

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Marco de procesos de desarrollo de software orientado a entornos productivos pequeños, especializados en soluciones de gestión.”

Trabajo de Diploma para optar por el título de  
Ingeniero Informático.

Autor(es): Claudia Núñez Sanz

Leoandris Sanchez Vargas

Tutor(es): Ing. Dorisbel Muro Fumero

Co-tutor: Ing. Beatriz Hernández Cervantes

Junio, 2010

## **DECLARACIÓN DE AUTORÍA**

Declaro que somos los únicos autores de este trabajo y autorizo a la Facultad 6 y a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2010.

---

Claudia Núñez Sanz

---

Leoandris Sanchez Vargas

---

Ing. Dorisbel Muro Fumero

---

Ing. Beatriz Hernández Cervantes

## **Datos de Contacto**

Tutor: Ing. Dorisbel Muro Fumero

Correo Electrónico: [dmuro@uci.cu](mailto:dmuro@uci.cu)

Ingeniero Informático, Universidad de Ciencias Informáticas, 2007.

Categoría Científica: Ingeniero.

Categoría Docente: Instructor.

Co-Tutor: Ing. Beatriz Hernández Cervantes

Correo Electrónico: [bcervantes@uci.cu](mailto:bcervantes@uci.cu)

Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas 2009.

Categoría Científica: Ingeniero.

Categoría Docente: Adiestrado.

## **RESUMEN**

El presente trabajo de diploma propone un marco de procesos de desarrollo de software orientado a entornos productivos pequeños y especializados en soluciones de gestión que se centrará en demostrar la relación existente o puntos de encuentro entre la guía del PMBOK y SCRUM como metodología ágil para desarrollo de software, además, se integra el Método ADD (Arquitectura Dirigida por Diseño) dentro del ciclo de desarrollo de software obteniendo una abstracción arquitectónica del modelo y los modelos de procesos industriales Línea de Producción unido a Fábricas de Software permitiendo una reducción de los niveles de incertidumbre en el desarrollo y la mantención de proyectos de software, obtener la estructura del proyecto, identificando áreas, procesos y macro procesos y un producto general partiendo de la integración de cada producto específico identificado por cada área o proceso a automatizar. El modelo fue aplicado en la empresa cubana SICS (Servicios Informáticos de Consultoría y Sistemas) obteniendo buenos resultados hasta el momento y está previsto aplicarse en DESOFT.

**Palabras claves:** Project Management Body of Knowledge (PMBOK), SCRUM.

## Tabla de contenido.

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
INTRODUCCIÓN.....	4
¿QUÉ ES UN MARCO DE PROCESO DE DESARROLLO DE SOFTWARE? .....	4
1.1 GESTIÓN DE PROYECTO. ....	4
1.1.1 <i>Fundamentos esenciales del Project Management Body of Knowledge (PMBOK)</i> . ....	4
1.2 INGENIERÍA DE SOFTWARE. ....	5
1.2.1 <i>Metodologías de desarrollo de software</i> . ....	5
1.3 COMPARACIÓN ENTRE METODOLOGÍA TRADICIONAL Y METODOLOGÍA ÁGIL. ....	5
1.3.2 <i>Fundamentos esenciales de SCRUM</i> .....	5
1.4 ELEMENTOS FUNDAMENTALES DE ADD (ARQUITECTURA DIRIGIDA POR DISEÑO).....	7
<b>CONCLUSIONES PARCIALES</b> .....	<b>8</b>
<b>CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN</b> .....	<b>9</b>
INTRODUCCIÓN.....	9
2.1 NIVEL 1: GESTIÓN DE PROYECTO.....	9
<i>Modelo de procesos de gestión de proyecto centrado en SCRUM</i> .....	9
2.1.2 <i>Fase: Especulación</i> .....	15
2.1.3 <i>Fase: Exploración</i> .....	40
2.1.4 <i>Fase: Revisión y Cierre</i> .....	60
2.1.5 <i>Expediente documental</i> .....	61
2.2 NIVEL 2: MÉTODO DE DESARROLLO DE SOFTWARE .....	61
<i>Introducción al nivel 2: Modelos de Procesos Industriales</i> .....	61
2.3 NIVEL 3: PLATAFORMA TECNOLÓGICA. ....	63

*Selección de la herramienta para plantear la plataforma tecnológica.....63*

Herramientas a utilizar durante la aplicación del modelo.....64

**CONCLUSIONES PARCIALES.....64**

**CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN .....65**

INTRODUCCIÓN.....65

3.1 PROCESO DE SELECCIÓN DE EXPERTOS.....66

3.2 CÁLCULO DEL COEFICIENTE DE COMPETENCIA.....66

3.3 ELABORACIÓN DEL CUESTIONARIO DE VALIDACIÓN.....67

3.4 ESTABLECIMIENTO DE CONCORDANCIA ENTRE LOS EXPERTOS.....67

3.5 ANÁLISIS DE LOS RESULTADOS.....68

**CONCLUSIONES PARCIALES.....69**

**CONCLUSIONES.....70**

**RECOMENDACIONES.....71**

**REFERENCIA BIBLIOGRÁFICA.....72**

**BIBLIOGRAFÍA.....74**

### **Introducción**

En la actualidad la globalización exige flexibilidad a las empresas del software para mantener su auge en los próximos años, debido a esto el ambiente de negocios ha cambiado y continuará variando en forma constante, para que las empresas desarrollen la capacidad de respuesta rápida ante nuevas oportunidades de negocio y retos competitivos al enfrentar cambios en las condiciones de mercado. De esta manera, la realidad de la mayoría de las empresas ha venido creciendo y mejorando sus procesos, lo que ha conllevado a la aplicación de nuevas prácticas en la industria de software siendo necesario reflexionar sobre el grado de adaptación de las metodologías utilizadas hasta el momento, que han sido metodologías pesadas o convencionales, basadas en una planificación extensa, propiciando que las empresas del software actuales se interesen por nuevos enfoques más ligeros y abiertos a los cambios, las "Metodologías Ágiles." (1)

Las metodologías ágiles en comparación con las metodologías convencionales, perciben cada respuesta al cambio como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de proyecto como un aspecto inherente al propio proceso de desarrollo de software. Como parte de este proceso la gestión de proyecto tiene que dar el paso de evolución apropiado para desarrollar nuevos productos y servicios en estos sectores y adaptarse a los cambios del entorno en el que trabaja ya que las circunstancias de los mercados y de las empresas no pueden cambiar.

La Industria Cubana del Software es de joven creación y su meta principal es informatizar la sociedad y elevar los niveles de desarrollo de software. En la actualidad los equipos de desarrollo cuentan con pocas personas para llevar a cabo todo el proceso de trabajo, no existe una cultura de producción de software en la que se realicen estudios por parte de equipos multidisciplinarios encaminados a la creación de un producto orientado a un mercado amplio, caracterizándose los desarrollos por ser locales, muy justos o a la medida, poco industriales y empíricos, por lo que se deduce la baja capacidad de reutilización de soluciones y por tanto, la necesidad de que los ciclos productivos sean cortos manteniendo así un elevado por ciento de efectividad enfocados al predominio de soluciones de gestión con una formalización o estandarización del tiempo y un estricto control del trabajo.

Para el desarrollo de esta nueva fuente de ingreso se utiliza el potencial que existe en la Universidad de Ciencias Informáticas, que tiene como uno de sus principales objetivos el desarrollo de software que cuente con la calidad requerida, muestra de ello es la aplicación en la universidad del modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software conocido como CMMI. Esta es una de las principales líneas en que se desarrolla el centro de altos

estudios actualmente. Aunque a lo largo de estos años la universidad ha alcanzado un alto grado de madurez y organización en cuanto a desarrollo y tecnología con los que se ha conseguido alcanzar un elevado conocimiento científico-técnico, la aplicación de buenas prácticas y se ha impulsado el desarrollo de tecnologías, es necesario reconocer que la universidad ha heredado algunas problemáticas de la Empresa Cubana del Software al existir una pobre estandarización e implantación de tendencias industriales en los modelos de procesos, las soluciones por lo general son hechas a la medida y por tanto, se deriva la baja capacidad de reutilización entre soluciones y proyectos lo que ha conllevado a que las grandes soluciones de la universidad no se comuniquen y por ende no se cuente con una suite completa que responda a las necesidades del centro en general.

Por tanto, se propone como **Problema científico:**

¿Cómo agilizar y estandarizar los procesos de desarrollo de software orientados a entornos productivos pequeños especializados en soluciones de gestión?

**Objeto de estudio:** Procesos de desarrollo de software

**Campo de acción:** Procesos de desarrollo de software para entornos productivos pequeños especializados en soluciones de gestión.

**Objetivo general:**

Elaborar un marco de procesos de desarrollo de Software orientado a entornos productivos pequeños, especializados en soluciones de gestión, para garantizar la agilidad y estandarización en los procesos productivos de estos entornos.

**Objetivos específicos:**

- ✓ Realizar un estudio del estado del arte de los procesos de desarrollo de software, específicamente, metodologías de desarrollo de software y gestión de proyecto.
- ✓ Proponer un marco de proceso de desarrollo de software para equipos productivos pequeños y especializados en soluciones de gestión.
- ✓ Documentar los procesos propuestos.
- ✓ Describir roles, artefactos y herramientas de los procesos propuestos.
- ✓ Validar la propuesta de modelo.

**Posibles resultados:**

Marco de procesos de desarrollo de software orientados a entornos productivos pequeños especializados en soluciones de gestión en cada uno de sus niveles: **Gestión de proyectos, Método de desarrollo de software y Herramientas y aplicaciones.**

**Tareas a cumplir por estudiantes:**

- ✓ Estudio del estado del arte del comportamiento de las metodologías de gestión de proyectos, desarrollo de software, así como sus herramientas y aplicaciones.
- ✓ Propuesta del marco de procesos de desarrollo de software orientado a entornos productivos pequeños, especializados en soluciones de gestión.
- ✓ Validación de la investigación según los resultados obtenidos.

El presente trabajo de diploma tiene como estructura:

- **Capítulo 1: Fundamentación Teórica.**

Este capítulo contiene los fundamentos teóricos para entender el problema a solucionar. Conceptos fundamentales, técnicas, tendencias y metodologías, así como los elementos fundamentales que permitirán definir las bases para el desarrollo de la investigación.

- **Capítulo 2: Desarrollo de la solución.**

Contiene la propuesta del modelo y descripción del desarrollo de la solución en tres niveles específicos. En el primer nivel se presenta el modelo de las actividades de SCRUM centradas en el proceso de gestión de proyecto referenciando el Project Management Body of Knowledge (PMBOK), así como su descripción y expediente documental. En el segundo nivel se especifica el método de desarrollo de software y el expediente documental y en el tercer nivel se da a conocer la herramienta escogida para la plataforma tecnológica así como las guías y manuales de instalación.

- **Capítulo 3: Evaluación de la solución.**

Se realiza la revisión y valoración de la propuesta, con el objetivo de validar la solución planteada. Se da a conocer la evaluación y las estadísticas que arrojaron los resultados de las encuestas aplicadas.

## **Capítulo 1: FUNDAMENTACIÓN TEÓRICA**

### **Introducción**

Este capítulo contiene los fundamentos teóricos para entender el problema a solucionar. Conceptos fundamentales, técnicas, tendencias, metodologías y herramientas utilizadas, así como los elementos fundamentales que permitirán definir las bases para el desarrollo de la investigación. Se llevará a cabo un estudio comparativo de las diferentes metodologías tanto ágiles como convencionales para la justificación de una posterior selección así como una descripción de algunas de las metodologías de gestión de proyecto.

### **¿Qué es un marco de proceso de desarrollo de software?**

*“El proceso de desarrollo de software, describe tres elementos fundamentales: marco común del proceso, conjunto de tareas y actividades de protección. El marco común de procesos, se define como un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software con independencia del tamaño o complejidad.” (2)*

### **1.1 Gestión de Proyecto.**

*“La Gestión de Proyectos se puede describir como un proceso de planteamiento, ejecución y control de un proyecto, desde su comienzo hasta su conclusión, con el propósito de alcanzar un objetivo final en un plazo de tiempo determinado, con un coste y nivel de calidad determinados, a través de la movilización de recursos técnicos, financieros y humanos. Incorporando variadas áreas del conocimiento, su objetivo final es obtener el mejor resultado posible del trinomio coste-tiempo-calidad.” (3)*

#### **1.1.1 Fundamentos esenciales del Project Management Body of Knowledge (PMBOK).**

EL PMBOK es una colección de procesos y áreas de conocimiento generalmente aceptadas como las mejores prácticas dentro de la gestión de proyectos; un estándar reconocido internacionalmente (IEEE Std 1490-2003) que provee los fundamentos de la gestión de proyectos que son aplicables a un amplio rango de proyectos. Reconoce 5 grupos de procesos básicos y 9 áreas de conocimiento comunes a casi todos los proyectos. Los procesos son descritos en términos de: Entradas (documentos, planes, diseños), Herramientas y Técnicas (mecanismos aplicados a las entradas) y Salidas (documentos, productos).

Las nueve áreas del conocimiento mencionadas en el PMBOK son: gestión de la integración, gestión del alcance, gestión del tiempo, gestión de la calidad, gestión de costos, gestión del riesgo, gestión de recursos humanos, gestión de la comunicación y gestión de compras y adquisiciones (4).

## **1.2 Ingeniería de Software.**

*“La Ingeniería de Software es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este, después que se utiliza. Esta es la disciplina que integra el proceso, los métodos, y las herramientas para el desarrollo de software “* (5). Consultar materiales complementarios: **Figura 1: Ingeniería de Software.**

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, su implementación, pone en práctica las pruebas necesarias para confirmar el correcto funcionamiento y el despliegue del producto con el apoyo de las herramientas que brindan el soporte automático.

