

## **Información Importante**

La Universidad Santo Tomás, informa que el(los) autor(es) ha(n) autorizado a usuarios internos y externos de la institución a consultar el contenido de este documento a través del Catálogo en línea de la Biblioteca y el Repositorio Institucional en la página Web de la Biblioteca, así como en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

Se permite la consulta a los usuarios interesados en el contenido de este documento, para todos los usos que tengan **finalidad académica**, nunca para usos comerciales, siempre y cuando mediante la correspondiente cita bibliográfica se le dé crédito al trabajo de grado y a su autor.

De conformidad con lo establecido en el Artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, la Universidad Santo Tomás informa que “los derechos morales sobre documento son propiedad de los autores, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.”

**Bibliotecas Bucaramanga**  
**Universidad Santo Tomás**

**Lineamientos para la planeación y estimación de actividades en el marco de la  
Fábrica de Software de la Fundación Cardiovascular de Colombia (FCV) basados en  
metodología SCRUM y PMI.**

**Ángela María Cortes Pabón**

**Trabajo de grado para optar el título de Especialista en Gerencia Comercial de  
proyectos de telecomunicaciones.**

**Director**

**Sergio Andrés Zabala Vargas**

**Magister en Administración de Proyectos**

**Universidad Santo Tomas, Bucaramanga**

**División de Ingenierías y Arquitectura**

**Facultad de Ingeniería de Telecomunicaciones**

**2017**

## Tabla de Contenido

1.	Introducción .....	7
2.	Formulación del problema .....	9
2.1	Problemática.....	9
2.2	Definiciones, acrónimos y abreviaturas .....	11
3.	Justificación del proyecto .....	12
3.1	Alcance.....	13
4.	Objetivos .....	14
4.1	General .....	14
4.2	Específicos .....	14
5.	Marco referencial del proyecto .....	15
5.1	Marco Teórico de la propuesta.....	15
5.1.1	Metodologías de gestión de proyectos. ....	15
5.1.2	Metodologías de proyectos ágiles. ....	17
5.2	Marco Histórico.....	19
5.3	Marco Tecnológico .....	21
5.4	Estado del arte .....	26
6.	Desarrollo del proyecto.....	35
6.1	Plan de las Fases y Etapas del Proyecto (WBS).....	35
Fuente: Autor .....		37
6.2	Descripción general del proceso de la Fábrica de Software.....	38
6.2.2	Planeación del Sprint .....	40
6.3	Diseño de la estrategia de intervención.....	49
6.3.1	Primera propuesta de intervención.....	49
6.3.2	Segunda propuesta de intervención.....	52
Fuente: Autor .....		54
6.3.3	Tercera propuesta de intervención: Ajuste al proceso de la Fábrica de Software e implementación de herramienta Testlink.....	56
6.3.4	Cuarta propuesta de intervención: Implementación de Herramienta SLACK.....	61
6.4	Resultados: Validación de la estrategia y medición del impacto. ....	62
6.5	Lineamientos para el mejoramiento de los procesos de la Fábrica de Software- Caso nuevo proyecto.....	65
6.6	Estrategias propuestas no implementadas del proyecto a la FCV. (Ideas futuro)73	
7.	Conclusiones .....	75
8.	Bibliografía .....	77

## Tabla de Tablas

Tabla 1. Indicador de gestión (Entregables liberados / Entregables planeados).....	9
Tabla 2. Historia de la Gestión de Proyectos .....	20
Tabla 3. Criterios generales de la revisión de estado de arte .....	26
Tabla 4 - Interacción actores - proceso .....	46
Tabla 5. Descripción de indicadores .....	47
Tabla 6. Puntos estimados y puntos realizados por mes por persona .....	48
Tabla 7. Resultados generales meses Junio, Julio y Agosto de 2016 .....	62
Tabla 8. Perfil nuevo integrante propuesto. ....	<b>¡Error! Marcador no definido.</b>

## Tabla de Figuras

Figura 1 . Diagrama de Gantt.....	22
Figura 2. Software Ganttter.....	23
Figura 3. Herramienta IceSrum.....	24
Figura 4. Herramienta Agilo for Trac .....	25
Figura 5. Herramienta Trello .....	26
Figura 6. Propuesta del desarrollo del proyecto presentada en el anteproyecto .....	35
Figura 7. EDT del proyecto.....	37
Figura 8. Proceso Fábrica de Software FCV.....	38
Figura 9. Recepción de solicitudes .....	40
Figura 10. Planeación del Sprint .....	41
Figura 11. Construcción de Entregable.....	42
Figura 12. Verificación del Entregable .....	43
Figura 13- Liberación del Ejecutable.....	44
Figura 14. Socialización.....	45
Figura 15. Puesta en producción .....	46
Figura 16. Indicadores por Sprint Fábrica de Software .....	48
Figura 17. Complemento Ganttter de Google Drive .....	50
Figura 18. Estructura para el ingreso de datos .....	50
Figura 19. Visualización Sprint en Gantt Project .....	51
Figura 20. Estructura tarea en Agilo for Trac del tipo Mejora .....	53
Figura 21. Fechas agregadas a la estructura de la tarea en Agilo for Trac .....	54
Figura 22. Cronograma planeación de sprint .....	55
Figura 23. Seguimiento a cronograma .....	56
Figura 24. Plan de pruebas herramienta Testlink.....	58
Figura 25. Ejecución caso de prueba Testlink .....	59
Figura 26. Ejemplo mensaje herramienta SLACK .....	62
Figura 27. Relación de % de Capacidad por meses (pre y post intervención).....	63
Figura 28. Relación de % de Cumplimiento por meses (pre y post intervención) .....	64

## RESUMEN

El presente documento socializa los resultados de la experiencia de proyecto de monografía titulada *Lineamientos para la planeación y estimación de actividades en el marco de la Fábrica de Software de la Fundación Cardiovascular de Colombia (FCV) basados en metodología SCRUM y PMI*; el cual se desarrolla como requerimiento de grado del programa de Especialización en Gerencia Comercial de Proyectos de Telecomunicaciones de la Universidad Santo Tomás- Seccional Bucaramanga.

Esta iniciativa permitió valorar un estado de arte del uso de metodologías de gestión de proyectos en organizaciones orientadas al desarrollo de software; la determinación de indicadores propios de la dinámica de la Fábrica de Software de la FCV, su medición, la generación de estrategias de intervención para el mejoramiento del desempeño y la validación de la misma. El proyecto propuso cuatro grandes intervenciones, las cuales giraron en torno a cambios metodológicos en el desarrollo de las actividades de la Fábrica, así como la inclusión de herramientas software de control y monitoreo de procesos; logrando un mejoramiento en indicadores de desempeño en varios puntos porcentuales.

**Palabras claves:** Scrum, PMI, gestión de proyectos de software, Herramientas de gestión de proyectos.

## **1. Introducción**

Este documento es la consolidación final del proyecto Lineamientos para la planeación y estimación de actividades en el marco de la Fábrica de Software de la Fundación Cardiovascular de Colombia (FCV) basados en metodología SCRUM y PMI, cuyo objetivo es proponer un conjunto de lineamientos y estrategias en el proceso de planeación y estimación de actividades para la Fábrica de Software de la Fundación Cardiovascular de Colombia (FCV); basados en metodología Scrum y PMI.

Este proyecto fue desarrollado como modalidad de grado de la Especialización en Gerencial Comercial de Proyectos de Telecomunicaciones adscrita a la Facultad de Ingeniería de Telecomunicaciones de la Universidad Santo Tomás-Bucaramanga; donde se aplican los conocimientos adquiridos en las áreas de la gestión de proyectos, en particular de proyectos ágiles. La principal motivación sobre el desarrollo el proyecto es el reconocimiento por parte de la autora, al desempeñarse al momento de la investigación como Analista Funcional de la Fábrica de Desarrollo de Software de la Fundación Cardiovascular de Colombia; donde se evidencia de primera mano las problemáticas al momento del inicio de las actividades del mes donde se listas aquellos requerimientos y no se logra una gestión eficiente del proceso.

El presente documento se desarrolla de la siguiente manera: En los primeros tres capítulos se muestran los elementos generales del anteproyecto, los cuales son la problemática, la justificación, los objetivos y la orientación metodológica. En el capítulo cinco se presenta un marco referencial y estado de arte, que permite al lector familiarizarse con la temática y ubicar conceptualmente el proyecto. Hacia el capítulo seis se presenta el diseño de estrategias y la descripción de lineamientos propios al quehacer de la Fábrica de software de la FCV, destacando los posibles mejoramientos que se puedan realizar.

Finalmente, se comparte un conjunto de conclusiones y recomendaciones para que sean estudiadas y consideradas por el equipo de la FCV; con la intención de lograr posibles implementaciones futuras que aporten en su quehacer como institución.

## 2. Formulación del problema

### 2.1 Problemática

En la planificación y gestión de proyectos, la correcta asignación de recursos (humanos, económicos, materiales, entre otros) es fundamental; así como el seguimiento de los mismos de forma responsable. En el caso particular de la Jefatura Fábrica de Software (Área) de la Fundación Cardiovascular de Colombia (FCV)<sup>1</sup> el proceso de desarrollo se viene ejecutando con metodología SCRUM, la cual se sustenta en el uso de buenas prácticas para el trabajo colaborativo y obteniendo el mejor resultado posible [1].

En este marco se ha encontrado que el desarrollar aplicaciones de forma eficiente requiere de un trabajo grupal organizado, siguiendo metodologías aceptadas para tal fin; sin embargo se han detectado en el equipo de trabajo del área las siguientes dificultades:

- Existen bajos porcentajes de cumplimiento entre las actividades o entregables liberados vs. Los entregables entregados por parte de la fábrica al cierre de mes.

Esto se puede evidenciar en la Tabla 1.

*Tabla 1. Indicador de gestión (Entregables liberados / Entregables planeados)*

Mes	Actividades o entregables planeados	Actividades o entregables liberados	Porcentaje de cumplimiento
Marzo	259	139	53,66 %
Abril	168	104	61,19 %
Mayo	104	65	62,5 %

*Nota:* La relación es obtenida del trabajo de campo realizado en la Fábrica de Software de la FCV.

---

<sup>1</sup> Adscrita a la Dirección de Tecnologías de la Fundación Cardio-Vascular de Colombia.

- El usuario final de los productos de la Fábrica de Software es la Jefatura de Help Desk. Se ha logrado determinar, a través de la observación en los encuentros, que falta mayor claridad y comprensión de los roles y funciones que cada uno debe desempeñar.
- No está debidamente analizada la carga laboral de los involucrados en la fábrica, por cual se llegan a generar sobrecargas laborales. Es importante también definir con claridad el rol del SCRUM MASTER vs. El equipo de recursos humanos de la fundación, como se resalta en [2]
- Existe una falta de socialización sobre puntos de planeación en el equipo de trabajo, lo que genera desconocimiento de las actividades a realizarse en el tiempo estimado por parte de los involucrados.

En síntesis, existen situaciones problemáticas que afectan el quehacer de la gestión de proyectos TI de la Fábrica de Software elegida para el desarrollo de la presente monografía; lo que exige la revisión de metodologías de gestión y su implementación, coherente con las necesidades del Área.

Particularmente se propone como alternativa de solución integrar algunas de las buenas prácticas de la metodología PMI en el desarrollo de los proyectos de la fábrica de software y recurrir a fortalecer la dinámica de trabajo con las recomendaciones de la metodología SCRUM en el desarrollo del producto.

## **2.2 Definiciones, acrónimos y abreviaturas**

Las siguientes son palabras definidas según el contexto donde se realizó el proyecto investigativo.

- SCRUM: Conjunto de buenas prácticas para el desarrollo de proyectos ágiles.
- PMI: Project Management Institute, organización internacional dedicada al estudio y promoción de la dirección de proyectos.
- TIC: Tecnologías de la información y las Comunicaciones
- FCV: Fundación Cardiovascular de Colombia
- SPRINT: Bloque temporal de división de trabajo para el desarrollo de proyectos
- STORY POINT: Medida para calcular el peso de una tarea
- AGILO: Herramienta utilizada en la Fábrica de Desarrollo para el seguimiento de tareas por SPRINT.

### **3. Justificación del proyecto**

Las metodologías de gestión de proyectos tradicionales presentan importantes dificultades en la gestión del tiempo y la flexibilidad de los procesos. Como indica [3], es posible mejorar estos aspectos recurriendo a metodologías ágiles como es el caso de SCRUM. Esta estrategia permite mejorar la gestión del proyecto, alinearse con los intereses del cliente, conocer y mantener el equipo motivado; apuntando al aumento de la productividad.

Ahora bien, como se destaca en la problemática la implementación de esta metodología no ha sido suficiente para la población objetivo del proyecto; por esta razón la presente monografía espera aportar en la integración con otras metodologías de gestión, como es el caso de la planteada por PMI. Se espera que esta integración sea más pertinente para la realidad de la Fábrica de Software.

Otra de las razones del desarrollo del presente proyecto es la consolidación profesional de la autora en su cargo de Analista funcional aportando al equipo de trabajo los conocimientos desarrollados en la Especialización de Gerencia Comercial de Proyectos de Telecomunicaciones; sobre todo en el actual momento que vive la institución de creación de nuevas sedes de mayor tamaño como lo es el Hospital Internacional de Colombia.

En síntesis, mejorar los procesos de diseño y desarrollo de software; desde una metodología integral de proyectos, ha trascendido de un valor agregado a una necesidad sentida. Es por esta razón que tanto la FCV como el autor de la iniciativa consideran de suma relevancia el desarrollo de la propuesta.

### **3.1 Alcance**

Al finalizar el presente proyecto la Fábrica de Desarrollo de Software de la FCV contará con los lineamientos necesarios para la implementación de estrategias basadas en PMI sobre las tareas de planeación y estimación llevadas a cabo en base la metodología SCRUM.

Gracias a las actividades propuestas, el personal que lleva a cabo estas actividades cuyos perfiles son: Jefe de la fábrica de software, Arquitectos de Software y Analistas Funcionales contarán con la capacidad de asignar tareas de acuerdo a la capacidad real del personal de desarrollo y estimar fechas de entrega de producto ajustadas a los interesados.

## **4. Objetivos**

### **4.1 General**

Proponer un conjunto de lineamientos y recomendaciones en el proceso de planeación y estimación de actividades para la Fábrica de Software de la Fundación Cardiovascular de Colombia (FCV); basados en metodología Scrum y PMI.

### **4.2 Específicos**

- Construir una línea base de conocimiento sobre la metodología Scrum y PMI asociado a la estimación y planeación de actividades.
- Identificar y cuantificar los indicadores de cumplimiento y productividad de la Fábrica de Software antes de la intervención propuesta en el proyecto.
- Implementar estrategias específicas para la estimación del tiempo y la planeación de entregables y actividades a realizar por la Fábrica de software de la Fundación Cardiovascular de Colombia (FCV); basadas en la integración de metodologías Scrum y PMI.
- Establecer el resultado de la aplicación del proyecto, a través del uso de dos instrumentos de medición para determinar mejoramiento en los indicadores establecidos en la Fábrica de Software.

## **5. Marco referencial del proyecto**

En esta sección del documento se busca que el lector cuente con las herramientas conceptuales básicas que enmarcan el proyecto, organizado a través de un marco referencial, marco histórico, marco tecnológico, un ejercicio básico de estado del arte.

### **5.1 Marco Teórico de la propuesta**

Se entiende la gestión de proyectos como un área interdisciplinaria donde convergen diferentes elementos entre ellos: el conocimiento técnico del área a trabajar, la gestión de los recursos asignados y las técnicas de dirección hacia los objetivos propuestos [4]. Para ello hoy en día se han desarrollado diferentes metodologías que nos permite realizar una gestión conjunta de todos los parámetros a medir y controlar en el desarrollo de un proyecto. A continuación se realiza un breve resumen de las metodologías de gestión de proyectos; así como metodologías ágiles las cuales son utilizadas comúnmente en el ámbito del desarrollo de proyectos de software.

#### **5.1.1 Metodologías de gestión de proyectos.**

La primera de las metodologías referenciadas en el presente proyecto es la desarrollada por el Project Management Institute o PMI; esta asociación ha contribuido con el Project Management Body of Knowledge (PMBOK) el cual, en su 5ta. Edición consolida pautas o normas dispuestas para la gestión de proyectos. El PMBOK se divide en 3 secciones: Marco Conceptual de la Dirección de Proyecto, Normas para la Dirección de un proyectos y las Áreas de Conocimiento, las cuales son [5]:

- Gestión de la Integración del Proyecto
- Gestión del Alcance del Proyecto

- Gestión del Tiempo del Proyecto
- Gestión de los Costos del proyecto
- Gestión de la Calidad del Proyecto
- Gestión de los Recursos Humanos del Proyecto
- Gestión de las Comunicaciones del proyecto
- Gestión de los Riesgos del proyecto
- Gestión de las Adquisiciones del proyecto
- Gestión de los Interesados del proyecto.

Basadas en el PMBOK y complementando lo ya descrito en se han desarrollado propuestas metodológicas para el desarrollo de proyectos como es el caso de PRINCE2. Los procesos de esta metodología interactúan con 8 componentes básicos [6]:

- Business Case: Define el rumbo del proyecto.
- Organización: Toma de decisiones sobre la inversión y el control presupuestario.
- Planes: Donde se detalla la planificación de las diferentes partes del mismo.
- Controles: Garantizar el cumplimiento.
- Gestión del riesgo: Análisis del riesgo y definición de estrategias.
- Gestión de la calidad
- Gestión de las configuraciones: Proporciona mecanismos de seguimiento y control de los entregables.
- Gestión del cambio: Verifica el impacto de cambios potenciales sobre el Business Case.

Otra metodología para la gestión de proyectos ha sido desarrollada por el International Project Management Association (IPMA) quien ha propuesto un programa de certificación para los directores de proyectos; de 4 niveles según el nivel de madurez a la hora de enfrentar desarrollo de distinta complejidad.

