**SPMP**

Alejandra Chacón

Alejandra Guerrero

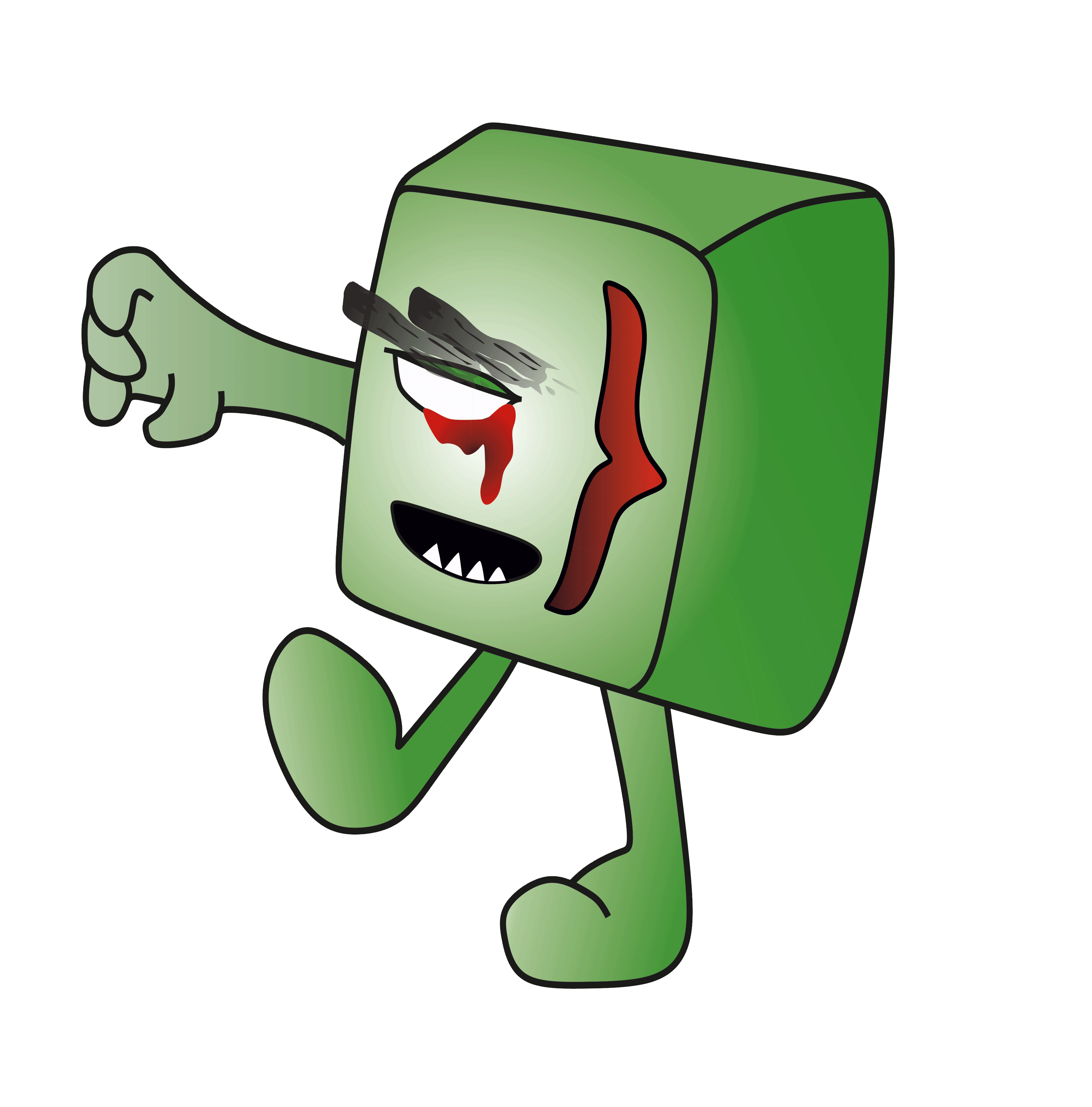
María José Mendoza

Oscar Moreno

Juan Pablo Rodríguez

Miguel Urquijo

Camilo Zamora



**The**

**Walking**

**Dev**

# HISTORIAL DE CAMBIOS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VERSIÓN | FECHA | SECCIÓN DEL DOCUMENTO | DESCRIPCIÓN DEL CAMBIO | RESPONSABLES |
| 0.1.0 | 16/02/2016 | 6.1, 6.2, 6.3, 8.4.1, 8.4.2 | Se agregaron las secciones al documento | Scrum Team |
| 0.1.0 | 16/02/2016 | 6.4, 6.5, 6.6 | Se agregaron las secciones al documento | Scrum Team |
| 0.2.0 | 22/02/2016 | 8.1, 8.1.1, 8.2, 8.2.1, 8.3 | Se agregaron las secciones al documento | Scrum Master y Scrum Team: Análisis y Documentación |
| 0.2.0 | 22/02/2016 | 9, 9.1, 9.2, 9.2.1, 9.2.2 | Se agregaron las secciones al documento | Scrum Team: Desarrollador |
| 0.2.0 | 22/02/2016 | 9.3, 9.3.1, 9.3.2, 9.3.3, 9.3.4 | Se agregaron las secciones al documento | Scrum Team: Administración de calidad y configuración |
| 0.2.1 | 1/03/2016 | 9.3.2, 9.3.3 | Se corrigieron y complementaron las secciones | Scrum Team: Administración de calidad y configuración |
| 0.2.2 | 5/03/2016 | 8.1.1, 8.2.1, 9.3.2 | Se corrigieron y complementaron las secciones | Scrum Team: Administración de calidad y configuración |
| 0.2.2 | 5/03/2016 | 6.1, 6.2, 6.3, 8.4.1, 8.4.2 | Se corrigieron y complementaron las secciones | Scrum Team: Administración de calidad y configuración |
| 0.2.3 | 6/03/2016 | 9.3.2, 9.3.3 | Se corrigieron y complementaron las secciones | Scrum Team: Análisis y Documentación |
| 0.3.0 | 2/03/2016 | 10.2, 10.3 | Se agregaron las secciones al documento | Scrum Team: Desarrollador |
| 0.3.0 | 3/03/2016 | 12.2, 12.3 | Se agregaron las secciones al documento | Scrum Master y Scrum Team: Análisis y Documentación |
| 0.3.0 | 4/03/2016 | 12.4, 12.5 | Se agregaron las secciones al documento | Scrum Team: Administración de calidad y configuración |
| 0.3.1 | 1/03/2016 | 9.1, 9.2.1 | Se complementaron las secciones con base en la experiencia obtenida al aplicarlos | Scrum Team: Desarrollador |
| 1.0.0 | 6/03/2016 | Todo | Se hizo evaluación de todas las secciones, se hicieron muchas correcciones y se asignaron tareas | Scrum Team: Análisis y Documentación |
| 1.1.0 | 8/03/2016 | Todo | Se agregaron las correcciones pendientes y se agregaron nuevas | Scrum Team: Análisis y Documentación |
| 1.2.0 | 10/03/2016 | 10.2, 10.3 | Se corrigieron y complementaron las secciones | Scrum Team |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# PREFACIO

Ian Sommerville afirma que “Una excelente planeación no garantiza el éxito en un proyecto de ingeniería de software. Sin embargo, una mala planeación lleva a que la probabilidad de fracaso sea alta.” [8]. Dada la importancia que tiene la planeación, haciendo uso del SPMP que es un documento completo y regido bajo el estándar IEEE 16326, se puede dar respuesta a lo que plantea Sommerville.

The Walking Dev, es un equipo de trabajo que gira entorno a desarrollar un producto de calidad y propiciar el éxito del proyecto, a través del presente SPMP que se compone de una gama de planes en distintas áreas y que se ejecutan a lo largo del proceso de ingeniería de software [6], busca lograr una buena administración y control de forma que se garantice la calidad del proceso y por tanto, la calidad del producto software D-EVent, un aplicativo para la gestión de eventos.

En el presente documento encontrará las actividades y detalles que componen cada plan, el modelo de desarrollo a seguir, la organización grupal establecida, las relaciones que deben haber entre los distintos procesos para que haya consistencia y uniformidad. Este SPMP es la base de trabajo de The Walking Dev, así mismo como el medio de comunicación que el equipo tiene con el cliente para realizar, definir, retroalimentar y evaluar tanto al proyecto como al equipo.

# TABLA DE CONTENIDOS

[HISTORIAL DE CAMBIOS 1](#_Toc445388557)

[PREFACIO 3](#_Toc445388558)

[TABLA DE CONTENIDOS 4](#_Toc445388559)

[LISTA DE TABLAS 7](#_Toc445388560)

[LISTA DE FIGURAS 8](#_Toc445388561)

[VISTA GENERAL DEL PROYECTO 9](#_Toc445388562)

[6.1 VISIÓN DEL PRODUCTO 9](#_Toc445388563)

[6.2 PROPÓSITO, ALCANCE Y OBJETIVOS 9](#_Toc445388564)

[PROPÓSITO 9](#_Toc445388565)

[ALCANCE 10](#_Toc445388566)

[OBJETIVOS 11](#_Toc445388567)

[6.3 SUPUESTOS Y RESTRICCIONES 11](#_Toc445388568)

[SUPUESTOS 11](#_Toc445388569)

[RESTRICCIONES 12](#_Toc445388570)

[6.4 ENTREGABLES 12](#_Toc445388571)

[DESCRIPCIÓN DE ENTREGABLES: 12](#_Toc445388572)

[6.5 RESUMEN DE CALENDARIZACIÓN Y PRESUPUESTO 14](#_Toc445388573)

[RESUMEN DE CALENDARIZACIÓN 14](#_Toc445388574)

[RESUMEN DE PRESUPUESTO 14](#_Toc445388575)

[6.6 EVOLUCIÓN DEL PLAN 15](#_Toc445388576)

[OBJETIVO 16](#_Toc445388577)

[RESPONSABLES 16](#_Toc445388578)

[DESARROLLO 16](#_Toc445388579)

[GLOSARIO 17](#_Toc445388580)

[CONTEXTO DEL PROYECTO 18](#_Toc445388581)

[8.1 MODELO DE CICLO DE VIDA 18](#_Toc445388582)

[SCRUM 18](#_Toc445388583)

[XP (Extreme Programming) 21](#_Toc445388584)

[8.1.1 ANÁLISIS DE ALTERNATIVAS Y JUSTIFICACIÓN 21](#_Toc445388585)

[8.2 LENGUAJES Y HERRAMIENTAS 29](#_Toc445388586)

[8.2.1 ANÁLISIS DE ALTERNATIVAS Y JUSTIFICACIÓN 33](#_Toc445388587)

[8.3 PLAN DE ACEPTACIÓN DEL PRODUCTO 34](#_Toc445388588)

[OBJETIVO 34](#_Toc445388589)

[RESPONSABLES 34](#_Toc445388590)

[DESARROLLO 35](#_Toc445388591)

[8.4 ORGANIZACIÓN DEL PROYECTO Y COMUNICACIÓN 36](#_Toc445388592)

[8.4.1 INTERFACES EXTERNAS 36](#_Toc445388593)

[8.4.2 ORGANIGRAMA Y DESCRIPCIÓN DE ROLES 39](#_Toc445388594)

[9. ADMINISTRACIÓN DEL PROYECTO 47](#_Toc445388595)

[9.1 MÉTODOS Y HERRAMIENTAS DE ESTIMACIÓN 47](#_Toc445388596)

[Objetivo General 47](#_Toc445388597)

[9.2 INICIO DEL PROYECTO 51](#_Toc445388598)

[9.2.1 ENTRENAMIENTO DEL PERSONAL 51](#_Toc445388599)

[9.2.2 INFRAESTRUCTURA 53](#_Toc445388600)

[9.3 PLANES DE TRABAJO DEL PROYECTO 55](#_Toc445388601)

[9.3.1 DESCOMPOSICIÓN DE ACTIVIDADES 55](#_Toc445388602)

[9.3.2 CALENDARIZACIÓN 55](#_Toc445388603)

[RESPONSABLES 56](#_Toc445388604)

[9.3.3 ASIGNACIÓN DE RECURSOS 57](#_Toc445388605)

[9.3.4 ASIGNACIÓN DE PRESUPUESTO Y JUSTIFICACIÓN 58](#_Toc445388606)

[10. MONITOREO Y CONTROL DEL PROYECTO 60](#_Toc445388607)

[10.1 ADMINISTRACIÓN DE REQUERIMIENTOS 60](#_Toc445388608)

[10.2 MONITOREO Y CONTROL DEL PROGRESO 60](#_Toc445388609)

[10.3 CIERRE DEL PROYECTO 61](#_Toc445388610)

[11. ENTREGA DEL PRODUCTO 65](#_Toc445388611)

[OBJETIVO 65](#_Toc445388612)

[RESPONSABLES 65](#_Toc445388613)

[RECURSOS REQUERIDOS 65](#_Toc445388614)

[ENTREGABLES 65](#_Toc445388615)

[12. PROCESOS DE SOPORTE 67](#_Toc445388616)

[12.1 AMBIENTE DE TRABAJO 67](#_Toc445388617)

[12.2 ANÁLISIS Y ADMINISTRACIÓN DE RIESGOS 70](#_Toc445388618)

[12.3 ADMINISTRACIÓN DE CONFIGURACIÓN Y DOCUMENTACIÓN 75](#_Toc445388619)

[OBJETIVOS 76](#_Toc445388620)

[RESPONSABLES 76](#_Toc445388621)

[HERRAMIENTAS 76](#_Toc445388622)

[Ítems de configuración: 76](#_Toc445388623)

[Versionamiento 77](#_Toc445388624)

[Proceso de control de cambios: 78](#_Toc445388625)

[Formato de Documentación 80](#_Toc445388626)

[Artefactos de Documentación 82](#_Toc445388627)

[12.4 MÉTRICAS Y PROCESO DE MEDICIÓN 82](#_Toc445388628)

[12.5 CONTROL DE CALIDAD 85](#_Toc445388629)

[BIBLIOGRAFÍA 91](#_Toc445388630)

# LISTA DE TABLAS

[Tabla 1. Duración de cada entrega del proyecto 14](#_Toc445388631)

[Tabla 2. Horas de trabajo por entrega 15](#_Toc445388632)

[Tabla 3. Presupuesto por entrega 15](#_Toc445388633)

[Tabla 4. Selección del modelo de ciclo de vida 27](#_Toc445388634)

[Tabla 5. Personas externas que interactúan con el desarrollo del proyecto 37](#_Toc445388635)

[Tabla 6. Entidades externas que interactúan con el desarrollo del proyecto 39](#_Toc445388636)

[Tabla 7 Computadores personales de los integrantes de The Walking Dev. 54](#_Toc445388637)

[Tabla 8. Probabilidades de un riesgo [44] 73](#_Toc445388638)

[Tabla 9. Impacto de un riesgo [44] 74](#_Toc445388639)

[Tabla 10. Matriz de riesgos [44] 74](#_Toc445388640)

[Tabla 11. Hoja de Información de Riesgo 75](#_Toc445388641)

[Tabla 12 Formato del código 81](#_Toc445388642)

[Tabla 13. Documentación – Administración de configuración 82](#_Toc445388643)

[Tabla 14. Calificación del software 85](#_Toc445388644)

[Tabla 15. Control de calidad 90](#_Toc445388645)

# LISTA DE FIGURAS

[Ilustración 1. Contenido y fechas de entrega 13](file:///C:\Users\María\Downloads\SPMP%20Linea%20Base%20v1.2.docx#_Toc445388646)

[Ilustración 2. Estructura Scrum [11] 19](file:///C:\Users\María\Downloads\SPMP%20Linea%20Base%20v1.2.docx#_Toc445388647)

[Ilustración 3. Ciclo de vida en cascada [12] 22](#_Toc445388648)

[Ilustración 4. Ciclo de vida en espiral [12] 24](#_Toc445388649)

[Ilustración 5. Proceso Racional Unificado [17] 26](#_Toc445388650)

[Ilustración 6. Proceso Racional Unificado [8] 26](#_Toc445388651)

[Ilustración 7. BPMN Modelo de ciclo de vida 29](#_Toc445388652)

[Ilustración 8. Organigrama 39](#_Toc445388653)

[Ilustración 9. Características del ScrumMaster [11] 41](file:///C:\Users\María\Downloads\SPMP%20Linea%20Base%20v1.2.docx#_Toc445388654)

[Ilustración10￼. Relating size and velocity [30] 47](#_Toc445388655)

[Ilustración11￼. Additional estimation effort yields very little value beyond a certain point [31] 48](#_Toc445388656)

[Ilustración 12. Planning [32] [49] 50](#_Toc445388657)

[Ilustración 13 Plan de entrenamiento 52](#_Toc445388658)

[Ilustración 14. Entregables 66](#_Toc445388659)

[Ilustración 15. BPMN Riesgos 72](#_Toc445388660)

[Ilustración 16. BPMN Control de cambios 78](#_Toc445388661)

[Ilustración 17. Pares por grupo 83](#_Toc445388662)

[Ilustración 18. Calidad de los documentos. 86](#_Toc445388663)

[Ilustración 19. Calidad de los códigos 87](#_Toc445388664)

[Ilustración 20. Calidad de los diseños 88](#_Toc445388665)

[Ilustración 21. Progreso del proyecto 89](#_Toc445388666)

# VISTA GENERAL DEL PROYECTO

## 6.1 VISIÓN DEL PRODUCTO

The Walking Dev espera que para el 2018 su aplicativo web D-EVent, que posee una implementación “responsive”, con lo cual se busca que sin importar la plataforma desde la cual sea abierta esta pueda ser visualizada de manera adecuada, se haya posicionado en el mercado a nivel nacional como el producto líder en el proceso de planificación de eventos y así ser un referente en la manera en la que se gestionan y programan todo tipo de eventos a pequeña y mediana escala. Por lo cual, se busca abarcar y llegar a todo tipo de persona natural que se encuentre entre los 18 y 70 años que esté interesada en llevar este tipo de actividades.

Con D-EVent los usuarios anteriormente mencionados podrán tener una guía de cómo organizar eventos que ellos deseen sin salir de sus casas, ni acudir a personas externas. Esta aplicación recomendará los mejores sitios para realizar los eventos, dependiendo del número de personas y del presupuesto, además permitirá tener una lista de regalos a la cual podrán acceder las personas que estén en la lista de invitados, de este modo podrán escoger el regalo perfecto. D-EVent también mantendrá actualizados a los usuarios con las tendencias actuales en la industria de organización de eventos.

Actualmente existen aplicaciones parecidas para lo que se proyecta que sea D-EVent. Una de estas aplicaciones es EventPlanner, la cual se encuentra disponible para dispositivos Android, cuenta con un conjunto de herramientas para ayudar a planificar y organizar su boda, fiesta o cualquier otro evento, tiene: lista de invitados con sus respectivas invitaciones, lista de tareas, lista de regalos y un planificador de presupuesto. Con EventPlanner se pueden organizar varios eventos al mismo tiempo, de fiestas simples a los eventos más importantes. Otra aplicación, disponible para IOS, es Pro Party Planner, la cual le ayudará a organizar eventos de manera efectiva mediante el seguimiento de los invitados, el presupuesto y la logística en un solo lugar. Esta aplicación a diferencia de EventPlanner tiene opciones de organización, como por ejemplo en Pro Party Planner el usuario podrá organizar a los invitados en las mesas a través de un mapa del sitio del evento.

## 6.2 PROPÓSITO, ALCANCE Y OBJETIVOS

### PROPÓSITO

A lo largo del desarrollo del proyecto se espera que todos los miembros de The Walking Dev integren todos los conocimientos previos vistos en asignaturas anteriores, junto con los adquiridos en la asignatura Ingeniería de Software para poder realizar un proyecto y al final realizar un producto (realización de eventos), teniendo un acercamiento a lo que es un proceso de desarrollo de software a nivel empresarial.

Además, aplicar los diferentes estándares y metodologías de calidad, documentación, riesgos, entre otros, diseñados por las diferentes entidades dedicadas a ello (IEEE, ACM, ente otros.) con el fin de desarrollar software de manera organizada, entregando un producto de calidad.

Al final del proyecto D-Event como aplicación para dispositivos móviles tiene como objetivo facilitar la organización de diferentes eventos tanto como empresariales, sociales, matrimoniales, familiares, entre otros.

*“Ayudar al mundo a reunirse al alcance de un clic”*

### ALCANCE

Las principales funciones a implementar en nuestro aplicativo web dirigido a la planificación de eventos son:

* Permitirá crear una cuenta como Usuario Creador (organizador).
* El Usuario Creador podrá seleccionar los invitados que desea que participen en el evento.
* Registrar en una base de datos la información de los usuarios e invitados, esto incluye todos los ítems que él tenga en la planificación de sus eventos.
* Suministrarle al usuario una lista de los pasos a seguir para realizar un evento que pueda diligenciar.
* Brindarle al usuario una gama de eventos predeterminados para escoger.
* Permitir al usuario agregar personas que colaboren en el evento.
* Permitir al usuario crear una lista de regalos.
* Generar recordatorios que le permitan al usuario estar al tanto de sus actividades.
* Generar plantillas predefinidas que el usuario podrá modificar para la organización y distribución de invitados.
* Recomendar al usuario establecimientos según la necesidad y el evento a realizar.
* Generar balance que describa la distribución del presupuesto.

Las características que no se van a implementar en la aplicación web son:

* Comunicación entre la aplicación y los sitios de eventos.
* Personalización de la aplicación por parte del usuario.
* Añadir nuevos tipos de eventos a la aplicación.

### OBJETIVOS

* General

Elaborar un aplicativo web con tecnología “responsive” que apoye la planificación de eventos.

* Específico

* Aplicar los conocimientos y técnicas aprendidas en clase y a lo largo de la carrera, en un proyecto similar a los desarrollados en el mundo laboral.
* Promover la capacidad de investigación en los integrantes del equipo que conlleven a la entrega de un producto de calidad.
* Fortalecer en los integrantes, habilidades de trabajo en equipo, organización, coordinación, comunicación que les permitan sobresalir y desenvolverse en su vida profesional.
* Desarrollar un aplicativo web que reinvente y facilite la forma tradicional de organizar eventos.
* Generar un ambiente de trabajo tranquilo en el que cada integrante del grupo cumpla puntual y responsablemente con las tareas asignadas, y se sienta a gusto con su rol desempeñado y con el producto que elabora.
* Incentivar a todos los miembros del grupo a tener una mentalidad curiosa frente a los conocimientos que se adquirirán a lo largo del desarrollo del producto.