### **1.2.1 Metodologías de desarrollo de software.**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, procesos, políticas y ayudas a la documentación para el desarrollo de productos de software. Tienen como finalidad garantizar la eficacia (cumplir los requisitos iniciales) y la eficiencia (minimizar las pérdidas de tiempo) en el proceso de generación de software. (6)

**SCRUM.** (Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle), dispone de la certificación SCRUM Certified; define un marco para la gestión de proyectos. Está especialmente indicada para proyectos con un rápido cambio de requisitos y compuesto por equipos multi-funcionales y auto-administrados. El trabajo a realizar durante cada una de las iteraciones es seleccionado por el equipo de desarrollo de una lista de requerimientos priorizados conocida como el "Product Backlog". De esta lista, el equipo selecciona los requerimientos más prioritarios que puede desarrollar completamente en 30 días y define y estima las tareas necesarias para cumplir con esos requerimientos; ese listado de tareas se conoce como el "Sprint Backlog". (7)

### **1.3 Comparación entre metodología tradicional y metodología ágil.**

El desarrollo de las metodologías ágiles ha demostrado que la participación de todo el equipo aporta mucho más talento innovador y diferencial. En las metodologías ágiles no se espera a desarrollar requisitos detallados antes de empezar el análisis o el desarrollo, más bien se parte de la visión general y el descubrimiento paulatino durante el desarrollo que dará paso a las circunstancias que se irán produciendo, dibujando el detalle de forma paralela al desarrollo. (Consultar materiales complementarios **Tabla 1: Comparación entre metodología tradicional y metodología ágil** y **Tabla 2: Comparación entre las diferentes metodologías ágiles.**)

#### **1.3.2 Fundamentos esenciales de SCRUM.**

Podemos identificar en el patrón de ciclo de vida de un modelo de desarrollo ágil cinco etapas:

**Concepto:** en esta fase se crea la visión del producto o servicio que se quiere obtener, pues se necesita conocer el alcance del proyecto. Se selecciona al equipo de personas que lo llevará a cabo. Esta información la deben compartir todos los integrantes del equipo.

**Especulación:** una vez que se dispone de la visión de lo que se quiere conseguir, el equipo especula y construye hipótesis sobre la información de la visión. En esta etapa se determinan las limitaciones impuestas por el entorno de negocio y se determina la primera aproximación de lo que se puede producir. Durante el desarrollo se confronta la realidad de lo que va obteniendo, dígase valor, posibilidades y la situación de negocio del entorno en cada momento.

**Exploración:** desarrollo de las funcionalidades que ha determinado el equipo para generar el siguiente incremento de producto.

**Revisión:** el equipo y los usuarios revisan las funcionalidades construidas hasta ese momento, trabajan y operan con el producto real para determinar su alineación y dirección con el objetivo.

**Cierre:** al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado. (8)

SCRUM plantea la realización de tres reuniones: planificación, seguimiento y revisión del Sprint.

**Planificación del Sprint:** en esta reunión se define el Product Backlog, el cual consiste en una lista priorizada de requisitos del sistema y es un documento vivo, que puede ser continuamente actualizado. En cada iteración el Product Backlog es revisado por el equipo.

**Seguimiento del Sprint:** se llevan a cabo breves reuniones diarias, para ver el avance de las tareas y el trabajo que está previsto para la jornada.

**Revisión del Sprint:** una vez finalizado el Sprint se realiza un análisis y revisión del incremento generado, se presentan los resultados finales y se recomienda siempre tener preparado un demo. (8)

La mayoría de las bibliografías dan a conocer entre otros 3 tipos de roles: el propietario del producto, el scrum máster y el equipo.

**Propietario del producto:** es la persona concedora del entorno de negocio del cliente y de la visión del producto, representa a todos los interesados en el producto final y es el responsable del Product Backlog, de obtener el resultado de mayor valor posible para los usuarios o clientes, de la financiación necesaria para el proyecto y de decidir cómo debe ser el resultado final del proyecto.

**Scrum Máster:** es el encargado de garantizar el funcionamiento de los procesos y de la metodología. Debe interactuar tanto con el equipo como con el cliente y con los gestores. Es responsable de garantizar

que el proceso sea entendido y seguido, asegurar que el Equipo se adhiera a los valores, prácticas y normas de SCRUM.

**Equipo de desarrollo:** es el equipo del proyecto y tiene la autoridad para decidir en las acciones necesarias y para auto-organizarse. Los miembros del equipo deben tener todas las habilidades necesarias para crear un incremento de trabajo.

SCRUM no requiere de ninguna práctica concreta para el desarrollo del software, sin embargo, sí dispone de **prácticas y herramientas** para la gestión de las diferentes fases.

**Product Backlog:** define los requisitos del sistema o el trabajo a hacer a lo largo del proyecto. Está compuesto por una lista de requisitos de negocio y técnicos, actualizados y priorizados.

**Sprint Backlog:** es una lista de trabajos que el equipo se compromete a realizar para generar el incremento previsto. Las tareas están asignadas a personas y tienen estimados el tiempo y los recursos.

**Estimación de esfuerzo:** es un proceso iterativo en el cual las estimaciones de los ítems del Product Backlog son reajustadas acorde a la información obtenida en la última iteración.

**Gráfico Burn-Down:** es una herramienta para gestionar y seguir el trabajo de cada Sprint y representa gráficamente el avance del Sprint.

**Planning Póker o estimación de Póker:** juego que ayuda al equipo a establecer una estimación de las tareas, cada participante dispone de 8 cartas con los números necesarios para representar la cifra de días que estima para la tarea.

### **1.4 Elementos fundamentales de ADD (Arquitectura Dirigida por Diseño).**

El método **ADD (Atributo-Driven Design o Arquitectura dirigida por diseño)** fue desarrollado por la Carnegie Mellon Software Engineering Institute. ADD es una aproximación a la definición de una arquitectura de software en el que el proceso de diseño se basa en la comprensión de la relación entre la calidad del software y la arquitectura de los mecanismos utilizados para alcanzar estas cualidades. Sigue un proceso recursivo que se descompone en sistema o elemento del sistema mediante la aplicación de las tácticas y los patrones arquitectónicos que satisfagan los requisitos de calidad.

Inicialmente la arquitectura no era tomada en cuenta en las metodologías ágiles o solo era considerada de un modo superficial. La carencia de prácticas relacionadas con el diseño formal de la aplicación es precisamente una de las debilidades de las metodologías ágiles. Esta falta de formalismo se halla justificada porque el diseño detallado lleva una inversión de tiempo considerable en decisiones y aspectos. Resulta difícil encontrar documentación sobre cómo definir una arquitectura en un proyecto de desarrollo de software guiado por una metodología ágil, en este caso SCRUM, por lo que sería de gran importancia

contar con un proceso capaz de acoplarse a una metodología de desarrollo de software ágil, en el que se distingan claramente los beneficios de aplicarlo en cada etapa y las consecuencias de no tenerlo en cuenta. Específicamente en el desarrollo de SCRUM se va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. Sin embargo, existen posiciones extremas por parte de las metodologías ágiles, que plantean que el código fuente es el diseño, es como decir que el edificio terminado es su diseño, cuando en la realidad se debe tener en cuenta que hay cosas que no pueden verse, como la sinergia que generan las partes como un todo que evita ver ciertos aspectos, dígame vistas y modelos, además, de que tampoco se puede analizar su estructura interna o esqueleto. (9)

SCRUM es una metodología pensada para recibir con manos abiertas los buenos cambios y con la utilización del método ADD se define la arquitectura conceptual del software, así como el diseño de sus procesos basados en los requerimientos de la calidad, es decir, se extrae la arquitectura.

### **Conclusiones Parciales.**

En este capítulo se realizó un análisis valorativo de los elementos fundamentales de los modelos y metodologías de desarrollo de software y gestión de proyectos existentes, teniendo en cuenta las principales características. Además, se establecen comparaciones entre las diferentes metodologías enfocando su uso hacia los modelos ágiles, describiendo así la aplicación de estos en las empresas para el desarrollo de software. Se realizó un análisis valorativo del ADD como método de abstracción arquitectónica partiendo del diseño y su importancia de aplicarlo en el modelo.

Tras el estudio realizado se ha podido apreciar que la información disponible presenta características importantes e imprescindibles y está orientada a perfeccionar la producción, propiciando un mejor resultado, aunque ninguno de los elementos estudiados cubre las necesidades para cumplir los objetivos del trabajo, este análisis proporcionó un conocimiento claro de cómo pueden aportar cada uno de estos enfoques a la construcción de la propuesta de la investigación.

## **Capítulo 2: DESARROLLO DE LA SOLUCIÓN**

### **Introducción**

En este capítulo se propone un marco de proceso de desarrollo de software basado en el estudio de las metodologías y marcos de trabajo analizados en el capítulo anterior. Como parte de la investigación se expone la fundamentación de los diferentes niveles propuestos para el desarrollo de este marco de procesos.

### **2.1 Nivel 1: Gestión de Proyecto**

#### **Modelo de procesos de gestión de proyecto centrado en SCRUM.**

Debido a que la metodología utilizada para el desarrollo de software no abarca todos los elementos de la gestión de proyectos se ha llevado a cabo un estudio para demostrar los puntos de encuentro entre la guía del PMBOK y SCRUM, donde se evidencia cómo pueden ser agrupadas las diferentes áreas de conocimiento de la guía con las fases de SCRUM destacando los artefactos generados más importantes. (Consultar Materiales complementarios **Tabla 3: Relación entre las actividades del PMBOK y SCRUM.**)

El estudio se basó en la realización de una encuesta que incluye preguntas acerca de todas las actividades que plantea la guía del PMBOK, el objetivo fue seleccionar las actividades de mayor prioridad en la gestión de proyectos y fue aplicada a especialista del área de gestión de proyecto. La encuesta arrojó como resultado las distintas actividades que no deben faltar en la gestión de un proyecto, además se evidencia que el Plan de Gestión de Proyecto como artefacto generado constituye el documento rector y este será actualizado según sea necesario. La esencia del Plan de Gestión de Proyecto no es más que la agilidad de realizar estimaciones, pensando en que todos los requerimientos del cliente son reales en este inicio (10). Sin perder el sentido por el cual existen las metodologías ágiles, refiriéndose así al principio del manifiesto ágil: *“La respuesta al cambio, por encima del seguimiento de un plan”* (11), se trata de que exista una planificación del proyecto. Incluye además, otros aspectos sobre la gestión de proyecto como son: la identificación del personal a trabajar en el proyecto, la planificación de la gestión de riesgos, calidad, planificaciones y cambios, recursos humanos, las métricas y los documentos de trazabilidad, calidad y presupuesto.

#### **Guía de actividades de los procesos centrado en SCRUM.**

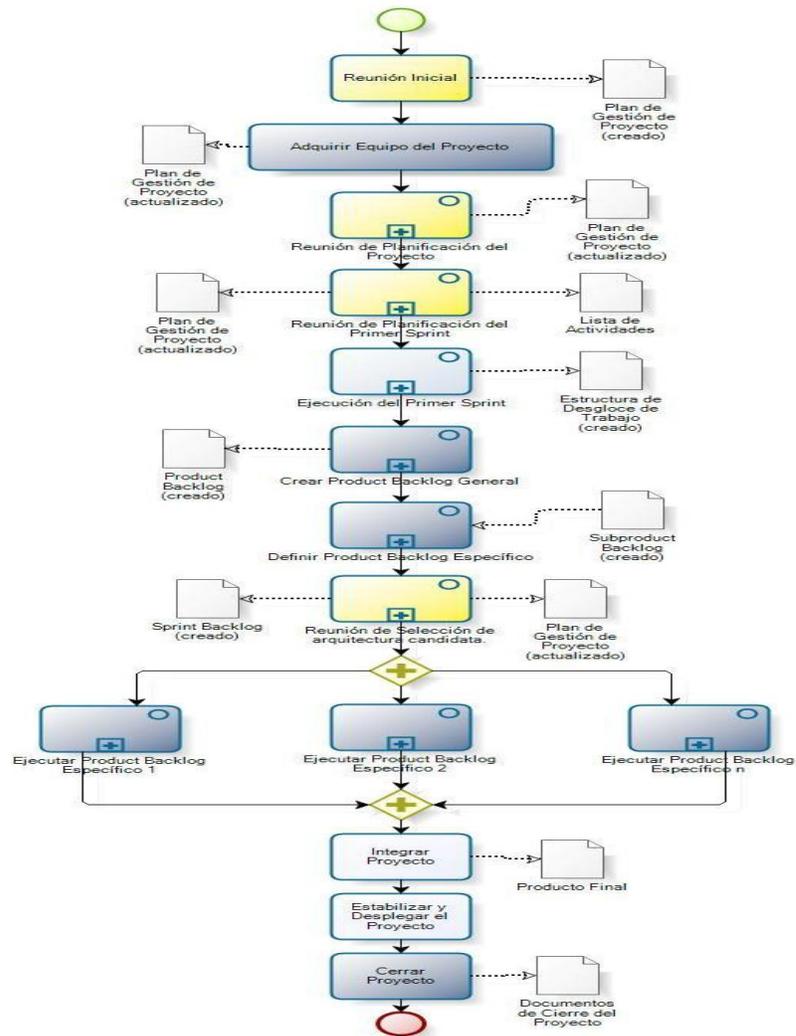


Figura 2: Modelo del marco de procesos de desarrollo de software especializado en soluciones de gestión.

2.1.1 Fase: Concepto.

Actividad # 1: Reunión Inicial.

Roles:

- ✓ Dueño del Producto
- ✓ Jefe de Proyecto
- ✓ Cliente

Artefactos de Entrada:

Artefactos de Salida:

- ✓ Enunciado del Trabajo del proyecto.
- ✓ Factores ambientales de la Empresa.
- ✓ Plan de Gestión de Proyecto (creado) dentro está contenido el Plan de Gestión del Alcance del Proyecto.

**Descripción:**

Se define el proyecto y los objetivos que deben cumplirse, además, se abordan y documentan las características y los límites del proyecto, sus productos y servicios relacionados, así como los métodos de aceptación y el control del alcance. Estos aspectos se desarrollan también en la actividad de gestión de proyecto que plantea el PMBOK: Enunciado del Alcance del Proyecto Preliminar. Esta actividad se desarrolla a partir de la información suministrada por el iniciador o el patrocinador. El contenido del alcance variará de acuerdo con el área de aplicación y la complejidad del proyecto y puede incluir algunos o todos los componentes identificados con anterioridad. En esta reunión se especifican:

- Objetivos del proyecto y del producto.
- Características del producto o servicio.
- Criterios de aceptación del producto.
- Límites del proyecto.
- Productos entregables del proyecto.
- Restricciones del proyecto.
- Asunciones del proyecto.
- Organización inicial del proyecto.
- Requisitos de gestión de la configuración del proyecto.
- Requisitos de aprobación.

**Herramientas y Técnicas:**

- ✓ Metodología de Dirección de Proyectos.

**Actividad # 2: Adquirir Equipo Proyecto.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto. específicamente Plan de Gestión de

**Artefactos de Salida:**

- ✓ Asignaciones del Personal del Proyecto.

Personal.

- ✓ Disponibilidad de Recursos.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión del Personal.