Estos niveles son: A – Certified Project Director, B – Certified Senior Project Manager, C – Certified Project Manager y D – Certified Management Associated; junto a 46 competencias de las cuales 15 refieren a competencias de comportamiento, 20 a competencias técnicas y 11 a competencias contextuales. Otra de las metodologías es la recientemente adquirida por la organización AXELOS como conjunto para las buenas prácticas para la gestión de servicios de tecnologías de la información conocida como ITIL o Information Technology Infrastructure Library.

Finalmente, cabe resaltar ITIL ofrece un marco de trabajo para las actividades del área de TI, proporcionando una descripción de los roles, tareas, procedimientos y responsabilidades que pueden ser adaptados en organizaciones de TI cuya finalidad sea mejorar la Gestión de sus Servicios; gracias a la cantidad de temas que cubre, se considera un elemento de referencia útil para las organizaciones, ya que permite fijar nuevos objetivos de mejora para la organización de TI, basándose en la calidad del servicio y en el desarrollo de los procesos de una manera eficaz y eficiente [7].

### **5.1.2 Metodologías de proyectos ágiles.**

En el año 2001 se incluye el término “Ágil” en el ámbito del desarrollo de proyectos de software. Bajo este concepto se crea *The Agile Alliance*, organización sin ánimo de lucro,

dedicada a promover los conceptos relacionados con el desarrollo ágil de software ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el **manifiesto ágil** [8] en él se registraron las siguientes consignas:

*“Valorar a las personas y su interacción, por encima de los procesos y las herramientas. Valorar el software que funciona, por encima de la documentación exhaustiva. Valorar la colaboración del cliente, por encima de la negociación contractual. Valorar la respuesta al cambio, por encima del seguimiento de un plan”.* [9]

Una de las metodologías más reconocidas para el desarrollo ágil de proyectos es SCRUM. La cual, es un proceso en el que se proponen aplicar un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible de un proyecto [10].

Esta metodología está especialmente indicada para entornos de alta complejidad, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la flexibilidad y la productividad son fundamentales.

- A continuación se presentan los términos básicos que representan SCRUM:
- Scrum Master: Es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer.
- Producto Owner: Conoce y maneja las prioridades del proyecto o producto.
- Sprint: Ciclo de trabajo; proceso organizado en un período de tiempo que se repite hasta finalizar el proyecto.
- Product Backlog: Corresponde con una o más tareas que provienen del product backlog. De este se sacan las tareas que formaran un sprint [11].

- Retrospectiva: Es un evento donde se da la oportunidad al equipo de tener una introspección; esta se realiza tan pronto como sea posible a la culminación del sprint. El objetivo de la retrospectiva es: inspeccionar cómo fue el último Sprint con respecto a las personas, relaciones, procesos y herramientas; identificar las cosas que salieron bien y las posibles mejoras y su ordenación; y, crear un plan en el que poner en marcha las mejoras.
- Daily meeting: Es un evento en un bloque de tiempo de 5 minutos en que el equipo sincroniza las actividades y hace un plan hasta la próxima reunión; también conocido como “reunión de pie” [12]. Normalmente los equipos la realizan diariamente donde se resume por persona: qué se ha hecho, qué voy a hacer y cuáles son los impedimentos que he encontrado en el desarrollo del proyecto.

Otra metodología a la que se hace referencia es la denominada Extreme Programming, la cual es una propuesta ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando el buen clima de trabajo. Basada en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios [13].

## **5.2 Marco Histórico**

Al realizar una búsqueda sobre la historia relacionada a la Gestión de Proyectos se encuentra con diferentes autores que sugieren el comienzo de esta área de conocimiento desde tiempos remoto o primeras civilizaciones. [14] [15].

A continuación se citan algunos de los acontecimientos que son ejemplo de la implementación de las prácticas de la Gestión de Proyectos:

- Construcción de la Gran Pirámide de Giza – 2570 a.C
- Construcción de la Gran Muralla China – 208 a.C
- Desarrollo del Diagrama de Gantt por Henry Gantt – 1917
- Creación del método de ruta crítica inventado por Dupont Corporation – 1957
- Invención de la Técnica de Revisión y Evaluación de Programas, desarrollada por la Armada de los Estados Unidos para el proyecto Polaris – 1958
- Aplicación de la Estructura de Desglose de Trabajo – 1962 [16]

Mediante la Tabla 2 se resume las fechas más importantes sobre la creación de metodologías o consolidación de entidades para la investigación y avances sobre la Gestión de Proyectos:

*Tabla 2. Historia de la Gestión de Proyectos*

<b>Fecha</b>	<b>Descripción</b>	<b>Importancia</b>
1965	Creación de la “International Project Management Association” (IPMA)	Fue la primera Asociación dedicada a la contribución de conocimiento acerca de la gestión de proyectos.
1969	Nace el “Project Management Institute” (PMI)	Es una de las organizaciones más importante al momento de referenciar sobre la implementación de la metodología a través de la guía PMBOK.
1975	Creación del método PROMPTII creado por Simpect Systems Limited	Es conocida como la primer metodología para la implementación de proyectos ágiles, reconociendo el áreas de los proyectos informáticos sobre estimados al implementar una de las metodologías ya nombradas.
1986	Creación de la metodología	Es la metodología más utilizada en el mundo

	SCRUM	para el desarrollo de proyectos ágiles.
1989	Desarrollo del Método de Desarrollo PRINCE a partir de PROMPTII	Esta metodología fue implementada por la Agencia Central de Informática y Telecomunicaciones del gobierno del Reino Unido, es uno de los referentes de herramientas fallidas ya que es considerada como difícil de manejar, demasiado rígida y solamente aplicable a grandes proyectos.
2008	El PMI lanza la 4ta. Edición del PMBOK	La publicación de nuevas versiones de los estándares propuesto por el PMI indica que la metodología ha sido utilizada y retroalimentada por diferentes entes y empresas alentando a la mejora continua en los procesos y avances en la investigación de la gestión de proyectos.
2012	Publicación de la 5ta. Edición de la Guía PMBOK	Se realiza la actualización del PMBOK generando modificaciones tanto en áreas de proyectos como en herramientas específicas. Se destaca la propuesta de incluir a los interesados como una nueva área de gestión.

Nota: La información presentada se ha adaptado de [4].

### 5.3 Marco Tecnológico

En esta sección se presenta el marco tecnológico que enmarca el presente proyecto. Principalmente se relacionan aplicativos software que facilitan la gestión de información de proyectos

- **GanttProject**

Aplicación de escritorio multiplataforma para la programación y gestión de proyectos. Es un proyecto software que surgió en la Universidad de Marne-la-Vallée en Francia. Su

principal función es la creación de diagramas de Gantt, lo que permite la visualización de jerarquía e interdependencia entre tareas. [17].

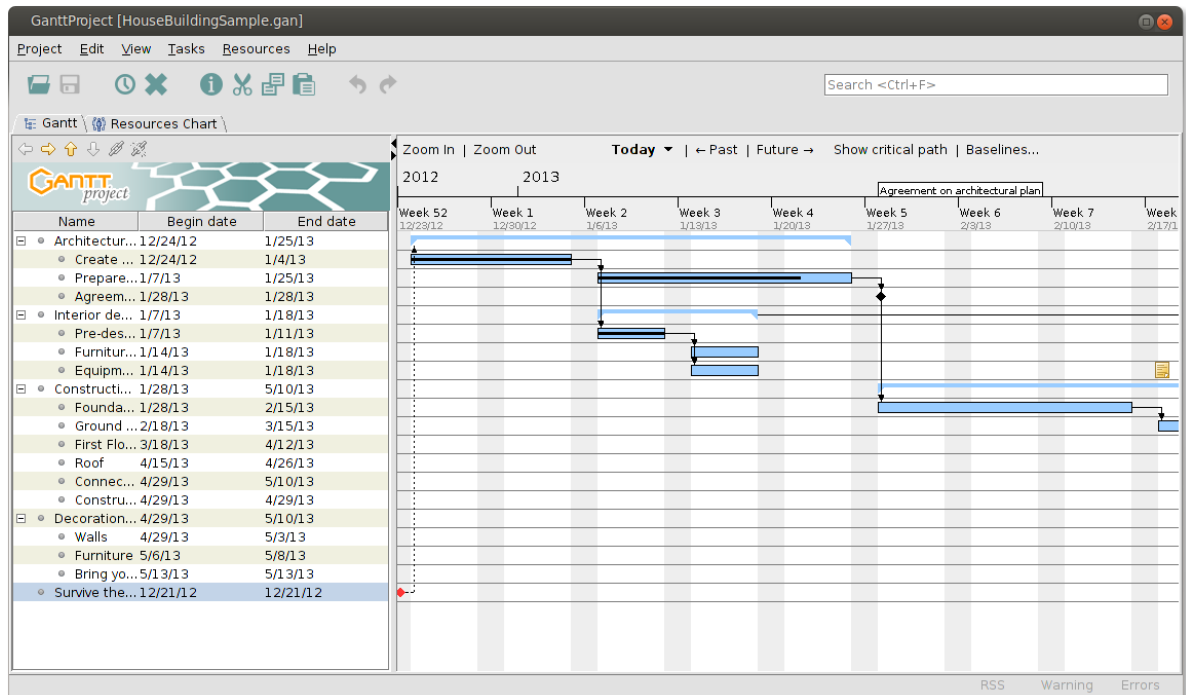


Figura 1 . Diagrama de Gantt

- **Ganttter**

Servicio web gratuito para la planificación de proyectos mediante la generación de diagramas de Gantt con fases, fases e hitos. Actualmente el servicio de alojamiento de archivos google Drive cuenta con una extensión para la creación de nuevos proyectos así como la generación de diagrama a partir de documentos Excel. [18]

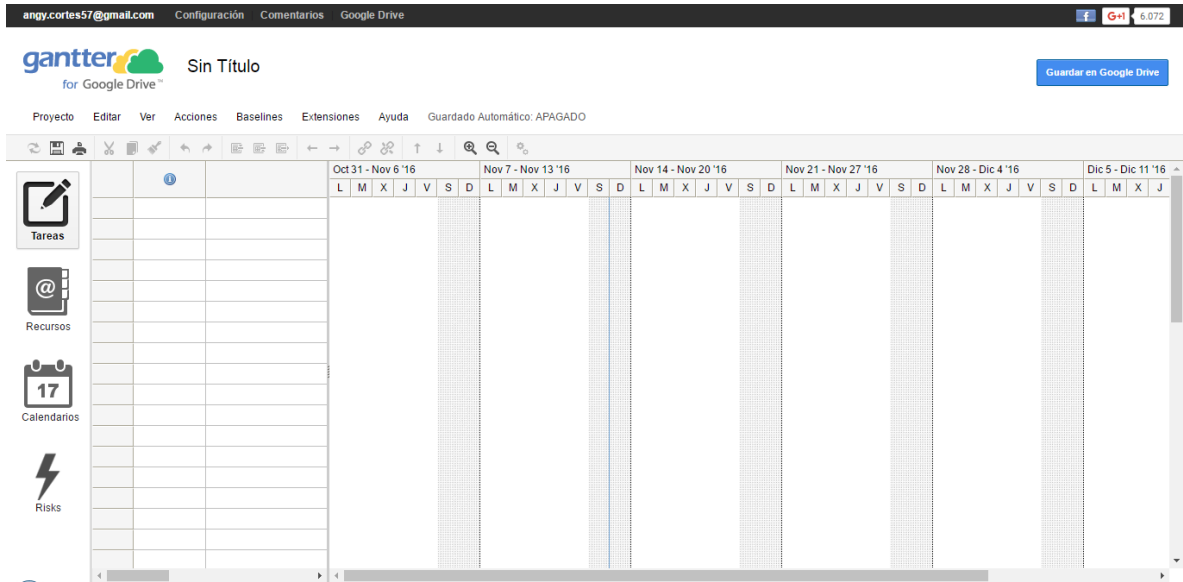


Figura 2. Software Gantt. Adaptado de:  
[https://www.researchgate.net/figure/264083084\\_fig4\\_Figure-5-Gantt-for-Google-Drive](https://www.researchgate.net/figure/264083084_fig4_Figure-5-Gantt-for-Google-Drive)

- **IceScrum**

Herramienta orienta especialmente a técnicas SCRUM, agrupa las tareas en un visual e intuitiva interfaz de post-it. Entre sus funcionalidades incluye: paneles de tareas, historiales de usuario, de asuntos, almacenamiento en la nube, histórico de problemas y solución de bugs.

[19]



Figura 3. Herramienta IceSrum. Adaptado de: <http://freecode.com/projects/icesrum>

- **Agilo for trac**

Herramienta web OpenSource basada en la metodología SCRUM, desde esta herramienta se apoya a los tres papeles SCRUM: Desarrollador, SCRUM Master y el propietario del producto, con diferentes características para apoyar las entradas al flujo de trabajo; así como la vinculación de tareas e historias y la trazabilidad completa de los requisitos a través de historias. [20]

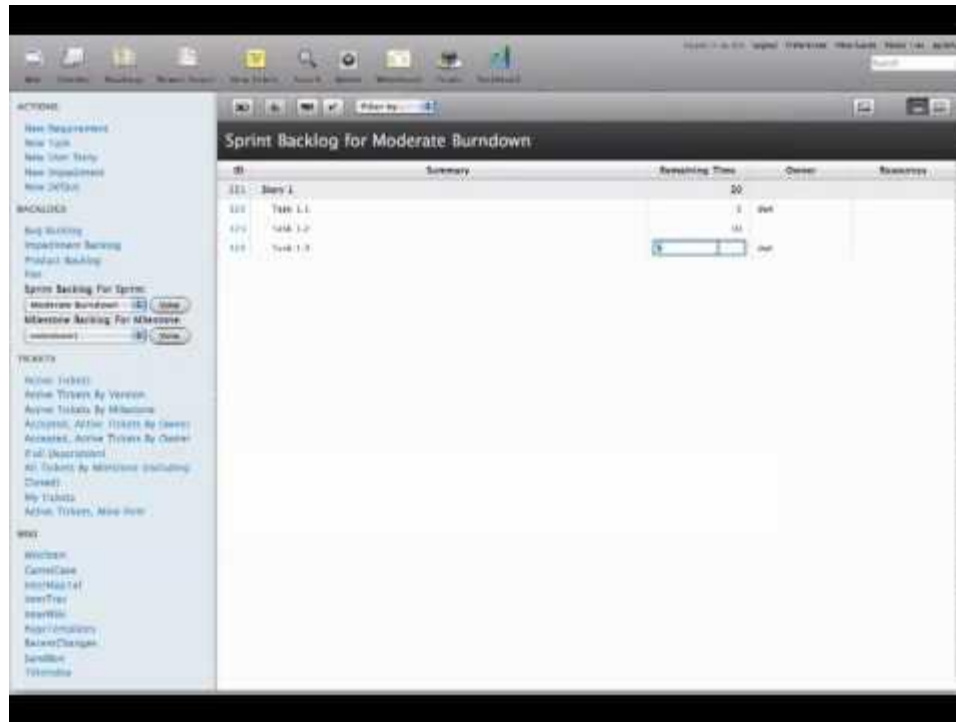


Figura 4. Herramienta Agilo for Trac. Adaptado de: <http://learn.agilest.org/wp-content/uploads/2016/05/scrum-tool-agilo-for-trac-backlo.jpg>

- **Trello**

Es un gestor de tareas que permite el trabajo de forma colaborativa mediante tableros compuestos por listas las cuales se encuentran en diferentes estados. Las tareas se agrupan en las siguientes lista: Por Hacer, en proceso y finalizada. [21]

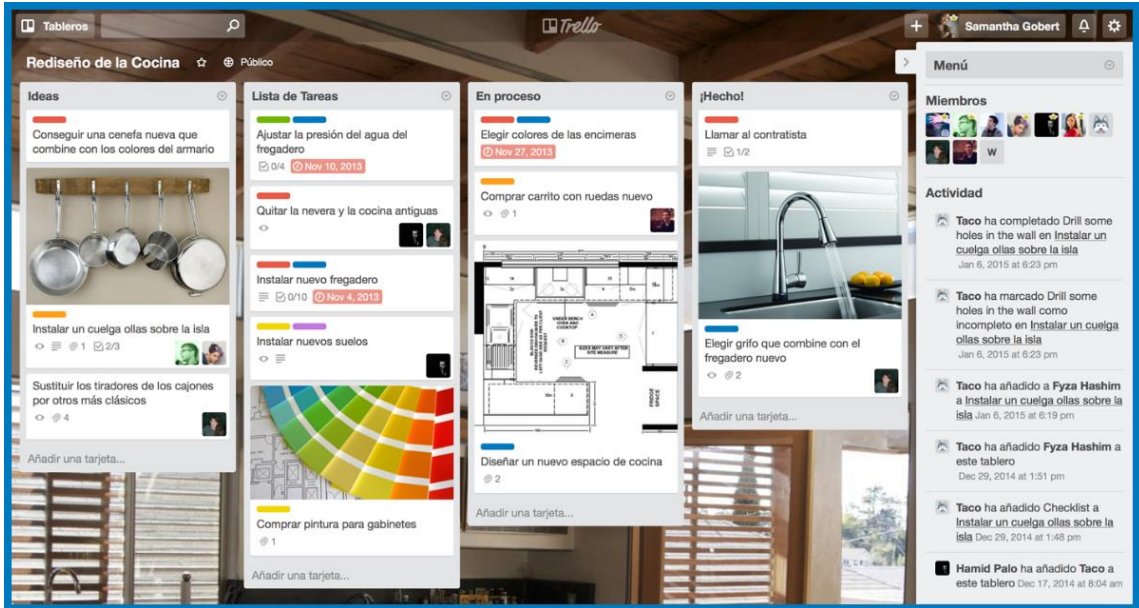


Figura 5. Herramienta Trello. Adaptado de: <https://trello.com/>

#### 5.4 Estado del arte

En esta sección se presenta y consolida la revisión de investigaciones realizadas por instituciones universitarias en conjunto con empresas interesadas en apropiar las herramientas PMI, la metodología Scrum; así como la integración de las mismas a sus actividades para la planeación y estimación de proyectos software, a través de metodologías ágiles. Se espera que con este ejercicio se logre consolidar diferentes experiencias, conclusiones, propuestas para la definición de los lineamientos que se proponen en el desarrollo del presente proyecto.