## 6.3 SUPUESTOS Y RESTRICCIONES

### SUPUESTOS

* El personal de The Walking Dev presentará una actitud colaboradora en todo momento del proyecto.
* El personal de The Walking Dev cuenta con el tiempo necesario para la realización del proyecto.
* Se cuenta con todas las herramientas necesarias para garantizar la calidad y el desarrollo del producto.
* Los miembros de The Walking Dev tienen conocimiento del paradigma orientado a objetos, tanto de sus características como de su uso.
* El cliente no hará modificaciones en los requerimientos planteados al iniciar el proyecto, estos se mantendrán hasta culminar.
* Todos los integrantes del grupo de The Walking Dev trabajarán y estarán presentes en todo el desarrollo del proyecto hasta su entrega final.

### RESTRICCIONES

* El personal de The Walking Dev tiene una disponibilidad de 8 horas por semana dedicadas a la asignatura y proyecto.
* Nuestro cliente y profesor Miguel Torres estableció unas fechas específicas para las diferentes entregas del proyecto, estas fechas son inamovibles:
  + Primera entrega 10 de marzo
  + Segunda entrega 21 de abril
  + Entrega final 24 de mayo
* Disponibilidad de la sala de redes, lugar donde se realizarán las reuniones.
* La aplicación deberá poseer persistencia y una comunicación cliente/servidor [1]
* Se debe contar con una interfaz gráfica de usuario [1].
* La aplicación debe desarrollarse mediante el paradigma orientado a objetos [1].
* El proyecto cuenta con recursos económicos limitados por lo que no se harán inversiones de dinero más allá de las previstas inicialmente.
* El desarrollo del producto usará un sistema de control de versiones para el código [1].
* La aplicación será desarrollada en su totalidad sin hacer uso de generadores de código [1].
* Habrá reuniones con el profesor Miguel Torres y con la monitora Erika Harker cada semana para mostrar los avances de los Sprints.
* El 70% del producto debe realizarse en menos de 3 meses.
* El equipo de trabajo de The Walking Dev consta de 7 personas, no se permitirá que personas ajenas al proyecto trabajen en este.

## 6.4 ENTREGABLES

Los entregables que recibirá el cliente se encuentran definidos en la *ilustración 1* y se describirán a continuación.

### DESCRIPCIÓN DE ENTREGABLES:

Software Project Management Plan (SPMP): Describe el proyecto, las personas que se encuentran involucradas, los procesos y estándares y lineamientos que deben seguirse durante el desarrollo del Proyecto. [2]

Casos de uso: Es una colección de escenarios con éxito y fallo relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo [3]

Primer prototipo: El primer prototipo debe tener implementado al menos el caso de uso más difícil. Tanto el primer prototipo como el segundo deben contener:

* Diseño y Documentación
* Código y Documentación del código
* Utilizar estándares de programación
* Ejecutable y manuales de instalación, administración y ejecución (acorde a lo que se está entregando) [1]

Ilustración 1. Contenido y fechas de entrega

Software Requirements Specification (SRS): Detallas los Requerimientos del sistema por parte del cliente. Este contendrá tanto los requerimientos funcionales y no funcionales de las necesidades, siendo clasificados como primordiales u opcionales. [2]

Software Design Description (SDD): Contiene un diseño del software a fondo. [2]

Segundo prototipo: Implementado al menos el 70% de los Casos de uso. [1]

Reporte de Pruebas, Manuales y Aplicación: Contiene información detallada de cómo funciona el sistema, e instruye al usuario como instalar y utilizar la herramienta. [2]

## 6.5 RESUMEN DE CALENDARIZACIÓN Y PRESUPUESTO

### RESUMEN DE CALENDARIZACIÓN

El proyecto de Ingeniería de Software está compuesto por tres Entregas, que representa cada uno de los entregables que se le harán al cliente. Cada Entrega tiene una duración de máximo 7 semanas y está dividido como se muestra a continuación.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Inicio | Fin | Semanas |
| Entrega 1 | 31/01/2016 | 10/03/2016 | 6 |
| Entrega 2 | 10/03/2016 | 21/04/2016 | 7 |
| Entrega 3 | 21/04/2016 | 24/05/2016 | 6 |

Tabla 1. Duración de cada entrega del proyecto

Para ampliar la información y conocer más detalles de cómo estará planeada cada entrega a lo largo del semestre ir a la sección 9.3.2 [CALENDARIZACIÓN](#_9.3.2_CALENDARIZACIÓN).

### RESUMEN DE PRESUPUESTO

De acuerdo a la cantidad de créditos que tiene la materia de Ingeniería de Software, que se encuentra establecida en el pensum con 4 créditos, cada miembro del grupo deberá trabajar ocho horas semanales aproximadamente, discriminadas de la siguiente manera:

* 4 horas teóricas: En estas los miembros deben documentarse teóricamente y resolver dudas de conceptos, necesarias para la implementación del proyecto.
* 4 horas prácticas: Aquí deben aplicar el concepto, fundamentar, desarrollar, crear todo lo que respecta al proyecto y sus entregables.

Se podrán manejarán más horas de trabajo o menos, teniendo en cuenta la complejidad de cada una de las diferentes actividades, pero esta será la línea base del grupo. De acuerdo a lo anterior se puede relacionar las horas de trabajo de la siguiente manera para cada Entrega:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Semanas | Horas individuo | Horas Equipo |
| Entrega 1 | 6 | 48 | 336 |
| Entrega 2 | 7 | 56 | 392 |
| Entrega 3 | 6 | 48 | 336 |

Tabla 2. Horas de trabajo por entrega

En la sección 9.3.4 Asignación de presupuesto y justificación, se mostrará detalladamente el costo de cada entrega con sus respectivos argumentos. A continuación se mostrará una tabla con el resumen de costos por entrega:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Entrega 1 | Entrega 2 | Entrega 3 | Total |
| Costo | 8,960,000.00 | $9,830,000.00 | $5,700,000.00 | $24,490,000.00 |

Tabla 3. Presupuesto por entrega

El presupuesto de este proyecto salió a partir de la relación entre hora laborada vs costo por persona que es de $ 10.000.00 COP y un estimado de 522 horas de trabajo da como resultado un costo total de $24,490,000.00 COP. Para soportar estos valores se ha tomado como base los datos generados por el Observatorio Laboral [4], donde se coge como base la carrera Ingeniería de Sistemas, sin ninguna derivación adicional en su nombre, donde se registra que para los graduados de 2013 el salario promedio de los mismos fue de $1,979,749.00 COP.

Al ser The Walking Dev un equipo conformado por personas cercanas a los últimos semestres, se ha hecho una reducción a ese precio base, generando un salario mensual de $1,600,000.00 COP para cada integrante por 40 horas de trabajo mensual, lo cual al desglosarlo en horas correspondería a los $ 10.000.00 COP obtenidos, es importante aclarar que en el documento se establecen 48 horas por persona o más, lo cual es el dato real, las 40 horas se emplean más es como referencia para el cálculo de costos.

Para ver informe del observatorio laboral:

Ver Anexo: Observatorio Laboral

Para ver la estimación del costo del proyecto:

[Ver Anexo: Estimación Costo Proyecto](file:///C:/Users/Estudiante/Downloads/Anexos/Estimación%20costo%20proyecto.xlsx)

Ofertas laborales que soportan el precio sugerido:

Ver link: [Oferta de trabajo](http://www.computrabajo.com.co/ofertas-de-trabajo/oferta-de-trabajo-de-ingeniero-sistemas-estudiante-de-ultimo-semestre--en-bogota-dc-DFDC57CAF2D6EE43)

## 6.6 EVOLUCIÓN DEL PLAN

### OBJETIVO

Que el lector entienda el proceso general para hacer cambios en el presente documento y hacerlos efectivos dentro del proyecto. [5]

### RESPONSABLES

* Scrum Master
* Scrum Team: Configuración y calidad

### DESARROLLO

Para realizar cambios en el presente documento, estos se deben aprobar por un integrante del equipo de The Walking Dev. Para ello se seguirá el control de cambios definido en el Plan de administración y documentación (ver Sección 12.3)

# GLOSARIO

**CASOS DE USO:** Es una colección de escenarios con éxito y fallo relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo [3]

**DISEÑO RESPONSIVE:** El diseño web responsive o adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles. [6]

**EVENTO:** Cualquier tipo de reunión realizada a pequeña y mediana escala (ej. matrimonios, cumpleaños, etc.)

**PRODUCT BACKLOG:** es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto [7]

**PRODUCT OWNER-CLIENTE:** es el responsable de maximizar el valor del producto y del trabajo del equipo de The Walking Dev [7]

**SCRUM MASTER:** El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado [7]

**SCRUM TEAM:** consiste en los profesionales que desempeñan el trabajo de entregar un incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. [7]

**SCRUM:** Un marco de trabajo por el cual las personas puede acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. [7]

**SPRINT BACKLOG:** Una lista de tareas que define el equipo de trabajo para un Sprint.

**SPRINT:** es una unidad de planeación en la que se valora el trabajo que se va a realizar, se seleccionan las particularidades por desarrollar y se implementa el software. [8]

**STAKEHOLDERS:** es una palabra del inglés que, en el ámbito empresarial, significa “interesado” o “parte interesada”, y que se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa. [9]

**UML:** es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas no software [3]

**XP:** es un método ágil, enfatiza en la colaboración, rapidez y la creación temprana del software, y el desarrollo de prácticas hábiles [10]

# CONTEXTO DEL PROYECTO

## 8.1 MODELO DE CICLO DE VIDA

Un modelo de ciclo de vida del software describe todas las actividades que intervienen en la creación de un producto de software; los estilos de desarrollo de los productos varían enormemente entre los diferentes tipos de proyectos que requieren diferentes tipos de tareas y diferente organización de estas.

La selección del modelo de ciclo de vida es realizada por todo el grupo y en ella se evalúan las necesidades y los problemas específicos del proyecto, para así elegir el modelo más adecuado para los procesos de desarrollo. La selección del ciclo de vida ayuda a asegurar que el desarrollo del proyecto sea eficiente y eficaz. Para esto se ha escogido como principal modelo Scrum, el cual va a ser complementado con algunas prácticas de XP. Más adelante, se realizará la explicación de las prácticas escogidas del modelo Scrum y de XP.

Debido a que existen muchas fuentes en las que encontramos modelos de ciclo de vida del método Scrum y que cada fuente se refiere a las fases de manera diferente aunque en el fondo describan lo mismo, por ejemplo, en el libro “Agile & Iterative Development” de Craig Larman, nos menciona cuatro fases: planear, puesta en marcha, desarrollo y publicación , o en el libro “Ingeniería de Software” de Ian Somerville que nos plantea tres fases con Scrum: la primera es la planeación del bosquejo, le sigue una serie de ciclos sprint, donde cada ciclo desarrolla un incremento del sistema y cada ciclo consta de 4 fases: valoración, selección, desarrollo y revisión y finalmente, la fase de cierre del proyecto concluye el proyecto, se tomó de referencia el libro “Essential Scrum” de Kenneth S. Rubin para describir el modelo de ciclo de vida de Scrum, ya que en este libro se encuentran detalladamente todas las fases, en especial el ciclo Sprint del que habla Ian Somerville, en el que nos enfocaremos con más detalle ya que es el corazón de Scrum.

### SCRUM

Scrum es un método ágil para desarrollar productos innovadores. Con este método, se crea un Product Backlog, que consiste en una lista priorizada de los elementos y otras capacidades necesarias para desarrollar un producto satisfactorio. Orientados por el Product Backlog se trabaja primero en los ítems más importantes o de mayor prioridad. Cuando se agotan los recursos como el tiempo, ningún trabajo que no se haya completado va a estar en un nivel de prioridad menor que el trabajo completado.

Ver Anexo: Product Backlog.

El trabajo por sí mismo se realiza en cortas iteraciones llamadas Sprint. Típicamente un sprint dura entre una semana a un mes, en donde el equipo completo está enfocado en producir un incremento del producto potencialmente entregable [11]. Las fases que se realizan en cada sprint se muestran en la ilustración 8.1, las cuales se explican a continuación:

* **PLANEACIÓN DEL SPRINT:** Durante la planeación del sprint, los equipos Scrum se ponen de acuerdo en un objetivo para el sprint, y el equipo de desarrollo determina los ítems que están ligados a él y que realmente pueden entregar al final del sprint.

Para adquirir confianza en lo que pueden entregar, los equipos Scrum crean un plan de cómo van a completar los ítems del Product Backlog. El Product Backlog y el plan forman el Sprint Backlog. El cliente comparte el primer objetivo del sprint, presenta la priorización del Product Backlog, y responde a las preguntas que tengan los integrantes sobre los ítems del Product Backlog.

La planeación del Sprint se hace cuando se inicia cada Sprint y no debe tomar más de 4 a 8 horas en hacerse. Durante la ejecución de la planeación el Scrum team trabaja diligentemente para determinar qué tanto del proyecto final puede realizar para la entrega y al final hacen un compromiso realista de la planificación del Sprint. El Scrum Master observa la actividad de planeación, hace preguntas de sondeo y ayuda para poder obtener un resultado exitoso. Aunque el Scrum

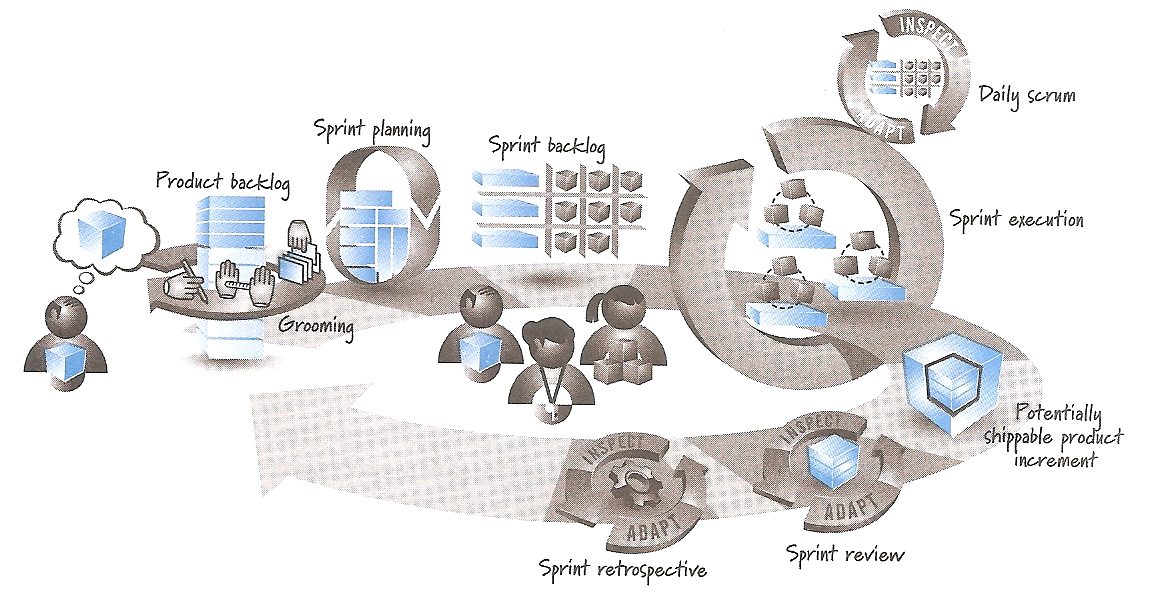
Master no puede decidir en nombre de los integrantes de los Scrum team sobre a que se deben comprometer a hacer, puede cuestionar el compromiso que está haciendo el equipo para asegurarse de que es realista y apropiado.

Ilustración 2. Estructura Scrum [11]

* **EJECUCIÓN DEL SPRINT:** Es el trabajo que realizan los Scrum Team para cumplir con el objetivo del sprint, es un proyecto pequeño en sí mismo donde se llevan a cabo todos los trabajos necesarios para realizar un incremento del producto potencialmente entregable. La ejecución del sprint dura la mayoría del tiempo que dura el sprint. Empieza después de planear el sprint y termina cuando la revisión del sprint empieza. Podría durar entre ocho a diez días.

Los integrantes de los Scrum team se auto-organizan y determinan la mejor manera para llegar al objetivo establecido durante la planeación del sprint. Además, cada día van actualizando el Sprint Backlog para estimar qué tanto esfuerzo se ha hecho para cada tarea no terminada; en el anexo Plantilla Sprint Backlog se puede ver un ejemplo de cómo se van anotando por día el esfuerzo que se ha tenido en las tareas.

El Scrum Master participa como entrenador, facilitador y remueve impedimentos, haciendo cualquier cosa posible para ayudar al equipo a que tenga éxito este no asigna trabjo al equipo ni les dice cómo deben hacer el trabajo. El cliente debe estar disponible durante la ejecución del sprint para resolver dudas, revisar el trabajo que se va realizando y hacer retroalimentación al equipo, discutir ajustes al objetivo del sprint si es necesario, y para verificar que se han cumplido los criterios de aceptación de los elementos del Product Backlog.

Ver Anexo: Plantilla Sprint Backlog.

Ver Anexo: Sprint Backlog.

REUNIONES SCRUM: Duran 15 minutos y se hacen diariamente durante un sprint. Sirven como una inspección, sincronización y adaptación del plan de la actividad para apoyar al equipo auto-organizado a realizar mejor su trabajo.

* **REVISIÓN DEL SPRINT:** Durante la revisión del sprint se inspecciona y adapta el resultado del trabajo, que se convertirá en un incremento potencial del producto. La entrada a la revisión del sprint es el Sprint Backlog que le da a todo el equipo la información del esfuerzo que se tuvo en las tareas del sprint.

La revisión del sprint provee una vista transparente del estado del producto. Es el momento para realizar preguntas, hacer observaciones y recomendaciones. El cliente, el Scrum Master y el Scrum team deben estar presentes en la revisión del sprint, así todos podrán escuchar la misma información y ser capaz de responder a preguntas relacionadas con el sprint y el incremento del producto.

* **RETROSPECTIVA DEL SPRINT:** la retrospectiva del sprint le da la oportunidad a todo el equipo Scrum de dejar de chocar por un momento y pensar, en esta fase, el equipo examina qué pasó, analiza la manera en la que se hizo el trabajo, identifica formas de mejorar, y hace planes para implementar esas mejoras.

Cualquier cosa que afecte el cómo el equipo hace el trabajo, está abierto a escrutinio y discusión, incluyendo procesos, prácticas, comunicación, entorno, herramientas, etc.

Esta fase es crucial para la mejora continua que Scrum ofrece, mientras que en otras organizaciones se espera hacer la retrospectiva en un lapso muy largo de tiempo, con Scrum se realiza en cada sprint lo que permite al equipo tener ventaja de puntos de vista y datos antes de que se pierdan. Debido a que en esta fase se reflexiona sobre el proceso, es necesario que solo el equipo Scrum esté presente, incluyendo el cliente, el Scrum Master y los Scrum team.

Finalmente, la fase de cierre del proyecto concluye el proyecto, completa la documentación requerida, como los marcos de ayuda del sistema y los manuales del usuario, y valora las lecciones aprendidas en el proyecto. [8]

### XP (Extreme Programming)

XP al igual que Scrum es un método ágil, de esta se tomará:

* La práctica Programación por pares, en donde los desarrolladores trabajan por parejas y cada uno comprueba el trabajo del otro; además, ofrecen apoyo para que se realice siempre un buen trabajo.
* La práctica de Refactorización, en donde se espera que todos los desarrolladores re factoricen de manera continua el código y, tan pronto como sea posible, se encuentren mejoras de éste. Lo anterior conserva el código simple y mantenible. [8]
* La práctica de Pruebas en donde todas las pruebas deben correr con un resultado binario: pasa o falla, de tal manera que no se requiera la intervención humana para cada prueba. Los test de aceptación se escriben con colaboración del cliente, ellos definen el nivel de aceptación. [10]

### 8.1.1 ANÁLISIS DE ALTERNATIVAS Y JUSTIFICACIÓN

Parte del éxito y de la calidad del proyecto depende del modelo de ciclo de vida y otras prácticas que se usen como base para la estructuración del mismo, debido a ello se hace necesario realizar una investigación de diferentes modelos que permitan visualizar las ventajas y desventajas que puedan tener al aplicarlo al proyecto, buscando cumplir los objetivos de este de la mejor forma y que sea el más apto teniendo en cuenta las características y las restricciones del proyecto además de las necesidades y habilidades del equipo de trabajo. En este orden de ideas, el tener información de diferentes modelos, permite tomar una mejor decisión a la hora de elegir los modelos y prácticas a usar.

Cabe resaltar que rara vez se aplica un modelo al pie de la letra, ya que no todas las etapas de un modelo cubren el proceso de desarrollo del proyecto debido a la naturaleza de este, por ello se combinan diferentes metodologías con sus prácticas, la unión de estas y su buena aplicación conducirán al éxito del proceso y, por lo tanto, como en reiteradas ocasiones lo ha afirmado el ingeniero Miguel Torres, conducirá a un producto de calidad.

#### Ciclo de vida en cascada

Es el primer modelo publicado sobre el ciclo de vida de software [8], ha sido el predecesor y la base de otros modelos. Consiste en varias etapas que no se superponen, es decir, primero se debe terminar la etapa que se está desarrollando para poder continuar con la siguiente ya que cada fase brinda información relevante para el comienzo de la siguiente.

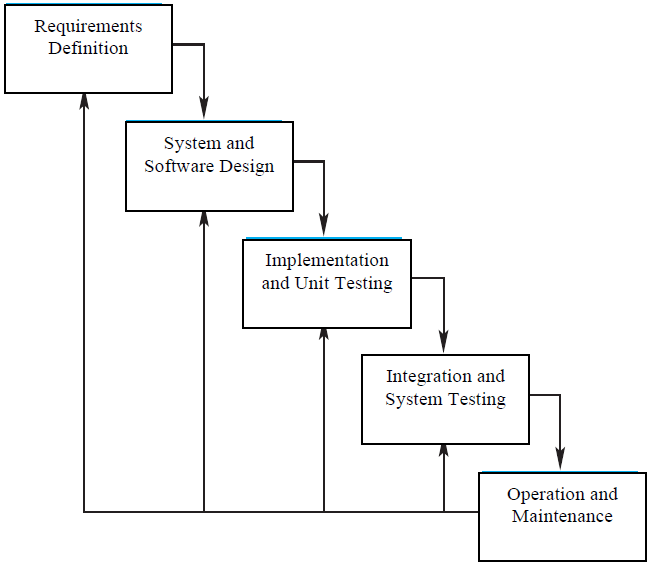


Ilustración 3. Ciclo de vida en cascada [12]

Ventajas [12]

* Fácil de entender y aplicar.
* Ampliamente utilizado y conocido.
* Refuerza buenos hábitos: definir antes de diseñar y diseñar antes de implementar.
* La documentación excesiva e inflexible permite la evolución y mantenimiento a externos al equipo de desarrollo original.
* La documentación permite que el proceso sea visible y monitorizado por los administradores contra el plan de desarrollo. [8]

Desventajas [12]

* No refleja la naturaleza iterativa del desarrollo exploratorio.
* Es poco realista esperar que los requisitos sean precisos tan tempranamente por lo que se dificulta responder a los posibles cambios.
* Postergar o ignorar cambios puede conllevar a que el sistema no haga lo que el usuario desea. [8]
* Es costoso realizar cambios.
* Retrasa el descubrimiento de errores graves ya que el software se entrega al final del proyecto.
* Difícil de integrar la gestión de riesgos.
* Para equipos y proyectos pequeños resulta más costoso que en organizaciones dedicadas al desarrollo de software debido a su alto grado de documentación y administración.
* Como no se ven resultados hasta el final puede ser desconcertante para el cliente.

