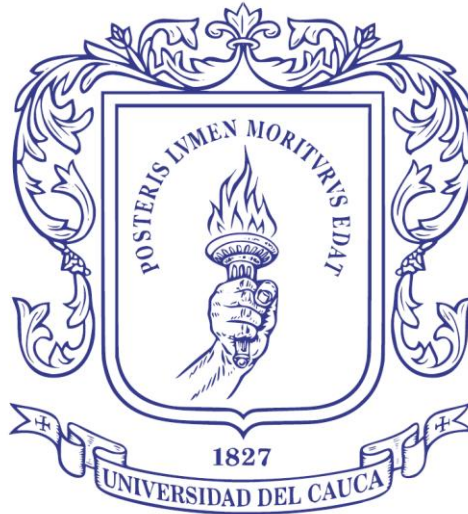


**PROPUESTA DE UN PROCESO DE DESARROLLO DE SOFTWARE
PARA UN FREELANCE, AUDACIOUS FREELANCE**

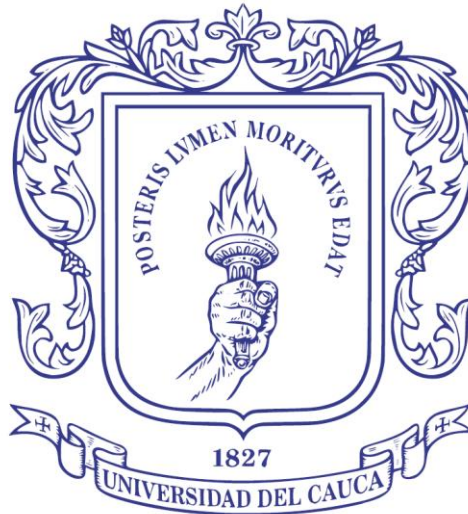


UNIVERSIDAD DEL CAUCA

Juan Camilo Cerón Estrada

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Línea de Investigación en ingeniería de procesos
Departamento de Ingeniería de Sistemas
Grupo IDIS
Popayán
Marzo de 2015**

PROPUESTA DE UN PROCESO DE DESARROLLO DE SOFTWARE PARA UN FREELANCE, AUDACIOUS FREELANCE



UNIVERSIDAD DEL CAUCA

Juan Camilo Cerón Estrada

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director:

Ph.D Julio Ariel Hurtado

Codirector:

Ph.D Francisco José Pino

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Línea de Investigación en ingeniería de procesos

Departamento de Ingeniería de Sistemas

Grupo IDIS

Popayán

Marzo de 2015

AGRADECIMIENTOS

Quiero agradecer a mi familia por motivarme a comenzar con este trabajo de grado y ayudarme a encontrar el camino que, en varias ocasiones perdí cuando la frustración me encontraba. Agradecer a mis directores quienes me dejaron trabajar a mi ritmo y a mi manera, muchas veces siguiendo mis caprichos, pero siempre estaban en disposición de brindarme un constante acompañamiento y la guía que me permitiría realizar un trabajo satisfactorio. También, agradecer a mis compañeros y amigos, Cesar Iván Zambrano y Juan Manuel Chagüendo, con quienes compartí desde inicio de la carrea hasta el final, pues ellos también corrieron junto a mí para acabar el proyecto de grado a tiempo, ayudándonos y motivándonos en el trascurso. Agradecer a María Camila Trujillo por estar siempre pendiente de este proyecto y ayudarme con correcciones. Y por último agradecer a mis amigos y socios Rúbi Bautista y Jhon Piñeros, quienes sacaron tiempo de donde no tenían para ayudarme con la evaluación del proyecto.

Juan Camilo Cerón

ÍNDICE GENERAL

CAPÍTULO 1. INTRODUCCIÓN	11
1.1. Contexto	11
1.2. Problema	11
1.3. Justificación	12
1.4. Objetivos	13
1.5. Metodología y descripción del trabajo	13
CAPÍTULO 2. MARCO CONCEPTUAL Y ESTADO DEL ARTE	15
2.1. Marco conceptual	15
2.1.1. Freelance	15
2.1.2. Proceso de desarrollo de software	16
2.1.3. Métodos ágiles de desarrollo de software	18
2.2. Estado del arte	18
2.2.1. Propuestas de procesos de software para un solo desarrollador	19
a. PSP	19
b. SIP	20
c. PCSE	20
d. PXP (Dzhurov)	21
e. PXP (Agarwal)	21
f. eXtreme Solo	22
g. AGILE SOLO	22
h. Otros aportes	22
2.2.2. Análisis	23
CAPÍTULO 3. ANÁLISIS Y CARACTERIZACIÓN DEL FREELANCE	24
3.1. Diseño del estudio	24
3.1.1. Estructura del cuestionario	26
3.2. Resultados y análisis	27
3.2.1. Aspectos personales	27
3.2.2. Experiencia como freelance	27
3.2.3. Situaciones comunes	28
3.2.4. Ideales como desarrollador	29

3.2.5. Conocimientos sobre procesos software	30
3.2.6. Análisis	32
3.3. Estudios relacionados.....	32
3.3.1. Reporte de la industria Freelance.....	32
3.3.2. Los conductores para el éxito en la carrera Freelance.....	33
3.4. Perfil del Freelance.....	33
3.4.1. Características personales	33
3.4.2. Características del contexto.....	34
3.4.3. Necesidades	34
a. Seguimiento al esfuerzo y costo de proyectos	35
b. Procesos de desarrollo flexible y ligero.	35
c. Gestionar el capital humano	35
d. Herramienta de gestión.....	35
e. Mercadeo y promoción de su trabajo.....	36
CAPÍTULO 4. DISEÑO DEL PROCESO AUDACIOUS FREELANCE.....	37
4.1. Meta proceso de formalización.....	37
4.1.1. Fase de planeación	37
4.1.2. Fase de ejecución.....	38
4.1.3. Fase de entrega.....	39
4.2. Criterios para el diseño del proceso	40
4.3. Modelos de referencia	41
4.3.1. Aporte de Scrum a Audacious Freelance	42
4.3.2. Aporte de eXtreme Programming a Audacious Freelance	44
4.4. Diseño iterativo	46
4.4.1. Audacious Freelance versión 0.5.....	46
4.4.2. Audacious Freelance versión 0.8.....	47
4.4.3. Audacious Freelance versión 1.0.....	48
4.4.4. Audacious Freelance versión 2.0.....	49
CAPÍTULO 5. AUDACIOUS FREELANCE – AF.....	52
5.1. Introducción	52
5.2. Enfoque	53
5.3. Propósito.....	53

5.4. Características.....	53
5.5. Filosofía	53
5.6. Las tres dimensiones.....	54
a. Desarrollo de software	54
b. Gestión de proyectos	54
c. Gestión del negocio	54
5.7. Artefactos.....	54
a. Lista de tareas	54
b. Lista de investigación.....	55
c. Lista de conocimiento	55
d. Productos de trabajo.....	55
5.8. Roles.....	55
a. Freelance Desarrollador	55
b. Freelance Administrador de proyectos	55
c. Freelance Empresario.....	56
d. Cliente.....	56
5.9. Ciclo de vida	56
5.10. Especificación del proceso – EPF	57
5.11. Diseño de instrumentos de soporte	58
5.11.1. Documentación web	59
5.11.2. Guía de Audacious Freelance	59
5.11.3. Audacious Freelance Dashboard - AFD	60
CAPÍTULO 6. EVALUACIÓN DEL PROCESO <i>AUDACIOUS FREELANCE</i>.....	64
6.1. Introducción	64
6.2. Ficha del estudio de caso	64
6.3. Contexto	66
6.3.1. Pregunta de Investigación	66
6.4. Diseño.....	67
6.5. Selección del caso.....	68
6.6. Procedimiento.....	69
6.7. Recopilación de datos	69
6.7.1. Indicadores y mediciones	70

6.8. Resultados y Análisis.....	71
6.8.1. Aspecto Calidad.....	72
6.8.2. Aspecto productividad.....	73
6.8.3. Aspecto Competitividad	74
6.8.4. Aspecto facilidad de adopción de AF.....	74
6.8.5. Respuesta a la pregunta de investigación	76
6.8.6. Sugerencias, ideas y problemas.....	76
6.9. Amenazas de validez.....	77
CAPÍTULO 7. CONCLUSIONES, LECCIONES APRENDIDAS, LIMITACIONES Y TRABAJOS FUTUROS.....	78
7.1. Conclusiones	79
7.2. Lecciones aprendidas.....	79
7.3. Limitaciones.....	80
7.4. Trabajos futuros.....	80
BIBLIOGRAFÍA.....	81