En primera instancia, la Tabla 3 presenta los elementos generales de la revisión. Cabe resaltar que en algunos momentos de la revisión se adjuntan citas más antiguas a la ventana de revisión, justificadas por su importancia en la discusión del contexto del proyecto.

Tabla 3. Criterios generales de la revisión de estado de arte

Concepto	Descripción
Palabras claves	Scrum, PMI y agile methodologies

Bases de datos revisadas	SCOPUS, Science-direct, IEEE explorer
Ventana de revision	2012-2017

Nota: Selección de criterios para la revisión de estado de arte.

#### 5.4.1 Metodologías ágiles de proyectos en el desarrollo de software- Consideraciones, aportes y retos.

El reto al implementar un proyecto de software por métodos no tradicionales (como puede entenderse una metodología ágil) es que se requiere un trabajo colaborativo mayor y una comunicación *cara-cara*. Experiencias de revisión de literatura contribuyen al desarrollo de esta sección, como es el caso propuesto por [22], por [23] y más recientemente por (Inayat, Salwah, Daneva, & Shamshirband, 2015). Estos últimos desarrollan una rigurosa pesquisa bibliográfica donde es posible encontrar metodologías diferentes aplicadas en contextos similares, y metodologías similares aplicadas en contextos diferentes.

Para sintetizar este ejercicio, se propone revisar las siguientes preguntas u objetivos de la revisión citada anteriormente; y se da cuenta de cada respuesta de forma sintética.

¿Cuáles son las prácticas adoptadas en el desarrollo de proyectos ágiles de software en estudios empíricos publicados?.

Entre las principales prácticas encontradas están:

- **Comunicación cara a cara (face-to-face)**, entre el equipo de desarrollo y los representantes del cliente. Esta es una de las características más importantes de las metodologías ágiles. Este aspecto es reforzado por [24] citado por [25]. La metodología motiva más a la creación de memorias (historias) en el trabajo con el usuario que el uso de

documentación completa y extensa. Aquí se logra que el cliente pueda redireccionar el proyecto a tiempo si el mismo apunta a la dirección equivocada.

- **Interacción y el involucramiento del cliente**, es presentada en las diferentes experiencias como la clave del éxito del proyecto. Se expone que la identificación del cliente o colectivos de interesados aseguran la correcta definición de los requerimientos y la priorización de los mismos. [26] citado por [25], refuerza este aspecto y motiva a la interacción constante y decidida.

- **Historia de usuario (user stories)**, son una herramienta que se crea para establecer los requerimientos específicos de los clientes. Esta estrategia permite concentrar la información en documentos cortos y bien acotados. El eje central de la historia de usuario es la definición clara de sus objetivos, enfocado en que es lo que realmente se quiere.

- **Requerimientos iterativos**, es decir que los requerimientos se van adaptando y actualizando en el dialogo constante con los interesados del proyecto. Los referentes revisados insisten en determinar claramente al inicio los alcances amplios de la propuesta y se pueden ir afinando aspectos concretos (detalles) del proyecto.

- **Priorización de requerimientos**, lo cual es parte de cada iteración de una metodología ágil. En métodos tradicionales la priorización ocurre una sola vez (inicio del proyecto), mientras que en procesos ágiles es continuo y se desarrolla en un ciclo desarrollador-cliente. En este punto vale la pena resaltar que uno de los grandes retos de las metodologías ágiles en cuanto a la priorización de requerimientos es que no solo se puede centrar en el valor del negocio (enfoque del cliente), sino que también se requiere verificar arquitectura de desarrollo, limitantes de diseño, entre otros aspectos.

- **Administración del cambio**, el cual es un reto no resuelto claramente en metodologías tradicionales. La constante relación *cara a cara* permite solicitar y gestionar cambios de forma efectiva. La claridad que los clientes ganan al mantener constante contacto permite refinar los requerimientos y evitar re-procesos futuros.

- **Equipos multifuncionales**, incluyen miembros de diferentes grupos funcionales que tienen objetivos similares. En las metodologías ágiles se encuentran desarrolladores, testers (validadores), diseñadores y administradores en el mismo espacio para dialogar y definir actividades.

- **Prototipado**, se presenta como una alternativa sencilla directa para revisar los requerimientos del cliente y ganar tiempo en la realimentación del mismo. El prototipado se realiza, como cita también [27], iniciando con los requerimientos más simples que tienen alta prioridad. El prototipado permite ver requerimientos que inicialmente no se haya determinado.

- **Pruebas antes de codificar**, lo que se puede entender como la importancia de escribir las pruebas (proponerlas) antes que escribir el código funcional. Esto permite contar con una realimentación detallada en caso de que las pruebas fallen. En este aspecto se encuentra, como se cita en [28], la posibilidad de contar con diferentes estrategias para la creación de las pruebas de validación, como es el caso del desarrollo ATDD (Acceptance Test-Driven Development). Este método relaciona prácticas tradicionales con prácticas ágiles; integrando por ejemplo documentación detallada con comunicaciones frecuentes.

- **Modelamiento de los requerimientos**, entre los cuales una de las técnicas más reconocidas consisten en el esbozo de las metas, generando gráficas intuitivas y de fácil lectura para los miembros del equipo, patrocinadores, director del proyecto, entre otras. Esta técnica

permite refinar constantemente los objetivos del proyecto y, como se cita en [29], potenciar la toma de decisiones mientras se afina la negociación de requisitos.

- **Administración y gestión de requerimientos**, se realiza manteniendo el backlog (lista de trabajo pendiente) o lista de características. Por ejemplo en scrum, el *product backlog* puede ser usado para mantener los requerimientos bien orientados.

- **Revisión de reuniones y aceptación de pruebas**, donde son desarrollados los requerimientos y los backlogs de producto que son constantemente revisados en reuniones. Esta es una forma de realizar chequeo y general balances para la medición de desempleo. Para el concepto de historia de usuario, la aceptación de pruebas se suele evaluar como aprobado o no aprobado.

- **Refabricación de código**, lo cual significa la revisión y modificación de la estructura del código, para poder adaptar cambios. Esta es una de las principales características de flexibilidad de las metodologías ágiles.

- **Conceptualizaciones compartidas**, se considera un elemento de soporte a la metodología que permiten recopilar, clarificar y consolidar los métodos ágiles. El trabajo presentado en [30] indica que dichos conceptos deben ser construidos y almacenados en repositorios individuales que se comunican a los actores, generando colaboración.

- **Emparejamiento para el análisis de requisitos**, el cual es una práctica que alienta (motiva) a los interesados a desarrollar sus roles. Es una estrategia pensada en la reducción en brechas de comunicación en los proyectos con metodologías ágiles.

- **Retrospectiva**, son los encuentros que se realizan tras cada iteración. Estos encuentros o reuniones permiten revisar el trabajo completado y determinar pasos futuros y posibles reprocesos.

- **Planeación continua**, la cual es una tarea rutinaria para equipos ágiles. El equipo de trabajo, en una metodología ágil, no se apega a un plan estático; sino que está siempre reevaluando y generando cambios constantes. Esta flexibilidad permite el constante cambio de requisitos.

El segundo aspecto a revisar se atiende en el siguiente interrogante:

¿Cuáles son los retos del desarrollo de proyectos de software por medio tradicional que resuelven las metodologías ágiles?

Los trabajos propuestos en [24] y en [31] indican cómo las metodologías ágiles han logrado mejorar aspectos de las estrategias tradicionales como:

- **Brechas de comunicación**, disminución de latencias entre la generación de la información y la recepción por las personas interesadas. Los encuentros cara a cara permiten contribuir en esta disminución.
- **Sobredimensionamiento**, es decir que el alcance del proyecto sobrepasa considerablemente los recursos existentes. Este punto en particular permite hacer trabajo no deseado o no requerido.
- **Validación de requerimientos**, el cual es una de los principales retos para una metodología tradicional. Los métodos ágiles [22] realizan constantes priorizaciones de sus requerimientos, ya que al final de cada iteración, la historia de usuario es demostrada al dueño del producto.

- **Documentación de requerimientos**, en metodologías ágiles se requiere menor cantidad de documentación; ya que los constantes encuentros permiten llevar el desarrollo e forma coherente con las necesidades. En este punto, un autor que realiza un interesante énfasis que vale la pena revisar es Bjarnason, presentado en [32]; resalta la importancia de generar una ingeniería de requisitos de forma sistemática y organizada.

Finalmente surge la siguiente inquietud: ¿Cuáles son los retos principales de las metodologías ágiles de proyectos?

Principalmente, [25, 32] resaltan que le principal reto de las metodologías ágiles es encontrar un balance entre agilidad y estabilidad, asegurando la calidad suficiente en el trabajo desarrollado por equipos multidisciplinarios. Otros aspectos que no han sido debidamente tratados son aquellos asociados a los requerimientos no funcionales, como son la seguridad de la información y la escalabilidad del sistema.

Finalmente, la metodología ágil propone siempre la importancia de el cara a cara; sin embargo es clave contar con la disponibilidad del cliente pues no es un aspecto que sea fácil de obtener y gestionar en un proceso de desarrollo.

#### **5.4.2 Integración de metodologías y adaptaciones para los diferentes entornos organizacionales**

Es común encontrar en las organizaciones, y no es la excepción en la Fábrica de Software de la FCV; la integración de múltiples metodologías de desarrollo y métricas de calidad; las cuales deben contar con un balance adecuado para aumentar la eficiencia en la gestión de los proyectos.

En [33] se presenta una interesante discusión en la integración de metodologías ágiles y el estándar ISO 21500; indicando que a pesar de ser dos metodologías con fundamentos que, a primera vista, pueden ser contradictorios, es posible encontrar puntos de mutua potencialización para tener mejores resultados en proyectos complejos. Los autores indican que la integración de los dos modelos se da extrayendo definiciones de alto nivel de uno (ISO), como es el ISOSG, ISOPG y el ISOP; con el desarrollo de proyectos en metodología ágil. Se encuentra que es posible con esto mejorar el seguimiento a aspectos financieros, legales y estándares de procedimiento de las compañías grandes.

Otra propuesta de discusión es la mostrada en [34] donde el autor destaca que aunque las estrategias tradicionales de gestión de proyectos y las ágiles tienen muchos aspectos en contraposición, es en los proyectos de desarrollo de software donde las misma cobran validez; logrando generar integraciones efectivas. Las ventajas y desventajas de las diferentes metodología se reflejan con mayor o menor relevancia en los diferentes tipos de proyecto y los tipos de organizaciones. El autor presenta una sentencia clave: *“la metodología debe adaptarse al proyecto y no viceversa”*; la cual justifica la iniciativa del presente trabajo de especialización.

El trabajo desarrollado en [35] propone una estrategia práctica para la integración de modelos de gestión fundamentados en el PMI, con el marco ágil de scrum; orientado a organizaciones pequeñas o medianas enfocadas al desarrollo de software. En términos generales los elementos básicos de gestión y administración son descargados a las estrategias del PMI, mientras que la interacción con los clientes cara a cara, las iteraciones realizadas y el control de calidad se enfoca desde la propuesta de SCRUM. Un caso de estudio de integración de estas metodologías en el sector transporte (gestión de información en dicho sector), es presentado en [36].

En conclusión, se cuentan con antecedentes que perfilan la posibilidad de integración de las dos metodologías; aprovechando las mejores prácticas de cada una de estas y sobre todo aquellas que se puedan trasladar y optimizar.

## 6. DESARROLLO DEL PROYECTO

### 6.1 Plan de las Fases y Etapas del Proyecto (WBS)

En la **Figura 6** se presenta con detalle las etapas propuestas para el desarrollo del presente proyecto.

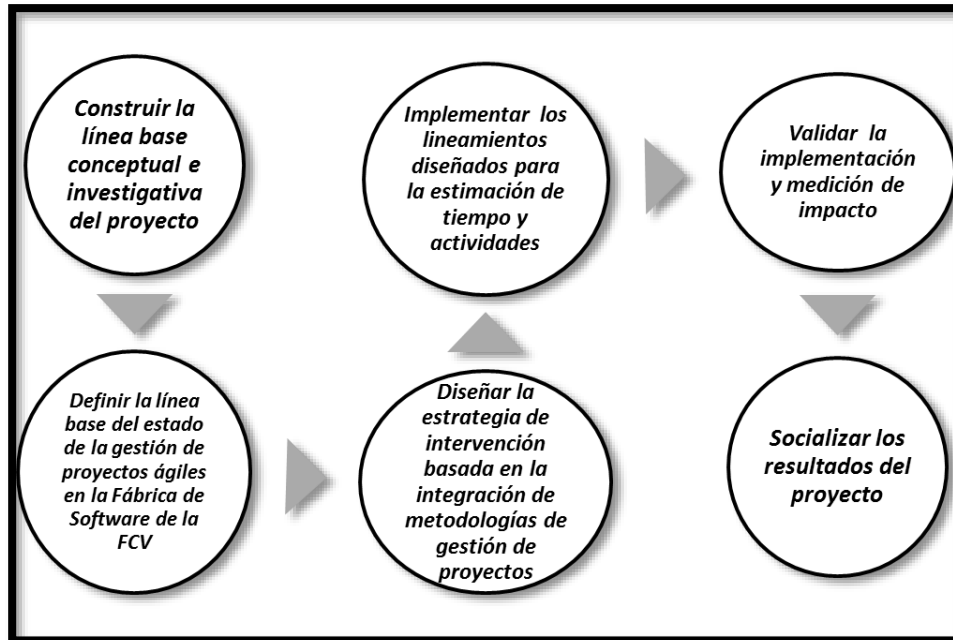


Figura 6. Propuesta del desarrollo del proyecto presentada en el anteproyecto

A continuación se desglosan las actividades:

**ACTIVIDAD 1- Construir la línea base conceptual e investigativa del proyecto:** En esta primera fase se consolidará la base documental del proyecto, principalmente en términos conceptuales e investigativos. En términos generales se profundizará sobre las temáticas propias del desarrollo del proyecto, así como se reforzará la revisión del estado del arte presentada en el anteproyecto. Como resultado de esta actividad se espera que el autor cuente con los fundamentos teóricos requeridos para el desarrollo de la iniciativa.

**ACTIVIDAD 2- Definir la línea base del estado de la gestión de proyectos ágiles en la Fábrica de Software de la FCV:** En esta fase se determinará el estado inicial sobre cómo se ejecuta la gestión de proyectos, a través de metodologías ágiles, en la población objetivo. En este punto se identificarán los indicadores de cumplimiento y productividad; los cual permitirá la realización una valoración pre-test. En este punto el análisis de los históricos de desempeño de la Fábrica de Software serán fundamentales para determinar la línea base.

**ACTIVIDAD 3- Diseñar la estrategia de intervención basada en la integración de metodologías de gestión de proyectos:** En esta tercera actividad del proyecto el autor generará, basado en los resultados de las actividades previas, lineamientos para la estimación de tiempo y actividades los cuales incluirán estrategias de metodologías SCRUM y PMI para la intervención al quehacer de trabajo en la fábrica. Esta estrategia será socializada inicialmente con el Director del Proyecto y con el Director (o delegado) del equipo de la Fábrica de Software de la FCV para su aprobación.

**ACTIVIDAD 4- Implementar los lineamientos diseñados para la estimación de tiempo y actividades:** En esta cuarta etapa se propone la implementación del / lo(s) lineamientos diseñados para el mejoramiento del desempeño en los indicadores seleccionadas. La ejecución se realizará en un periodo de tres (3) meses, donde se pueda realizar el seguimiento del mismo número de iteraciones.

**ACTIVIDAD 5- Validar la implementación y medición de impacto:** En esta etapa el ejercicio se centra en analizar las valoraciones pos-test; que unidas al seguimiento realizado en el desarrollo del proyecto (al terminar cada iteración) pueda generar conclusiones sobre la intervención realizada. En este punto se preparará un documento con lineamientos y recomendaciones específicas para la Fábrica de Software.

**ACTIVIDAD 6- Socializar los resultados del proyecto:** En esta fase final, el autor de la propuesta socializará, ante la Universidad y la Empresa; los resultados del trabajo y las recomendaciones propuestas.

Finalmente, en la

se presenta el Esquema de Desglose de Trabajo del proyecto, el cual se relaciona directamente con los objetivos propuestos.

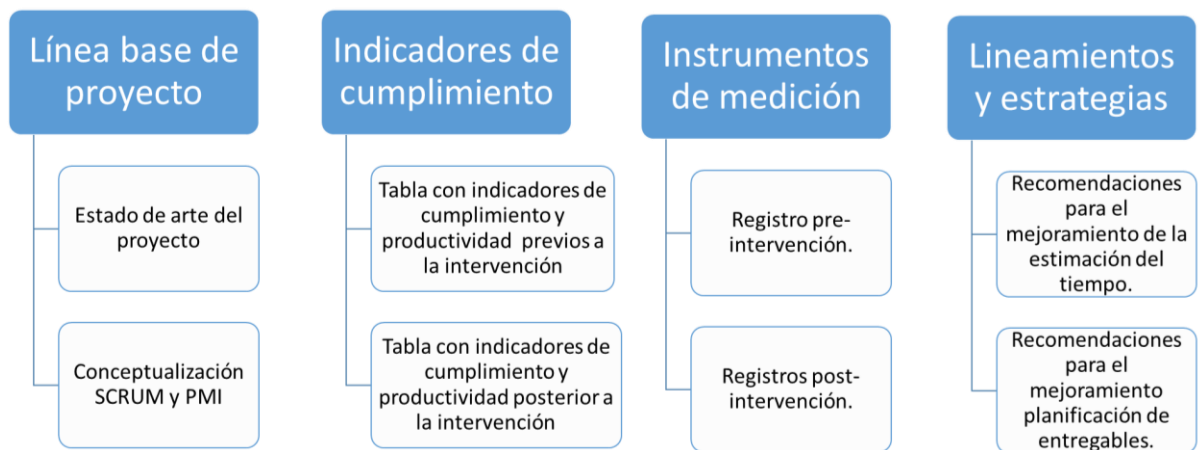


Figura 7. EDT del proyecto.

