entregables, que pueden ser documentos o componentes del producto>

Para cada entregable aceptado, se da por entendido que:

- El entregable ha cumplido los criterios de aceptación establecidos en la documentación de requerimientos y definición de alcance.
- Se ha verificado que los entregables cumplen los requerimientos.
- Se ha validado el cumplimiento de los requerimientos funcionales y de calidad definidos.
- Se ha realizado la transferencia de conocimientos y control al área operativa.
- Se ha concluido el entrenamiento que se definió necesario.
- Se ha entregado la documentación al área operativa.

Se autoriza al Gerente de Proyecto a continuar con el cierre formal del proyecto o fase, lo cual deberá incluir:

- Evaluación post-proyecto o fase.
- Documentación de lecciones aprendidas.
- Liberación del equipo de trabajo para su reasignación.
- Cierre de todos los procesos de procura y contratación con terceros.
- Archivo de la documentación del proyecto.

Una vez concluido el proceso de cierre, el Patrocinador (Sponsor) del proyecto deberá ser notificado para que el Gerente de Proyectos sea liberado y reasignado.

Aprobaciones

Patrocinador	Fecha	Firma

Anexo I. Encuesta aplicada a la empresa XYZ

UNIVERSIDAD DE LAS AMERICAS

MAESTRIA EN GERENCIA DE SISTEMAS

La presente encuesta es parte del trabajo de titulación de la carrera de Maestría en Gerencia de Sistemas de la Universidad de las Américas. La información consignada en este documento será mantenida confidencialmente y únicamente se utilizarán los datos estadísticos totales con fines académicos.

Datos de la Empresa Encuestada

Nombre de la Empresa: XYZ _____ **Ciudad:** Quito _____

Persona que llena la encuesta: Roberto Toapanta_ **Cargo:** Arquitecto de aplicaciones _____

Email de contacto: _____ (opcional) **Fecha:** 24/12/2014 _____

Número de empleados de su empresa?

1 – 5	
6 – 10	
11 – 20	
21 – 49	X
50 o más	

Cuestionario

1. ¿Los Ingenieros/Programadores, trabajan en varios proyectos en forma simultánea a tiempo exclusivo?

	Nunca
	Casi Nunca
X	A veces
	Casi Siempre
	Siempre

2. ¿Cuál considera que es el nivel de aplicación de metodologías de gestión de desarrollo en su empresa?

	Muy Bajo
	Bajo
X	Regular
	Aceptable
	Muy Alto

3. ¿Está establecida la política de la calidad y los objetivos de la calidad?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

4. ¿Están definidas las responsabilidades y autoridad entre ellas la función de calidad?

	Prácticamente no están
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

5. ¿Los roles están bien definidos y limitan o aumentan las posibilidades de las personas?

	Prácticamente no están
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

6. ¿Los proyectos anteriores han sido entregados en tiempo y forma o han sufrido retrasos?

	Prácticamente no se entregan
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

7. ¿Asegura la dirección la disponibilidad de los recursos necesarios: Humanos, instalaciones y equipos?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

8. ¿Se tienen identificados los requisitos de los clientes tanto los especificados por ellos como los no especificados, así como los requisitos legales y reglamentarios?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

9. ¿Se revisan los requisitos del producto o servicio antes de adquirir un compromiso con el cliente?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

10. ¿El cliente final está contento con el producto?

	Prácticamente no
	Parcialmente (en ocasiones puntuales)
X	Generalmente (en la mayoría de los casos)
	Sistemáticamente y en casi todas los caso
	Siempre y de forma total, y somos un ejemplo para el sector.

11. ¿Se realizan planes para el personal (admisión, formación, desarrollo, etc.) evaluando el rendimiento y las necesidades de desarrollo de todas las personas?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

12. ¿Es prioridad satisfacer al cliente mediante la entrega temprana y continua de software con valor?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

13. ¿Se entrega software funcional frecuentemente?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

14. ¿Se acepta que los requerimientos cambien incluso en etapas tardías del desarrollo?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

15. ¿Los proyectos se desarrollan en torno a individuos motivados?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

16. ¿El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

17. ¿El software funcionando es la medida principal de progreso?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

18. ¿Los promotores, desarrolladores y usuarios son capaces de mantener un ritmo constante de forma indefinida?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

19. ¿Se analizan e interpretan las métricas y datos obtenidos?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

20. ¿El equipo de desarrollo y jefes de producto están físicamente en la misma área?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

21. ¿Los desarrolladores conocen la visión completa del producto?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

22. ¿Las decisiones de cambio son discutidas con el equipo de desarrollo o tomadas por la parte gerencial?

