**7.1 Modelo de Ciclo de vida**

Todo proyecto debe de tener un modelo de ciclo de vida que defina como va a ser desarrollado. Esto con el propósito de darle un orden a las actividades que se van a ejecutar y así lograr la terminación adecuada del proyecto. El ciclo de vida que el grupo Active escogió para el proyecto es una adaptación de Scrum con XP.

**7.1.1 Scrum**

Es un marco de trabajo ligero para manejo de proyectos, principalmente usado para el desarrollo de software. Describe un acercamiento iterativo e incremental para el trabajo del proyecto. (Scrum Alliance, s.f.) Los componentes principales de Scrum son:

**Roles**:

***Scrum Team***

Es una colección de individuos que trabajan juntos para entregar el producto. Para lograr ser efectivo, el equipo debe tener un objetivo común, seguir normas establecidas y mostrar respeto entre sus integrantes. El *scrum team* se trata como un conjunto, es decir, un error o un logro se atribuye al equipo, no a individuos.

Las características notables de un *scrum team* son:

* Los integrantes comparten y siguen las mismas reglas
* Todo el equipo es responsable de su entrega
* El equipo tiene poder de decisión
* Trabaja de la manera más autónoma posible
* Se auto-organiza
* Las habilidades dentro del equipo están balanceadas
* Es pequeño e indivisible
* Integrantes trabajan tiempo completo con el equipo

Por otro lado, las reglas de un *scrum team* deben contener unos puntos clave que son el tiempo y ubicación de las reuniones, la definición de terminado, como se debe programar y que herramientas se va a usar. (Scrum Alliance, s.f.)

***Scrum Master***

Es el líder que le sirve al equipo, no al contrario. Se ve como un protector del equipo, por ende, debe asegurar que todos aquellos involucrados con el proyecto y en especial el equipo de desarrollo, puedan concentrarse en su trabajo sin distracciones. Esto incluye proteger al equipo del *product owner,* o de problemas organizacionales y distracciones internas, por ejemplo, arreglar computadores o generar un ambiente más callado de trabajo.

Además de proteger al equipo, el *scrum master* debe proteger el proceso de Scrum como tal. Debe ser experto en los procesos que componen a Scrum y como se deben aplicar. Deber asegurar que el equipo se mantenga dentro de Scrum. (International Scrum Institute, s.f.)

***Scrum Product Owner:***

Es de donde emerge el valor de negocio. Es el responsable de definir que se va a hacer y de priorizar el trabajo. Debe saber que se debe entregar y porque es importante para los clientes, para el mercado y para la organización. Debe ser un guía experto capaz de llevar el equipo a completar el proyecto. Lo más importante y distinto de este rol es que el *product owner* deber involucrarse en el proyecto y permanecer activo durante su ejecución. (SCRUMstudy, 2016)

**Proceso:**

Scrum es un proceso iterativo e incremental. Para este, se requiere de ciertos artefactos y procesos propios de Scrum que son:

**Product Backlog:**

Es una lista de todas las cosas que necesitan hacerse dentro del proyecto. Reemplaza los artefactos de especificación de requerimientos tradicionales. Los ítems pueden ser tanto técnicos como centrados a usuarios. El dueño del Scrum Product Backlog es el Scrum Product Owner

Un Scrum Product Backlog se diferencia de una lista simple de actividades dado a las siguientes características:

* Toda entrada en el Scrum Product Backlog le adiciona valor al cliente.
* Las entradas están priorizadas y ordenadas de manera acorde.
* El nivel de detalle depende de la posición de la entrada dentro del Scrum Product Backlog
* Todas las entradas son estimadas.
* El Scrum Product Backlog es un documento vivo, es decir cambiante durante el desarrollo del proyecto.
* El Scrum Product Backlog no especifica acciones ni actividades de bajo nivel.

Dado a que es un documento vivo, el Scrum Product Backlog cambia a medida que el proyecto se desarrolla. Si se encuentran nuevos ítem en el desarrollo del proyecto, se deben agregar al Scrum Product Backlog.