## 6.2 Descripción general del proceso de la Fábrica de Software

A continuación se presenta, el proceso llevado a cabo al momento de desarrollar un proyecto por la Dirección de Tecnologías de la Información y las comunicaciones (Ver Figura 8). Se describirá los actores que participan en el proceso y se demarcará los momentos en donde se realizará intervención para el desarrollo del presente proyecto.

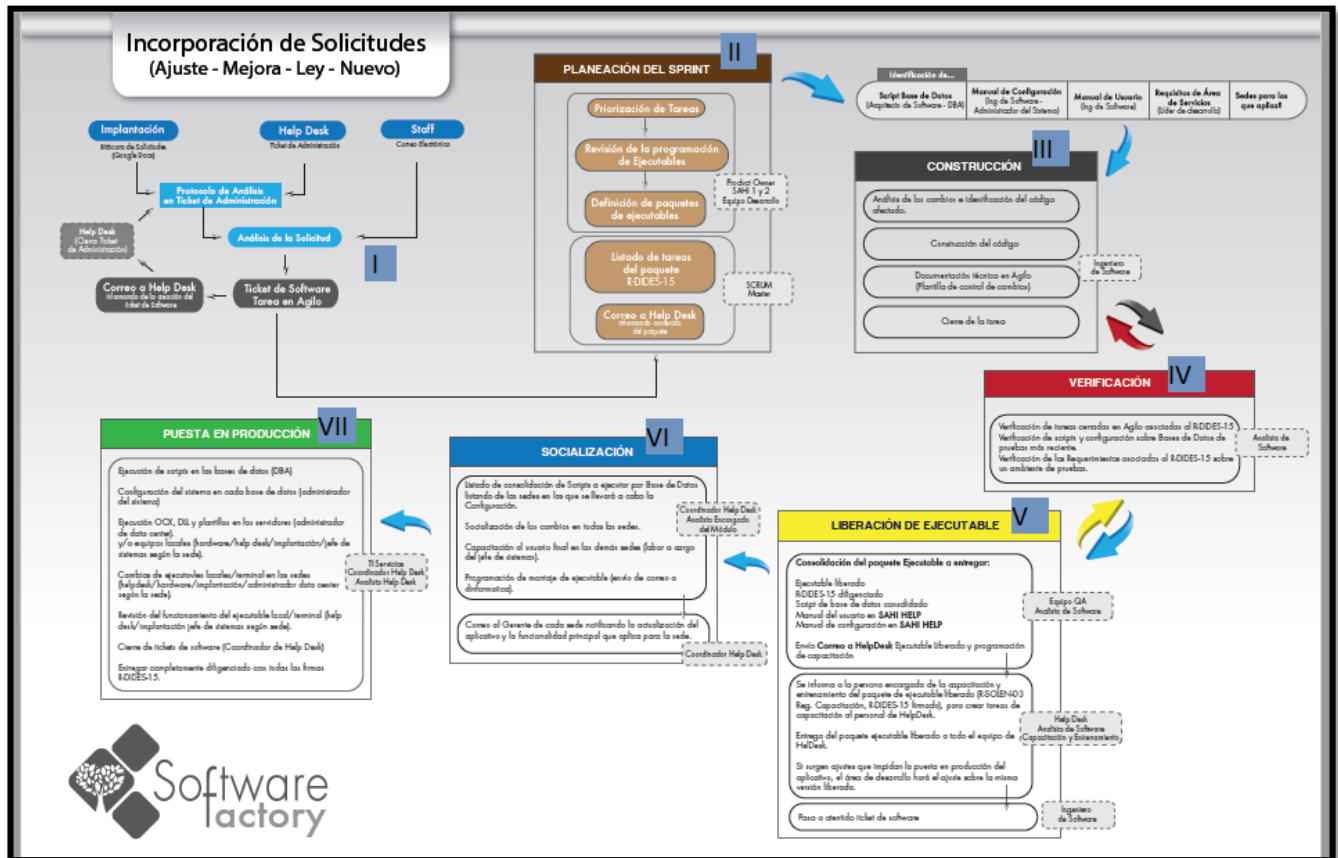


Figura 8. Proceso Fábrica de Software FCV

### 6.2.1 Recepción de solicitud

Este subproceso es considerado el inicio a la elaboración de un proyecto o el desarrollo de una nueva versión. El cual es llevado a cabo diariamente.

Los actores que hacen parte del proceso son:

- **Implantación**, son quienes reciben observaciones al momento de poner en marcha una nueva funcionalidad o instalar uno de los sistemas al cliente.
- **Help Desk**, son los encargados de realizar el soporte de 1er y 2do nivel conociendo cuando alguno de ellas debe ser categorizado como ajuste ó mejora dentro del proceso.
- **Staff**, así se conoce a los usuarios clientes quienes pueden realizar solicitudes de cambios. Ejemplo: Presidente de la Fundación.

Una vez es recibida la solicitud esta se convierte en un ticket en el sistema de mesa de ayuda, el cual es direccionado a la Fábrica de Software mediante un usuario genérico que permite el fácil conocimiento de los tickets que han sido asignados a la Fábrica. El proceso se sintetiza en la Figura 9.

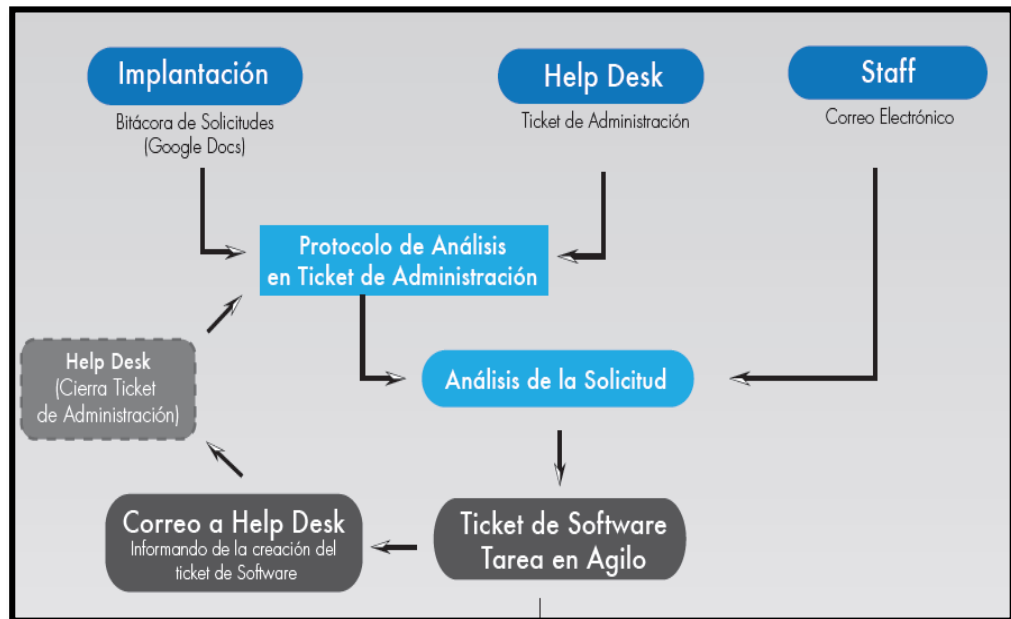


Figura 9. Recepción de solicitudes. Adaptado de: Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

### 6.2.2 Planeación del Sprint

Esta es una de las herramientas estipuladas por la metodología SCRUM, como se establece en la guía de SCRUM:

*“Es una caja de tiempo establecido de un mes o menos durante la cual se crea un incremento de producto realizable, utilizable y potencialmente liberable. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior. Estos se componen de: Planeación del Sprint, Revisión del Sprint y la Retrospectiva del Sprint ó Cierre de Sprint. Durante un Sprint no se deben realizar cambios que pongan en peligro el objetivo del Sprint; el alcance puede ser aclarado y renegociado entre el propietario del producto y el equipo de desarrollo...” [37]*

Para contextualizar la definición anterior el Sprint en la Fábrica de Software equivale a 1 mes, en donde a comienzos del mismo se realiza la Planeación del Sprint

mediante el Comité de Control de Cambios en donde participan: Jefe de Implantación, Jefe de la Fábrica de Software, Arquitectos de Software y Analistas Funcionales de Software; con el objetivo de revisar el cierre del Sprint anterior, cuáles de los proyectos o versiones se deben continuar desarrollando, actualización en la priorización de actividades, entre otros.

Una vez se conoce los temas que se deben llevar a cabo los Analistas Funcionales junto al Jefe de la Fábrica y los arquitectos ejecutan las actividades de la Figura 10. Este subproceso es el de mayor riesgo identificado con respecto a lo expuesto en la problemática donde se evidencia bajos porcentajes de cumplimiento, nula participación del usuario final o Product Owner para la definición de requerimientos así como deficiencias en la asignación de carga laboral; por lo tanto el desarrollo de la propuesta se centrará en las actividades realizadas en él.

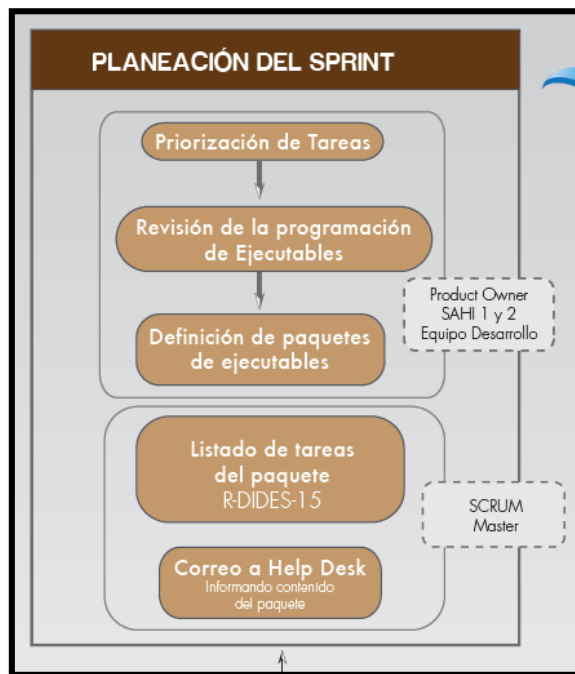


Figura 10. Planeación del Sprint. Adaptado de: Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

### 6.2.3 Construcción

En este punto es llevado a cabo el desarrollo de lo definido al comienzo de un Sprint. En él intervienen los Arquitectos, Analistas y Desarrolladores de Software. El detalle se observa en la Figura 11.

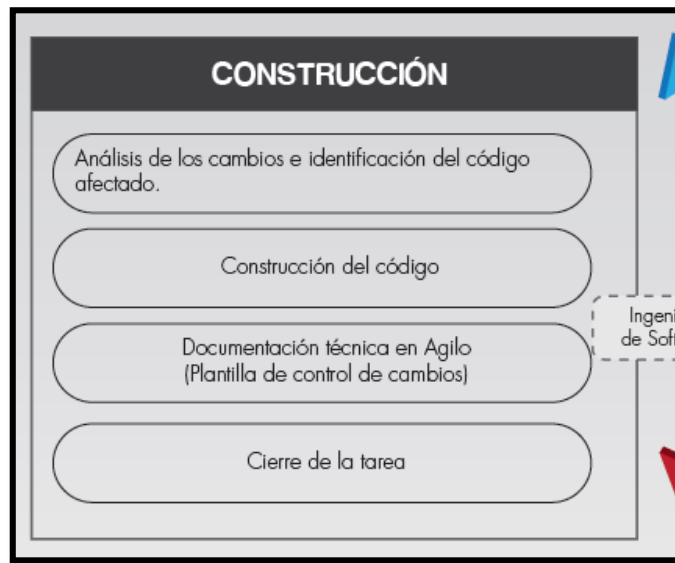


Figura 11. Construcción de Entregable. Adaptado de: Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

### 6.2.4 Verificación del Entregable

Luego de la construcción de un producto es llevado a cabo la verificación de las tareas, en donde se recopilan como documentación Script de base de datos, configuración para la puesta en marcha, manual de usuarios, entre otros documentos. En él intervienen: Analistas Funcionales y Analistas Tester. Para comenzar con la verificación de un ejecutable es de vital importancia contar con el Plan de Verificación en donde se dispone

para cada desarrollo los escenarios que se deben ejecutar con los resultados esperados, esto permite un control sobre los pasos que se deben llevar a cabo al momento de probar una funcionalidad en el sistema. El detalle es mostrado en la Figura 12.

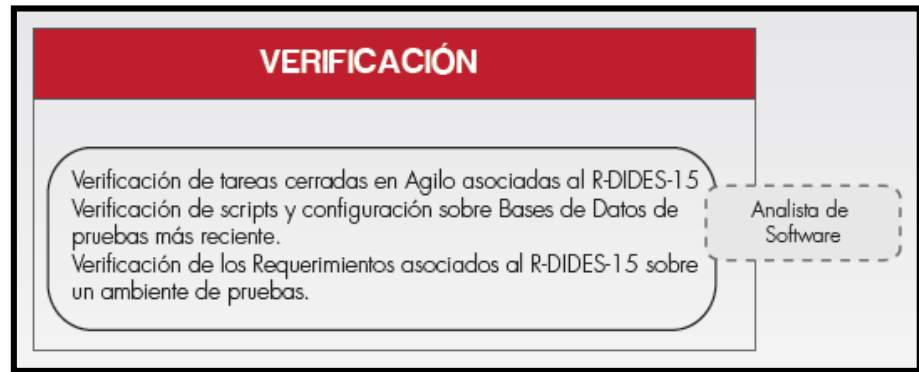


Figura 12. Verificación del Entregable. Adaptado de Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

### 6.2.5 Liberación de ejecutable

Este subproceso es llevado a cabo durante un Sprint según la cantidad de versiones o productos que cumplan a satisfacción los Planes de Verificación. En él los Analistas Tester, informan a los interesados sobre la liberación de los requerimientos solicitados. Los procesos de verificación y liberación de ejecutable pueden llevarse a cabo varias veces durante un sprint. Ver Figura 13.

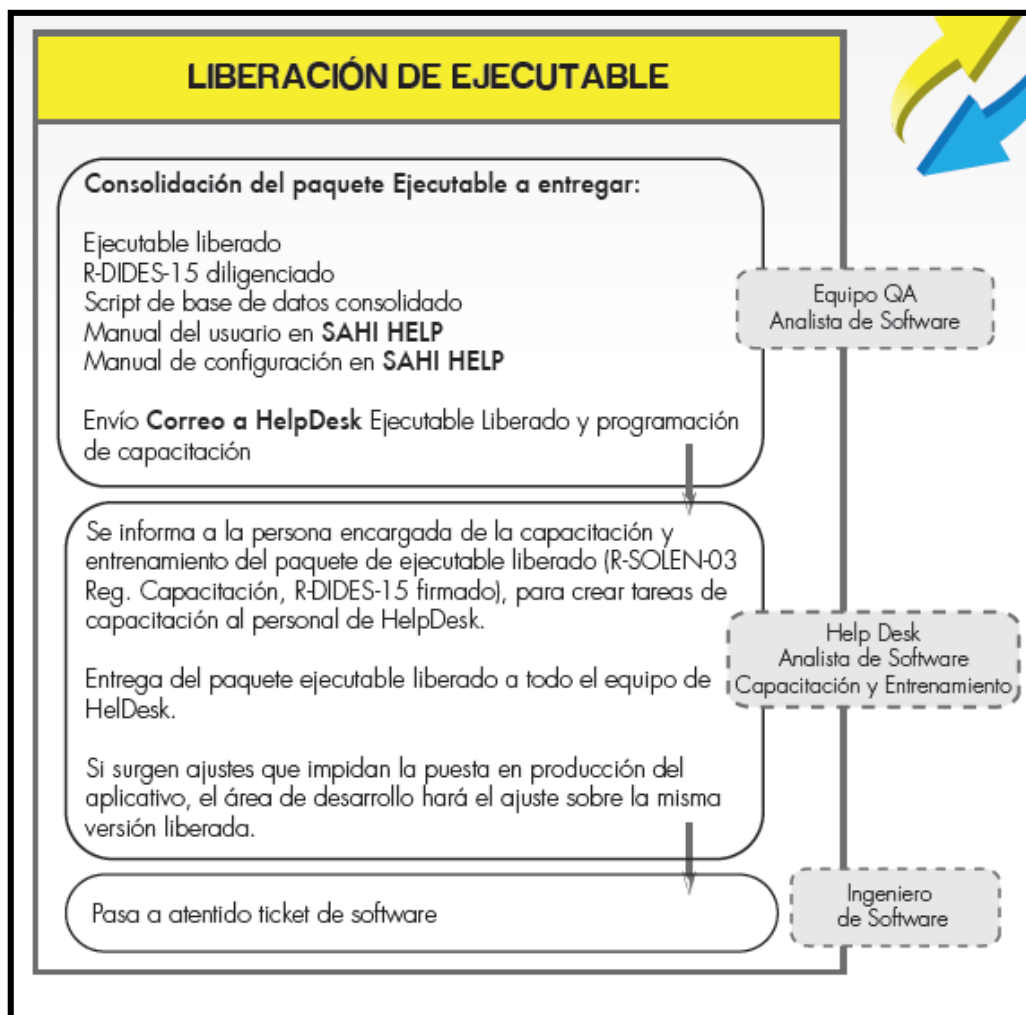


Figura 13 Liberación del Ejecutable. Adaptado de: Sistema de Gestión de Calidad

Fundación Cardiovascular de Colombia

### 6.2.6 Socialización

En este punto el Analista de la mesa de ayuda (personal externo a la Fábrica de Software) encargado de cada uno de los requerimientos que han sido liberados se encarga de socializar a los interesados. Ver Figura 14.