¿Cuándo se recomienda?

* Cuando los requerimientos se entiendan bien y sea improbable el cambio radical durante el desarrollo del sistema.
* Cuando el sistema a desarrollar tiene rigurosos requerimientos de seguridad, fiabilidad y protección. [8]
* Cuando se desarrolla software que proporcionará un servicio a otras aplicaciones como bases de datos relacionales, compiladores o sistemas operativos seguros. [13]

Con base a lo anterior y a la naturaleza del proyecto a desarrollar no es recomendable usar el modelo de ciclo de vida en cascada ya que los requerimientos no están definidos desde el principio y conforme va evolucionando, será necesaria la retroalimentación y la realización de cambios. Adicionalmente al no haber interacción con el cliente ni revisiones internas no se pueden realizar mejoras significativas que propicien la calidad del producto final.

Por otro lado, el modelo al ser tan estricto e inflexible en sus fases, puesto que se necesita que se termine la etapa actual antes de proseguir, no permite que se desarrollen actividades en paralelo lo cual es necesario para el equipo de trabajo debido a las distintas habilidades con las que cada uno cuenta y a su disponibilidad de tiempo.

#### Ciclo de vida en espiral

Es un marco del proceso de software dirigido por el riesgo y su gestión, lo cual lo diferencia de otros modelos. Cada fase se representa como un ciclo en la espiral, no es lineal ni retrocede [8]. Adicionalmente, espiral combina características del modelo de prototipos en etapas y el modelo de cascada, dichos prototipos son incrementos en el grado de definición e implementación del sistema mientras disminuye su grado de riesgo y por ende los cambios a realizar en el sistema, ya que supone que el cambio es resultado de riesgos del proyecto. [14] [15]

Otra característica del ciclo de vida en cascada es su fase de revisiones cuyo objetivo principal es asegurar que las partes interesadas (clientes, usuarios, etc.) estén comprometidas y se puedan hacer mejoras en el sistema y asegurar la calidad de este.

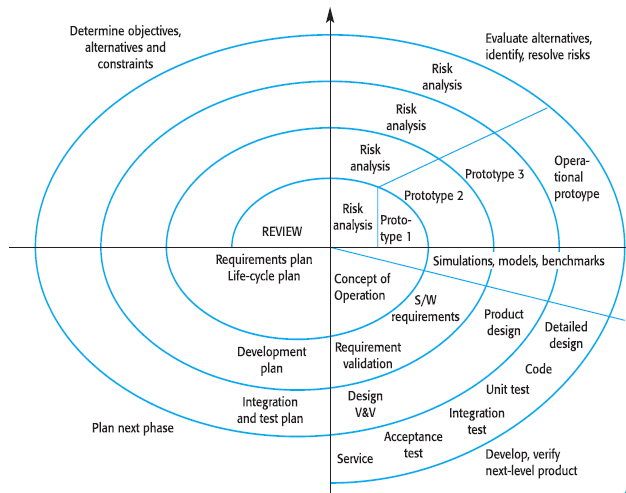


Ilustración 4. Ciclo de vida en espiral [12]

Ventajas [12]

* Alto grado de análisis de riesgos.
* Capacidad de reutilización.
* Mejor productividad. [16]
* Es bueno para proyectos grandes o críticos.
* Se desarrolla el software desde inicio del ciclo de vida y no al final.
* Eliminación de errores desde etapas tempranas del ciclo de vida.
* El compromiso de los involucrados en la revisión de los avances del proyecto conlleva a un sistema de calidad.

Desventajas

* Requiere que los contratos con los stakeholders sean flexibles ya que no se puede estimar con certeza el costo total del sistema y el tiempo requerido.
* Puede ser un modelo costoso de usar.
* El análisis de riesgos requiere conocimientos muy específicos.
* Dependencia de la capacidad del equipo de desarrollo para identificar los riesgos correctamente para el siguiente ciclo.
* El éxito del proyecto depende altamente de la fase de análisis de riesgos.
* No funciona bien para los proyectos más pequeños.
* Es complejo.

A primera vista, como equipo se había mencionado hacer uso de este modelo de ciclo de vida de software debido a su desarrollo por incrementos y a las pruebas y revisiones que se hacen. Adicionalmente, porque como prototipo se pensaba entregar los diferentes documentos que hacen parte del proyecto de clase.

Pero de acuerdo con la información anterior, no es conveniente emplearlo para el desarrollo de nuestro software debido a que el proyecto no es grande, hay una restricción de tiempo para entregar el producto final y el análisis de riesgos requiere de tiempo y aptitudes que no son fuertes en el grupo de trabajo. Aunque es necesario realizar el análisis y la gestión de riesgos durante la evolución del sistema, no tendrá el alto nivel de granularidad que plantea este modelo.

#### RUP: Proceso racional unificado

Es un modelo de ciclo de vida moderno en fases, que se vinculan más con la empresa que con actividades del proceso; a diferencia de otros con una sola perspectiva, este define tres: perspectiva dinámica (fases a través del tiempo), estática (actividades del proceso) y práctica (buenas prácticas a usar durante el proceso). Permite el desarrollo por incrementos tanto por fases como en general.

Una de sus características es la importancia que le da al modelado visual que apoya en UML, aspecto que se relaciona con el hecho de que RUP es dirigido por casos de uso y no por la alta especificación (documentación) o análisis de riesgos. [8] [13]

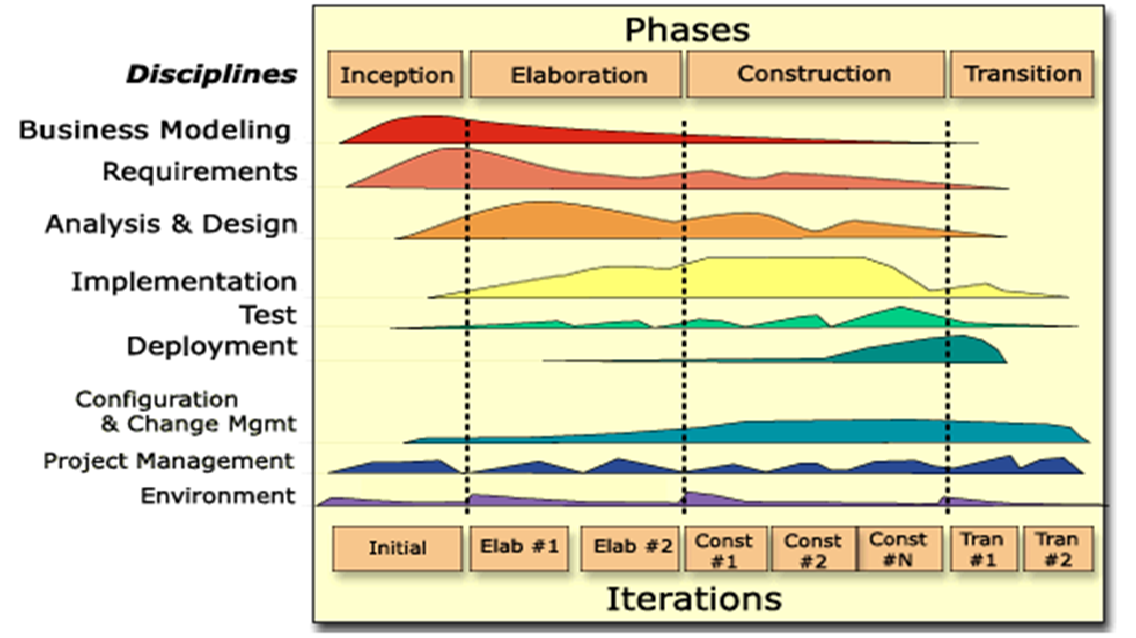


Ilustración 5. Proceso Racional Unificado [17]

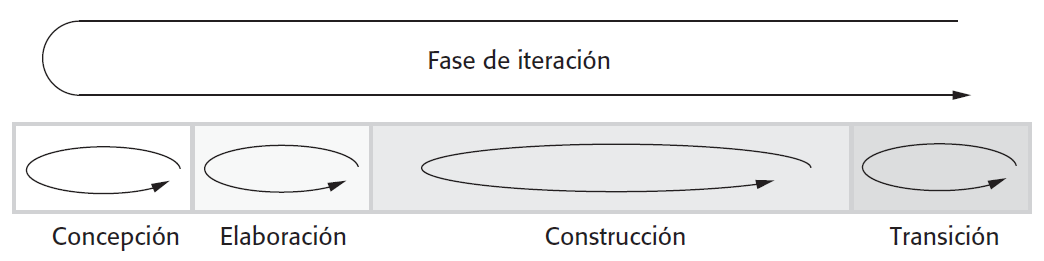


Ilustración 6. Proceso Racional Unificado [8]

Ventajas

* Permite el desarrollo del sistema en paralelo.
* Garantiza que el software cumpla con los estándares de calidad de la organización.
* Control y análisis de cambios antes de aceptarlos.
* Permite la reutilización de componentes.
* Realizar cambios en etapas tempranas.

Desventajas

* La fase de transición es una actividad costosa y en ocasiones problemática.
* No es adecuado para todo tipo de desarrollo.

Para realizar la elección del ciclo de vida, por un lado, nos apoyamos en los criterios definidos en el Construx “Lifecycle model selection” : se compararon los ciclos de vida anteriores; el resultado se muestra a continuación en la Tabla 4: Selección del modelo de ciclo de vida, el rango de calificación que se usó es Alto, medio y bajo. La columna “Proyecto” relaciona el nivel de importancia que tiene el aspecto en el presente proyecto.

Tabla 4. Selección del modelo de ciclo de vida

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Aspectos | Proyecto | Scrum | XP | Cascada | Espiral | RUP |
| Entendimiento de los requerimientos al inicio del proyecto | **Medio** | Medio | Medio | Alto | Medio | Alto |
| Entendimiento de la arquitectura del sistema | **Bajo** | Bajo | Bajo | Alto | Medio | Alto |
| Confiabilidad | **Medio** | Medio | Medio | Alto | Alto | Alto |
| Planeación por adelantado | **Medio** | Medio | Bajo | Alto | Alto | Medio |
| Gestión de riesgos | **Medio** | Bajo | Bajo | Medio | Alto | Bajo |
| Calendarización rígida | **Medio** | Alto | Alto | Alto | Alto | Bajo |
| Correcciones y cambios durante el proceso | **Alto** | Alto | Alto | Bajo | Alto | Alto |
| Progreso visible al cliente | **Alto** | Alto | Medio | Bajo | Alto | Medio |
| Progreso visible al gerente | **Alto** | Alto | Alto | Alto | Alto | Alto |
| Afinidad |  | **7** | **5** | 2 | 4 | 3 |

El resultado de la tabla anterior apoya la inclinación que tiene el equipo de trabajo por los métodos agiles debido a que por la naturaleza del proyecto percibimos que era el que más se acoplaba; como decidimos no hacer la planeación total del proyecto desde el principio, sino por el contrario, planear por partes y enriquecer las secciones conforme se avanza, Scrum y XP son los modelos que usaremos a lo largo del semestre.

#### CONCLUSIÓN: MODELO DE CICLO DE VIDA QUE SE USARÁ EN EL PROYECTO

Todo el equipo de The Walking Dev aprobó usar Scrum y XP como modelos de ciclo de vida base debido a la naturaleza del proyecto, a los resultados de la tabla 1 y a que por criterio propio se quiso experimentar con Scrum. Por lo tanto, se utilizarán las fases descritas anteriormente de Scrum, así como se muestra en el diagrama BMPN (Ilustración 7).

Primero se realiza el Product Backlog con base a los requerimientos establecidos, los cuales son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito [8].

A partir de los requerimientos se planean los Sprints, y no con base a las historias de usuario que plantea Scrum, que son una descripción simplificada de un requerimiento, debido a que todo el equipo de The Walking Dev y el cliente conocen el uso de requerimientos, y no a las historias de usuario. En consecuencia, se realizarán casos de uso para describir los requerimientos funcionales.

Posteriormente se ejecutarán los Sprints, en los cuales se adaptó de la práctica de XP, programación en parejas, así cada par de miembros del equipo está sentado junto a una pantalla de ordenador para escribir el código. Estos equipos son auto-organizados para así determinar la mejor manera de lograr el objetivo establecido en los Sprints. La documentación que se hará durante todo el proyecto será tenida en cuenta como parte del producto a entregar, ya que ni Scrum ni XP tienen prácticas respecto a la documentación.

En cada Sprint se realizarán reuniones que según Scrum se hacen diariamente, pero como este proyecto es de magnitud pequeña y los integrantes de The Walking Dev no están solamente dedicados a este proyecto, ya que son estudiantes, se realizarán dos veces por semana. Al finalizar el Sprint se entregará el incremento del producto, en el cual se harán las pruebas de XP para así aceptar o rechazar el incremento del producto que se realizó en el Sprint. Después se realiza la retrospectiva del Sprint, para revisar lo que se hizo y así encontrar oportunidades de mejora. Por último, al finalizar todos los Sprints se hará el cierre del proyecto.

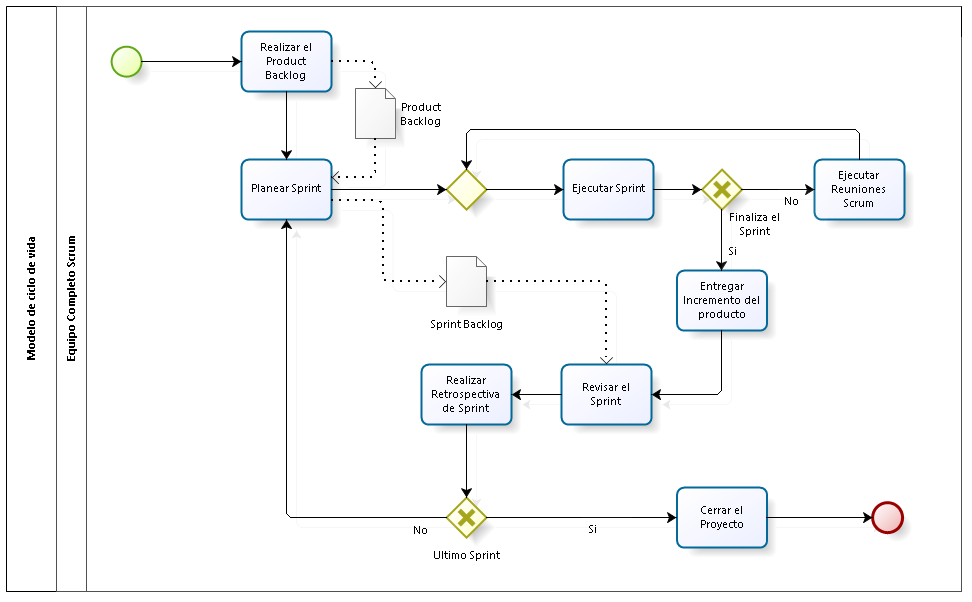


Ilustración 7. BPMN Modelo de ciclo de vida

## 8.2 LENGUAJES Y HERRAMIENTAS

**Lenguajes de Programación:**

Los lenguajes de programación que se utilizarán para desarrollar la aplicación D-Event serán JavaScript y PHP, los cuales son lenguajes orientados a objetos.

**JavaScript** es un lenguaje de programación que se utiliza principalmente para crear páginas web capaces de interactuar con el usuario, con JavaScript se accede al mundo de las páginas web dinámicas [18].

**PHP** es uno de los lenguajes de programación más populares para el desarrollo Web y se ha convertido en la opción para el desarrollo de aplicaciones web orientadas a bases de datos sobre todo por su escalabilidad, facilidad de uso y el amplio soporte para diferentes bases de datos y formatos de estos [19].

Mediante el uso de estos dos lenguajes de programación y las herramientas Atom y Laravel desarrollaremos nuestro aplicativo web ya que son los recursos necesarios para su creación.

**Herramientas:**

A continuación, se hará una descripción de las herramientas que utilizaremos a lo largo del desarrollo de nuestro proyecto y que nos facilitarán y permitirán completar cada una de sus etapas.

* **Comunicación:**

1. **WhatsApp Messenger**: Es una aplicación de mensajería multiplataforma que permite enviar y recibir mensajes. Ya que es una de las aplicaciones de mensajería instantánea más usada hoy en día es una herramienta que nos permitirá una comunicación rápida en The Walking Dev.
2. **Slack**: Permite organizar las conversaciones de grupos de cualquier índole en diferentes canales. Además, nos da la posibilidad de compartir todo tipo de archivos. Al igual que Trello los servicios prestados por esta herramienta están disponibles tanto desde navegador web como en aplicación móvil, además no necesita de una configuración especifica solo el registro en su página web: [Link de registro](https://slack.com/).

* **Organización:**

1. **One drive**: Es una herramienta la cual nos permite acceder nuestros archivos desde cualquier lugar y en cualquier dispositivo, de igual manera podemos compartir y trabajar conjuntamente con nuestro equipo de trabajo. One drive está disponible para su uso desde navegador web además de contar con una aplicación tanto para dispositivos móviles como para computadoras. Esta herramienta no necesita de una configuración específica y se puede descargar desde aquí: [Link de descarga](https://onedrive.live.com/about/es-419/download/).
2. **Trello**: Es una manera gratuita, flexible y visual de organizarlo todo con cualquiera [20]. Esta herramienta nos permite organizar las tareas a realizar en las diferentes etapas del proyecto dándonos así la posibilidad de saber que no se ha hecho, que se está haciendo y que ya se hizo. Esta herramienta nos ofrece sus servicios tanto desde navegador web como en aplicación móvil; no necesita de una configuración específica y se puede realizar el registro desde aquí: [Link de registro](https://trello.com/).
3. **Microsoft Project**: Facilita la administración de los proyectos y la colaboración con las herramientas adecuadas para jefes de proyecto, equipos de proyecto y responsables de toma de decisiones [21], de esta manera Project nos ayudara a administrar nuestro proyecto de inicio a fin. Esta herramienta es de pago así que usaremos la prueba gratuita que ofrece Microsoft: [Link de descarga](https://products.office.com/es/Project/try-or-buy-project-management-software).
4. **Microsoft Excel 2013**: Herramienta de hoja de cálculo la cual nos permite la creación de plantillas, métricas y formatos necesarios en las diferentes etapas del desarrollo de nuestro proyecto. Excel no necesita de alguna configuración específica y desde el siguiente enlace se puede descargar su prueba gratuita: [Link de descarga](https://products.office.com/es-co/excel).
5. **Microsoft Word 2013**: Herramienta para la creación y manejo de documentos la cual usaremos en todo el desarrollo del proyecto. Al igual que Excel no necesita de una configuración específica y se puede descargar su prueba gratuita desde el siguiente enlace: [Link de descarga](https://products.office.com/es-co/word).
6. **Microsoft Power Point 2013**: Herramienta para la creación de presentaciones la cual usaremos en nuestro proyecto con el fin de exponer al cliente el trabajo realizado en cada una de las entregas. La prueba gratuita de esta herramienta se puede descargar desde el siguiente enlace: [Link de descarga](https://products.office.com/es/powerpoint).
7. **Zotero 4.0**: Es una herramienta que ayuda a recolectar, organizar y exportar referencias bibliográficas directamente desde nuestro navegador de internet . Además de los plugins que se pueden descargar de esta herramienta para cada uno de los navegadores web, se puede descargar una aplicación para computador desde el siguiente enlace: [Link de descarga](https://www.zotero.org/download/).
8. **Google Drive**: Es un servicio de almacenamiento en la nube el cual usaremos como repositorio de algunos de los archivos de nuestro proyecto. Google drive es completamente gratis y se puede usar desde navegador web, aplicación móvil y aplicación para computador; esta herramienta no necesita de ninguna configuración específica y desde el siguiente enlace se puede descargar la aplicación para computador: [Link de descarga](https://www.google.com/intl/es/drive/download/).
9. **Planning Poker**: La aplicación Planning Poker es una forma divertida de para comprender mejor los requerimientos de un proyecto ágil de software. Mediante el uso de esta herramienta será más claro identificar qué es lo que el cliente quiere y se podrán estimar las tareas a realizar mediante el consenso de todo el grupo de trabajo. Planning póker es una aplicación gratuita para Android.

* **Diseño:**

1. **Bizagi 3.0**: Es una herramienta para el manejo de los procesos de negocio mediante la construcción de diagramas BPMN (Business Process Model and Notation). Mediante el uso de Bizagi podremos crear diagramas y modelos de procesos que serán de gran importancia en nuestro proyecto. Bizagi no necesita una configuración específica y desde el siguiente enlace se puede descargar su prueba gratuita: [Link de descarga](http://www.bizagi.com/es/productos/bpm-suite/modeler).
2. **Enterprise Architect**: Es una herramienta de diseño UML y Análisis de negocio principalmente usada para para modelar y documentar. Entre otras cosas Enterprise Architect nos ayudara a diseñar el diagrama de casos de uso de nuestro proyecto, parte fundamental del desarrollo de este. Esta herramienta es de pago, así que para el desarrollo del proyecto usaremos su prueba gratuita la cual se puede descargar del siguiente enlace: [Link de descarga](https://www.sparxsystems.es/enterprisearchitect/).
3. **Adobe Illustrator**: Es una aplicación de gráficos vectoriales que te permite crear logotipos, iconos, bocetos, tipografías y complejas ilustraciones para cualquier formato. Con esta herramienta crearemos los iconos y logotipos de nuestro proyecto, además de imágenes para los documentos. Adobe Illustrator es una herramienta de pago así que usaremos su prueba gratuita la cual se puede descargar desde el siguiente enlace: [Link de descarga](https://creative.adobe.com/es/products/download/illustrator?promoid=KSPFH).

