



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

“LAS HISTORIAS DE USUARIO COMO TÉCNICA PARA EL PERFIL
BÁSICO DE LA 29110”

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

SERGIO ANGEL CÁRDENAS TLAXCALA

DIRECTOR DE TESIS:

M. en C. MARÍA GUADALUPE ELENA IBARGÜENGOITIA GONZÁLEZ

México, D.F.

2011.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



Índice

Introducción	3
Objetivo	3
Antecedentes	4
Capítulo I Técnicas Ágiles	6
1.1 Historias de Usuario	6
1.1.2 Características de las Historias de Usuario.	8
1.2 Pruebas de Aceptación.....	11
1.3 Plan de Iteración.	12
1.4 Estimación	12
1.5 Velocidad.....	13
1.6 Liberaciones	14
1.7 Riesgos de las Historias de Usuario.....	14
Capítulo II Descripción de la ISO/IEC 29110.....	16
2.1 Roles	18
2.2 Administración de Proyectos	19
2.3 Desarrollo de Software.....	23
Capítulo III Mapeo de las Historias de Usuario a la ISO/IEC 29110.....	29
3.1 Administración de Proyectos	31
3.1.1 Planeación del Proyecto.....	31
3.1.2 Ejecución del Plan del Proyecto	38
3.1.3 Evaluación y Control del Proyecto.	41
3.1.4 Cierre del Proyecto.....	42
4.2 Desarrollo de Software.....	43
4.2.1 Inicio del Desarrollo de Software	43
4.2.2 Análisis de Requerimientos de Software	44
4.2.3 Arquitectura de Software y Diseño Detallado.....	48
4.2.4 Construcción del Software	51
4.2.5 Integración de Software y Pruebas	54
4.2.6 Entrega del Producto.....	57
Capítulo IV Ejemplos.....	59



4.1 Administración de Proyectos.	60
4.1.1 Planeación del Proyecto.....	60
4.1.2 Ejecución del Plan del Proyecto	69
4.1.3 Evaluación y Control del Proyecto	72
4.1.4 Cierre del Proyecto.....	73
4.2 Desarrollo de Software.....	74
4.2.1 Iniciación de la implementación del Software	74
4.2.2 Análisis de Requerimientos del Software.....	74
4.2.3 Arquitectura de Software y Diseño Detallado.....	79
4.2.4 Construcción del Software	80
4.2.5 Integración de Software y Pruebas	83
Tabla 4.42. Configuración del Software.	86
4.2.6 Entrega del Producto.....	87
Conclusiones y trabajo futuro	91
Referencias.....	93



Introducción

La industria del desarrollo del software se ha convertido en una parte importante dentro de la economía de muchos países, y está formada principalmente por empresas de menos de 25 empleados, las cuales enfrentan problemas al implantar procesos de mejora dentro de sus organizaciones, los cuales les permitirían obtener una mayor estabilidad, tanto económica como técnica. Por ello, la organización internacional ISO formó un grupo de trabajo llamado el WG24 que determinó la generación de un estándar para el desarrollo de software para entidades pequeñas, denominada ISO/IEC 29110, aprobada por parte de la comunidad internacional en el 2011. Con este estándar la industria del software tendrá un conjunto de buenas prácticas que pueden ser realizadas durante la ejecución de su proceso de desarrollo de software.

Objetivo

Para las Very Small Enterprise (VSE's), que en este trabajo se denominarán las Pequeñas y Medianas Empresas (PyME's), es importante contar con guías y técnicas [Laporte, 2008] que les permitan apoyar la realización de las actividades derivadas por la norma.

Las propuestas ágiles son bien vistas por las PyME's [Coleman, 2008], por lo cual será de gran utilidad poder apoyar a las actividades de la 29110 con estas técnicas ágiles, ya que las metodologías ágiles tienen la filosofía de *ser adaptables en lugar de predictivas* [XP, 2003], “dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas” [Canós, 2010], que pueden ayudar a minimizar la complejidad del desarrollo del software. “Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo” [Canós, 2010].

Una técnica ágil para especificar los requisitos del software, son las Historias de Usuario [Canós, 2010] las cuales se caracterizan por “ser independientes, negociables, y de gran valor tanto para el cliente como para el programador” [Cohn, 2004], además de que pueden apoyar a la planeación y estimación del proyecto.

Teniendo en cuenta lo expuesto anteriormente, el presente trabajo propone una estrategia que utiliza las técnicas y propuestas descritas por las Historias de Usuario para guiar la realización de actividades y tareas descritas en los procesos de Administración de Proyectos y Desarrollo de Software que forman parte de la ISO 29110.

El objetivo es proveer a las pequeñas y medianas empresas de una guía de trabajo ágil completa, que les permita llevar a cabo la implantación de modelos como la ISO 29110 mediante la técnica de Historias de Usuario.



Antecedentes

Las pequeñas empresas productoras de software tienen la necesidad de implementar un conjunto de buenas prácticas dentro de su organización, para implementar una mejora de procesos y lograr el desarrollo de un producto con calidad. El SEI ha generado distintos modelos de calidad, que permiten asegurar la calidad de los productos de software pero resultan ser muy costosos, debido a que han sido generados para grandes organizaciones.

La norma ISO 29110 surge como respuesta a esta necesidad ya que tiene en cuenta algunos elementos de los estándares internacionales creados por SEI e ISO, entre los cuales se destacan CMMI, ISO/IEC 12207, ISO/IEC 15504 [Oktaba, 2008]. El objetivo de esta norma es proveer a las PyME's de un marco de trabajo que se adecue a sus necesidades y programas de mejora.

La ISO/IEC 29110 está compuesta por varios documentos los cuales se dividen en tres categorías:

- *Visión General:* Documento introductorio al conjunto de documentos. Este informe incluye los conceptos principales necesarios para comprender y utilizar los documentos de ISO/IEC 29110.
- *Perfiles:* Los perfiles se definen con el propósito de empaquetar referencias a y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las PyME's [García, 2010]. Preparar un perfil implica producir dos tipos de documentos:

Marco de trabajo y taxonomía (TR29110-2): Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO/IEC 29110.

Especificaciones de perfil (TR29110-4): Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de "entrada". Para cada perfil existe un documento de este tipo.

- *Guías:* Documentos orientados al usuario. Las guías contienen directrices de aplicación sobre cómo realizar los procesos para alcanzar los niveles de madurez. Se desarrollan para la implantación de los procesos y para la evaluación [García, 2010]. Están compuestas por:

Guías de evaluación (TR29110-3): describe el proceso a seguir para realizar una evaluación que determine las capacidades de proceso y la madurez organizacional.

Guías de ingeniería de gestión (TR29110-5): Las guías de ingeniería y gestión proporcionan orientación sobre la implementación y uso de un perfil. Para cada perfil, existe una guía de este tipo.



La estructura de la norma facilita su adopción, acorde a las necesidades del negocio y de las PyME's, lo cual las proveerá de un modelo de referencia que se adapte a su organización.

No obstante, su estructura no provee guías concretas que apoyen la ejecución de las tareas y actividades para obtener los productos propuestos.

Las propuestas ágiles son bien vistas por las PyMES [Coleman, 2008], por lo cual será de gran utilidad poder apoyar a las actividades de la ISO/IEC 29110 con metodologías ágiles. Se propone la utilización de plantillas basadas en ésta técnica ágil para facilitar la adopción de las guías y así facilitar el cumplimiento de los objetivos de la ISO 29110.



Capítulo I Técnicas Ágiles

De acuerdo con [Augustine, 2005] citado por [Qumer, 2007], “la naturaleza del desarrollo de software ha cambiado con la transición de la economía industrial a una economía del conocimiento”. De acuerdo con [Qumer, 2007] el incremento del conocimiento del desarrollo del software se genera de forma intensiva, el cual en su mayoría, es tácito y que según [Oualid, 2009] este hecho limita la escalabilidad.

Según [Oualid, 2009] “los proyectos de gran escala comúnmente entregan un software de bajo valor para el negocio, gracias a la incomprensión de los stakeholders” o como menciona [Cockburn, 2001] citado por [Cohen, 2004] Revisión de Tesis.

“clientes pobres resultan en un sistema pobre”; por lo cual [Oualid, 2009] considera que *“las metodologías ágiles se presentan como una alternativa viable para encarar la actual crisis del software en grandes proyectos. Sin embargo, también sufren de limitaciones debido a problemas de escalabilidad y debilidades en su perspectiva de negocio”*.

Además de que “la ausencia de una visión compartida es uno de los factores más importantes para que los proyectos de software fallen” [Qumer, 2007], por ello [Qumer, 2007] menciona que las metodologías ágiles no deben adoptarse de golpe sino en un proceso a mediano o largo plazo.

También [Qumer, 2007] argumenta que “el conocimiento de la ingeniería ágil y los principios de la administración deben ser integrados con los principios del desarrollo de software ágil con el fin de captar y gestionar conocimientos relacionados y utilizarlos para mejorar el rendimiento, aprendizaje y decisión realizados en un ambiente de desarrollo ágil”.

“Los métodos ágiles a menudo son bienvenidos tanto por administradores como por programadores ya que satisfacen la necesidad de liberarse de los altos costos que son impuestos por los métodos de desarrollo de software tradicionales” [Qumer, 2007].

1.1 Historias de Usuario

En esta sección se presentarán las técnicas utilizadas por las historias de usuario:

- Las características de las historias de Usuario
- Pruebas de Aceptación
- Plan de Iteración
- Estimación
- Velocidad
- Liberaciones



En la Figura 1.1 se muestra una visión general de la relación que puede existir entre las técnicas de historias de usuario y el proceso de Administración del Proyecto de la 29110. Esta relación se explicara a mayor detalle a lo largo de este trabajo.

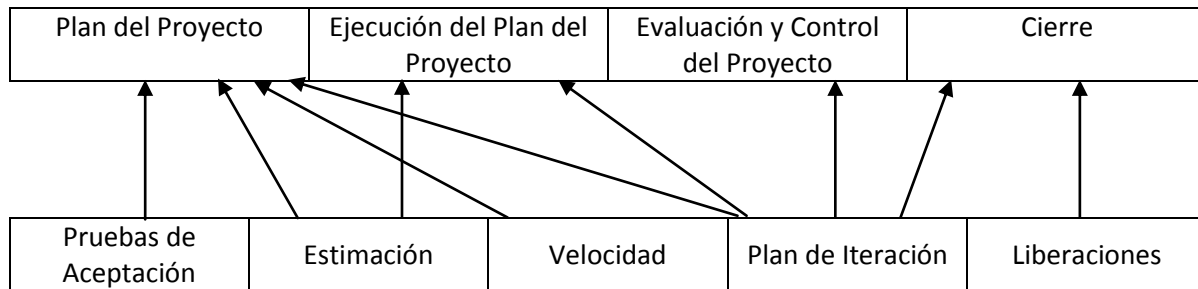


Fig. 1.1. Relación entre técnicas de historias de usuario y administración del proyecto.

El mayor problema que enfrenta un proyecto de desarrollo de software es el incompleto entendimiento del negocio por parte del equipo de desarrollo, ya que el cliente no sabe explicar con claridad sus necesidades, y es precisamente aquí donde se determinan sus alcances y objetivos del proyecto. La mayor parte de este entendimiento se pierde o no se toma en cuenta debido a que los costos y las fechas de entrega acaparan la mayor atención.

“Muchos proyectos fallan debido a la falta de participación del cliente” [Cohn, 2004], lo que priva a los desarrolladores de conocer la visión y expectativas que éste tiene del producto. Las Historias de Usuario promueven que el cliente se involucre en el proyecto, convirtiéndolo en un participante más del desarrollo. “Las Historias de Usuario entran en escena cuando tenemos un proceso que nos brinda información continua y confiable del desarrollo del proyecto” [Cohn, 2004].

Las Historias de Usuario describen funcionalidad que es valiosa para el usuario del software. Las Historias de Usuario están compuestas por tres aspectos [Cohn, 2004]:

- 1) Una descripción escrita de la historia que es utilizada para planificar y recordar.
- 2) Conversaciones acerca de la historia que sirven para profundizar en los detalles de la historia.
- 3) Pruebas que transmiten y documentan detalles y por lo tanto se pueden utilizar para determinar cuándo una historia está incompleta.

Tradicionalmente las Historias de Usuario se anotan en tarjetas de papel, por lo que los aspectos anteriores se conocen como *tarjeta*, *conversación*, y *confirmación* [Cohn, 2004].

Mientras la tarjeta contiene el texto de la historia, los detalles se trabajan en la conversación y son recuperados en la confirmación. Las tarjetas representan las necesidades del cliente más que un documento y no son necesariamente la parte más importante [Cohn, 2004]. Las Historias de



Usuario son utilizadas para definir los requerimientos del cliente y a partir de ellas se puede establecer una estimación y planeación para el desarrollo del software.

1.1.2 Características de las Historias de Usuario.

Las Historias de usuario conforman la parte central de muchas metodologías del desarrollo ágil, ya que estas definen lo que se debe construir. Por ello [Cohn, 2004] describe sus características:

- Las Historias de Usuario deberán ser lo más independientes posibles unas de otras. Las dependencias entre las historias dificultan la priorización, planeación, y estimación del proyecto. Para reducir la dependencia, comúnmente se pueden combinar varias historias, o se pueden dividir de otra manera.
- Las Historias de Usuario son negociables, no son contratos, existen simplemente para apoyar al desarrollo. La parte escrita de la historia es sólo una descripción, en la que no se incluyen detalles, ya que éstos son tratados durante la conversación. La conversación con el cliente puede verse afectada si la parte escrita contiene demasiados detalles, pero si existen algunos que sean muy importantes, estos se pueden incluir como notas. Esto ayudará al cliente y al programador a resumir su conversación, y servirá de guía para añadir detalles a la historia.
- Las Historias de Usuario deben tener algún valor ya sea para el usuario o para el comprador, una forma de lograr esto es haciendo que el cliente escriba las historias, pues es éste quien tiene la mejor visión del alcance del producto. Los clientes comienzan a escribir historias cuando se sienten cómodos con la idea de que son notaciones y pueden modificarse.
- Los desarrolladores deben tener la capacidad de estimar cuanto tiempo les tomara desarrollar la historia, y todo esto depende del conocimiento de los desarrolladores a nivel técnico y del dominio del tema. Estos factores tendrán efecto durante la priorización y la planificación.
- Por ello son ideales en el desarrollo y planeación de proyectos comprometidos en el tiempo, ya que se pueden realizar entregas de software funcional y de valor para el usuario en un corto tiempo. Además de que las historias de usuario permiten a los programadores ir de un nivel de abstracción alto a uno más bajo de forma sencilla y clara.
- Las Historias de Usuario deben ser breves, es decir, mientras más pequeña sea una historia ésta se podrá implementar más fácilmente. A una historia muy larga la llamaremos épica. En caso de tener una historia épica, ésta se podrá dividir en varias que serán más pequeñas, lo cual es más conveniente. Este es un ejemplo de cómo dividir una historia:



“El usuario debe ingresar los datos del cliente como su teléfono, dirección, código postal así como hacer una descripción del producto que se ha adquirido”.

“El usuario debe ingresar los datos del cliente como su teléfono, dirección, código postal”.

“El usuario debe hacer una descripción del producto que ha sido adquirido por el cliente”

- El tamaño pequeño de las historias las hace muy manejables y convenientes para usarlas en los planes de liberación, además de que son fáciles de manejar para los programadores. Las Historias de Usuario posponen los detalles y a cambio dan mayor importancia a un mejor entendimiento de lo que se requiere.
- Existirán ocasiones en que las historias no se podrán dividir y si se logra hacerlo, habrá una gran dependencia entre ellas. También existirán casos en que las historias ya no deberán ser divididas, pues se tendría demasiada granularidad y esto podría generar incongruencias durante el desarrollo, a continuación se presenta un ejemplo de este caso:

“El usuario debe ingresar los datos del cliente como su teléfono, dirección, código postal”.

“El usuario debe ingresar el teléfono del cliente”

“El usuario debe ingresar la dirección del cliente”

“El usuario debe ingresar el código postal del cliente”

- Las historias se deben codificar y probar ya que con ello confirmaremos que la historia ha sido bien comprendida e implementada adecuadamente.
- La parte más tangible de las Historias de Usuario es la tarjeta donde se escriben las necesidades a cubrir, pero es en la conversación donde se recopilan la mayor parte de esa información (grabada, escrita, etc). Las Historias se escriben en tarjetas de papel o electrónicas si se cuenta con alguna herramienta, y deben ser escritas en voz activa. A continuación se muestra en la figura 1.2, un ejemplo de un formato de tarjeta:

Nombre de la Historia: Datos del Cliente	ID de la Historia: 25
El usuario debe ingresar los datos del cliente como su teléfono, dirección y código postal.	

Fig. 1.2 Formato de Tarjeta.



- La principal característica de las Historias de Usuario es que explotan la comunicación entre el grupo de desarrollo y el cliente, la conversación es la clave, aunque no se anota en la tarjeta [Cohn, 2004].
- Las Historias de Usuario surgen como una técnica útil y de gran valor para la generación de un buen producto, pues engloba la comunicación oral y escrita de las ideas principales, dejando de lado la información de los detalles más superfluos. Debe entenderse que la participación del cliente es crucial tanto al principio como a lo largo de todo el proyecto, lo que implica que estará involucrado en tomas de decisiones continuamente, así como para que el equipo de desarrollo pueda consultarle cualquier duda.
- Es en esta comunicación verbal donde radica el verdadero potencial de esta técnica, además de que es aplicable a lo largo del proyecto y no solo en la adquisición de requerimientos. Las historias son ideales para un desarrollo iterativo, esto se debe a que no se deben escribir todas las historias antes de comenzar a desarrollar la primera.
- Las Historias de Usuario comprenden una descripción escrita del problema a resolver y son una nota recordatoria de lo que se debe realizar. De nuevo, el punto importante es el de la conversación, ya que es aquí donde se dejan fuera los detalles más simples y que pueden generar confusión, así que solo se toman las necesidades de mayor funcionalidad. Los detalles que se han dejado de lado por ahora, pueden ser agregados como comentarios a las tarjetas o se pueden retomar como otras historias de usuario.
- En las historias no debe existir un lenguaje dominante, ya sea por parte del desarrollador o del cliente, pues en ambos casos se puede llegar a perder información valiosa y generar confusiones. Los clientes pueden perder de vista la magnitud y dificultad del proyecto, así como su aplicación práctica. El grupo de desarrollo puede perder un gran conocimiento de lo que el cliente realmente necesita.
- Las Historias de Usuario deberán ser escritas por el cliente, ya que es él quien tiene una mejor y mayor visión del proyecto, además de que conservar el lenguaje del negocio es clave para realizar la priorización de las mismas, esto es, deben ser escritas en el lenguaje del usuario y no en el del desarrollador.
- La información que se genera en el equipo de trabajo debido a la comunicación verbal con el cliente, brinda la oportunidad de incrementar el conocimiento del propio programador en lenguajes diferentes al técnico, lo cual le ofrece una visión más general y enriquecedora.



- El equipo del cliente debe estar formado por varias personas que van a utilizar el software, así evitamos depender de la visión de una sola persona, incrementamos la funcionalidad, y cumplimos de mayor manera con las expectativas del cliente. Este equipo puede estar integrado por gente que va a utilizar el software (usuarios), administradores de producto, etc. Este equipo incrementará en gran medida el valor del software, pues podrá aportar nuevas ideas durante el desarrollo.
- Una ventaja de las historias es que son muy prácticas y con un poco de experiencia se pueden generar de manera rápida y con mayor información.

1.2 Pruebas de Aceptación.

Se utilizan para demostrar que la historia ha sido correcta y completamente codificada. Una razón para escribir las pruebas de aceptación es la de expresar muchos de los detalles que resultaron de la conversación entre el cliente y los desarrolladores al ser escritas las historias de usuario [Cohn, 2004].

Las expectativas del cliente son de gran importancia y que el sistema cumpla con ellas, demuestra que el grupo de desarrollo tiene gran interés en cumplirlas, así que una buena forma de registrar esas expectativas es dentro de las Historias de Usuario como pruebas de aceptación, las cuales pueden escribirse en la parte posterior de la tarjeta y ser agregadas o eliminadas en cualquier momento, de preferencia antes de comenzar a programar.