	Prácticamente no se realiza
	Se realiza parcialmente (en ocasiones puntuales)
X	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

23. ¿Existe rotación de las personas y roles o cada individuo trabaja en su especialidad?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

24. ¿Las personas pueden elegir su trabajo o tareas basadas en sus conocimientos o les son asignadas?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

25. ¿Su empresa Realiza pruebas de software?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

26. ¿Existe un programa de mejora continua que afecta a todas las actividades de la empresa empleando herramientas adecuadas y estableciendo objetivos de mejora?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

27. ¿Cuáles considera que son las principales dificultades en la aplicación de estándares de control de calidad?

	Falta de entrenamiento del personal
	Falta de apoyo a nivel de consultoría externa
X	Falta de tiempo
	Falta de presupuesto
	No se considera necesario

28. ¿Se controlan las no conformidades y se asegura que el producto no conforme es identificado y controlado para prevenir una utilización o entrega no intencionada?

	Prácticamente no se realiza
X	Se realiza parcialmente (en ocasiones puntuales)
	Se realiza generalmente (en la mayoría de los casos)
	Se realiza sistemáticamente y en casi todas las áreas
	Se realiza siempre y de forma total, y somos un ejemplo para el sector.

	1	2	3	4	5
Total Marcados		11	16		
Totales multiplicados por puntos	*1	*2	*3	*4	*5
Suma total de Puntos	70				

Anexo J. Historias de usuario Caso de Estudio

Historia de Usuario	
Iteración:1	
Número: 1	Nombre Historia de Usuario:Generación de Ticket de compra
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Todos los usuarios	
Prioridad en Negocio: Alta	
Riesgo en Desarrollo: Alta	
<p>Descripción:</p> <p>Como usuario de la aplicación, quiero Generar un Ticket de compra, entonces:</p> <p>Al ingresar a la pantalla el usuario debe visualizar la siguiente información: usuario, oficina, área y fecha, ésta información debe ser heredada de los parámetros iniciales del sistema, es decir información presentada de acuerdo al usuario que está logueado.</p> <p>Es obligatorio que se registrarse en el sistema el login de la persona que realizó el pedido y la fecha y hora de proceso.</p> <p>Debe existir un campo Nro. Ticket, en el cual se debe generar un número secuencial de ticket que debe iniciar con las siglas del departamento que genera el ticket por ejemplo Contabilidad CON y el secuencial unidos por un guion (sin ceros). Este número debe ser generado automáticamente por el sistema.</p> <p>Se debe ingresar los detalles del pedido, toda esta información deberá presentarse en un grid, en el cual se podrá modificar, y eliminar los registros de acuerdo a las necesidades del usuario.</p> <p>Además para el ingreso de la información se debe presentar una pantalla adicional sobre puesta (popup)</p> <p>Debe existir la posibilidad de atar archivos adjuntos como pdf, sin ser obligatorios.</p> <p>Esta información debe estar en capacidad de ser grabada y a la vez enviada a un esquema de seguimiento del ticket (WorckFlow) para que sea aprobada y revisada por otros usuarios del sistema.</p> <p>Al realizar esta funcionalidad de seguimiento, se debe notificar a cada usuario que debe atender el requerimiento en curso.</p> <p>El flujo de aprobación de ésta primera pantalla deberá ser la siguiente: viajar a la estación del jefe de área u oficina y luego al gerente de área. Cada uno deberá aprobar o rechazar la solicitud enviada. A continuación el pedido se direccionará al área de adquisiciones.</p> <p>Cada nivel de aprobación debe registrar su observación ya sea de aprobación o rechazo.</p> <p>Si se rechaza se enviará un correo a la persona que realizó el requerimiento, desde cualquier nivel que se rechace.</p>	