* **Desarrollo:**

1. **Github**: Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git, de esta manera nos permitirá manejar el versionamiento del código de nuestro proyecto. Es una herramienta gratuita, no necesita de una configuración específica y su aplicación para computador se puede descargar desde el siguiente enlace: [Link de descarga](https://desktop.github.com/).
2. **Digital Ocean**: Es un proveedor de infraestructura en la nube centrado en la simplificación de la infraestructura web para desarrolladores de software. Esta herramienta es la que nos dará el hosting de nuestro aplicativo web, Digital Ocean es exclusivamente de pago y ofrece sus servicios desde $5 dólares por mes.
3. **MySQL**: Es un sistema de gestión de bases de datos que nos permitirá desarrollar nuestro proyecto permitiéndonos guardar y gestionar la información de los usuarios de nuestra aplicación. La herramienta se puede descargar desde el siguiente enlace: [Link de descarga](https://www.mysql.com/downloads/).
4. **Laravel**: Es un framework de código abierto para desarrollar aplicaciones y servicios web en PHP de una manera elegante y simple, de esta manera Laravel se define como el framework de PHP para artesanos web. En el siguiente enlace se puede encontrar la guía de instalación de Laravel: [Link guía de instalación](https://laravel.com/docs/4.2).
5. **Atom 1.5.3**: Es un editor de texto moderno y accesible, una herramienta que puede personalizar para hacer cualquier cosa y también utilizar de manera productiva sin tener que tocar un archivo de configuración. Esta herramienta nos ayudara a crear el código de la aplicación a desarrollar en nuestro proyecto. Es una herramienta gratuita y está disponible para su descarga a través del siguiente enlace: [Link de descarga](https://atom.io/).

### 8.2.1 ANÁLISIS DE ALTERNATIVAS Y JUSTIFICACIÓN

Inicialmente en la parte de comunicación se empleaba la aplicación WhatsApp pero con el paso del tiempo encontramos que no era muy efectiva ya que al tener un grupo de conversación activa con todo el equipo del proyecto, los mensajes de mayor relevancia se iban perdiendo a medida que surgía un nuevo tema de conversación acerca del proyecto.

Con base a lo anterior, se decidió cambiar la forma de comunicación y usar Slack, ya que esta herramienta permite crear canales de conversación de una manera organizada y además nos permite compartir archivos de todo tipo, a diferencia de WhatsApp que solo permite compartir archivos multimedia. De esta manera, acordamos usar WhatsApp únicamente para convocar reuniones extraordinarias o para informar a todos los miembros del proyecto de alguna eventualidad imprevista.

Por otro lado, en lo que refiere a la creación, gestión y edición de texto optamos inicialmente por usar el editor de texto de Google (Google Docs) ya que permite trabajar sobre los documentos en simultaneo, pero esto fue hasta que se empezaron a presentar problemas con el formato de los documentos a causa de la incompatibilidad de Word con Google Docs.

Para solucionar el problema con los formatos, se pensó cambiar a One drive, herramienta mediante la cual se puede compartir archivos y editar documentos creados en Word y Excel sin problema alguno debido a la compatibilidad entre ellos ya que tanto Word como One drive son herramientas de Microsoft .

Además de Google Drive y One drive como herramientas para compartir archivos y editar documentos conjuntamente, se contempló la idea de usar Dropbox para ese mismo fin; idea que fue descartada debido a que no es una herramienta muy eficaz a la hora de editar documentos en simultaneo, con esto, decidimos finalmente usar One Drive como nuestro repositorio principal de información seguido de Google Drive, lugar donde se guardarán también algunos archivos como artículos relevantes encontrados.

Para lo que tiene que ver con la organización de las actividades y tareas se pensó inicialmente en usar la herramienta Google Keep ya que da la posibilidad de crear listas de chequeo con el fin de saber si algo ya se hizo o no, pero esta herramienta no nos aportaba mucho más que eso, por lo cual empezamos a usar Trello, aplicación más completa para este fin.

Finalmente, en lo que respecta al análisis y modelado decidimos usar Enterprise Architect y Bizagi, ya que son herramientas profesionales muy completas las cuales nos facilita el desarrollo de nuestro proyecto gracias a que nos permiten crear los diagramas base de la aplicación que vamos a crear, además de que no son nuevas para nosotros, pues todos los integrantes de The Walking Dev ya las hemos usado anteriormente para diferentes proyectos y/o actividades.

## PLAN DE ACEPTACIÓN DEL PRODUCTO

Para esta fase se tendrá en cuenta las normas y criterios establecidos por el cliente desde el inicio del semestre y otros establecidos por The Walking Dev.

### OBJETIVO

Dar a conocer los criterios que se tendrán en cuenta para dar como terminada y aceptada una entrega de forma que sea de satisfacción tanto para el cliente como para el equipo de trabajo. [5]

### RESPONSABLES

* Scrum Team: Calidad y configuración

Al estar en contacto con los otros sub-equipos y examinar otros factores que también afectan la calidad del proyecto, son los más adecuados para llevar a cabo la labor de verificar que los criterios definidos sean cumplidos o tomar medidas en pro de su cumplimiento.

* Scrum Master

Como conoce con claridad cuáles son los requerimientos de mayor importancia y monitoriza el avance del proyecto, puede decidir si acepta según su criterio alguna tarea.

* Scrum Team: Análisis y documentación

Deben estar al tanto de los criterios establecidos para la aceptación de una tarea o sección para que en el desarrollo de ellas sean de satisfacción del cliente y del equipo, además ello garantiza que no tenga que rehacer su trabajo.

### DESARROLLO

El cliente Miguel Eduardo Torres, solicita tres entregas del producto: en la ilustración que se puede observar en la sección 6. 4 Entregables en la que se muestran los documentos y elementos a presentar en las fechas establecidas para cada entrega. Adicionalmente, estos tres entregables del producto deben tener en cuenta la hora en que debe presentarse: debe ser en el intervalo de 4:00pm a 4:15pm, en el salón de clase, a través de un medio digital: DVD o USB Storage, y debe incluir un archivo Leame.txt que detalle el contenido de este. Por otro lado, en cada entrega se debe realizar un reporte gerencial que consiste en un análisis detallado del estado del proyecto.

La segunda y tercera entrega deberán contener obligatoriamente las correcciones pertinentes a la anterior entrega y la actualización del SPMP. [1]

Cada prototipo debe utilizar estándares de programación y contener:

* Diseño y documentación
* Código y documentación del código
* Ejecutable y manuales de instalación, administración y ejecución

Lo anterior en cuanto al cliente se refiere. En cuanto a los integrantes de The Walking Dev se ha fijado un formato para la entrega de documentos, Ver Anexo: Plantilla Para Documentos, en el cual se ha definido portada, tipo de fuente y se han establecidos los formatos para títulos y subtítulos. Cada documento debe estar correctamente referenciado usando formato IEEE y debe contener una tabla con el historial de cambios Ver Anexo: Plantilla Historial de Cambios. El equipo decidirá dar como terminado una sección cuando sus documentos se adecuen a la

Por otro lado, el cliente ha definido unos criterios con los cuales se evaluará cada entrega Ver Anexo: Rúbricas. El equipo de The Walking Dev decidió que una sección del proyecto estará finalizada y lista para presentar al cliente cuando consideren que se cumplen dichos criterios que están descritos en la Plantilla de Calidad de Documento, Ver Anexo: Plantilla Calidad de Documentos.

## 8.4 ORGANIZACIÓN DEL PROYECTO Y COMUNICACIÓN

### 8.4.1 INTERFACES EXTERNAS

En este numeral se procederá a describir toda aquella persona o entidad que afectan/influyen de alguna manera en el proyecto que realiza The Walking Dev. Por ende se ha desglosado a los actores en dos tablas que se pueden visualizar a continuación:

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE | DESCRIPCIÓN | RESPONSABILIDADES | DATOS DE CONTACTO |
| Miguel Eduardo Torres Moreno MsC | Ingeniero de Sistemas de la Universidad Nacional de Colombia 1999, Magíster en Ciencias de la Computación de Mississippi State University en Estados Unidos 2003. Docente de la materia de Ingeniería de Software de la Pontificia Universidad Javeriana. | - Guía el proyecto de software  - Define los requerimientos a ejecutar en el proyecto  - Revisa y aprueba documentos en forma responsable  - Establece los parámetros con los que se debe analizar, planear, diseñar y ejecutar el proyecto  - Monitoriza el avance en el desarrollo del software | E-mail:  [metorres@javeriana.edu.co](mailto:metorres@javeriana.edu.co)  Tel: (+57 1) 320 8320  Ext. 5316 |
| Erika Jennifer Harker | Estudiante de las carreras de Ingeniería de Sistemas e Ingeniería Industrial de la Pontificia Universidad Javeriana. Monitora de la materia Ingeniería de Software. | -Revisa y aprueba documentos en forma responsable | E-mail: [eharker@javeriana.edu.co](mailto:eharker@javeriana.edu.co)  Celular: (+57) 301 3412386 |
| Daniel Serrano | Ingeniero de Sistemas Pontificia Universidad Javeriana. Empleado full time en Google. Asesor en el Capítulo Javeriano ACM | -Guiar y dar herramientas para el desarrollo del aplicativo web  -Brindar material complementario para los entrenamientos | E-mail:  daniel-serrano@javeriana.edu.co  Celular: (+57) 316 6960472 |
| Alfredo Santama-ria | Estudiante de Ingeniería de Sistemas de noveno semestre. Presidente del capítulo Javeriano ACM | -Apoyar el reforzamiento de conocimiento básico en lo que respecta a programación web  -Ayuda con la configuración de servidores | E-mail: alfredo.santamaria@javeriana.edu.co  Celular: (+57) 318 6017866 |
| Edgar Enrique Ruiz García | Ingeniero de Sistemas del Politécnico Grancolombiano, con una especialización en Seguridad de Redes de la Universidad Católica de Colombia y una certificación internacional como Instructor CCNA de Cisco Systems. | -Avalar el uso de las salas solicitadas  -Enviar los correos respectivos para el préstamo de salas | E-mail: eruiz@javeriana.edu.co  Tel: (+57 1) 320 8320 Ext. 5347 |

Tabla 5. Personas externas que interactúan con el desarrollo del proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE DE LA ENTIDAD | DESCRIPCIÓN | RESPONSABILIDADES | DATOS DE CONTACTO |
| Biblioteca | La Biblioteca es una división de la universidad que se encuentra adscrita a la Vicerrectoría Académica, responsable de apoyar las labores de docencia, investigación y servicio. Tiene la responsabilidad de adquirir, conservar y ofrecer información en las áreas de conocimiento pertinentes para el logro de los fines de la Universidad. | - Proporcionar materiales para la consulta e investigación  - Facilitar espacios en los que se puedan llevar a cabo reuniones con el grupo  - Brindar materiales adicionales como escáner, fotocopiadora, etc. Esenciales para el montaje del proyecto | Email: [biblioteca@javeriana.edu.co](mailto:biblioteca@javeriana.edu.co)  PBX: +57 1 320 8320 Ext. 2132, 2135  Dirección: Carrera 7 # 41 - 00 Bogotá, Colombia |
| Carrera Ing. Sistemas | Esta disciplina se ocupa de proponer, modelar, planear, diseñar, construir, adaptar, gestionar y liderar soluciones que mejoren la calidad de vida de las personas y las organizaciones, aprovechando al máximo las tecnologías de información y comunicaciones. Esto habilita la toma de decisiones, la generación de ventajas competitivas y la innovación. | - Ofrecer los programas base con los cuales se montará y ejecutará el proyecto  - Al igual que la biblioteca brindar espacios de trabajo para el grupo  - Mantener habilitados recursos de información como lo es UVirtual. | Encargado: José Hernando Hurtado Rojas E-mail: [hhurtado@javeriana.edu.co](mailto:hhurtado@javeriana.edu.co)  Tel: (+57 1) 320 8320 Ext. 5310 |
| Digital Ocean | Es un proveedor Estadounidense de servidores virtuales privados, basado en la ciudad de Nueva York. La compañía alquila facilidades de centros de cómputo existentes | -Permitir el despliegue del aplicativo en la web  -Ofrecer herramientas para el mantenimiento de los archivos dentro del servidor  -Facilitar un dominio | Enlace de contacto:  https://www.digitalocean.com/company/contact/ |
| Operarios salas de computo | Son los encargados de facilitar el uso de las salas de cómputo dispuestas en la carrera, dan ingreso a los salones, organizan y les hacen mantenimiento a los equipos existentes. | -Permitir el ingreso a la sala, previamente pedida por el equipo  -Velar porque los espacios se encuentren vacíos a las horas de las reuniones | Encargado: Ramiro Augusto Lamilla Diaz  Email: rlamilla@javeriana.edu.co |

Tabla 6. Entidades externas que interactúan con el desarrollo del proyecto

### 8.4.2 ORGANIGRAMA Y DESCRIPCIÓN DE ROLES

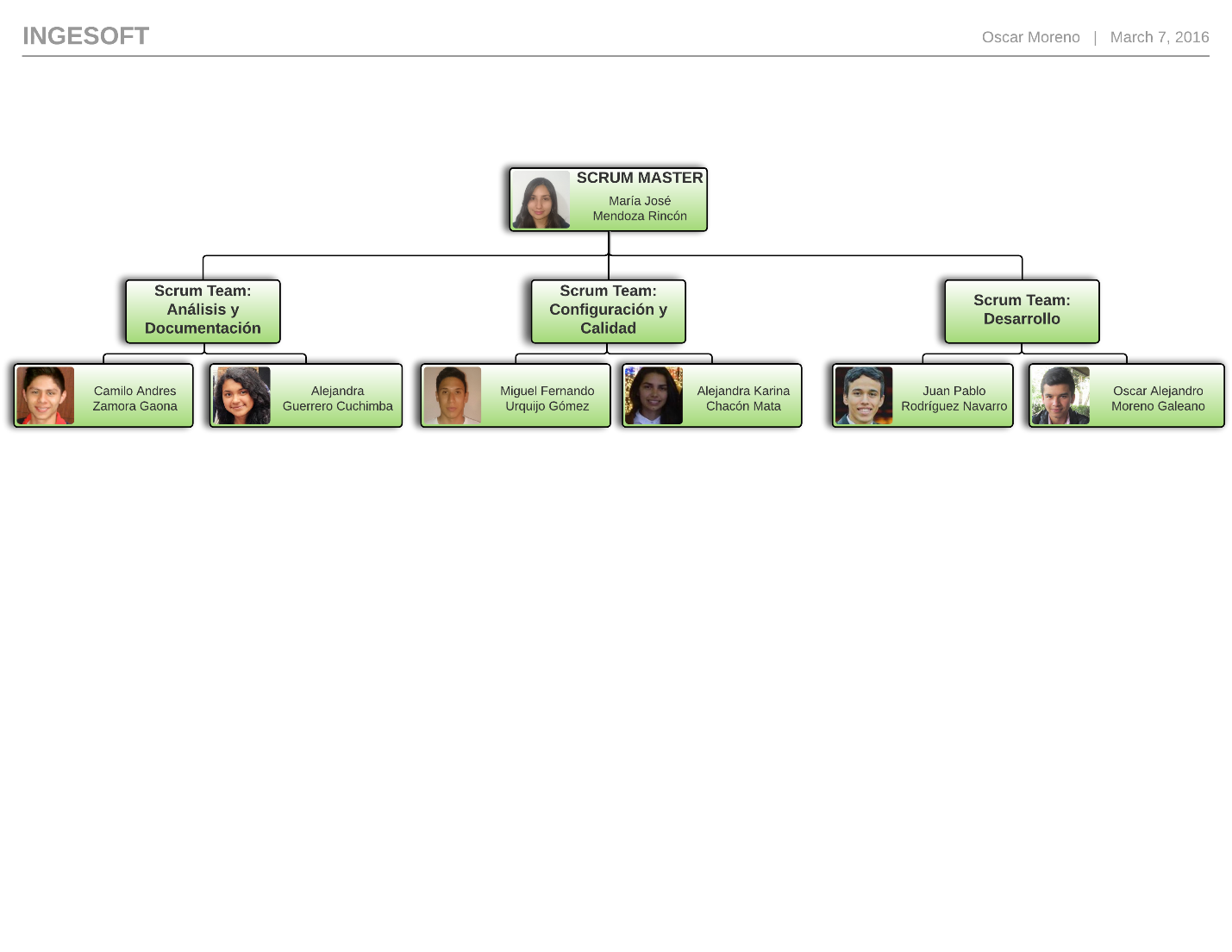


Ilustración 8. Organigrama

#### DESCRIPCIÓN DE ROLES

#### SCRUM MASTER

DEFINICIÓN

El ScrumMaster es el responsable de asegurar que Scrum es entendido y adoptado, hacen esto asegurándose de que el equipo trabaja ajustándose a la teoría, prácticas y reglas de Scrum. Es un líder que está al servicio del equipo, debe facilitar el proceso de esta metodología y proteger los principios ágiles. También necesita proteger el equipo eliminando los impedimentos que ocurran, asegurando que el equipo no se distraiga de la tarea en cuestión. Cuando haya una discusión entre los integrantes de los Scrum Teams, el ScrumMaster toma la decisión final. En ocasiones, los miembros del equipo solamente siguen lo que el ScrumMaster sugiere sobre cómo hacer las tareas. [22]

El ScrumMaster dirige las reuniones, identifica el retraso inicial que se completará en el sprint, y empíricamente mide el progreso hacia el objetivo de la entrega de este conjunto incremental de la funcionalidad del producto. Asegura que todo el mundo realice un progreso, registra las decisiones tomadas en las reuniones y mantiene las reuniones de Scrum cortas y concretas. Trabaja constantemente para reducir el riesgo del producto a través de la entrega incremental de las características, respuesta rápida al desarrollo de obstáculos, y el continuo seguimiento de la entrega de los elementos de retraso. [23]

PERFÍL DEL ROL

* **Experto**: Debe ser un experto en Scrum. Debe entender los problemas técnicos que el equipo tiene que abordar y las tecnologías que va a usar el equipo para crear las soluciones.
* **Interrogativo**: Debe usar sus habilidades de liderazgo en relación con sus conocimientos de procesos, técnica y de negocio para realizar buenas preguntas. Se dedican a la indagación intencional, realizando preguntas que haga que la gente se detenga y diga “Hmm, nunca pensé en eso. Ahora que tú lo preguntaste, me hace pensar en que podría haber otra manera de hacerlo”.
* **Paciente**: Dado que el ScrumMaster prefiere no dar respuestas, necesita ser paciente, dando tiempo a sus equipos a llegar a respuestas apropiadas por sus propios medios.
* **Colaborador**: Debe tener excelentes habilidades de colaboración para el trabajo con el dueño del producto, el equipo de desarrollo y todas las otras partes, incluso cuando no se vean directamente involucrados con Scrum. El ScrumMaster siempre está buscando oportunidades para ayudar a los miembros del equipo de Scrum, logrando un nivel envidiable de colaboración entre el equipo.
* **Protector**: Debe ser muy protector con su equipo. El ScrumMaster es partidario de asegurar la protección del equipo dentro de un gran contexto de toma de decisiones empresariales económicamente viables. Con aguda sensibilidad hacia las necesidades tanto de protección del equipo y de negocios; ayuda al equipo a lograr un equilibrio saludable. Cuando los equipos tengan debe ayudar a superar estas dificultades reforzando cómo usar Scrum más efectivamente.
* **Transparente**: Es transparente en muchas formas de comunicación. Cuando se trabaja con los miembros del equipo, no hay actividades desconocidas o no comunicadas; lo que se ve y se escucha del ScrumMaster es lo que hay que hacer. Promueve la comunicación transparente fuera del equipo. Sin un acceso transparente a la información es difícil para la organización inspeccionar y adaptarse a alcanzar los resultados de negocio deseados usando Scrum. [11]

Ilustración 9. Características del ScrumMaster [11]

Características del ScrumMaster

Experto

Paciente

Interrogativo

Colaborador

Protector

Transparente

#### SCRUM TEAM: ANALISTA Y DOCUMENTADOR

* ANALISTA

DEFINICIÓN

El Analista Funcional es el vínculo de unión entre el usuario y el área informática de la entidad. Su misión consiste en elaborar el análisis de las funcionalidades para el proyecto, así como actualizar y mejorar las conceptualizaciones; es decir, debe trabajar de manera conjunta con el desarrollo funcional de las aplicaciones informáticas, asegurando su correcta explotación y su óptimo rendimiento [24]. De esta manera el analista junto con el cliente, quien es el experto en el problema a tratar, realizan el estudio, el análisis y finalmente llegan a la especificación del sistema que se requiere construir [25]. Otro aspecto importante es que el analista debe permanecer en continua comunicación y trabajo con el administrador del proyecto, esto con el fin de estudiar las necesidades del cliente y los requisitos del sistema en conjunto. Para resumir el trabajo del analista, se puede decir que su función es la de convertir la informalidad de las necesidades del cliente a especificaciones y requerimientos formales desde el punto de vista de ingeniería de software [24].

RESPONSABILIDADES

El análisis se puede dividir en dos fases:

* Especificación de requisitos del usuario.
* Especificación de requisitos de software. [25]

El analista realiza actividades como las siguientes para cumplir con sus objetivos:

* Entrevistar al cliente.
* Verificar si los requisitos especificados son los correctos.
* Realizar en análisis de los requisitos.
* Analizar la estructura básica del sistema.
* Generar los diagramas de la arquitectura. [25]

El rol del analista es muy importante ya que el éxito del proyecto dependerá de una buena especificación de los requerimientos.

* DOCUMENTADOR

DEFINICIÓN

La gestión documental se ha convertido en una necesidad para las organizaciones donde se genere gran cantidad de información e implica el uso de diversas herramientas o métodos de proceder que permiten un mejor funcionamiento de la organización [26].

De esta manera, durante todo el proceso de desarrollo de software se genera una gran cantidad de documentación la cual debe ser almacenada en el repositorio del proyecto [25], esto con el fin de conocer su historia y trayecto, además toda esta documentación necesita ser actualizada conforme se va desarrollando el proyecto. Por ello el objetivo principal del rol de documentador es el de mantener la información generada durante el proceso de desarrollo.

La documentación se suele clasificar en dos categorías:

* **Documentación de procesos:** Registran la información del proceso de desarrollo y manutención del sistema.
* **Documentación de producto:** Estos documentos describen el desarrollo del producto desde los puntos de vista técnico (documentación de sistema) y usuario del sistema (documentación de usuario).

La calidad de la documentación generada es de gran importancia, debido a que la utilidad del sistema se degrada si no hay información adecuada de cómo usar o entender sus características [25]. Para obtener esta calidad, la documentación del proyecto debe seguir estándares.

RESPONSABILIDADES

El rol de documentador debe cumplir con actividades como las siguientes:

* Diseñar y construir un repositorio de información compartido donde se almacenará la información.
* Mantener el repositorio de información.
* Especificar el formato que será usado para elaborar la documentación.
* Durante las reuniones de revisión el documentador elaborará las actas de la reunión.
* Asegurarse de que los documentos mantienen el estándar de documentación definido en el proyecto
* Elaborar el manual de uso del sistema para el usuario final.

El trabajo realizado por el o los documentadores es de suma importancia ya que en base a este el administrador del proyecto podrá elaborar planes y podrá realizar la evaluación del proyecto además de todo lo dicho anteriormente.

#### SCRUM TEAM: CALIDAD Y CONFIGURACIÓN

DEFINICIÓN

Los integrantes de esta rama, en lo que respecta a la configuración serán los encargados de facilitar y definir cuáles son las herramientas necesarias para facilitar las tareas de: administración, coordinación, el intercambio de información, manejar los cambios que se realizan además de la estandarización en el proyecto, ya sea en la documentación o en la parte de desarrollo [7].