**Scrum User Stories:**

Normalmente, los ítems dentro de un Scrum Product Backlog son descritos como historias de usuario. Esto es una historia corta acerca de alguien usando el producto. Tiene nombre, una narrativa pequeña, criterio de aceptación y condiciones para que la historia se complete. Ejemplos de plantillas para historia de usuario son (International Scrum Institute, s.f.):

Como un [actor], Yo [quiero|debo] [acción] para [logro]

Como un [actor], Yo [quiero|debo] [logro]

**Actor:** El “dueño” de la historia

**Acción:** Lo que el actor hace. Si es obligatoria debe ir precedida de “debo”. De lo contrario se debe usar “quiero”

**Logro:** Lo que el actor quiere lograr al ejecutar la acción.

**Sprint:** Es el corazón de Scrum. Es un lapso de un mes o menos, en el un incremento de producto terminado, usable y potencialmente liberable se crea. Son de tiempo constante durante el proyecto y deben empezar donde el otro termina. (Scrum.Org, s.f.)

**Sprint Planning:** El trabajo que se va a ejecutar en un Sprint, es planificado en el Sprint Planning. [guide] Debe comenzar con un WHAT-Meeting. El propósito de esta sesión es de definir un Sprint Backlog realista con los ítems que pueden ser implementados completamente hasta el final del Sprint. Después se debe hacer un HOW-Meeting que se define las tareas concretas necesarias para la implementación completa de las entradas del Scrum Product Backlog. (The Interantional Scrum Institute)

**Sprint Backlog:** Es un set de ítems del Product Backlog seleccionados para un Sprint, adicionando un plan para entrega del incremento del producto y completar el objetivo del Sprint. (Scrum.Org, s.f.)

**Scrum Burndown Chart:** es una herramienta de medición visual que muestra el trabajo hecho por día contra la tasa de trabajo proyectada para la entrega actual. Su propósito es el de habilitar que el proyecto esté en el camino correcto para hacer la entrega de la solución esperada dentro del tiempo deseado.

**Definition of Done (DOD, por sus siglas en inglés):** En el momento que un item del Scrum Product Backlog se describe como “terminado”, todos deben saber que significa “terminado”. Por ende, se debe definir que significa “terminado” para el Scrum Team. La definición de terminado debe ir expandiéndose a medida que el equipo madure para tener un estándar. (Scrum.Org, s.f.)

**Daily Scrum Meeting / Daily Standup Meeting:** es una reunión diaria que se caracteriza por ser cortae idealmente al principio de cada día de trabajo. Cada miembro que trabaje para completar el sprint debe participar. En la reunión, todos deben responder las siguientes preguntas:

* ¿Qué has logrado desde la reunión anterior?
* ¿Qué vas a lograr para la siguiente reunión?
* ¿Qué te impide que logres tus tareas?

Todos los miembros del equipo deben atender a la reunión y se recomienda que estén de pie. Adicionalmente se recomienda que la reunión no dure más de 15 minutos. La falta de tiempo hace que no se pueda ignorar cualquier inconveniente que los integrantes reporten y el Scrum Master debe anotarlos.

**Sprint Review Meeting:** es una reunión que se lleva a cabo al final de cada sprint. En esta, se hace seguimiento de que ítems se completaron del Product Backlog durante el sprint. Se puede hacer mediante demostraciones de funcionalidad. Aquello que no esté completo no se debe demostrar. Esta reunión debe ser informal y rápida.

**Sprint Retrospective Meeting:** esta reunion se debe hacer después de la *Sprint Review Meeting* y se reúnen el Scrum Master y el Scrum Team. Esta reunion debe tener un tiempo definido. El propósito es responder tres preguntas acerca del sprint pasado:

* ¿En que nos fue bien durante el sprint?
* ¿En que nos fue mal durante el sprint?
* ¿Cómo podemos mejorar para el siguiente sprint?

Scrum agrega valor al proyecto, en el aspecto gerencial, por medio de todos los artefactos y procesos mencionados anteriormente. En especial da claridad a la cantidad de trabajo hecho y por hacerse y además propone roles que ayudan a la simplicidad estructural del grupo. Es por lo anterior que el grupo Active lo escogió.

**7.1.2 Extreme Programming (XP)**

*Extreme programming* es un marco de trabajo ágil para desarrollo de software que busca producir software de mayor calidad, y mayor calidad de vida para el equipo de desarrollo. XP es el más específico de los marcos de trabajo agiles en cuanto a prácticas apropiadas para desarrollo de software (Agile Alliance, s.f.)