Las pruebas se escriben durante la charla que tiene lugar entre el cliente y el grupo de desarrollo, así se pueden registrar más detalles. Al comienzo de cada iteración se pueden revisar las pruebas escritas y de ser necesario se escribirán nuevas y/o se reescribirán las existentes a fin de garantizar que la historia sea bien implementada.

Las Historias de Usuario deben ser escritas de manera que puedan ser probadas, ya que la confirmación se refiere a las pruebas con las que el cliente validará el funcionamiento del producto, por ejemplo:

- “Se debe poder ingresar al sistema los datos del cliente en los campos correspondientes”
- “Se deben recuperar los datos del cliente sin alteraciones”

Dichas pruebas se diseñarán con el cliente y con un desarrollador especializado en pruebas (el de mayor experiencia en esta área). Éstas determinarán cuando una Historias de Usuario está completa.

Teniendo en cuenta la complejidad y las pruebas a realizar a cada Historia de Usuario, el programador estimará la medida de cada una de ellas, con el fin de determinar el tiempo que tardará en realizarlas.



Se debe evitar probar solo lo que se les pidió, es decir, se debe entender que la realización de las pruebas es parte del proceso de codificación y no esperar hasta el final para realizarlas. También existe la probabilidad de que el cliente pruebe todo, excepto lo que quería probar, es en ese momento cuando las notas realizadas en las historias son de gran ayuda.

1.3 Plan de Iteración.

La planeación es crítica en cualquier proyecto. La planeación de los proyectos permite identificar riesgos, lo cual nos ayuda a responder a ellos. Con la planeación se determina si se podrán cubrir los requerimientos del sistema.

Tanto el cliente como el grupo de desarrollo establecerán un periodo de tiempo dentro del cual, los desarrolladores deberán entregar las Historias que hayan realizado, a este periodo se le llamará iteración, ésta puede durar de una a cuatro semanas, y se debe mantener durante todo el proyecto.

Como indica [Cohn, 2004], las iteraciones cortas permiten correcciones más frecuentes del proyecto y mayor visibilidad en su progreso.

El mantener un periodo constante de producción genera que el proyecto entre en un ciclo natural de desarrollo, y esto provoca que el equipo de trabajo sea más consistente con sus entregas.

El cliente puede realizar cambios antes de comenzar una nueva iteración [Cohn, 2004]. Conforme las iteraciones avanzan podemos realizar una mejor medición de la velocidad, además de que se pueden agregar mayores detalles.

La selección de qué es lo que se deberá entregar en cada liberación es el verdadero problema, ya que se tiene que priorizar las nuevas características y estimar si es posible entregarlas a tiempo, además se debe tener en cuenta, qué tan importantes son para la funcionalidad [Cohn. 2004].

Se debe generar un plan de iteración donde se describan las Historias que serán incluidas en cada iteración.

1.4 Estimación

La estimación es parte importante en la planeación, ya que nos permite tomar decisiones respecto a los tiempos de ejecución y entrega del proyecto. Como lo plantea [Cohn, 2004], uno de los principales propósitos de la estimación, es que sea capaz de responder preguntas acerca del esfuerzo total esperado en un proyecto, y que eventualmente tendremos que convertir las estimaciones en tiempo.

En este caso [Cohn, 2004] establece que las estimaciones de las historias necesitan ser de propiedad colectiva del equipo. Por ello es importante que una parte del equipo esté involucrado durante las estimaciones.



Las estimaciones de las historias las realiza el equipo de trabajo, el cliente solo participa como observador. Pero durante la priorización el cliente siempre tendrá la razón, así que se le debe informar cual podría ser la entrega de historias ya implementadas con mayor lógica.

El equipo de trabajo debe tomar en consideración toda la complejidad que puede envolver a la historia y en base a ello estimar cuanto esfuerzo se requiere para completarla. Con estas estimaciones se les da prioridad a las historias y se decide cuales serán las primeras en desarrollarse.

Con estas estimaciones se puede generar un plan de liberación, en el cuál se debe contemplar la realización de las primeras iteraciones, las historias de mayor prioridad o riesgo, en las siguientes las de mediano valor, y así sucesivamente.

1.5 Velocidad

El esfuerzo que se realiza al desarrollar una historia de usuario dentro del tiempo de duración de una iteración, se debe poder medir de alguna forma, ya que esta medida nos permitirá estimar cuanto trabajo podemos desarrollar por iteración, y a la vez cuantas iteraciones se requieren para terminar las historias del proyecto y por ende la duración aproximada del mismo.

A esta medición la denominaremos velocidad, pero para poder comprender mejor la manera en que trabaja, necesitamos definir otra “unidad de medida” la cual es llamada *Story points* o puntos por historia.

Según [Cohn, 2006] los puntos por historia son una unidad de medida para expresar el tamaño total, función u otra característica de una historia de usuario. El número de puntos por historia asociados a una historia de usuario representa el tamaño total de ésta. No hay una fórmula para definir la medida de una historia, pero una característica interesante de los puntos por historia es que cada equipo los define como mejor les parezca [Cohn, 2004].

Una de estas definiciones podría ser por ejemplo:

- Un día ideal de trabajo (sin interrupciones) se puede tomar como un punto por historia
- Una semana ideal de trabajo puede ser un punto por historia
- Según la complejidad de la historia se tomaría como punto por historia.

De acuerdo con [Cohn, 2004] al final de una iteración el equipo contabiliza cuantos puntos por historia realizó. El equipo utiliza este total para “pronosticar” cuantos puntos serán capaces de completar en las siguientes estimaciones de igual duración.

Bajo este contexto, la velocidad se puede definir como el número de historias de usuario que un equipo completa (o espera completar) durante una iteración [Cohn, 2004].



Así, con la velocidad definida, se podría estimar el tiempo en el cual el proyecto estaría completado. Un ejemplo de esto sería imaginar, que el equipo de trabajo desarrollo 32 puntos en una iteración, y que el proyecto tiene un total de 320 puntos por historia. En este caso, si la velocidad fuese constante, el proyecto se completaría en $320 / 32 = 10$ iteraciones.

Para poder establecer la velocidad del equipo, más apegada a la realidad, se deberá esperar dos o tres iteraciones para así poder realizar los ajustes necesarios. Pero si existe un registro histórico de proyectos similares, podríamos establecer la velocidad que se considero en ese proyecto.

1.6 Liberaciones

Una liberación está compuesta de una o varias iteraciones, de aquí se genera un plan de liberación que pretende lograr la mayor funcionalidad dentro del tiempo establecido. Para poder realizar una liberación, el cliente deberá priorizar cuales Historias son de mayor valor o importancia para el proyecto.

Dentro de este plan de liberación se establece que historias se desarrollarán dentro de la primera iteración y cuales en la segunda, esto sucederá sucesivamente hasta que el total de las historias se integren en el total de iteraciones.

En el caso de que alguna historia no se complete en su iteración correspondiente, se puede integrar en la siguiente, aunque esto incrementa el riesgo del proyecto, ya que podría modificar su duración aproximada.

Se debe tener especial cuidado con el monitoreo de la velocidad y la estimación de las historias, ya que esto puede modificar el plan completo.

El producto puede mejorarse después de cada liberación, ya que se pueden agregar nuevas características y/ó mejorar las existentes, esto se puede realizar teniendo una lista de áreas a mejorar.

Las historias de usuario son descripciones de funcionalidad, no tareas individuales de ingeniería que deben ser realizadas [Cohn, 2006].

1.7 Riesgos de las Historias de Usuario

Algunos riesgos a considerarse cuando se utilizan historias de usuario son:

- Las historias de usuario inconclusas en alguna iteración anterior, ya que estas pueden modificar las fechas de entrega, en este caso se puede acudir con el cliente, para determinar la forma de incluirlas.



-
- Las historias de alto riesgo son las que presentan una mayor dependencia entre ellas, las que no se pueden separar. Se debe consultar al cliente para tomar la mejor decisión.
 - No tener al cliente in-situ. Se pueden concertar juntas extraordinarias con el cliente para esclarecer las dudas del equipo y formular o modificar las Historias de Usuario.
 - No cubrir las pruebas de aceptación. En este caso se deben realizar pruebas unitarias antes de que se termine el plazo de la iteración.

Lo anterior permite generar un plan de acción, el cual permitirá integrar nuevas acciones para los riesgos que se generen debido a la integración de nuevas historias.



Capítulo II Descripción de la ISO/IEC 29110

Los esfuerzos por desarrollar un modelo de mejora de la calidad a nivel internacional, que le permita a las PyME's alcanzar un programa de mejora de procesos que las lleven a alcanzar certificaciones internacionales, han logrado que la ISO haya creado al grupo de trabajo SC7-WG24, el cual ha sido el encargado de desarrollar la norma ISO/IEC 29110 basándose en la norma MOPROSOFT (NMX-I-059-NYCE).

La ISO/IEC 29110 está compuesta por varios documentos, los cuales se dividen en tres categorías:

- *Visión General (TR)*: Documento introductorio al conjunto de documentos. Este informe incluye los conceptos principales necesarios para comprender y utilizar los documentos de ISO/IEC 29110. Introduce los aspectos del negocio, características y requerimientos de las PyME's, y aclara la razón de ser de los perfiles específicos de las PyME's, documentos, estándares y guías [ISO 29110].

- *Perfiles (International Standard Profile ISP)*: Los perfiles se definen con el propósito de empaquetar referencias a y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las PYME'S [García, 2010]. Preparar un perfil implica producir dos tipos de documentos:

Marco de trabajo y taxonomía (TR29110-2): Este documento establece la lógica detrás de la definición y aplicación de los perfiles. Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (clasificación) de los perfiles ISO/IEC 29110.

Especificaciones de perfil (TR29110-4): Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de “entrada”. Para cada perfil existe un documento de este tipo.

- *Guías (o Reporte Técnico por siglas en inglés TR)*: Documentos orientados al usuario. Las guías contienen directrices de aplicación sobre cómo realizar los procesos para alcanzar los niveles de madurez. Se desarrollan para la implantación de los procesos y para la evaluación basado en temas del dominio, las prácticas empresariales y los riesgos [García, 2010]. Están compuestas por:

Guías de evaluación (TR29110-3): describe el proceso a seguir para realizar una evaluación que determine las capacidades de proceso y la madurez organizacional. Las guías de evaluación son aplicables a todos los perfiles.



Guías de ingeniería de gestión (TR29110-5): Las guías de ingeniería y gestión proporcionan orientación sobre la implementación y uso de un perfil. Para cada perfil, existe una guía de este tipo.

La figura 2.1 muestra la organización de estos documentos para facilitar su uso y comprensión [García, 2010].

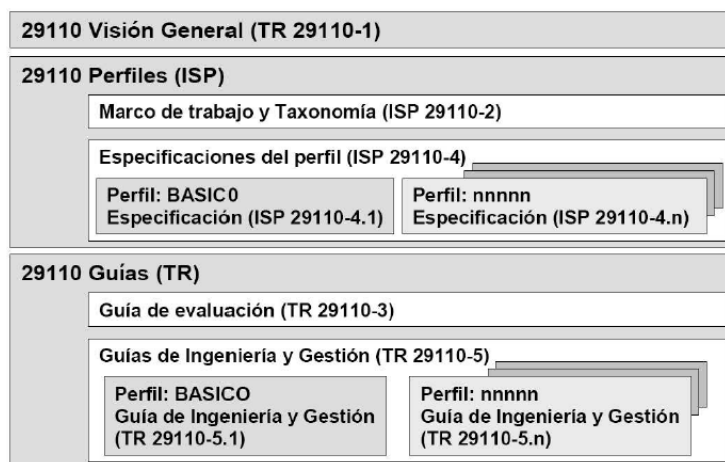


Figura 2.1. Familia de documentos de la ISO/IEC 29110.

El presente trabajo se centra en el uso y manejo del Perfil Básico de la Guía de Ingeniería y Gestión (TR 29110-5.1) que se encuentra dentro de las Guías (TR, Reportes Técnicos en español) de la ISO 29110. Esta guía describe la administración de proyectos y los procesos de desarrollo de software [ISO 29110].

El propósito del perfil básico es definir una implementación del software y una guía de administración de proyectos para un subconjunto de procesos de la ISO/IEC 12207, apropiadas a las características y necesidades de las PyME's. La principal razón de incluir la administración de proyectos es que el negocio principal de las PyME's es el desarrollo de software y su éxito financiero depende del éxito del proyecto [ISO 29110]. En el caso de México existen empresas cuyo negocio principal es diferente al del software, más sin embargo cuentan con un área de sistemas pequeña.

El cliente provee la descripción del producto como una entrada y recibe una Configuración del Software como el resultado del proceso de desarrollo de software, que es controlado por el proceso de administración de proyectos [ISO 29110].

El proceso de Administración de Proyectos recibe la Descripción del Proyecto para elaborar el Plan del Proyecto, el cual sirve para guiar la ejecución del proceso de Desarrollo de Software. Los subprocesos de Evaluación y Control del Proyecto comparan el Registro del Estado del Proyecto contra el Plan del Proyecto, para eliminar desviaciones a través de Acciones Correctivas o cambios al Plan del Proyecto. Finalmente el subproceso de Cierre del Proyecto entrega la Configuración del



Software y obtiene el Documento de Aceptación por parte del cliente para formalizar el cierre del proyecto [ISO 29110].

El proceso de Desarrollo de Software comienza con un subproceso de Iniciación que recibe el Plan del Proyecto que guiará la ejecución de los subprocesos siguientes: Análisis de Requerimientos del Software, Arquitectura y Diseño Detallado del Software, Construcción del Software, Integración del Software y Pruebas, y Entrega del Producto [ISO 29110].

2.1 Roles

Los roles definidos dentro de este perfil básico están descritos en la tabla 2.1 [ISO 29110] y durante la descripción de las actividades se hará referencia a éstos por su abreviación.

Rol	Abreviación	Descripción
Analista	AN	Conocimientos y experiencia en la obtención, especificación y análisis de requerimientos.
Cliente	CL	Conocimiento de los procesos del cliente y capacidad de explicar los requerimientos del cliente.
Diseñador	DIS	Conocimiento y experiencia en los componentes del software y diseño de arquitecturas.
Responsable de Manuales	RM	Conocimiento en técnicas de redacción y experiencia sobre el desarrollo de software y su mantenimiento.
Programador	PR	Conocimiento y/o experiencia en programación, integración y pruebas.
Revisor	RE	Conocimiento de técnicas de revisión y experiencia sobre el desarrollo de software y su mantenimiento.
Administrador de Proyecto	AP	Capacidad de liderazgo con experiencia en toma de decisiones, planeación estratégica, manejo de personal, delegación y supervisión, finanzas y desarrollo de software.
Líder Técnico	LT	Conocimiento y experiencia en el desarrollo de software y su mantenimiento.
Responsable de Pruebas	RP	Conocimiento y experiencia en la planeación y en el rendimiento de la integración y sistemas de pruebas.
Usuario	US	Ninguno
Diseñador de Interfaz de Usuario	DIU	Conocimiento en diseño de interfaces de usuario y criterio ergonómico.
Equipo de trabajo	ET	Conocimiento y experiencia de acuerdo a su rol.

Tabla 2.1. Roles definidos en el perfil básico.



2.2 Administración de Proyectos

El propósito de la administración de Proyectos es establecer y llevar a cabo de una manera sistemática las tareas del Desarrollo del Proyecto, que permitan cumplir con los objetivos del proyecto dentro del tiempo y costo esperados [ISO 29110].

La Administración de Proyectos tiene como finalidad cumplir con los siguientes siete objetivos:

O1. El Plan del Proyecto para la ejecución del proyecto es desarrollado de acuerdo a la Descripción del Proyecto y validado con el cliente, las tareas y recursos necesarios para completar el trabajo son clasificados y estimados.

O2. El progreso del proyecto es monitoreado comparándolo con el Plan del Proyecto, registrado en el Registro del Estado del Proyecto; las acciones correctivas para remediar las desviaciones del plan son tomadas cuando los objetivos del proyecto no son logrados; el cierre del proyecto es realizado para obtener el Documento de Aceptación.

O3. La Solicitud de Cambios es dirigida a través de su recepción y análisis.

O4. Se llevan a cabo reuniones de revisión con el equipo de trabajo y el cliente, los acuerdos son registrados y se les da seguimiento.

O5. Los riesgos son identificados en el desarrollo y durante la realización del proyecto.

O6. Se desarrolla una estrategia de control de versiones del software; los elementos de la Configuración del Software son identificados, y definidos dentro de la línea base; las modificaciones y liberaciones de los elementos son controlados y puestos a disposición del cliente y del equipo de trabajo, así como el almacenamiento, manejo y entrega de los elementos.

O7. El aseguramiento de calidad del software es realizado para garantizar que los productos de trabajo y los procesos cumplan con el Plan del Proyecto y la Especificación de Requerimientos.

Los productos de entrada de este proceso son: Descripción del Proyecto, Configuración del Software y la Solicitud de Cambios. Los productos de salida son: Plan del Proyecto, Documento de Aceptación y Repositorio del Proyecto.

La Administración de Proyectos está compuesta por los siguientes subprocesos:



- **Planeación del Proyecto:** Cumple con los objetivos 1, 5 y 6. Los subprocesos de la Planeación del Proyecto, documentan los detalles necesarios de planeación para administrar el proyecto. Este subproceso lo desarrollan: AP, LT Y CL.

Dentro de este subproceso se efectúan las siguientes tareas:

A1.1 Revisar la Descripción del Proyecto

A1.2 Definir conjuntamente con el Cliente el Protocolo de Entrega de cada uno de los entregables especificados en la Descripción del Proyecto

A1.3 Identificar el número de ciclos y las actividades específicas que se desarrollarán para producir los entregables y sus componentes identificados en la Descripción del Proyecto. Incluir verificación, validación y revisiones con el cliente y el equipo de desarrollo, de las actividades para asegurar la calidad de los productos de trabajo. Identificar las actividades que se deberán realizar al Protocolo de Entrega. Documentar el número de ciclos y actividades.

A1.4 Establecer el tiempo estimado de cada actividad

A1.5 Identificar y documentar los recursos: humanos, materiales, equipo y herramientas, incluyendo el entrenamiento requerido del equipo de trabajo para realizar el proyecto. Generar el calendario cuando los recursos y entrenamiento son necesarios.

A1.6 Conformar el Equipo de Trabajo, asignando roles y responsabilidades de acuerdo con la Descripción del Proyecto.

A1.7 Asignar fechas de inicio y fin a cada una de las actividades para generar el Calendario de las Actividades del Proyecto tomando en cuenta los recursos asignados, la secuencia y dependencia de las actividades.

A1.8 Calcular y Documentar el Costo y Esfuerzo estimado del Proyecto

A1.9 Identificar y documentar los riesgos que pueden afectar al proyecto.

A1.10 Documentar la Estrategia de Control de Versiones en el Plan del Proyecto.

A1.11 Generar o actualizar el Plan del Proyecto antes de iniciar un nuevo ciclo. Además, el Plan del Proyecto puede ser actualizado debido a la Solicitud de Cambios realizada por el usuario o derivados del proyecto.

A1.12 Incluir la descripción del producto y entregables en el Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo.

A1.13 Verificar el Plan del Proyecto.



A1.14 Corregir los defectos encontrados en el Plan del Proyecto con base en el Reporte de Verificación y obtener la aprobación de las correcciones.

A1.15 Validar el Plan del Proyecto.

A1.16 Corregir los defectos encontrados en el Plan del Proyecto con base en el Reporte de Validación y obtener la aprobación de las correcciones.

A1.17 Establecer o preparar el repositorio del proyecto utilizando la Estrategia de Control de Versiones.

A1.18 Incorporar el Plan del Proyecto dentro del Repositorio del Proyecto en la línea base.

- **Ejecución del Plan del Proyecto:** Cumple con los objetivos 2, 3, 4 y 5. El subproceso de la Ejecución del Plan del Proyecto, implementa el plan documentado en el proyecto. Este subproceso lo ejecutan: AP, LT, ET y CL.

Dentro de este subproceso se efectúan las siguientes tareas:

A2.1 Acordar con el Líder Técnico la asignación de tareas al Equipo de trabajo.

A2.2 Revisar el Plan del Proyecto y registrar los datos actuales en el Registro de Estado de Progreso

A2.3 Registrar, analizar y evaluar la solicitud de cambio para costo, calendario e impacto técnico, e incluir los cambios aprobados en el Plan del Proyecto. La solicitud de cambio puede ser iniciada externamente por el cliente o internamente por el equipo de trabajo.