**Descripción:**

Se obtienen los recursos humanos necesarios para completar el proyecto. Para llevar a cabo la selección del personal se pueden realizar encuestas, formularios y entrevistas que permitan conformar el Equipo de trabajo de acuerdo con sus competencias. Esta actividad forma parte del área de conocimiento del PMBOK: Gestión de Recursos Humanos.

**Herramientas y Técnicas:**

- ✓ Adquisición.
- ✓ Encuestas.
- ✓ Formularios.
- ✓ Entrevistas.

**Proceso # 3: Reunión de Planificación del Proyecto.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Descripción:**

Este proceso da inicio a la planificación del proyecto donde se desarrollan actividades como Planificación de Recursos Humanos, Planificación de Comunicación, Planificación de Gestión de Riesgos y Planificación de Calidad.

**Actividad # 3.1: Planificar Recursos Humanos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Plan de Gestión de Proyecto actualizado en el acápite Plan de

Gestión de Personal donde se especifican Roles y Responsabilidades.

**Descripción:**

Se determinan los roles del proyecto, las responsabilidades y las relaciones de informe y todos estos aspectos deben quedar reflejados en el plan de gestión de personal. Además, puede incluir los criterios para eximir a los miembros del proyecto, la identificación de las necesidades de formación, los planes relativos a recompensas y reconocimiento, consideraciones sobre cumplimiento, polémicas de seguridad y el impacto del plan de gestión de personal sobre la organización. Esta actividad forma parte del área de conocimiento del PMBOK: Gestión de Recursos Humanos.

**Herramientas y Técnicas:**

- ✓ Organigramas y Descripciones de Cargos.
- ✓ Teoría de la Organización.

**Actividad # 3.2: Planificar Comunicación**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de las Comunicaciones.

**Descripción:**

Se determinan las necesidades de información y comunicación de los interesados; quién necesita, qué información, cuándo la necesitará, cómo le será suministrada y por quién. Se determina el medio tecnológico, los formatos y las fuentes para emitir la información, se definen los sistemas de trabajo y la frecuencia con que serán emitidos los informes o partes, además de la estructura de cada uno y los objetivos que persigue. Esta actividad forma parte del área de conocimiento del PMBOK: Gestión de Comunicación del Proyecto.

**Herramientas y Técnicas:**

- ✓ Análisis de Requisitos de Comunicaciones y Tecnología de las Comunicaciones

**Actividad # 3.3: Planificar Gestión de Riesgos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Riesgos.

**Descripción:**

Se decide cómo abordar y llevar a cabo las actividades de gestión de riesgos de un proyecto, garantizando que el nivel, el tipo y la visibilidad de la gestión de riesgos sean acordes con el riesgo y la importancia del proyecto para la organización, a fin de proporcionar recursos y tiempo suficiente para las actividades de gestión de riesgos, y para establecer una base acordada para evaluar los riesgos. Esta actividad forma parte del área de conocimiento del PMBOK: Gestión de Riesgos del Proyecto.

**Herramientas y Técnicas:**

- ✓ Reuniones de Planificación y Análisis.

**Actividad # 3.4: Planificar Calidad.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Métricas de Calidad.
- ✓ Plan de Gestión de Proyecto actualizado en el acápite Plan de Gestión de Calidad especificando la Listas de Control de Calidad y el Plan de Mejoras del Proceso.

**Descripción:**

Se identifican qué normas de calidad son relevantes para el proyecto y determinar cómo satisfacerlas. Debería realizarse de forma paralela a los demás procesos de planificación del proyecto. Los cambios requeridos en el producto para cumplir con las normas de calidad identificadas pueden requerir ajustes en el coste o en el cronograma. La calidad deseada del producto puede requerir un análisis detallado de riesgos de un problema identificado. Esta actividad forma parte del área de conocimiento del PMBOK: Gestión de Calidad del Proyecto.

**Herramientas y Técnicas:**

- ✓ Análisis Coste-Beneficio.
- ✓ Coste de Calidad.
- ✓ Herramientas Adicionales de Planificación de Calidad.

**2.1.2 Fase: Especulación.**

**Proceso # 4: Reunión de Planificación del Primer Sprint.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Descripción:**

En esta reunión se definen las actividades a Ejecutar en el Primer Sprint, se especifican concretamente los riesgos y se realiza un análisis de los recursos y costos de las actividades.

**Actividad # 4.1: Definir Actividades**

**Roles:**

- ✓ Dueño del Producto.
- ✓ Scrum Máster.
- ✓ Cliente.

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión del Alcance.

**Artefactos de Salida:**

- ✓ Lista de actividades.

**Descripción:**

Definir las actividades implica documentar el trabajo que se planifica realizar. En la actividad se identificarán los productos entregables al nivel más bajo de la estructura de desglose del trabajo que se

denomina paquete de trabajo. Los paquetes de trabajo del proyecto están descompuestos en componentes más pequeños denominados actividades del cronograma, para proporcionar una base con el fin de controlar el cronograma, ejecutar las actividades ya definidas y planificadas, y supervisar y controlar el trabajo del proyecto, de tal modo que se cumplan los objetivos del proyecto.

**Herramientas y Técnicas:**

- ✓ Descomposición.
- ✓ Plantillas.

**Actividad # 4.2: Analizar Riesgos de las actividades.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.
  - Plan de Gestión de Cronograma.
  - Plan de Gestión de Costes.
  - Plan de Gestión de Riesgos.
- ✓ Registro de Riesgos.

**Artefactos de Salida:**

- ✓ Registro de Riesgo (actualizado).

**Descripción:**

Es una actividad en la cual se identifican los riesgos de las actividades y se analizan tanto cualitativa como cuantitativamente. La identificación de riesgos determina qué riesgos pueden afectar al proyecto y documenta sus características. Es iterativa porque se pueden descubrir nuevos riesgos a medida que el proyecto avanza a lo largo de su ciclo de vida. La Identificación de Riesgos suele llevar al proceso Análisis Cualitativo de Riesgos y como alternativa, puede llevar directamente al proceso Análisis Cuantitativo de Riesgos. El Análisis Cualitativo evalúa la prioridad de los riesgos identificados usando la probabilidad de ocurrencia, el impacto correspondiente sobre los objetivos del proyecto si los riesgos efectivamente ocurren, así como otros factores. El Análisis Cuantitativo se realiza respecto a los riesgos priorizados en el proceso Análisis Cualitativo por tener un posible impacto significativo sobre las demandas concurrentes del proyecto.

**Herramientas y Técnicas:**

- ✓ Matriz de probabilidad e impacto.

- ✓ Evaluación de urgencia del riesgo.
- ✓ Técnicas de análisis cuantitativo de riesgos y modelado.

**Actividad # 4.3: Estimar los Recursos de las actividades.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Lista de Requerimientos.
- ✓ Disponibilidad de Recursos.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Requisitos de Recursos de las Actividades.
- ✓ Estructura de Desglose de Recursos.
- ✓ Cambios Solicitados.

**Descripción:**

La estimación de recursos de las actividades involucra determinar cuáles son los recursos (personas, Equipos, o material), qué cantidad de cada recurso se utilizará y cuándo estará disponible cada recurso para realizar las actividades del proyecto. Estimación de Recursos de las Actividades se coordina estrechamente con el proceso Estimación de Costes.

**Herramientas y Técnicas:**

- ✓ Análisis de Alternativas.
- ✓ Estimación Ascendente.

**Actividad # 4.4: Estimar los Costes de las actividades.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto
  - Plan de Gestión del Cronograma.
  - Plan de Gestión del Personal.

**Artefactos de Salida:**

- ✓ Estimaciones de Costes de las Actividades.
- ✓ Cambios Solicitados.
- ✓ Requisitos para la financiación del proyecto.

- Registro de Riesgos.
- ✓ Plan de Gestión de Proyecto actualizado en el acápite Plan de Gestión de Costes específicamente en la línea base de coste.

**Descripción:**

Implica desarrollar una aproximación de los costes de los recursos necesarios para completar cada requisito del cronograma. Al hacer una aproximación de los costes, el estimador debe considerar las posibles causas de variación de las estimaciones de costes, incluyendo los riesgos. La actividad incluye sumar los costes estimados de actividades individuales o paquetes de trabajo a fin de establecer una línea base de coste.

**Herramientas y Técnicas:**

- ✓ Determinación de Tarifas de Costes de Recursos.
- ✓ Suma de Costes.
- ✓ Conciliación del Límite de la Financiación.

**Proceso # 5: Ejecución del Primer Sprint.**

**Roles:**

- ✓ Scrum Máster
- ✓ Dueño del Producto
- ✓ Equipo

**Descripción:**

A partir de una reunión con el cliente se definen los procesos y macro procesos que componen las diferentes áreas del proyecto, de igual manera se priorizan estas áreas de acuerdo con su nivel de importancia dentro de la empresa.

**Actividad # 5.1: Desarrollar y Gestionar el Equipo de Desarrollo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Equipo
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Asignaciones del Personal del Proyecto.

**Artefactos de salida:**

- ✓ Evaluación del Rendimiento del Equipo.

- ✓ Plan de Gestión de Personal.
- ✓ Disponibilidad de Recursos.
- ✓ Roles y Responsabilidades: Una lista de los roles y las responsabilidades del personal se utiliza para supervisar y evaluar el rendimiento.
- ✓ Evaluación del rendimiento del Equipo.
- ✓ Cambios Solicitados.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Personal.

**Descripción:**

Desarrollar el Equipo del Proyecto mejora las competencias e interacciones de los miembros del Equipo a fin de elevar el rendimiento del proyecto mediante talleres y cursos de capacitación. Gestionar el Equipo del Proyecto implica hacer un seguimiento del rendimiento de los miembros del Equipo, proporcionar retroalimentación, resolver polémicas y coordinar cambios a fin de mejorar el rendimiento del proyecto. El Scrum Máster gestiona los conflictos, resuelve las polémicas y evalúa el rendimiento de los miembros del Equipo.

**Herramientas y Técnicas:**

- ✓ Reubicación.
- ✓ Reconocimiento y Recompensas.
- ✓ Observación y Conversación.
- ✓ Gestión de Conflictos.

**Subproceso # 5.2: Actividades de Desarrollo.**

**Roles:**

- ✓ Cliente
- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Descripción:**

Se identifican áreas de procesos y subprocesos asociados a cada área identificada, se obtendrán modelos globales y diagramas de la organización, se obtendrá un mapa de procesos detallado y el diagrama del proceso general, facilitando la base para todo el desarrollo posterior ya que garantiza que el cliente y desarrolladores tengan una visión de la organización.

**Actividad # 5.2.1: Taller con el Cliente.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo
- ✓ Cliente

**Artefactos de entrada:**

- ✓ Información del negocio.

**Artefactos de salida:**

- ✓ Lista de áreas de procesos.

**Descripción:**

En esta actividad se identifican las áreas de procesos y se priorizan, permitiendo una mayor abstracción de lo que quiere el cliente, para posteriormente facilitar el proceso de identificación de un Product Backlog por cada área de proceso identificada.

**Actividad # 5.2.2: Obtener Modelo Global.**

**Roles:**

- ✓ Cliente
- ✓ Dueño del Producto
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Información del negocio.
- ✓ Lista de áreas de procesos.

**Artefactos de salida:**

- ✓ Organigrama.
- ✓ Objetivos de la organización.
- ✓ Cadena de valor.

**Descripción:**

En esta actividad se identificarán los contenedores de información que serán las personas o documentos que proporcionarán la información necesaria para llevar a cabo el proceso descriptivo para el modelo, teniendo esto identificado se obtendrán modelos globales y diagramas de la organización que tiene como objetivo el entendimiento general de la misma. La obtención de estos modelos es la base para todo el desarrollo posterior del proceso de modelado del negocio ya que garantiza que el cliente y desarrolladores tengan una visión de la organización permitiendo que sean identificados los macro procesos de la empresa.

**Actividad # 5.2.3: Taller con el Equipo**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Lista de áreas de procesos.
- ✓ Organigrama.

**Artefactos de salida:**

- ✓ Lista de procesos asociados a cada área de procesos.

**Descripción:**

A partir del área del proceso, en esta actividad se obtienen los procesos específicos de cada área de proceso.

**Actividad # 5.2.4: Confeccionar Mapa de Procesos.**

**Roles:**

- ✓ Scrum Máster
- ✓ Dueño del Producto
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Modelos globales.
- ✓ Inventario de proceso.

**Artefactos de salida:**

- ✓ Mapa de proceso.

**Descripción:**

Con la realización de esta actividad se obtendrá un mapa de procesos detallado donde se podrán apreciar las entradas y salidas de cada uno de los macro procesos y su interacción entre ellos.

**Herramientas y Técnicas:**

- ✓ Business Process Management (BPM).
- ✓ Bizagi.

**Actividad # 5.2.5: Modelar el Proceso General.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Mapa de procesos.
- ✓ Organigrama.

**Artefactos de salida:**

- ✓ Diagrama del Proceso General.

**Descripción:**

Se lleva a cabo la realización de la matriz de integración de procesos contra el sistema, se realizan los modelos conceptuales, el diagrama del proceso general, teniendo en cuenta las interacciones, se especifican los roles del negocio y se identifican los problemas que existen en el negocio y sus metas.

**Herramientas y Técnicas:**

- ✓ BPM.
- ✓ Visual Paradigm.
- ✓ Bizagi.

**Actividad # 5.2.6: Indicadores de Procesos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Especificación de proceso de negocio.

**Artefactos de salida:**

- ✓ Listado de indicadores de proceso.

**Descripción:**

Se definirán los indicadores para evaluar el desempeño tanto del negocio como de los procesos. Esta se realizará a partir de un análisis del proceso y sus principales objetivos definiendo indicadores de resultados así como de rendimiento permitiendo monitorizar y optimizar el proceso.

**Herramientas y Técnicas:**

- ✓ BPM.
- ✓ Visual Paradigm.
- ✓ Bizagi.

**Subproceso # 5.3: Reunión Diaria de SCRUM.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Descripción:**

Es una reunión corta que se celebra todos los días a una hora prefijada. Todo el Equipo asiste a la reunión. Incluye la actividad clave de la Reunión Diaria de SCRUM, Informar progreso y obstáculo. Es la oportunidad del Equipo de informar a los demás sobre el progreso y los obstáculos. Alguien anota los

bloqueos, y el Scrum Máster se responsabiliza de ayudar a los miembros del Equipo a resolverlos. En esta reunión se controlan el cronograma, los riesgos y se gestiona el Equipo con el fin de mejorar el rendimiento del proyecto, además se actualiza la Pila de Entrega y la gráfica de trabajo restante del Sprint.