En cuanto a calidad debe ayudar en la creación y corroboración tanto para documentos como para software que se cumpla esta característica a lo largo del ciclo de vida del proyecto; debe interceder y preocuparse para que los requerimientos asignados y dependiento de su prioridad sean asignados de manera correcta, que sean comprensibles para el resto del grupo coherentes, etc. Para poder ejecutar y llevar a cabo estas tareas los integrantes de este equipo deben trabajar de manera paralela con el resto de grupos, ya que si se espera al final de los ciclos de vida para tomar correctivos, y hay algún inconveniente en las tareas presentadas por los otros equipos se puede causar pérdidas graves para el proyecto en lo que respecta a tiempo y recursos.

RESPONSABILIDADES

**Actividades de la configuración:**

* Definir cuáles son las herramientas que usarán los diferentes equipos en el desarrollo del proyecto.
* Verificar que el uso que se les da a las herramientas sea el adecuado, y en caso de desconocimiento, instruir al grupo en cómo se maneja correctamente.
* Definir un glosario con el fin de que haya uniformidad en la terminología usada por el equipo.
* Establecer los parámetros con los cuales un archivo puede ser modificado, con el fin de generar concurrencia en los mismos.
* Definir como se deben manejar las versiones de los archivos de los diferentes tipos ya sea documentación o código.

Tomando como referencia el estándar ISO/IEC 25010:2011 se definen las siguientes actividades para asegurar la calidad [27]:

* Validar que los requerimientos establecidos sean comprensibles para todo el equipo.
* Establecer cuáles serán los criterios para evaluar tanto el software como la documentación entregada.
* Establecer plantillas y reglas sobre cómo se debe realizar la documentación.
* Verificar la coherencia y concordancia de los diferentes documentos.
* Verificar que toda investigación esté debidamente soportada por referencias, estas últimas deben provenir de fuentes confiables y estar en el formato correcto [28].

INTERACCIÓN CON OTROS ROLES

* **Scrum Master:** Debe ser el primer referente al que debe acudir este equipo a la hora de tomar decisiones debido a que el master conoce con claridad cuáles son los requerimientos de mayor importancia y con los que se tendrán que definir las reglas de trabajo antes de que comience a ejecutarse un sprint, velando por la estandarización de los procesos y de que estén disponibles las herramientas acordadas dentro del plan de trabajo.
* **Analistas y documentadores:** Debe estar en constante comunicación ya que de la mano generan investigaciones y producciones que sean coherentes, debidamente ordenados y que cumplan con unos parámetros previamente establecidos, de la misma forma las partes deben conocer las herramientas que se manejarán y dominarlas lo necesario.
* **Desarrolladores:** Debe apoyar los procesos de ejecución, verificando constantemente que los requerimientos trabajados sean debidamente abordados y que cumplan las condiciones establecidas por el cliente, por lo que entra también a jugar como tester del producto. [25]

PERFÍL DEL ROL

La persona que desarrolle este cargo es alguien que tiene conocimiento de todas las ramas del proyecto, posee un criterio para tomar decisiones a nivel de software y a nivel de documentación, de la misma forma se caracteriza por ser ordenado ya que debe transmitir con tiempo y anterioridad a los otros equipos cuáles serán las métricas con las cuales serán evaluadas las tareas asignadas. Es alguien con capacidades comunicativas bien desarrolladas por lo que todas las ideas y planteamientos deben quedar claros para los integrantes del proyecto. [25]

#### SCRUM TEAM: DESARROLLADOR

DEFINICIÓN

El equipo de desarrollo consistirá de miembros que encarguen de realizar entregas por cada incremento realizado al final de cada Sprint durante el semestre y al final entreguen el producto ya sea terminado o lleve mínimo el 70% de realización. El equipo de desarrollo es estructurado, organizado y totalmente responsable para organizar y gestionar su propio ritmo de trabajo de acuerdo al calendario. La sinergia entre miembros optimiza la eficiencia y efectividad del equipo de desarrollo.[7]

RESPONSABILIDADES

* Entender los diferentes diagramas realizados por el área de diseño.
* Realizar el código respectivo a los diagramas.
* Actualizar el repositorio con los cambios realizados.
* Probar que el código funcione en diferentes dispositivos.
* Documentar el código realizado. [29]

PERFÍL DEL ROL

El equipo de desarrollo tiene las siguientes características:

* Ser organizados. Ningún otro miembro del equipo ni siquiera el Scrum Master indica cómo realizar el producto a lo largo de los Incrementos.
* El equipo de desarrollo cuenta con todas las habilidades necesarias para crear un Incremento de producto al final de cada Sprint.
* No existe jerarquía en el equipo de desarrollo, todos son desarrolladores, no importa el cargo que desempeñe dentro del grupo.
* No existen sub-grupos, no importa diferentes procesos que se tengan que aplicar, como pruebas o análisis de negocio.
* Los miembros del equipo de desarrollo pueden tener distintas habilidades y áreas en las que se puedan enfocar, pero la responsabilidad recae en el equipo de desarrollo en general.
* El tamaño del equipo de desarrollo debería ser lo bastante pequeño como para permanecer ágil y lo bastante grande como para completar la iteración al final del Sprint. [7]

# 9. ADMINISTRACIÓN DEL PROYECTO

## 9.1 MÉTODOS Y HERRAMIENTAS DE ESTIMACIÓN

### Objetivo General

Definir y fundamentar los métodos y herramientas que se emplearán para la estimación de los recursos con los que cuenta D-Event, en los que se tendrán en cuenta: recursos humanos, disponibilidad de software, disponibilidad de hardware y tiempo. Todo esto con el fin de poder tener un cronograma con el cuál el equipo de The Walking Dev pueda trabajar y distribuirse la carga adecuadamente para cumplir con las entregas en las fechas indicadas y con contenido de calidad. Vale aclarar que al ser una estimación pueden realizarse cambios en el plan propuesto.

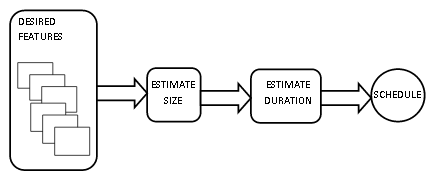


Ilustración10￼. Relating size and velocity [30]

#### Responsables

La estimación estará a cargo de la Scrum Master y una persona de experiencia que componga el equipo de configuración y control de calidad, en este caso Miguel Urquijo. El resto del equipo también tendrá voz y voto en las decisiones de este ítem, más los que guiarán las sesiones donde se realizarán las estimaciones serán las personas mencionadas anteriormente.

#### Recursos requeridos

Para llevar a cabo la estimación es necesario:

* Contar con todo el equipo de trabajo
* Cartas con las cuales se llevará a cabo el Planning Poker
* Un espacio en el que se desarrollará la dinámica
* Microsoft Excel en el que se consignará toda la información y análisis de la estimación

#### Plan de trabajo

Para la estimación se tomaron en cuenta las metodologías propuestas por Somerville en el capítulo 23 sección 5 [8], en el cual se plantean dos formas de realizar esta planeación:

1. Técnicas basadas en la experiencia: La estimación en los recursos asignados a cada requerimiento se basa en la experiencia que posee algún miembro del equipo, en nuestro caso contamos con un integrante de The Walking Dev, el cual ya ha cursado con anterioridad la materia y conoce de primera mano la dificultad de las actividades y tareas que se deben desarrollar a lo largo del proyecto.
2. Modelado algorítmico de costo: Aquí se le da una valoración a las diferentes actividades, tareas y requerimientos, definidos según unos atributos establecidos por el equipo de trabajo, los cuales se tendrán un valor y un costo dentro del proyecto con el fin de calcular cual es el esfuerzo que implica la realización del mismo, a medida que va avanzando el proyecto.

Dadas las definiciones anteriores en este proyecto se implementarán ambos modelos.

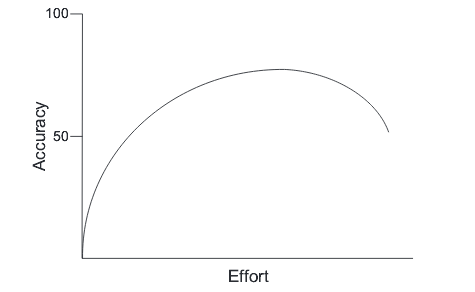


Ilustración11￼. Additional estimation effort yields very little value beyond a certain point [31]

Es importante recalcar que sin importar la metodología escogida y el tiempo invertido en la estimación, nunca se logrará alcanzar a dar una respuesta 100% acertada al costo final y al tiempo establecido para la realización del proyecto tal como lo muestra esta gráfica.

Ahora para aterrizar más estas metodologías es necesario acercarnos al modo de trabajo que el equipo ha escogido, por lo que se aproximaran los conceptos empleados en lo que respecta a implementaciones ágiles.

**DERIVAR UN ESTIMADO**

Esta metodología se compone de diversas técnicas, pero para este proyecto se emplearán 2 de las más usadas:

* Opinión de un experto: Esta metodología se basa en Técnicas basadas en la experiencia la cual ya fue abordada al principio de esta sección, pero es necesario hacer la aclaración de que al implementarse en las formas agiles se deben tener en cuenta una serie de consideraciones. Tal como se aclara en el libro Agile Estimating and Agile Planning [31] este puede resultar contraproducente ya que para realizar estimaciones de este tipo, es necesario que la persona conozca de manera profunda todos los campos de acción, por lo que se aconseja que intervenga todo el grupo para también aportar su opinión, para sacar aproximados más cercanos a la realidad.
* Desagregación: En esta técnica se busca como su nombre lo indica descomponer los requerimientos y actividades en tareas, todo esto con el fin de poder estimar eventos más pequeños, ya que al tener actividades más grandes y menos detalladas es más complejo y menos preciso realizar una estimación adecuada.

En este tipo de estimación, se afirma que al usar varias técnicas combinadas los resultados obtenidos serán mejores, con base en lo anteriormente descrito y otra modalidad llamada analogía surge el Planning Poker.

El **Planning Poker** es una manera en la que todo el grupo entra en acción para la realización de la estimación sobre las diferentes actividades, requerimientos y tareas, que a diferencia de otras técnicas tiene resultados de manera ágil pero así mismo suelen ser confiables. Es una manera didáctica de tomar decisiones ya que emplea cartas, las cuales tienen una numeración previamente definida por The Walking Dev, con las cuales se encargará el equipo de decidir cuál es la prioridad y el costo en lo que a recursos respecta al ítem analizado, la Scrum Master (María José) debe leer una descripción breve del ítem, con la finalidad de que todos sepan qué es lo que se está evaluando. Vale aclarar que todos participan y tienen sus propias cartas ya que todos tienen voz y voto.

En el caso de que las cartas presentadas posean una diferencia significativa no se debe descartar a la primera alguna, al contrario se ha de escuchar los argumentos de quienes presentaron las cartas, con el fin de que el equipo conozca dos puntos de vista igualmente válidos que pueden llegar a dar bases para la toma de una mejor decisión. Al final el ideal es que se llegue a un estimado similar, en caso de no conseguirlo se debe realizar otra ronda, hasta conseguirlo.

Es importante hacer la claridad de que a la hora de realizar este tipo de estimaciones por esta metodología se debe tener en cuenta el Esfuerzo vs. Exactitud, el objetivo y fin último de estas técnicas es alcanzar la mayor exactitud a partir del menor costo, tal como se ve en la Ilustración 11 en la que es mejor aproximarse a la mayor exactitud por el lado en el que haya que emplear un menor esfuerzo.

The Walking Dev se ajusta perfectamente a las condiciones para llevar a cabo un Planning Poker ya que es un grupo pequeño, por lo cual las deliberaciones se pueden realizar de manera rápida. El proyecto está dividido en diferentes módulos, los cuales están demarcados en las especificaciones de las entregas presentadas por el cliente Miguel Torres, en algunos casos se llegará a un mayor nivel de granularidad para su estimación, todas las decisiones tomadas a través de esta técnica sobre las diferentes actividades, tareas y requisitos deben estarán consignadas en un documento en Excel, en el que se debe especificar:

1. El nombre del ítem
2. Una descripción en la que se resuma claramente que se hará
3. La prioridad dentro del proyecto
4. Costo de la realización y ejecución

Para esta primera entrega se ha generado una estimación temprana consignada en el siguiente anexo:

Ver Anexo: Estimación Casos de Uso.

**Planning Poker**



Ilustración 12. Planning [32] [49]

Se usará Planning Poker herramienta que está descrita en la sección 8.2 Lenguajes y Herramientas.

## 9.2 INICIO DEL PROYECTO

### 9.2.1 ENTRENAMIENTO DEL PERSONAL

#### OBJETIVOS

* Identificar las herramientas a utilizar para el proceso de entrenamiento.
* Crear grupos interdisciplinarios para fortalecer el proceso de entrenamiento.
* Promover y orientar al interior del grupo el aprendizaje de las herramientas identificadas durante el proceso.
* Controlar y realizar seguimiento a los procesos identificados y especificados en el proyecto.

#### RESPONSABLES

* El Scrum Team de desarrollo es el encargado de planear e impartir los entrenamientos de las herramientas de desarrollo. Y el equipo de Configuración y Calidad se encargan de las herramientas de control de versiones.

#### RECURSOS REQUERIDOS

* Se usará una plantilla en Microsoft Excel como la herramienta para realizar el plan de entrenamiento del personal.

Ver Anexo: Formato Plan Entrenamiento.

#### PUESTA EN MARCHA

Realizar un análisis de los equipos para identificar las diferentes herramientas que requieren de entrenamiento, así como determinar si ya existe algún conocimiento previo sobre estas; dado el caso que no se tenga tal conocimiento se contactara con las personas encargadas del entrenamiento de dicha herramienta para que proceda con el inicio del proceso de aprendizaje.

El proceso de aprendizaje estará programado para los días jueves de cada semana con una intensidad horaria de dos horas por sesión; con la finalidad de obtener la mayor capacitación posible en la respectiva herramienta.

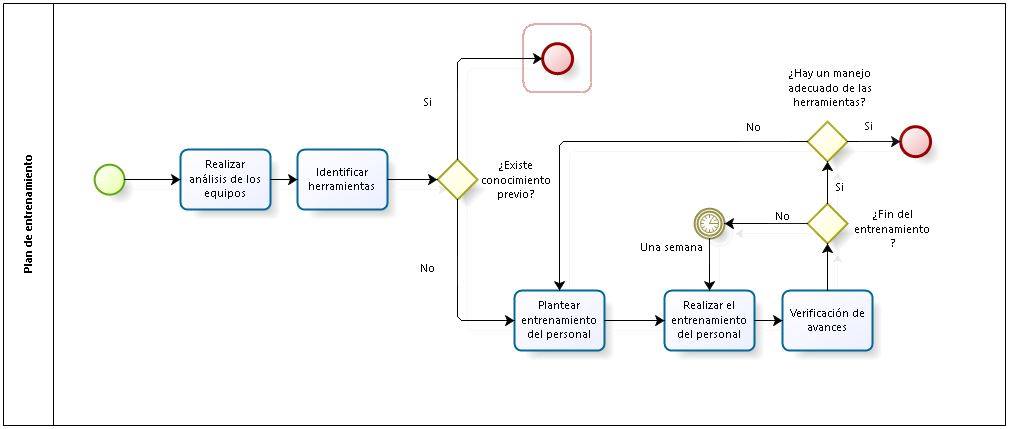


Ilustración 13 Plan de entrenamiento

#### Supervisión y control

Con la finalidad de conocer el progreso en conjunto y promover el aprendizaje en equipo, se realizará semanalmente la verificación de los avances en el proceso de aprendizaje de la herramienta, así como la consigna de las respectivas dudas, con un enfoque práctico hacia la mejora del proceso. Siendo responsabilidad de la consigna tanto del avance como de las dudas de los entes respectivos para cada función.

Para la supervisión y control del proceso de aprendizaje se tendrán en cuenta:

* El proceso en sí, siendo necesario su correcto funcionamiento, en cuyo caso si se observa que el manejo de las herramientas de los integrantes del equipo de trabajo es deficiente después de haber tenido las sesiones de entrenamiento, será necesario replantear el mismo.
* La herramienta:
* En caso tal de que a lo largo del proyecto se encuentre una herramienta disfuncional esta será removida.
* De encontrarse una herramienta con alguna mejora sobre otra que este siendo utilizada en el proceso, dicha herramienta será sustituida.

### 9.2.2 INFRAESTRUCTURA

#### OBJETIVO

A partir de lo definido en la sección [8.2 Lenguajes y herramientas](#_8.2_LENGUAJES_Y), se debe establecer todos los recursos necesarios para el debido montaje y funcionamiento, en los cuales se tienen en cuenta los elementos ya existentes y se plantean de igual forma los que son necesarios conseguir, ya sea a nivel de software o hardware, para que a lo largo del proyecto no se presente ningún percance a causa del fallo en la infraestructura.

#### RESPONSABLES

El montaje y la verificación de toda la infraestructura estará a cargo de las personas que componen el Scrum Team de Desarrollo, los cuales deberán proveer a lo largo del proyecto: las herramientas necesitadas en las diferentes áreas, la instalación de los diversos componentes de software, así como probar que estén funcionando correctamente; de igual forma deben asegurar que los diferentes requerimientos de hardware cumplan con lo estipulado en el documento presente, y que se disponga de los espacios sin contratiempos oportunamente. Es importante recalcar que para disminuir la probabilidad de riesgo en este ámbito es necesario de que se haga una constante verificación y mantenimiento oportuno a las herramientas con el fin de evitar fallos graves a futuro.

#### RECURSOS REQUERIDOS

#### Software

Los recursos necesarios en esta sección están previamente definidos (ver sección 8.2-Lenguajes y herramientas), donde se describe cual es la herramienta, quien interactúa con la misma y como es la interacción dentro del proyecto.

#### Hardware

Además de los recursos de hardware que posee cada integrante del grupo, mostradas en la tabla 7, es necesario contar con un servidor web, el cual servirá para realizar la funcionalidad Cliente-Servidor de la cual necesita la aplicación.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre del integrante | Sistema Operativo | Procesador | Modelo | Tipo de Sistema | RAM | Disco Duro |
| Alejandra Chacón | Windows 10 | Intel Core i5 | Notebook HP Pavilion DM4-1090LA | 64 bits | 4 Gb | 500 Gb |
| Alejandra Guerrero | Windows 10 | Intel Core i5 | HP 14-AC116LA | 64 bits | 4 Gb | 1 Tb |
| Camilo Zamora | Windows 10 | Intel Core i5 | Sony Vaio VPCSB45FL | 64 bits | 4 Gb | 500 Gb |
| Juan Pablo Rodríguez | Windows 8.1 / Ubuntu 14.10 | AMD A10 | HP Envy 15 | 64 bits | 8 Gb | 1 Tb |
| María José Mendoza | Windows 10 | Intel Core i7 | Asus Zenbook UX303L | 64 bits | 8 Gb | 1 Tb |
| Miguel Urquijo | Windows 7 | Intel Core i5 | HP G4-2955LA NOTEBOOK PC | 64 bits | 8 Gb | 750 Gb |
| Oscar Moreno | Windows 10 | AMD A10 | Asus X555D | 64 bits | 8 Gb | 1 Tb |

Tabla 7 Computadores personales de los integrantes de The Walking Dev.

#### Físicos

Los recursos físicos necesarios son: Lugares en los cuales se puedan llevar las reuniones de The Walking Dev y salas de programación en las cuales se puedan desarrollar las diferentes pruebas de software.

#### Plan de trabajo

La instalación de las mismas se hará en las diferentes máquinas de los integrantes de The Walking Dev, según sea su rol, y le competa el uso de la herramienta seleccionada, esta se debe llevar a cabo desde el principio del proyecto, y no cuando se llegue a necesitar, ya que se podrían producir contratiempos mientras se realiza este proceso. Es necesario asegurar una conexión a internet ya que será la fuente de conexión y comunicación dentro del grupo, y será un recurso fundamental para el funcionamiento de algunos programas.

## 9.3 PLANES DE TRABAJO DEL PROYECTO

### 9.3.1 DESCOMPOSICIÓN DE ACTIVIDADES

La descomposición del trabajo total del proyecto es utilizada en la planeación para dar una claridad a todo el grupo de lo que se va a hacer generando una estructura base de la cual se puede partir al iniciar el trabajo, generalmente involucra las siguientes actividades:

* Identificación de los entregables y trabajos relacionados.
* Estructuración y organización del WBS.
* Descomposición de los niveles superiores del WBS hasta los niveles inferiores de los componentes detallados.
* Desarrollo y asignación de códigos de identificación a los componentes del WBS.
* Verificación de que el grado de descomposición del trabajo es necesaria y suficiente. [33]

El WBS (Work Breakdown Structure [34]), es un entregable o una agrupación de la orientación del producto de elementos de trabajo del proyecto de manera gráfica para organizar y subdividir el trabajo total del alcance del proyecto.

Este es una herramienta de vital importancia para la planeación, control y monitoreo de todas las actividades que realiza el equipo de trabajo, ya que da una claridad de lo que se debe hacer para el desarrollo del proyecto. [33]

Ver Anexo: WBS del Proyecto

### 9.3.2 CALENDARIZACIÓN

#### OBJETIVOS ESPECÍFICOS

* Describir las tareas detalladamente, con iteraciones cortas para que se pueda realizar una revisión mucho más concreta y eficiente gracias a su sencillez, facilidad de uso y bajo coste [35].
* Realizar un diagrama de Gantt, que muestre las fechas correspondientes y los diferentes Sprint de cada tarea y su respectiva duración, gracias a una fácil y cómoda visualización de las acciones a realizar, permite realizar el seguimiento y control del progreso de cada una de las etapas de un proyecto. [35]

#### RESPONSABLES

* Scrum Master.
* Cada miembro del grupo es responsable de llevar a cabo sus respectivas tareas antes del tiempo límite estipulado (Sprint).

#### RECURSOS REQUERIDOS

* Microsoft Project se usa para poder realizar el diagrama de Gantt donde se especificarán las actividades y su duración.

#### PUESTA EN MARCHA

Durante cada Sub-Sprint se hará una monitorización clara del progreso para descubrir con facilidad los puntos críticos, los períodos de inactividad y para calcular los retrasos en la ejecución, adicionalmente se revisará si todas las tareas fueron realizadas sin ningún contratiempo, de no ser así se podrá mejorar o reasignar tareas para que se pueda cumplir con las fechas estipuladas para la entrega del Sprint final teniendo en cuenta la calidad de cada tarea y la cantidad de miembros que la ejecutan, revisando que el tiempo de duración sea realista [35].