Al finalizar el flujo en el nivel del gerente de área se debe enviar un correo a la persona que realizó el requerimiento.
<p>Observaciones:</p> <p>Esta pantalla deberá presentar la información la siguiente información:</p> <p>Pantalla de Adquisiciones</p> <ul style="list-style-type: none"> - Usuario: Se debe especificar el login del usuario. - Área: Área a la que pertenece el usuario. - Oficina: Oficina a la que pertenece el usuario. - Fecha y hora: Fecha y hora en la que se ingresa el requerimiento. - Tipo Bien: Catálogo que especificará si Activo Fijo u Otros - Tipo Requisición: Administrativo o cuenta propia - Carga de Archivos: Campo en el cual se deberá permitir que el usuario añada archivos al sistema.
<p>Validación:</p> <ul style="list-style-type: none"> - Debe validarse los campos que sean números ejemplo cantidad deberá solamente aceptar valores numéricos. - Previo al proceso de guardar la información, el sistema validará información de campos obligatorios. - Previo al proceso de actualizar la información, el sistema validará información de campos obligatorios. - Siempre deben existir mensajes para indicar al usuario sobre las acciones que se van a realizar.

Generado por el (Autor)

Historia de Usuario	
Iteración:1	
Número: 2	Nombre Historia de Usuario:Análisis de compra y aprobación de ticket de compra
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Administradores de Adquisiciones / Usuario generador de la Adquisición	
Prioridad en Negocio: Alta	
Riesgo en Desarrollo: Alta	
<p>Descripción:</p> <p>Como usuario de la aplicación, quiero realizar el análisis de un Ticket de compra, entonces: Ingresar con login y password para acceder al sistema y así tener los permisos necesarios para ingresar a la pantalla para ver los tickets de compra.</p> <p>Luego visualizar todas los tickets pendientes que tiene para su revisión en la pantalla correspondiente a tickets pendientes.</p> <p>El sistema permitirá revisar los detalles de la compra, pro no modificarlos.</p>	

Mediante una nueva funcionalidad permitir en la pantalla de ticket de compras ingresar información de posibles proformas de proveedores a los que se va a comprar lo indicado en el ticket.

Pantalla de Análisis de compra y aprobación de ticket de compra

Se debe incluir los campos que existen en la pantalla de adquisiciones para que sean visualizados por el usuario que realizará el análisis de compra y aprobación del ticket de compra. Estos campos no tendrán opción a ser modificados por el usuario aprobador.

A continuación se deberá ingresar los proveedores ésta información debe ser presentada en un grid, en el cual se podrá modificar, y eliminar los registros de acuerdo a las necesidades del usuario.

Además se desarrollará una pantalla (pop-up) en la que se ingresará los siguientes campos con la información de los proveedores:

- Identificación
- Nombres de los proveedores
- Valor
- Calificación
- Excepción Proveedor único
- Observaciones.

También permitirá la creación de nuevos proveedores

Se deberá tener un campo para observaciones en esta etapa.

El ticket deberá dirigirse al jefe de adquisiciones, el mismo aprobará o rechazará el ticket.

En caso de ser aprobado deberá dirigirse a la etapa de presupuestación.

Observaciones:

Esta pantalla deberá presentar la siguiente información:

Pantalla de Análisis de compra y aprobación de ticket de compra

- Información de la pantalla de adquisiciones.
- Grid con información sobre nombre de proveedores: Se presentará la información de los proveedores

Validación:

- Debe validarse los campos que sean números ejemplo cantidad deberá solamente aceptar valores numéricos.
- Previo al proceso de guardar la información, el sistema validará información de campos obligatorios.
- Previo al proceso de actualizar la información, el sistema validará información de campos obligatorios.
- Siempre deben existir mensajes para indicar al usuario sobre las acciones que se van a realizar.