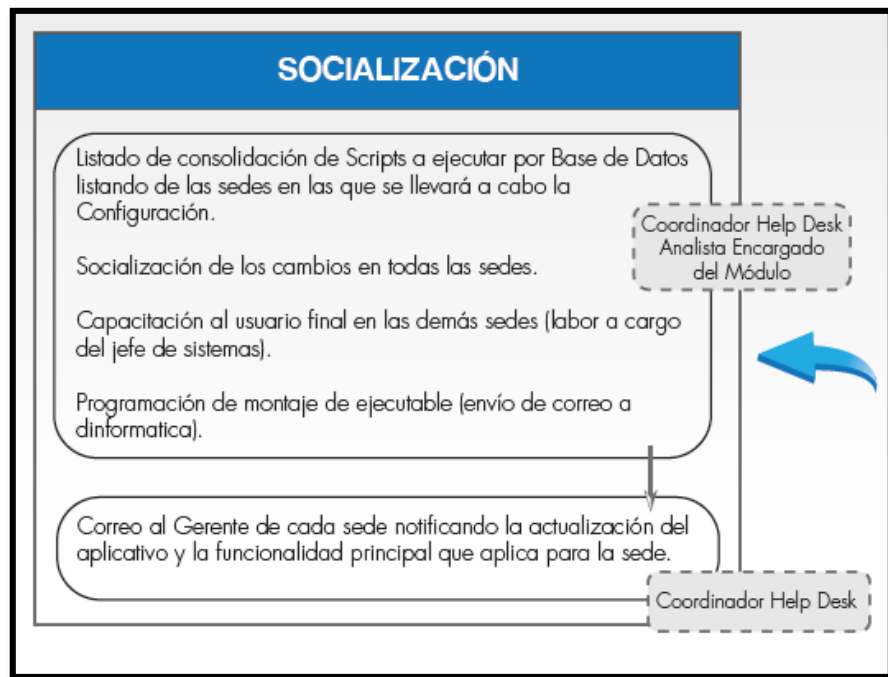


Figura 14. Socialización. Adaptado de: Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

### 6.2.7 Puesta en producción

Para finalizar el equipo de implantación junto a los equipos de Help Desk de cada cliente se encargan de definir la fecha para la puesta en marcha de una versión o producto. Este punto no requiere intervención por parte de la Fábrica de Software. Ver Figura 15.

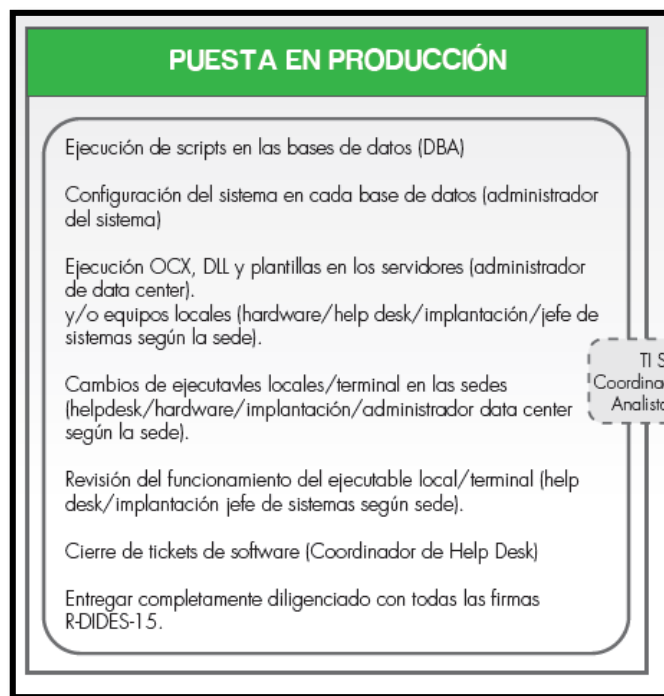


Figura 15. Puesta en producción. Adaptado de: Sistema de Gestión de Calidad Fundación Cardiovascular de Colombia

Para un mejor entendimiento sobre la interacción de los actores con los subprocesos se ha diseñado una matriz RACI, en donde: R : Responsible, A: Accountable, C: Consulted, I: Informed. Esta tabla es mostrada en Tabla 4.

Tabla 4 - Interacción actores - proceso

Roles / Proceso	Director TIC	Product Owner (Cliente solicitud)	Jefe de proyectos e implantación	Jefe de Fábrica de Software	Equipo desarrollo Fábrica de Software (Analistas funcionales, arquitectos, analistas tester y desarrolladores)	Equipo Help Desk DTIC	Equipo Help Desk Sistemas Cliente
Recepción de solicitud		I	R	I	I	A	A
Planeación del Sprint	I		C	R	A		

Construcción				R	A		
Verificación				R	A		
Liberación del Ejecutable	I		I	R	A		
Socialización		I	R			A	I
Puesta en producción	I	C	R	I	I	A	R

Para continuar con el estado de la gestión de proyectos en la Fábrica de Software de la FCV, es necesario realizar un análisis sobre los indicadores de calidad internos los cuales son tomados en cada cierre de sprint.

Considérese capacidad como el volumen máximo que una entidad puede producir con unas cantidades de factores productivos dados [38]. Ahora bien cumplimiento se entiende como el porcentaje de actividades cerradas vs. Cantidad de tareas planeadas. A continuación en la Tabla 5. Descripción de indicadores se presenta

*Tabla 5. Descripción de indicadores*

<b>Nombre del indicador</b>	<b>Representación</b>	<b>Numerador</b>	<b>Denominador</b>	<b>Dirección</b>	<b>Rango deseado</b>
Capacidad	Porcentaje (%)	Número de tareas cerradas	Número de tareas en el mes	Arriba	80 - 100
Cumplimiento	Porcentaje (%)	Número de tareas planeadas cerradas en el mes	Número de tareas planeadas en el mes	Arriba	80 - 100

En la Figura 16, se presenta la capacidad y el cumplimiento para los meses de Febrero, Marzo, Abril y Mayo los cuales serán la medida de comparativa del presente proyecto siendo estos el punto inicial de revisión para el alcance propuesto.

2016												
Sprint	Activ Mes	Plan Mes	Peso Plan Mes	Cerradas					Peso Cerradas	Plan Cerradas	Capacida d	Cumplimi ento
				Tareas	Nuevos	Mejoras	Ajustes	Total				
Enero												
Febrero	157	132	412	40	1	56	17	114	398	113	73%	86%
Marzo	259	157	524	60	0	53	26	139	401	100	54%	64%
Abril	168	100	426	36	14	30	24	104	439	80	62%	80%
Mayo	104	70	208	10	5	40	10	65	272	40	63%	57%

Figura 16. Indicadores por Sprint Fábrica de Software. Adaptado de: Tomado de Seguimiento a Montaje de ejecutables hoja Indicadores x Sprint de la Fábrica de Software

Al realizar la recopilación de los datos se determina que ambos indicadores se encuentran por debajo del resultado esperado. Esto debido a diferentes factores que se encuentran al revisar a detalle la toma de datos; en cada uno de los meses el número de tareas que fueron cerradas planeadas es menor al número de tareas planeadas lo que supone que durante el transcurso de los meses se incluyó un considerable número de tareas no tenidas en cuenta en la planeación de las actividades.

Adicional a los indicadores internos, se capturan los datos para estos mismos meses de los puntos realizados por cada uno de los integrantes que conforman el equipo de la Fábrica de software (ver Tabla 6. Puntos estimados y puntos realizados por mes por persona), esto permite identificar la carga asumida; así como reconocer lo estimado al comienzo del sprint y el resultado obtenido en el cierre.

Tabla 6. Puntos estimados y puntos realizados por mes por persona

Nombre	Mar		Abr		May	
	Total Est.	Total	Total Est.	Total	Total Est.	Total Real.
Tester 1	17	14,5	13	17	15	19
Analista Funcional 1	15	10	11	37	19	13

Tester 2	16	8,5	13	18	12	18
Analista Funcional 2	2	27,5	11	6,5	13	13,5
Tester 3	24	14	18	23	21	16
Arquitecto 1	13	38,5	19	1	19	8
Desarrollador 1	20	27	20	8	21	15,5
Desarrollador 2	15	15	13	16	12	10
Desarrollador 3	23	21	25	20	23	19,5
Arquitecto 2	18	31	21	8	15	14
Desarrollador 4	22	24	24	30	24	21
Desarrollador 5	27	28	27	16	24	28
Desarrollador 6	18	21	20	17,5	16	9
Tester 4	35	23,5	29	45	31	23
Desarrollador 7	0	0	0	44	15	11
Jefe Fábrica	20	14	14	7	10	8
Desarrollador 9	30	28,5	32	31,5	33	17
Desarrollador 10	21	12	19	21	21	14
Desarrollador 11	33	20	23	0	11	0
	<b>415</b>	<b>381</b>	<b>385</b>	<b>366,5</b>	<b>373</b>	<b>277,5</b>

Nota: La información presentada es adaptada del Plan de Equipos hoja Story points Desarrollo2016 de la Fábrica de Software

Como resultado se encuentra:

- El 54% de los desarrolladores realizan menor cantidad de punto que los estimados a su capacidad.
- Entre los Tester la cantidad de puntos estimados a su capacidad tiene una marcada dispersión esto se puede deber a una sobrecarga de actividades.
- Cada uno de los meses que fueron medidos no tuvieron cumplimiento de los puntos estimados.

### **6.3 Diseño de la estrategia de intervención**

#### **6.3.1 Primera propuesta de intervención**

La primera propuesta estudiada para la Fábrica de software fue la implementación de GanttProject como extensión de google drive para el seguimiento de las actividades planeadas en el Sprint. Se tomó en consideración las ventajas ya conocidas que brinda esta

herramienta para el seguimiento y control en la ejecución de proyectos [39]. Para ello se realizó la exportación de las actividades de un sprint las cuales son creadas desde la herramienta Agile for Trac, luego esta información fue copiada en una Hoja de Cálculo de Google y luego se utilizó el complemento Gantter.

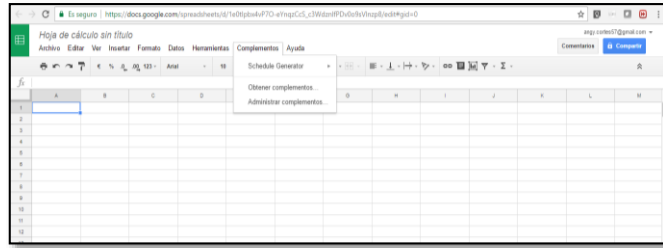


Figura 17. Complemento Gantter de Google Drive

Una vez se han ingreso los datos obtenidos de la herramienta Agilo en el complemento se solicitan los siguientes datos: Nombre de la tareas, fecha de inicio, fecha de finalización, recurso y notas; de los cuales mediante la exportación se contaba solo con el nombre y el recurso asignado a la tarea. Luego se debía llevar a cabo de manera manual agregar las fechas de inicio y fin estimadas para la realización de cada tarea según la experiencia y conocimiento sobre la velocidad de desarrollo o análisis del recurso asignado a la tarea por parte del equipo Staff quienes llevaban a cabo esta acción.

A	B	C	D	E
Enter your project tasks in the template below. Once you are ready to generate a Gantter project schedule from the tasks, select the "Create project from template sheet" option.				
TaskName	StartDate	FinishDate	Resource	Notes

Figura 18. Estructura para el ingreso de datos

A continuación el complemento genera el diagrama de Gantt requerido.

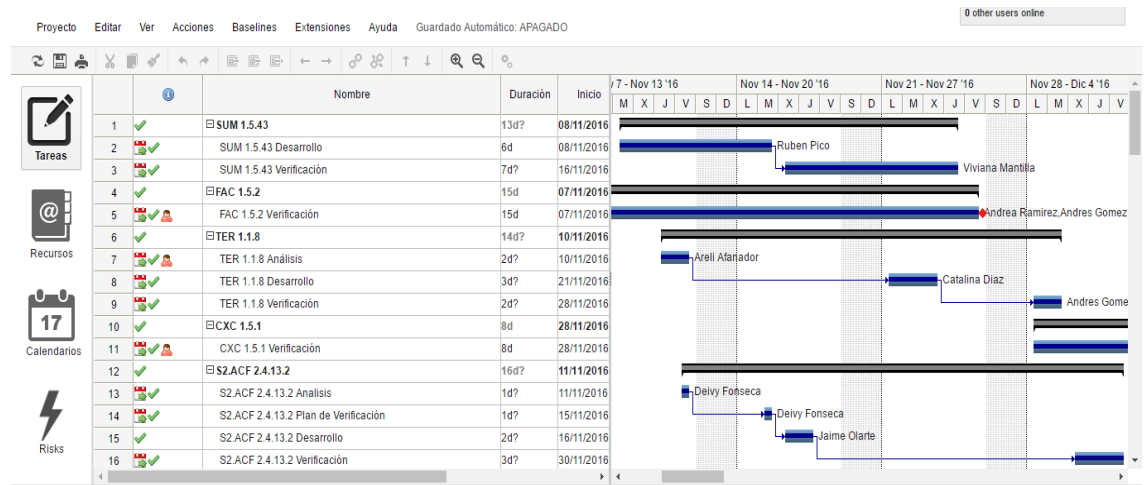


Figura 19. Visualización Sprint en Gantt Project. Adaptado de: Fábrica de Software - Fundación Cardiovascular de Colombia

Luego de la implementación de la estrategia durante los tres meses de seguimiento al proyecto se encuentre lo siguiente: En efecto, el complemento Gantter permite la visualización cronológica del proyecto permitiendo encontrar las rutas de mayor riesgo para el desarrollo secuencial de las tareas; así mismo al contar con la asignación de recursos permite la revisión de las cargas entre los miembros del equipo. Uno de los inconvenientes encontrados durante esta implementación fue la actualización de la información manualmente lo que demanda tiempo por parte del equipo staff y conocimiento de los Sprint para que no se pierdan las secuencias de las tareas. Por lo tanto se llega a la conclusión que esta estrategia no es la mejor solución para los objetivos planteados y no se implementó como lineamiento en las tareas de planeación y estimación de la Fábrica de Software.

### **6.3.2 Segunda propuesta de intervención**

Con la primera propuesta se logró identificar la utilidad de contar con las fechas asociadas al desarrollo de una tarea lo que permite escalar a diagramas cronológicos, adicional de tener comparativas entre puntos de estimación y fechas reales de producción. Para ello mediante el conocimiento empírico de la herramienta Agilo for Trac; utilizada en la institución para la creación de las tareas se crearon nuevos campos relacionados a cada uno de los tipos de tareas existentes, las cuales son: Nuevo, tarea, mejora y ajuste. Los tres campos propuestos son:

- Fecha Inicio: En este espacio se debe relacionar la fecha en la cual es comenzada. Esta acción debe ser realizada por el dueño (owner) de la tarea.
- Fecha Fin: Fecha calculada de terminación de la tarea según puntos de estimación y velocidad de desarrollo de la persona asignada a la misma. Esta acción debe ser realizada por el equipo Staff.
- Fecha de terminación: Fecha real del cierre de la tarea. Debe ser actualizada por la persona que llevó a cabo la tarea.

El detalle de esta sección es mostrado en la Figura 20.

Properties

Type: Mejora

Summary:

Reporter: oscgom01

Description:

**B I A** You may use [WikiFormatting](#) here.

Priority: Media

Component: NA

Sprint: Febrero2017

Cliente Solicitante:

Scripts SQL:

Milestone: Febrero2017

Version:

TicketSW:

Cliente Aplica:

Figura 20. Estructura tarea en Agilo for Trac del tipo Mejora

The image shows a web form for task management in Agile for Trac. It includes a rich text editor for technical documentation, a grid of input fields for task details, and another rich text editor for data model observations. The form is populated with the following values:

Documentación Técnica:		
MTécnico:	Enlace a SahiHelp. (Dilig)	MUsuario: Enlace a SahiHelp. (Dilig)
Pruebas Unitarias:	NO	Pruebas de Carga: NO
Anexos:	SI Ó NO, En caso que se	Tiempo(horas):
Planeada:	Si	Equipo: Java
Observaciones Modelo Datos:	---No Aplica	
FechaInicio:	01/09/2016 00:00:00.0	FechaFin: 01/09/2016 00:00:00.0
Proyecto:		Storypoints: 0
Date Completed:		Owner:

Figura 21. Fechas agregadas a la estructura de la tarea en Agile for Trac

Para hallar el campo Fecha Fin el equipo Staff debe realizar un cálculo, el cual se explica a continuación:

- Esta tarea se debe realizar luego de la reunión de control de cambios donde se definan las prioridades de las versiones o proyectos a desarrollar.
- Se debe realizar la estimación en puntos de cada una de las tareas.
- Tomar los puntos de la tarea y los puntos estimados de la capacidad de la persona que realizará la tarea, estos puntos dividirlos en 4 de las semanas que componen un sprint.
- Así conociendo las prioridades se toma persona por persona y se va desglosando las tareas a su cargo según las prioridades dadas.