#### CALENDARIO

A pesar de que en general el proyecto se basa en metodologías ágiles para su ejecución no está demás el uso de diferentes herramientas que nos permitan ampliar la visión del trabajo a realizar ya que, aunque la planeación ágil es buena manejando la incertidumbre, la calendarización con diagramas Gantt nos aporta una mayor claridad de las cosas que por seguro hay que hacer. [36]

* Diagrama de Gantt, Ver Anexo: Diagrama de Gantt.
* Tabla primer Sprint, Ver Anexo: Lista de Actividades Sprint 1.
* Tabla segundo Sprint, Ver Anexo: Lista de Actividades Sprint 2.
* Lista general de Sprints, Ver Anexo: Lista de Sprints.

Los martes se hará la finalización del sprint donde se generará los reportes de progreso. Después de revisado se realiza el siguiente sprint planning con el fin de definir las tareas que se van a hacer en la siguiente iteración ágil.

Es importante resaltar como se hará cada actividad dentro del proyecto durante las diferentes iteraciones ([ver sección: 8.1 Modelo de ciclo de vida](#_8.1_MODELO_DE)) en la ilustración (ver ilustración: Ilustración 7. BPMN Modelo de ciclo de vida)

Después de haberse definido cada actividad para los grupos se desarrolla la actividad durante el tiempo establecido en la calendarización, luego se realiza una revisión por parejas para rectificar que el documento o archivo contenga los mínimos requerimientos para ser aceptado. Al final de cada Sprint se hace una revisión los días martes donde se hace un reporte grupal de todas las actividades; si cumple con las normas de calidad el documento o archivo será aceptado y guardado en el repositorio, de lo contrario se tendrá que repetir la actividad.

### 9.3.3 ASIGNACIÓN DE RECURSOS

Para una mejor trazabilidad de las tareas a realizar, se han designado responsables según los roles creados donde también se busca una revisión de calidad a cada una de ellas por todos los integrantes del equipo. [33]

#### OBJETIVO GENERAL

* Generar una claridad en los encargados de cada tarea a realizar. [37]

#### OBJETIVOS ESPECÍFICOS

* Llevar un control del trabajo que está haciendo cada uno de los integrantes del equipo. [37]
* Llevar control del peso laboral de los integrantes para lograr un equilibrio. [37]

#### RESPONSABLES

El Scrum Master estará encargado del cumplimiento y verificación de las tareas asignadas, más todos los integrantes del proyecto velarán por la actualización de las tareas y estarán comprometidos con cada una de las tareas encargadas.

#### RECURSOS REQUERIDOS

* Microsoft Excel: Se consignará el formato de las tareas asignadas.

#### MATRIZ DE ASIGNACIÓN DE RESPONSABILIDADES

La guía del PMBOK sugiere hacer una RAM (Responsibility Assignment Matrix) ya que es un método útil para la especificación de las responsabilidades de cada uno de los miembros, para este caso se usa PASI (Primary responsibility, Approval authority,

Supporting responsibility,Information only) para describir la participación de cada uno de los miembros del equipo en todas las actividades del proyecto definidas en el WBS porque es el que mejor se adapta a los tipos de responsabilidades que se tiene dentro del modo de trabajo ágil que se está implementando donde hay mucho soporte entre las integrantes. [38]

A continuación, se encuentra toda la lista de tareas y los encargados de ejecutarlas. [39]

Ver Anexo: Recursos Asignados.

### 9.3.4 ASIGNACIÓN DE PRESUPUESTO Y JUSTIFICACIÓN

El presupuesto para el desarrollar de la aplicación The Walking Dev se ha calculado a partir de la cantidad de horas trabajadas y los recursos humanos necesarios para realizar cada tarea existente en el cronograma del proyecto.

La hora que se paga a cada miembro del equipo es de $10.000 [4], que es el valor que se les está pagando actualmente a los estudiantes recién egresados.

El costo total del proyecto, así como la especificación de precio por tarea se encuentra en el anexo de Estimación Costo Proyecto.

Ver Anexo: Estimación Costo Proyecto.

Además del costo del proyecto por iteraciones se va a tener adicionalmente un formato de flujo de caja en donde se indicarán los gastos que se relaciona con el pago a cada uno de los integrantes del grupo e ingresos a las multas pagadas por incumplir con las entregas.

Ver Anexo: Flujo de Caja.

# 10. MONITOREO Y CONTROL DEL PROYECTO

## 10.1 ADMINISTRACIÓN DE REQUERIMIENTOS

Este plan se realizará para la segunda entrega.

## 10.2 MONITOREO Y CONTROL DEL PROGRESO

#### OBJETIVOS

* Describir actividades para gestionar y controlar el progreso individual y grupal.
* Describir actividades para controlar adecuadamente el cronograma.
* Conocer las diferentes herramientas y metodologías que se utilizaran para gestionar el progreso.

#### RESPONSABLES

* Scrum Master. Llevará la organización y será el encargado del monitoreo.

#### RECURSOS

* Contar con el equipo de trabajo.
* Planning Poker.

#### PUESTA EN MARCHA

El equipo de The Walking Dev para el monitoreo y control del progreso ha planteado tres estrategias para gestionar los avances individuales y grupales, permitiendo así llevar un seguimiento al cumplimiento de los acuerdos y el cronograma.

#### Trabajo por parejas

Para una correcta distribución y supervisión de las tareas el equipo se encuentra manejando una organización doble, la cual permite un apoyo de primera mano entre los dos integrantes de cada área, así de ser necesaria alguna otra colaboración se puede contar con las demás áreas. [40]

#### Planning Poker

Como se había mencionado en la sección 9.1 esta es una herramienta que permitirá mediante el uso de cartas, la organización y posterior selección de tareas pendientes mediante un estudio de prioridad y costos. Teniendo todos los integrantes voz y voto para dichas decisiones.

#### Sprint Planning Meeting

Semanalmente se llevarán a cabo encuentros que buscan la exposición de las tareas previas ya desarrolladas y la asignación de nuevas tareas para cada área teniendo en cuenta la variable del tiempo y los puntos restantes. [8]

Adicionalmente se debe tener en cuenta el uso del sprint backlog cuya función es la consigna de horas dedicadas al progreso de la tarea de cada área, este se renueva semanalmente en cada una de las reuniones consignando en el mismo la asignación de las nuevas tareas. [11]

Siendo el encargado de la supervisión y asignación el Scrum Master.

#### Monitoreo y Acciones correctivas.

Permite determinar las tareas que están incompletas o que requieren correcciones, generando la reasignación de tareas a todo el equipo comenzando con aquella persona que incumplió, ocasionando una supervisión de parte de todo el grupo hacia el trabajo de dicha persona quien será monitoreada hasta cumplir con sus funciones de forma correcta con un plazo de 12 horas las cuales posterior a ser excedidas se procede a hacer el correctivo contemplado en “reglas”.

## 10.3 CIERRE DEL PROYECTO

#### OBJETIVO

En este numeral se dará a conocer cuáles son los pasos que se deben seguir para llevar a cabo el cierre del proyecto, especificando como se ven implicados los clientes, interfaces externas y los miembros del equipo The Walking Dev.

#### RESPONSABLES

Los responsables directos será el Scrum Team de configuración y calidad. El Scrum Master así mismo tendrá la potestad para supervisar he interactuar en esta tarea.

#### RECURSOS Requeridos

* Onedrive: Repositorio al que se accederá para la consulta de documentos.
* Word: Ahí se llevarán las actas y se realizarán los comentarios pertinentes.

#### METODOLOGÍA

Para hacer un cierre de proyecto correcto, es necesario basarnos en el modelo de ciclo de vida de Scrum, por lo que para entender mejor los conceptos y partes mostradas a continuación se recomienda revisar la sección 8.1 MODELO DE CICLO DE VIDA.

#### Scrum review :

1. **Desempeño del equipo:** Teniendo en cuenta la rúbrica establecida al inicio del proyecto y con la cual se establecieron los diversos sprints y el Product Backlog, se procederá a evaluar el desempeño de las diversas áreas del equipo, las cuales a lo largo de las semanas irán realizando unos entregables que corresponden a la iteración, estos deben ser revisados y retroalimentados por el resto del equipo. Todo esto será con el fin de:
   1. Verificar la calidad del trabajo realizado.
   2. Verificar cuales son las actividades y tareas faltantes dentro del proyecto.
   3. Evaluar los riesgos ya establecidos, así como los nuevos.
   4. Evaluar las fortalezas y debilidades dentro del grupo.

La información de estas evaluaciones quedará registrada en las actas del equipo:

Ver Anexo: Actas.

1. **Reunión fin de ciclo:** El equipo se debe reunir en una sesión final en la que se debe comenzar a realizar la integración de las partes que componen el entregable, en esta se debe verificar la coherencia, la no ambigüedad del documento y que cumpla las especificaciones del Product Owner. Para de Walking Dev es muy importante la calidad y la forma en la que entrega sus productos por lo cual en la sección 8.3 Plan de Aceptación se especifican más requerimientos para evaluar las entregas.
2. **Hacer una entrega preliminar al Product Owner:** A lo largo del proyecto se sostendrán reuniones con el Product Owner, las cuales permitirán verificar que el trabajo realizado por los equipos es el esperado por el cliente, se anotarán las recomendaciones y comentarios, los cuales deben estar analizados y corregidos en la siguiente iteración. Para la última iteración se presentará el documento final, la idea es verificar que se cumple lo pedido y que el documento está correctamente finalizado, de no ser así el equipo debe corregir las diferentes fallas antes de la fecha de entrega.
3. **Entrega:** El producto se entregará en un disco extraíble o en su defecto en CD, antes de realizar la entrega es necesario realizar las pruebas pertinentes en los dispositivos del equipo, con el fin de evitar errores y verificar que todo este correctamente guardado.

#### Scrum retrospective:

1. **Retro-alimentación del Sprint:** Con base en el 1. y después de haber realizado la evaluación se debe comenzar a hacer un análisis de los resultados obtenidos, en la que se tendrán en cuenta el rendimiento de los equipos, y con base a eso se asignarán las nuevas tareas, de igual forma se deben establecer quienes serán merecedores de reconocimientos, o en su defecto de penalizaciones. Dependiendo de cuál haya sido el desempeño se evaluará si se le da el total del puntaje asignado a la actividad, de lo contrario tendrá una semana más para corregir los errores, en la segunda revisión si no cumple con los requisitos propuestos el equipo verá afectada su nota.
2. **Análisis de riesgos:** También es importante tener en cuenta que se deben mirar cuales son los riesgos que se traían de precedente en el sprint anterior, y a partir de eso mirar si afectaron o no la entrega de los equipos. De no verse afectados, mirar si es necesaria su vigencia para el siguiente Sprint; de afectar, mirar cual será el porcentaje que se le dará al riesgo en el nuevo Sprint y cuáles pueden ser las medidas a tomar para contrarrestar su efecto.
3. **Reporte gerencial:** A partir de todos los ítems mencionados anteriormente se generará un reporte por entrega donde se dará razón de como fue el desarrollo del proyecto, en el que se tendrá en cuenta: el personal, el producto y el proceso. Todo esto quedará registrado en un balance presentado por el Scrum Master.

# 11. ENTREGA DEL PRODUCTO

## OBJETIVO

Se mostrarán cuáles son los entregables que se deben suministrar al cliente en las fechas previamente acordadas, y con los cuales se busca respaldar el compromiso adquirido.

## RESPONSABLES

Los responsables directos será el Scrum Team de configuración y calidad. El Scrum Master así mismo tendrá la potestad para supervisar he interactuar en este rol.

## RECURSOS REQUERIDOS

Herramientas especificadas en la sección 8.2 Lenguajes y Herramientas.

## ENTREGABLES

Ilustración 14. Entregables

Para conocer más de cada uno de estos entregables ir a la [sección 8.3 Plan de Aceptación del Producto](#_PLAN_DE_ACEPTACIÓN). En este caso se discriminan y se presentan como un ítem separado la presentación y el prototipo, ya que estos tienen una fecha y un formato de presentación distintos al de los otros 3 entregables.

* **Prototipo:** Al ser un aplicativo web este debe ser montado en un servidor, el cual permitirá realizar el despliegue del software y que a la hora de mostrar el demo cualquier persona pueda acceder a la URL seleccionada, por lo que debe estar disponible antes de la fecha de entrega del SDD y de la presentación final.
* **Presentación:** Este será un espacio donde el equipo de The Walking Dev deberá estar presente, en el que se llevará a cabo la sustentación del desarrollo y de la implementación del proyecto. Para esto se deberá contar con unas láminas que servirán de soporte en la explicación y que estará acompañada de algunos elementos que se repartirán entre los jueces/clientes con el fin de dar a conocer el producto en mención.

# 12. PROCESOS DE SOPORTE

## 

## 12.1 AMBIENTE DE TRABAJO

El ambiente de trabajo es importante para la coherencia de todos los documentos que se hagan, ya que se consta de un espacio donde todos los miembros puedan interactuar entre ellos con el fin, de mejorar todas las tareas que se realicen; en él también se busca cuidar de cada uno de los integrantes en cuánto a varios aspectos. [41]

#### OBJETIVO GENERAL

* Crear un ambiente de trabajo que aumente la productividad del equipo. [41]

#### OBJETIVOS ESPECÍFICOS

* Buscar que las personas estén motivadas a trabajar.
* Crear un ambiente colaborativo entre los miembros del grupo.
* Crear un espacio para la resolución de dudas y correcciones entre los documentos creados. [41]

#### RESPONSABLES

Los encargados de verificar que estas reglas se sigan y que se genere un buen ambiente de trabajo son el Scrum Team de configuración y calidad, con ayuda y colaboración del Scrum Master para la organización de salidas, buscando la disponibilidad y participación de todos los miembros.

#### RECURSOS REQUERIDOS

* Word: para la descripción de las reglas.
* Whatsapp: para la disponibilidad de salidas y la comunicación entre miembros en trabajos no presenciales.

#### ESTRATEGIA

Para un mejor ambiente de trabajo nos basamos en un concepto llamado capital social, utilizado por Magaret Heffernan, emprendedora y escritora de libros que hablan de temas como gestión de alto rendimiento del talento, innovación continua y el papel de líderes en servir el talento que contratan, quien explica cómo las relaciones entre los miembros de su equipo de trabajo mejora la productividad de una compañía al fomentar la colaboración entre todos, a partir de la empatía generada en encuentros sociales. [42]

A pesar de que una empresa de software esté comprendida por un equipo de trabajo con un nivel de tecnicidad y experticia muy alto no asegura un buen desempeño en el desarrollo del proyecto, ya que la calidad de los ladrillos no son el único factor que influye en la robustez de la pared sino también cómo están conectados entre sí, es por ello que además del intelecto hace falta confianza, conocimiento global de lo que hace el equipo, reciprocidad y normas compartidas que creen calidad de vida  y resiliencia en el equipo. Todas las herramientas que logren estas características dentro del grupo de trabajo hacen parte del capital social. [43]

#### REGLAS

Siguiendo los principios dados por Margaret Heffernan se proponen las siguientes reglas y actividades:

* Tener una participación activa dentro de las redes de comunicación.
* Organizar una salida (mínimo al mes), fuera del ámbito laboral.
* Escribir mensajes motivacionales y alegrar el ambiente cada semana.
* Compartir una comida en jornadas largas de trabajo conjunto.
* Preguntar cómo va el trabajo de cada uno de sus compañeros (semanalmente).
* Tener pausas activas en jornadas largas de trabajo.

#### ACTAS

Para la sustentación del esfuerzo y cumplimiento de las actividades realizadas por cada uno de los miembros, queda registrado por escrito lo hablado en las reuniones Scrum, donde a su vez se pautan las tareas para el día siguiente en actas, las mismas serán redactadas por un miembro diferente en el siguiente orden:

* Juan Pablo Rodríguez
* Alejandra Guerrero
* Miguel Urquijo
* Alejandra Chacón
* Oscar Moreno
* Camilo Zamora
* María José Mendoza

#### REUNIONES CON EL PROFESOR

Siendo las reuniones con el profesor un factor importante para el buen desarrollo del proyecto, se decide establecer las siguientes normas:

* Todos los miembros del equipo deben estar informados de lo que se habló en las reuniones (apuntes, notas de voz, etc.).
* Cada integrante debe formular como mínimo una pregunta con respecto a la parte del proyecto que le compete, para dichas reuniones; estas estarán guardadas en Slack.
* Los participantes a las reuniones deben ser en lo posible, los que hayan manifestado una mayor cantidad de dudas al equipo, sin hallar una respuesta clara con respecto al trabajo a realizar.

#### COMUNICACIÓN ENTRE LOS INTEGRANTES

Debido a que además de las reuniones ya establecidas como lo son los Sprints, los integrantes del equipo no tienen la misma disponibilidad horaria, se estipulan diferentes herramientas para la comunicación explicadas a continuación:

* **Trello:** utilizada para el seguimiento de las tareas asignadas teniendo estos tres estados: to do (tareas a realizar), on doing (tareas en progreso), done (tareas completadas.
* **Whatsapp:** únicamente para el cumplimiento de las reglas pautadas para el fomento de un buen ambiente de trabajo, comunicados importantes y dudas urgentes.
* **Slack:** usada con el fin de obtener información pertinente para la realización del proyecto como: lluvias de ideas, información sobre estados del proyecto, entre otras.

#### AMONESTACIONES E INCENTIVOS

A pesar que dentro de la estructura empresarial del equipo se pretende fomentar un ambiente colaborativo, es imprescindible que haya un esfuerzo en cada una de las entregas por parte de todos los integrantes donde el incumplimiento de las políticas acordadas será motivo de faltas y luego de una acumulación de las mismas, su posterior expulsión del equipo.

#### FALTAS Y MEMORANDOS

En primera instancia, hay que aclarar que todos los miembros deben estar al tanto de las actividades que hacen los demás como se acordó anteriormente en el documento, razón por la cual en caso de notar cualquier irregularidad en el trabajo de algún compañero están en el deber de alentarlo a retomar el trabajo con el debido compromiso, en caso de que este comportamiento sea reiterativo, se recurre a ponerle una falta.

También se pondrá una falta a la persona que llegue después de 15 minutos de haber empezado una reunión.

La falta quedará registrada en las actas y se cobrará una multa de 10.000 pesos, cada tres multas se mandará un memorando explicando el porqué de la falla al correo de [TheWalkingDev@javeriana.edu.co](mailto:TheWalkingDev@javeriana.edu.co) además de la aprobación de todos los integrantes del equipo; luego de tres memorandos el integrante del equipo queda expulsado del equipo.

El dinero cobrado de las multas será guardado y se gastará en el participante al que se le haya dado el reconocimiento o en su defecto se aportará a alguna de las salidas de grupo mensuales.

#### INCENTIVOS

Para cada entrega habrá un reconocimiento al integrante que haya dado un valor agregado al trabajo hecho, el cual consiste en un incremento en su nota final. Cada miembro del equipo hará su postulación en las reuniones Scrum del Sprint siguiente a la entrega y se hará la votación, exponiendo los méritos del compañero.

## 12.2 ANÁLISIS Y ADMINISTRACIÓN DE RIESGOS

El propósito de la siguiente subsección es exponer el plan de gestión de riesgos que se definió para el presente proyecto. Se entiende por gestión de riesgos la aplicación sistemática de políticas, procedimientos y prácticas para establecer el mejor curso de acción en condiciones de incertidumbre mediante la identificación, análisis, evaluación, tratamiento y monitoreo del riesgo, de tal forma que permita minimizar amenazas, gestionar los recursos, continuar activamente las actividades del ciclo de vida y mejorar la calidad del producto. La IEEE afirma que “esta disciplina es clave para la toma de decisiones eficaces y la comunicación de los resultados dentro de las organizaciones” [44] [45] [46].

En este orden de ideas, esta subsección abarca el proceso de identificación de riesgos, priorización y acciones de tratamiento de los problemas de mayor probabilidad e impacto, junto con sus responsables y recursos necesarios para llevar a cabo dichas acciones.

#### OBJETIVOS

* Identificar los posibles riesgos que pueden surgir en el transcurso del semestre de desarrollo del proyecto
* Describir los principales riesgos encontrados y plantear actividades de mitigación y contingencia de riesgos
* Ejecutar el plan de riesgos y las actividades pertinentes para la disminución del riesgo y la calidad del producto
* Fomentar habilidades de control y contingencia de riesgos en el equipo de trabajo

#### RESPONSABLES

* Principalmente, el equipo de configuración y calidad ya que el rol y las funciones que cumplen es el más apropiado para el monitoreo y revisión de las actividades de tratamiento de riesgos.
* En general el Scrum Team debe tener la cultura y la disposición necesarias para llevar a cabo esta labor ya que requiere de tiempo adicional, que a lo largo se refleja principalmente en la calidad del proceso y del producto.
* El Scrum Master se encargará de la organización de la toma de decisiones basadas en los riesgos identificados.

#### RECURSOS

* Microsoft Excel como herramienta para hacer la identificación y priorización de riesgos, así como las especificaciones y controles de aquellos más críticos.

#### DESCRIPCIÓN Y APLICACIÓN DEL PROCESO DE ADMINISTRACIÓN DE RIESGOS

El proceso de administración de riesgos consiste en cinco fases como se puede ver en la ilustración 15, las cuales se explicarán a continuación. Ya que durante el desarrollo del proyecto se encuentran nuevos riesgos, este proceso se realiza en cada sprint y así el repositorio de riesgos se mantiene actualizado.

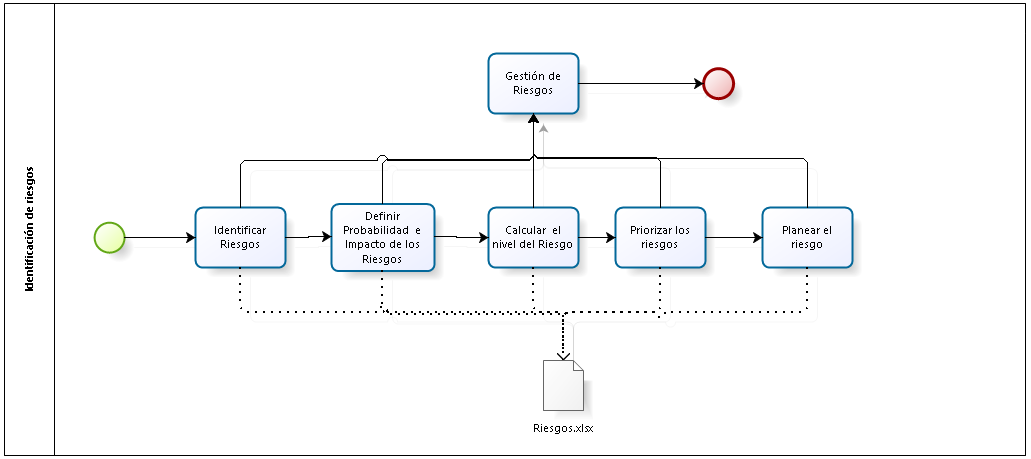


Ilustración 15. BPMN Riesgos

#### Identificar Riesgos:

La identificación del riesgo se define como las actividades y métodos a usar para descubrir los riesgos [47]. A continuación, algunas actividades y métodos que contiene:

1. Aplicar una combinación de técnicas de identificación que se refieren a todos los aspectos del proyecto de software, en particular aquellas áreas que son críticas para el cumplimiento de los criterios de éxito del proyecto y los que son obligatorios.
2. Conversar con equipos de proyecto de software que tengas actividades similares o tengan experiencia para identificar un riesgo.
3. Asegurarse que las descripciones de los riesgos muestran claramente su origen, y tengan los contactos de los expertos en la materia para entender totalmente el detalle.
4. Documentar los supuestos y el contexto que pueden producir los riesgos. [44]

Para el presente proyecto, la identificación de riesgos puede encontrarla en el anexo de Riesgos.