Generado por el (Autor)

Historia de Usuario	
Iteración:2	
Número: 3	Nombre Historia de Usuario:Aprobación proveedor y compra del bien o servicio
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Administrador de adquisiciones / usuario que genero el ticket de adquisición	
Prioridad en Negocio: Alta	
Riesgo en Desarrollo: Alta	
<p>Descripción:</p> <p>Como usuario de la aplicación, quiero realizar la aprobación de un Ticket de compra, entonces:</p> <p>El usuario debe ingresar con su login y password para acceder al sistema y así tener los permisos necesarios para ingresar a la pantalla para generar los tickets de compra.</p> <p>El usuario visualizará todas los tickets pendientes que tiene para su revisión en la pantalla correspondiente a tickets pendientes.</p> <p>Pantalla de aprobación proveedor y compra</p> <p>Debe presentarse la información de la pantalla de análisis de compra y aprobación de ticket de compra. Estos campos no tendrán opción a ser modificados por el usuario aprobador.</p> <p>En el grid de bienes a adquirir se podrá seleccionar una cantidad aprobada o seleccionar toda la cantidad de cada uno de los proveedores, se selecciona a los proveedores y estos pasan a un nuevo grid de proveedores adjudicados para la compra.</p> <p>Según el rango en el valor, se direcciona al nivel de aprobación correspondiente con toda la información anterior, permitiendo modificar adquisición (cantidad) aprobación ciertos ítems.</p>	
<p>Observaciones:</p> <p>Esta pantalla deberá presentar la información la siguiente información:</p> <p>Pantalla aprobación proveedor y compra</p> <ul style="list-style-type: none"> - Información de la pantalla de adquisiciones. - Información de la pantalla de análisis de compra y aprobación de ticket de compra. - Grid con información de los bienes a adquirir - Grid con información de los proveedores para la aprobación de la compra. - Grid con información de los proveedores adjudicados. - Observaciones. 	
<p>Validación:</p> <ul style="list-style-type: none"> - La hora y fecha de entrada y salida de una etapa es muy importante puesto que permitirá tener un control sobre los tiempos de procesamiento en cada etapa. 	

- Debe validarse los campos que sean números ejemplo cantidad deberá solamente aceptar valores numéricos.
- Previo al proceso de guardar la información, el sistema validará información de campos obligatorios.
- Previo al proceso de actualizar la información, el sistema validará información de campos obligatorios.
- Siempre deben existir mensajes para indicar al usuario sobre las acciones que se van a realizar.

Generado por el (Autor)

Historia de Usuario	
Iteración:2	
Número: 4	Nombre Historia de Usuario:Compra del bien o servicio
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Administrador de adquisiciones	
Prioridad en Negocio: Alta	
Riesgo en Desarrollo: Alta	
<p>Descripción:</p> <p>Como usuario de la aplicación, quiero aprobar la compra de un Ticket de compra, entonces:</p> <p>El usuario debe ingresar con su login y password para acceder al sistema así tener los permisos necesarios para ingresar a la pantalla para generar los tickets de compra.</p> <p>El usuario visualizará todas los tickets pendientes que tiene para su revisión en la pantalla correspondiente a tickets pendientes.</p> <p>Pantalla de Compra de bien – servicio</p> <p>Se debe incluir los campos que existen en la pantalla de adquisiciones para que sean visualizados por el usuario que realizará el análisis de compra y aprobación del ticket de compra. Estos campos no tendrán opción a ser modificados por el usuario aprobador.</p> <p>Adicionalmente debe presentarse la información de la pantalla de análisis de compra y aprobación de ticket de compra. Estos campos no tendrán opción a ser modificados por el usuario aprobador.</p> <p>En esta etapa deberá existir un comentario u observación.</p>	
<p>Observaciones:</p> <p>Esta pantalla deberá presentar la información la siguiente información:</p> <p>Pantalla de Compra de bien – servicio</p> <ul style="list-style-type: none"> - Información de la pantalla de adquisiciones. 	

- Información de la pantalla de análisis de compra y aprobación de ticket de compra.
- Grid con información de los proveedores adjudicados para la aprobación de la compra.
- Información de requiere contrato
- Observaciones.

Validación:

- La hora y fecha de entrada y salida de una etapa es muy importante puesto que permitirá tener un control sobre los tiempos de procesamiento en cada etapa.
- Debe validarse los campos que sean números ejemplo cantidad deberá solamente aceptar valores numéricos.
- Previo al proceso de guardar la información, el sistema validará información de campos obligatorios.
- Previo al proceso de actualizar la información, el sistema validará información de campos obligatorios.
- Siempre deben existir mensajes para indicar al usuario sobre las acciones que se van a realizar.