Un caso ejemplo se muestra a continuación: Tarea estimada en 8 puntos (La realizará el tester #2) cuya tarea anterior finaliza el 24 de Mayo.

Si se remite a Tabla 6 es posible observar que: El tester #2 tiene una capacidad estimada de 12 puntos, es decir, realiza 3 puntos por semana, por lo tanto la persona asignada requerirá 2 semanas y media para la realización de la tarea; concluyendo esta tarea para una fecha fin estimada del 10 de Junio. Al realizar este cálculo con cada una de las tareas a desarrollarse durante un sprint se obtiene un cronograma de actividades por versiones como el que se muestra en la **Figura 22**.

PENDIENTE	TERMINADO A TIEMPO	TERMINADO POSTERIOR	NO PROGRAMADO
Martes	Miércoles	Jueves	Viernes
FEBRERO			
	1 Inicio Verificación 2.4.14 (Eduardo Silva, Viviana Mantilla) Inicio Análisis y Desarrollo FAC 1.5.4.1 (Arelí Afanador)	2 Inicio Verificación APL V.1.1.5 (Andrea Ramirez) Inicio - Fin Análisis, Desarrollo, Verificación CXC 1.5.3.4 (Ruben Pico, Anyury Duarte)	3 Fin Desarrollo Ref y Contrarreferencia (Mónica García y Duvan Vargas) Inicio Desarrollo SUM V.1.5.44 (Diego Herrera) Fin Verificación APL V.1.1.5 (Andrea Ramirez)
7 FE: Fin Desarrollo V.2.4.13.5 (Wilmer, Jaime Olate, Juan Luis) Fin Verificación ADP PHP 1.0.0 (Sandra Aguilar)	8 FE: Liberación S1.HCE V.1.5.46 (Carlos Diaz) Continuación Desarrollo V.2.4.15 (Wilmer Muñoz, Juan Luis Jaime) Continuación Verificación ADP V.1.5.30 (Sandra Aguilar) Fin Desarrollo CXC V.1.5.4 (Giobany Cortés) Fin Análisis CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación 2.4.13.5 (Angela Cortes, Viviana Mantilla) Inicio Desarrollo CXC V.1.5.3.5 (Giobany Cortés)	9 Fin Desarrollo CXC V.1.5.3.5 (Giobany Cortés) Inicio Soporte Configuración Parafiscales UGPP (Giobany Cortés)	10 Fin Análisis y Desarrollo Mantenimiento de equipos (Johnatan Nararajo) Fin Verificación FAC V.1.5.4.1 (Andrea Ramirez) Fin Desarrollo SUM V.1.5.44 (Diego Herrera) Fin Desarrollo FAC V.1.5.5 (Ruben Pico) Fin Desarrollo FE 1.1.0.1 (Duvan Vargas) 27/Feb/2017 Fin Plan de Verificación S2.ACF V.2.4.17 (Andrés Gómez) Inicio Verificación CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación Informe Cumplimiento de Agenda por Consulta Externa (Sandra)
14 Entrega tareas 2.4.15 (Juan Luis, Wilmer Muñoz) Fin Verificación CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación Informes Consulta Externa (Sandra Aguilar)	15 Continuación tareas 2.4.16 (Juan Luis, Wilmer Muñoz) Fin Plan Verificación FAC V.1.5.5 (Rubén Pico) Inicio Verificación CXC V.1.5.3.6 (Anyury Duarte)	16 Inicio Desarrollo CNT V.1.4.5 (Giobany Cortés) Continuación Desarrollo PDV V.1.2.0 (Ruben Pico y Diego Herrera) 17/Feb/2017 Inicio Desarrollo ADM V.1.3.26.3 (Diego Herrera) Inicio-Fin Desarrollo CIT V.1.3.8.4 (Diego Herrera)	17 Liberación SUM V.1.5.44 (Andrea Ramirez) Fin Verificación Informes Consulta Externa (Sandra Aguilar) Inicio-Fin Verificación CIT V.1.3.8.4 (Diego Herrera)

Figura 22. Cronograma planeación de sprint. Adaptado de: Fábrica de Software

Durante el transcurso del Sprint el equipo Staff se encarga de actualizar el estado actual del cronograma en donde se utilizan colores para indicar si la tarea se encuentra: Pendiente (rojo), Terminado a tiempo (Verde), Terminado posterior (azul) en este caso se

incluye la fecha de cierre, No programado (Naranja) como se muestra en Figura 23. Seguimiento a cronograma.

PENDIENTE	TERMINADO A TIEMPO	TERMINADO POSTERIOR	NO PROGRAMADO
Martes	Miércoles	Jueves	Viernes
	1 Inicio Verificación 2.4.14 (Eduardo Silva, Viviana Mantilla) Inicio Análisis y Desarrollo FAC 1.5.4.1 (Arelí Afanador)	2 Inicio Verificación APL V.1.1.5 (Andrea Ramírez) Inicio - Fin Analisis, Desarrollo, Verificación CXC 1.5.3.4 (Ruben Pico, Anyury Duarte)	3 Fin Desarrollo Ref y Contrareferencia (Mónica García y Duvan Vargas) 08/Feb/2017 Inicio Desarrollo SUM V.1.5.44 (Diego Herrera) Fin Verificación APL V.1.1.5 (Andrea Ramirez )
7 FE: Fin Desarrollo V.2.4.13.5 (Wilmer, Jaime Olarte, Juan Luis) 8/Feb/2017 Fin Verificación ADP PHP 1.0.0 (Sandra Aguilar) 14/Feb/2017	8 FE: Liberación S1.HCE V.1.5.46 (Carlos Díaz) 27/Feb/2017 Continuación Desarrollo V.2.4.15 (Wilmer Muñoz, Juan Luis Jaime) Continuación Verificación ADP V.1.5.30 (Sandra Aguilar) Fin Desarrollo CXC V.1.5.4 (Giobany Cortés) Fin Análisis CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación 2.4.13.5 (Angela Cortés, Viviana Mantilla) Inicio Desarrollo CXC V.1.5.3.5 (Giobany Cortés)	9 Fin Desarrollo CXC V.1.5.3.5 (Giobany Cortés) Inicio Soporte Configuración Parafiscales UGPP (Giobany Cortés)	10 Fin Análisis y Desarrollo Mantenimiento de equipos (Johnatan Naranjo) Fin Verificación FAC V.1.5.4.1 (Andrea Ramirez) Fin Desarrollo SUM V.1.5.44 (Diego Herrera) Fin Desarrollo FAC V.1.5.5 (Ruben Pico) Fin Desarrollo FE 1.1.0.1 (Duvan Vargas) 27/Feb/2017 Fin Plan de Verificación S2.ACF V.2.4.17 (Andrés Gómez) Inicio Verificación CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación Informe Cumplimiento de Agenda por Consulta Externa (Sandra)
14 Entrega tareas 2.4.15 (Juan Luis, Wilmer Muñoz) Fin Verificación CXC V.1.5.3.5 (Anyury Duarte) Inicio Verificación Informes Consulta Externa (Sandra Aguilar)	15 Continuación tareas 2.4.16 (Juan Luis, Wilmer Muñoz) Fin Plan Verificación FAC V.1.5.5 (Rubén Pico) Inicio Verificación CXC V.1.5.3.6 (Anyury Duarte)	16 Inicio Desarrollo CNT V.1.4.5 (Giobany Cortés) Continuación Desarrollo PDV V.1.2.0 (Ruben Pico y Diego Herrera) 17/Feb/2017 Inicio Desarrollo ADM V.1.3.26.3 (Diego Herrera) Inicio-Fin Desarrollo CIT V.1.3.8.4 (Diego Herrera)	17 Liberación SUM V.1.5.44 (Andrea Ramirez) Fin Verificación Informes Consulta Externa (Sandra Aguilar) Inicio-Fin Verificación CIT V.1.3.8.4 (Diego Herrera)

Figura 23. Seguimiento a cronograma. Adaptado de: Fábrica de Software

### 6.3.3 Tercera propuesta de intervención: Ajuste al proceso de la Fábrica de Software e implementación de herramienta Testlink

Como tercera propuesta se realizó la implementación de una herramienta para la Gestión de Pruebas (verificación de ejecutables) llamada Testlink,. La Fábrica de Software contaba con esta herramienta instalada desde el año 2012 pero esta no había sido implementada al proceso. Como se detalla en [40] los aspectos destacados sobre las herramientas de Gestión de pruebas son:

- Planeación de las pruebas
- Selección y ejecución de los Sprints

- Aplicación de las pruebas
- Gestión de las pruebas
- Métricas de las pruebas

El primer paso para la implementación de la herramienta fue la generación de los casos de prueba relacionados a una versión de uno de los módulos del sistema (Ver **Figura 24**).

TestLink 1.9.7 (Ghost) : admin [admin] [ Mi Configuración | Cerrar la sesión ]

Inicio | Requisitos | Especificación | Ejecutar | Resultados | Gestión de Usuarios | Eventos | [S2] | Proyecto de Pruebas | SAH2.0

Plan de Pruebas : Test Plan SAH2.0 - HCE (Build : Versión 2.4.13.5 - sispro, oncología)

Opciones

Plan de Pruebas: Test Plan SAH2.0 - HCE

Build a ejecutar: Versión 2.4.13.5 - sispro, oncología

Actualizar el Árbol de Navegación después de cada operación:

Exportar plan

Filtros

Expandir árbol | Recoger árbol

SAH2.0 / Test Plan SAH2.0 - HCE (159)(3, 154, 1, 1)

- Historia Clínica Electronica (159)(3, 154, 1, 1)
  - Planes de prueba por proceso (159)(3, 154, 1, 1)
    - Orden de medicamentos (37)(2, 34, 0, 1)
      - Medicamentos no pos (CTC, SISPRO / MIPRES) (37)(2, 34, 0, 1)
        - 52-162:Tarea #12499 Esc 1 OrdMed Hospitalaria no pos proceso #1 apagado y proceso #2
        - 52-38:Tarea #12499 Esc 1 OrdMed Hospitalaria no pos proceso #1 apagado y proceso #2
        - 52-40:Tarea #12499 Esc 2 OrdMed Hospitalaria no pos proceso #1 encendido y proceso #2
        - 52-41:Tarea #12499 Esc 3 OrdMed Hospitalaria proceso #1 apagado y proceso #2 no cre
        - 52-42:Tarea #12499 Esc 4 OrdMed Hospitalaria no pos atención sin régimen en sispro
        - 52-43:Tarea #12499 Esc 5 OrdMed Hospitalaria no pos atención con régimen en sispro
        - 52-44:Tarea #12499 Esc 6 OrdMed Hospitalaria no pos atención con régimen en sispro, g
        - 52-45:Tarea #12499 Esc 7 OrdMed Hosp atención con régimen en sispro, sin #solicitud, c
        - 52-46:Tarea #12499 Esc 8 OrdMed Hosp con régimen en sispro, sin #solicitud, check plat
        - 52-47:Tarea #12499 Esc 9 OrdMed Hosp atención con 2 contratos de diferentes niveles
        - 52-48:Tarea #12499 Esc 10 OrdMed Hosp atención con contrato que no requiere justific
        - 52-49:Tarea #12499 Esc 11 OrdMed Hosp atención con contrato de régimen sispro, sin m
        - 52-50:Tarea #12499 Esc 12 OrdMed ambulatoria no pos proceso #1 apagado y proceso #2
        - 52-51:Tarea #12499 Esc 13 OrdMed ambulatoria no pos proceso #1 encendido y proceso #2
        - 52-52:Tarea #12499 Esc 14 OrdMed ambulatoria proceso #1 apagado y proceso #2 no c
        - 52-53:Tarea #12499 Esc 15 OrdMed ambulatoria no pos atención sin régimen en sispro
        - 52-54:Tarea #12499 Esc 16 OrdMed ambulatoria no pos atención con régimen en sispro
        - 52-55:Tarea #12499 Esc 17 OrdMed ambulatoria no pos atención con régimen en sispro,
        - 52-56:Tarea #12499 Esc 18 OrdMed Amb atención con régimen en sispro, sin #solicitud,
        - 52-57:Tarea #12499 Esc 19 OrdMed Amb con régimen en sispro, sin #solicitud, check pla
        - 52-58:Tarea #12499 Esc 20 OrdMed Amb atención con 2 contratos de diferentes niveles
        - 52-59:Tarea #12499 Esc 21 OrdMed Amb atención con contrato que no requiere justific
        - 52-60:Tarea #12499 Esc 22 OrdMed Amb atención con contrato de régimen sispro, sin m
        - 52-155:Tarea #12499 Esc 23 OrdMed Hospitalaria no pos proceso #1 encendido y proces
        - 52-156:Tarea #12499 Esc 24 OrdMed Ambulatoria no pos proceso #1 encendido y proces
        - 52-164:Tarea #12499 Esc 25 Ord Med Hospitalaria, repetir orden que tenía medicamento

Resumen

Escenario 1 - OrdMed Hospitalaria no pos proceso #1 apagado y proceso #2 apagado

Ruta de Validación:

Inicio / Modulo HCE / Paciente Actual / Orden de Medicamentos

Criterios de Aceptación:

Solicitud de Número de solicitud sispro / mipres para atenciones cuyo régimen de contrato pertenezca a sispro.

Precondiciones

- La constantes del proceso #1 de sispro INSM debe estar apagada (0 = Apagada | 1 = Encendido)
- La constantes del proceso #2 de sispro ISM2 debe estar apagada (0 = Apagada | 1 = Encendido)
- El contrato de nivel 1 de la atención debe necesario registrar justificación de procedimientos (IndJusProc de conContraro, 0 = No requiere | 1 = requiere)

```
select ac.IdContrato, g.IdLista, g.destitem, g.desformula from admAtencionContrato ac
inner join conContrato c on ac.IdContrato = c.IdContrato
inner join genLista g on c.IndRegimen = g.IdItem
where ac.IdAtencion = XXXX and g.IdGrupo = 91
```

Tipo de ejecución : Manual

Entorno de pruebas(plataforma, navegador, versión):

Versión del aplicativo:

Base de datos:

Usuarios de prueba:

Ruta de verificación:

Observaciones:

Tipo de Prueba:

Nº	Pasos	Resultados Esperados	Ejecución
1	1. Ingrese con un médico a la orden de procedimientos, a una atención de un paciente cuyo contrato pertenezca a sispro. Y a u paciente cuyo contrato NO pertenezca a sispro.		Manual
2	2. Seleccione un medicamento no pos.	Al cargarlo a la grilla se deberá desplegar el formulario de justificación de procedimientos no pos para ser Manual diligenciado. (El medicamento se visualiza cargado en la grilla). Si se cancela el CTC, luego da clic en guardar si visualiza el dialogo de confirmación, y da guardar se vuelve a visualizar el CTC para ser diligenciado.	
3	3. Guarde la orden	La información de la orden y de la justificación deberá guardarse correctamente en la tabla hceFormuOrden, hceFormulacionDet, hceJustificaMed (Pruebe con diferentes tipos de procedimientos. Ej: Alto riesgo).	Manual

Notas/Descripción

Resultado

No Ejecutado  
 Pasado  
 Fallado  
 Bloqueado

Figura 24. Plan de pruebas herramienta Testlink. Adaptado de: Fábrica de Software

A continuación se asignaron los casos a un Tester para realizar la ejecución del mismo. Como se muestra en la **Figura 25** la herramienta Testlink permite la revisión al detalle sobre la ejecución de un caso de pruebas para el ejemplo el mismo pasó correctamente con fecha y personas que realizó la prueba.

Resultados del Caso de Prueba en la Build Versión 2.4.13.5 - sispro, oncologia

Notas del Plan de Pruebas

Descripción de la Build

Imprimir    Mostrar el historial completo de ejecuciones    Importar Resultados XML

Suite de Pruebas : Historia Clinica Electronica/ Planes de prueba por proceso/ Orden de medicamentos/ Medicamentos no pos (CTC, SISPRO / MIPRES)/

Caso de Prueba ID S2-162 :: Versión : 1  
Tarea #12499 Esc 1 OrdMed Hospitalaria no pos proceso #1 apagado y proceso #2 apagado -OK  
No hay un Tester asignado

Última Ejecución (cualquier build) - Build : S2.HCE V.2.4.14

Fecha : 22/02/2017 15:49:14 - Testeado por : admin - Build : S2.HCE V.2.4.14 - Estado : Pasado

Última Ejecución (build actual) - Build : Versión 2.4.13.5 - sispro, oncologia

Fecha	Build	Testeado por	Estado	Versión del Caso de Prueba	Archivos Adjuntos	Borrar	Modo de ejecución
10/02/2017 10:50:45	Versión 2.4.13.5 - sispro, oncologia	admin	Pasado	1			

Notas

Figura 25. Ejecución caso de prueba Testlink. Adaptado de: Fábrica de Software

A continuación se remite la respuesta de uno de los Tester la Fábrica sobre la siguiente pregunta: ¿En qué ha contribuido para su trabajo la herramienta Testlink?. : “... “La herramienta Testlink nos ha permitido agilizar nuestro trabajo ya que al llevar a cabo la verificación de una versión solo debemos seleccionar los casos que debemos ejecutar. Así mismo ya no debemos llevar diferentes archivos para la documentación tanto del plan como de la ejecución de las pruebas.”