ÍNDICE DE FIGURAS

Figura 1: Satisfacción del comprador en proyectos de software Freelance	12
Figura 2: Elementos de un proceso de software	17
Figura 3: Proceso para diseño de cuestionarios online por Lumsden	24
Figura 4: How many years of experience do you have with IT projects?	28
Figura 5: How common are these situations in IT projects?	29
Figura 6: What aspects do you prioritize when you have a project?	30
Figura 7: Do you use a formal methodologies/processes for your work?	31
Figura 8: Do you know about this software development methodologies/processes/frameworks?	31
Figura 9: Fase de planeación del meta proceso.....	37
Figura 10: Fase de ejecución del meta proceso.....	39
Figura 11: Fase de entrega del meta proceso.....	40
Figura 12: Dimensiones de Audacious Freelance	40
Figura 13: Ciclo de vida de Scrum.....	42
Figura 14: AF v. 0.5 Desarrollo de software	46
Figura 15: AF v 0.8 (a) Gestión de proyectos, (b) Desarrollo de software, (c) Tareas parte 1, (d) Tarea parte 2.....	47
Figura 16: AF v 1.0. (a) Gestión de proyectos, (b) Desarrollo de software, (c) Tareas parte 1, (d) Tarea parte 2.....	48
Figura 17: AF v. 2.0 Gestión del negocio	49
Figura 18: AF v. 2.0 Gestión de proyectos	50
Figura 19: AF v. 2.0 Desarrollo de software - Fase pre-juego.....	51
Figura 20: AF v. 2.0 Desarrollo de software – Fase juego	51
Figura 21: AF v. 2.0 Desarrollo de software - Fase post-juego	51

Figura 22: Elementos de Audacious Freelance.....	52
Figura 23: Ciclo de vida de Audacious Freelance	56
Figura 24: EPF Method Framework.....	57
Figura 25: Arquitectura de Audacious Freelance en EPF	58
Figura 26: Logo de Audacious Freelance.....	59
Figura 27: Instrumento de soporte de Audacious Freelance - Documentación web	59
Figura 28: Audacious Freelance Dashboard - Gestión Lista de tareas	60
Figura 29: Audacious Freelance Dashboard - Gestión Lista de investigación	61
Figura 30: Audacious Freelance Dashboard - Gestión Lista de conocimiento.....	61
Figura 31: Audacious Freelance Dashboard - Compartir en redes sociales	62
Figura 32: Audacious Freelance Dashboard - Inicio.....	62
Figura 33: Audacious Freelance Dashboard - Reportes de proyecto.....	63
Figura 34: Audacious Freelance Dashboard - Reporte grafico de desempeño.....	63
Figura 35: ¿Qué métricas de calidad cree usted que Audacious Freelance mejora?	72

ÍNDICE DE TABLAS

Tabla 1: Necesidades del Freelance y soluciones desde la perspectiva de procesos software	41
Tabla 2: Análisis de los eventos de Scrum.....	43
Tabla 3: Análisis de los artefactos de Scrum.....	44
Tabla 4: Análisis de las prácticas de XP	45
Tabla 5: Ficha del estudio de caso	64

CAPÍTULO 1. INTRODUCCIÓN

1.1. Contexto

Freelance o *Freelancer*, es un término que se utiliza para designar a un trabajador autónomo que realiza labores propias de su ocupación. Este modelo de trabajo, en el área del desarrollo de software, nació durante los años noventa como producto de las tendencias del *outsourcing*¹ y el *offshoring*² de tecnologías informáticas [1], y el mercado de software ha sido potenciado en los últimos años por el apogeo de la web y los dispositivos móviles, causando así un enorme crecimiento de aplicaciones [2] relativamente poco complejas y de propósito específico que pueden llegar a ser desarrolladas por un solo una sola persona.

Esta oportunidad de trabajo no ha sido inadvertida por los desarrolladores *Freelance* de software de todo el mundo, quienes intentan abrirse campo en este mercado laboral el cual, particularmente, facilita su trabajo, pues son muchas las plataformas que tienen como finalidad acercar al desarrollador de aplicaciones software al usuario final, pasando por su comercialización, como el caso de las plataformas de distribución digital como *Google Play*³, *App Store*⁴, o *Steam*⁵, o los ya populares *Freelance marketplaces* [3] que actúan de intermediario entre los clientes con proyectos de desarrollo de software y los desarrolladores. Esta facilidad de mercado, hace que la competencia sea grande, por tanto, los desarrolladores *Freelance* deberán pensar en mejorar aspectos que los puedan diferenciar y los hagan más competitivos.

Una de estas mejoras podría ser la inclusión de procesos de desarrollo de software que les permita mejorar calidad y desempeño, sin embargo, en el mundo de la ingeniería de software, pocos se han dedicado a estudiar y mejorar el proceso de desarrollo de software por una sola persona [1] y menos aún los procesos concernientes al negocio del software, como la gestión de proyectos, planeación y venta. Incluso las metodologías ágiles presentan problemas al ser adoptadas por una sola persona [4].

1.2. Problema

Si bien, la modalidad de trabajo de desarrollo de software como *Freelance* nació en los años 90, es hasta mediado de la última década que se ha visto el verdadero florecimiento de este modelo de mercado. Debido a esto, son pocas las propuestas formales de un

¹ Es el proceso económico empresarial en el que se delega los recursos orientados a cumplir ciertos servicios a una sociedad externa, dedicada a la prestación de diferentes servicios especializados, por medio de un contrato.

² Es una subcontratación de procesos de negocios de un país a otro, usualmente en busca de costos más bajos o mano de obra. Incluye procesos como producción, manufactura, servicios e incluso innovación o investigación y desarrollo (I+D)

³ Google Inc, "Google Play." <http://play.google.com/>.

⁴ Apple Inc, "App Store."

⁵ VALVE Corp, "Steam." <http://store.steampowered.com/>

proceso que permita estructurar, planificar y controlar el desarrollo de software para una sola persona [1], y éstas son, en su mayoría, adaptaciones de modelos ágiles poco documentadas, las cuales se enfocan sólo en el proceso de desarrollo de software como tal, pero no exploran ni definen adecuadamente el modelo de trabajo *Freelance*, sus dinámicas y el manejo del negocio.

De esta manera, el desarrollo de software como *Freelance* es una buena forma de entrar al mercado laboral [5] y ser exitoso aprovechando estos momentos de auge de las tecnologías informáticas en donde el mercado es exigente, crítico y estricto en términos de calidad y productividad [6]. Para que un desarrollador pueda sobrevivir en este entorno, es necesario que consiga mejorar su desempeño, optimizando el tiempo y el esfuerzo con un proceso que lo forme y lo guíe [7], puesto que los proyectos *Freelance* se exceden en tiempo y recursos, sin lograr la calidad esperada y así la satisfacción del cliente [6] como se muestra en la Figura 1, convirtiéndose en un problema recurrente.

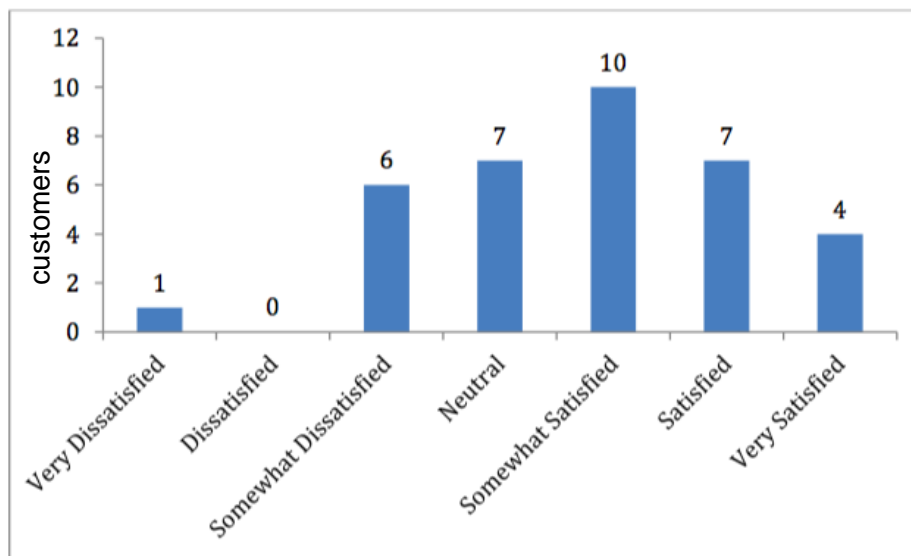


Figura 1: Satisfacción del comprador en proyectos de software *Freelance*

Tomado de Walter

Se hace necesario definir adecuadamente un proceso para el desarrollo de software orientado a una sola persona y que consiga lograr altos mejor niveles de productividad y calidad de servicio en el marco de soluciones dentro del mercado *Freelance*.