Generado por el (Autor)

Anexo K. Product Backlog aplicado al Caso de Estudio

Desarrollo ágil: Pila de Producto (Product Backlog)

Proyecto: Generación de Ticket de compra

Elaborado por: Santiago León

ID de la Historia	Enunciado de la Historia	Alias	Estado	Dimensión / Esfuerzo	Iteración (Sprint)	Prioridad	Comentarios
1	a el área de administrativo de la empresa y que este sea validado por las deudas usuarios, para que de acuerdo a valores y otros factores estos sean aprobados, o productos o servicios que los usuarios están solicitados comprar en base a diferentes y realizar el proceso de compra de un bien o servicio solicitado por un usuario y validado por las deudas usuarios	Generación de Ticket de compra	Planificada	50	Sprint 1	1	
2	Análisis de compra y aprobación de ticket de compra	Aprobación proveedor y compra del bien o servicio	Planificada	50	Sprint 1	2	
3	Compra del bien o servicio	Aprobaciones por montos	Nueva	100	Sprint 2	3	
4	compras intermedio validado por montos de acuerdo a los mismos se realicen	Aprobaciones por montos	Nueva	100	Sprint 2	4	
5	compras intermedio validado por montos de acuerdo a los mismos se realicen	Aprobaciones por montos	Nueva	30	Sprint2	4	

Anexo L. Metáfora aplicado al Caso de Estudio

Identificación de la Metáfora	
Creación de Tickets de Compra	
Metáfora del Sistema	
<p>Se llena un documento físico, en el que se ingresan datos como:</p> <p>Nombre del solicitante fecha de solicitud oficina Área Número de Solicitud</p> <p>Luego de esto en un párrafo más abajo se llena la información del bien o servicio que se desea adquirir se describe la cantidad, la descripción del bien y si amerita una observación que ayude a tener más claro lo que se quiere comprar luego esta solicitud es entregada a la oficina de administración para que sea procesada, quedando el solicitante a la espera de una respuesta Positiva o negativa sobre la adquisición que se solicitó, el administrativo procede a sellar el documento como recibido.</p>	
Información de aprobación de la Metáfora	
Santiago León	María Elena Romero
Firma del Entrenador (Coach)	Firma del cliente

Generado por el (Autor)

Anexo M. Tarjetas CRC aplicado al Caso de Estudio

Clase: Compra	
Responsabilidad 1 Mantenimiento de entidad Compra	Colaborador 1 Detalle Compra
. Ingreso	Colaborador 2 Proformas
. Actualización	
. Eliminación	Colaborador 3 Definición de nemónico
Responsabilidad 2 Parámetros de Despliegue .Recibir iteración .Relación Cabecera detalle compra	

Responsabilidad 3 Obtención de información de uno o varios tickets de compra .Consultar por Id de Compra .Consultar por parámetros Id, Rango de Fechas, Estados	
Comentarios	
Permite realizar el mantenimiento de una compra, creación, actualización, eliminación, búsqueda, y su asociación con el detalle de la compra, así como la generación de los nemónicos por cada una de las áreas donde se genera el ticket	

Generado por el (Autor)

Clase: Detalle Compra	
Responsabilidad 1 Mantenimiento de entidad Detalle Compra	Colaborador 1 Compra
. Ingreso	Colaborador 2 Archivos Adjuntos
. Actualización	
. Eliminación	Colaborador 3 Pasajes
Responsabilidad 2 Parámetros de Despliegue .Recibir iteración .Relación compra	Colaborador 4 Proveedores
Responsabilidad 3 Obtención de información del detalle de la compra, basado en el id principal de la compra .Relación compra	
Responsabilidad 4 Relación con archivos adjuntos por cada detalle .Relación archivos adjuntos	
Responsabilidad 5 Relación con Pasajes .Relación pasajes	
Responsabilidad 6 Relación proformas y proveedores .Relación proformas .Relación proveedores	

Comentarios

Permite la creación de los detalles de una cabecera de compra en la que se detalla que bien o servicio se procede a solicitar, en el caso de ser un pasaje se procede a la asociación con la clase de pasajes en la que se detalla destino, origen, personas que viajan hora y fechas de salida y retorno.