Una vez comprobaron las ventajas de la herramienta, se debió tomar la decisión sobre en qué punto del proceso era más eficiente llevar a cabo la tarea de crear cada uno de los casos de prueba, esto en relación a cada uno de los actores de una versión. Para ello en conjunto con el equipo Staff se decidió actualizar el proceso de Planeación del Sprint (Ver sección 6.2.2 Planeación de Sprint), en donde una vez realizada la priorización de temas mediante el comité de control de cambios (integración con el Product Owner) se debía realizar la creación de las tareas y posterior documento de sus casos de pruebas en la herramienta Testlink. Las ventajas encontradas sobre la decisión son:

- Para el Analista Funcional, el relacionar los casos de prueba en la creación de la tarea permite globalizar el proceso tratado en el análisis de una tarea, así se profundiza en el detalle sobre el requerimiento que se pasará a un Desarrollador.
- La herramienta Testlink para la gestión de pruebas permite la reutilización de los casos en diferentes versiones, apoyando así el proceso de gestión de la documentación. Anteriormente los planes de prueba quedaban en un archivo de texto asociados a la versión por lo que era rigurosa la revisión y reutilización del mismo.
- Para el Desarrollador, recibir una tarea con los casos de pruebas relacionados permite llevar a cabo la ejecución de los mismos antes de ser entregada la tarea por lo tanto esta debe llegar al Tester con un primer filtro de verificación realizado.
- Para el Tester, contar con una herramienta de gestión de pruebas, permite estandarización de las mismas ya que esta cuenta con una misma estructura para todos los casos, tema que se encontraba abierto a interpretaciones al ser el documento de texto el formato fuente para la consolidación de los planes de prueba.

- La herramienta genera automáticamente los planes de prueba en extensión pdf.

#### **6.3.4 Cuarta propuesta de intervención: Implementación de Herramienta**

##### **SLACK**

Siendo una de las problemáticas identificadas la comunicación entre el Product Owner con el equipo Scrum se implementó una herramienta de comunicación interna llamada SLACK. La cual tiene un comportamiento similar a una red social, en donde se crean temas de comunicación, para el caso de la Fábrica de crearon 3: General, visual basic y Java.

En General se relacionaban los temas tratados en reuniones con el Product Owner, o comunicaciones sobre tomas de decisiones importante que debían ser conocidas por todo el equipo.

En Visual Basic y JAVA se trataban temas de interés para ser compartidas en la comunidad de cada lenguaje de programación; ejemplo la creación de un nuevo Bean del proyecto.

Adicional la herramienta permite etiquetar personas del equipo como se realiza en las redes sociales ejemplo: @angelacortes para personalizar los mensajes; así como marcar temas especiales con el conocido #, ejemplo: El día de hoy se realizará el #cierredesprint.

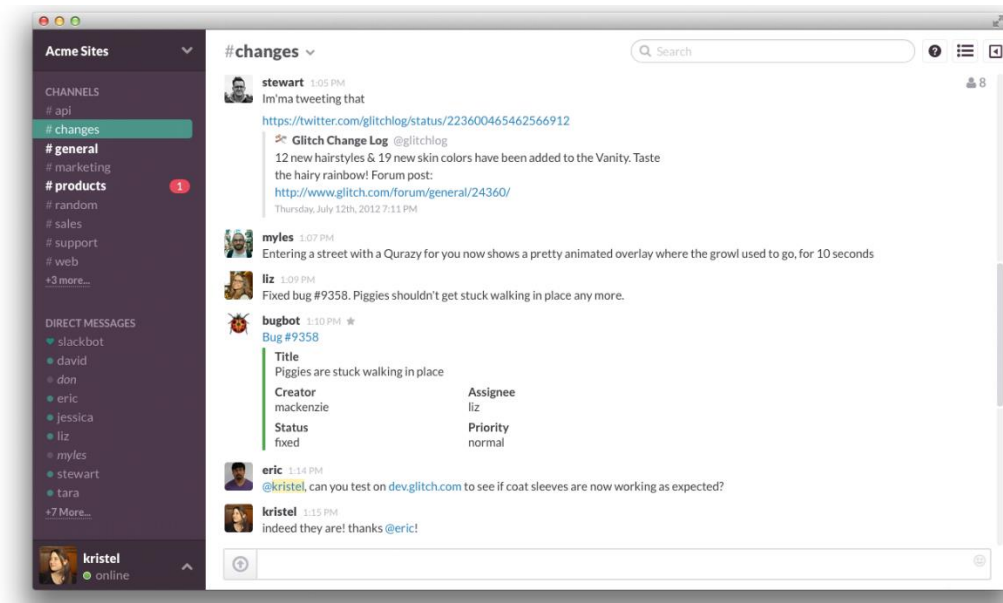


Figura 26. Ejemplo mensaje herramienta SLACK. Adaptado de:

<http://pp.e3.pe/ima/0/0/0/1/3/13883.png>

#### 6.4 Resultados: Validación de la estrategia y medición del impacto.

Finalizada la implementación de las estrategias se realiza la medición de los indicadores de capacidad y cumplimiento; con la intención de contrastar los resultados previos de la intervención con dicho registro. Dichos resultados son presentados en la Tabla 7.

Tabla 7. Resultados generales meses Junio, Julio y Agosto de 2016

Plan Mes	Peso Plan Mes	Cerradas					Peso Cerradas	Plan Cerradas	Capacida d	Cumplimi ento
		Tareas	Nuevos	Mejoras	Ajustes	Total				
123	208	50	3	60	24	137	272	82	90%	67%
109	254	56	0	47	27	130	287	106	100%	97%
113	279	58	5	61	24	148	327	112	100%	99%

En la Figura 27 se presenta la relación de los seis meses de medición del % de **Capacidad** (descrito en 6.2); siendo los primeros tres pre-intervención y los siguientes obtenidos luego de la intervención propuesta por la autora del proyecto. El promedio de los tres primeros meses es: 67%, mientras que en los meses de la intervención se logra una media del 97%. Cabe resaltar que en los últimos dos meses de intervención la capacidad se cuantificó a un 100%; siendo esto un mejoramiento más que relevante en el quehacer de la Fábrica de Software. El otro indicador relevante para la discusión es el porcentaje de cumplimiento, el cual es presentado con detalle en la **Figura 28**. Se observa un incremento de dicho indicador con un efecto de inercia, esperado, de un mes (cuarto mes, primero luego de la intervención, aún no se observa un cambio considerable); sin embargo los meses cinco y seis presentan un incremento destacado, con cambio en promedio de 60% (pre-intervención) al 88% (post-intervención).

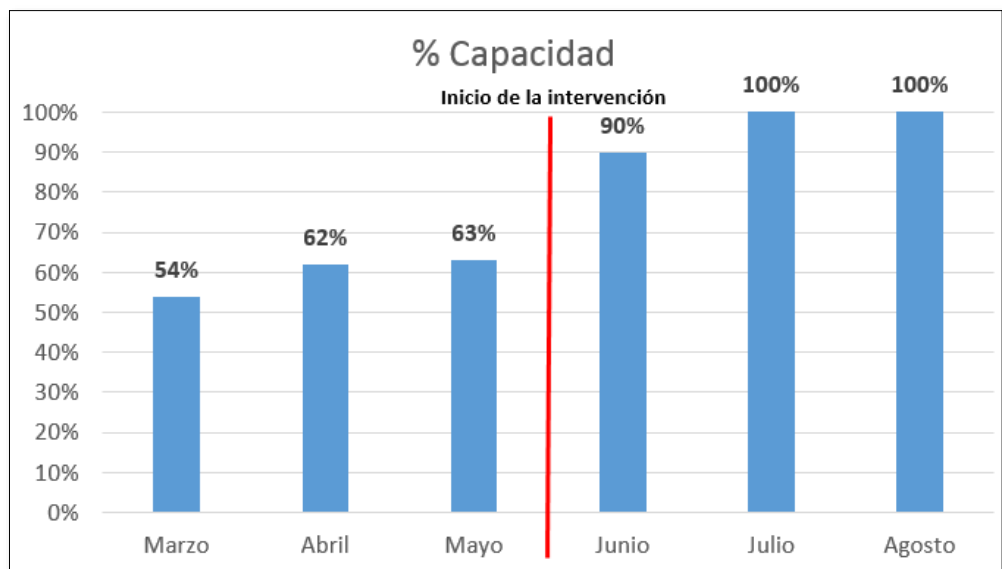


Figura 27. Relación de % de Capacidad por meses (pre y post intervención)

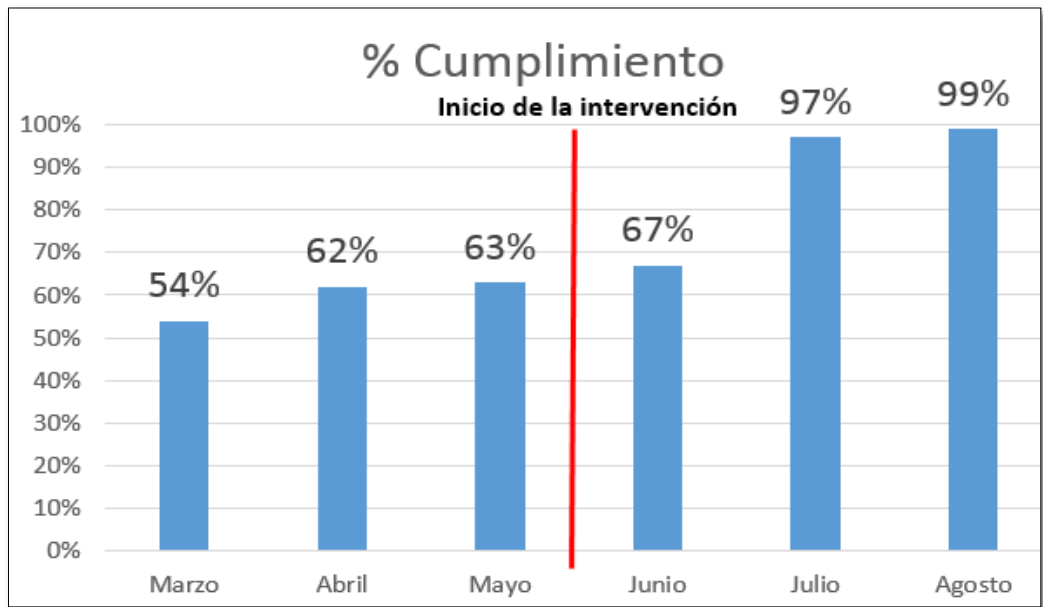


Figura 28. Relación de % de Cumplimiento por meses (pre y post intervención)

Sobre la revisión del impacto se realizó una entrevista al Jefe de la Fábrica de Software a quien se le preguntó: ¿Cuál considera usted que ha sido el impacto sobre la planeación y estimación de las tareas de los lineamientos puestos en marcha durante la ejecución del proyecto de especialización propuestos por la autora? a continuación se registra su respuesta:

Los lineamientos que han sido implementados en la Fábrica han sido positivos, ya que permiten contribución en cuento al equipo; ya que permiten la interacción de cada uno de los integrantes como la integración con los interesados al proceso de desarrollo.

Las mejoras propuesta han permitido incrementar los valores de los 2 indicadores más importantes que dictaminan cual es el manejo dado a la Fábrica y me han permitido presentar de una manera más clara información a la Dirección de Tecnologías. Así mismo se estableció un proceso claro para la planeación y cierre de la Fábrica el cual puede ser realizado por cualquier persona que conforme el equipo Staff.

Estos resultados, si bien motivadores debidos a los grandes cambios cuantitativos; deben seguir siendo monitoreados por el equipo de trabajo; determinando el nivel de afectación con cambios de personal, inicio de otros proyectos donde el recurso humano participe, entre otros.

### **6.5 Lineamientos para el mejoramiento de los procesos de la Fábrica de Software- Caso nuevo proyecto.**

Para presentar de forma concreta los lineamientos que surgen del presente trabajo de investigación, se toma como referencia el inicio de un nuevo proyecto, para el desarrollo de uno o más productos, de la Fábrica de Software. En ese sentido se diseñan las actividades, procesos y roles que integraran las fases de inicio y planificación, Ejecución, Entrega y cierre; que compondrán la estrategia elaborada para la elaboración de artefactos de gestión por parte de la Fábrica de Software necesarios para la correcta gestión y manejo de los proyectos.

### 6.5.1 Fase de inicio y planificación

En esta fase se describe las actividades necesarias para el establecimiento de un proyecto. Una vez el cliente (área de implantación) interno haya realizado la solicitud del nuevo proyecto en la tabla X y Y se

Actividades	<ol style="list-style-type: none"><li>1. Creación de proyecto.</li><li>2. Actas de seguimiento.</li><li>3. Creación de grupo SLACK.</li></ol>
Roles	Jefe de fábrica de Software, analista funcional, arquitecto, personal de implantación, usuarios,

Tabla 8. Proceso 1: Levantamiento de requerimientos

El Jefe de Fábrica o Analista funcional deberá ser el encargado de la **creación del proyecto**, se deberá crear un *Documento de google*, el cual deberá ser compartido a cada una de las personas involucradas en el proyecto; este será la línea base de documentación y se debe ir actualizando según se den las reuniones de levantamiento de requerimientos con los involucrados quien deberán validar la información registrada mediante la aprobación de las actas de seguimiento. Se recomienda en cada iteración del proyecto realizar una descarga de la versión del documento la cual debe quedar resguardada en unas de las carpetas del servidor de archivos que cuente con respaldo.

Se propone que el documento se componga de los siguientes capítulos:

- **Definición:** En esta sección se busca recopilar la necesidad del cliente; cuya comprensión permite al equipo tener una idea clara sobre la solución a desarrollar. Para establecer con claridad la problemática se recomienda realizar un árbol de problemas donde junto con el cliente se identifiquen las causas y efectos en relación a la necesidad del cliente.
- **Actividades:** Se deben listar las actividades que debe llevar a cabo la Fábrica para el desarrollo del proyecto. Ejemplos de estas actividades son: Separación de modelo de datos, arquitectura de software, construcción de mockups, modelamiento de procesos, entre otras.
- **Funcionalidades:** Definición de las funcionalidades que se implementarán en el proyecto.
- **Integración entre las funcionalidades:** En esta sección se deben definir roles, permisos, diagramas de proceso que permitan la comprensión de las funcionalidades.
- **Desarrollo del proyecto:** Se deben definir cada una de las tareas (especificaciones de desarrollo del proyecto) que se deben llevar a cabo para el desarrollo del proyecto; estas en base a los lineamientos de las historias de usuario que propone SCRUM. Las cuales luego serán gestionadas desde las herramientas Agilo for Tract y Testlink con la que cuenta la Fábrica.

En la **Tabla 9. Plantilla tareas de proyecto** se presenta una plantilla de tarea:

<b>Campo</b>	<b>Descripción</b>
Título de la tarea	El formato utilizado es: Tarea #XXX – PRO.NOM Nombre de la tarea. El campo XXX será el identificador obtenido

	desde la herramienta agilo, el campo PRO.NOM es la etiqueta corta establecida para llamar al proyecto, el nombre de la tarea debe contener el propósito de la misma (sin ambigüedades) para una fácil búsqueda y trazabilidad de la misma.
Requerimientos funcionales y no funcionales.	Descripción o lista de los requerimiento que componen la tarea.
Diagrama de proceso	Figura con el proceso que interactúa en la funcionalidad que se implementará en la tarea. Este debe ser modelado en la herramienta Enterprise Architect.
Mockup	Si se requiere diseño de interfaz recibido por parte del diseñador de producto de software.
Modelo de datos	Diagrama de modelo de datos. Se podrán anexar scripts para la creación o actualización del modelo de datos.
Configuraciones necesarias	Definición de las configuraciones o condiciones establecidas en el sistema para el desarrollo de la tarea.
Escenarios de Verificación	Documentar cada uno de los escenarios que se deben probar en la etapa de verificación del producto. Estos escenarios se deberán ingresar en la herramienta Testlink.

*Tabla 9. Plantilla tareas de proyecto*

Se deben realizar reuniones de seguimiento del proyecto donde se atenderán temas como levantamiento de requerimientos, revisión de temas críticos, entre otros cada semana máximo 15 días entre cada reunión. Una vez culminada el secretario encargado de levantar la información de la reunión deberá presentar un resumen sobre los temas tratados en la herramienta SLACK (previamente creado un grupo del proyecto por el Jefe de la fábrica de software donde se incluyen los interesados). Las actas de seguimiento deben ser realizadas por el equipo de Staff, resumiendo los temas más importantes de la reunión; resultando los cambios que se pueden haber generado sobre la definición de requerimientos. Estas actas deberán ser socializadas vía correo electrónico para la aprobación y verificación de la información contenida por parte de todos los interesados.

<b>Proceso 2: Planeación de proyecto</b>	
Actividades	<ol style="list-style-type: none"> <li>1. Comité de control de Cambios.</li> <li>2. Estimación y programación</li> </ol>
Roles	Jefe de fábrica de Software, analista funcional, arquitecto, personal de implantación.

*Tabla 10. Proceso 2: Planeación de proyecto*

En el proceso 2 propuesto se ejecutaran las actividades correspondientes a la planeación del proyecto para ello luego de la reunión de inicio del proyecto y retroalimentación de la información el equipo Staff. La Fábrica de Software debe llevar a cabo una reunión denominada Comité de Control de Cambios (la cual debe ser llevada a

cabo una semana antes de cada inicio de Sprint) en donde participará el equipo Staff y el personal de implantación; esta reunión deberá ser liderada por el Jefe de implantación quien es el responsable de conocer las necesidades de los usuarios y las prioridades de los desarrollos que deben ser llevados a cabo por la Fábrica de Software. En esta reunión se deben revisar el cierre del mes anterior para conocer los porcentajes de cumplimiento de cada uno de los proyectos, y revisar cada uno de los temas y requerimientos de los usuarios; el Jefe de implantación deberá exponer la criticidad de cada uno de los requerimientos y si se deben desarrollar en el Sprint entrante.

Luego del Comité de Control de Cambios el equipo Staff deberá realizar la estimación y programación de las actividades que deberán ser realizadas durante el Sprint a comenzar. La estimación de actividades permitirá conocer si todas las actividades solicitadas por el equipo de implantación podrán ser llevadas a cabo, identificar el personal que llevará a cabo cada una de las tareas y así llegar a término con el cumplimiento de los proyectos.