El problema en general que aborda este proyecto es el cómo elevar la calidad, productividad y competitividad del desarrollo de software *Freelance* desde un enfoque de gestión de procesos.

1.3. Justificación

Según Fedesoft [8], actualmente la ingeniería de software vive su época dorada en el país [9]; el desarrollo de software en el último año creció más que el resto de la economía

Colombiana [10]. Cifras muestran que el 62% del comercio de software se divide en igual proporción entre desarrollo y *outsourcing* de software y hay un considerable 5.56% de profesionales en software que trabajan independientemente [11]. A nivel mundial, esta tendencia también es evidente, son muchos los casos de éxito de soluciones de software brindadas por desarrolladores independientes [12] que logran hacerse campo laboral y obtener altos ingresos. Este es el caso de las aplicaciones Braid [13], Fez [14], el Minecraft [15], y el reciente Flappy birds [16]. Los estudios realizados por A. Van Den Born y A. Van Witteloostuijn concluyen es que este modelo de trabajo seguirá en auge [5] y se espera que para el 2020, los *Freelance* harán el 50% del trabajo que actualmente hacen los trabajadores de tiempo completo [17]

Estos datos evidencian la necesidad de trabajar en los problemas de la industria *Freelance*, por ello, en este trabajo de grado se busca establecer un proceso ágil de desarrollo de software para una persona, que esté bien definido y que mejore su calidad de su trabajo profesional y la calidad de sus productos.

1.4. Objetivos

Objetivo General

Proponer un proceso de desarrollo de software, denominado *Audacious Freelance*, para ser ejecutado por una sola persona y que responda a las necesidades de un desarrollador de software *Freelance*.

Objetivos Específicos

- Caracterizar el desarrollo de software por una sola persona como *Freelance*.
- Proponer un modelo de proceso de desarrollo de software basado en la caracterización realizada y que se rija por los principios del manifiesto ágil.
- Evaluar y contrastar el desempeño de *Audacious Freelance* mediante la ejecución de dos proyectos piloto de desarrollo de software, uno de control con desarrollo *ad-hoc* y el otro ejecutando *Audacious Freelance*.

1.5. Metodología y descripción del trabajo

Para el desarrollo de este trabajo de grado se estudiaron las propuestas de desarrollo de software para una sola persona que fueron presentadas en otros trabajos, se analizaron y se encontraron oportunidades de mejora. Posteriormente, se definió el modelo de procesos siguiendo el enfoque metodológico dado por el meta-proceso definido por Ruiz [18], para ello, mediante un estudio exploratorio se caracterizó al desarrollador *Freelance* estableciendo sus dinámicas, para con esto, proponer un proceso software que sea acorde con las necesidades encontradas del *Freelance*. Éste proceso se denominó *Audacious Freelance*. Finalmente esta nueva propuesta se evaluó en un caso de estudio embebido, de acuerdo a los lineamientos establecidos por Runeson [19], que permitió conocer el aporte de *Audacious Freelance* en la ingeniería de software.

Primero se presenta un marco conceptual del proyecto en el capítulo 2; definiendo los conceptos concernientes en este estudio y mostrando el estado del arte con los estudios relacionados con procesos de desarrollo de software por una sola persona.

En el capítulo 3 de éste documento se define y caracteriza al *Freelance* mediante el uso de definiciones formales y por medio de un estudio exploratorio en el que se usó la encuesta como herramienta de colección de información la cual fue difundida por medio de la web, principalmente por grupos de redes sociales y foros especializados en programación.

En el capítulo 4, y de acuerdo a la información previamente recolectada sobre las dinámicas, condiciones y contexto de un *Freelance*, se determinó sus falencias permitiendo proponer un proceso de desarrollo de software que se ajuste a sus necesidades junto con herramientas para lograr la implementación del proceso por parte de los *Freelance*.

En el capítulo 5, se evalúa *Audacious Freelance* en un caso de estudio el cual consiste en la realización de dos proyectos de desarrollo de similares características, uno de control y otro ejecutando *Audacious Freelance* con el fin contrastar los resultados y así poder determinar el aporte de este proyecto.

Finalmente, en el capítulo 6, se presentan las limitaciones del desarrollo de este proyecto, así mismo, las conclusiones obtenidas y los trabajos futuros.

CAPÍTULO 2. MARCO CONCEPTUAL Y ESTADO DEL ARTE

2.1. Marco conceptual

2.1.1. Freelance

El trabajador *Freelance* también llamado *Freelancer*, es “una persona que actúa con independencia y sin estar afiliado con o autorizado por una organización.”⁶, un “trabajador por cuenta propia que es contratado para diferentes compañías en tareas particulares”⁷, este modelo de trabajo *Freelance* en el desarrollo de software a nivel formal ha sido poco abordado [5].

El *Freelance* puede ser considerado como un híbrido entre empleado y empresario [5]. Empleado porque generalmente son contratados por empresas para trabajos particulares en contratos cortos para únicos proyectos, en donde básicamente venden su conocimiento profesional, algo intangible, que los separa de cualquier otra relación vendedor-empresa en la cual se intercambie mercancía tangible. Por otro lado, también es considerado empresario, entendiéndolo como una empresa unipersonal [1], porque los *Freelance* trabajan bajo su propio riesgo y sin el apoyo de una organización que los respalde. Ellos escogen su manera de trabajar, sin cumplir horario ni exigencias aparte de qué se va a entregar. Son autónomos y pueden tomar decisiones técnicas deliberadamente; no tienen jefes, tienen clientes.

En Estados Unidos, la IRS⁸ habla de tres criterios para determinar si una persona es un *Freelance* o un empleado, esta son: conductual (*behavioral*), financiero (*finacial*) y tipo de relación (*type of relationship*).

- **Conductual:** Este criterio evalúa el control que tiene el empleador sobre el trabajador, es decir, qué tanto puede el empleador inmiscuirse en la forma como se desempeña el trabajador, indicando cuando, como y donde trabajar. Entre más control tenga el empleador sobre el trabajador, es más considerado un empleado que un *Freelance*. Los factores que se involucran en este criterio son: tipo de instrucciones dadas, grado de la instrucción, sistema de evaluación y entrenamiento.
- **Financiero:** Este criterio evalúa la forma en como el empleado es retribuido por su trabajo. Los factores que determinan este criterio son: inversión al importante al trabajador, reembolso de gastos (viáticos), oportunidad de ganancia o pérdida

⁶ Definición del diccionario Marriam-Webster

⁷ Definición del diccionario Oxford

⁸ Internal Revenue Service

(comisiones), y formas de pago. En trabajador estará más cercano a un empleado, entre más relaciones económicas tenga con el empleador.

- **Tipo de relación:** Este criterio identifica el tipo de relación que tiene el empleador con el trabajador. Los factores para este criterio son: tipo de contrato, beneficios para trabajadores, la permanencia de la relación de trabajo, y los servicios prestados como actividades clave del negocio.

En Colombia, un trabajador *Freelance* se puede asociar al término de *trabajador independiente* o *contratista*, siendo así las personas que no están vinculadas a una empresa mediante un contrato de trabajo, sino mediante un contrato de servicios y son remunerados bajo la figura de honorarios. Este tipo de contrato tiene las siguientes características:

- Prestación de servicios específicos.
- Autonomía e independencia del trabajador desde el punto de vista técnico y científico.
- No subordinación continúa.
- No cumplimiento de horarios.
- La vigencia del contrato es temporal y, por lo tanto, su duración debe ser por tiempo limitado.
- No se genera ninguna relación laboral y por ende no hay lugar al pago de prestaciones sociales.
- La afiliación al sistema integral de seguridad social se debe realizar como trabajador independiente, esto es, asume la totalidad de las cotizaciones.

Para el propósito de este trabajo, se está interesado en los trabajadores *Freelance* en el sector del desarrollo de software.

2.1.2. Proceso de desarrollo de software

Desde un punto de vista general, un proceso consiste en un conjunto de actividades relacionadas que transforman una o más entradas en salidas mientras consumen recursos para completar dicha transformación [20].