A2.4 Conducir reuniones de revisión con el equipo de trabajo y el cliente, y se registran los acuerdos.

A2.5 Realizar el Respaldo de acuerdo a la Estrategia de Control de Versiones.

A2.6 Realizar la recuperación del Repositorio usando el Respaldo del Repositorio, en caso de ser necesario.

- **Evaluación y Control del Proyecto:** Cubre el objetivo 2. Los subprocesos de Evaluación y Control del Proyecto monitorean y evalúan el rendimiento del plan de los objetivos documentados. Lo desarrolla: AP.

Dentro de este subproceso se efectúan las siguientes tareas:

A3.1 Evaluar la información requerida por el Plan del Proyecto. Las acciones correctivas o solicitudes de cambio son establecidas y seguidas.



- **Cierre del Proyecto:** Cubre el objetivo 2. Los subprocesos del Cierre del Proyecto proporcionan la documentación del proyecto y productos de acuerdo con los requisitos del contrato. Lo ejecutan: AP y CL.

Dentro de este subproceso se efectúan las siguientes tareas:

A4.1 Formalizar la finalización del ciclo o del Proyecto acorde al Protocolo de entrega establecido en el Plan del Proyecto, y conseguir el Documento de Aceptación.

En la figura 2.2 [ISO 29110] se presenta el diagrama de flujo del proceso de Administración de Proyectos formado por los subprocesos: Plan del Proyecto, Ejecución del Plan del Proyecto, Evaluación y Control del Proyecto, y Cierre; junto con sus documentos de entrada, internos y de salida.

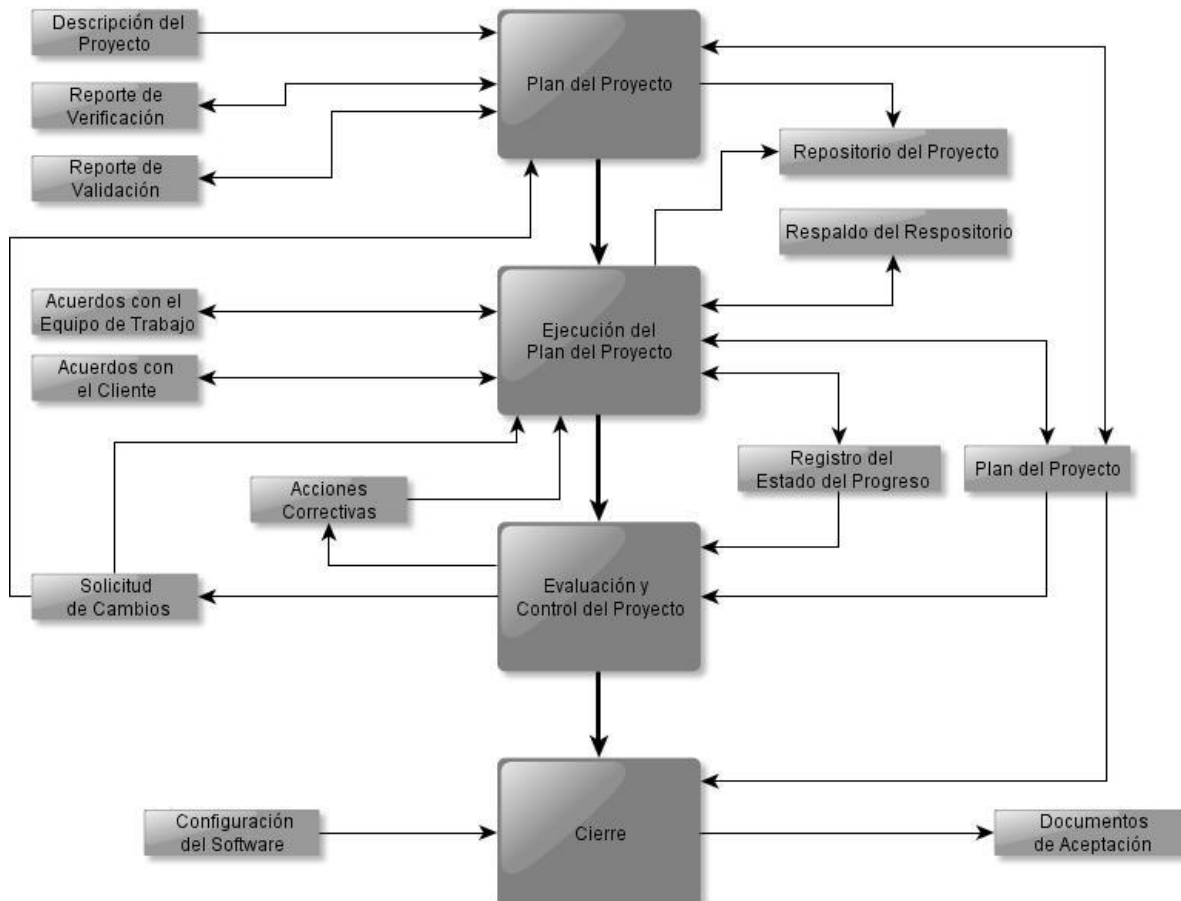


Fig. 2.2. Diagrama de Flujo de Administración de Proyectos.



2.3 Desarrollo de Software

El propósito del proceso de Desarrollo de Software es la realización sistemática del análisis, diseño, construcción, integración y pruebas de subprocesos para los productos de software, nuevos o modificados, de acuerdo a la especificación de requerimientos.

El Desarrollo de Software tiene por meta desarrollar y cumplir con los siguientes objetivos:

O1. Las tareas de los subprocesos son realizadas a través del cumplimiento del Plan de Proyecto actual.

O2. Los requerimientos del Software están: definidos, analizados para su corrección y realización de pruebas, aprobados por el usuario, integrados a la línea base y han sido comunicados; los cambios realizados son evaluados para el costo, calendario e impacto técnico, previo a ser procesado.

O3. La arquitectura del software y el diseño detallado es desarrollado e integrado a la línea base, éstos describen los elementos del software; las interfaces internas y externas del mismo, la consistencia y trazabilidad de los requerimientos del software, son establecidos.

O4. Los componentes del software definidos en el diseño son producidos; la unidad de prueba es definida y realizada para verificar la consistencia con los requerimientos y el diseño, la trazabilidad de los requerimientos y el diseño son establecidos.

O5. La Configuración del software es producida, realizando la integración de los componentes del software, y verificada, utilizando *Casos de Prueba* y *Procedimientos de Prueba*. Los resultados son registrados en el Reporte de Incidentes de Prueba, los defectos son corregidos, la consistencia y trazabilidad del Diseño de Software son establecidos.

O6. Una Configuración del Software, que cumpla con la Especificación de Requerimientos convenida con el cliente, que incluya manuales de usuario, operación y mantenimiento es integrado a la línea base y almacenada en el Repositorio del Proyecto.

O7. Las tareas de verificación y validación para todos los productos de trabajo requeridos, son realizadas, utilizando el criterio definido para lograr que los productos de salida sean consistentes con los productos de entrada en cada subproceso, los defectos son identificados, y registrados en el reporte de Verificación y Validación, y son corregidos.



Los Productos de entrada para este subproceso son: El Plan del Proyecto y Repositorio del Proyecto. Los productos de salida son: La Configuración del Software y Solicitud de Cambios.

El desarrollo de software está conformado por los siguientes subprocesos:

- **Inicio del Desarrollo del Software:** Cubre el objetivo 1. El subproceso de inicio del desarrollo del software, asegura que el Plan del Proyecto establecido anteriormente es aceptado por el equipo de trabajo. Este subproceso lo desarrolla el ET.

Dentro de este subproceso se efectúan las siguientes tareas:

A1.1 Revisión del Plan de Proyecto actual por los integrantes del equipo de trabajo, a fin de lograr un entendimiento común y obtener su compromiso con el proyecto.

- **Análisis de Requerimientos del Software:** Cumple con los objetivos 2, 6 y 7. El subproceso de Análisis de Requerimientos de Software, analiza los requerimientos acordados con el cliente y establece los requerimientos validados del proyecto. Este subproceso lo ejecuta el LT, AN, CL, US, DIU, RE, RP y RM.

Dentro de este subproceso se efectúan las siguientes tareas:

A2.1 Asignar tareas a los miembros del equipo de trabajo acorde a su rol, basado en el Plan del Proyecto actual.

A2.2 Documentar o actualizar la Especificación de Requerimientos. Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos. Analizar los requerimientos identificados para determinar el alcance y viabilidad, considerando el ambiente del negocio del cliente o limitaciones del proyecto. Elaborar o actualizar el prototipo de interfaz de usuario. Generar o actualizar la Especificación de Requerimientos.

A2.3 Verificar la Especificación de Requerimientos.

A2.4 Corregir los defectos encontrados en la especificación de Requerimientos, basándose en el Reporte de Verificación, y obtener la aprobación de correcciones.

A2.5 Validar la Especificación de Requerimientos.

A2.6 Corrección de defectos encontrados en la especificación de requerimientos basado en el Reporte de Validación y obtención de aprobación de correcciones.

A2.7 Documentar la versión preliminar del Manual de Usuario o actualizar el Manual actual.



A2.8 Verificar el Manual de Usuario

A2.9 Corregir los defectos encontrados en el Manual de Usuario basado en el Reporte de Verificación y obtención de aprobación de las correcciones.

A2.10 Incorporar la Especificación de Requerimientos, y el Manual de usuario a la Configuración del Software en la línea base.

- **Arquitectura del Software y el Diseño Detallado:** Cubre los objetivos 3, 6 y 7. El subproceso de Arquitectura del Software y el Diseño Detallado transforma los requerimientos del software a la arquitectura del sistema y al diseño detallado. Este subproceso es realizado por: LT, AN, DIS, DIU y RE.

Dentro de este subproceso se efectúan las siguientes tareas:

A3.1 Asignar tareas a los integrantes del equipo de trabajo relacionadas a su rol de acuerdo con el Plan del Proyecto actual.

A3.2 Comprender la Especificación de Requerimientos

A3.3 Documentar o actualizar el Diseño del Software. Analizar la Especificación de Requerimientos para generar el Diseño de la Arquitectura, su arreglo en subsistemas y componentes define las interfaces internas y externas. Describe en detalle, la apariencia y el comportamiento de las interfaces, basándose en la Especificación de Requerimientos de una manera que los recursos para su implementación puedan ser previstos. Describe el detalle de los componentes que permiten su construcción en una forma evidente. Generar o actualizar el Registro de Trazabilidad.

A3.4 Verificar el diseño del Software, y el Registro de Trazabilidad

A3.5 Corregir los defectos encontrados en el Diseño del Software, y en el Registro de Trazabilidad basado en el Reporte de Verificación y obtención de aprobación de correcciones

A3.6 Incorporar el Diseño del Software y el Registro de Trazabilidad en la Configuración del software como parte de la línea base.

- **Construcción del Software:** Cumple con los objetivos 4, 6 y 7. El subproceso de Construcción del Software desarrolla el código y datos del software desde el Diseño del Software. El LT, PR, RP y RE ejecutan este subproceso.

Dentro de este subproceso se efectúan las siguientes tareas:



A4.1 Asignar tareas a los miembros del equipo de trabajo relacionados a sus roles, de acuerdo con el Plan del Proyecto actual.

A4.2 Comprender el Diseño del Software.

A4.3 Elaborar o actualizar los componentes del software. Implementar o actualizar Componentes base en la parte detallada del Diseño de Software. Definir y aplicar pruebas unitarias para verificar que la funcionalidad de cada componente concuerda con la parte detallada del Diseño del software. Corregir los defectos encontrados hasta lograr que las pruebas unitarias sean exitosas. Actualizar el Reporte de Trazabilidad incorporando los componentes construidos o modificados.

A4.4 Establecer los Casos de Prueba y los Procedimientos de Prueba para la integración de las pruebas basado en la Especificación de Requerimientos.

A4.5 Verificar los Casos de Prueba y los Procedimientos de Prueba.

A4.6 Corrección de defectos encontrados en los Casos de Prueba y los Procedimientos de Prueba basados en el Reporte de Verificación y obtener la aprobación de las correcciones.

A4.7 Incorporar Componentes, Reporte de Trazabilidad, Casos de Prueba y los Procedimientos de Prueba en la Configuración del Software como parte de la línea base.

- ***Integración del Software y Pruebas:*** Cubre los objetivos 5, 6 y 7. El subproceso de Integración del Software y Pruebas asegura que los componentes integrados del software satisfacen los requerimientos del software. Este subproceso lo llevan a cabo el LT, PR, RM y el RE.

Dentro de este subproceso se efectúan las siguientes tareas:

A5.1 Asignar tareas a los miembros del equipo de trabajo relacionados a sus roles, de acuerdo con el Plan del Proyecto actual.

A5.2 Entender los Casos de Prueba y los Procedimientos de Prueba.

A5.3 Integrar el Software utilizando Componentes y realizando pruebas con Casos de Prueba y Procedimientos de Prueba. Documentar los resultados en el Reporte de Incidentes de Pruebas. Corregir los defectos encontrados. Actualizar el Reporte de Trazabilidad, si es necesario.

A5.4 Documentar el Manual Operacional o actualizar el manual actual, si es necesario.

A5.5 Verificar el Manual Operacional, si es necesario.



A5.6 Corregir los defectos encontrados en el Manual Operacional basado en el Reporte de Verificación y obtener la aprobación de las correcciones, si es necesario.

A5.7 Documentar el Manual de Usuario o actualizar el actual, si es necesario.

A5.8 Verificar el Manual de Usuario, si es necesario.

A5.9 Corregir los defectos encontrados en el Manual de Usuario basado en el Reporte de Verificación y obtener la aprobación de las correcciones, si es necesario.

A5.10 Incorporar el Software, Registro de Trazabilidad, Reporte de Incidencias en Pruebas, Manual Operacional y Manual de Usuario, en la Configuración del Software como parte de la línea base.

- **Entrega del Producto:** Cubre los objetivos 6 y 7. El subproceso de Entrega del Producto, proporciona el producto del software integrado, al cliente. El RM, RE y LT ejecutan este subproceso.

Dentro de este subproceso se efectúan las siguientes tareas:

A6.1 Entender la Configuración del Software.

A6.2 Documentar el Manual de Mantenimiento o actualizarlo.

A6.3 Verificar el Manual de Mantenimiento.

A6.4 Corregir los defectos encontrados en el Manual de Mantenimiento basado en el Reporte de Verificación y obtener la aprobación de las correcciones, si es necesario.

A6.5 Incorporar el Manual de Mantenimiento como línea base para la Configuración del Software.

A6.6 Realizar la entrega final de acuerdo con el Protocolo de Entrega.

En la figura 2.3 [ISO 29110] se presenta el diagrama de flujo del proceso de Desarrollo de Software formado con los subprocesos: Inicio de Implementación del Software, Análisis de Requerimientos del Software, Arquitectura de Software y Diseño detallado, Construcción del Software, Integración de Software y Pruebas, y Entrega del Producto; junto con sus documentos de entrada, internos y de salida.

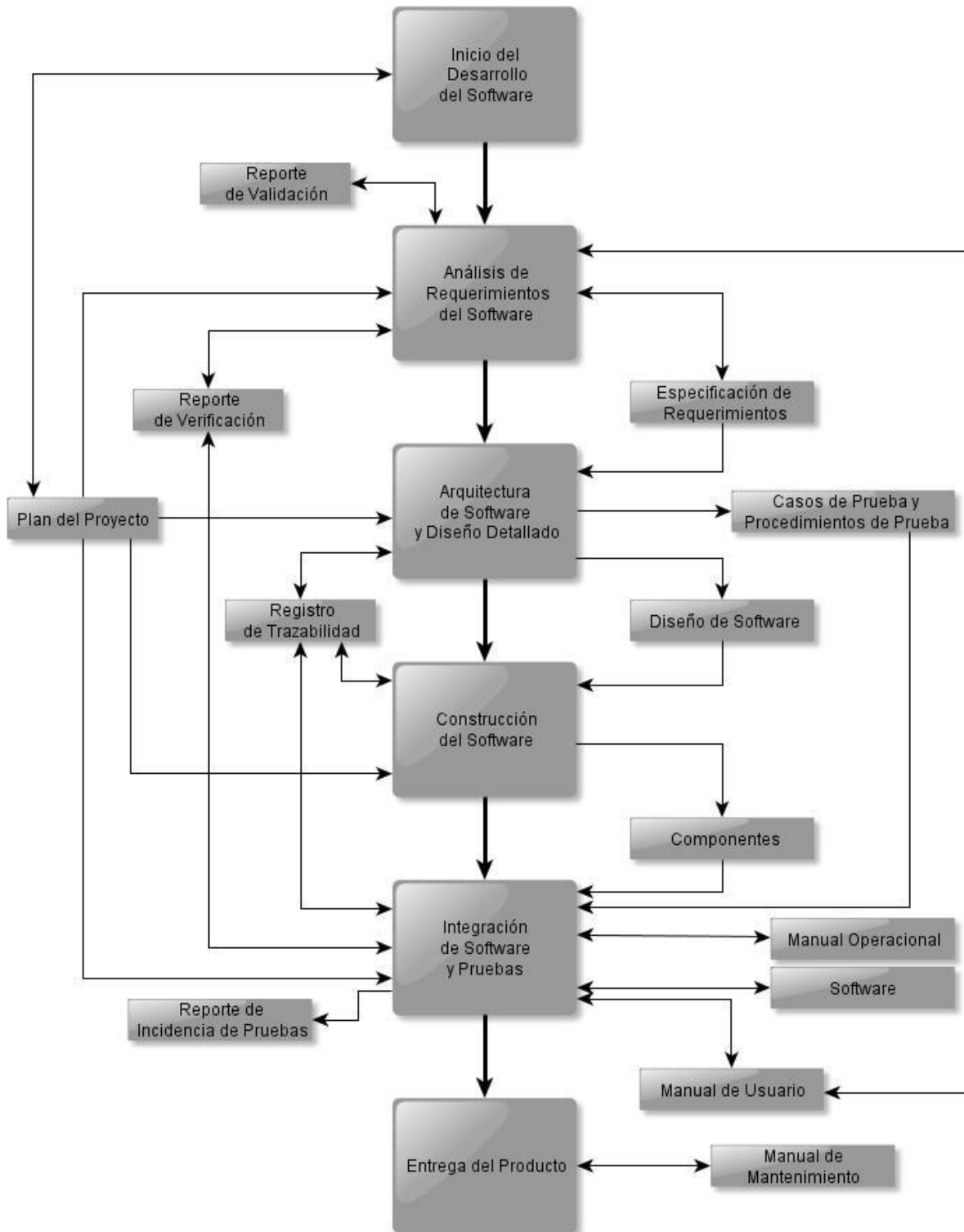


Fig. 2.3. Diagrama de Flujo del Desarrollo de Software.



Capítulo III Mapeo de las Historias de Usuario a la ISO/IEC 29110

El estándar ISO/IEC 29110 promueve la producción de un software con calidad mediante un proceso definido, su estructura provee guías que apoyan la ejecución de las tareas y actividades para obtener los productos propuestos. La propuesta de este trabajo tiene por objetivo enriquecerla con la utilización de técnicas para lograrlo.

Las metodologías ágiles se sugieren como una buena opción para generar las guías que necesitan las pequeñas y medianas empresas (PyME's) de software para implementar los procesos descritos por la norma, ya que han tenido un gran crecimiento en los últimos años [García, 2010] e incluyen una serie de técnicas ágiles.

Las Historias de Usuario son algunas de las técnicas ágiles que se especializan en la captura de los requerimientos del software, así como en su descripción, implementación y pruebas; y también son útiles para la planificación y estimación del proyecto. En este trabajo se propone una forma de apoyar los procesos de la ISO/IEC 29110 mediante la utilización de la técnica de Historias de Usuario.

Con el fin de crear una guía que utilice a las Historias de Usuario como técnica para realizar las tareas que describen las actividades descritas por los procesos de administración de proyectos y de desarrollo de software de la ISO/IEC 29110 [Cárdenas, 2010], se realizó un análisis de los conceptos propuestos por las Historias de Usuario, así como el de sus características, cómo implantarlas, y qué resultados se esperan de su ejecución, para poder identificar su adecuada correspondencia con las tareas que se describen dentro de las actividades de la ISO/IEC 29110.