Ver anexo: Riesgos.

#### Definir Probabilidad e Impacto de los Riesgos

Tras la definición de los riesgos, se debe asignar atributos a cada riesgo. Se le debe asignar el impacto y la probabilidad de ocurrencia. Estimar la probabilidad puede ser una práctica difícil. Afortunadamente, es suficiente para estimar la probabilidad el uso de una escala relativa como la que se muestra en la Tabla 12.2.1:

|  |  |  |
| --- | --- | --- |
| Etiqueta | Probabilidad | Descripción |
| Muy bajo | 0% - 20% | Muy improbable del que riesgo ocurra |
| Bajo | 21% - 40% | Improbable del que riesgo ocurra |
| Moderado | 41% - 60% | Posible que ocurra el riesgo |
| Alto | 61% - 85% | Probable del que riesgo ocurra |
| Muy alto | 86% - 100% | Muy probable del que riego ocurra |

Tabla 8. Probabilidades de un riesgo [44]

Los valores para el atributo impacto son los siguientes:

|  |  |
| --- | --- |
| IMPACTO | DESCRIPCIÓN |
| Insignificante | Si se produce el evento de riesgo no habrá efecto alguno. Todos los requerimientos pueden ser satisfechos |
| Menor | Si se produce el evento de riesgo, el programa se encontrará con un aumento bajo en los costos y en el tiempo. Se cumplirán los requisitos mínimos aceptable y la mayoría de los requisitos secundarios pueden ser satisfechos |
| Moderado | Si se produce el evento de riesgo, el programa se encontrará con un aumento moderado en los costos y en el tiempo. Se cumplirán los requisitos mínimos aceptable pero los requisitos secundarios no pueden ser satisfechos |
| Serio | Si se produce el evento de riesgo, el programa se encontrará con grandes aumentos de costos y tiempo. Se cumplirán los requisitos mínimos aceptable pero los requisitos secundarios no pueden ser satisfechos |
| Crítico | Si se produce el evento de riesgo, el programa fallará. No se cumplen los requisitos mínimos aceptables |

Tabla 9. Impacto de un riesgo [44]

#### Calcular el nivel del riesgo

Para calcular el nivel del riesgo se usa la matriz de tolerancia donde se muestra claramente cuando un riesgo es de tolerancia baja, media o alta dependiendo de su probabilidad e impacto.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TOLERANCIA** | | **Impacto** | | | | |
| **Insignificante** | **Menor** | **Moderado** | **Serio** | **Crítico** |
| **Probabilidad** | **Muy alto** |  |  |  |  |  |
| **Alto** |  |  |  |  |  |
| **Moderado** |  |  |  |  |  |
| **Bajo** |  |  |  |  |  |
| **Muy bajo** |  |  |  |  |  |

|  |  |
| --- | --- |
|  | BAJO |
|  | MEDIO |
|  | ALTO |

Tabla 10. Matriz de riesgos [44]

#### Priorizar los riesgos

En esta fase se ordenan los riesgos en cuanto a su tolerancia. Después todo el equipo realiza la priorización de los riesgos haciendo un top 10 con los riesgos menos tolerables.

Para el presente proyecto, la priorización de riesgos puede encontrarla en el anexo de Riesgos en la hoja llamada Top 10.

Ver anexo: Riesgos.

#### Planear el riesgo

El plan de riesgo se define como las actividades y métodos para desarrollar alternativas para la resolución de riesgo [47]. Para esto se llena la siguiente plantilla por cada riesgo del top 10.

|  |  |  |  |
| --- | --- | --- | --- |
| HOJA DE INFORMACIÓN DE RIESGO | | | |
| ID Riesgo: | **Fecha:** | **Probabilidad:** | **Impacto:** |
| Descripción: Breve descripción del riesgo | | | |
| Plan de prevención: Estrategias para disminuir la probabilidad de que el riesgo suceda [47] | | | |
| Plan de mitigación: Estrategias para disminuir el impacto que puede generar el riesgo una vez ocurra [47] | | | |
| Plan de contingencia: Asumiendo que los esfuerzos de reducción han fracasado y que el riesgo se ha convertido en una realidad, que se hará para responder rápidamente al impacto que causará en el proyecto [48] | | | |
| Recursos: Recursos necesarios para ejecutar los planes anteriores. | | | |
| Responsables: Roles responsables de realizar el plan de prevención, mitigación y contingencia en caso de ser necesario. | | | |

Tabla 11. Hoja de Información de Riesgo

La planeación y acciones de tratamiento de los anteriores riesgos puede encontrarla en el anexo de Riesgos.

Ver anexo: Riesgos.

#### Gestión de Riesgos:

Esta actividad es paralela a todas las demás actividades, ya que el documento Riesgos.xlsx se ésta actualizando constantemente. En caso de que haya un riesgo que se convirtió en problema, que se presentó, se recurrirá al documento y se mirará los planes de contingencia del riesgo para así aplicarlos. También se estará evaluando los riesgos menos tolerables y así poder aplicar los planes de prevención y mitigación para reducir el impacto y la probabilidad de los riesgos.

## 12.3 ADMINISTRACIÓN DE CONFIGURACIÓN Y DOCUMENTACIÓN

### OBJETIVOS

* Definir una línea base que permita la coordinación del trabajo
* Definir el formato a manejar para la administración de los ítems de configuración
* Facilitar el trabajo concurrente y coordinarlo de forma que se garantice su consistencia
* Especificar los documentos a entregar facilitando su comprensión a través de la documentación
* Definir los formatos para llevar a cabo una documentación uniforme y entendible [49]

### RESPONSABLES

* Scrum Team de configuración y calidad, realizan el control de versiones. Además, verifican que los cambios al código y a los documentos sean documentados como corresponde
* Scrum Team de análisis y documentación, responsables de verificar que todo documento sea entregado con los formatos descritos en esta sección
* Scrum Team de desarrollo, que gestionan y coordinan los cambios aceptados a realizar en el software

### HERRAMIENTAS

* Configuración: Git, OneDrive, Microsoft Word y Microsoft Excel (ver sección 8.2)
* Documentación: Microsoft Word, Enterprise Arquitect, Bizagi (ver sección 8.2)

Adicionalmente, se emplearán plantillas que a lo largo de la sección se mostrarán junto con su uso. Cabe resaltar que en el transcurso del proyecto se irá enriqueciendo está sección con otras herramientas que, por la etapa actual del proyecto, aún no han sido definidas.

### Ítems de configuración:

A continuación, se describirán cada uno de los ítems de configuración identificados en el proyecto:

* SPMP: El Plan de Gestión de Proyectos de software (SPMP) regido por el estándar ISO/IEC/IEEE 16326-2009 [50] contiene la panificación de cada uno de los procesos a llevar a cabo en el desarrollo de un proyecto de software. Hace parte de este un conjunto de planes que al ser parte del SPMP también estarán sometidos a control de cambios y versiones.
* SRS: La Especificación de Requerimientos de Software (SRS) regido por el estándar IEEE 830-1998 [51] es la descripción del sistema de software a desarrollar, estableciendo los requerimientos funcionales y no funcionales, dando así las bases para un acuerdo entre los clientes y los contratistas.
* SDD: La Descripción del Diseño de Software (SDD) regido por el estándar IEEE 1016-2009 [52] muestra detalladamente el diseño de un producto de software y su arquitectura, esto con el fin de conocer todos los componentes importantes en el desarrollo e implementación del sistema.
* Requerimientos funcionales: Los requerimientos funcionales son todas las características con las que debe cumplir un producto de software, así, es una de las maneras de verificar que el sistema cumpla con lo que se pactó con el cliente.
* Manual de usuario e instalación: Son documentos técnicos mediante los cuales se le brinda asistencia al usuario guiándolo en los procesos de instalación y uso del sistema.
* Planes de pruebas: En el plan de pruebas se especifica de manera minuciosa y detallada las funciones a probar del sistema, además de cómo serán ejecutadas esas pruebas y quienes serán los responsables.
* Código fuente: Es el conjunto de líneas de texto en el cual está escrito mediante el uso de un lenguaje de programación el funcionamiento del sistema.
* Casos de uso: Es el conjunto de secuencias de acciones que relacionan y describen la interacción entre el sistema y los actores que intervienen en este. En este documento se describe mediante los casos de uso todo el funcionamiento del sistema.

### Versionamiento

Tanto el código como la documentación se versionará mediante el siguiente formato, dependiendo si es línea base o no:

1. Para documentos Línea base:

“Nombre del ítem o documento” + LíneaBase + v”X”.”Y”.”Z”

1. Documentos internos que se integraran a una línea Base:

“Nombre del ítem o documento” + v”X”.”Y”.”Z”

El formato anterior muestra:

* Nombre del ítem o documento: Es el nombre con el cual se identifica el documento o archivo.
* X: Numero que indica la cantidad de veces que dicho ítem o documento ha sido revisado por el cliente.
* Y: Número que indica el número de cambios significativos que han sido realizados sobre el ítem o documento. Un cambio significativo tiene que ver por ejemplo con un cambio en el contenido.
* Z: Numero que indica el número de correcciones leves por ejemplo correcciones de ortografía, referencias, redacción, formato, etc.

### Proceso de control de cambios:

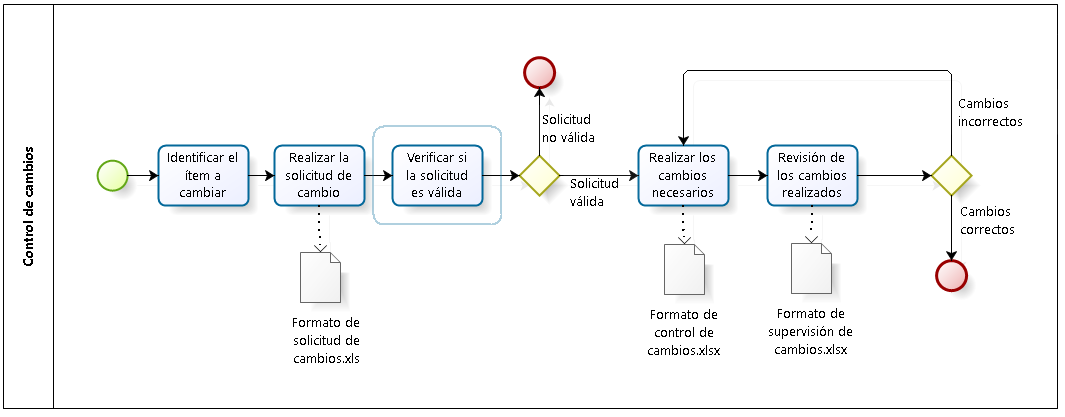


Ilustración 16. BPMN Control de cambios

El proceso de control de cambios se le hará únicamente a los documentos o archivos que sean línea base, es decir elementos de configuración que han sido revisados, entregados y aprobados por el cliente, y que pueden ser cambiados sólo a través de una solicitud de cambio [53]. Para llevar a cabo este proceso se usará una serie de formatos que han sido adaptados para nuestro proyecto [54], debe existir un formato de cada tipo por cada documento que sea línea base.

A continuación, se describe paso a paso el proceso de control de cambios ilustrado en el diagrama BPMN Control de cambios:

1. Identificar el ítem sobre el que se realizará el cambio ya sea por un error en el trabajo realizado o porque se le quiere añadir alguna nueva característica [53].
2. Realizar una solicitud de cambio. El autor de la solicitud de cambio especifica el ítem que quiere cambiar, su versión, el problema a ser resuelto y propone una solución [53]. Esta petición se hará mediante el Formato de solicitud de cambios [8].

Ver anexo: Formato de Solicitud de Cambios.

1. Después de enviar una petición de cambio, ésta se verifica para asegurarse de que sea válida [8], si la petición de cambio es válida se procede a solucionar el problema.
2. Cuando se hayan hecho las modificaciones necesarias se deberá llenar el Formato de control de cambios con el fin de llevar el control de cada uno de ellos [55] [54].

Ver anexo: Formato de Control de Cambios.

1. Mediante el Formato de supervisión de cambios el equipo de configuración y calidad realizara la revisión del cambio realizado para determinar si cumple con el objetivo planteado en la solicitud del cambio [55] [54].

Ver anexo: Formato de Supervisión de Cambios.

1. Finalmente, si el cambio realizado solucionó el problema inicial se procederá a almacenarlo en el repositorio del proyecto, sino los responsables del cambio deberán seguir trabajando en ello. Si no se corrigen los problemas de manera oportuna los responsables serán sancionados de acuerdo a las reglas del grupo de trabajo.

El control de cambios y revisión de los elementos preliminares a las líneas base, es decir documentos que se crean y se trabajan antes de unirse y convertirse en la estructura final en la que se basará el todo el proyecto, se llevará a cabo mediante revisión por pares: (ver sección 12.4)

### Formato de Documentación

Para que haya consistencia entre los distintos documentos que se entregan a lo largo del semestre, se especificaron las fuentes, tamaño, color y etiquetas que el equipo debe seguir en el desarrollo de documentos. Estas especificaciones se encuentran en el anexo Plantilla Para Documentos. Todos los documentos serán redactados en idioma español, bajo las normas de gramática y ortografía definidas por la Real Academia Española.

En los documentos se incluyen anexos, para facilitar la búsqueda de los archivos anexos, estos serán nombrados anteponiendo la sección a la que hace parte y luego su nombre como se muestra a continuación:

*X.Y.Z Nombre del archivo*

Siendo X, Y, Z las secciones y subsecciones a las que hace parte dicho anexo.

Para el almacenamiento de los archivos en esta primera etapa se tendrá una carpeta compuesta exclusivamente por los anexos y otra carpeta con las actas del semestre.

Por otra parte, para el desarrollo de código se consideró también necesario establecer formatos que permitan la homogeneidad de este y facilitar la comprensión del código.

**PHP**

El portal *Pear* define en una sección las distintas buenas prácticas y otras especificaciones para la documentación de código en PHP (a partir de la propuesta de [56]

* Utilizar identación de 4 espacios
* Mantener líneas de aproximadamente 75-85 caracteres de longitud para mejor legibilidad
* Las asignaciones deben estar separadas por espacio *Ejemplo:* var  =  foo();

|  |  |
| --- | --- |
| Aspecto | Ejemplo |
| Definición de clases  Las declaraciones de clase tienen su llave de apertura en una nueva línea. El nombre de esta se escribirá usando UperCamelCase. | <?php class Foo\_Bar {     //... code goes here  } ?> |
| Definición de funciones  El corchete de apertura tiene su propia línea, pero no sucede lo mismo para sentencias de control dentro de las funciones, las cuales tendrán su corchete de entrada al final de la línea. Los corchetes de salida siempre tendrán su propia línea. | <?php function fooFunction($arg1, $arg2 = '') {     if (condition) {         statement;     }     return $val; } ?> |
| Definición de atributos se emplea lowerCamelCase | private intraTribuX |
| Llamadas a funciones  Llamadas sin espacios entre el nombre de la función, el paréntesis de apertura, y el primer parámetro; espacios entre las comas y cada parámetro, y no hay espacio entre el último parámetro, el paréntesis de cierre, y el punto y coma. | <?php $var = foo($bar, $baz, $quux); ?> |
| Condicionales  Un espacio entre la palabra de control y el paréntesis de apertura.  Usar siempre corchetes, incluso en situaciones en que es opcional. Ello reduce la probabilidad de errores y aumenta la legibilidad. | <?php if ((condition1) || (condition2)) {     action1; } elseif ((condition3) && (condition4)) {     action2; } else {     defaultaction; } ?> |
| Comentarios iniciales  Son los comentarios y otras especificaciones que se definen al inicio del archivo. | /\*\*  \* Short description for file  \*  \* PHP version 5  \*   \* @package    PackageName  \* @author     Original Author   \* @author     Another Author   \* @version    SVN: $Id$  \* @date Last revision   \*/ |
| Comentarios para otras estructuras. Puede componerse de atributos. | /\*  \* This is a "Block Comment."  The format  \* is the same as docblock Comments   \* except there is only one sterisk at   \* the top.   \*/ |

Tabla 12 Formato del código

**JavaScript**

Aunque el lenguaje de programación es diferente, se mantienen los mismos principios que se tienen con PHP para garantizar la homogeneidad. Para más información consulte [Google JavaScript Style Guide](https://google.github.io/styleguide/javascriptguide.xml).

### Artefactos de Documentación

|  |  |
| --- | --- |
| Artefacto | Creación o refinamiento |
| SPMP | Es el primer documento que se crea ya que es la base para el desarrollo del proyecto por lo que nace al inicio de este. Al estar compuesto de la planeación (responsables, calendario, herramientas, etc.), se refina constantemente y se actualiza hasta el final del proyecto ya que se pueden ir agregando componentes que debido a la naturaleza del proyecto y al ciclo de vida elegido, aún no han sido definidos. |
| SRS | Se crea en el segundo sprint y se enriquece también en las etapas posteriores puesto que puede que al final del proyecto, según el tiempo disponible, se modifique el alcance. |
| SDD | Se crea en el tercer sprint y debe estar en constante cambio ya que conforme se agregan, modifican o eliminan funcionalidades y requerimientos se ve afectado el diseño. |
| Manual de usuario | Se crea al tener el primer prototipo y se refina conforme se van realizando mejoras en el software final. |
| Manual de instalación | Se crea al tener el primer prototipo y se refina conforme se van realizando mejoras en el software final. |

Tabla 13. Documentación – Administración de configuración

## 12.4 MÉTRICAS Y PROCESO DE MEDICIÓN

#### OBJETIVO

Su propósito es medir la calidad del producto, mediante métricas en cada Sprint / iteración del Scrum en base a algunas clasificaciones factor que se puede utilizar para obtener una visión global de las pruebas de calidad de la prueba. [57]

Se realizarán revisiones por parejas. Esta revisión se realizará dentro de cada grupo de trabajo antes de acabar el Sprint para presentarlo los días martes en la revisión general.

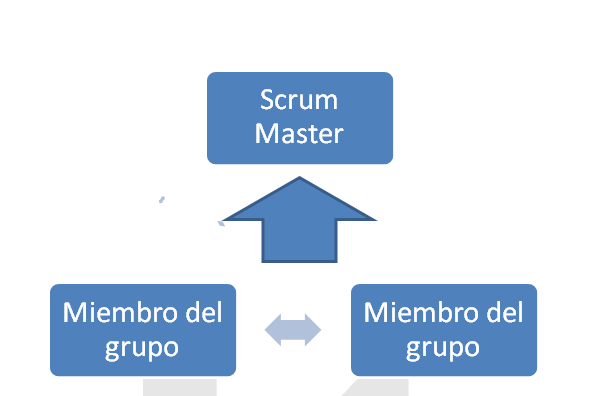


Ilustración 17. Pares por grupo

Cada miembro del grupo al realizar un producto se lo muestra a su pareja de área para que este lo revise y le haga las respectivas correcciones, antes de terminar el Sprint. Luego al momento de la reunión de los martes se reúnen con el Scrum Master para la revisión formal.

Plantilla de calidad: En el siguiente documento se puede visualizar una plantilla con la cual se evaluará la calidad de los productos entregados por cada uno de los miembros, y donde se dirá si un documento aprobó la revisión de calidad.

Ver anexo: Plantilla Calidad de Documentos.

#### DESARROLLO

La medición se hará de forma cualitativa o cuantitativa teniendo en cuenta los diferentes procesos. Para ello se hará énfasis en medir:

* Desarrollo y gestión de la solución técnica.
* Gestión de proyecto.
* Gestión de la organización. [58]

#### DOCUMENTOS

Se usará una plantilla Plantilla Para Documentos donde está especificado el formato con el que el grupo The Walking Dev trabajara para todo el proyecto, además se tendrán lo estipulado en la sección 12.3 Administración de Configuración y Documentación.

Para medir la calidad de un documento se tendrá en cuenta los siguientes aspectos para verificar que si cumple con los requisitos y puede ser entregado al cliente.

* ¿La idea principal del documento está clara?
* ¿Hay coherencia entre el texto y el tema?
* ¿Se hace un correcto uso de la plantilla?
* ¿Se aplica correctamente el formato de referencias en el documento?
* ¿El texto está escrito con coherencia?
* ¿Cumple con todos los requisitos del plan de control de cambios?

Plantilla de calidad: En el siguiente documento se puede visualizar una plantilla con la cual se evaluará la calidad de los productos entregados por cada uno de los miembros, y donde se dirá si un documento aprobó la revisión de calidad y contendrá todos los requisitos anteriores, Ver Anexo: Plantilla Calidad de Documentos.

La calificación que se le asignará a un documento en el proceso de calidad, será una calificación de si o no; y consistirá en si cumple con los requisitos descritos en (ver sección: [12.5 Control de Calidad](#_12.5_CONTROL_DE)) Ésta calificación permitirá controlar y modificar los estados de todos los documentos entregados permitiendo realizar las respectivas correcciones hasta que el documento pase el proceso de calidad.

#### TIEMPO

El tiempo es un factor muy importante cuando se está haciendo un proyecto, debido a esto The Walking Dev hará medición del trabajo realizado por cada miembro del grupo. Para observar si el tiempo que cada miembro le está dedicando al desarrollo del proyecto es equivalente al trabajo realizado de acuerdo a la dificultad de cada tarea en el Sprint se implementara un formato, Ver Anexo: Sprint Backlog.

Para contrastar si el porcentaje de trabajo realizado es coherente con el tiempo invertido por el miembro del grupo se incluirá una medida de páginas por tiempo. Esta revisión se hará con cada uno de los miembros del grupo en las reuniones de los Sprints para ver si se está cumpliendo con esta métrica en base a la plantilla del SPMP donde se especifica cuantas páginas promedio tiene cada producto [57].