**Actividad # 5.3.1: Informar Progreso y Obstáculo.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.
- ✓ Gráfica Burn Down.

**Artefactos de salida:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión del Cronograma, Plan de Gestión del Personal y Plan de Gestión de Riesgos.
- ✓ Cambios solicitados.

**Descripción:**

Es la actividad clave de la Reunión Diaria de SCRUM. En ella cada miembro del Equipo informa sobre tres cosas a los otros miembros del Equipo: qué han hecho desde la última reunión, qué tienen planificado hacer antes de la siguiente reunión y bloqueo o impedimento que tengan. Alguien anota los bloqueos, y el Scrum Máster se responsabiliza de ayudar a los miembros del Equipo a resolverlos.

**Herramientas y Técnicas:**

- ✓ Gráfico Burn Down.

**Subproceso # 5.3.2: Actualizar Pila de Sprint y Gráfica de Trabajo Restante de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo

**Descripción:**

Llegados a este punto se han terminado algunos elementos del Sprint. El Dueño de Producto es responsable de asegurarse que estos cambios se reflejan en la Pila de Entrega (y más ampliamente, en la Pila de Producto). Además, SCRUM incluye una gráfica de Trabajo Restante de la Entrega del Sprint que

muestra el progreso hecho hacia la fecha de entrega. Incluye las actividades de control de cronograma y alcance.

**Actividad # 5.3.2.1: Controlar Cronograma de las Tareas del Sprint**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión del Cronograma.
- ✓ Línea Base del Cronograma.
- ✓ Solicitudes de Cambio Aprobadas.

**Artefactos de salida:**

- ✓ Datos del Modelo de Cronograma (Actualizaciones).
- ✓ Gráfica de trabajo restante actualizada.
- ✓ Línea Base del Cronograma (Actualizaciones).
- ✓ Sprint Backlog actualizado.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión del Cronograma.

**Descripción:**

El control del cronograma implica: determinar el estado actual del cronograma del proyecto, influir sobre los factores que crean cambios en el cronograma, gestionar los cambios reales a medida que suceden. El control del cronograma es una parte del proceso Control Integrado de Cambios.

**Herramientas y Técnicas:**

- ✓ Informe del Avance.
- ✓ Sistema de control de Cambios del Cronograma.
- ✓ Gráfica Burn-Down.

**Actividad # 5.3.2.2: Controlar Alcance de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Product Backlog.
- ✓ Plan de Gestión del Alcance del Proyecto.

**Artefactos de salida:**

- ✓ Product Backlog actualizado.
- ✓ Plan de Gestión de Proyecto actualizado en el acápito Plan de Gestión del Alcance específicamente en la Línea Base del Alcance.
- ✓ Cambios Solicitados.

**Descripción:**

Se encarga de influir sobre los factores que crean cambios en el alcance del proyecto y de controlar el impacto de dichos cambios, asegura que todos los cambios solicitados y las acciones correctivas recomendadas se procesen a través del proceso Control Integrado de Cambios del Proyecto. El control del alcance del proyecto también se usa para gestionar los cambios reales cuando se producen, y está integrado con los demás procesos de control. Los cambios no controlados a menudo se denominan corrupción del alcance del proyecto. Los cambios son inevitables, con lo cual se impone algún tipo de proceso de control de cambios.

**Herramientas y Técnicas:**

- ✓ Sistema de Control de Cambios.
- ✓ Análisis de Variación.
- ✓ Replanificación.

**Subproceso # 5.4: Actividades de Gestión.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Descripción:**

En este subproceso se llevan a cabo las actividades que permiten un control de las tareas que se ejecutan en el Sprint con el objetivo de elevar al máximo la calidad del proyecto en función del trinomio coste, tiempo y alcance.

**Actividad # 5.4.1: Controlar Coste de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Producto

- ✓ Equipo
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Línea Base de Coste.
- ✓ Requisitos para la financiación del proyecto.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de salida:**

- ✓ Estimaciones de Costes (Actualizaciones).
- ✓ Línea Base de Coste (Actualizaciones)
- ✓ Mediciones del Rendimiento.
- ✓ Cambios Solicitados.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de gestión de Costes.

**Descripción:**

El control de costes del proyecto busca las causas de las variaciones positivas y negativas. Incluye influir sobre los factores que producen cambios en la línea base de coste, asegurarse de que los cambios solicitados sean acordados, gestionar los cambios reales cuando y a medida que se produzcan, asegurar que los posibles sobrecostes no excedan la financiación autorizada periódica y total para el proyecto, realizar el seguimiento del rendimiento del coste para detectar y entender las variaciones con respecto a la línea base de coste, registrar todos los cambios pertinentes con precisión en la línea base de coste, evitar que se incluyan cambios incorrectos, inadecuados o no aprobados en el coste o en el uso de recursos informados, informar los cambios aprobados a los interesados pertinentes y actuar para mantener los sobrecostes esperados dentro de límites aceptables.

**Herramientas y Técnicas:**

- ✓ Sistema de Control de Cambios del Coste.
- ✓ Análisis de Medición del Rendimiento.
- ✓ Gestión de Variación.

**Actividad # 5.4.2: Realizar Control Calidad de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

**Artefactos de salida:**

- ✓ Plan de Gestión de Calidad.
- ✓ Métricas de Calidad.
- ✓ Listas de Control de Calidad.
- ✓ Información sobre rendimiento del trabajo.
- ✓ Solicitudes de cambio aprobadas.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Calidad.
- ✓ Línea Base de Calidad (Actualizaciones)
- ✓ Mediciones de control de calidad.
- ✓ Productos entregables validados.
- ✓ Cambio Solicitados.

**Descripción:**

Realizar control de calidad implica supervisar los resultados específicos obtenidos de cada tarea, para determinar si cumplen con las normas de calidad relevantes e identificar los modos de eliminar las causas de resultados insatisfactorios. Las normas de calidad incluyen los objetivos de los procesos y productos del proyecto y llevar a cabo acciones para eliminar las causas de un rendimiento insatisfactorio del proyecto. Además, de que se realizan las actividades planificadas y sistemáticas de calidad a fin de garantizar que el proyecto utilice todos los procesos necesarios para satisfacer los requisitos.

**Herramientas y Técnicas:**

- ✓ Diagrama de Causa y Efecto.
- ✓ Diagramas de Control.
- ✓ Inspección.
- ✓ Revisión de Reparación de Defectos.

**Actividad # 5.4.3: Controlar Riesgos de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión de Riesgos.
- ✓ Registro de Riesgos.

**Artefactos de salida:**

- ✓ Registro de Riesgos (Actualizaciones).
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Riesgos.

**Descripción:**

En la actividad se determinan acciones para mejorar las oportunidades y reducir las amenazas a las Tareas del Sprint. Incluye la identificación y asignación de una o más personas para que asuma la responsabilidad de cada respuesta a los riesgos acordados y financiados. Aborda los riesgos en función de su prioridad, introduciendo recursos y actividades en el presupuesto, cronograma y Plan de Gestión de Proyecto, según sea necesario.

**Herramientas y Técnicas:**

- ✓ Estrategias para Riesgos Positivos u Oportunidades.
- ✓ Estrategia Común ante Amenazas y Oportunidades.
- ✓ Estrategia de Respuesta para Contingencias.

**Actividad # 5.4.4: Dar Respuesta a los Riesgos de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión de Riesgos.
- ✓ Registro de Riesgos.

**Artefactos de salida:**

- ✓ Registro de Riesgos (Actualizaciones)
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Riesgos.
- ✓ Cambios Solicitados.

**Descripción:**

La actividad implica elegir estrategias alternativas, ejecutar un plan para contingencias o de reserva, adoptar acciones correctivas y modificar el Plan de Gestión de Proyecto. El propietario da respuesta a los riesgos e informa periódicamente al director del proyecto acerca de la efectividad del plan, de cualquier efecto no anticipado y cualquier corrección sobre la marcha que sea necesaria para gestionar el riesgo correctamente. También se incluye la actualización de los activos de los procesos de la organización, incluidas las bases de datos de las lecciones aprendidas del proyecto y las plantillas de gestión de riesgos para beneficio de proyectos futuros.

**Herramientas y Técnicas:**

- ✓ Reevaluación de los Riesgos.
- ✓ Análisis de Variación y de Tendencias.
- ✓ Análisis de Reserva.

**Actividad # 5.5: Reunión de Revisión del primer Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint terminado.
- ✓ Feedback (demo) del producto.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de salida:**

- ✓ Entregables.
- ✓ Plan de Gestión de Proyecto actualizado.

**Descripción:**

Reunión donde el Equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir. En función de los resultados mostrados y de los cambios en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto. La revisión incluye un demo de lo que el Equipo construyó durante el Sprint.

**Actividad # 5.6: Reunión Retrospectiva del primer Sprint.**

**Roles:**

- ✓ Scrum Máster
- ✓ Equipo
- ✓ Dueño del Producto

**Descripción:**

Con el objetivo de mejorar de manera continua su productividad, el Equipo analiza cómo ha sido su manera de trabajar durante la iteración: qué cosas han funcionado bien, cuáles hay que mejorar, qué cosas quiere probar hacer en la siguiente iteración, qué ha aprendido y cuáles son los problemas que podrían impedirle progresar adecuadamente. El facilitador se encargará de ir eliminando los obstáculos identificados que el propio Equipo no pueda resolver por sí mismo. Se actualiza el cronograma y se lleva a cabo un taller con el Equipo para su preparación.

**Proceso # 6: Crear Product Backlog.**

**Roles:**

- ✓ Scrum Máster

- ✓ Cliente
- ✓ Dueño del Producto

**Descripción:**

En este proceso se toma la lista de áreas construida anteriormente en la Ejecución del Primer Sprint y a partir de ella se crea y se define la estructura de desglose del trabajo, se realiza el cronograma general del proyecto, se identifican las prioridades de entrega, se estima el precio del producto, se aplican las métricas de costo y tiempo, se acuerda la definición de completado y se definen los casos de pruebas generales, toda esta información complementada tendrá como nombre Pila del Producto o Product Backlog. Esta pila de producto existe y evoluciona a lo largo de la vida del proyecto; es el plan de trabajo del producto. La Pila del Producto es actualizada continuamente por el Dueño de Producto para reflejar los cambios de necesidad del cliente, nuevas ideas, movimientos de los competidores y dificultades técnicas. Durante este proceso se crea el plan de gestión del alcance del proyecto que documenta cómo se definirá, verificará y controlará el alcance del proyecto.

**Actividad # 6.1: Crear estructura de desagregación del trabajo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Lista de áreas de procesos.

**Artefactos de salida:**

- ✓ Product Backlog creado.

**Descripción:**

En esta actividad se organiza y define el alcance total del proyecto. La EDT subdivide el trabajo del proyecto en porciones de trabajo más pequeñas y fáciles de manejar, donde cada nivel descendente de la EDT representa una definición cada vez más detallada del trabajo del proyecto. El trabajo planificado comprendido dentro de los componentes de la EDT del nivel más bajo, denominados paquetes de trabajo, puede programarse, supervisarse, controlarse y estimarse sus costes. Además, se planifica el cronograma general del proyecto por cada uno de los componentes del EDT identificados.

**Herramientas y Técnicas:**

- ✓ Product Backlog.

**Actividad # 6.2: Crear Cronograma.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog creado.

**Artefactos de salida:**

- ✓ Cronograma del Proyecto.
- ✓ Datos del Modelo de Cronograma.

**Descripción:**

En esta actividad se determinan las fechas de inicio y finalización planificadas para las actividades del proyecto. El desarrollo del cronograma exige que se revisen y se corrijan las estimaciones de duración y las estimaciones de los recursos para crear un cronograma del proyecto aprobado que pueda servir como línea base con respecto a la cual poder medir el avance.

**Herramientas y Técnicas:**

- ✓ Software de Gestión de Proyectos.

**Actividad # 6.3: Identificar las prioridades de entrega.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

- ✓ Product Backlog (actualizado).

**Descripción:**

En esta actividad se define la prioridad con que se deben garantizar la realización de cada una de las actividades en dependencia a la importancia que requiera su ejecución, esto se lleva a cabo a través de la planificación del póker, juego en que el Equipo debe estimar el esfuerzo de las tareas con criterio de “juicio de expertos”.

**Herramientas y Técnicas:**

- ✓ Planificación del Póker.

**Actividad # 6.4: Estimar precio del producto.**

**Roles:**

- ✓ Scrum Máster

- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

- ✓ Product Backlog (actualizado).

**Descripción:**

En esta actividad en dependencia de los costos y recursos que se necesiten para la elaboración o ejecución de cada una de las actividades que darán composición al producto, se estimará un precio del producto que estará en correspondencia al esfuerzo realizado por la ejecución de las mismas.

**Herramientas y Técnicas:**

- ✓ Product Backlog.

**Actividad # 6.5: Aplicar métricas de costo y tiempo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

Product Backlog (actualizado).

**Descripción:**

Se aplican las métricas de coste y tiempo para indicar la calidad del producto, evaluar la productividad del Equipo y evaluar los beneficios en términos de productividad y calidad, permitiendo la utilización adecuada y efectiva de recursos, con el objetivo de establecer una línea base para la estimación del proyecto.

**Herramientas y Técnicas:**

- ✓ Product Backlog.

**Actividad # 6.6: Planificar Riesgos.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 3.3: Planificar Gestión de Riesgos.**

**Actividad # 6.7: Acordar definición de completado.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

- ✓ Definición de completado.

**Descripción:**

Se acuerda cómo será completado el producto en función de determinar hasta dónde se quiere llegar, teniendo en cuenta las especificidades de cada área. Además, se debe asegurar que el incremento de producto es potencialmente entregable al cliente al finalizar cada iteración, no hay tareas pendientes que puedan impedir utilizar los resultados del proyecto lo antes posible. Debe incluir lo necesario para considerar de manera clara que el cliente obtendrá lo que necesita según sus criterios de aceptación y de calidad.

**Actividad # 6.8: Definir casos de prueba.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Definición del alcance.

**Artefactos de salida:**

- ✓ Plan de Prueba (creado).

**Descripción:**

Se definen los casos de prueba especificando por cada requerimiento como debe ser posible de probar, por tanto, debería asociarse un caso de prueba a cada requerimiento. Un caso de prueba para requisitos generalmente consiste en definir las entradas, salidas del software y acciones del usuario para que se pueda completar lo que expresa el requisito.