Además se procede a adjuntar archivos en formato pdf para describir de manera gráfica el bien o servicio a adquirir.

Generado por el (Autor)

Anexo N. Tarjetas de Tareas aplicados al Caso de Estudio

Tarea Sprint 1	
Número Tarea: 1	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Cabecera Compra Base de Datos	
Estado: En proceso	
Tiempo Estimado:	4 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 3
Fecha Inicio:	Fecha Fin:
Programador Responsable: Luis Serrano	
<p>Descripción:Generación de estructura de cabecera para la entidad compra y sus procedimientos almacenados para inserción, actualización, eliminación, y búsqueda, con sus correspondientes claves foráneas y asociaciones a otras estructuras de la base de datos.</p> <p>Asociación al correspondiente esquema de trabajo.</p> <p>Mantener los estándares de programación especificados.</p>	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 2	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Detalle Compra Base de Datos	
Estado: En proceso	
Tiempo Estimado:	4 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 3
Fecha Inicio:	Fecha Fin:
Programador Responsable: Luis Serrano	
<p>Descripción: Generación de estructura para la entidad detalle compra y sus procedimientos almacenados para inserción, actualización, eliminación, y búsqueda, con sus correspondientes claves foráneas y asociaciones a otras estructuras de la base de datos.</p> <p>Asociación al correspondiente esquema de trabajo.</p> <p>Mantener los estándares de programación especificados.</p>	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 3	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Archivos Adjunto Base de Datos	
Estado: En proceso	
Tiempo Estimado:	4 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 3
Fecha Inicio:	Fecha Fin:
Programador Responsable: Luis Serrano	
<p>Descripción: Generación de estructura para la entidad Archivos Adjuntos y sus procedimientos almacenados para inserción, actualización, eliminación, y búsqueda, con sus correspondientes claves foráneas y asociaciones a otras estructuras de la base de datos.</p> <p>Asociación al correspondiente esquema de trabajo.</p> <p>Mantener los estándares de programación especificados.</p> <p>Asociación a las estructuras correspondientes como detalleCompra</p>	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 4	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Cabecera Compra Acceso a Datos y Core	
Estado: Planificada	
Tiempo Estimado:	3 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha Fin:
Programador Responsable: Eduardo Bustillos	
<p>Descripción: Generación de métodos de cabecera para la entidad compra para inserción, actualización, eliminación, y búsqueda, con sus correspondientes tipos de datos y validaciones, así como los parámetros de entrada y salida necesarios.</p> <p>Generación de entidades para trasportar la información por todas las capas de la aplicación.</p>	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 5	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Detalle Compra Acceso a Datos y Core	
Estado: Planificada	
Tiempo Estimado:	3 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha Fin:
Programador Responsable: Eduardo Bustillos	
Descripción: Generación de métodos para la entidad detalle compra para inserción, actualización, eliminación, y búsqueda, con sus correspondientes tipos de datos y validaciones, así como los parámetros de entrada y salida necesarios. Generación de entidades para trasportar la información por todas las capas de la aplicación.	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 6	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Archivos Adjunto Acceso a Datos y Core	
Estado: Planificada	
Tiempo Estimado:	2 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 2
Fecha Inicio:	Fecha Fin:
Programador Responsable: Eduardo Bustillos	
Descripción: Generación de métodos para la entidad archivos adjuntos para inserción, actualización, eliminación, y búsqueda, con sus correspondientes tipos de datos y validaciones, así como los parámetros de entrada y salida necesarios. Generación de entidades para trasportar la información por todas las capas de la aplicación. Asociación e búsqueda por parámetro iddetallecompra, filtros de búsqueda por tipo de archivo y padre del archivo Eliminación de archivos física del repositorio	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 7	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Cabecera Compra Lógica del Negocio	
Estado: En proceso	
Tiempo Estimado:	8 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 5
Fecha Inicio:	Fecha Fin:
Programador Responsable: Luis Serrano	
Descripción: Generación de métodos, procesos y entidades que permitirán la manipulación, validación y preservación de la información de la información en el repositorio, así como las validaciones correspondientes y procesos lógicos del negocio, como validaciones de compras por montos, validaciones de proveedores entre otras.	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 8	Historia de Usuario (1. Generación de Ticket de compra)
Nombre Tarea: Detalle Compra Lógica del Negocio	
Estado: En proceso	
Tiempo Estimado:	5 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 4
Fecha Inicio:	Fecha Fin:
Programador Responsable: Luis Serrano	
Descripción: Generación de métodos, procesos y entidades que permitirán la manipulación, validación y preservación de la información de la información en el repositorio.	