Se analizó la estructura del perfil básico de la ISO/IEC 29110, se identificaron sus actividades, tareas y objetivos, así como los productos o entregables. Como consecuencia de este análisis se puede determinar que las Historias de Usuario podrían dirigir varias actividades con las cuales se obtengan los productos que se indican. Se encontró que la estimación y planeación, según las Historias de Usuario, pueden ser utilizadas para la planeación de todo el proyecto.

Posteriormente se realizaron diagramas que describen las relaciones entre las actividades de la ISO/IEC 29110 para identificar su correlación y poder identificar con mayor facilidad la aplicación de esta técnica en cada una de sus tareas de acuerdo a su definición, con ello también se logró obtener la información necesaria que se deberá integrar en la documentación como la requiere la norma en los procesos de Administración de Proyectos, Desarrollo y Mantenimiento de Software.

Una vez establecidas las técnicas que se podrían utilizar para cada tarea se definió la relación entre roles con apoyo en las responsabilidades de dicho rol y las tareas propuestas por las Historias de Usuario. Con ello se pretende proporcionar una estructura de comunicación al interior de la organización.



A continuación se describe el mapeo para la ejecución de los procesos de la ISO/IEC 29110 utilizando las Historias de Usuario. Se muestra como las actividades y los productos de trabajo propuestos por la ISO/IEC 29110 pueden ser implementadas siguiendo técnicas descritas por las Historias de Usuario. Se describirán y ejemplificarán cada uno de los posibles productos para el perfil básico de dicha norma.

La figura 3.1 muestra como las Historias de Usuario apoyan en la ejecución de la actividad “Planificación del Proyecto” del proceso “Administración de Proyectos” de la ISO/IEC 29110, siguiendo la siguiente nomenclatura, la cual se utilizara a lo largo de este trabajo:

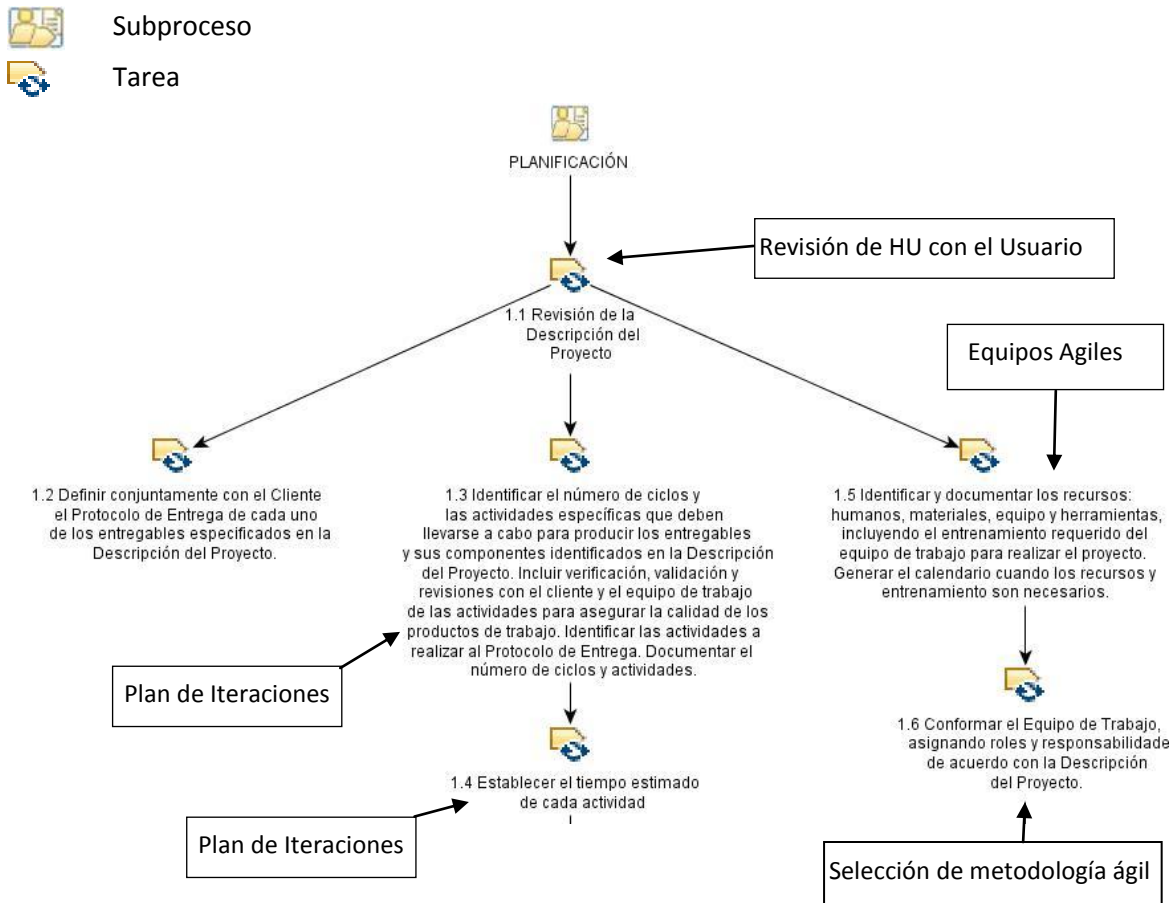


Fig. 3.1. Mapeo de tareas del subproceso Planificación del Proyecto con Historias de Usuario.

Al analizar las técnicas propuestas por las Historias de Usuario se identificó la información que se obtiene de aplicarlas. De este análisis se observó que existen tareas de la ISO/IEC 29110 a las cuales no es posible aplicar ninguna de las técnicas ágiles. En este caso se puede apoyar la realización de estas actividades mediante el uso de otras técnicas de la ingeniería de software (revisiones entre colegas, e.t.c.).



3.1 Administración de Proyectos

3.1.1 Planeación del Proyecto

A continuación se presenta una forma de cómo se pueden usar técnicas de historias de usuario dentro de las tareas de la ISO/IEC 29110, para ello se presenta un diagrama de la dependencia entre tareas (Fig. 3.2), y posteriormente se describe cada una de las tareas que pertenecen a la actividad de *Planeación del Proyecto* que forma parte del proceso de Administración de Proyectos relativo al perfil básico.

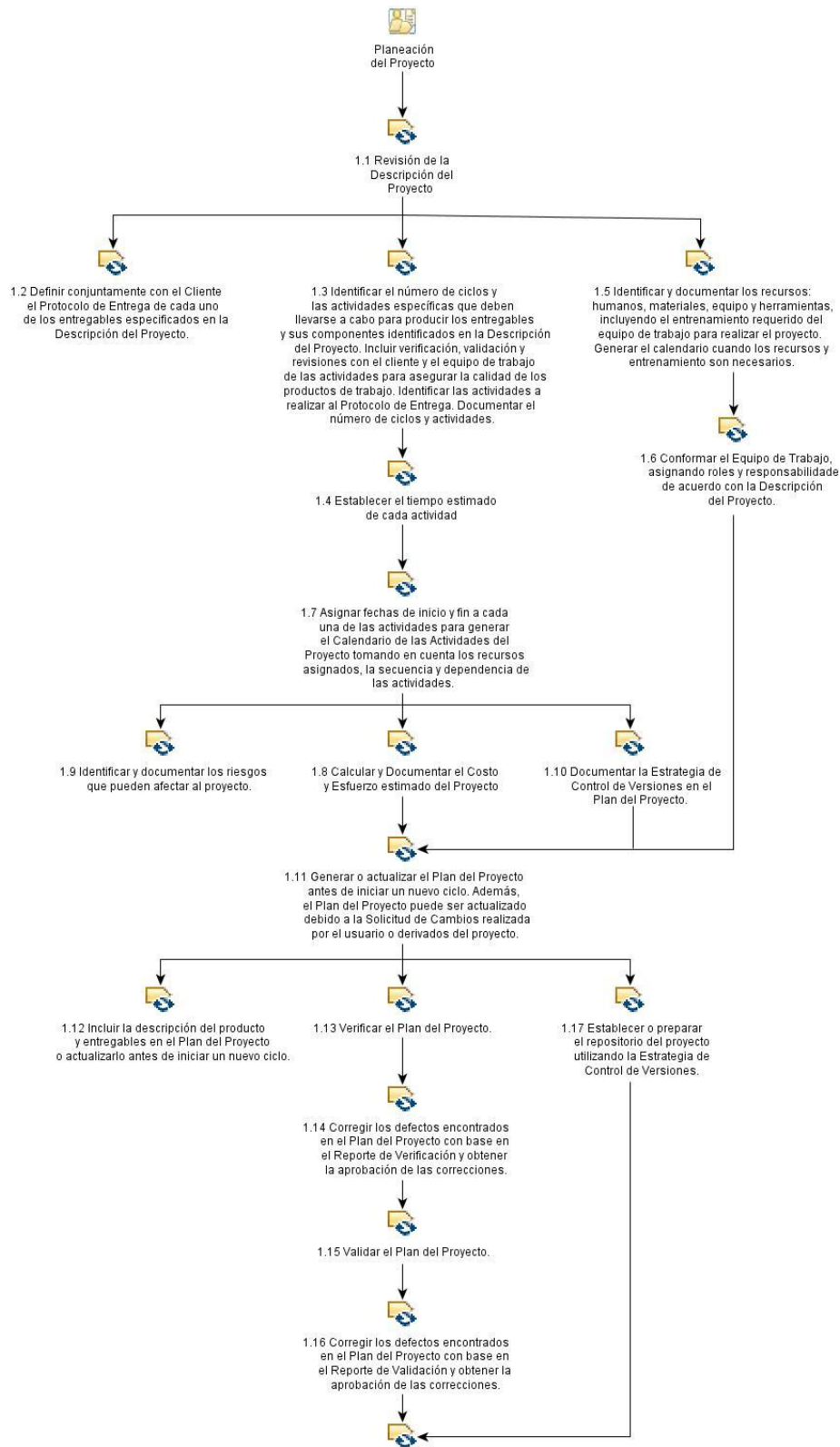


Fig. 3.2. Diagrama de dependencia de tareas de la Planificación del Proyecto.



A1.1 Revisar la Descripción del Proyecto

En este punto se puede realizar la definición de las Historias de Usuario. Se puede usar una lluvia de ideas para encontrar las historias de usuario básicas, con el objetivo de comprender las necesidades del negocio y las expectativas del cliente. Todas las observaciones se anotan y las más importantes para el cliente se incorporan dentro de la descripción del proyecto.

A1.2 Definir con el Cliente el Protocolo de Entrega de cada uno de los entregables especificados en la Descripción del Proyecto

Se establecen los parámetros y procedimientos para realizar la entrega del software funcional y su respectiva documentación. Acordar con el cliente que la entrega se hará por ciclos en los que se obtendrá siempre un software funcional, aunque no necesariamente útil para él.

A1.3 Identificar el número de ciclos y las actividades específicas que deben llevarse a cabo para producir los entregables y sus componentes identificados en la Descripción del Proyecto. Incluir verificación, validación y revisiones con el cliente y el equipo de trabajo de las actividades para asegurar la calidad de los productos de trabajo. Identificar las actividades a realizar al Protocolo de Entrega. Documentar el número de ciclos y actividades.

Para identificar el número de ciclos se estima el número de iteraciones y la cantidad de trabajo a realizar dentro de cada una de ellas. Entre el equipo de trabajo y el cliente se establece el periodo de tiempo que durará cada iteración, y se debe aclarar que la duración de dichas iteraciones se deberá mantener de manera constante.

A1.4 Establecer el tiempo estimado de cada actividad

El equipo de trabajo puede estimar o establecer la cantidad de trabajo que puede realizar a lo largo de una iteración.

A1.5 Identificar y documentar los recursos: humanos, materiales, equipo y herramientas, incluyendo el entrenamiento requerido del equipo de trabajo para realizar el proyecto. Generar el calendario cuando los recursos y entrenamiento son necesarios.

Se conforma el equipo de trabajo el cual es auto organizado, comparten información, y que comprenden su responsabilidad. Se determina si cuenta con la preparación y experiencia necesaria, se documenta esta información y se identifican las necesidades de capacitación.

A1.6 Conformar el Equipo de Trabajo, asignando roles y responsabilidades de acuerdo con la Descripción del Proyecto.

Esta tarea se implementa dependiendo de la metodología ágil que se haya elegido para trabajar XP, Scrum, etc.



A1.7 Asignar fechas de inicio y fin a cada una de las actividades para generar el *Calendario de las Actividades del Proyecto* tomando en cuenta los recursos asignados, la secuencia y dependencia de las actividades.

Se lleva a cabo una reunión con el cliente para realizar la estimación de cada una de las historias de usuario ya sea que se estime con los puntos por historia o por días ideales.

El número de historias que puede llevar a cabo un equipo en una iteración, se llama *velocidad* y servirá para planear las siguientes iteraciones. Al terminar la primera iteración se calcula cuántas Historias puede desarrollar el equipo en el plazo de una iteración.

La determinación de qué historias de usuario se incluyen en cada iteración se llama *priorización* y la realiza el usuario, con apoyo del equipo de desarrollo, tratando de llevar una cierta lógica en la secuencia. Si existe alguna discusión en este punto, el cliente siempre tendrá la última palabra.

Se debe tomar en cuenta que las Historias de usuario de mayor riesgo o las que tengan dependencia con otras historias son las que se deben implementar primero, además de los factores técnicos.

En este momento se pueden estimar las fechas de inicio y finalización del proyecto, así como un *Calendario de actividades* basado en el plan de Liberación.

A1.8 Calcular y Documentar el Costo y Esfuerzo estimado del Proyecto

Se debe informar al usuario el costo en tiempo y esfuerzo.

A1.9 Identificar y documentar los riesgos que pueden afectar al proyecto.

Se consideran los riesgos como internos, externos, tecnológicos, de procesos, de personal. Se identifican los posibles riesgos que corre el proyecto y se analizan, posteriormente se crea un plan para minimizarlos o eliminarlos. Esta planificación se integra dentro de cada iteración para poder identificarlos y darles seguimiento.

Se puede utilizar la tabla 3.1 como una forma de registrar dichos riesgos:

Descripción	Prioridad	Impacto	Plan de Contingencia

Tabla 3.1. Identificación de riesgos.

Algunos riesgos a considerarse cuando se utilizan historias de usuario se mencionan en la sección 1.7 del capítulo I.



A1.10 Documentar la Estrategia de Control de Versiones en el Plan del Proyecto.

El control de versiones se lleva a cabo conforme a las iteraciones acordadas, es decir, cuando se termina una iteración se libera una versión intermedia. Las versiones completas podrán especificarse según el plan de liberaciones.

La estrategia de control de versiones debe contar con un mecanismo de almacenaje que describa el tipo de información se almacenará, se debe contemplar la posibilidad de realizar cambios en la información almacenada, se deberá contar con registros históricos de los movimientos realizados. Se deberá establecer si la información se almacenará de forma centralizada o distribuida.

A1.11 Generar o actualizar el Plan del Proyecto antes de iniciar un nuevo ciclo. Además, el Plan del Proyecto puede ser actualizado debido a la Solicitud de Cambios realizada por el usuario o derivados del proyecto.

Para integrar el Plan del Proyecto se deben integrar los siguientes:

- Alcance del Proyecto
- Descripción del Proyecto
 - Ciclos o iteraciones
 - Actividades
 - Estimados de cada iteración y actividad
- Protocolo de entrega
- Recursos: humanos, materiales y herramientas.
- Equipo de trabajo: Roles y responsabilidades
- Estrategia de Control de Versiones
- Plan de Riesgos
- Plan de Iteraciones
- Plan de Liberación

A1.12 Incluir la descripción del producto y entregables en el Plan del Proyecto o actualizarlo antes de iniciar un nuevo ciclo.

Se puede realizar un Plan del proyecto general y uno por cada iteración.

Se analiza la posibilidad de re-estimar las historias de usuario, en caso de ser necesario, para ajustar los planes de liberación e iteración al calendario de trabajo. También se considera la posibilidad de integrar las historias de usuario incompletas dentro de una nueva iteración, realizando las modificaciones necesarias al plan de iteración. Todo se debe realizar antes comenzar un nuevo ciclo y se actualiza el Plan del proyecto.

A1.13 Verificar el Plan del Proyecto.

En este punto se verifica el Plan del proyecto.



A1.14 Corregir los defectos encontrados en el *Plan del Proyecto* con base en el *Reporte de Verificación* y obtener la aprobación de las correcciones.

Una vez que se tienen identificados los defectos del Plan del Proyecto, se corrigen y se reintegra el documento.

A1.15 Validar el Plan del Proyecto.

Validar todo el plan y la descripción total del proyecto con el cliente.

Se puede generar una lista de verificación para ir comprobando que los objetivos del proyecto sean cubiertos.

A1.16 Corregir los defectos encontrados en el *Plan del Proyecto* con base en el *Reporte de Validación* y obtener la aprobación de las correcciones.

Se modifica el Plan corrigiendo los defectos y se obtiene la aprobación de las correcciones.

A1.17 Establecer o preparar el repositorio del proyecto utilizando la *Estrategia de Control de Versiones*.

Utilizando la propuesta del control versión de A1.10, se define la estructura del repositorio.

A1.18 Incorporar el Plan del Proyecto dentro del Repositorio del Proyecto en la línea base.

Se incorpora el Plan del Proyecto en el Repositorio.

En la tabla 3.2 se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Planificación del Proyecto, incluyendo los roles responsables. A lo largo del documento se respetará la siguiente nomenclatura: las columnas indican el rol encargado de realizar las tareas. Los renglones indican las tareas que forman al subproceso. La información que se encuentra en cada celda indica la técnica de historias de usuario que se puede aplicar para realizar cada tarea, las celdas que se encuentran en blanco indican que no existe técnica para aplicar.



Planeación del Proyecto			
	AP	AP/LT	AP/CL
A1.1		Historias de Usuario Lluvia de ideas	
A1.2			Plan de Iteraciones
A1.3	Estimación de iteraciones Duración de iteración		
A1.4		Estimación de cantidad de trabajo por iteración	
A1.5	Se crea el equipo de trabajo		
A1.6	Se conforma el equipo de trabajo Se elige la metodología de trabajo		
A1.7	Estimación de Historias Se estima la velocidad Se da prioridad a las Historias		
A1.8			
A1.9		Historias de Riesgo	
A1.10	Plan de Iteraciones		
A1.11	Plan de iteraciones Plan de Liberación		
A1.12		Re-estimación de Historias	
A1.13			
A1.14			
A1.15			Lista de Verificación
A1.16			
A1.17			
A1.18			

Tabla 3.2. Relación de Actividades, Roles y Técnicas Ágiles dentro de Planeación del Proyecto.



3.1.2 Ejecución del Plan del Proyecto

Ahora se presenta la forma de cómo se implementan las técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se presenta la dependencia de las tareas (Fig. 3.3), posteriormente se describen las tareas que pertenecen a la actividad de *Ejecución del Plan del Proyecto* que forma parte del proceso de Administración de Proyectos relativo al perfil básico.



Fig. 3.3. Dependencia de tareas de la Ejecución del Plan del Proyecto.



A2.1 Acordar con el Líder Técnico la asignación de tareas al *Equipo de Trabajo*.

Se comienza por realizar el Plan del proyecto de esta iteración y se realiza la estimación de las historias a desarrollarse dentro de esta iteración.

Dejar que los programadores tomen las historias que desean realizar, haciendo hincapié en que son responsables de realizar completamente las historias que han seleccionado dentro de la iteración planificada.

Se debe tener en cuenta que algunas veces los programadores no terminarán a tiempo las historias que seleccionaron, por ello debe existir una mentalidad de compañerismo dentro del equipo de trabajo, para que con ello si algún programador ha terminado sus historias entonces apoye al que se encuentra retrasado.

A2.2 Revisar el Plan del Proyecto y registrar los datos actuales en el Registro de Estado de Progreso

Se realizan reuniones diariamente para que cada integrante del Equipo de Trabajo pueda informar de sus avances y tiempos dedicados, y se comparan con los estimados. Toda esta información se conjunta dentro del Registro del Estado de Progreso.

A2.3 Registrar, analizar y evaluar la solicitud de cambio para costo, calendario e impacto técnico. Incluir los cambios aprobados en el Plan del Proyecto. La solicitud de cambio puede ser iniciada externamente por el cliente o internamente por el equipo de trabajo.

Se registran los cambios en la historia de usuario ya sea en la misma tarjeta o rescribiéndola, tomando en cuenta la modificación de las pruebas de aceptación.