**Proceso # 7: Definir Product Backlog Específico.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente

- ✓ Dueño del Producto

**Descripción:**

En el proceso se definen los Subproducto Backlog a implementar en las siguientes semanas, estos son escogidos del Product Backlog General. Es importante destacar que van a existir tantos Product Backlog Específicos como Áreas de procesos se hayan identificado.

**Actividad # 7.1: Crear estructura de desagregación del trabajo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de entrada:**

- ✓ Lista de procesos.

**Artefactos de salida:**

- ✓ Subproducto Backlog creado.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 6.1: Crear estructura de desagregación del trabajo.**

**Herramientas y Técnicas:**

- ✓ Product Backlog.

**Actividad # 7.2: Crear Cronograma.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Subproducto Backlog creado.

**Artefactos de salida:**

- ✓ Cronograma del Proyecto.
- ✓ Datos del Modelo de Cronograma.

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.2: Crear cronograma.**

**Herramientas y Técnicas:**

- ✓ Software de Gestión de Proyectos.

**Actividad # 7.3: Identificar las prioridades de entrega.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Subproducto Backlog.

**Artefactos de salida:**

- ✓ Subproducto Backlog (actualizado).

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.3: Identificar las prioridades de entrega.**

**Herramientas y Técnicas:**

- ✓ Planificación del Póker.

**Actividad # 7.4: Estimar precio del producto.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Subproducto Backlog.

**Artefactos de salida:**

- ✓ Subproducto Backlog (actualizado).

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.4: Estimar precio del producto.**

**Actividad # 7.5: Aplicar métricas de costo y tiempo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Subproducto Backlog.

**Artefactos de salida:**

- ✓ Subproducto Backlog (actualizado).

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.5: Aplicar métricas de costo y tiempo.**

**Actividad # 7.6: Planificar Riesgos.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 3.3: Planificar Gestión de Riesgos.**

**Actividad # 7.7: Acordar definición de completado.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Subproducto Backlog.

**Artefactos de salida:**

- ✓ Definición de completado.

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.7: Acordar definición de completado.**

**Herramientas y Técnicas:**

- ✓ BPM.
- ✓ Visual Paradigm.
- ✓ Bizagi.

**Actividad # 7.8: Definir casos de prueba.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Definición del alcance.

**Artefactos de salida:**

- ✓ Plan de Prueba (actualizado).

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Actividad # 6.8: Definir casos de prueba.**

**Proceso # 8: Reunión de Selección de arquitectura candidata.**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

En esta reunión se analiza la arquitectura del negocio como el conjunto de elementos relacionados entre sí por medio de una funcionalidad determinada que representa la estructura y la organización de un “Sistema de Negocio”. Incluye la descripción de los procesos y estructuras básicas del negocio, y las diferentes vistas (la de mercado, de procesos de negocio, de organización, de recursos humanos, del dominio, geográfica y de comunicación), esto permite que se lleve a cabo la transformación organizacional que se orienta a la evolución gradual de cada uno de los componentes que la integran, ejemplo: estrategias, técnicas, metodología e incluso personal, sugiere analizar adónde se quiere llegar en cada línea de producción, identificar cuáles son las competencias actuales, conformar una alineación estratégica que no es más que la correspondencia entre lo que se dice y lo que se hace, tener identificada la disponibilidad de recursos y que se evidencie la existencia de la operacionalización de la estrategia, es decir, convertir en acciones y tareas las estrategias identificadas.

**Actividad # 8.1: Definir Línea Base.**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Descripciones del negocio.

**Artefactos de salida:**

- ✓ Documento de Arquitectura creado.

**Descripción:**

Durante esta actividad se identifican, describen y especifica el uso de todas las herramientas que brindarán el soporte a la arquitectura del proyecto.

**Actividad # 8.2: Identificar los escenarios arquitectónicos.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

- ✓ Documento de arquitectura actualizado.

**Descripción:**

Se definen los estándares y guías para la adquisición y despliegue de herramientas, bloques de construcción de aplicaciones, servicios de infraestructura, y componentes de conectividad de red.

**Herramientas y Técnicas:**

- ✓ Tormenta de ideas.

**Actividad # 8.3: Identificar las necesidades técnicas y tecnológicas del Equipo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Cliente
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Product Backlog.

**Artefactos de salida:**

- ✓ Product Backlog (actualizado).

**Descripción:**

Se identifica a nivel de detalle cuántas PC necesita el Equipo para trabajar, cuántos servidores y todas las necesidades en cuanto a hardware se refiere, además del tipo de sistema operativo, plataforma donde se implementará el software, lenguaje que se utilizará para programar, condiciones físicas de las PC donde se trabajará, entre otras especificaciones que se acordarán en dependencia de las necesidades de cada proyecto.

**Herramientas y Técnicas:**

- ✓ Product Backlog.

**Actividad # 8.4: Identificar restricciones del diseño.**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Documento de Arquitectura.

**Artefactos de salida:**

- ✓ Documento de arquitectura actualizado.

**Descripción:**

Se define el gestor de base de datos, el lenguaje de programación con que el Equipo de desarrollo trabajará, se especificará el framework a utilizar y se determina el rendimiento en cuanto a soporte técnico.

**Actividad # 8.5: Identificar componentes**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Descripción del negocio.

**Artefactos de salida:**

- ✓ Lista de componentes identificados.

**Descripción:**

Se identifican los componentes que serán la unidad de construcción elemental del diseño físico permitiendo saber si es reusable a través de la aplicación, si encapsula sus funciones y datos así como la contención de otros componentes.

**Actividad # 8.6: Priorizar componentes.**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Lista de componentes identificados.

**Artefactos de salida:**

- ✓ Lista de componentes priorizados.

**Descripción:**

Se le asigna una prioridad a cada componente de acuerdo con la repercusión que tengan estos en el diseño.

**Actividad # 8.7: Asociar requisitos a los componentes.**

**Roles:**

- ✓ Dueño Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

**Artefactos de salida:**

- ✓ Lista de componentes priorizados.
- ✓ Matriz de Asociación.
- ✓ Product Backlog.

**Descripción:**

Se lleva a cabo un análisis de los requisitos y se identifican cuáles pueden ser asociados a los componentes definidos anteriormente.

**2.1.3 Fase: Exploración.**

**Proceso # 9: Ejecutar Product Backlog específico.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se comienza a ejecutar el Product Backlog a partir de la desagregación realizada con el objetivo de lograr un mayor entendimiento de cada una de las áreas que debe cubrir el proyecto en la informatización.

**Subproceso # 9.1: Reunión de Planificación del Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se procede a la desagregación de los requisitos en tareas que posteriormente serán llevadas a funcionalidades identificándose cada una de esas tareas estimando coste y recursos.

**Actividad # 9.1.1: Definir Actividades.**

**Roles:**

- ✓ Equipo
- ✓ Dueño del Producto
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión del Alcance.

**Artefactos de salida:**

- ✓ Sprint Backlog creado.

- ✓ Product Backlog.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 4.1: Definir Actividades.**

**Actividad # 9.1.2: Analizar Riesgos de las Actividades.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto.
- ✓ Registro de Riesgos.

**Artefactos de Salida:**

- ✓ Registro de Riesgo (actualizado).

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 4.2: Analizar Riesgos de las actividades.**

**Actividad # 9.1.3: Estimar Recursos de las Actividades.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Lista de Requerimientos.
- ✓ Disponibilidad de Recursos.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de Salida:**

- ✓ Requisitos de Recursos de las Actividades.
- ✓ Estructura de Desglose de Recursos.
- ✓ Cambios Solicitados.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 4.3: Estimar Recursos de las Actividades.**

**Actividad # 9.1.4: Estimar Costes de las Actividades.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster

- ✓ Cliente

**Artefactos de Entrada:**

- ✓ Plan de Gestión de Proyecto
  - Plan de Gestión del Cronograma.
  - Plan de Gestión del Personal.
  - Registro de Riesgos.

**Artefactos de Salida:**

- ✓ Estimaciones de Costes de las Actividades.
- ✓ Cambios Solicitados.
- ✓ Requisitos para la financiación del proyecto.
- ✓ Plan de Gestión de Proyecto actualizado en el acápite Plan de Gestión de Costes específicamente en la línea base de coste.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 4.4: Estimar Costes de las Actividades.**

**Subproceso # 9.2: Desarrollar Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se lleva a cabo el desarrollo del producto mediante actividades tanto de desarrollo de software como de gestión de proyectos, que permiten llevar a cabo el análisis, diseño, implementación, prueba, control de calidad, control de riesgos y la actualización del cronograma a seguir. Este subproceso se aplicará de igual manera en todos los Sprint identificados.

**Actividad # 9.2.1: Desarrollar y Gestionar el Equipo de Desarrollo.**

**Roles:**

- ✓ Scrum Máster
- ✓ Equipo
- ✓ Dueño del Producto

**Artefactos de entrada:**

- ✓ Asignaciones del Personal del

**Artefactos de salida:**

- ✓ Evaluación del Rendimiento del

Proyecto.

- ✓ Plan de Gestión de Personal.
- ✓ Disponibilidad de Recursos.
- ✓ Roles y Responsabilidades: Una lista de los roles y las responsabilidades del personal se utiliza para supervisar y evaluar el rendimiento.
- ✓ Evaluación del rendimiento del Equipo.

Equipo.

- ✓ Cambios Solicitados.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Personal.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.1: Desarrollar y Gestionar el Equipo de Desarrollo.**

**Subproceso # 9.2.2: Actividades de Desarrollo.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

En el subproceso se llevan a cabo las actividades de desarrollo de software, es donde se refina el diseño y la arquitectura del proyecto, se lleva a cabo la implementación y se realizan las pruebas.

**Subproceso # 9.2.2.1: Definir Requisitos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Este subproceso forma parte de las tareas de desarrollo donde se definen los requisitos como parte del ciclo de vida de desarrollo de software, a un nivel muy detallado dígase tareas.

**Actividad # 9.2.2.1.1: Taller con el Cliente.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster

- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.

**Artefactos de salida:**

- ✓ Sprint Backlog actualizado.

**Descripción:**

Se identifican y priorizan los requisitos que serán analizados para conformar el diseño, permitiendo una mayor abstracción de lo que quiere el cliente.

**Herramientas y Técnicas:**

- ✓ Sprint Backlog.

**Actividad # 9.2.2.1.2: Confirmar que haya suficiente información de los requisitos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.

**Artefactos de salida:**

- ✓ Sprint Backlog actualizado.

**Descripción:**

Se conforma que exista suficiente información sobre los requisitos y se priorizan de los mismos para considerar el impacto potencial en la estructura de la arquitectura. Se determina si existe suficiente información sobre los atributos de requerimientos de calidad del sistema para hacer más fácil la selección de los patrones de diseño, ya que estos se obtienen de la prioridad que se le dé a cada requerimiento.

**Actividad # 9.2.2.1.3: Descripción de requisitos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.

**Artefactos de salida:**

- ✓ Sprint Backlog (actualizado).

**Descripción:**

Se describen los requisitos anteriormente identificados y priorizados con el objetivo de obtener una detallada explicación de lo que se quiere hacer con cada uno de ellos.

**Actividad # 9.2.2.1.4: Verificar y refinar los requisitos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se verifica que la descomposición de los elementos esté encaminada a los objetivos de los requerimientos, los atributos de requerimiento de calidad y el diseño.

**Actividad # 9.2.2.1.5: Identificar reglas.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se extrae la información originada de las políticas, reglas y regulaciones del proyecto y se describe la información como una colección de reglas referenciadas a partir de las descripciones de los requisitos.

**Actividad # 9.2.2.1.6: Identificar precondiciones.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se identifican las precondiciones que el sistema debe cumplir. Es responsabilidad del que realiza la llamada garantizar que se cumplan estas propiedades. Si las precondiciones no se cumplen, el método se puede comportar de cualquier manera, entre otras cosas, bloquear el programa, continuar con resultados incorrectos, informar al usuario del problema o recuperarse del problema con dignidad.

**Actividad # 9.2.2.1.7: Definir casos de prueba.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Definición del alcance.

**Artefactos de salida:**

- ✓ Caso de prueba.

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Subproceso # 6.8: Definir Casos de Prueba.**

**Actividad # 9.2.2.1.8: Revisión con el cliente.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Documentación del Sprint.

**Artefactos de salida:**

- ✓ Criterios de aceptación.

**Descripción:**

Se lleva a cabo con el objetivo de mostrarle al cliente todo lo que quedó definido en cuanto a los requisitos y reglas del proyecto en general y que el cliente de su punto de vista, así como su aprobación.

**Herramientas y Técnicas:**

- ✓ Criterios de aceptación.

**Subproceso # 9.2.2.2: Definir ADD.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se lleva a cabo todo lo referente al diseño de los productos que conformarán el producto general. El ADD como método para extraer la arquitectura parte del análisis con los requisitos y diseña toda la estructura lógica, convirtiéndose en la arquitectura del producto en sí.

**Actividad # 9.2.2.2.1: Definir patrones de diseño.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Documento de arquitectura.
- ✓ Lista de componentes identificados.

**Artefactos de salida:**

- ✓ Documento de arquitectura actualizado

**Descripción:**

Se definen los patrones que expresan esquemas para definir estructuras del diseño con las que construir el software. Se describen además, clases y objetos comunicándose entre sí para resolver un problema del diseño general en un contexto particular. Un patrón de diseño identifica: Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad.

**Actividad # 9.2.2.2.2: Elegir un elemento del sistema para descomponer**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se eligen los elementos del sistema que serán el centro del diseño. Es una actividad propia del ADD.

**Actividad # 9.2.2.2.3: Identificar Candidatos Arquitectónicos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Después de elegir el elemento del sistema a descomponer, se priorizan los requerimientos que afectan a este elemento. Se le asigna un rango según el impacto en la arquitectura (impacto alto, impacto medio e impacto bajo) a cada requerimiento. En caso de que un requerimiento esté definido como de alto impacto y en el transcurso del desarrollo se demuestre lo contrario pues es eliminado de la lista, mientras otros pueden ser añadidos a esta. Es una actividad propia del ADD.

**Actividad # 9.2.2.2.4: Elegir un concepto de diseño que satisfaga los controladores de arquitectura.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster

- ✓ Equipo

**Descripción:**

Se eligen los tipos de elementos que aparecerán en la arquitectura y los tipos de relaciones entre ellos. El diseño y los requerimientos de atributo de calidad son usados para determinar los tipos de elementos, relaciones y las interacciones. Se identifican las funcionalidades asociadas a los distintos tipos de elementos. Es una actividad propia del ADD.