Este marco de trabajo se caracteriza por cinco valores con los cuales se definen un set de reglas para cada parte del desarrollo de software. Según la definición del autor Don Wells, estos son:

**Valores:**

* **Simplicidad:** Se hará lo que es necesario y lo que se pide, pero no más. Esto maximiza el valor creado por la inversión hecha. Se tomarán simples y pequeños pasos para lograr nuestro objetivo y mitigar las fallas a medida que aparezcan. Crearemos algo que nos haga orgullosos y lo mantendremos a largo plazo por costos razonables.
* **Comunicación:** Todos son parte de un mismo equipo y nos comunicamos a de cara a cara a diario. Trabajaremos juntos en todo desde requerimientos hasta código. Crearemos la mejor solución juntos.
* **Retroalimentación:** Tomaremos cada compromiso de iteración seriamente por medio de la entrega de software funcional. Demostraremos el software de manera temprana y a menudo y escucharemos atentamente y haremos los cambios necesarios. Hablaremos sobre el proyecto y adaptaremos nuestros procesos a él, no al revés.
* **Respeto:** Todos dan y sienten el respeto que merecen como un integrante valorado del equipo. Todos contribuyen valor, aunque sea solo con entusiasmo. Los desarrolladores respetan la experiencia del cliente y viceversa. Gerencia respeta nuestro derecho de aceptar responsabilidad y de tener autoridad sobre nuestro trabajo.
* **Coraje:** Diremos la verdad acerca nuestro progreso y estimaciones. No documentamos excusas para el fracaso porque planeamos tener éxito. Nosotros no le tememos a nada porque nadie nunca trabaja solo. Nos adaptaremos a los cambios cuando sucedan.

**Reglas:**

Las siguientes reglas son el producto natural de la maximización de los valores mencionados previamente. Es por esto que Don Wells menciona explícitamente que XP no es un set de reglas si no una manera de trabajar en harmonía con valores personales y corporativos y sugiere agregar reglas propias que reflejen estos valores.

**Reglas de Planificación**

* Historias de usuario son descritas.
* La planificación de entregas crea el horario de entregas.
* Has entregas pequeñas y frecuentes.
* El proyecto se divide en iteraciones.

**Reglas de Gerencia**

* Dale al equipo un espacio abierto de trabajo dedicado.
* Ten un ritmo sostenible.
* Una reunión de pie al comienzo de cada día.
* La velocidad de proyecto es medida. Esto es simplemente cuanto trabajo se está haciendo en tu proyecto.
* Mueve personas
* Arregla el XP cuando se rompa. Esto significa que XP se debe adaptar en caso de que no funcione. Se sugiere empezar con las reglas básicas y modificarlas para ir acorde al equipo.

**Reglas de Diseño**

* Simplicidad.
* Escoge una metáfora del sistema.
* Usa tarjetas CRC (Clases, Responsabilidades y Colaboración) para decisiones de diseño.
* Crea soluciones de pico (“*spike solutions*”) para reducir el riesgo. Esto es un programa simple que explore soluciones potenciales a un solo problema.
* Ninguna funcionalidad se agrega tempranamente**.**
* Refactoriza cuando y donde sea posible.

**Reglas de Programación**

* El cliente siempre está disponible.
* El código debe ser escrito bajo estándares acordados.
* Programa las pruebas unitarias primero**.**
* Toda la producción de código se hace en parejas.
* Solo un par integra código a la vez.
* Integra código frecuentemente**.**
* Prepara una computadora dedicada solo para integración.
* Haz uso de propiedad colectiva.

**Reglas de Pruebas**

* Todo el código debe tener pruebas unitarias.
* Todo el código debe pasar todas las pruebas unitarias antes de entregarse.
* Cuando un *bug* es encontrado, pruebas son creadas.
* Pruebas de aceptación se corren frecuentemente y el resultado es publicado.

*Extreme Programming* ofrece técnicas más detalladas de trabajo, orientadas a la sana convivencia y motivación del grupo para cumplir con el objetivo de la manera más positiva posible. Adicionalmente, ofrece flexibilidad y propone una metodología para el desarrollo de software por pares lo cual puede otorgar un valor agregado pertinente al proceso de producción. Es por esto por lo que se escogió para ser usado en el proyecto.