Entre el cliente y el equipo de trabajo se analizan los riesgos que conllevan los cambios solicitados, así como sus beneficios, para determinar si deben realizarse tomando en cuenta los aspectos de calendario, costo e impacto técnico.

En caso de existir una modificación que afecte las fechas de entrega, se puede hacer uso de la división de historias para acelerar su implementación. También se deben analizar los cambios dentro del plan de liberaciones. En el plan de iteraciones se harán los cambios necesarios y se realizarán al comienzo de la iteración correspondiente. Se analizan los planes de iteración para verificar su convergencia.

A2.4 Conducir reuniones de revisión con el equipo de trabajo y el cliente, y se registran los acuerdos.

Periódicamente se deben realizar reuniones de revisión del estado de avance en que se encuentran las historias de usuario, si las historias no han sido bien comprendidas por los programadores, se solicita al cliente que aclare las dudas. Si existen nuevas historias de usuario, se integran de acuerdo con el plan de iteraciones.



En caso de que exista una gran dependencia entre historias, se analiza la integración de éstas dentro de otra iteración o se analiza su división.

A2.5 Realizar el *Respaldo* de acuerdo a la *Estrategia de Control de Versiones*.

Se realiza el respaldo dentro del repositorio según la estructura que haya sido definida, esto se puede realizar al finalizar cada iteración.

A2.6 Realizar la recuperación del *Repositorio* usando el *Respaldo del Repositorio*, en caso de ser necesario.

Se respalda el repositorio periódicamente.

A continuación se presenta la tabla 3.3, como relación de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Ejecución del Plan del Proyecto, incluyendo los roles responsables.

Ejecución del Plan del Proyecto			
	AP	AP/LT	AP/CL/ET
A2.1		Estimación de Historias Asignación de Historias	
A2.2			
A2.3		Re-escritura de Historias Plan de Iteración Plan de Liberación	
A2.4			Reunión con Cliente Plan de Iteraciones
A2.5	Plan de Iteraciones		
A2.6			

Tabla 3.3. Relación de Actividades, Roles y Técnicas Ágiles dentro de Ejecución del Plan del Proyecto.



3.1.3 Evaluación y Control del Proyecto.

Esta sección muestra la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se presenta la dependencia de las tareas (Fig. 3.4), posteriormente se describen las tareas que pertenecen a la actividad de *Evaluación y Control del Proyecto* que forma parte del proceso de Administración de Proyectos relativo al perfil básico.



Fig. 3.4. Dependencia de tareas de Evaluación y Control del Proyecto.

A3.1 Evaluar la información requerida por el Plan del Proyecto. Las acciones correctivas o solicitudes de cambio son establecidas y seguidas.

Se analizan las historias de usuario y de ser necesario se integran nuevas historias de usuario que haya generado la solicitud de cambio o las acciones correctivas con sus correspondientes pruebas de aceptación.

Se analizan los planes de iteración de acuerdo a la estimación que se realice respecto a los cambios solicitados. Al finalizar la iteración correspondiente se verifica que los cambios se hayan realizado, y que las pruebas hayan sido exitosas.

En la tabla 3.4 se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Evaluación y Control del Proyecto, incluyendo los roles responsables.

Evaluación y Control del Proyecto	
	AP
A3.1	Análisis de Historias Análisis de Plan de Iteración

Tabla 3.4. Relación de Actividades, Roles y Técnicas Ágiles dentro de Evaluación y Control del Proyecto.



3.1.4 Cierre del Proyecto.

Aquí observamos la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se presenta la dependencia de las tareas (Fig. 3.5), posteriormente se describen las tareas que pertenecen a la actividad de *Cierre del Proyecto* que forma parte del proceso de Administración de Proyectos relativo al perfil básico.



Fig. 3.5. Dependencia de tareas de Cierre del Proyecto.

A4.1 Formalizar la finalización del ciclo o del Proyecto acorde al Protocolo de entrega establecido en el Plan del Proyecto, y conseguir el Documento de Aceptación.

Se realiza una reunión con el equipo de trabajo para verifica que las historias de usuario correspondientes al ciclo o en su totalidad al proyecto, se encuentren completamente implementadas y validadas por el usuario. Todo esto de acuerdo a los planes de iteración y/o liberación.

Se puede realizar una reunión de lecciones aprendidas y se integra a la base de conocimientos.

La tabla 3.5 presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte del Cierre del Proyecto, incluyendo los roles responsables.

Cierre del Proyecto	
	AP/CL
A4.1	Verificación de Historias Plan de Iteración

Tabla 3.5. Relación de Actividades, Roles y Técnicas Ágiles dentro del Cierre del Proyecto.



4.2 Desarrollo de Software

4.2.1 Inicio del Desarrollo de Software

Se presenta la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se presenta la dependencia de las tareas (Fig. 3.6), posteriormente se describen las tareas que pertenecen a la actividad de *Inicio del Desarrollo de Software* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.



Fig. 3.6. Dependencia de tareas de Inicio del Desarrollo de Software.

A1.1 Revisión del Plan de Proyecto actual por los integrantes del equipo de trabajo, a fin de lograr un entendimiento común y obtener su compromiso con el proyecto.

En esta actividad el equipo de desarrollo y el cliente deben analizar tanto el plan de iteración como el de liberación. Se determina que el equipo deberá reunirse al inicio de cada iteración para revisar el estado de las historias, y si existen se integrarán nuevas historias de usuario. Se analiza si la estimación de las historias de usuario ha sido adecuada.

Se analiza si las historias que se incluyeron en esta iteración, integran el mayor valor posible del proyecto. También se deben de tomar en cuenta los diferentes tipos de historias de usuario, es decir, las que son de mayor prioridad para el usuario y que son fundamentales para el sistema, y deben integrarse de manera inmediata. Las que son importantes pero no se tiene una solución a corto plazo. Las que se pueden dejar fuera en caso de llegar al tiempo límite, y las que se podrían incluir dentro de la siguiente iteración.

Es en este momento cuando el cliente puede cambiar la prioridad de las historias a desarrollar, y el equipo de trabajo puede expresar sus dudas o comentarios sobre la complejidad de dichas historias.

A continuación la tabla 3.6 se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte del Inicio del Desarrollo de Software, incluyendo los roles responsables.

Inicio del Desarrollo de Software	
	ET
A1.1	Se analizan planes de iteración y liberación Se establece el periodo de reuniones Integración de nuevas historias Priorización de historias

Tabla 3.6. Relación de Actividades, Roles y Técnicas Ágiles dentro del Inicio del Desarrollo de Software.



4.2.2 Análisis de Requerimientos de Software

En este punto se presenta la manera de cómo se implementan las técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se presenta la dependencia de las tareas (Fig. 3.7), posteriormente se describen las tareas que pertenecen a la actividad de *Análisis de Requerimientos de Software* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.

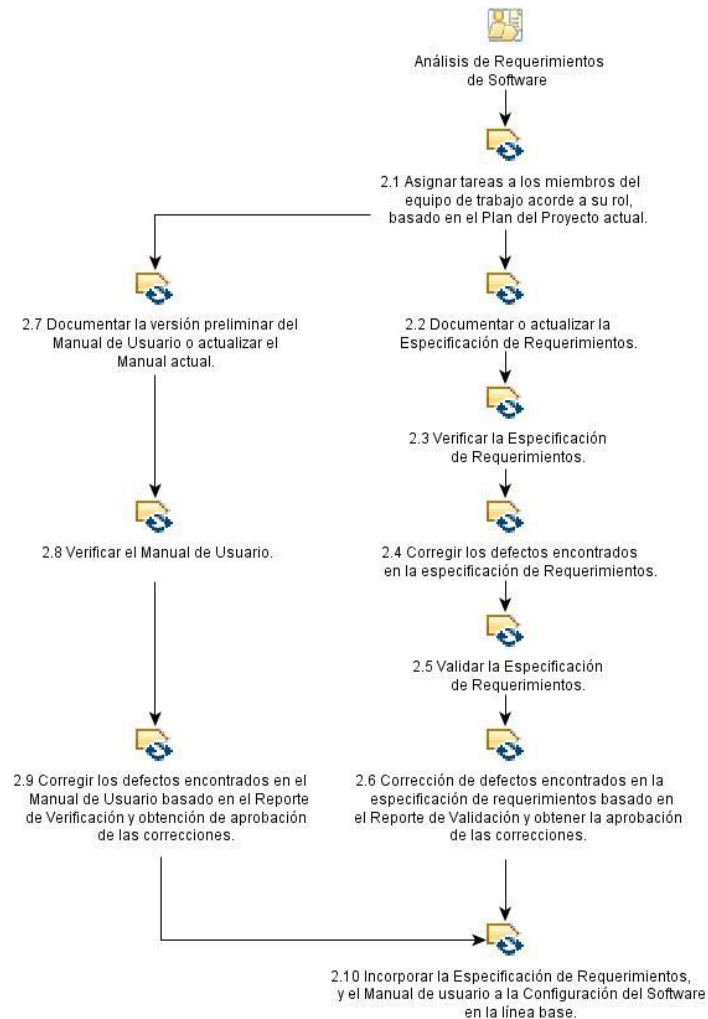


Fig. 3.7. Dependencia de tareas de Análisis de Requerimientos de Software.

A2.1 Asignar tareas a los miembros del equipo de trabajo acorde a su rol, basado en el Plan del Proyecto actual.

Se realiza una reunión con el equipo de trabajo y el cliente, es en este momento cuando los miembros del equipo toman la decisión de hacerse responsables por las historias que han escogido y que se deben realizar de acuerdo al plan de iteración. El equipo se auto-organiza y los integrantes se asignan varios roles como: Arquitecto, Programador, Tester, Diseñador de



Interfaces, etc. Para ello se realiza una revisión de las historias de usuario incluidas dentro de la iteración actual. Este tipo de reuniones deben realizarse al inicio de cada iteración.

A2.2 Documentar o actualizar la Especificación de Requerimientos. Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos. Analizar los requerimientos identificados para determinar el alcance y viabilidad, considerando el ambiente del negocio del cliente o limitaciones del proyecto. Elaborar o actualizar el prototipo de interfaz de usuario. Generar o actualizar la Especificación de Requerimientos.

Una vez que el cliente haya redactado las Historias de Usuario que completen el proyecto, derivadas de la lluvia de ideas, el equipo de trabajo podrá realizar la estimación de dichas historias. El cliente las podrá priorizar y determinar en qué iteración se integrarán.

Los programadores podrán realizar la selección de historias, se realiza una relación entre sus nombres y las historias de usuario que eligieron. El programador debe estar consciente de la cantidad de trabajo que está adquiriendo. Ahora los programadores podrán pedirle al cliente que aclare sus dudas de las historias que deben implementar, así podrán analizar la complejidad de la historia de usuario y decidir si la dividen, si necesitan un compañero de programación, etc.

El cliente puede integrar nuevas historias si lo cree necesario, su integración se determinará de acuerdo al plan de iteraciones.

Se establecerán las pruebas bajo las cuales el cliente aceptará que la historia ha sido implementada correctamente. Estas pruebas se anotarán al reverso de la tarjeta de cada historia. También se realizarán cambios a las pruebas de aceptación si el cliente las solicita, además de verificar la dependencia entre ellas.

A2.3 Verificar la Especificación de Requerimientos.

Se realiza una revisión con el cliente de las historias de usuario que se implementarán en la presente iteración. El equipo de trabajo debe exponerle su conocimiento de los objetivos del proyecto, con ello se logra conocer a profundidad el conocimiento adquirido por parte de los programadores y con ello verificar si se ha comprendido los requerimientos del sistema.

Se determina que los desarrolladores hayan aclarado las dudas, que hayan determinado la forma en la cual desarrollarán sus historias de usuario, y que hayan descrito las pruebas de aceptación correctamente.

A2.4 Corregir los defectos encontrados en la especificación de Requerimientos, basándose en el Reporte de Verificación, y obtener la aprobación de correcciones.

Si existen mal entendidos en las historias de usuario, se pueden modificar directamente en la tarjeta o se pueden incluir como notas. Esto también se puede realizar a las pruebas de aceptación de dicha historia.



A2.5 Validar la Especificación de Requerimientos.

El equipo de trabajo valida con el cliente que las historias integradas en dicha iteración sean claras, para que no contengan ambigüedades, así como las pruebas de aceptación. También se valida que no se realicen cambios una vez iniciada la iteración.

A2.6 Corrección de defectos encontrados en la especificación de requerimientos basado en el Reporte de Validación y obtener la aprobación de las correcciones.

El equipo de trabajo verifica junto con el cliente que las historias sean claras y que no contengan ambigüedades. Una vez aceptadas las historias por parte del cliente, se obtiene la aprobación de las correcciones. Nuevamente no se aceptan cambios durante la iteración, sino a su final.

A2.7 Documentar la versión preliminar del Manual de Usuario o actualizar el Manual actual.

Se genera el Manual de Usuario según las historias de Usuario a desarrollar en esta iteración y se conjunta con las versiones de iteraciones anteriores.

A2.8 Verificar el Manual de Usuario.

Se revisa el Manual de usuario con el usuario y se verifica que sea claro.

A2.9 Corregir los defectos encontrados en el Manual de Usuario basado en el Reporte de Verificación y obtención de aprobación de las correcciones.

Se realiza la corrección de defectos del Manual de Usuario. Se reintegra el manual.

A2.10 Incorporar la Especificación de Requerimientos, y el Manual de Usuario a la Configuración del Software en la línea base.

Se incorporan el Manual de Usuario y la Especificación de Requerimientos al depósito del proyecto.



A continuación en la tabla 3.7, se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte del Análisis de Requerimientos de Software, incluyendo los roles responsables.

Análisis de Requerimientos de Software						
	AN/LT	AN/CL/US/DIU	RE	AN/DIU	CL/RP/US	AN/DIU
A2.1	Asignación de Historias Asignación de roles					
A2.2		Estimación de historias Priorización de historias Pruebas de aceptación				
A2.3			Revisión con Cliente			
A2.4				Modificación de historias y pruebas		
A2.5					Validación con cliente de historias claras	
A2.6						Verificación con cliente de historias claras

Tabla 3.7. Relación de Actividades, Roles y Técnicas Ágiles dentro del Análisis de Requerimientos de Software.

Nota: Las tareas 2.7, 2.8, 2.9 y 2.10 no se incluyen en la tabla pues no existe técnica ágil que se pueda aplicar para ellas.



4.2.3 Arquitectura de Software y Diseño Detallado

Ahora se presenta la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se describen, en la figura 3.8, las tareas que pertenecen a la actividad de la *Arquitectura y Diseño Detallado de Software* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.



Fig. 3.8. Dependencia de tareas de Arquitectura de Software y Diseño Detallado.



A3.1 Asignar tareas a los integrantes del equipo de trabajo relacionadas a su rol de acuerdo con el Plan del Proyecto actual.

Cada desarrollador selecciona las Historias de Usuario que va a desarrollar. Aquí se determina que el equipo de desarrollo se auto-organice.

A3.2 Comprender la Especificación de Requerimientos.

En esta tarea se revisan las Historias de Usuario asignadas a cada desarrollador para aclarar nuevas dudas y comentarios. El equipo de desarrollo debe contar con el apoyo del cliente para aclarar sus dudas y comprender mejor la visión que se tiene del proyecto.

A3.3 Documentar o actualizar el Diseño del Software. Analizar la Especificación de Requerimientos para generar el Diseño de la Arquitectura, su arreglo en subsistemas y componentes definen las interfaces internas y externas. Describe en detalle, la apariencia y el comportamiento de las interfaces, basándose en la Especificación de Requerimientos de una manera que los recursos para su implementación puedan ser previstos. Describe el detalle de los componentes que permiten su construcción en una forma evidente. Generar o actualizar el Registro de Trazabilidad.

Se plantea una arquitectura inicial basándose en: funcionalidad, atributos de calidad, e.t.c., dependiendo del proyecto. En cada iteración se reevalúa y se puede decidir si se reorganiza.

Se puede verificar si el diseño del software sigue siendo el adecuado para el proyecto. Se genera el Registro de Trazabilidad.

A3.4 Verificar el Diseño del Software, y el Reporte de Trazabilidad.

Al inicio de cada iteración, se deben analizar las historias de usuario a desarrollar, y se deben integrar a los módulos de la arquitectura establecida.

A3.5 Corregir los defectos encontrados en el Diseño del Software, y en el Registro de Trazabilidad basado en el Reporte de Verificación y obtención de aprobación de correcciones.

Se corrigen los defectos encontrados en el Diseño.

A3.6 Incorporar el Diseño del Software y el Registro de Trazabilidad en la Configuración del software como parte de la línea base.

Se incorpora el Diseño del Software y el Registro de Trazabilidad al depósito del proyecto.



La tabla 3.8 presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Arquitectura de Software y Diseño Detallado, incluyendo los roles responsables.

Arquitectura de Software y Diseño Detallado					
	AN/DIS/LT	AN/DIS	AN/DIS/DIU	RE	LT
A3.1	Asignación de historias				
A3.2		Revisión de historias con Cliente			
A3.3			Reorganizar arquitectura inicial		
A3.4				Análisis e integración de historias a la arquitectura	
A3.5					
A3.6					

Tabla 3.8. Relación de Actividades, Roles y Técnicas Ágiles dentro de la Arquitectura de Software y Diseño Detallado.



4.2.4 Construcción del Software

Aquí se presenta la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se describen (en la figura 3.9) las tareas que pertenecen a la actividad de la *Construcción del Software* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.

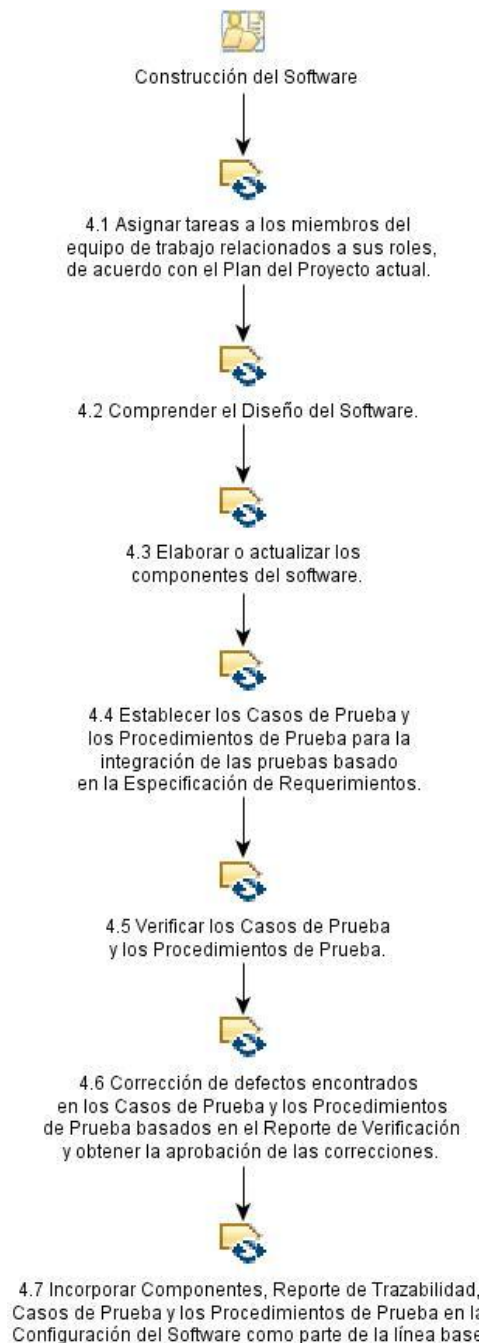


Fig. 3.9. Dependencia de tareas de Arquitectura de Software y Diseño Detallado.



A4.1 Asignar tareas a los miembros del equipo de trabajo relacionados a sus roles, de acuerdo con el Plan del Proyecto actual.

Cada desarrollador desarrolla las Historias de Usuario que eligió y realiza las pruebas de aceptación de dicha historia.

A4.2 Comprender el *Diseño del Software*.

El desarrollador debe entender a relación entre los módulos de la arquitectura.

A4.3 Elaborar o actualizar los componentes del software. Implementar o actualizar Componentes base en la parte detallada del Diseño de Software. Definir y aplicar pruebas unitarias para verificar que la funcionalidad de cada componente concuerda con la parte detallada del Diseño del software. Corregir los defectos encontrados hasta lograr que las pruebas unitarias sean exitosas. Actualizar el Reporte de Trazabilidad incorporando los componentes construidos o modificados.