**Actividad # 9.2.2.2.5: Crear diseño de clases**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.

**Artefactos de salida:**

- ✓ Diseño de clases.

**Descripción:**

Se crea el diseño de clases que no son más que clases con un mayor nivel de detalle, que se conciben para satisfacer las funcionalidades del sistema, teniendo en consideración la tecnología en la cual se implementará el diseño.

**Herramientas y Técnicas:**

- ✓ Visual Paradigm.

**Actividad # 9.2.2.2.6: Crear diseño de bases de datos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño de clases.
- ✓ Lista de componentes identificados.

**Artefactos de salida:**

- ✓ Diseño de base de datos.

**Descripción:**

Se diseña la base de datos, definiéndose su estructura que permitirá almacenar datos, reconocer el contenido y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usarán, esto puede ayudar a realizar un proceso del negocio para alcanzar un valor

agregado para el cliente. En este proceso se definen las clases persistentes, se refinan las clases, se clasifican las clases y los atributos, se realiza el diagrama de transición de estado, se obtienen las restricciones estáticas y las fórmulas dinámicas y se convierten las clases al medio de almacenamiento.

**Herramientas y Técnicas:**

- ✓ Visual Paradigm

**Subproceso # 9.2.2.2.7: Elaborar diseño lógico.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se elabora el diseño lógico que consiste entre otras cosas, en la realización de dos actividades de ADD:

- ❖ Crear instancias de elementos arquitectónicos y atribuir las responsabilidades: los elementos instanciados son responsabilidades asignadas de acuerdo con los tipos de elementos. Estos se derivan en requerimientos funcionales asociados a los controladores candidatos de la arquitectura. Estos elementos deben ser representados por la secuencia de las responsabilidades con los elementos hijos. Es una actividad propia del ADD.
- ❖ Definir interfaces para crear instancias de elementos: se definen los servicios y las propiedades requeridas brindadas por los elementos del diseño. En ADD estos servicios y estas propiedades son referidas las interfaces de los elementos. Es una actividad propia del ADD.

**Actividad # 9.2.2.2.8: Diseñar la integración.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Lista de componentes identificados.
- ✓ Documento arquitectura.

**Artefactos de salida:**

- ✓ Diseño de integración.

**Descripción:**

Se diseña la integración del sistema basado en los componentes y los patrones de diseños previamente identificados.

**Herramientas y Técnicas:**

- ✓ Bizagi.
- ✓ BPM.
- ✓ Visual Paradigm.

**Actividad # 9.2.2.2.9: Diseñar prototipo de interfaz.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint Backlog.

**Artefactos de salida:**

- ✓ Diseño del prototipo de interfaz de usuario.

**Descripción:**

Se lleva a cabo el diseño de la interfaz del sistema con la que el cliente interactúa.

**Herramientas y Técnicas:**

- ✓ Visual Paradigm.

**Actividad # 9.2.2.2.11: Diseñar casos de prueba.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Especificaciones funcionales del diseño.
- ✓ Especificación de los requisitos.

**Artefactos de salida:**

- ✓ Diseños de casos de prueba.

**Descripción:**

Se diseñan los casos para guiar las pruebas. Con este diseño se podrá verificar si el producto satisface los requerimientos del usuario tal y como se describe en las especificaciones de los requerimientos y si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

**Subproceso # 9.2.2.3: Implementar Funcionalidades**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Se implementan las clases y subsistemas encontrados durante el diseño. La implementación parte del resultado general del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario y ejecutable enlazándose en uno o más de estos últimos, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

**Herramientas y Técnicas:**

- ✓ Entornos de desarrollo especificado en el negocio.

**Actividad # 9.2.2.3.1: Implementar diseño de clases.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño de clases.

**Artefactos de salida:**

- ✓ Clases implementadas.

**Descripción:**

Se implementan las clases identificadas durante el diseño, debido a que son abstracciones de clases directamente utilizables en la implementación, por lo que las clases se mapean directamente en el lenguaje de programación, así como sus relaciones tienen un significado directo en dichos lenguajes.

**Actividad # 9.2.2.3.2: Implementar diseño de interfaz.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño prototipo de interfaz de usuario.

**Artefactos de salida:**

- ✓ Prototipo funcional de interfaz de usuario.

**Descripción:**

Se implementa el diseño de la interfaz de usuario propuesto con anterioridad.

**Actividad # 9.2.2.3.3: Implementar base de datos.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño de base de datos.

**Artefactos de salida:**

- ✓ Base de datos implementada.

**Descripción:**

Se implementa el diseño de la base de datos propuesto con anterioridad.

**Actividad # 9.2.2.3.4: Implementar diseño de prueba.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño de casos de prueba.

**Artefactos de salida:**

- ✓ Pruebas implementadas.

**Descripción:**

Se realiza con el objetivo de implementar el diseño de las pruebas propuesto con anterioridad.

**Actividad # 9.2.2.3.5: Implementar integración.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño integración.

**Artefactos de salida:**

- ✓ Integración implementada.

**Descripción:**

Se implementa el diseño de la integración del sistema propuesto con anterioridad para la iteración actual.

**Actividad # 9.2.2.3.6: Implementar precondiciones.**

**Roles:**

- ✓ Dueño del Producto

- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseño de precondiciones.

**Artefactos de salida:**

- ✓ Precondiciones implementadas

**Descripción:**

Esta actividad se lleva a cabo con el objetivo de implementar las precondiciones previamente identificadas.

**Subproceso # 9.2.2.4: Probar.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

El objetivo de la prueba de software es descubrir errores. Para conseguir este objetivo se planifica y se ejecuta una serie de pasos que van revisando todos los elementos del software. En todas las fases del desarrollo del proyecto hay que probar el software que se va construyendo, aunque como el grueso de la programación se realiza en la construcción, es en esa fase en la que se centran los mayores esfuerzos de este flujo. La etapa de prueba es tan o más importante que todas las realizadas hasta el momento puesto que en ella se refleja la calidad con que ha sido llevada a cabo la proyección del sistema.

**Actividad # 9.2.2.4.1: Prueba unitaria.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Casos de prueba.

**Artefactos de salida:**

- ✓ Plan de pruebas actualizado.

**Descripción:**

Es la prueba enfocada a los elementos testeables más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos y que ellos funcionen como se espera.

**Herramientas y Técnicas:**

- ✓ Pruebas de caja blanca.

**Actividad # 9.2.2.4.2: Prueba de integración.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Casos de prueba.

**Artefactos de salida:**

- ✓ Plan de pruebas actualizado.

**Descripción:**

Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Estas pruebas descubren errores en las especificaciones de las interfaces de los paquetes.

**Herramientas y Técnicas:**

- ✓ Pruebas basadas en hilos.
- ✓ Pruebas basadas en el uso.

**Actividad # 9.2.2.4.3: Ejecutar casos de prueba.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Diseños de casos de prueba.

**Artefactos de salida:**

- ✓ Plantilla de No conformidades

**Descripción:**

Al ejecutar los casos de prueba diseñados previamente según las necesidades del software, se está llevando a cabo el proceso de revisión del software y por tanto, la calidad, en caso de que se incurra en errores, estos son recogidos en un documento hasta que se erradiquen finalmente.

**Herramientas y Técnicas:**

- ✓ Pruebas de Caja Blanca
- ✓ Pruebas de caja Negra

**Actividad # 9.2.2.4.4: Ejecutar prueba funcional**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Casos de prueba.

**Artefactos de salida:**

- ✓ Plan de pruebas actualizado.

**Descripción:**

La prueba funcional se lleva a cabo con el objetivo de asegurar el trabajo apropiado de los requisitos, verificar el procesamiento, recuperación e implementación adecuada. Estas pruebas están basadas en 3 aspectos fundamentales: Pruebas de funcionalidad: pruebas fijando su atención en la validación de las funciones, métodos y servicios. Pruebas de seguridad: Asegurar que los datos o el sistema solamente es accedido por los actores deseados. Pruebas de volumen: Enfocada en la verificación de las habilidades de los programas para manejar grandes cantidades de datos, tanto entrada como salida.

**Herramientas y Técnicas:**

- ✓ Pruebas de caja negra.

**Subproceso # 9.2.3: Reunión Diaria de SCRUM.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Subproceso # 5.3: Reunión Diaria de SCRUM.**

**Subproceso # 9.2.4: Actividades de Gestión.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Para documentarse sobre el subproceso favor de dirigirse a: **Subproceso # 5.4: Actividades de Gestión.**

**Actividad # 9.2.4.1: Realizar Control Calidad de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Producto

- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión de Calidad.
- ✓ Métricas de Calidad.
- ✓ Listas de Control de Calidad.
- ✓ Información sobre rendimiento del trabajo.
- ✓ Solicitudes de cambio aprobadas.

**Artefactos de salida:**

- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Calidad.
- ✓ Mediciones de control de calidad.
- ✓ Productos entregables validados.
- ✓ Cambio Solicitados.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.4.2: Realizar Control Calidad de las Tareas del Sprint.**

**Actividad # 9.2.4.2: Planificar Respuesta de los Riesgos de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión de Riesgos.
- ✓ Registro de Riesgos.

**Artefactos de salida:**

- ✓ Registro de Riesgos (Actualizaciones).
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Riesgos.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.4.3: Planificar Respuesta de los Riesgos de las Tareas del Sprint.**

**Actividad # 9.2.4.3: Dar Respuesta a los Riesgos de las Tareas del Sprint.**

**Roles:**

- ✓ Dueño del Equipo
- ✓ Equipo

**Artefactos de entrada:**

**Artefactos de salida:**

- ✓ Plan de Gestión de Riesgos.
- ✓ Registro de Riesgos.
- ✓ Registro de Riesgos (Actualizaciones).
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de Gestión de Riesgos.
- ✓ Cambios Solicitados.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.4.4: Dar Respuesta a los Riesgos de las Tareas del Sprint.**

**Actividad # 9.2.5: Reunión de Revisión del Sprint.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Sprint terminado.
- ✓ Feedback (demo) del producto.
- ✓ Plan de Gestión de Proyecto actualizado.

**Artefactos de salida:**

- ✓ Entregables.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.5: Reunión de Revisión del Primer Sprint.**

**Actividad # 9.2.6: Reunión Retrospectiva del Sprint.**

**Roles:**

- ✓ Scrum Máster
- ✓ Equipo
- ✓ Dueño del Producto

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.6: Reunión Retrospectiva del Primer Sprint.**

**Actividad # 9.3: Integrar el Sprint.**

**Roles:**

- ✓ Equipo

**Artefactos de entrada:**

- ✓ Entregables.

**Artefactos de salida:**

- ✓ Demo del Sprint.

**Descripción:**

Se integran cada uno de los Sprint que fueron desarrollados. Estos son un producto terminado, pero a su vez son también parte de un producto final que engrana todos los Sprint integrados.

**Herramientas y Técnicas:**

- ✓ La integración se lleva a cabo a través de la herramienta que se definió previamente en el negocio.

**Subproceso # 9.4: Reunión de Revisión del Sprint Integrado.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.5: Reunión de Revisión del primer Sprint.**

Aunque es necesario destacar que en esta reunión específicamente se revisan los Sprint integrados.

**Actividad # 9.4.1: Controlar Alcance del Sprint Integrado.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Product Backlog.
- ✓ Plan de Gestión del Alcance del Proyecto.

**Artefactos de salida:**

- ✓ Product Backlog actualizado.
- ✓ Plan de Gestión de Proyecto actualizado en el acápite Plan de Gestión del Alcance específicamente en la Línea Base del Alcance.
- ✓ Cambios Solicitados.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.3.2.2: Controlar Alcance de las Tareas del Sprint.**

Aunque es necesario destacar que en esta actividad se controla el alcance del Sprint integrado.

**Actividad # 9.4.2: Controlar Coste del Sprint Integrado.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Línea Base de Coste.
- ✓ Requisitos para la financiación del proyecto.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de salida:**

- ✓ Estimaciones de Costes (Actualizaciones).
- ✓ Mediciones del Rendimiento.
- ✓ Cambios Solicitados.
- ✓ Plan de Gestión de Proyecto actualizado específicamente en el acápite Plan de gestión de Costes.

**Descripción:**

Para documentarse sobre la actividad favor de dirigirse a: **Actividad # 5.4.1: Controlar Coste de las Tareas del Sprint.**

Aunque es necesario destacar que en esta actividad se controla el coste del Sprint integrado.

**Actividad # 9.4.3: Mostrar Requerimientos Completados del Sprint Integrado.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster

**Artefactos de entrada:**

- ✓ Sprint terminado.
- ✓ Feedback (demo) del producto.
- ✓ Plan de Gestión de Proyecto.

**Artefactos de salida:**

- ✓ Entregables.

**Descripción:**

El Equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto. Es la oportunidad de que el Dueño de Producto vea lo que está pasando con el producto y con el Equipo y la oportunidad del Equipo de saber cómo va el Dueño de Producto y el mercado.

#### **2.1.4 Fase: Revisión y Cierre.**

##### **Actividad # 10: Integrar el Proyecto.**

###### **Roles:**

- ✓ Equipo

###### **Artefactos de entrada:**

- ✓ Demo del Sprint.

###### **Artefactos de salida:**

- ✓ Producto Final.

###### **Descripción:**

Se lleva a cabo la integración de todos los demos realizados en cada una de las líneas de trabajo para obtener el producto general, y se realiza al finalizar todos los Sprint que conformarán dicho producto. Además, se llevan a cabo las pruebas necesarias para verificar la calidad del producto final, dígame prueba del sistema y pruebas de aceptación.

###### **Herramientas y Técnicas:**

- ✓ La integración se lleva a cabo a través de la herramienta que se definió previamente en el negocio.

##### **Actividad # 11: Estabilización y Despliegue del Proyecto**

###### **Roles:**

- ✓ Dueño del Producto
- ✓ Equipo
- ✓ Scrum Máster
- ✓ Cliente

###### **Artefactos de entrada:**

- ✓ Producto Final.

###### **Artefactos de salida:**

- ✓ Producto Final Desplegado.

###### **Descripción:**

El despliegue se lleva a cabo en dependencia de las especificaciones que se hayan determinado durante el desarrollo del proyecto, teniendo en cuenta el aseguramiento de la logística en cuanto a servidores, hardware y software en general.