En el contexto de la ingeniería de software existen varias definiciones sobre proceso de desarrollo de software, que muchas veces incluyen además del desarrollo, el mantenimiento de software. Algunas de éstas son:

- “Conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener software y los productos de trabajo asociados (planes de proyecto, diseño de documentos, código, pruebas y manuales de usuario)”⁹

⁹ *Software Engineering Institute (SEI), 1995*

- “Proceso o conjunto de procesos usados por una organización o proyecto, para planificar, gestionar, ejecutar, monitorizar, controlar y mejorar sus actividades software relacionadas”¹⁰
- “El proceso software define cómo se organiza, gestiona, mide, soporta y mejora el desarrollo, independientemente de las técnicas y métodos usados”¹¹
- “Un proceso es una entidad implícitamente o explícitamente definida para representar, adaptar, instanciar y ejecutar un conjunto de tareas parcialmente ordenadas incluyendo sus participantes, entradas, salidas y conocimiento asociados”¹²

De acuerdo al SWEBOK¹³, Un proceso de software como mínimo debe describir las entradas, las actividades de transformación y las salidas generadas. Las actividades también pueden estar divididas en tareas, que se ejecutan como un procedimiento, para describir la transformación de manera más detallada (Figura 2). Un proceso además también puede incluir subprocesos, comunicados entre sí, cumpliendo las condiciones de entrada y salidas de los procesos asociados, para dar forma a un proceso más grande. Un proceso de software también puede incluir roles y competencias, técnicas, prácticas y herramientas, así como también enfoques y criterios de mediciones que se utilizan para determinar la eficiencia y la eficacia de la realización del proceso.

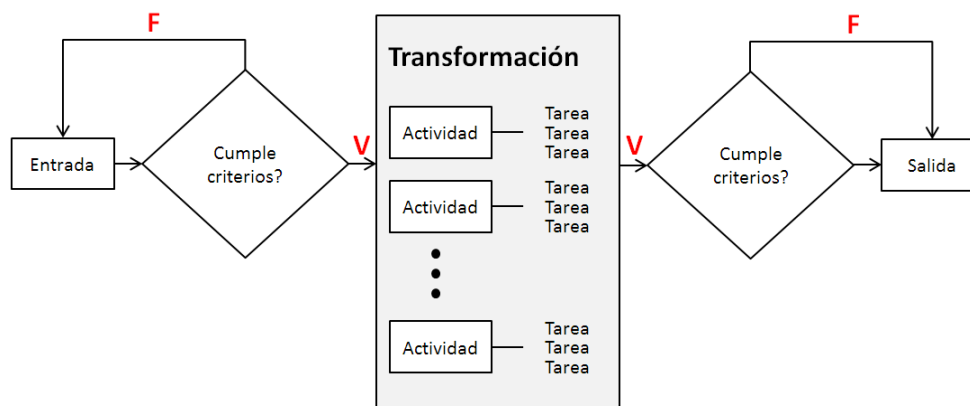


Figura 2: Elementos de un proceso de software

Los procesos de software se especifican por las siguientes razones: para facilitar la comprensión humana, la comunicación y la coordinación; para ayudar a la gestión de

¹⁰ International Organization for Standardization (ISO), 1998

¹¹ J. C. Derniame et al., 1999

¹² J. A. Hurtado et al., 2011

¹³ Software Engineering Body of Knowledge

proyectos de software; para medir y mejorar la calidad de los productos de software de una manera eficiente; para apoyar la mejora de procesos; y para proporcionar una base para el apoyo automático de ejecución del proceso [20].

Algo que se debe tener en cuenta, es que un proceso de software no puede ser general y no hay un proceso que sea mejor que los demás. Los procesos de software deben ser seleccionados y adaptados (*tailoring*) de acuerdo a los proyectos y contextos en donde se quiere aplicar dicho proceso, como por ejemplo: tecnologías a usar, tamaño del proyecto, riesgo del proyecto, naturaleza de los requerimientos, prácticas de la industria, etc. Se requiere esfuerzo cuando se decide introducir y mejorar un proceso de software, sin embargo, la experiencia ha mostrado que el esfuerzo gastado es bien recompensado debido a que resulta una mejor en la eficiencia, optimizando el trabajo antes realizado [20].

El *ciclo de vida del desarrollo de software* (SDLC por sus siglas en inglés), incluye los procesos de planeación y gestión para transformar requisitos de software en un producto software para entregar. El *ciclo de vida de un producto software* (SPLC por sus siglas en inglés) además de contar con los procesos de SDLC, incluye procesos asociados a despliegue, mantenimiento, soporte, evolución, retiro, gestión de la configuración, aseguramiento de calidad, entre otros.

2.1.3. Métodos ágiles de desarrollo de software

El desarrollo ágil de software es un cambio de paradigma en cuanto a la forma como se estaban desarrollando los proyectos software. El término “ágil” en el contexto de métodos de desarrollo de software fue acuñado en 2001, en Utha - EEUU, en donde un grupo de expertos de la industria propusieron una serie de valores y principios, consignados en el *manifiesto ágil*¹⁴, con el fin de disminuir la burocracia de los métodos existentes hasta el momento y priorizar la comunicación interna entre el grupo desarrolladores y con él cliente, para con esto, poder desarrollar software de manera más rápida y que a la vez, los proyectos puedan responder a los cambios en el transcurso de su ejecución. De esta manera, se le llama “ágil” a los métodos, metodologías, procesos y frameworks de desarrollo de software que aplican lo expuesto en el *manifiesto ágil*. Algunos de los métodos ágiles son: *eXtreme Programming*, *Scrum*, *Adaptative Software Development*, *Crystal Clear*, *Feature Driven Development*.

2.2. Estado del arte

Para realizar la revisión de propuestas en la literatura científica, se siguió la plantilla del protocolo presentado por Biolchini [21]. Este protocolo consta de cinco partes generales, sin embargo, para propósito de este estudio y de alcance, se desarrollaron las primeras cuatro que a continuación se explican brevemente.

¹⁴ <http://www.agilemanifesto.org/iso/es/>

- 1. Formulación de la pregunta:** Para esta actividad, la pregunta de investigación fue ¿Qué metodologías, procesos o frameworks de desarrollo de software para un *Freelance* existen? La lista de términos que se usaron fue: *Software, Freelance, solo, single-person, one-person, personal, methodology, process* y *framework*. Se esperaba como resultado un listado de alternativas metodológicas para desarrollo de software diseñados para ser ejecutados por una sola persona y cuáles eran las motivaciones para dichas alternativas.
- 2. Selección de fuentes:** De acuerdo a la lista de términos anterior, se usó la siguiente cadena con conectores lógicos “AND” y “OR para realizar las búsquedas: (*Freelance* OR solo OR single-person OR one-person OR personal) AND software AND (methodology OR process OR framework). La lista de fuentes seleccionada para realizar la revisión fue: *ProQuest, ScienceDirect, IEEE Xplore, Wiley, ACM, Sage journals, Google Scholar* (como motor de búsqueda de bibliografías).
- 3. Selección de los estudios:** Para la selección de los estudios, se usó el procedimiento descrito por Pino en [22].
- 4. Extracción de la información:** Posteriormente, a los estudios que se encontraron relevantes, se les extrajo la información sobre el proceso (metodología o framework) que proponen, aspectos que cubre el proceso, la influencia (si la tuvieron) de otros procesos de desarrollo software, la metodología usada para la adaptación, herramientas de gestión del proceso y la selección de criterios para el diseño del proceso. Posteriormente, fueron agregados a la herramienta software de gestión bibliográfica *Zotero*¹⁵ la cual, además de gestionar las referencias y su visualización en diferentes estilos, también permite realizar anotaciones a las referencias sobre la información importante que fue extraída.

Después de todo, son pocas las propuestas que se encuentran de procesos software para una sola persona [1] y la mayoría de ellas son adaptaciones de metodologías ágiles poco especificadas. Las propuestas encontradas se dividieron en dos, propuestas originales y las adaptadas. Las propuestas originales son aquellas que fueron concebidas desde cero para ser ejecutadas por una sola persona. Las propuestas adaptadas, a diferencia de las anteriores, son adaptaciones de métodos ágiles, diseñados para grupos de trabajo, y que han sido cambiadas de tal manera que puedan ser ejecutadas por un solo desarrollador.

2.2.1. Propuestas de procesos de software para un solo desarrollador

Propuestas originales

a. PSP

Personal Software Process [23], es el proceso de desarrollo de software para una sola persona mejor definido hasta el momento. Se trata de un conjunto de prácticas, procesos y guías que tienen como propósito mejorar, gestionar y controlar el trabajo de desarrollo de una persona. Cubre áreas de administración

¹⁵ <https://www.zotero.org/>

de proyectos, ingeniería de software y administración de calidad. PSP básicamente se trata de un proceso que ayuda a mejorar a un ingeniero de software con la premisa de que, si es bueno trabajando individualmente, será bueno trabajando en equipo.