Cada miembro del equipo de trabajo desarrolla sus historias de usuario, de acuerdo al diseño y arquitectura previamente seleccionados. Se debe tener en cuenta la cantidad de historias de usuario que han quedado pendientes de otras iteraciones.

Se pueden comenzar a realizar las pruebas de aceptación, estas se realizan hasta poder verificar la completa implementación.

A4.4 Establecer los *Casos de Prueba* y los *Procedimientos de Prueba* para la integración de las pruebas basado en la *Especificación de Requerimientos*.

Se pueden comenzar a realizar las pruebas de aceptación, estas se realizan hasta poder verificar la completa implementación.

En esta tarea se deben de considerar las pruebas de aceptación diseñadas por el cliente y el responsable de pruebas. Estas pruebas fueron verificadas y validadas por el cliente en el análisis de requerimientos. Se define el orden de integración de las Historias de Usuario según la especificación de requerimientos.

A4.5 Verificar los *Casos de Prueba* y los *Procedimientos de Prueba*.

Se pueden ejecutar las pruebas de aceptación y verificar que sean las adecuadas para dicha historia de usuario.

A4.6 Corrección de defectos encontrados en los *Casos de Prueba* y los *Procedimientos de Prueba* basados en el Reporte de Verificación y obtener la aprobación de las correcciones.

En caso de que existan defectos en las pruebas de aceptación, se corrigen en conjunto con el cliente. Si éste está conforme, se obtiene su aprobación a las correcciones.



A4.7 Incorporar Componentes, Reporte de Trazabilidad, Casos de Prueba y los Procedimientos de Prueba en la Configuración del Software como parte de la línea base.

Se integran a la Configuración del Software las historias de usuario completadas y validadas por el usuario a través de sus pruebas de aceptación, así como el estado que guardan las historias incompletas y que serán integradas en otra iteración. También se integra el plan de iteración actualizado.

A continuación en la tabla 3.9, se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Construcción del Software, incluyendo los roles responsables.

Construcción del Software				
	PR	RE	LT	RP
A4.1			Desarrollo de historias	
A4.2				
A4.3	Pruebas de Aceptación			
A4.4				Pruebas de Aceptación Orden de integración
A4.5		Ejecución de pruebas de aceptación		
A4.6				Corrección de pruebas de aceptación
A4.7			Plan de iteración	

Tabla 3.9. Relación de Actividades, Roles y Técnicas Ágiles dentro de la Construcción del Software.



4.2.5 Integración de Software y Pruebas

Ahora se presenta la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se describen las tareas (Fig. 3.10) que pertenecen a la actividad de la *Integración y Pruebas de Software* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.

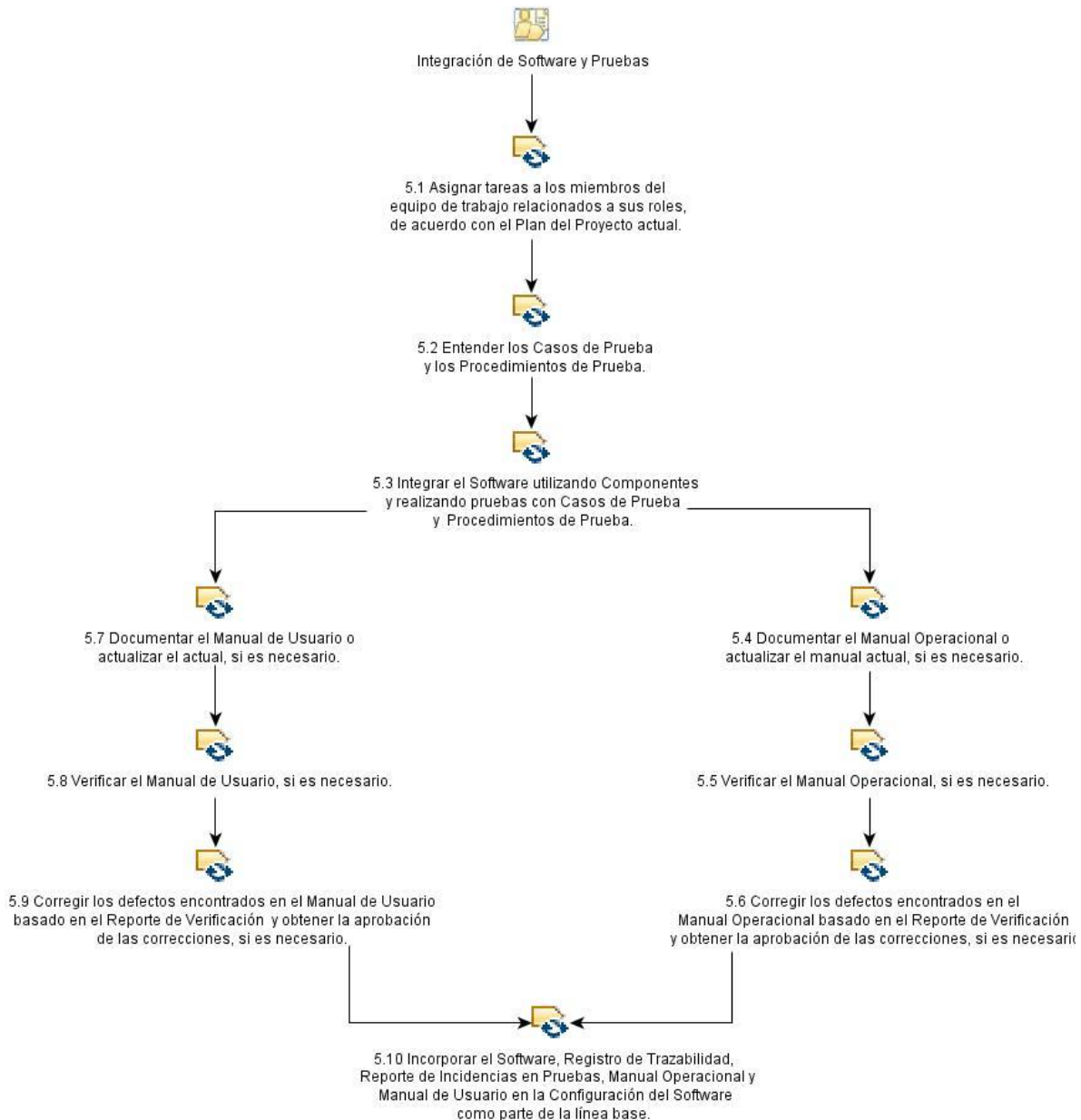


Fig. 3.10. Dependencia de tareas de Integración de Software y Pruebas.



A5.1 Asignar tareas a los miembros del equipo de trabajo relacionados a sus roles, de acuerdo con el Plan del Proyecto actual.

Al final de cada iteración, se realizarán las pruebas de aceptación, que han sido definidas para cada historia de usuario que haya sido completada, para que ésta sea validada.

A5.2 Entender los *Casos de Prueba* y los *Procedimientos de Prueba*.

Se deben analizar las pruebas de aceptación descritas para las historias de usuario.

A5.3 Integrar el Software utilizando *Componentes* y realizando pruebas con *Casos de Prueba* y *Procedimientos de Prueba*. Documentar los resultados en el *Reporte de Incidentes de Pruebas*. Corregir los defectos encontrados. Actualizar el *Reporte de Trazabilidad*, si es necesario.

Se realiza la integración de cada una de las historias de usuario. Posteriormente se realizan pruebas en conjunto, incluyendo a las implementadas en iteraciones anteriores. Se corrigen los defectos encontrados.

A5.4 Documentar el *Manual Operacional* o actualizar el manual actual, si es necesario.

El plan de iteraciones se toma como guía para el manual operacional, se incluyen las historias que han sido desarrolladas completamente.

A5.5 Verificar el *Manual Operacional*, si es necesario.

Se puede realizar una prueba al manual operacional.

A5.6 Corregir los defectos encontrados en el *Manual Operacional* basado en el *Reporte de Verificación* y obtener la aprobación de las correcciones, si es necesario.

Se corrigen los defectos del manual operacional. Se obtiene la aprobación de correcciones.

A5.7 Documentar el *Manual de Usuario* o se actualiza, si es necesario.

Se genera el Manual de Usuario.

A5.8 Verificar el *Manual de Usuario*, si es necesario.

Se verifica el Manual del Usuario con el cliente y se obtiene la aprobación de correcciones.

A5.9 Corregir los defectos encontrados en el *Manual de Usuario* basado en el *Reporte de Verificación* y obtener la aprobación de las correcciones, si es necesario.

Se corrigen los defectos del Manual de Usuario.



A5.10 Incorporar el Software, Registro de Trazabilidad, Reporte de Incidencias en Pruebas, Manual Operacional y Manual de Usuario en la Configuración del Software como parte de la línea base.

Se incorporan las historias de usuario que hayan sido validadas, los resultados de las pruebas y el manual de operación como línea base.

En la tabla 3.10, se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Integración de Software y Pruebas, incluyendo los roles responsables.

Integración de Software y Pruebas				
	RM	PR	RE	LT
A5.1				Pruebas de Aceptación
A5.2		Análisis de pruebas de aceptación		
A5.3		Integración de historias		
A5.4	Plan de Iteraciones			

Tabla 3.10. Relación de Actividades, Roles y Técnicas Ágiles dentro de la Integración de Software y Pruebas.

Nota: Las tareas 5.5, 5.6, 5.7, 5.8, 5.9 y 5.10 no se incluyen en la tabla pues no existe técnica ágil que se pueda aplicar para ellas.



4.2.6 Entrega del Producto

Ahora se presenta la forma de cómo se implementan técnicas de historias de usuario dentro de las tareas de la siguiente actividad de la ISO/IEC 29110, para ello se describen las tareas (Fig. 3.11) que pertenecen a la actividad de la *Entrega del Producto* que forma parte del proceso de *Desarrollo de Software* relativo al perfil básico.



Fig. 3.11. Dependencia de tareas de Entrega del Producto.



A6.1 Entender la Configuración del Software.

Se deben analizar los planes de Iteración para entender los cambios y el desarrollo del proyecto.

A6.2 Documentar el Manual de Mantenimiento o actualizarlo.

Se describe o actualiza el Manual de Mantenimiento, que puede ser todo el conjunto de documentos generados en la iteración.

A6.3 Verificar el Manual de Mantenimiento.

Se verifica el Manual de Mantenimiento

A6.4 Corregir los defectos encontrados en el Manual de Mantenimiento basado en el Reporte de Verificación y obtener la aprobación de las correcciones, si es necesario.

Se corrigen los errores del Manual de Mantenimiento y se obtiene su aprobación de ser necesario.

A6.5 Incorporar el Manual de Mantenimiento como línea base para la Configuración del Software.

Se incorpora el Manual de Mantenimiento a la Configuración del Software

A6.6 Realizar la entrega final de acuerdo con el Protocolo de Entrega.

Se Realiza la entrega del producto, además de las historias de usuario que fueron desarrolladas, así como los planes de Liberación y de Iteración. También se integran las pruebas realizadas, y la documentación generada.

En la tabla 3.11, se presenta un resumen de las actividades en donde se pueden utilizar las técnicas ágiles y que forman parte de la Entrega del Producto incluyendo los roles responsables.

Entrega del Producto			
	RM	RE	LT
A6.1	Plan de Iteración		
A6.2			
A6.3			
A6.4			
A6.5			
A6.6			Plan de Liberación Plan de Iteración

Tabla 3.11. Relación de Actividades, Roles y Técnicas Ágiles dentro de la Entrega del Producto.



Capítulo IV Ejemplos

Para demostrar la aplicación de la norma ISO/IEC 29110 a través de las Historias de Usuario, se plantea un proyecto donde se pueden manejar las técnicas descritas con anterioridad, éste se basó en el proyecto desarrollado durante el curso escolar de Tecnología Orientada a Objetos, en el Posgrado de Ciencia e Ingeniería de la Computación de la UNAM, siendo un proyecto típico escolar. Durante este proyecto se consideró apropiado basarse en sus objetivos. Se seleccionó este proyecto dado que no se contaba con un proyecto real para poder aplicar dichas técnicas.

A lo largo del ejemplo se obtienen los documentos de salida que solicita la norma y que son mencionados en la sección 2.2 y 2.3.

La compañía Firstjob es líder en outsourcing y desea desarrollar un sistema web que apoye al usuario (**persona, empresa**) al consultar una bolsa de trabajo, el cual será usado por las **personas** que estén en busca de empleo y por las **empresas** que soliciten empleados para ocupar algunas **vacantes**.

Firstjob espera que el sistema cumpla con las siguientes características:

1. La **empresa** se podrá registrar en el sistema y dar de alta, baja y cambiar sus datos. Para todos sus accesos tendrá una clave y una contraseña. Los datos de la empresa son: nombre, RFC, dirección, teléfono, fax, descripción (a que se dedica la empresa), correo electrónico, página web, clave contraseña y contacto (la persona que está dando de alta a la empresa).
2. Las **personas** se podrán registrar y dar de alta, de baja y cambiar sus datos por medio de su clave y contraseña. Los datos de la persona son: nombre completo, dirección, teléfono, correo electrónico, puesto que solicitan, curriculum vitae (CV), clave y contraseña. El CV debe incluir además de sus datos personales, información como habilidades, cursos que ha tomado, nivel de estudios, en los casos que se requiera la experiencia laboral se dará el lugar de trabajo, periodo laborado y puesto desempeñado, los idiomas que habla, escribe o lee, y todos los demás datos relevantes para el puesto que están solicitando.
3. Las **vacantes** son las plazas disponibles ofrecidas por las empresas. Las empresas podrán publicar, quitar y cambia sus vacantes. Para publicar una vacante se darán los siguientes datos: nombre de la vacante, requisitos, descripción (de qué se trata el empleo), sueldo, horario de trabajo, tipo de contratación (por horas, tiempo completo, días no hábiles), forma de contactarse con la empresa para concertar una cita, contacto (la persona que está dando de alta la vacante). Los siguientes datos serán tomados de los datos de la empresa: teléfono, fax, correo electrónico, nombre de la empresa.



4. Habrá diversos tipos de operaciones.

- 4.1 Las personas podrán consultar las vacantes publicadas sin necesidad de estar dadas de alta o registradas. Las consultas a las vacantes pueden ser: por tipo de trabajo, horario, nombre de la empresa y/o ubicación de la misma o sin restricción alguna.
- 4.2 Las personas pueden postularse para las vacantes registrándose (dándose de alta) en el sistema.
- 4.3 Las empresas podrán consultar a las personas que se han postulado para sus vacantes en la bolsa de trabajo para iniciar el contacto con ellas.

Dependiendo de la metodología ágil que se seleccione se asignarán los roles, en este ejemplo se utilizan los roles indicados por el estándar ISO/IEC 29110.

4.1 Administración de Proyectos.

4.1.1 Planeación del Proyecto

A1.1

Esta actividad se realiza entre AP, LT y el CL.

AP: ¿Qué necesita la empresa?

CL: Necesita poder registrarse en el sistema, modificar sus datos, registrar y modificar sus vacantes, consultar a las personas que se han postulado.

AP le explica al CL qué es una Historia de Usuario y le pide que escriba algunas, de aquí se generan las siguientes historias:

- H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña”
- H2 “La empresa puede ingresar sus datos”
- H3 “La empresa puede dar de alta sus vacantes”
- H4 “La empresa puede quitar sus vacantes”

AP, LT: ¿Que necesita el cliente?

CL: Deberá poder darse de alta, podrá registrarse, deberá incluir información de su CV, podrá revisar las vacantes.

El CL crea otras historias:

- H5 “El usuario puede registrarse” Nota: Podrá hacerlo desde Internet
- H6 “El cliente puede modificar su CV”



A1.2

El AP en conjunto con el CL generan de común acuerdo el Protocolo de Entrega.

AP: Se entregará el proyecto por ciclos, dentro de los cuales se presentará software funcional, que al final se integrará completamente dando como resultado el sistema completo.

CL: El software deberá ser presentado completamente dentro de las fechas establecidas.

A continuación se presenta un ejemplo de Protocolo de Entrega, vea Tabla 4.1.

Protocolo de Entrega	
La entrega se realizará en las instalaciones del cliente, se entregarán toda la documentación necesaria y las historias de usuario totalmente implementadas. Deberá estar presente un representante del Cliente, uno del equipo de trabajo y un usuario. La entrega deberá realizarse en la fecha establecida.	
Fecha:	Responsable:

Tabla 4.1. Protocolo de Entrega

A1.3

El AP y el CL han decidido establecer el tiempo de cada iteración en 3 semanas. El equipo de trabajo ha determinado que utilizará los días ideales de trabajo como medida de estimación de trabajo, así se ha establecido que dentro de cada iteración se realizarán 5 días ideales de trabajo. La tabla 4.2 presenta la forma de expresar lo anterior.

Número de Ciclos	
Duración de Iteración:	3 semanas
Unidad de Medida:	Días ideales de trabajo
Cantidad de trabajo por iteración:	5 días de trabajo ideal

Tabla 4.2. Número de Ciclos.

A1.4

El AP y el LT han determinado que podría avanzar 5 días de trabajo ideales por iteración.

A1.5

El AP decidió que 8 personas integran el equipo de trabajo, de los cuales 4 ya han tenido experiencia en proyectos similares.



A1.6

El AP establece que se elije una metodología ágil como la metodología para trabajar. El equipo se auto organiza y se auto asignan los roles convenientes al proyecto.

A1.7

En las Actividades A1.3 y A1.4 se determinó que la medida de medición serían los días ideales de trabajo. Se estableció que la velocidad del equipo para la primera iteración es de 5 días de trabajo ideal (debido a que no se cuenta con un registro de velocidad previo, también se toman en cuenta las historias que se desarrollarán en la primera iteración. Se aclara que la velocidad puede ser modificada en la siguiente iteración).

Para realizar la estimación el equipo de desarrollo utilizó el planning poker, donde cada integrante del equipo de trabajo expresa cuántos días ideales de trabajo se utilizarán para implementar la historia, las estimaciones más altas y bajas de cada ronda dan su explicación de por qué votaron así y se llega a un acuerdo, posteriormente se vuelve a votar sobre la misma historia y se determina cuál será su valor. Los resultados se anotan en la tabla 4.3.

Planning Poker			
Historia de Usuario	Ronda	Valor de estimación	Estimación Final
1	1	2,3,3,4,5,3,1,8	2
	2	2,2,3,3,2,3,2,2	
2	1	5,4,6,3,7,5,6,5	3
	2	4,3,3,4,5,3,3,3	

Tabla 4.3. Resultados del Planning Poker.

Se realiza la división de Historia de Usuario 6, ya que actualizar el CV se divide en:

- H7 “Actualizar nombre”
- H8 “Actualizar Habilidades”

Quedando la relación de la Historias de Usuario a desarrollar como se muestra en la tabla 4.4.

Historias de Usuario a Desarrollar	
ID	Historia
H1	La empresa puede registrarse en el sistema, utilizando su clave y contraseña
H2	La empresa puede ingresar sus datos
H3	La empresa puede dar de alta sus vacantes
H4	La empresa puede quitar sus vacantes
H5	El usuario puede registrarse
H6	El cliente puede modificar su CV
H7	Actualizar nombre
H8	Actualizar Habilidades

Tabla 4.4. Historias de Usuario a Desarrollar.



Una vez realizada la estimación se llevó a cabo la priorización, la cual fue realizada entre el equipo de trabajo y el CL. De esta priorización se obtuvo el Plan de Iteraciones, tal como se muestra en la tabla 4.5.

Plan de Iteración	
Iteración	Historias de Usuario
1	H1 H2
2	H3 H4
3	H5 H6
4	H7 H8

Tabla 4.5. Plan de Iteración.

Se estima que la cantidad total de días ideales totales del proyecto es: 20

Se estima que estas Historias de Usuario estarán listas en:

$$\text{Total de días ideales} / \text{velocidad} = 20 / 5 = 4 \text{ iteraciones}$$

1 iteración = 3 semanas así 4 iteraciones * 3 semanas = 12 semanas totales

A1.8

El AP le informa al CL el costo del proyecto, tanto en tiempo como en esfuerzo. Esto se muestra en la tabla 4.6.

Duración del Proyecto
<p>En el proyecto actual se han estimado en 8 Historias de Usuario, las cuales se han priorizado de la siguiente manera:</p> <ul style="list-style-type: none"> - H1, H2 durante la primera iteración - H3, H4 durante la segunda iteración - H5, H6 durante la tercera iteración - H7, H8 durante la cuarta iteración <p>Con las iteraciones establecidas en 3 semanas y con el esfuerzo por iteración se ha estimado que el proyecto podría durar 12 semanas.</p>

Tabla 4.6. Duración del Proyecto.