Generado por el (Autor)

Tarea Sprint 1	
Número Tarea: 9	Historia de Usuario (2. Análisis de compra y aprobación de ticket de compra)
Nombre Tarea: Generación de métodos y procesos para proformas	
Estado: En proceso	
Tiempo Estimado:	8 (horas)
Tipo de Tarea : Desarrollo	Puntos Estimados: 5
Fecha Inicio:	Fecha Fin:
Programador Responsable: Eduardo Bustillos	
Descripción: Generación de métodos, procesos y entidades que permitirán la manipulación, validación y preservación de la información de la información en el repositorio sobre proformas y proveedores asociados a un ticket de compra.	

Generado por el (Autor)

Anexo O. Casos de Pruebas

Plantilla de Casos de Pruebas de Software

Proyecto Generación de tickets de Compra

Elaborado por: Santiago León

Ciclo de Pruebas Validación del proceso de generación de tickets de compras.

Id	Caso de Prueba	Descripción	Fecha	Área Funcional / Sub proceso	Funcionalidad / Característica	Datos / Acciones de Entrada	Resultado Esperado	Requerimientos de Pruebas
1	Generación de ticket	Generación de ticket de compra con ingreso de los ítems a comprar	05/10/2014	Todas las áreas		solicitar en compra, ingresar	Generación de ticket con éxito	creados (Catálogos, parámetros)
2	Generación de ticket	Generación de ticket de compra con ingreso de los ítems a comprar	05/10/2014	Todas las áreas	Validación de campos numéricos, alfanuméricos	cantidad, ingreso de datos alfanumérico	Datos ingresados con éxito	Ninguno
3	Validación en compra	Revisión de compra en jefatura del área	06/10/2014	Todas las áreas	Validación de adquisiciones	Datos de compra	Datos ingresados con éxito	
4	Generación de proformas	Generación de proformas	08/10/2014	Administrativas	Validación de información de proformas	Datos de compra	de datos de proformas con valores	Ninguno
5	proveedores	proveedores enlazados a una proforma	08/10/2014	Administrativas	Validación de proveedores	Datos compra y proforma	proveedores a proformas	Ninguno

Anexo P. Plan de Pruebas de integración

Nombre Proyecto

Plan de Pruebas de Integración

Elaborado por: Santiago León

Proyecto: Generación de Ticket de compra

Nota: Para cada caso de prueba se debe mostrar la siguiente información:

- Número del caso de prueba: Número secuencia que hace referencia a los casos de pruebas definidos.
- Componentes a los que hace referencia cada caso de prueba
- Prerrequisitos que se deben cumplir para realizar cada caso de prueba
- Descripción de cada uno de los pasos a realizar para realizar el caso de prueba
- Los datos que se utilizarán de entrada
- La salida que se espera de ejecutar cada paso
- Las columnas sombreadas, correspondientes a 'Resultados' se rellenarán una vez ejecutadas las pruebas, obteniendo así el Informe de Resultado de Pruebas de Integración

Número del Caso de Prueba	Componente	Descripción de lo que se Probará	Prerrequisitos
CA01	Generación de ticket compra Ingreso de datos de adquisición	Creación de una compra con los datos de cabecera automáticos y el ingreso de los datos del bien o servicio a solicitar	Carga de datos básicos de la aplicación como son Área, departamento, usuario, fecha del sistema
CA04	Ingreso de proveedores ingreso de proformas	Enlace de datos adquisición con una proforma y proveedores	Carga de datos de detalle del bien o servicio a adquirir
C01			