Se encuentran varios problemas en PSP, entre estos: es un proceso muy estricto, consume mucho esfuerzo en toma de información, en el manejo de tablas para la toma de decisiones y para encontrar oportunidades de mejora, además, se requiere tener un buen conocimiento de la especificación del proceso y usualmente requiere de la instrucción de un experto [1], [24].

b. SIP

Solo Iterative Process [25], fue propuesto por Václav Rajlic. Este proceso de desarrollo de software para una sola persona ayuda al desarrollador a realizar proyectos de mejor calidad y, a diferencia de los procesos anteriormente descritos, involucra áreas de procesos como gestión de tiempo, gestión de *stakeholders*, requerimientos y aspectos relacionados con el negocio del software. Una de sus principales prácticas es hacer seguimiento constante del tiempo y esfuerzo usado en los proyectos, con el fin de lograr estimaciones futuras más certeras.

SIP es, si bien una adaptación de un proceso llamado SC [26], al tratarse del mismo autor, quien hace la adaptación, se ha catalogado como original. SC Consta de 8 actividades que tienen que ver con el desarrollo de software como tal y que son realizadas iterativamente en SIP. SC cuenta con una herramienta desarrollada como un *plug-in* de Eclipse que sirve como un asistente para el desarrollador.

Propuestas adaptadas de métodos ágiles

c. PCSE

Practitioner Centered Software Engineering [27], es un proceso personal de auto superación para ingenieros de software, que tiene como objetivo controlar, gestionar y mejorar el desempeño de un profesional del software, con el fin de planear certeramente, incrementar la disciplina, y fortalecer el *QPPC*¹⁶. Para lograr esto, PCSE hace uso de prácticas y elementos adoptados de varios procesos de software como *PSP*¹⁷, *TSP*¹⁸, *XP*¹⁹, *FDD*²⁰, *Scrum*²¹ y *RUP*²². Estos elementos son

¹⁶ *Quality, Predictability, Productivity and Customer satisfaction*

¹⁷ *Personal Software Process*

¹⁸ *Team Software Process*

¹⁹ *eXtrememe Programming*

²⁰ *Feature-Driven Development*

²¹ *Framework de desarrollo de software*

²² *Rational Unified Process*

agrupados en actividades típicas como: análisis, diseño de arquitectura, planeación del proyecto, iteración, construcción, revisión, refactorización, integración y despliegue. Las prácticas son acompañadas por algunos artefactos que las soportan desde calendarios hasta matrices para calcular la dimensión del proyecto. Estos artefactos son algo pesados e involucran bastantes factores que podrían hacer de PCSE una metodología poco liviana y que al igual que PSP, requiere de un aprendizaje previo.

El mayor inconveniente, es que no está bien documentado, solo se conoce un estudio ejecutando PCSE [7], en donde hacen un caso de estudio implementando PCSE para un desarrollo web y dado la naturaleza del proyecto, los desarrolladores deben hacer ajustes no menores en las prácticas y en los artefactos.

d. PXP (Dzhurov)

Personal Extreme Programming [1], es una propuesta de proceso inspirado en PSP pero más fácil de seguir, minimiza la documentación, conserva las prácticas básicas de PSP y usa algunas prácticas de XP para reforzar el control de calidad y la planeación de proyectos. El objetivo de PXP es mejorar el desempeño y la calidad del trabajo de los desarrolladores de software independientes por medio de constantes retrospecciones, las cuales buscan oportunidades de mejora. Gran parte de las actividades del proceso, son soportadas por una herramienta creada para este propósito. El proyecto fue confrontado con desarrollos *ad-hoc*²³, mostrando buenos resultados en la identificación temprana de errores, sin embargo, el tiempo de planeación consumido fue significativamente mayor que en los *ad-hoc*. PXP es aún muy teórico y poco especificado.

e. PXP (Agarwal)

Extreme Programming for a Single Person Team [28], aunque tiene el mismo nombre que el proceso anterior, éste es una adaptación únicamente de XP. En este caso, la estrategia para adaptar XP a un sólo desarrollador, consiste en tomar cada una de las doce prácticas de XP y hacer una versión alternativa para que puedan ser hechas por un sólo desarrollador. Eso implica una serie de cambios en el proceso original, por ejemplo en el manejo de las *historias de usuario* que ahora son más “ligeras” y la práctica de *programación en parejas* que ha sido cambiada, entre otras. Algo interesante de este proceso, a diferencia de los demás, es que muestra el *script*²⁴ del proceso de PXP, permitiendo establecer un orden coherente para la implementación del proceso.

²³ Proyectos planeados en el momento de desarrollo y sólo para el proyecto en particular, no están formalizados ni estandarizados.

²⁴ Una lista de pasos secuenciales para la ejecución de un proceso

f. eXtreme Solo

Es un caso de estudio [29] de la aplicación de XP para un sólo desarrollador. Gareth Cronin, adapta XP, dejando a un lado las prácticas concernientes a equipos de trabajo como la programación en parejas, para hacer el desarrollo de un software. El proceso XP, no es un proceso adecuado para un desarrollador individual, pero su propio creador, Kent Beck, sugiere que, para un desarrollador individual, es la mejor alternativa entre los demás procesos. Sus conclusiones son muy reveladoras: el uso de XP adaptado comparado al desarrollo *ad-hoc*, define un marco de trabajo que mejora muchos aspectos. Entre estos, los más destacados son las pruebas unitarias y las iteraciones cortas y planeadas. Sin embargo, al quitar prácticas como programación en parejas que son insignias de XP, hizo que tampoco se pudiera realizar adecuadamente otras prácticas como refactorización, conclusión a la que también llegó Edward Akpata [30]. Este estudio evidencia ciertas falencias en la adaptación de procesos originalmente definidos para pequeños o grandes equipos de trabajo al desarrollo hecho por una sola persona.

g. AGILE SOLO

Es otra adaptación de procesos ágiles, creada por Anna Nystrom [2], con el fin de conseguir un proceso de desarrollo para una sola persona. En este caso se trata de la colección de prácticas que deben ser cumplidas obligatoriamente en cada fase. Para evaluar el proceso, se hizo un proyecto piloto que trataba del desarrollo de una aplicación para dispositivos móviles, al culminar, y a modo de conclusión, Nystrom se da cuenta que su proceso no se ajusta a las particularidades de su proyecto y aconseja que lo que se debe hacer en todo caso es un *tailoring*²⁵ de AGILE SOLO para poder tener mejores resultados de acuerdo a las características del proyecto a desarrollar.

h. Otros aportes

Por otro lado, y de manera menos formal, Rody Middelkoop [4], desglosa *Scrum*, *XP* y *UP*²⁶ en sus actividades más relevantes y da unas pautas, desde su punto de vista como consultor, de las actividades que se deben o no hacer si se está haciendo un desarrollo individual. Edward Akpata [30] también analiza XP concluyendo que algunas prácticas pueden ser apropiadas por un solo desarrollador.

Es de gran valor la manera en la que logra dar alternativas para el trabajo individual, siendo un buen referente para tener en cuenta a la hora de realizar la incorporación prácticas a un nuevo proceso orientado a una sola persona.

²⁵ Adaptación de un proceso a un contexto

²⁶ Unified Process

2.2.2. Análisis

Luego de realizar un análisis de los estudios mencionados, son varias las falencias que se encuentran y es por eso que se ha encontrado la posibilidad de aportar al campo de la ingeniería de software con este proyecto.

El principal inconveniente con estas propuestas es que, generalmente son adaptaciones poco formales de modelos ágiles, diseñados originalmente para grupos de trabajo y adaptados para ser usados por una sola persona, muchas veces eliminando características esenciales de los procesos software. Además, estos estudios se enfocan en el desarrollo de software como tal, y no tienen en cuenta el contexto ni las dinámicas de un *Freelance*. Estas dinámicas van más allá del desarrollo de un producto/servicio software, pues un *Freelance* requiere procesos transversales a los de una empresa, como procesos de gestión y producción. Estos deben ser ligeros, y correctamente especificados para el que *Freelance* pueda aprenderlos por su cuenta y deben contar con herramientas software que apoyen en la gestión de sus proyectos, que como se dijo anteriormente, complementan la especificación de un proceso software.

CAPÍTULO 3. ANÁLISIS Y CARACTERIZACIÓN DEL FREELANCE

3.1. Diseño del estudio

Para lograr el primer objetivo específico, se realizó un estudio exploratorio con el propósito de determinar las características de trabajo y el comportamiento de un *Freelance* del software, considerando aspectos como su formación académica, entorno de trabajo, experiencia laboral, área de desempeño, situaciones cotidianas y conocimientos técnicos sobre ingeniería de software con el fin de determinar qué factores disminuyen la probabilidad de lograr proyectos exitosos en el desarrollo de software como *Freelance*.