**A1.9**

Entre el AP y el LT analizan los riesgos y generan un Plan de Riesgos, como se hace en la tabla 4.7.

Plan de Riesgos			
Riesgo	Probabilidad de Ocurrencia	Indicador	Plan de Contingencia
No contar con las herramientas necesarias para desarrollar el proyecto y el conocimiento de su uso.	1	Significativo	Comprar las licencias necesarias, conseguir software de licencia libre.
Realizar cambios en el producto debido a fallas no contempladas	0.5	alto	Adquirir documentación suficiente sobre otras técnicas de programación.
Tener un bajo nivel de conocimiento en programación Java y de bases de datos	0.5	alto	Identificar al personal que cuenta con bajo nivel de Java y capacitarlo.
Cambios en los requerimientos.	0.5 *	significativo	Realizar las modificaciones necesarias en el diseño, comentar con el usuario y determinar si se aborta algún subsistema para seguir adelante.
No cumplir con los tiempos estimados de entrega.	1	moderado	Verificar el calendario y realizar las modificaciones necesarias, para que se ajusten a la entrega final.

Tabla 4.7. Plan de Riesgos. *Depende del requerimiento.

A1.10

Se define la Estrategia del Control de Versiones con la participación del AP. Ver tabla 4.8.

Estrategia de Control de Versiones	
Mecanismo de Almacenaje	Texto, html
Tipos de Cambio	Añadir, borra, renombrar, mover
Registro Histórico	Se generarán reportes que indiquen tipo y fecha de modificación.
Clasificación	Centralizado

Tabla 4.8. Estrategia de Control de Versiones

Se establece que cada elemento de software contenga la versión y revisión correcta que le corresponde a la línea base del proyecto. El nombre de los elementos almacenados será de la forma: Nombre-Versión.extensión.

Donde:

Nombre se refiere al nombre del documento

Versión se refiere a la versión del documento

Extensión se refiere al tipo de documento



A1.11

El AP genera el Plan del Proyecto, tal como se muestra en la tabla 4.9.

Plan del Proyecto							
<p>- Objetivo</p> <p><i>Generar una bolsa de trabajo, para que tanto personas como empresas tengan acceso a información concerniente a la vinculación laboral. El producto debe apoyar de manera simple y eficiente al usuario (persona/sistema), facilitando el registro y modificación de las vacantes de las empresas.</i></p>							
<p>- Descripción del Proyecto</p> <p><i>La compañía Firstjob es líder en outsourcing y desea desarrollar un sistema que apoye al usuario (persona, empresa) al consultar una bolsa de trabajo, el cual será usado por las personas que estén en busca de empleo y por las empresas que soliciten empleados para ocupar algunas vacantes.</i></p>							
<p>- Protocolo de Entrega</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #4F81BD; color: white; text-align: center;">Protocolo de Entrega</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="padding: 5px;"> <p><i>La entrega se realizará en las instalaciones del cliente, se entregarán toda la documentación necesaria y las historias de usuario totalmente implementadas. Deberá estar presente un representante del Cliente , uno del equipo de trabajo y un usuario.</i></p> </td> </tr> <tr> <td style="width: 50%; padding: 5px;">Fecha:</td> <td style="width: 50%; padding: 5px;">Responsable:</td> </tr> </tbody> </table>		Protocolo de Entrega		<p><i>La entrega se realizará en las instalaciones del cliente, se entregarán toda la documentación necesaria y las historias de usuario totalmente implementadas. Deberá estar presente un representante del Cliente , uno del equipo de trabajo y un usuario.</i></p>		Fecha:	Responsable:
Protocolo de Entrega							
<p><i>La entrega se realizará en las instalaciones del cliente, se entregarán toda la documentación necesaria y las historias de usuario totalmente implementadas. Deberá estar presente un representante del Cliente , uno del equipo de trabajo y un usuario.</i></p>							
Fecha:	Responsable:						
<p>- Recursos</p> <p><i>Se decidió que 8 personas integran el equipo de trabajo, de los cuales 4 ya han tenido experiencia en proyectos similares, el encargado de la base de datos carece de conocimientos de Java. Los demás tienen conocimientos de Java aun que no tan profundos.</i></p>							
<p>- Equipo de trabajo</p> <p>Depende de la metodología ágil que se seleccione, el equipo define los roles que se ejecutarán. En este ejemplo se utilizan los roles de la ISO/IEC 29110.</p> <p>AN: Tiene conocimientos y experiencia en la obtención, especificación y análisis de requerimientos. CL: Tiene conocimientos de los procesos del cliente y la habilidad de explicar los requerimientos. DIS: Tiene conocimiento y experiencia en componentes de Software y diseño de arquitectura. RM: Tiene conocimientos en técnicas de redacción y experiencia en el desarrollo y mantenimiento de Software.</p>							



PR: Tiene conocimiento y/o experiencia en programación, integración y pruebas.
 RE: Tiene conocimientos en técnicas de revisión y experiencia en el desarrollo y mantenimiento de software.
 AP: Tiene capacidad de liderazgo con experiencia en toma de decisiones, planeación estratégica, administración de personal, delegación y supervisión, finanzas, y desarrollo de software.
 LT: Tiene experiencia en el desarrollo y mantenimiento de Software.
 RP: Tiene conocimiento y experiencia en la planeación y desempeño, en la integración y pruebas de sistemas.
 DIU: Tiene conocimiento en diseño de interfaces de usuario y criterio ergonómico.
 ET: Tienen conocimiento y experiencia acorde a su rol.

- Estrategia de Control de Versiones

Estrategia de Control de Versiones	
Mecanismo de Almacenaje	Texto, html
Tipos de Cambio	Añadir, borra, renombrar, mover
Registro Histórico	Se generarán reportes que indiquen tipo y fecha de modificación.
Clasificación	Centralizado

Se establece que cada elemento de software contenga la versión y revisión correcta que le corresponde a la línea base del proyecto. El nombre de los elementos almacenados será de la forma: Nombre-Versión.extensión.

Donde:

Nombre se refiere al nombre del documento

Versión se refiere a la versión del documento

Extensión se refiere al tipo de documento

- Plan de Riesgos

Plan de Riesgos			
Riesgo	Probabilidad de Ocurrencia	Indicador	Plan de Contingencia
No contar con las herramientas necesarias para desarrollar el proyecto y el conocimiento de su uso.	1	Significativo	Comprar las licencias necesarias, conseguir software de licencia libre.
Realizar cambios en el producto debido a fallas no contempladas	0.5	alto	Adquirir documentación suficiente sobre otras técnicas de programación.
Tener un bajo nivel de conocimiento en programación Java y de	0.5	alto	Identificar al personal que cuenta con bajo nivel de Java y



bases de datos			capacitarlo.
Cambios en los requerimientos.	0.5	significativo	Realizar las modificaciones necesarias en el diseño, comentar con el usuario y determinar si se aborta algún subsistema para seguir adelante.
No cumplir con los tiempos estimados de entrega.	1	moderado	Verificar el calendario y realizar las modificaciones necesarias, para que se ajusten a la entrega final.

- **Plan de Iteraciones** (Apoyándose en la tabla 4.4)

Plan de Iteración	
Iteración	Historias de Usuario
1	H1 H2
2	H3 H4
3	H5 H6
4	H7 H8

- **Plan de Liberación**

Plan de Liberación	
Primera iteración	Del 06/12/10 al 24/12/10
Segunda iteración	Del 27/12/10 al 07/01/11
Tercera iteración	Del 10/01/11 al 28/01/11
Cuarta iteración	Del 31/01/11 al 18/02/11
Primera Liberación	21/02/11

Tabla 4.9. Plan del Proyecto

A1.12

Como aún no se efectúa la primera iteración, no se realiza re estimación alguna ni se integran historias inconclusas a otra iteración. No se modifican los calendarios. El AP y el LT generan un plan del proyecto por cada iteración.



A1.13

El AP y el LT verifican el Plan del Proyecto.

A1.14

El AP corrige los defectos del Plan del proyecto

A1.15

El AP y el CL validan el Plan del Proyecto.

A1.16

El AP corrige los defectos del Plan del proyecto

A1.17

El AP se apoya en A1.1 y en el plan de iteraciones. Como el AP conoce a qué programador se le asignaron ciertas historias, puede establecer la siguiente relación que se muestra en la tabla 4.10:

Estrategia de Control de Versiones por programador			
Programador	Historia de Usuario	Fecha de respaldo	Carpeta
Programador 1	H1	12/12/10	//Programador1.xxx
Programador 2	H3	11/12/10	//Programador2.xxx

Tabla 4.10. Estrategia de Control de Versiones por programador.

El repositorio se define como un espacio al que pueda acceder todo el equipo de trabajo. Se encuentra en el disco duro de la computadora donde se está desarrollando el producto, pero también en un servidor de respaldo de versiones, la estructura de dicho repositorio es **C:/TOO/Firstjob/ProyectoBolsaTrabajo** y a demás se cuenta con un respaldo en USB y correo electrónico.

A1.18

El AP incorpora el Plan del proyecto en el repositorio y en la línea base.



4.1.2 Ejecución del Plan del Proyecto

A2.1

El Plan del Proyecto para esta iteración es generado por el AP y el LT. Observe la tabla 4.11.

Plan del Proyecto de la Primera iteración											
<p>- Objetivo</p> <p><i>Generar una bolsa de trabajo, para que tanto personas como empresas tengan acceso a información concerniente a la vinculación laboral. El producto debe apoyar de manera simple y eficiente al usuario (persona/sistema), facilitando el registro y modificación de las vacantes de las empresas.</i></p>											
<p>- Descripción del Proyecto</p> <p><i>La compañía Firstjob es líder en outsourcing y desea desarrollar un sistema que apoye al usuario (persona, empresa) al consultar una bolsa de trabajo, el cual será usado por las personas que estén en busca de empleo y por las empresas que soliciten empleados para ocupar algunas vacantes.</i></p>											
<p>- Recursos</p> <p><i>Se decidió que 8 personas integran el equipo de trabajo, de los cuales 4 ya han tenido experiencia en proyectos similares, el encargado de la base de datos carece de conocimientos de Java. Los demás tienen conocimientos de Java aun que no tan profundos.</i></p>											
<p>- Estrategia de Control de Versiones</p> <table border="1"> <thead> <tr> <th colspan="2">Estrategia de Control de Versiones</th> </tr> </thead> <tbody> <tr> <td>Mecanismo de Almacenaje</td> <td>Texto, html</td> </tr> <tr> <td>Tipos de Cambio</td> <td>Añadir, borra, renombrar, mover</td> </tr> <tr> <td>Registro Histórico</td> <td>Se generarán reportes que indiquen tipo y fecha de modificación.</td> </tr> <tr> <td>Clasificación</td> <td>Centralizado</td> </tr> </tbody> </table>		Estrategia de Control de Versiones		Mecanismo de Almacenaje	Texto, html	Tipos de Cambio	Añadir, borra, renombrar, mover	Registro Histórico	Se generarán reportes que indiquen tipo y fecha de modificación.	Clasificación	Centralizado
Estrategia de Control de Versiones											
Mecanismo de Almacenaje	Texto, html										
Tipos de Cambio	Añadir, borra, renombrar, mover										
Registro Histórico	Se generarán reportes que indiquen tipo y fecha de modificación.										
Clasificación	Centralizado										
<p><i>Se establece que cada elemento de software contenga la versión y revisión correcta que le corresponde a la línea base del proyecto. El nombre de los elementos almacenados será de la forma: Nombre-Versión.extensión.</i></p> <p><i>Donde:</i></p> <ul style="list-style-type: none"> <i>Nombre se refiere al nombre del documento</i> <i>Versión se refiere a la versión del documento</i> <i>Extensión se refiere al tipo de documento</i> 											



- Plan de Riesgos			
Riesgo	Probabilidad de Ocurrencia	Indicador	Plan de Contingencia
No contar con las herramientas necesarias para desarrollar el proyecto y el conocimiento de su uso.	1	Significativo	Comprar las licencias necesarias, conseguir software de licencia libre.
Realizar cambios en el producto debido a fallas no contempladas	0.5	alto	Adquirir documentación suficiente sobre otras técnicas de programación.
Tener un bajo nivel de conocimiento en programación Java y de bases de datos	0.5	alto	Identificar al personal que cuenta con bajo nivel de Java y capacitarlo.
Cambios en los requerimientos.	0.5	significativo	Realizar las modificaciones necesarias en el diseño, comentar con el usuario y determinar si se aborta algún subsistema para seguir adelante.
No cumplir con los tiempos estimados de entrega.	1	moderado	Verificar el calendario y realizar las modificaciones necesarias, para que se ajusten a la entrega final.

- Plan de Iteraciones	
Plan de Iteración	
Iteración	Historias de Usuario
1	H1 H2

Tabla 4.11. Plan del Proyecto de la Primera iteración.

Los programadores han realizado la selección de Historias de usuario que se comprometen a desarrollar dentro de la iteración correspondiente. Esto se hace en una reunión donde se muestran las historias en un muro, para que estas sean visibles a todos, de todo esto se genera la tabla 4.12:



Asignación de Historias	
Programador	Historias
Programador 1	H1
Programador 2	H2

Tabla 4.12. Asignación de Historias.

A2.2

El AP determina que se realizarán juntas diarias para establecer el avance de las Historias y se registran en la tabla 4.13.

Registro de Estado del Progreso			
Programador	Tiempo dedicado a la Historia (Días Ideales)	Avance (%) Comparación con Estimados	Fecha
Programador 2	2	70	14/12/10

Tabla 4.13. Registro de Estado del Progreso.

A2.3

Aún no se generan cambios y no se ha pedido ninguna modificación (por ello no se llena la tabla 4.14). El AP junto con el LT analizan los cambios sugeridos. Se utilizará el formato mostrado en la tabla 4.14.

Formato de Solicitud de Cambios			
Nombre		Fecha	
Equipo		Proyecto	
Flujo de trabajo		Componente	
Información del Elemento			
ID de la Historia		Responsable	
Historia			
Información del Cambio			
Causa del Cambio			
Descripción			
Impacto			
Beneficios			
Estado del Cambio			
Aceptado		Rechazado	
Causas del Rechazo			
Firma		Fecha	

Tabla 4.14. Formato de Solicitud de Cambios



A2.4

Durante la reunión con el equipo de trabajo se utilizará este formato (Tabla 4.15) para que el cliente apoye al programador. Esta actividad es realizada por el AP, CL y el ET.

Registro de Reuniones			
Programador	Avance (%)	Solicita revisión con el cliente	Fecha
Programador 1	30	No	14/12/10

Tabla 4.15. Registro de Reuniones.

A2.5

El AP realiza el respaldo del avance de los programadores dentro de la estructura definida del repositorio. Lleva un control como el que se muestra en la tabla 4.16.

Respaldo del Repositorio	
Versión	Fecha de Respaldo
Ver. 1.0	15/12/10

Tabla 4.16. Respaldo del Repositorio

A2.6

En caso de ser necesario el AP utilizará la información almacenada en el repositorio y se apoyará en el Registro de Estado del Progreso. Se respalda el repositorio periódicamente.

4.1.3 Evaluación y Control del Proyecto

A3.1

El AP analiza las historias de usuario y evalúa la integración de nuevas historias de usuario. Ver tabla 4.17.

Registro de Acciones Correctivas		
Registro de Historias de Usuario Nuevas		
Historia de usuario	iteración	
H7	1	
Historia de Usuario	Solicitud de Modificación (S/N)	Aceptación de Modificación
H1	N	
Historia de Usuario Nueva	Solicitud de Modificación (S/N)	Aceptación de Modificación
H7	N	

Tabla 4.17. Registro de Acciones Correctivas.



4.1.4 Cierre del Proyecto

A4.1

Se realiza una reunión con el AP, ES y el CL, y si se determina que las historias se han completado totalmente, se da por terminado el proyecto y se genera el documento de aceptación. En caso contrario se revisan las historias que se están implementando y las que están listas, se analizan los planes de iteración y liberación. Se puede observar la tabla 4.18.

Documento de Aceptación		
<ul style="list-style-type: none"> • El equipo de trabajo realizó la verificación de todas las historias de usuario, y validó cada uno de ellas junto con el cliente. Se entrega el producto funcionando completamente. • La documentación se entregará en un CD-ROM que incluirá también el manual de usuario, manual de instalación, y la documentación generada en el desarrollo. • El Cliente será el encargado de la administración del sistema y de la base de datos. • La entrega se realiza en las instalaciones del Cliente. 		
Acepta Cliente:	Acepta Empresa:	Fecha:

Tabla 4.18. Documento de Aceptación.



4.2 Desarrollo de Software

4.2.1 Iniciación de la implementación del Software

A1.1

Se realiza la junta con el CL y el equipo de trabajo para analizar las nuevas Historias de Usuario que serán integradas, después se analizan los planes de iteración para observar como afectarán estas nuevas historias a todo el proyecto. Esto se realizará al inicio de cada iteración.

Para ello se utiliza la tabla 4.19:

Registro de Historias de Usuario Nuevas	
Historia de usuario	iteración
H8	1

Tabla 4.19. Registro de historias de usuario nuevas.

En este punto se analizan cuales historias son las que se continuarán desarrollando y cuales tentativamente pueden dejarse fuera de la presente liberación. Esto se observa en la tabla 4.20.

Implementación de Historias de Usuario			
Historia	Se implementa	No se implementa	Observaciones
H1	En proceso		Continúa en proceso

Tabla 4.20. Implementación de Historias de Usuario.

4.2.2 Análisis de Requerimientos del Software

A2.1

El LT y el AN asignan roles a los integrantes del Equipo de trabajo. Esto se observa en la tabla 4.21.

Asignación de Roles	
Integrante	Rol
Programador 1, 2, 3	PR
	CL
Programador 3	RP
Programador 4	LT
Programador 5	AP
Programador 6	DIS
Programador 7	DIU

Tabla 4.21. Asignación de Roles.

Se analizan las nuevas historias y el Programador las acepta. Esto se observa en la tabla 4.22.

Asignación de Historias Nuevas			
Programador	Historias	Historias Nuevas	Iteración
Programador 1	H1		
Programador 2	H2		

Tabla 4.22. Asignación de Historias Nuevas.



A2.2

Entre el AN, US, DIU y el CL han determinado las historias que deberán ser implementadas, y se integran a la especificación de requerimientos junto con sus pruebas de aceptación, esto se muestra en la tabla 4.23.

Especificación de Requerimientos	
-	H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña” * Prueba: Se escribe clave y contraseña validos y se ingresa al sistema
-	H2 “La empresa puede ingresar sus datos” * Prueba: Se ingresan los datos, posteriormente se consultan y se observa que todos se encuentre correctamente.
-	H3 “La empresa puede dar de alta sus vacantes” * Prueba: La empresa da de alta sus vacantes y se verifica que se hayan guardado exitosamente.
-	H4 “La empresa puede quitar sus vacantes” * Prueba: Se eliminan la vacantes y se verifica que se hayan borrado totalmente.
-	H5 “El usuario puede registrarse” Nota: Podrá hacerlo desde Internet * Prueba: Se hace un registro desde internet y se verifica que se pueda entrar al sistema.
-	H6 “El cliente puede modificar su CV” * Prueba: El cliente modifica su CV y se verifica que se hayan guardado exitosamente.
-	H7 “Actualizar nombre” * Prueba: Se verifica que el nombre se haya guardado correctamente.
-	H8 “Actualizar Habilidades” * Prueba: Se verifica que se hayan actualizado las habilidades.

Tabla 4.23. Especificación de Requerimientos.

A2.3

El RE genera un documento con las historias a ser desarrolladas en esta iteración y se registran os comentarios de los programadores y el CL. Observe la tabla 4.24.

Reporte de Verificación		
Programador	Historia	Observaciones
Programador 1	H1	
Programador 2	H2	

Tabla 4.24. Reporte de Verificación.



A2.4

Se modifican los malos entendidos encontrados directamente en la tarjeta, esto también se hace en las pruebas de aceptación. Esta actividad la realiza el AN y el DIU.

A2.5

El CL, US y el RP validan que la Historia sea clara. Observe la tabla 4.25

Reporte de Validación	
Historia de Usuario	Observaciones
H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña”	La historia es clara y sin ambigüedades

Tabla 4.25. Reporte de Validación.