**Actividad # 12: Cerrar Proyecto.**

**Roles:**

- ✓ Dueño del Producto
- ✓ Scrum Máster
- ✓ Equipo

**Artefactos de entrada:**

- ✓ Plan de Gestión de Proyecto.

**Artefactos de salida:**

- ✓ Archivos del Proyecto.
- ✓ Documentos de Cierre del Proyecto.

**Descripción:**

Se realiza el cierre del Plan de Gestión de Proyecto. Esta actividad incluye finalizar todas las actividades completadas a lo largo de todos los Grupos de Procesos de Dirección de Proyectos para cerrar formalmente el proyecto o una fase del proyecto y transferir el proyecto completado o cancelado según corresponda. Se desarrollan dos procedimientos para establecer las interacciones necesarias para realizar las actividades de cierre a lo largo de todo el proyecto o de una fase del proyecto:

- **Procedimiento de Cierre Administrativo.**
- **Procedimiento de Cierre del Contrato.**

**Herramientas y Técnicas:**

- ✓ Metodología de Dirección de Proyectos.
- ✓ Sistema de Información de la Gestión de Proyectos.
- ✓ Juicio de Expertos.

**2.1.5 Expediente documental**

**Sistema de Trabajo:** dirigirse a Compendio 1.

**2.2 Nivel 2: Método de desarrollo de software.**

**Introducción al nivel 2: Modelos de Procesos Industriales.**

Los modelos de desarrollo de software son la guía a seguir cuando se trata de crear un producto de software.

Existen **diferentes modelos de desarrollo de software** entre los más comunes se encuentran:

- ✓ Modelo en cascada: consiste en la realización continua del proceso de desarrollo de software.

- ✓ Desarrollo por prototipos: propicia una buena comunicación con los clientes que pueden evaluar sus funcionalidades del sistema en etapas tempranas del desarrollo.
- ✓ Desarrollo en espiral: la característica fundamental de este modelo es que el desarrollo de los estados se repite.
- ✓ Desarrollo con técnicas de cuarta generación (RAD): las características fundamentales son: participación de los usuarios en todas las etapas, el uso de herramientas automáticas y el uso de equipos de desarrollo pequeños.
- ✓ Desarrollo basado en componentes o paquetes: define la forma de sus interfaces y los mecanismos para interconectarlos entre ellos.
- ✓ Fábrica de software: modelo de servicios cuya forma de trabajo es una analogía de los procesos de producción industriales, permitiendo una reducción de los niveles de incertidumbre en el desarrollo y mantención de proyectos de software. Ofrece beneficios directos sobre la confiabilidad y nivel de satisfacción de los productos entregados, proporcionando una mayor claridad presupuestaria y un calendario acotado de los proyectos e incidencias asociadas a las aplicaciones de software de la empresa.
- ✓ Líneas de producción: Según el Instituto de Ingeniería de Software (SEI) una línea de producción de software se define de la siguiente manera: “Una LPS es un conjunto de sistemas de software compartiendo características comunes y administradas que satisface las necesidades específicas de un segmento de mercado particular o misión y que son desarrolladas de forma prescrita a partir de un conjunto común de elementos clave”.

Debido a que en la Universidad de las Ciencias Informáticas se ha comenzado a percibir un desarrollo de los modelos industriales específicamente fábricas de software y líneas de producción se ha determinado escoger como modelo de desarrollo industrial la integración de ambos. La elección se basa principalmente en que la integración de los modelos se adecua a los entornos productivos descritos en la investigación, cubriendo algunas necesidades del desarrollo de software en cuestión. A través de los mismos se obtiene la estructura del proyecto, identificando áreas, procesos y macro procesos, permitiendo además, obtener un producto general partiendo de la integración de cada producto específico identificado por cada área o proceso a automatizar. Se citan a continuación los distintos modelos que proponen las fábricas de software, asimismo se describen una serie de pasos que conforman el modelo que integra líneas de producción con fábricas de software.

**Los modelos de producción que proponen las fábricas de software son:**

1. Modelo basado en la norma ISO 9001 y CMM. (División de los elementos en cinco entidades: Técnicas, Proceso, Trabajadores involucrados, Gestión de la factoría y Activos del proceso, Herramientas y componentes de código).
2. Modelo Eureka. (Desarrollo distribuido. Constituido por: procesos, reglas, herramientas e información, personas y computadoras).
3. Modelo Clasificadorio. (Propone 5 agrupaciones: Factoría de Proyecto Ampliada (Construcción de la arquitectura), Factoría de Proyectos de Software (Ciclo de vida), Factoría de Proyectos Físicos (Diseño, implementación y prueba), Factoría de Programas (Menor de las unidades de componentes de código)).
4. Modelo propuesto por Basili. (Desarrollo basado en componentes y que los mismos pueden ser desarrollados por una factoría de componentes y otro tipo de factoría de integración o desarrollo de software).
5. Modelo Replicable (Modelo más maduro y más referenciado, característica fundamental propone 5 áreas de proceso: Área de producción de análisis de sistema o modelado de negocio, Área de producción de diseño de software, Área de construcción de software, Área de producción de componentes de infraestructura o activos del proceso, Área de producción de componentes de código).

**Figura 3: Modelo de línea de producción integrado a fábricas de software.**

### **2.3 Nivel 3: Plataforma tecnológica.**

#### **Selección de la herramienta para plantear la plataforma tecnológica.**

Para determinar la plataforma tecnológica se tuvo en cuenta un estudio acerca de las herramientas que más se adecuan a la propuesta presentada. El análisis se basó en determinar qué herramientas cubrirían parcial o totalmente las necesidades del modelo, y se llegó a la conclusión de que IceScrum es la herramienta que mejor cubre los requisitos del mismo. Para documentarse sobre la comparación dirigirse a materiales complementarios:

#### **Tabla 4: Comparación entre las herramientas para la plataforma tecnológica.**

**IceScrum:** Tecnología: Java 1.5 con el motor de servlets. Viene equipada con HSQLDB, pero puede ser configurado para trabajar con otros DBM como MySQL. Licencia: GPL. Los productos son el nivel más alto a construir, y cada implementación puede tener varios productos. Cada producto tiene un Backlog único y una hoja de ruta. Un Backlog contiene historias de usuarios, defectos y las historias técnicas. Un plan de trabajo contiene varias versiones, cada una de ellas tiene un plan de liberación individual. Un plan de

lanzamiento consiste en múltiples Sprint. Cada Sprint contiene historias, que a su vez contienen las tareas y pruebas de aceptación. Los impedimentos pueden ser rastreados para cada producto. IceScrum incluye una tabla de tareas y vistas de la iteración en una pizarra, permitiendo que las tareas se puedan arrastrar y soltar. También permite arrastrar y soltar el ranking de las historias de usuario en el backlog. IceScrum es el único producto revisado que tiene una función de la planificación de póquer. Esta característica permite a un equipo distribuido jugar póquer en la planificación para la estimación de las características e historias de los usuarios. Los usuarios pueden tener cualquiera de los roles de SCRUM. (12)

### **Herramientas a utilizar durante la aplicación del modelo.**

**Bizagi:** es un modelador de procesos que permite representar de forma esquemática todas las actividades y decisiones que se toman en el negocio. Cumple con el estándar Business Process Modeling Notation (BPMN). Reduce el tiempo de salida al mercado de nuevas ideas y estrategias de negocio y facilita un mejoramiento continuo de procesos. La herramienta se utilizó para modelar el marco de procesos propuesto. (13)

**Subversion:** es un sistema centralizado de control de versiones que permite realizar el seguimiento de los cambios en archivos empleados en proyectos de software. Hace posible recuperar antiguas versiones de los archivos, examinar el cambio en la historia de los mismos y puede ser accedido a través de la red, para ser usado por diferentes personas. Trabaja con el modelo copiar-modificar-fusionar, en el cual en cada cliente se crea una copia de los archivos y directorios. Así se puede trabajar en paralelo, modificando sus copias privadas. Finalmente, estas copias son fusionadas en una versión final. La herramienta puede ser utilizada para el control de las versiones de los entregables propuestos en el marco de desarrollo. (14)

### **Conclusiones Parciales.**

En el primer nivel de este capítulo, se propuso el modelo de procesos de gestión de proyectos centrados en SCRUM. Del mismo se describieron los procesos, subprocesos y actividades que lo conforman por fases, se especificaron roles, artefactos y herramientas y técnicas en general, además, se propuso un plan de trabajo tipo que contiene todas las reuniones del modelo general. En el segundo nivel se dan a conocer los modelos de procesos industriales y se justifica el porqué se escogió Líneas de Producción unido a Fábricas de Software. Se da a conocer además, el modelo del método de desarrollo de software por fases integrado al ADD o método de arquitectura dirigida por diseño. En el último nivel se da a conocer la herramienta para plantear la plataforma tecnológica, así como manuales de usuario, guías de administración y de instalación.

## **Capítulo 3: EVALUACIÓN DE LA SOLUCIÓN**

### **Introducción**

Desde el comienzo de una investigación, los estudiosos se hacen una pregunta: ¿Cómo dar validez al resultado de su indagación?

Con el objetivo de demostrar la confianza con que la propuesta puede ser estudiada o utilizada por otras personas, se necesita manifestar de alguna manera que la solución dada, puede funcionar con total éxito. Como procedimiento para llevar a cabo una exposición de eficacia surge el Método de Experto o Delphi el cual se utilizó para la aceptación del modelo que se presenta en el Capítulo 2, con la misión de recoger información de un grupo de personas de las que se supone sean acreedoras de un conocimiento elevado de la materia a tratar.

El método consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas al tema principal. Las estimaciones de los expertos se realizan en sucesivas rondas anónimas y con la máxima autonomía por parte de los participantes. El método Delphi procede por medio de la interrogación a expertos con la ayuda de cuestionarios sucesivos, a fin de poner de manifiesto, convergencias de opiniones y deducir eventuales consensos (15).

El Método Delphi consta de algunas características que se citan a continuación:

- Anonimato: durante el procedimiento del método ningún experto conoce la identidad de los otros que componen el tribunal, esto a su vez impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de los otros miembros, permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen y por último el experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no será conocida.
- Iteración y realimentación controlada: la iteración se consigue al presentar varias veces el mismo cuestionario. Además, se van presentando los resultados obtenidos con los cuestionarios anteriores, facilitando que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión en caso de que los argumentos presentados le parezcan más apropiados que los suyos.
- Respuesta del grupo en forma estadística: la información que se presenta a los expertos no es solo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo con lo que se ha obtenido. (16)

### **3.1 Proceso de selección de Expertos.**

El término “Experto” se aplica a la persona que tiene mucha experiencia o es muy hábil en una actividad (17). Experto es una persona en sí o un grupo de ellas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia (18).

Partiendo de las definiciones anteriores se deduce que el experto debe tener grandes cualidades a causa de su potencial, su práctica y la capacidad de integración de los conocimientos adquiridos con el fin de expresar juicios que se estiman pertinentes.

Desde el punto de vista de la calidad de la solución presentada, se define que los expertos deben tener las siguientes competencias: graduado de Nivel Superior, vinculación al desarrollo de productos informáticos, conocimientos acerca de la Gestión de Proyecto y conocimientos acerca de las Metodologías de Desarrollo de Software Ágil.

Determinar el número de expertos es fundamental para llevar a cabo la realización del mismo, se sugiere que el rango se encuentre entre 7 y 30 expertos. Para la validación de esta investigación se escogieron 12 posibles expertos donde todos estuvieron de acuerdo con formar parte del panel. Con el objetivo de obtener el coeficiente de competencia de cada uno, se llevó a cabo la realización de la encuesta # 1 (Consultar materiales complementarios **Encuesta #1**)

### **3.2 Cálculo del Coeficiente de Competencia.**

El coeficiente de Competencia se determina a partir de la encuesta anterior, para determinar el valor de la competencia de cada experto, mediante la fórmula:  $K = \frac{1}{2} (kc + ka)$ .

Donde **kc** es el coeficiente de conocimiento, obtenido de la primera pregunta de la encuesta y **ka** es el coeficiente de argumentación, este se obtiene de la tabla que recoge la autoevaluación del posible experto. Para calcular el coeficiente de argumentación se le asignan valores a las categorías seleccionadas por los posibles expertos según la siguiente escala: (Consultar materiales complementarios **Tabla 5: Grados de influencia en la determinación del coeficiente de argumentación.**) Con estos elementos es suficiente para obtener el coeficiente de competencia K. A partir del resultado de la fórmula para obtener K se puede decir que la interpretación de tales coeficientes de competencia es el siguiente:

- Si  $0.8 < k < 1.0$ , el coeficiente de competencia es alto
- Si  $0.5 < k < 0.8$ , el coeficiente de competencia es medio.
- Si  $k < 0.5$  el coeficiente de competencia es bajo.

Para validar la presente investigación se seleccionaron 7 expertos, de ellos presentaron un coeficiente de competencia alto 4 expertos y medio 3 expertos. Estos resultados se muestran en la siguiente tabla: (Consultar materiales complementarios **Tabla 6: Coeficiente de competencia de los expertos.**)

### **3.3 Elaboración del Cuestionario de Validación.**

Para llevar a cabo la validación de la propuesta se confecciona el Cuestionario de Validación, el cual consta de 17 preguntas que giran en torno a los aspectos principales del modelo presentado, asegurando con las mismas la validación del modelo general. El objetivo del documento es lograr la interacción entre los expertos, pero de manera de que la opinión de un experto no pueda influir sobre la opinión de otro. Las respuestas que brindan los expertos están valoradas en 5 categorías de modo que puedan dar su opinión exacta. A continuación se presenta el cuestionario realizado: (Consultar materiales complementarios: **Encuesta 2**)

### **3.4 Establecimiento de Concordancia entre los Expertos.**

Para que la propuesta tenga una mayor validez es necesario que exista un acuerdo entre los integrantes del panel de expertos, esto se comprueba mediante el cálculo del Coeficiente de Concordancia de Kendall el cual ayuda a precisar el grado de coincidencia de las valoraciones arrojadas por los expertos. Para determinar este coeficiente se construye una tabla de aspectos a evaluar contra expertos, estos datos fueron obtenidos al aplicar la encuesta para la validación del procedimiento. Luego de haber confeccionado dicha tabla, se siguen los siguientes pasos:

**Paso 1:** Determinar la suma de los valores numéricos asignados a cada aspecto a evaluar, según el criterio dado por cada experto ( $R_j$ ). (Consultar materiales complementarios **Tabla 7: Suma de los valores numéricos asignado por experto.**)

**Paso 2:** Determinar el valor medio de las  $R_j$ , dado por la sumatoria de los  $R_j$  entre  $N$ , siendo  $N$  el total de aspectos a evaluar (los aspectos serán las preguntas del cuestionario, en este caso  $N = 17$ ).  $\overline{R_j} = \frac{\sum_{j=1}^n R_j}{N}$

Sustituyendo los valores en la fórmula se obtiene el siguiente resultado:

$$\overline{R_j} = \frac{2(34)+5(28)+2(32)+2(31)+3(30)+24+29+25+26}{17} = 29.29$$

**Paso 3:** Determinación de la desviación media  $S$ , dada por la diferencia entre cada  $R_j$  y el valor de la media al cuadrado. Para calcular el valor de  $S$  se emplea la siguiente fórmula:  $S = \sum_{j=1}^n (R_j - \overline{R_j})^2$ . Al sustituir los valores en la fórmula se obtiene como resultado que  **$S = 131.46$**

**Paso 4:** Determinación del cuadrado del número total de expertos,  $K$ . En este caso  $K = 7$ , por lo que  **$K^2 = 49$** .