Esta actividad hizo uso de la encuesta descriptiva [31], usando el cuestionario como instrumento cualitativo de recolección de información y la técnica usada fue CAWI²⁷. Esta técnica consiste en realizar la encuesta en una herramienta web, y difundirla a través del internet, siguiendo los lineamientos propuestos por Lumsden [32], compilados en el siguiente proceso:

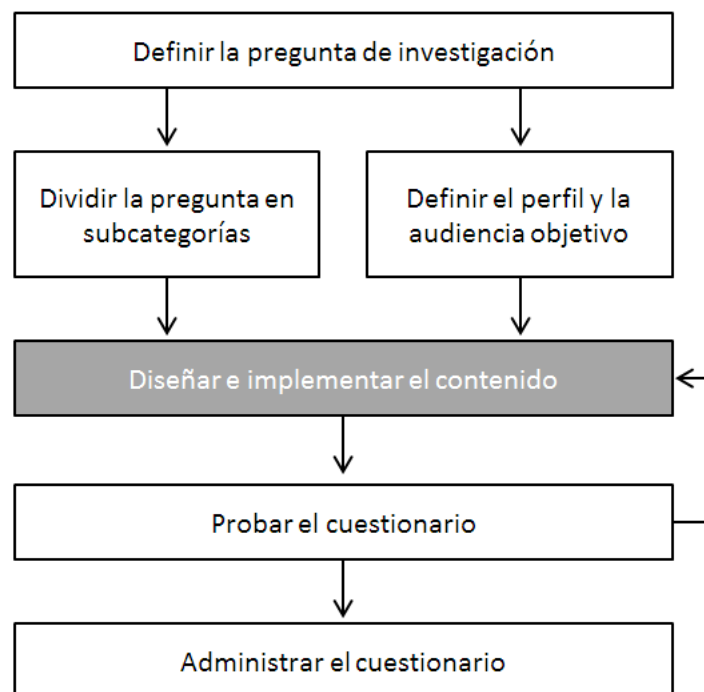


Figura 3: Proceso para diseño de cuestionarios online por Lumsden

²⁷ Computer Assisted Web Interview

A continuación se explica brevemente los pasos para la ejecución del proceso:

- 1. Definir la pregunta de investigación:** El propósito del cuestionario era conocer y establecer, qué perfil tienen los *Freelance* en el campo del desarrollo de software.
- 2. Dividir la pregunta en subcategorías:** En esta actividad, se deben dividir la pregunta de investigación en subcategorías con el fin de ir organizando el contenido y las preguntas de la encuesta para esto las subcategorías fueron:
 - ¿Cuáles son aspectos personales de un *Freelance*?
 - ¿Cómo ha sido su experiencia como *Freelance*?
 - ¿Cuáles son las situaciones frecuentes que tienen que vivir los *Freelance*?
 - ¿Cuáles son sus ideales?
 - ¿Qué conocimiento tienen sobre proceso software?
- 3. Definir el perfil y la audiencia objetivo:** Los cuestionarios son diseñados para recolectar información sobre un grupo particular de individuos, de esta manera, y teniendo en cuenta la internet como herramienta principal para ellos, la audiencia objetivo fueron *Freelance* involucrados en el desarrollo de software alrededor del mundo, particularmente quienes participan de grupos en redes sociales y foros especializados en el tema.
- 4. Diseñar e implementar el contenido:** De acuerdo a lo anterior, se dividió el contenido en secciones que permitieran dar respuesta a cada pregunta que se había formulado. Para crear el cuestionario, se usó el servicio gratuito de *Google Forms*²⁸.
- 5. Probar el cuestionario:** El cuestionario fue probado por algunas *Freelance* con el fin de determinar si existía algún mal entendido en las preguntas, ambigüedades, o si hacía a falta secciones, entre otras.
- 6. Administrar el cuestionario:** Una vez el cuestionario ha sido diseñado y corregido, es momento de ser administrado. El cuestionario fue difundido por medio de grupos de redes sociales como: *Desarrolladores PHP Freelance*, *Desarrolladores Java – Colombia*, *Comunidad de desarrolladores de videojuegos*, entre otros. Y en algunos foros especializados como: *programmers.stackexchange.com*, *codecall.net* y *codeproject.com*

El cuestionario tuvo 54 preguntas que se redactaron en inglés y fueron agrupadas según el tipo de información que se quería determinar, de esta manera: 5 preguntas fueron sobre aspectos personales, 9 sobre su experiencia como *Freelance*, 16 preguntas sobre situaciones comunes que se presentan en su trabajo, 11 sobre sus ideales como desarrollador y 13 preguntas sobre conocimientos de procesos software. La encuesta estuvo publicada durante una semana en Mayo de 2014 y fue respondida por 66 *Freelance* del área del desarrollo de software de distintos lugares del mundo.

²⁸ Google Inc, "Google Forms." <http://drive.google.com/>.

3.1.1. Estructura del cuestionario

El cuestionario se dividió en cinco secciones así:

Aspectos personales

Al ser una encuesta difundida libremente por internet, es decir, no se sabía exactamente a qué muestra estaba dirigida, su intención era poder determinar la población. Algunas preguntas fueron:

- ¿Qué edad tiene?
- ¿Cuál es el nivel más alto de educación que ha completado?

Experiencia del Freelance

Su intención fue conocer algunos antecedentes de su trabajo, principalmente orientado a los proyectos que desarrollan. Algunas preguntas fueron:

- ¿Cuántos años de experiencia tiene usted con los proyectos de TI?
- ¿Cuál es el tiempo promedio del desarrollo de un solo proyecto?
- ¿Cuántas horas trabaja a la semana?
- ¿Alguna vez ha subcontratado?
- ¿En qué sector de industria trabaja?

Dinámicas Frecuentes

Esta sección busca determinar las dinámicas comunes en los proyectos TI evaluadas mediante una escala nominal de acuerdo a su recurrencia (nunca, raramente, usualmente, frecuentemente, siempre). Algunas de las situaciones fueron:

- Usted tiene proyectos simultáneos.
- Usted estima certeramente el tiempo del proyecto.
- Usted sabe exactamente qué hacer en el proyecto.
- El cliente es técnicamente competente.
- La comunicación con el cliente es adecuada.
- Usted tiene problemas con los requerimientos.

Ideales del Freelance

Esta sección fue diseñada para hacer una confrontación con respecto a otras preguntas. Su intención fue determinar qué ideales tienen los *Freelance* al hacer su trabajo. Fue diseñada como una matriz en donde las filas representan aspectos de un proyecto que les gustaría priorizar y las columnas en rangos de 0 a 5, donde 0 significa una prioridad nula y 5 que son de más alta prioridad. Se diseñó para que los encuestados pudieran evaluar con máxima prioridad todos los aspectos. Algunas de los aspectos a priorizar fueron:

- Calidad en uso.
- Calidad externa.
- Satisfacción del cliente.
- Presupuesto del proyecto.
- Tiempos de entrega.

Conocimiento sobre procesos software

Las preguntas de esta sección fueron sobre procesos, metodologías y frameworks, enfocándose en los que se consideran ágiles y cómo los encuestados estaban familiarizados con éstos. Esta sección fue diseñada a manera de matriz en donde las filas son procesos, metodologías y frameworks y las columnas se refieren a un grado de familiaridad (experto, lo conoce bien, conoce algo, ha escuchado algo, no tiene idea). Algunos de los procesos, metodologías y frameworks fueron:

- Extreme Programming
- Scrum
- Personal Software Process
- Test-Driven Development

El cuestionario completo se encuentra en el *Anexo A* de este estudio.

3.2. Resultados y análisis

La encuesta fue respondida por 66 personas dejando una colección de buenos datos. A continuación se muestran los más relevantes junto con un breve análisis.

3.2.1. Aspectos personales

La encuesta fue respondida en un 88% por personas entre las edades de 18 y 34 años. El 14% está finalizando sus estudios profesionales, 49% es profesional y el 16% tiene título de maestría. Esto nos dice que la mayoría de los encuestados tienen conocimientos formales de desarrollo de software.