A2.6

Si existen ambigüedades en la especificación de requerimientos según el reporte de validación, el AN y el DIU las corrigen.

A2.7

El RM genera o actualiza el Manual de Usuario que se observa en la tabla 4.26.

Manual de Usuario							
- H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña”							
<p>Bolsa de Trabajo Firstjob</p> <table border="1"><tr><td>Clave</td><td>*****</td></tr><tr><td>Contraseña</td><td>****</td></tr></table> <table border="1"><tr><td>Aceptar</td><td>Cancelar</td></tr></table>		Clave	*****	Contraseña	****	Aceptar	Cancelar
Clave	*****						
Contraseña	****						
Aceptar	Cancelar						
* Prueba 1: Se escribe clave y contraseña validos y se ingresa al sistema							



Bolsa de Trabajo Firstjob

Clave	*****
Contraseña	****

Aceptar	Cancelar
---------	----------

"La clave es incorrecta"

* Prueba 2: Se escribe clave invalida

Bolsa de Trabajo Firstjob

Clave	***
Contraseña	*****

Aceptar	Cancelar
---------	----------

"La contraseña es incorrecta"

* Prueba 3: Se escribe contraseña invalida

Bolsa de Trabajo Firstjob

Clave	
Contraseña	

Aceptar	Cancelar
---------	----------

"La clave y contraseña es requerida"

* Prueba 4: No se escribe contraseña ni clave.

Tabla 4.26. Manual de Usuario.

A2.8

El RE revisa el Manual de Usuario con el CL.

**A2.9**

El RM realiza la corrección de defectos del Manual de Usuario

A2.10

El LT incorpora el Manual de usuario y la especificación de Requerimientos al depósito del proyecto que es: **C:/TOO/Firstjob/ProyectoBolsaTrabajo/Configuración del software**. Ver tabla 4.27.

Configuración del Software	
-	Especificación de Requerimientos
	Especificación de Requerimientos
-	<ul style="list-style-type: none"> H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña” * Prueba: Se escribe clave y contraseña validos y se ingresa al sistema H2 “La empresa puede ingresar sus datos” * Prueba: Se ingresan los datos, posteriormente se consultan y se observa que todos se encuentre correctamente. H3 “La empresa puede dar de alta sus vacantes” * Prueba: La empresa da de alta sus vacantes y se verifica que se hayan guardado exitosamente. H4 “La empresa puede quitar sus vacantes” * Prueba: Se eliminan la vacantes y se verifica que se hayan borrado totalmente. H5 “El usuario puede registrarse” Nota: Podrá hacerlo desde Internet * Prueba: Se hace un registro desde internet y se verifica que se pueda entrar al sistema. H6 “El cliente puede modificar su CV” * Prueba: El cliente modifica su CV y se verifica que se hayan guardado exitosamente. H7 “Actualizar nombre” * Prueba: Se verifica que el nombre se haya guardado correctamente. H8 “Actualizar Habilidades” * Prueba: Se verifica que se hayan actualizado las habilidades.



<p>- Manual de Usuario</p> <table border="1" style="margin: auto;"> <tr> <th style="background-color: #d9e1f2;">Manual del Usuario</th> </tr> <tr> <td style="text-align: center;"> <p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p> </td> </tr> </table>	Manual del Usuario	<p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p>
Manual del Usuario		
<p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p>		

Tabla 4.27. Configuración del Software.

4.2.3 Arquitectura de Software y Diseño Detallado

A3.1

Los programadores han realizado la selección de Historias de usuario que se comprometen a desarrollar dentro de la primera iteración, ver tabla 4.28. Esta actividad es guiada por el LT, AN y el DIS.

Asignación de Historias	
Programador	Historias
Programador 1	H1
Programador 2	H2

Tabla 4.28. Asignación de Historias.

El equipo de trabajo se auto-organizará.

A3.2

El AN y el DIS guía esta actividad y se apoyan en el CL. La tabla 4.29 muestra los requerimientos.

Especificación de Requerimientos	
Historias	Observaciones
<ul style="list-style-type: none"> - H1 “La empresa puede registrarse en el sistema, utilizando su clave y contraseña” <ul style="list-style-type: none"> * Prueba 1: Se escribe clave y contraseña validos y se ingresa al sistema. * Prueba 2: Se escribe prueba invalida, se envía mensaje de error. * Prueba 3: Se escribe contraseña incorrecta, se envía menaje de error. * Prueba 4: No se escribe clave ni contraseña, se envía mensaje de error. 	



<ul style="list-style-type: none"> - H2 “La empresa puede ingresar sus datos” <ul style="list-style-type: none"> * Prueba: Se ingresan los datos, posteriormente se consultan y se observa que todos se encuentre correctamente. 	
---	--

Tabla 4.29. Especificación de requerimientos

A3.3

En este caso el AN, DIS y el DIU proponen la arquitectura Modelo-Vista-Controlador, la cual se evaluará a lo largo de la iteración, en caso de que se observen problemas se podrá realizar un cambio de arquitectura. Se genera el registro de trazabilidad como el mostrado en la tabla 4.30.

Registro de Trazabilidad				
	Clases		Paquetes	
Historia	C1	C2	P1	P2
H1	✓	✓	✓	
H2		✓		✓

Tabla 4.30. Registro de trazabilidad.

A3.4

Al inicio de cada iteración el RE verificará el Diseño del Software.

A3.5

El DIS, DIU y el AN corrigen los defectos encontrados.

A3.6

Se incorporan el diseño del Software y el Registro de trazabilidad al depósito del Proyecto que se encuentra en: **C:/TOO/Firstjob/ProyectoBolsaTrabajo**. El LT es los encargado de esta actividad.

4.2.4 Construcción del Software**A4.1**

Cada programador comienza la implementación de la historia que le ha sido asignada, así como las pruebas que se solicitan. El LT supervisa esta actividad.



A4.2

El PR puede realizar una grafica como la que se muestra en la figura 4.1, donde se explique la forma en que trabaja la arquitectura.

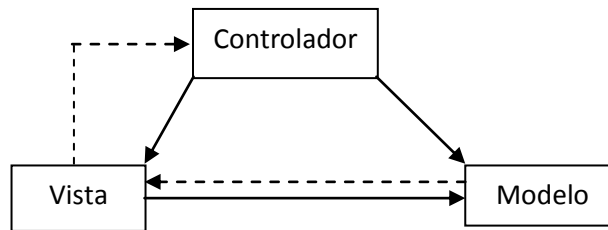


Fig. 4.1. Grafica de la arquitectura.

A4.3

Cada PR notifica el avance en la historia que está desarrollando, esto lo hace en la tabla 4.31 y 4.32.

Componentes				
Programador	Historia	Porcentaje de Avance	Pruebas Realizadas	Éxito
Programador 1	H1	20%	La indicada	No
Programador 2	H3	10%	Ninguna	No

Tabla 4.31. Componentes.

Registro de Trazabilidad				
Historia	Clases		Paquetes	
	C1	C2	P1	P2
H1	✓	✓	✓	
H2		✓		✓

Tabla 4.32. Registro de trazabilidad.

A4.4

Se realizan pruebas de aceptación por parte del RP.

Casos de Prueba		
Historia	Pruebas de Integración	Observaciones
H1	Las indicadas	Adecuadas
H2	Las indicadas	Consultar al Cliente

Tabla 4.33. Casos de Prueba.



A4.5

El RE puede realizar una lista con las historias y verificar sus pruebas.

Reporte de Verificación	
Historia	Se especificaron pruebas?
H1	Si
H2	No

Tabla 4.34. Reporte de Verificación.

A4.6

El RP revisa las correcciones con el cliente y las anota en la tabla 4.35.

Procedimientos de Pruebas			
Historia	Defectos	Corrección	Aceptación
H1	Si	Si	Si
H2	No	No	Si

Tabla 4.35. Procedimiento de Pruebas.

A4.7

Entre el LT integra la configuración del software, tal como se observa en la tabla 4.36.

Configuración del Software				
Componentes				
Programador	Historia	Porcentaje de Avance	Pruebas Realizadas	Éxito
Programador 1	H1	20%	La indicada	No
Programador 2	H2	10%	Ninguna	No
Registro de Trazabilidad				
Historia	Clases		Paquetes	
	C1	C2	P1	P2
H1	√	√	√	
H2		√		√



Procedimientos de Pruebas			
Historia	Defectos	Corrección	Aceptación
H1	Si	Si	Si
H2	No	No	Si

Plan de Iteración	
Iteración	Historias de Usuario
1	H1 H2
2	H3 H4
3	H5 H6
4	H7 H8

Tabla 4.36. Configuración del Software.

4.2.5 Integración de Software y Pruebas

A5.1

El LT verifica si las historias han sido completadas satisfactoriamente y deposita sus resultados en la tabla 4.37.

Estado de las Historias		
Historias	Realización de Pruebas	Aceptación
H1	Si	Si
H2	Si	No

Tabla 4.37. Estado de las historias.

A5.2

El PR debe ejecutar las pruebas establecidas por el CL.

Ejecución de Pruebas	
Historia	Observaciones de Pruebas
H1	Adecuadas
H2	Solicitud de revisión con el cliente

Tabla 4.38. Ejecución de pruebas.



A5.3

El PR realiza las pruebas de integración.

Reporte de incidentes de pruebas	
Historias Integradas	Resultados De Pruebas
H1 H2	Aceptables pero aun re requiere evaluar las pruebas de H2 con el Cliente

Tabla 4.39. Reporte de incidentes de pruebas.

Se actualiza el registro de trazabilidad como se muestra en la tabla 4.40.

Registro de Trazabilidad				
	Clases		Paquetes	
Historia	C1	C2	P1	P2
H1	√	√	√	
H2		√		√
H3	√			

Tabla 4.40. Registro de Trazabilidad.

A5.4

Según el Plan de iteraciones, las historias que ya están listas se toman como base para el Manual Operacional. El RM se encarga de esta actividad.

Manual Operacional
<p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p>

Tabla 4.41. Manual operacional.

A5.5

El RE realiza una prueba al Manual Operacional

A5.6

El RM corrige los defectos en el Manual Operacional

A5.7

Se genera el Manual de Usuario por parte del RM.

**A5.8**

El RE verifica el Manual de Usuario.

A5.9

Los defectos del Manual de Usuario son corregidos por el RM.

A5.10

El LT integra la Configuración del Software a la línea base, tal como se muestra en la tabla 4.42.

Configuración del Software				
Componentes				
Programador	Historia	Porcentaje de Avance	Pruebas Realizadas	Éxito
Programador 1	H1	20%	La indicada	No
Programador 2	H2	10%	Ninguna	No
Registro de Trazabilidad				
	Clases		Paquetes	
Historia	C1	C2	P1	P2
H1	√	√	√	
H2		√		√
Procedimientos de Pruebas				
Historia	Defectos	Corrección	Aceptación	
H1	Si	Si	Si	
H2	No	No	Si	
Plan de Iteración				
Iteración	Historias de Usuario			
1	H1 H2			
2	H3 H4			
3	H5 H6			



4	H7 H8
---	----------

Reporte de incidente de pruebas	
Historias Integradas	Resultados De Pruebas
H1 H2	Aceptables pero aun re requiere evaluar las pruebas de H2 con el Cliente

Manual Operacional

Manual Operacional
<p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p>

Manual de Usuario

Manual del Usuario
<p>Bolsa de trabajo de FirstJob</p> <p>Ver 1.0</p>

Tabla 4.42. Configuración del Software.



4.2.6 Entrega del Producto

A6.1

Se analiza el plan de Iteración que se presenta en la tabla 4.43, por parte del RM.

Plan de Iteración	
Iteración	Historias de Usuario
1	H1 H2
2	H3 H4
3	H5 H6
4	H7 H8

Tabla 4.43. Plan de iteración.

A6.2

El RM Genera el Manual de Mantenimiento como se observa en la tabla 4.44.

Manual de Mantenimiento
Bolsa de trabajo de FirstJob
Ver 1.0

Tabla 4.44. Manual de Mantenimiento.

A6.3

Se verifica el Manual de Mantenimiento, esto lo realiza el RE.

A6.4

El RM corrige los errores del Manual de Mantenimiento y obtiene su aprobación



A6.5

El LT incorpora el Manual de Mantenimiento a la Configuración del Software que se muestra en la tabla 4.45.

Configuración del Software				
Componentes				
Programador	Historia	Porcentaje de Avance	Pruebas Realizadas	Éxito
Programador 1	H1,H3	100%	La indicada	Si
Programador 2	H2,H4,H7	100%	La indicada	Si
Programador 3	H5, H8	100%	La indicada	Si
Registro de Trazabilidad				
Historia	Clases		Paquetes	
	C1	C2	P1	P2
H1	√	√	√	
H2		√		√
H3	√			
H4		√	√	
H5	√			√
H6	√			√
H7		√	√	
H8	√		√	
Procedimientos de Pruebas				
Historia	Defectos	Corrección	Aceptación	
H1	Si	Si	Si	
H2	Si	Si	Si	
H3	Si	Si	Si	
H4	Si	Si	Si	
H5	Si	Si	Si	
H6	Si	Si	Si	
H7	Si	Si	Si	
H8	Si	Si	Si	



Plan de Iteración	
Iteración	Historias de Usuario
1	H1
	H2
2	H3
	H4
3	H5
	H6
4	H7
	H8

Reporte de incidente de pruebas	
Historias Integradas	Resultados De Pruebas
H1	Aceptadas por el Cliente
H2	
H3	
H4	
H5	
H6	
H7	
H8	

Manual Operacional

Manual Operacional
Bolsa de trabajo de FirstJob
Ver 1.3

Manual de Usuario

Manual del Usuario
Bolsa de trabajo de FirstJob
Ver 1.5



Manual de Mantenimiento	
Manual de Mantenimiento	
Bolsa de trabajo de FirstJob	
Ver 1.4	

Tabla 4.45. Configuración del Software.

A6.6

Se realiza la entrega de la documentación y software desarrollado, según el Protocolo de entrega. Se debe generar un documento de aceptación de entrega firmado por el CL y el Equipo de trabajo. EL LT se encarga de esta actividad.

Entrega de Documentos	
Documento	Entrega
Entrega de Historias de Usuario Implementadas	√
Plan de Iteración	√
Plan de Liberación	√
Pruebas	√
Entrega del producto terminado	√

Tabla 4.46. Listado de entrega de documentos.



Conclusiones y trabajo futuro

En este trabajo se presenta una guía que permite sustentar la implementación de un marco de referencia de procesos como lo es la ISO/IEC 29110, con la técnica de Historias de Usuario. Dicha guía se desarrolló para todo el perfil básico.

En cada subproceso se identificaron el total de sus actividades, tomándose éstas como el 100%, y las actividades donde se pueden aplicar las técnicas de historias de usuario, como el porcentaje desconocido. Se estableció regla de tres, para obtener así el porcentaje de las actividades cubiertas por las técnicas de historias de usuario, con lo que se obtuvieron los resultados que se muestran en la tabla C.1:

Aplicación de Técnicas Ágiles	
Subproceso	Porcentaje
Plan del Proyecto	66.66
Ejecución del Plan del Proyecto	66.66
Evaluación y Control	100
Cierre del Proyecto	100
Inicio del Desarrollo	100
Análisis de Requisitos	60
Arquitectura de Software	66.66
Construcción del Software	85.71
Integración del Software	40
Entrega del Producto	33.33

Tabla C.1 Aplicación de técnicas ágiles a los subprocesos de la ISO/IEC 29110.

Las técnicas de historias de usuario se pueden aplicar en un promedio del 71.9 % (obtenido de la tabla C.1) al total de las actividades descritas por la norma ISO/IEC 29110. Cabe destacar que las historias de usuario no tienen aplicación en actividades como: corrección de defectos, creación y respaldos en repositorios, documentación y verificación de manuales.

Por lo que se puede concluir que las técnicas de historias de usuario se podrían aplicar a gran parte de la norma ISO/IEC 29110.

En base a la experiencia adquirida durante la elaboración de la presente guía, se concluye lo siguiente:

- Las historias de usuario podrían resultar en una herramienta práctica que proporcionen un buen sustento para la implementación de los procesos de la ISO/IEC 29110 para generar software que cumpla con las expectativas del cliente.
- Se podrían aplicar técnicas de Historias de Usuario para sustentar a la ISO/IEC 29110 sin que estas pierdan sus fundamentos ágiles como lo son su sencillez, y su énfasis en la comunicación verbal.



- El equipo de trabajo debe estar conformado por personal técnico capacitado y que además cuenten con la visión de trabajo en equipo. Esto se debe a que las metodologías ágiles resaltan al individuo y sus relaciones sobre los procesos y herramientas [Ágil, 2001].
- Como técnica ágil, el cliente deberá involucrarse a lo largo del proyecto, ya que se prefiere la colaboración con el cliente sobre la negociación contractual [Ágil, 2001]. Su participación facilitaría la priorización y estimación de las Historias de Usuario.
- Las técnicas ágiles se pueden presentar como una buena opción para la planeación de proyectos, lo cual resultaría ideal para las PyME's, además de que son fáciles de implementar aún con poca experiencia.
- Además de proponer técnicas ágiles para el desarrollo de las actividades, también se propone la implementación de buenas prácticas que existen dentro la Ingeniería de Software.

Como trabajo futuro se pretende validar la implementación de la estrategia propuesta. Si bien es un trabajo que pretende apoyar a las PyME's para la adopción de la ISO/IEC 29110, también resulta en una invitación a las organizaciones para implementar las técnicas y metodologías ágiles dentro de su organización, aún cuando esto implique un cambio dentro de sus políticas.



Referencias

- [ISO 29110]**
ISO/IEC 29110 Draft version, BER-005 29110 TR VSEP Overview WD2 15CL
- [Laporte, 2008]**
Laporte, C., S. Alexandre, and A. Renault, “Developing International Standards for Very Small Enterprises”. IEEE Computer, 2008. 41(3): p. 98-101.
- [Coleman, 2008]**
Coleman, G. and R. O’Connor, “Investigating software process in practice: A grounded theory perspective”. Journal of Systems and Software, 2008. 81(5): p. 772-784.
- [XP, 2003]**
Fowler M., “La Nueva Metodología”,
<http://www.programacionextrema.org/articulos/newMethodology.es.html>
- [Canós, 2010]**
Canós, J., “Metodologías Ágiles en el Desarrollo de Software”; UPV;
<http://www.willydev.net/descargas/prev/TodoAgil.pdf>, 2011.
- [Cohn, 2004]**
Cohn, M., “User Stories Applied for Agile Software Development”; Addison-Wesley, 2004.
- [Oktaba, 2008]**
Oktaba, H., Piattini, M., Pino, F., Orozco, M. Julia., Alquicira, C., “COMPETISOFT: Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos”; Ra-Ma, 2008.
- [García, 2010]**
García, C., Garzás, J., Piattini, M. “La mejora de procesos en pequeñas empresas y la ISO/IEC 29110”; Kibele Consulting, www.kybeleconsulting.com.
- [Augustine, 2005]**
Augustine, S., . Managing Agil Projects. Pearson Education Inc., USA. 2005.
- [Qumer, 2007]**
Qumer, A., Henderson-Sellers, B., “A framework to support the evaluation, adoption and improvement of agil methods in practice”. The journal of systems and software, 2007 pp. 1899-1901.
- [Oualid, 2009]**
Oualid, K. Ghislain, L., “Agile development: Issues and avenues requiring a substantial enhancement of the business perspective in large projects”. University of Quebec, Montreal. 2009
- [Cockburn, 2001]**
Cockburn, A., Highsmith, J., “Agile software development: The business of innovation”, IEEE Computer (Sept. 2001) pp. 120-122.
- [Cohen, 2004]**
Cohen, D., Lindvall, M., Costa, P., “An Introduction to Agile Methods, Fraunhofer Center for Experimental Software Engineering”, 2004.
- [Cohn, 2006]**
Cohen, M., “Agile Estimating and Planning”, Prentice Hall, 2006.
- [Cárdenas, 2010]**
Cárdenas, S., Pino, F., Ibargüengoitia, G., Piattini, M., “Procesos de Software de la ISO/IEC29110 Guiados por Historias de Usuario”. IIISIC 2010.
- [Ágil, 2001]**
Manifiesto por el Desarrollo Ágil de Software, página web: <http://agilemanifesto.org/iso/es/>