Paso	Descripción de pasos a seguir	Datos Entrada	Salida Esperada	¿OK?	Observaciones
1	Verificar los datos básicos de la aplicación	Área Fecha del Sistema Usuario Departamento	Ninguno	OK	Ninguna
2	Ingreso de datos de bien o servicio a adquirir cantidad y detalle	Cantidad y descripción del bien o servicio	Ninguno	Ok	Ninguna
3	Grabar y generar ticket de compra	Ninguno	Mensaje de que el ticket se generó con éxito y el número de ticket generado	OK	ninguna
C04					
Paso	Descripción de pasos a seguir	Datos Entrada	Salida Esperada	¿OK?	Observaciones
1	Seleccionar ticket generado	Datos del Ticket de compra	Ninguna	OK	Ninguno
2	Ingresar datos de proforma	Datos de Proveedor: Identificación Nombre proveedor Calificación Actividad	Validación de existencia del proveedor	OK	Ninguno
3	Ingreso de Datos de la proforma	Valor de la proforma Observación Dato de proveedor recurrente	Ninguna	OK	Ninguna

Generado por el (Autor)

Anexo Q. Acta de cierre del proyecto aplicado al caso de estudio**Acta de Cierre de
Proyecto o Fase*****Módulo de Generación para tickets de Compra******Fecha:29/01/2015*****Información del Proyecto****Datos**

Empresa / Organización	Empresa XYZ
Proyecto	Módulo para Generación de tickes de compra
Fecha de preparación	29/01/2015
Cliente	Representante: María Elena Romero
Patrocinador principal	Luis Caicedo
Gerente de Proyecto	Roberto Murillo

Patrocinador / Patrocinadores

Nombre	Cargo	Departamento / División	Rama ejecutiva (Vicepresidencia)
María Elena Romero	Contadora	Contabilidad	Financiero
Luis Caicedo	Jefe financiero	Contabilidad	Finanzas
Roberto Murillo	Jefe de sistemas	Sistemas	Tecnología

Razón de cierre

Mediante la presente se informa que la **versión final**, del sistema de Generación de tickes de compra, se encuentra actualmente instalado en los servidores del ambiente de producción de la empresa XYZ, cumpliendo con lo estipulado en el contrato que mantienen la mencionada empresa y SL Software.

Por medio de la presente, se da cierre formal al proyecto, por las razones especificadas en la siguiente ficha:

Marcar con una "X" la razón de cierre:

Entrega de todos los productos de conformidad con los requerimientos del cliente.	X
Entrega parcial de productos y cancelación de otros de conformidad con los requerimientos del cliente.	
Cancelación de todos los productos asociados con el proyecto.	X

Aceptación de los productos o entregables

A continuación se establece cuales entregables de proyecto han sido aceptados:

Entregable	Aceptación (Si o No)	Observaciones
Módulo de Proveedores	Si	
Módulo de Creación de Adquisiciones	Si	
Flujos de Aceptación de Órdenes de Pago	Si	
Módulo de Creación de Pagos	SI	

Para cada entregable aceptado, se da por entendido que:

- El entregable ha cumplido los criterios de aceptación establecidos en la documentación de requerimientos y definición de alcance.
- Se ha verificado que los entregables cumplen los requerimientos.
- Se ha validado el cumplimiento de los requerimientos funcionales y de calidad definidos.
- Se ha realizado la transferencia de conocimientos y control al área operativa.
- Se ha concluido el entrenamiento que se definió necesario.
- Se ha entregado la documentación al área operativa.

Se autoriza al Gerente de Proyecto a continuar con el cierre formal del proyecto o fase, lo cual deberá incluir:

- Evaluación post-proyecto o fase.
- Documentación de lecciones aprendidas.
- Liberación del equipo de trabajo para su reasignación.
- Cierre de todos los procesos de procura y contratación con terceros.
- Archivo de la documentación del proyecto.

Una vez concluido el proceso de cierre, el Patrocinador (Sponsor) del proyecto deberá ser notificado para que el Gerente de Proyectos sea liberado y reasignado.

Aprobaciones

Patrocinador	Fecha	Firma
María Elena Romero	20/01/2015	
Luis Caicedo	22/01/2015	
Roberto Murillo	23/01/2015	