3.2.2. Experiencia como freelance

El 78% de los proyectos *Freelance* tiene una duración menor a los 5 meses, indicando que se trata de trabajos relativamente pequeños pero muy ágiles y concretos, siendo la Web como el principal sector de la industria seguida por los móviles y aplicaciones de escritorio en último lugar. El 54% realizan proyectos propios, lo cual refuerza la idea de que el *Freelance* es un legítimo proactivo y emprendedor. Un 47% subcontrata confirmando su espíritu como empresario y entusiasta de su trabajo pues el 88% manifiesta ser feliz y desea continuar con su trabajo como *Freelance*. Sus entornos de trabajo son variados, el 88% dice trabajar parte del tiempo desde la casa, sin ninguna supervisión, el 33% trabaja en oficinas al menos una parte del tiempo y el 40% lo hace junto compañeros de trabajo. El 29% dedica entre 30 y 40 horas semanales a su trabajo

como *Freelance* y el 32% dedica más de 40 horas. En la Figura 4 se muestra la experiencia que los encuestados tienen en proyectos de software *Freelance*.

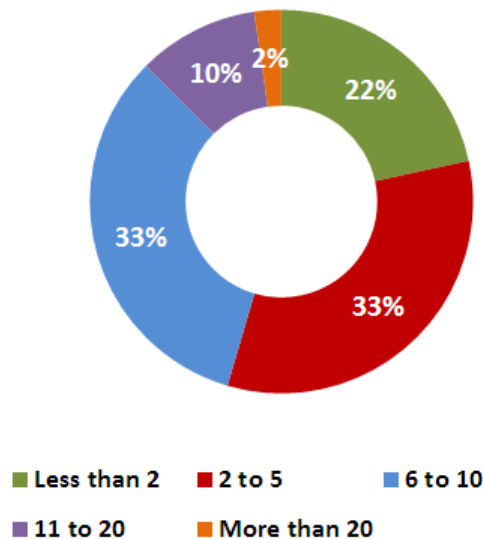


Figura 4: How many years of experience do you have with IT projects?

3.2.3. Situaciones comunes

Como se ve en la Figura 5, en donde la condición ideal es cuando las crestas de las líneas están hacia la parte superior izquierda, el porcentaje de éxito de los proyectos TI *Freelance* es considerable, sin embargo, esto es desde las perspectivas del desarrollador; desde la perspectiva del cliente, el estudio hecho por Walter [6] es un poco más pesimista, ya que la mayoría de proyectos resulta medianamente satisfactorio. Esto sugiere que el concepto de “proyecto exitoso” no esté claro para los *Freelance*. Con respecto a la relación entre la estimación del costo y del tiempo de un proyecto junto con el seguimiento de horas gastado y el costo invertido (reuniones con el cliente, transporte, herramientas de trabajo, etc.), el porcentaje de personas que lo hacen usualmente o rara vez es alto, lo que sugiere que el concepto de proyecto exitoso no considera el desfase que éste pueda tener en tiempo, esfuerzo y costo.

Un resultado valioso es la relación entre su percepción de proyectos exitosos y el conocer aquello que se debe hacer, es decir, contar con requerimientos de calidad (concretos, no ambiguos, verificables, etc.) y tener el conocimiento para hacer el proyecto, es decir, contar con el conocimiento técnico para desarrollarlo. Existe una importante relación en donde, tener buen conocimiento de los requerimientos y tener los conocimientos técnicos para el proyecto es indicio de un proyecto exitoso.

Otro resultado importante en esta sección es que alrededor del 43% de los *Freelance* tienen frecuentemente proyectos simultáneos. Esto indica que son personas dinámicas y flexibles. Esto implica que requieren hacer seguimiento de sus proyectos con el fin hacer el cambio de contexto adecuadamente al momento de alternar entre dos proyectos.

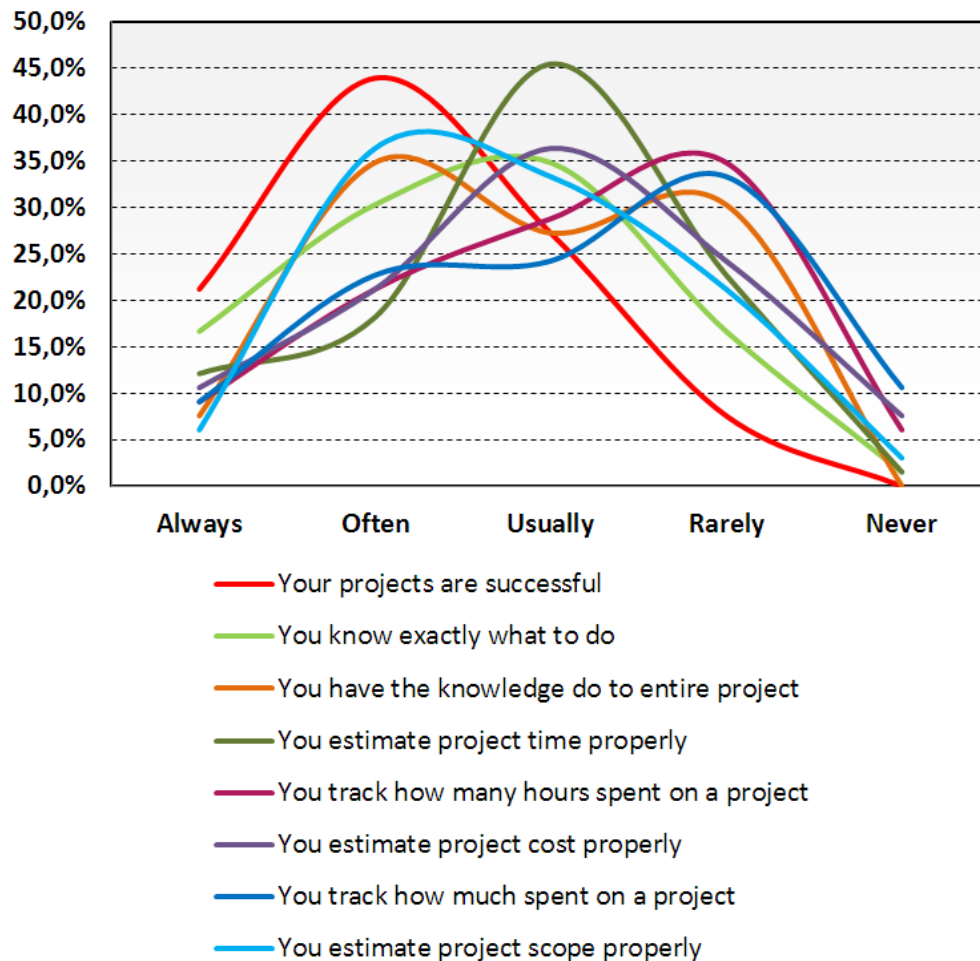


Figura 5: How common are these situations in IT projects?

3.2.4. Ideales como desarrollador

Esta sección busca conocer qué elementos prioriza un *Freelance* al momento de hacer un proyecto, es decir, qué elementos considera de más importancia. En esta sección el encuestado podía marcar cada aspecto de 0 a 5, siendo 5 la opción que indica que le importa sustancialmente y 0 todo lo contrario. Los encuestados pudieron marcar la opción 5 en todo, sin embargo no todos lo hicieron.

En la Figura 6, se muestra poca importancia a tres aspectos considerados como claves a la hora de considerar un proyecto exitoso, el presupuesto, las fechas de entrega y el alcance. Estos datos corroboran la información de la sección anterior.

Definitivamente en lo que más se enfocan los *Freelance* es en la satisfacción del cliente, algo que es coherente entre sí. De nuevo hay que insistir que esto es desde la perspectiva del desarrollador *Freelance*. Como se dijo anteriormente, los datos mostrados por Walter, que son desde la perspectiva del cliente, son algo contradictorios con la percepción del cliente. Esto podría indicar que el desarrollador sólo se está enfocando en su producto o servicio que está desarrollando pero deja a un lado muchos factores importantes respecto a la gestión de sus proyectos.

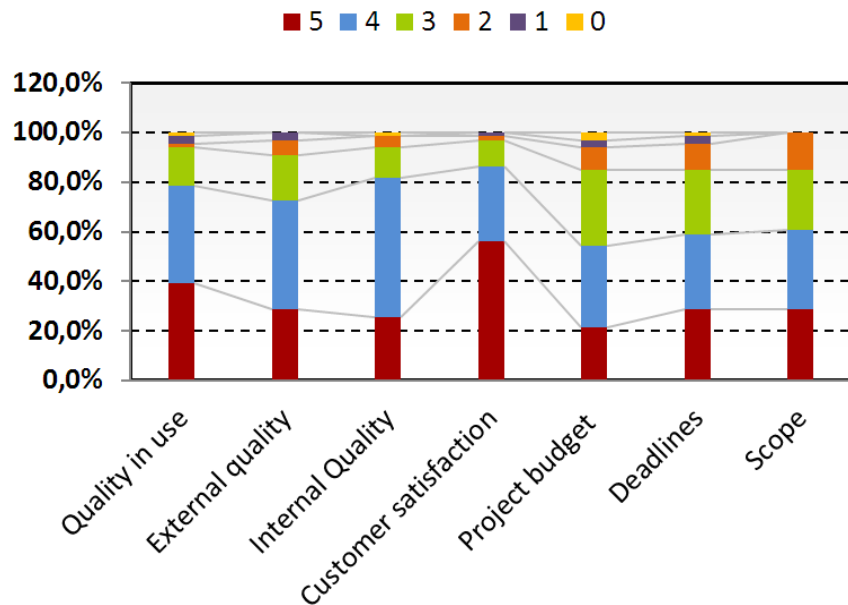


Figura 6: What aspects do you prioritize when you have a project?