**Paso 5:** Determinación del cubo del número total de aspectos a evaluar, N. En este caso  $N = 17$ , por lo que  $N^3 = 4913$ .

**Paso 6:** Determinación de la diferencia  $N^3$  y N y su multiplicación por  $K^2$ . Obteniéndose como resultado: **239904**.

Una vez obtenidos todos los valores necesarios para el cálculo de W se sustituyen dichos valores en la siguiente fórmula:  $W = \frac{12 * S}{K^2(N^3 - N)} = \frac{12 * 131.46}{239904} = 0.006$

El coeficiente W siempre tiene que ser positivo y oscilar en el intervalo entre 0 y 1. Con este valor, se puede calcular Chi cuadrado real, con el propósito de verificar si existe o no concordancia entre los expertos. Chi cuadrado real se calcula de la siguiente forma:  $X^2 = K(N - 1)W$ . Sustituyendo los valores en la fórmula se obtiene el siguiente resultado:  $X^2 = 7(17 - 1)0.006 = 0.672$

El resultado de  $X^2$  obtenido se compara con el de las tablas estadísticas de Siegel, y si:  $X^2 \text{ real} < X^2(\alpha, N - 1)$  existe una concordancia entre el panel de expertos.

Para  $\alpha = 0.1$  y  $(N - 1) = 16$ , quedaría que  $0.67 < 23.5418$  por lo que se llega a la conclusión que existe concordancia entre los expertos.

### **3.5 Análisis de los Resultados.**

A continuación se recogen en la siguiente tabla los resultados desde el punto de vista estadístico, para arribar a conclusiones sobre el nivel de aceptación de los especialistas seleccionados. (Consultar materiales complementarios **Tabla 8: Frecuencias acumuladas**.)

Computados todos los datos de las encuestas, se prosigue con el siguiente paso del método, que consiste en la realización de la siguiente tabla en la que cada número en la fila se obtiene sumándole el anterior excepto en la primera y además, desaparece la última fila: (Consultar materiales complementarios **Tabla 9: Frecuencias absolutas acumuladas**.)

Luego se prosigue con el llenado de la siguiente tabla la cual se obtiene dividiendo cada elemento de la tabla por el número de expertos: (Consultar materiales complementarios **Tabla 10: Frecuencias relativas acumuladas**.)

Finalmente, se buscan las imágenes de los valores de la tabla anterior por medio de la función (Dist. Normal. Standard Inv.). Estas imágenes se representan en la siguiente tabla la cual además, posee algunos nuevos elementos como son:

- Suma de las columnas.
- Suma de filas.

- Promedio de las columnas.
- Los promedios de las filas se obtienen de forma similar, en este caso también se divide por cuatro porque quedan 4 categorías ya que la última se eliminó.
- Para hallar N, se divide la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.
- El valor N-P da el valor promedio que otorgan los expertos para cada indicador propuesto.

Las sumas obtenidas en las cuatro primeras columnas dan los puntos de cortes. (Consultar materiales complementarios **Tabla 11: Puntos de corte.**) Los puntos de corte se utilizan para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados. Con ellos se opera del modo siguiente: (Consultar materiales complementarios **Tabla 12: Rangos obtenidos a partir de los puntos de corte.**)

Como resultado final se obtiene el grado de adecuación de los elementos sometidos a la valoración de los expertos: (Consultar materiales complementarios **Tabla 13: Categorización de aspectos.**)

### **Conclusiones Parciales**

En este capítulo se propuso el Método Delphi para la validación del modelo presentado, características principales del mismo, ventajas y desventajas. Se escogieron 12 posibles expertos donde se tuvieron en cuenta las competencias que estos debían cumplir y de acuerdo a los resultados arrojados quedaron un total de 7 expertos seleccionados. En primer orden se les aplicó la Encuesta de Autovaloración para obtener el grado de competencia de estos y a continuación se les realizó la Encuesta de Valoración de la propuesta, la cual poseía las afirmaciones necesarias para validar el modelo. Después de un análisis estadístico de los resultados obtenidos en la encuesta fueron definidos por los expertos como Muy Adecuado y Bastante Adecuado todos los elementos.

### **CONCLUSIONES**

Como resultado de la investigación se han definido los conceptos relacionados con la gestión de proyecto y las metodologías de desarrollo de software, de igual manera se realizaron dos análisis comparativos, uno entre las metodologías livianas y robustas y otro entre las ágiles entre sí, con el fin de justificar las metodologías escogidas. Este análisis proporcionó una noción clara de cómo pueden aportar cada uno de estos enfoques a la construcción de la propuesta del modelo.

Partiendo de este estudio se describieron las actividades fundamentales propuestas que componen el marco de proceso de desarrollo de software, de igual manera se definieron cada uno de sus roles, especificando las competencias que deben reunir para poder acometer cada tarea, conjuntamente se detallaron los artefactos de entrada y salida y las herramientas y técnicas que resultan necesarios para la aplicación de este modelo. Se propuso además, la integración del Método ADD (Arquitectura Dirigida por Diseño) en aras de resolver la ausencia de escenarios arquitectónicos que por defecto traen consigo las metodologías ágiles.

Seguidamente se realizó la validación del modelo a través del método Delphi dentro del contexto de los métodos de expertos, para esto se definieron las competencias que debían cumplir los expertos, quedando 7 a los que se les aplicaron las encuestas. Después de un análisis estadístico de los resultados obtenidos en la encuesta fueron evaluados como Muy Adecuado y Bastante Adecuado todos los elementos de la propuesta. Como parte del proceso de validación se está aplicando el modelo en el centro SICS el cual ha arrojado un buen nivel de aceptación por parte de los clientes, por lo que se concluye que fueron cumplidos todos los objetivos específicos planteados en la investigación.

## **RECOMENDACIONES**

- 1 . Aplicar el modelo en proyectos para valorar su desempeño y ajuste a las necesidades de los proyectos orientados a entornos productivos pequeños especializados en soluciones de gestión.
2. Mejorar la usabilidad del modelo con en una herramienta que permita la gestión de todas las tareas con todos los artefactos, técnicas y herramientas asociadas.

**Referencia Bibliográfica**

1. **Riola, Jose Carlos Carvajal.** *Metodologías ágiles: Herramientas y modelos de desarrollo para aplicaciones Java EE como metodología empresarial.* 2008.
2. [www.scribd.com](http://www.scribd.com). *www.scribd.com*. [En línea] [Citado el: 20 de Enero de 2010.] <http://www.scribd.com/doc/8255409/Metodologias-para-la-geston-y-desarrollo-de-Software>.
3. [www.knoow.net](http://www.knoow.net). *www.knoow.net*. [En línea] [Citado el: 15 de enero de 2010.] <http://www.knoow.net/es/cieeconcom/gestion/gestiondeproyectos.htm>.
4. **Project Management Institute, Inc.** *Guía de los Fundamentos de la Dirección de Proyectos Tercera Edición (Guía del PMBOK®).* Pennsylvania 19073-3299 EE.UU.: Four Campus Boulevard Newtown Square.
5. **Pressman, Roger.** *Ingeniería del Software: Un Enfoque Practico.* s.l. : McGraw-Hill., 2006.
6. *Conferencia 1 Ingeniería de Software.*
7. [scrumenespanol](http://scrumenespanol.blogspot.com). *scrumenespanol*. [En línea] <http://scrumenespanol.blogspot.com/2007/12/scrum-en-pocas-palabras.html>.
8. **Deemer, Pete, y otros.** *Información Básica de SCRUM (The SCRUM Primer).* 2009.
9. **Mena López, Grette Leydi y Escalona Herrera, Arianna.** *Expediente para la documentación técnica de la arquitectura de software de sistemas de gestión.* 2009.
10. **Rodrigo Corral.** *Blog de Rodrigo Corral con Scrum.* [En línea] 2007. <http://geeks.ms/blogs/rcorral/archive/2007/01/29/tengo-un-plan-ser-gil.aspx>.
11. **Beck, Kent.** *Manifiesto for Agile Software Development.* [En línea] 3 de 2001. [Citado el: 3 de 3 de 2010.] <http://www.agilemanifiesto.org>.
12. [www.icescrum.org](http://www.icescrum.org). *www.icescrum.org*. [En línea] <http://www.icescrum.org/>.
13. [www.scribd.com](http://www.scribd.com). *www.scribd.com*. [En línea] [Citado el: 3 de marzo de 2010.] <http://www.scribd.com/doc/22659190/Manual-De-BizAgi>.

14. [www.antartec.com](http://www.antartec.com). [www.antartec.com](http://www.antartec.com). [En línea]  
<http://www.antartec.com/opensource/2008/11/antartec-es-open-source-parte-5-subversion-svn/>.
15. **Barcelona, Ariel y Godet, Michel.** *El método Delphi. Una Técnica de previsión para la incertidumbre.* Paris : Manuel de Prospective Strategique. Dunod, 1996.
16. **Ferrer, Raymundo Llerena y Rey, Pedro Martinez.** *Propuesta de modelo para desarrollar el modelado del negocio en proyectos BPM/ SOA.* Ciudad de la Habana : s.n., 2009.
17. [En línea] [Citado el: 28 de abril de 2010.] <http://es.thefreedictionary.com/experto>.
18. **GONZÁLEZ ALMAGUER, ARMÍN.** *El Método Delphi y el procesamiento estadístico de los datos obtenidos de la consulta a los expertos.* s.l. : ISP "José de la Luz y Caballero".

**Bibliografía**

**Bachmann, Felix y Bass, Len.** *Introduction to the Attribute Driven Design Method.* s.l. : Proceedings of the 23rd International Conference on Software Engineering (ICSE'01), 2001.

**Barcelona, Ariel y Godet, Michel.** *El método Delphi. Una Técnica de previsión para la incertidumbre.* Paris : Manuel de Prospective Strategique. Dunod, 1996.

**Beck, K.** *Extreme Programming Explained. Embrace Change.* s.l. : Pearson Education, 1999.

**Beck, Kent.** *Manifiesto for Agile Software Development.* [En línea] 3 de 2001. [Citado el: 3 de 3 de 2010.] <http://www.agilemanifiesto.org>.

**Cockbun, A.** *Agile Software Development.* s.l. : Addison-Wesley, 2001.

*Conferencia 1 Ingeniería de Software.*

**Deemer, Pete, y otros.** *Información Básica de SCRUM (The SCRUM Primer).* 2009.

**Ferrer, Raymundo Llerena y Rey, Pedro Martinez.** *Propuesta de modelo para desarrollar el modelado del negocio en proyectos BPM/ SOA.* Ciudad de la Habana : s.n., 2009.

**García Borrego, Félix, y otros.** *Manual de usuario gestor encuesta Opina.* 1.02RC2.

**GONZÁLEZ ALMAGUER, ARMÍN.** *El Método Delphi y el procesamiento estadístico de los datos obtenidos de la consulta a los expertos.* s.l. : ISP "José de la Luz y Caballero".

**G. Figueroa, Roberth, J. Solís, Camilo y A. Cabrera, Armando.** *Metodologías tradicionales vs metodologías ágiles.*

**Highsmith, J y Orr, K.** *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems.* s.l. : Dorset House, 2000.

**Mena López, Grette Leydi y Escalona Herrera, Arianna.** *Expediente para la documentación técnica de la arquitectura de software de sistemas de gestión.*

**Muro Fumero, Dorisbel y Lazo Ochoa, Rene.** *Integración del modelo de procesos de desarrollo con el modelo de procesos de gestión de proyecto.* 2010.

**Palacio, Juan.** *Flexibilidad con Scrum.* 2007.

**Pressman, Roger.** *Ingeniería del Software: Un Enfoque Practico.* s.l. : McGraw-Hill., 2006.

**Project Management Institute, Inc.** *Guía de los Fundamentos de la Dirección de Proyectos Tercera Edición (Guía del PMBOK®).* Pennsylvania 19073-3299 EE.UU. : Four Campus Boulevard Newtown Square.

**Riola, Jose Carlos Carvajal.** *Metodologías ágiles: Herramientas y modelos de desarrollo para aplicaciones Java EE como metodología empresarial.* 2008.

**Rodrigo Corral.** *Blog de Rodrigo Corral con Scrum.* [En línea] 2007.  
<http://geeks.ms/blogs/rcorral/archive/2007/01/29/tengo-un-plan-ser-gil.aspx>.

**Schwaber, K, Beedle, M y Martin, R.C.** *Agile Software Development with SCRUM.* s.l. : Prentice Hall., 2001.

*scrumenespanol. scrumenespanol.* [En línea] <http://scrumenespanol.blogspot.com/2007/12/scrum-en-pocas-palabras.html>.

**Stapleton, J.** *Dsdm Dynamic Systems Development Method: The Method in Practice.* s.l. : Addison-Wesley, 1997.

**Solis Álvarez, Camilo Javier y Figueroa Díaz, Roberth Gustavo.** *Metodologías Tradicionales vs. Metodologías Ágiles.*

*www.scribd.com. www.scribd.com.* [En línea] [Citado el: 20 de Enero de 2010.]  
<http://www.scribd.com/doc/8255409/Metodologias-para-la-geston-y-desarrollo-de-Software>.

*www.knoow.net. www.knoow.net.* [En línea] [Citado el: 15 de enero de 2010.]  
<http://www.knoow.net/es/cieeconcom/gestion/gestiondeproyectos.htm>.

*www.proyectosagiles.org. www.proyectosagiles.org.* [En línea] [Citado el: 2 de abril de 2010.]