La estimación se debe realizar de la siguiente manera:

1. Se deben listar cada uno de los requerimientos, los cuales se dividirán según la línea de producción Médico - Asistencial, Corporativo – Financiero; cada uno de estos deberá asociarse a una nueva versión del módulo o proyecto al cual pertenece.
2. El Arquitecto y Analista funcional líderes de cada uno de los equipos deberán realizar una reunión con el personal acorde a los requerimientos y se deben estimar cada uno de los requisitos utilizando la metodología propuesta por SCRUM [41]. Al finalizar esta

reunión cada equipo contará con el listado de requerimientos y sus puntos estimados de desarrollo o testing.

3. El Jefe de la Fábrica junto con los analistas funcionales deberá realizar la programación de los ejecutables según las versiones o proyectos definidos en el primer punto. Esto se deberá realizar según la criticidad identificada en el Comité de Control de Cambios; para ello por proyectos se deberá relacionar un “owner” o encargado a cada una de las tareas luego se deberá proponer una fecha de inicio y para cada una de las personas involucradas según sus puntos y velocidad de realización de tareas (esta velocidad se conoce gracias al indicador mensual de puntos realizados por personas) definir la fecha en la cual termina su participación en el ejecutable; la mayor fecha será la de finalización del ejecutable (la finalización puede ser interna o externa ya que el ejecutable puede pasar a testing). Estas fechas deberán asignarse en un cronograma en el documento compartido de google de nombre *Estimación Fábrica de Software*. De resultado se obtendrá un cronograma como el presentado en la **Figura 22. Cronograma planeación de sprint**.
4. El Jefe de la Fábrica de Software deberá socializar el cronograma al Jefe de implantación y este podrá plantear algunas modificaciones. Una vez dado su visto bueno. El equipo Staff deberá realizar una socialización a todo el equipo de trabajo de la Fábrica de Software ya que este cronograma plasma los compromisos adquiridos por el equipo.

### **6.5.2 Fase de Ejecución**

En esta fase son llevadas cabo cada una de las tareas relacionadas a proyecto a desarrollar; así como el seguimiento de la planeación realizada. Gracias a los daily meeting de la metodología SCRUM el equipo de desarrollo diariamente socializará lo realizado, lo pendiente y aquellos detalles que pueden obstaculizar el trabajo del equipo. Según el progreso el Jefe de la Fábrica deberá actualizar el cronograma y deberá recordar las metas según las fechas propuestas.

En esta fase también los analistas funcionales deberán realizar la socialización y reconocimiento de los proyectos a verificar a los Analistas Tester quienes deberán identificar los escenarios a revisar en la plataforma TestLink. Este es el punto para resolver dudas previas a la ejecución de los escenarios de prueba.

En este punto es importante el apoyo por parte del equipo Staff hacia los Desarrolladores y Analistas Tester para agilizar la ejecución de las tareas asignadas y la resolución de conflictos o ambientes de inseguridad.

### **6.5.3 Fase de Entrega y Cierre**

Esta fase es llevada a cabo una vez se haya ejecutado todos los Sprint de desarrollo y se encuentren validadas cada una de las versiones que componen un proyecto de desarrollo.

Como cierre se deberá realizar una reunión de entrega en donde participe el personal de implantación y los Analistas Funcional y Tester de la fábrica de software que hicieron

parte del proyecto (también se podrá incluir la participación del Arquitecto de Software si el proyecto requirió un componente crítico en este ámbito).

En esta socialización se deberán exponer cada uno de la documentación generada por la Fábrica: documento del proyecto, manual de usuario, manual de actualización de versión y toda la documentación necesaria para la comprensión del proyecto en la cual se podrá basar el equipo al realizar la socialización e implantación del proyecto al usuario final.

El cierre de un proyecto se deberá realizar mediante un acta de cierre la cual se deberá disponer en la carpeta donde se encuentren todos los documentos relacionados al proyecto.

**6.6 Estrategias propuestas no implementadas del proyecto a la FCV. (Ideas futuro)**

Adicional a las propuestas anteriormente mencionadas y validadas; se presentó la propuesta sobre la creación de un perfil o rol dentro de la Fábrica de Software, como **Gestor de Proyectos de Software**. El perfil propuesto por el autor del proyecto es presentado en la **¡Error! No se encuentra el origen de la referencia..**

<b>Habilidades cualitativas</b>	<b>Habilidades profesionales</b>	<b>Responsabilidades</b>
Comunicación Asertiva Persona extrovertida que pueda comunicarse en público Organizado, que pueda realizar actividades de documentación y redacción.	Profesional en Ingeniería de sistemas, telecomunicaciones o afines. Certificación SCRUM MASTER Especialización en Gerencia de proyectos o afines. Contar con experiencias en	Documentación de los proyectos llevados a cabo por la Fábrica de Software. Actualización de las herramientas donde se lleve el control de los proyectos. Apoyo al equipo Staff sobre el control de la ejecución de

	desarrollo de proyectos ágiles	pruebas de verificación. Realización de actas de constitución, seguimiento y cierre de proyectos.
--	--------------------------------	------------------------------------------------------------------------------------------------------

*Tabla 11. Perfil nuevo integrante propuesto.*

## 7. CONCLUSIONES

- La implementación de cualquier metodología de gestión y administración de proyecto, dentro de un entorno organizacional específico, se debe evaluar en cuanto a cuáles son las herramientas y lineamientos que aportaran ventajas en las tareas realizadas.
- En el proceso de implementación de una estrategia de gestión de proyectos la comunicación entre los diferentes actores es fundamental, propendiendo por el éxito y la efectividad de la implementación. Se debe involucrar a todas las personas del equipo realizando comunicaciones asertivas para dar a entender el alcance de las herramientas que se quieran implementar y que estas no sean vistas como una carga laboral.
- La implementación de herramientas que integran procedimientos debidamente descritos y soluciones tecnológicas efectivas (como las propuestas en la intervención de este proyecto) permitieron el mejoramiento de los indicadores seleccionados para la evaluación del desempeño de la fábrica de software. En algunos casos esta mejora se produjo en más de 30 puntos porcentuales, lo cual es un resultado muy satisfactorio de la iniciativa.
- Al implementar una herramienta además de ser puesta en marcha se debe identificar el punto del proceso donde esta interviene para tomar mayor provecho de la misma. El seguimiento y actualización de los datos debe ser preciso ya que se puede acumular las actividades y luego ser torna difícil la interpretación de los datos.

- En la presente propuesta, fue una gran ventaja que la autora del proyecto participara activamente en los procesos de la Fábrica de software de la FCV; pues permitió contar con información precisa y a tiempo, así como en la aprobación del desarrollo de cada intervención y el apoyo de la alta dirección organizacional para la ejecución del proceso y la validación de resultados.

## 8. BIBLIOGRAFÍA

- Fundación Cardiovascular de Colombia , «Indicador actividades planeados vs. actividades estimadas,» Floridablanca, 2015.
- 1] actividades estimadas,» Floridablanca, 2015.
- Ágril Ibérica, «¿Qué hacemos con recursos humano en Agile?,» 17 Febrero 2013. [En línea]. Available: <http://agilib.org/2013/02/17/que-hacemos-con-recursos-humanos-en-agile/>. [Último acceso: 10 Octubre 2015].
- 2] 2013. [En línea]. Available: <http://agilib.org/2013/02/17/que-hacemos-con-recursos-humanos-en-agile/>. [Último acceso: 10 Octubre 2015].
- Aprende y piensa, «Prototipado Ágil y Metodología #SCRUM. Seminario Avanzado #innovape,» 07 Septiembre 2014. [En línea]. Available: <http://www.aprendeypiensa.com/2014/09/prototipado-agil-y-metodologia-scrum.html>. [Último acceso: 10 Octubre 2015].
- 3] Avanzado #innovape,» 07 Septiembre 2014. [En línea]. Available: <http://www.aprendeypiensa.com/2014/09/prototipado-agil-y-metodologia-scrum.html>. [Último acceso: 10 Octubre 2015].
- J. C. González, «Lider de proyecto,» 2015. [En línea]. Available: [http://www.liderdeproyecto.com/articulos/fase\\_conceptual\\_metodologia\\_administracion\\_proyectos\\_software.html](http://www.liderdeproyecto.com/articulos/fase_conceptual_metodologia_administracion_proyectos_software.html). [Último acceso: 10 10 2015].
- 4] [http://www.liderdeproyecto.com/articulos/fase\\_conceptual\\_metodologia\\_administracion\\_proyectos\\_software.html](http://www.liderdeproyecto.com/articulos/fase_conceptual_metodologia_administracion_proyectos_software.html). [Último acceso: 10 10 2015].
- Project Management Institute, Fundamentos de la Dirección de Proyectos (Tercera Edición), 2004.
- 5] (Tercera Edición), 2004.
- S. Blanco-Cuaresma, «Marble Station,» 28 Mayo 2008. [En línea]. Available: <http://www.marblestation.com/?p=657>. [Último acceso: 01 10 2015].
- 6] <http://www.marblestation.com/?p=657>. [Último acceso: 01 10 2015].
- InteracTIC, ITIL - Mejores Practicas, Bogotá: Centro de investigaciones de las Telecomunicaciones CINTEL, 2008.
- 7] Telecomunicaciones CINTEL, 2008.

P. L. y. M. C. P. José H. Canós, «Metodologías Ágiles en el Desarrollo de  
8] Software,» *Universidad Politécnica de Valencia*, p. 8.

S. Blanco-Cuaresma, «Metodologías ágiles de gestión de proyectos (Scrum,  
9] DSDM, Extreme Programming - XP,...),» 7 Junio 2008. [En línea]. Available:  
<http://www.marblestation.com/?p=661>. [Último acceso: 01 10 2015].

Proyecto Agiles, «Qué es SCRUM,» [En línea]. Available:  
10 <http://proyectosagiles.org/que-es-scrum/>. [Último acceso: 01 10 2015].

]

C.-A. Vásquez, «Glosario SCRUM,» 03 Marzo 2014. [En línea]. Available:  
11 <http://www.buenastareas.com/ensayos/Glosario-Scrum/48191679.html>. [Último acceso:  
] 01 10 2015].

eduScrum, La Guía de eduScrum - Las reglas del juego, eduScrum, 2013.

12

]

Universidad Unión Bolivariana, «Ingeniería de Software,» [En línea]. Available:  
13 [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html).

]

G. Montero, «Ideas sencillas para la gestión,» 16 Mayo 2012. [En línea].  
14 Available: [http://www.ideassencillas.com/2012/05/la-historia-de-la-gestion-de-](http://www.ideassencillas.com/2012/05/la-historia-de-la-gestion-de-proyectos.html)  
] [proyectos.html](http://www.ideassencillas.com/2012/05/la-historia-de-la-gestion-de-proyectos.html). [Último acceso: 09 2016].

D. Haughey, «Líder de Proyecto,» 2012. [En línea]. Available:  
15 [http://www.liderdeproyecto.com/manual/breve\\_historia\\_sobre\\_la\\_administracion\\_de\\_p](http://www.liderdeproyecto.com/manual/breve_historia_sobre_la_administracion_de_p)  
] [royectos.html](http://www.liderdeproyecto.com/manual/breve_historia_sobre_la_administracion_de_p). [Último acceso: 09 2016].

C. Ardillas, «Historia de la Administración de Proyectos. Línea del Tiempo.,» 5  
16 Septiembre 2014. [En línea]. Available: [https://prezi.com/yn4ypu7sqzby/historia-de-la-](https://prezi.com/yn4ypu7sqzby/historia-de-la-administracion-de-proyectos-linea-del-tiempo/)  
] [administracion-de-proyectos-linea-del-tiempo/](https://prezi.com/yn4ypu7sqzby/historia-de-la-administracion-de-proyectos-linea-del-tiempo/). [Último acceso: 09 2019].

O. -. B. School, «GanttProject: Análisis del Software,» 2016. [En línea].  
17 Available: [http://www.obs-edu.com/int/blog-project-management/diagramas-de-](http://www.obs-edu.com/int/blog-project-management/diagramas-de-gantt/ganttproject-analisis-del-software)  
] [gantt/ganttproject-analisis-del-software](http://www.obs-edu.com/int/blog-project-management/diagramas-de-gantt/ganttproject-analisis-del-software). [Último acceso: 09 2016].

jjvelasco, «Gantter, una interesante aplicación para gestionar proyectos,» 22  
18 Junio 2012. [En línea]. Available: [https://hipertextual.com/archivo/2012/06/gantter-](https://hipertextual.com/archivo/2012/06/gantter-gestor-de-proyectos/)  
] [gestor-de-proyectos/](https://hipertextual.com/archivo/2012/06/gantter-gestor-de-proyectos/). [Último acceso: Septiembre 2016].

FREELANCERS, «8 Herramientas para la gestión de proyectos profesionales,»  
19 29 Junio 2016. [En línea]. Available: [https://www.lancetalent.com/blog/8-herramientas-](https://www.lancetalent.com/blog/8-herramientas-para-la-gestion-de-proyectos-profesionales/)  
] [para-la-gestion-de-proyectos-profesionales/](https://www.lancetalent.com/blog/8-herramientas-para-la-gestion-de-proyectos-profesionales/). [Último acceso: Septiembre 2016].

Similar.io, «Agilo for Trac,» 2016. [En línea]. Available:  
20 <http://www.similar.io/es/product/Agilo-for-Trac/17028>. [Último acceso: Septiembre  
] 2016].

A. Garcia, «<http://sinapsis.agency/herramienta-de-gestion-de-proyectos-trello/>,»  
21 6 Febrero 2014. [En línea]. [Último acceso: Septiembre 2016].

]

E. Hasnain, «An overview of published agile studies: A systematic literature  
22 review,» *Proceedings of the national software engineering conference*, pp. 1-6, 2010.

]

T. Silva, A. Martin y S. M., «User-centered design and agile methods: A  
23 systematic review.,» de *Agil. conf*, 2011.

]

L. Cao y B. Ramesh, «Agile Requirements engineering practices. An empirical  
24 study,» *IEEE Software*, pp. 60-67, 2008.

]

I. Inayat, S. M. Salwah, M. Daneva y S. Shamshirband, «A systematic literature  
25 review on agile requirements engineering practices and challenges,» *Computers in Human  
] Behavior*, p. 915–929, 2015.

M. Daneva, E. Van der Veen, C. Amrit, S. Ghaisas, K. Sikkel, R. Kumar y N.  
26 Ajmeri, «Agile requirements prioritization in large-scale outsourced system projects:  
] An empirical study,» *Journal of Systems and software*, pp. 1333-1353, 2013.

A. De Lucia y A. Qusef, «Requirements engineering in agile software  
27 development,» *Journal of emerging technologies in web intelligence*, pp. 308-313, 2010.

]

B. Haugset y T. Stalhane, «Automated acceptance testing as an agile

28 requirements engineering practice,» de *45th Hawaii international conference on system*  
] *science*, 2012.

N. Ernst, A. Borgida, I. Jureta y Mylopuolos, «Agile requirements engineering  
29 via paraconsistent reasoning,» *Information System*, pp. 1-17, 2013.

]

N. Abdulla, S. Honiden, H. Sharp, B. Nuseibeh y D. Notkin, «Communication  
30 patterns of agile requeriments engineering,» de *Proceedings of the 1st workshop on*  
] *agile requirements engineering*, 2011.

E. Wolfgang, «Working with user stories,» de *Agile requirements engineering*  
31 *workshop*, 2011.

]

E. Bjarnason, K. Wnuk y B. Regnell, «A case study on benefits and side-effects  
32 of agile practices in large-sacle requirements engineering,» *Agile RE*, 2011.

]

J. Binder, L. Aillaund y L. Schilli, «The project management cocktail model: An  
33 approach for balancing agile and ISO 21500,» *Procedia - Social and Behavioral*  
] *Sciences*, vol. 119, pp. 182-191, 2014.

M. Spundak, «Mixed agile/traditional project management methodology –  
34 reality or illusion?,» de *27th IPMA World Congress*, 2013.

]

A. Zapata, L. Bastidas y R. Rojas, «Marco ágil para PMI en Pequeñas y  
35 Medianas empresas de desarrollo de software,» Universidad San Buenaventura, Cali,  
] 2014.

E. Guzman, «Methodological proposal using SCRUM and PMBOK, for the  
36 Project Management Technologic in the Office Informatics in a Unit Executing the  
] Sector Transport,» UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS, 2015.

S. Team, «SCRUM GUIDES - The Sprints,» Julio 2016. [En línea]. Available:  
37 <http://www.scrumguides.org/scrum-guide.html#events-sprint>. [Último acceso:  
] Septiembre 2016].

Enciclopedia de Economía, 2009. [En línea]. Available:  
38 <http://www.economia48.com/spa/d/capacidad-de-produccion/capacidad-de->  
] [produccion.htm](http://www.economia48.com/spa/d/capacidad-de-produccion/capacidad-de-produccion.htm).

M. P. L. E. M. y. A. G. Lornel Rivas, «Selección de Herramientas para la  
39 Gestión de Proyectos de Software en Pequeñas y Medianas Empresas,» *Departamento*  
] *de Procesos y Sistemas, Universidad Simón Bolívar, Caracas Venezuela*.

F. E. G. Ramírez, «Testing Ágil de Software con Herramientas libres y abiertas,»  
40 *Revista Científica TEKNOS*, vol. 7, nº 2, p. 9, Diciembre 2011.  
]

ScrumSense, Un mejor Scrum, 2009.

]

M. d. I. P. Social, «Resolución No. 1403,» Colombia, 2007.

42

]

H. E. E. Aguirre, «Códigos de Barras, Usos y Aplicaciones,» Ciudad de México,

43 2003.

]

Fundación Cardiovascular de Colombia, «Portal FCV,» 2013. [En línea].

44 Available: <http://www.fcv.org/site/fcv/inicio/fcv-quienes-somos/mision-y-vision>.

]