**7.1.3 Plan Utilizado**

Para la llevar a cabo el proyecto, el grupo Active decidió usar la metodología ágil de Scrum en conjunto con XP. La razón principal para esta decisión es la flexibilidad y simplicidad relativa en comparación a los demás modelos de ciclo de vida. Adicionalmente, no se aplicará en su totalidad ambos marcos de trabajo, si no, se adaptará para mejor coincidir con el tamaño y alcance del proyecto y con los recursos disponibles. Se tomarán los aspectos gerenciales de Scrum para complementar con las técnicas de desarrollo que ofrece XP.

El modelo de ciclo de vida usado tomará ciertos elementos de Scrum y otros de XP para poder llevar a cabo el proyecto. Dentro de los artefactos de Scrum, se tomó en cuenta el Product Backlog. Este se usará para hacer seguimiento del trabajo efectuado y el trabajo por hacer. Por otro lado, se hará uso de los sprints de scrum. Estos nos permiten organizar el trabajo y dividir las tareas entre diferentes sprints para lograr llevarlas a cabo. Se hará uso de la definición y funcionalidad del scrum master para gerenciar a los integrantes del grupo. Adicionalmente, se hará uso de la idea de entregas frecuentes y funcionales, diseño simple, pruebas frecuentes y la programación por pares descrita en XP para poder llevar a cabo el desarrollo del producto.

El grupo se dividió en varios grupos funcionales que trabajan como scrum teams con los mismos scrum masters. Cada grupo funcional representa un grupo de actividades que se deben llevar a cabo durante el proyecto. A su vez cada grupo funcional es dividido por roles. Los roles son responsables de actividades específicas dentro de las actividades de cada grupo. Sin embargo, los grupos funcionales se ven como un solo *lane* en el *Anexo XX: Diagrama BPMN de Modelo de Ciclo de vida* y se tratan como un solo scrum team*.*

El comienzo del proyecto se da con las especificaciones dadas por la Scrum Product Owner, que en este caso será la Ingeniera Anabel Montero Posada. Es ella quien le suministró al grupo Active los requisitos para la aprobación del producto y las especificaciones para el proyecto. Después de esto, se definió el set de roles y responsabilidades necesarias para llevar a cabo el proyecto. Estos roles se definieron dentro de grupos funcionales que describen un conjunto de roles que llevan a cabo procesos a fines para llevar a cabo el proyecto. Cada rol y grupo funcional son descritos en la sección *7.4.2 Organigrama y descripción de roles.* Adicionalmente, se definió los procesos necesarios para la ejecución del proyecto. Esto incluye: análisis de riesgos, selección de modelo de ciclo de vida, selección de herramientas, definición de propósito, alcance y objetivos, definición de reglas de convivencia, etc.

El proyecto se ejecutará mediante varios sprints. Antes de empezar los sprints, se hará una elección de actividades. Cada sprint tendrá una lista con las tareas necesarias y posibles a ejecutar dentro del sprint. Durante el transcurso de los sprints, se harán Daily Scrum Meetings, donde cada integrante deberá reportar el trabajo efectuado, el trabajo que va a hacer y en caso tal de tener problemas o contratiempos, avisarlos para darles solución. Adicionalmente, se harán múltiples revisiones por el Product Owner del trabajo efectuado para así tener una retroalimentación directa para mejorar la calidad del trabajo.

# Bibliography

Agile Alliance. (n.d.). *What is Extreme Programming (XP)? | Agile Alliance*. Retrieved from Agile Alliance: https://www.agilealliance.org/glossary/xp/#q=~(filters~(postType~(~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)

International Scrum Institute. (n.d.). *Scrum Roles – The Scrum Team - International Scrum Institute*. (International Scrum Institute) Retrieved 8 16, 2017, from http://www.scrum-institute.org/Scrum\_Roles\_The\_Scrum\_Team.php

Scrum Alliance. (n.d.). *Scrum Roles Demystified - Scrum Alliance*. (Scrum Alliance) Retrieved 8 16, 2017, from https://www.scrumalliance.org/agile-resources/scrum-roles-demystified

Scrum.Org. (n.d.). *The Scrum Guide™*. Retrieved 8 21, 2017, from https://www.scrumguides.org/scrum-guide.html

SCRUMstudy. (2016). *A Guide to the Scrum Body of Knowledge (SBOK™ Guide) – 2016 edition .* VMEdu Inc.

The Interantional Scrum Institute. (n.d.). *Scrum Revealed.*

Wells, D. (2013, Octubre 8). *Extreme Programming: A Gentle Introduction*. Retrieved Agosto 20, 2017, from http://www.extremeprogramming.org/