#### CÓDIGO

The Walking Dev ha adoptado por métricas estándar de eficiencia de Ingeniería del Software, donde por medio de una medición de cuantas líneas de código se escriben por hora contra el número de errores al compilar nos da el rendimiento y así poder medir la eficiencia de los programas con la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| LINEAS DE CÓDIGO x HORA | ERRORES | RENDIMIENTO |
|  |  |  |

Tabla 14. Calificación del software

## 12.5 CONTROL DE CALIDAD

La gestión de aseguramiento de calidad se encarga que todos los entregables y las actividades relacionadas con los procesos definidos sean revisadas y auditadas para verificar que cumplen con los estándares y las líneas guías adoptadas por Walking Dev. [53]

La calidad del software se puede medir con alguna de las siguientes características de software: [8]

* Fiabilidad: Grado en que un programa satisface las especificaciones definidas por el cliente.
* Portabilidad: Esfuerzo requerido para transferir el programa desde un sistema hardware/software a otro.
* Eficiencia: Cantidad de recursos de computadora y de código requeridos por un programa para llevar a cabo sus funciones.
* Ingeniería humana: Grado en que el software facilita la funcionalidad y el rendimiento asignados al elemento humano.
* Facilidad de prueba: Esfuerzo requerido para probar un programa de forma que se asegure que se realiza su función requerida.
* Facilidad de uso: Esfuerzo requerido para aprender y trabajar con un programa, así como preparar la entrada e interpretar la salida de un programa.
* Facilidad de mantenimiento: Esfuerzo requerido para localizar y reparar un error en un programa.

En cada Sprint, se garantizará la calidad para que cada entrega cumpla con los estándares ya estipulados.

#### OBJETIVO GENERAL

Garantizar la calidad de cada uno de los productos realizados durante los Sprint. [5]

#### CALIDAD DE LOS DOCUMENTOS

Para garantizar la calidad de los documentos, se realizará un proceso de calidad entre el Scrum Team y el Scrum Master, donde se tiene como base para la revisión los estándares especificados en la sección 12.3 Administración de configuración y documentación.

Cada uno de los grupos subirá los documentos al repositorio, después se hará una revisión de calidad con las siguientes métricas:

* Estructura del documento: Teniendo en cuenta las plantillas SPMP, SRS y SDD.
* Ortografía.
* Fuentes bibliográficas.
* Redacción.
* Formato de las referencias: Las referencias deben estar en formato IEEE.

Si el documento presentado no cumple con las métricas anteriores, se devuelve el documento al grupo con los comentarios correspondientes a modificar, y si por el contrario el documento cumple con todos los parámetros establecidos, se aprueba el documento y queda listo para anexarlo al documento final.

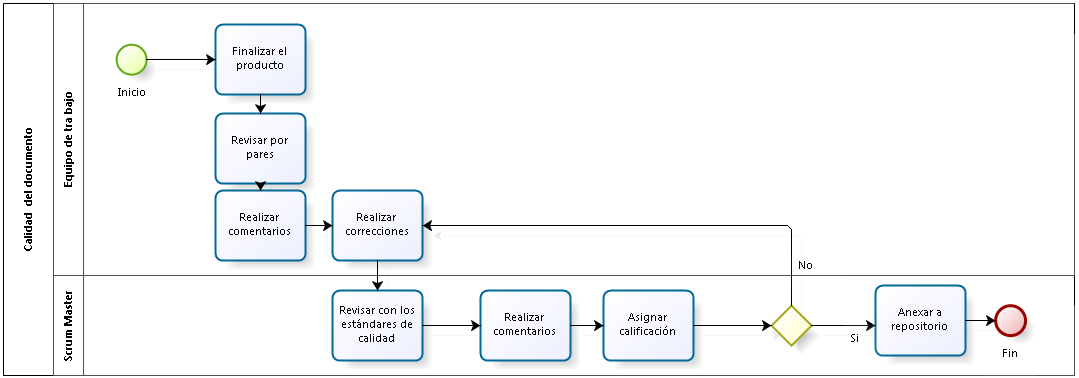


Ilustración 18. Calidad de los documentos.

#### CALIDAD DE LOS CÓDIGOS

Para garantizar la calidad del código se realizará un proceso de revisión con el desarrollador y el Scrum Team.

El desarrollador subirá su código al repositorio. Para la revisión de calidad del código, el Scrum Team tendrá en cuenta los siguientes requerimientos, Ver Anexo: Plantilla Calidad de Documentos.

* Coherencia entre el diseño y el código.
* Buenas prácticas de programación.
* Documentación del código.

El Scrum Master revisará el código y si cumple con las especificaciones pasará a la fase de evaluación y quedará listo para anexarlo al documento final, si no se le hará las respectivas correcciones.

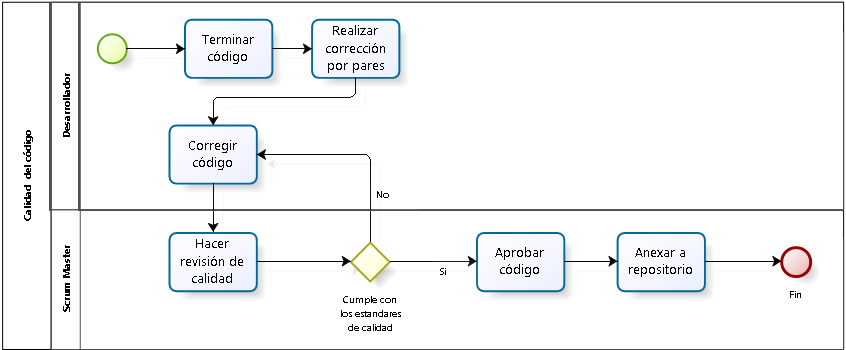


Ilustración 19. Calidad de los códigos

#### CALIDAD DE LOS DISEÑOS

Para asegurar la calidad del diseño se debe llevar a cabo un proceso de revisión que involucra al desarrollador y a los Scrum Team.

Se deberán revisar las siguientes métricas:

* Patrones de diseño. [59]
* Coherencia con los requerimientos.

El Scrum Master debe garantizar que los diagramas realizados tengan los estándares para el lenguaje UML [53]. Si el diagrama no cumple con estos estándares el diagrama será revisado para su corrección. Si el diagrama no presenta errores, el Scrum Master lo aprobará para anexarlo al documento final.

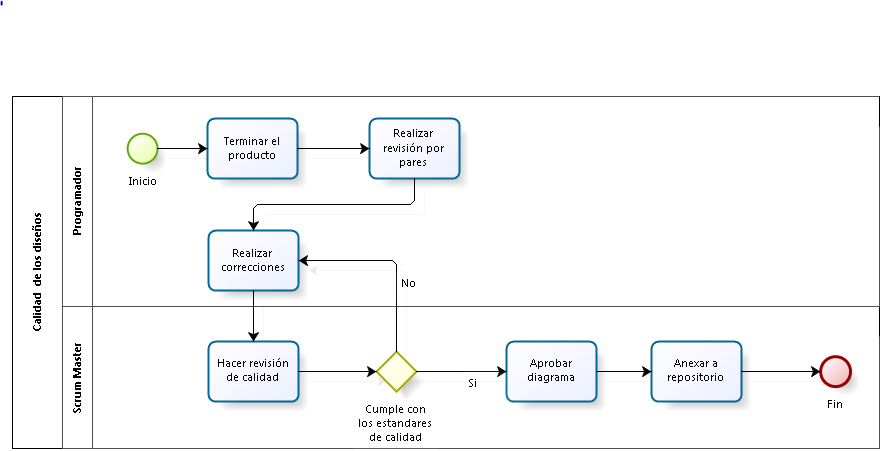


Ilustración 20. Calidad de los diseños

#### PROGRESO DEL PROYECTO

Se realizará una revisión constante del documento y cada una de las diferentes actividades según el calendario. El documento final deberá contar con cada una de las diferentes partes ya corregidas y con los diferentes procesos de calidad aprobados por el Scrum Master y luego será anexado al documento final.

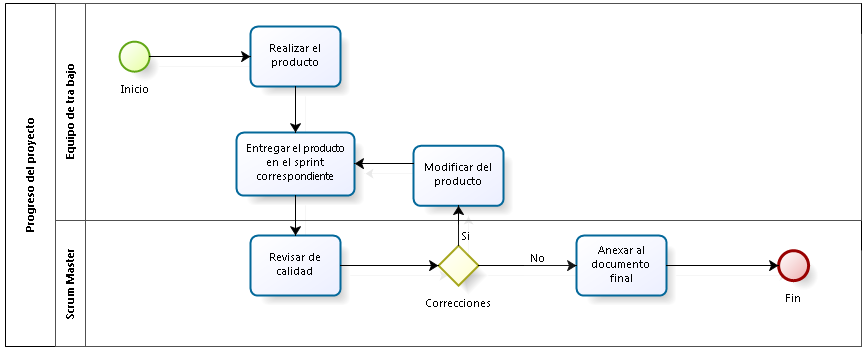


Ilustración 21. Progreso del proyecto

#### CONTROL DE CALIDAD

En la siguiente tabla se podrá observar un resumen más claro sobre el proceso que se va a llevar a cabo y en qué momento se llevará a cabo.

|  |  |  |  |
| --- | --- | --- | --- |
| PROCESO | MOMENTO DEL PROYECTO EN QUE OCURRE | RESPONSABLE | HERRAMIENTA |
| Calidad de documentos | Se realiza durante todo el proyecto en cada Sprint. | Scrum Master y Scrum Team | * Plantillas. * Microsoft Word. * Excel. |
| Calidad de software | Se realizará durante el tercer Sprint. Cuando este realizado el producto y se revise para poder anexarlo al documento final. | Scrum Master y Scrum Team de programación | * Github. * Paginación * Color * Tipo de letra * Indexación * Justificación * Sangría |
| Calidad de diseño | Se realiza durante toda la segunda y tercera iteración. | Scrum Master y Scrum Team de programación | * Bizagi Modeler. |
| Progreso del proyecto | Se debe realizar en cada sprint, que según el calendario cada martes todas las semanas. | Scrum Master | * Excel. |

Tabla 15. Control de calidad

# BIBLIOGRAFÍA

[1] M. Torres, «Blackboard Pontificia Universidad Javeriana - Reglas de clase,» [En línea]. Available: http://uvirtual.javeriana.edu.co/webapps/portal/frameset.jsp?tab\_group=courses&url=%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2Ffile%3Fcmd%3Dview%26content\_id%3D\_382553\_1%26course\_id%3D\_8388\_1%26framesetWrapped%3Dtrue.

[2] S. H. Bautista, «Pegasus - Pontificia Universidad Javeriana,» 2012. [En línea]. Available: http://pegasus.javeriana.edu.co/~CIS1230IS09/Documentos/SPMP.pdf.

[3] C. Larman, UML y patrones una introducción al análisis y diseño orientado a objetos y al proceso unificado, Pearson Educación, 2003.

[4] Ministerio de educación, «VINCULACIÓN LABORAL DE RECIÉN GRADUADOS,» Ministerio de educación, [En línea]. Available: http://bi.mineducacion.gov.co:8380/eportal/web/men-observatorio-laboral/taza-de-cotizacion-por-programas;jsessionid=0BB48B1FCF599E0BD577781F40649064 .

[5] «Plantilla SPMP,» [En línea]. Available: http://uvirtual.javeriana.edu.co/webapps/portal/frameset.jsp?tab\_group=courses&url=%2Fwebapps%2Fblackboard%2Fexecute%2Fcontent%2Ffile%3Fcmd%3Dview%26content\_id%3D\_382553\_1%26course\_id%3D\_8388\_1%26framesetWrapped%3Dtrue.

[6] 40 de fiebre, «40 de fiebre,» [En línea]. Available: http://www.40defiebre.com/que-es/diseno-responsive/.

[7] K. Schwaber y . J. Sutherland, «Scrum Guides,» 2013. [En línea]. Available: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf.

[8] I. Sommerville, Ingeniería de Software, Naucalpan de Juárez, Estado de México: Pearson, 2011.

[9] «Significados,» [En línea]. Available: http://www.significados.com/stakeholder/.

[10] C. Larman, Agile and iterative development a manager's guide, Boston, Massachusetts: Addison-Wesley, 2004.

[11] K. S. Rubin, Essential Scrum a practical guide to the most popular agile process, Addison-Wesley, 2012.

[12] A. G. NMA Munassar, «A Comparison Between Five Models of Software Engineering,» *IJCSI International Journal of Computer Science Issues,* vol. 7, nº 5, 2010.

[13] N. B. Ruparelia, «Software development lifecycle models,» *ACM SIGSOFT Software Engineering Notes,* vol. 35, nº 3, pp. 8-13, 2010.

[14] B. Boehm, «Spiral Development: Experience, Principles, and Refinements,» 2010.

[15] R. K. S. K. L. Harikrishnan Natarajan, «A Comparison Between Present and Future Models Of Software Engineering,» *IJCSI International Journal of Computer Science Issues,* vol. 10, nº 2, 2013.

[16] D. C. J. Shikha maheshwari, «A comparative study of software development life cycle models,» *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 2, nº 5, 2012.

[17] FACULTAD DE INGENIERIA , «UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA,» [En línea]. Available: http://yaqui.mxl.uabc.mx/~molguin/as/RUP\_files/image003.jpg .

[18] J. Mohedano, J. M. Saiz y P. Salazar Román, Iniciación a javascript, Ministerio de Educación de España, 2012.

[19] V. Vaswani, Fundamentos de PHP, McGraw-Hill Interamericana, 2010.

[20] Trello, Inc., «Trello,» [En línea]. Available: https://trello.com. [Último acceso: 20 Febrero 2016].

[21] Microsoft, «Project,» [En línea]. Available: https://products.office.com/es/Project/project-and-portfolio-management-software. [Último acceso: 28 Febrero 2016].

[22] R. C. Lee, «The Success Factors of Running Scrum: A Qualitative Perspective,» 2012.

[23] L. Rising y N. S. Janoff, «The Scrum Software Development Process For Small Teams,» 2000.

[24] Instituto Geográfico Agustín Codazzi, «Rol: Analista Funcional,» [En línea]. Available: http://geoservice.igac.gov.co/mds/igac/ciaf/roles/analista\_igac.html.

[25] D. F. Padilla, «Universidad Catolica de Maule,» 2003. [En línea]. Available: http://www.eici.ucm.cl/Academicos/R\_Villarroel/descargas/ing\_sw\_1/Roles\_desarrollo\_software.pdf.

[26] Y. V. Lorenzo, D. M. Sanchidrian y L. A. Pérez, «Universidad de las Ciencias Informáticas,» [En línea]. Available: http://publicaciones.uci.cu/index.php/SC/article/download/89/88.

[27] ISO, «ISO/IEC 25001:2014(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Planning and management,» [En línea]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:25001:ed-2:v1:en.

[28] IEEE, «IEEE Standard for Configuration Management in Systems and Software Engineering,» *IEEE-SA Standards Board,* 2012.

[29] M. Perez, « «Roles y responsabilidades en un equipo de desarrollo de software»,» 2015.

[30] E. B. Evita Coelho, «Effort Estimation in Agile Software Development using Story Points,» *International Journal of Applied Information Systems,* vol. 3, 2012.

[31] M. Cohn, Agile Estimating and Planning, Pearson , 2006.

[32] Google, «Play Store,» [En línea]. Available: https://play.google.com/store/apps/details?id=com.leanify.planning.poker&hl=es\_419.

[33] H. Kerzner, Project Management Proffessional Practical Guide, John Wiley & Sons, Inc., 1998.

[34] U.S. Department of Energy, «Work Breakdown,» *Office of Engineering and Construction Management,* 2003.

[35] OBS Business School, «OBS Business School,» 2014. [En línea]. Available: http://www.obs-edu.com/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve/.

[36] Hansoft, «Scrum, Kanban, and Gantt scheduling in one project – ¿nonsense or necessity?,» 2014.

[37] IEEE, «1998-1490 IEEE guide - adoption of PMI standard - a guide to the project management body of knowledge,» 1998.

[38] M. L. S. y. J. Erwin, «Role & Responsibility,» 2000.

[39] Project Management Prepcast, «PMP® Exam Tip: The Responsibility Assignment Matrix (RAM),» [En línea]. Available: https://www.project-management-prepcast.com/free/pmp-exam/tips/303-pmp-exam-tip-the-responsibility-assignment-matrix-ram.

[40] Space and Naval Warfare Systems Center San Diego, «Peer Review Process,» 20 Noviembre 2002. [En línea]. Available: http://uvirtual.javeriana.edu.co/bbcswebdav/pid-382550-dt-content-rid-1584187\_1/courses/004070\_1610\_2415/PeerReview\_Process.pdf.

[41] O. Lawson, «The importance and remarkable effects of a positive work environment in business,» 2012.

[42] TED, «TED,» [En línea]. Available: http://www.ted.com/talks/margaret\_heffernan\_why\_it\_s\_time\_to\_forget\_the\_pecking\_order\_at\_work#t-803527.

[43] M. Heffernan, The Surprising Element That Will Make Your Company Thrive, 2015.

[44] L. S. C. W. y. F. S. Li Xiaosong, «he Application of Risk Matrix to Software project Risk management,» *IEEE,* 2009.

[45] IEEE, «Systems and software engineering — Life cycle processes — Risk management,» *Software & Systems Engineering Standards Committee,* 2016.

[46] H.-P. Berg, «RISK MANAGEMENT: PROCEDURES, METHODS AND EXPERIENCES,» 2010.

[47] E. M. Hall, Managing risk methods for software systems development, Addison-Wesley, 1998.

[48] R. S. Pressman, Ingeniería del software un enfoque práctico, Mcgraw Hill/Interamericana Editores, 2010.

[49] IEEE, «IEEE Standard for Configuration,» 2012.

[50] IEEE (Institute of Electrical and Electronics Engineers), *16326-2009 ISO/IEC/IEEE International Standard Systems and Software Engineering--Life Cycle Processes--Project Management,* IEEE-SA Standards Board, 2009.

[51] IEEE (Institute of Electrical and Electronics Engineers), *830-1998 IEEE Recommended Practice for Software Requirements Specificacitions,* IEEE-SA Standards Board, 1998.

[52] IEEE (Institute of Electrical and Electronics Engineers), *1016-2009 IEEE Standard for Information Technology--Systems Design--Software Design Descriptions,* IEEE-SA Standards Board, 2009.

[53] B. Brugge y A. H. Dutoit, Prentice Object Oriented Software Engineering Using UML Patterns and Java, Prentice Hall, 2012.

[54] NASA, «Software and Hardware Configuration Management,» 9 Julio 2014. [En línea]. Available: http://www.nasa.gov/sites/default/files/ivv\_10\_-\_ver\_i.doc. [Último acceso: 8 marzo 2016].

[55] B. Pierre y R. E. Fairley, *Swebok v3.0 Guide to the Software Engineering Body of Knowledge,* 2014.

[56] Pear, «Naming Conventions Pear,» [En línea]. Available: https://pear.php.net/manual/en/standards.naming.php.

[57] I. Kayes, «Agile Testing: Introducing PRAT as a Metric of Testing Quality in Scrum,» vol. 36, nº 2, pp. 1-5, 2011.

[58] J. Palacio, Gestión de proyectos Scrum Manager, 2014.

[59] sparxsystems, «Patrones UML,» [En línea]. Available: http://www.sparxsystems.com.ar/resources/developers/uml\_patterns.html.

[60] K. Schwaber, Agile Project Management With Scrum, Redmond, Washington: Microsoft Press, 2004.

[61] Pontificia Universidad Javeriana, «Departamento de Ingeniería de Sistemas,» [En línea]. Available: http://ingenieria.javeriana.edu.co/facultad/departamentos/ingenieria-sistemas/profesores.

[62] Pontificia Universidad Javeriana, «Biblioteca General Alfonso Borrero Cabal S.J.,» [En línea]. Available: http://www.javeriana.edu.co/biblos/biblioteca-general.

[63] Pontificia Universidad Javeriana, «Carrera de Ingeniería de Sistemas,» [En línea]. Available: http://www.javeriana.edu.co/carrera-ingenieria-de-sistemas.

[64] S. Wolff, «Scrum Goes Formal: Agile Methods for Safety-Critical Systems,» 2012.

[65] M. T. Gallego, «Quimbiotec - Gestión de proyectos informáticos,» [En línea]. Available: http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf.

[66] J. Palacio, «Navegapolis - El modelo Scrum,» [En línea]. Available: http://www.navegapolis.net/files/s/NST-010\_01.pdf.

[67] Software Engineering Institue, «A Framework for Software Product Line Practice, Version 5.0,» [En línea]. Available: http://www.sei.cmu.edu/productlines/frame\_report/config.man.htm.

[68] IEEE, « IEEE Standard for System and Software Verification and Validation IEEE Computer Society,» *IEEE-SA Standards Board,* 2012.

[69] CxOne, «Life Cycle Model Selection,» *Construx Software,* 2002.

[70] M. Chacon, «User Stories Versus Use Cases,» Scrum Alliance , 2015. [En línea]. Available: https://www.scrumalliance.org/community/articles/2015/october/user-stories-vs-use-cases.

[71] H. A. Amani Mahdi Mohammed Hamed, «Popular Agile Approaches in Software Development: Review and Analysis,» *IEEE,* 2013.

[72] «Zotero,» [En línea]. [Último acceso: 28 Febrero 2016].

[73] WhatsApp Inc., «Whatsapp,» [En línea]. Available: https://www.whatsapp.com/?l=es. [Último acceso: 20 Febrero 2016].

[74] Slack, «Slack,» [En línea]. Available: https://slack.com/is. [Último acceso: 20 Febrero 2016].

[75] «Planning Poker,» [En línea]. Available: https://play.google.com/store/apps/details?id=com.leanify.planning.poker&hl=es\_419. [Último acceso: 5 Marzo 2016].

[76] Microsoft, «OneDrive,» [En línea]. Available: https://onedrive.live.com/about/es-419/. [Último acceso: 20 Febrero 2016].

[77] «Laravel,» [En línea]. Available: https://laravel.com/. [Último acceso: 28 Febrero 2016].

[78] «Digital Ocean,» [En línea]. Available: https://www.digitalocean.com/help/. [Último acceso: 28 Febrero 2016].

[79] «Bizagi,» [En línea]. Available: http://www.bizagi.com/es/. [Último acceso: 28 Febrero 2016].

[80] «Atom,» [En línea]. Available: https://atom.io/. [Último acceso: 28 Febrero 2016].

[81] Adobe Systems Incorporated, «Adobe Illustrator,» [En línea]. Available: http://www.adobe.com/la/products/illustrator.html. [Último acceso: 28 Febrero 2016].

[82] «GitHub,» [En línea]. Available: https://github.com/. [Último acceso: 05 Marzo 2016].

[83] Google, «Play Store,» [En línea]. Available: https://play.google.com/store/apps/details?id=com.mmobile.app.event&hl=es.

[84] Apple, «Itunes Apple,» [En línea]. Available: https://itunes.apple.com/us/app/pro-party-planner/id526140208?mt=8.

[85] C. A. D. Mosquera, «Consultas laborales,» 2016. [En línea]. Available: http://consultas-laborales.com.co/index.php?option=com\_content&view=article&id=441:2015-12-30-00-49-00&catid=1:laboral&Itemid=86.