3.2.5. Conocimientos sobre procesos software

Esta sección busca conocer principalmente la disposición a utilizar proceso para el mejoramiento de la labor por parte de los *Freelance*. Los resultados de la Figura 7 indican que hay una buena disposición por parte del *Freelance* para usar procesos que permitan mejorar su trabajo. También es interesante ver los problemas por los cuales no los usan.

En la Figura 8 se muestra que tanto saben sobre algunos procesos, metodologías y frameworks. La mayoría de los procesos corresponden a procesos catalogados como ágiles. Los 6 primeros son orientados a grupos pequeños y los 6 últimos son diseñados para ser ejecutados por una sola persona. *XP* y *Scrum* son los procesos que más

conocen. Es importante recalcar que la mayoría de los encuestados presentan un conocimiento escaso acerca de estos procesos.

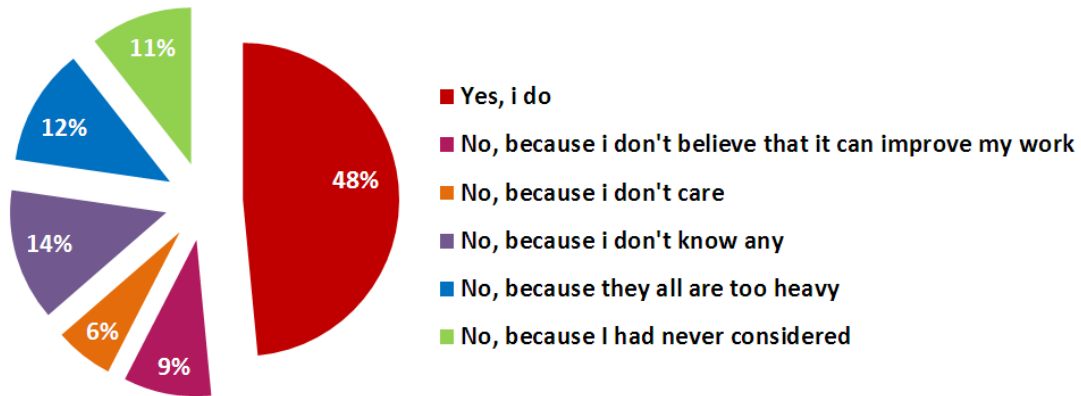


Figura 7: Do you use a formal methodologies/processes for your work?

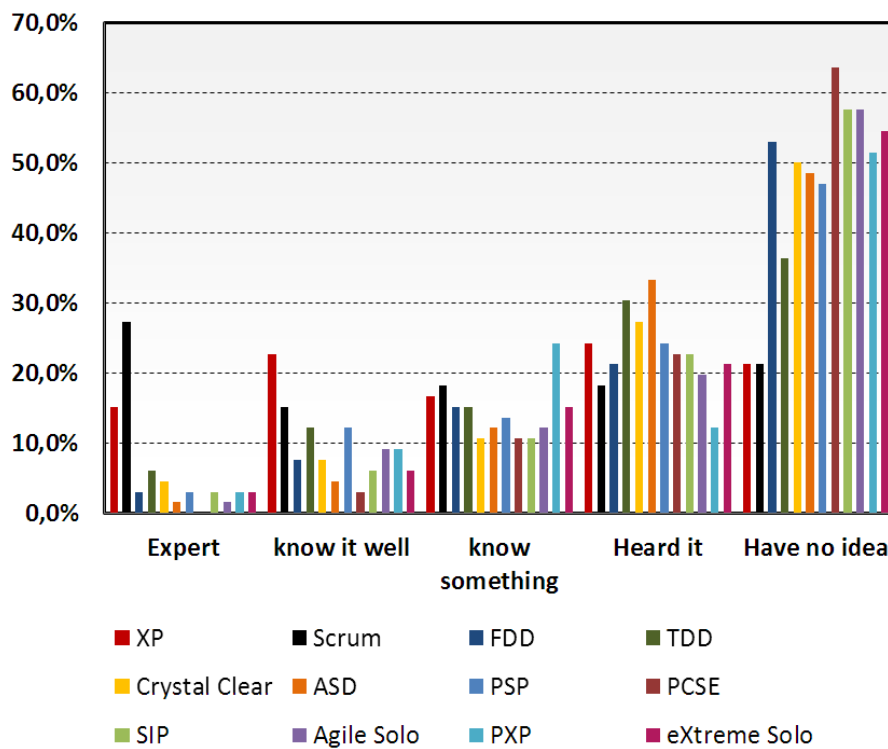


Figura 8: Do you know about this software development methodologies/processes/frameworks?

3.2.6. Análisis

Los resultados del estudio sugieren una importante y no siempre evidente diferencia entre la perspectiva del desarrollador y la del cliente en el contexto del mercado *Freelance*. El desarrollador parece que no percibiera más aspectos que el desarrollo del producto o servicio para identificar el éxito de los proyectos. No tiene en cuenta los tres aspectos de gestión de proyectos, el presupuesto, los tiempos y el alcance lo que dificulta realmente tener proyectos exitosos en la práctica.

Por otro lado se confirmó el carácter emprendedor y empresario que tiene el *Freelance*. Se encontraron también habilidades como ser proactivo, autodidacta, dinámico, flexible, capaz de adaptarse a nuevos contextos puesto que sus proyectos son de poca duración, pero ocurren de manera simultánea.

3.3. Estudios relacionados

Con el fin de realizar una correcta caracterización, se revisaron dos estudios con respecto al modelo de trabajo *Freelance*. Uno de ellos, similar al realizado en este proyecto pero con una población bastante más grande y multidisciplinar. El otro, que se ha citado varias veces en este documento, tiene que ver con aspectos para el éxito en la carrera de un *Freelance*. A continuación se indican lo que se consideró más relevante de cada estudio.

3.3.1. Reporte de la industria Freelance

Este reporte [33] se realizó en Agosto de 2012 y trata acerca de aspectos relacionados con la experiencia como *Freelance* en una población de 1500 *Freelance* de alrededor del mundo en 50 campos y profesiones diferentes. El 7.3% está relacionado con la industria del software.

Los datos más relevantes son:

- El tiempo desempeñándose como *Freelance* es de: 3-5 años un 25%, de 6-10% un 16% y más de 10 años un 34%.
- El 66% es *Freelance* de tiempo completo.
- Los aspectos que más les gusta del modelo de trabajo *Freelance* es la flexibilidad de tiempo, variedad de proyectos a ejecutar, ser su propio jefe, trabajar desde donde quieran y tomar sus propias decisiones.
- El 62% de los *Freelance* es feliz con su forma de trabajo.
- Los cuatro retos más importantes del *Freelance* son: encontrar clientes, vivir el estrés de tener mucho trabajo en algunos momentos y muy poco en otros (inestabilidad laboral), tener buena balance entre la vida y el trabajo, gestionar el tiempo y estar productivo.
- Un 60% de los *Freelance* es pagado por proyectos y 35% por horas.
- El 77% de los *Freelance* es optimista de su trabajo durante los siguientes 12 meses.
- El 74% de los clientes de los *Freelance* son empresas.

- El medio más común para encontrar clientes es a través de referencias y voz a voz.
- El 72% cree que es empresario y tienen mejores resultados por eso, pues invierten más tiempo en promover su trabajo y buscar más agresivamente clientes.

3.3.2. Los conductores para el éxito en la carrera Freelance

En este estudio [5] se realiza de una encuesta a 1600 *Freelance* en los Países Bajos con el fin de desarrollar un modelo de éxito para la carrera profesional *Freelance*. Para crear este modelo, se basa en la idea del *Freelance* como empleado y empresario, teniendo en cuenta los modelos de éxito en la literatura para las carreras individuales, así también, como para la de empresarios. Algunas de sus conclusiones son:

- El modelo *Freelance* formalmente ha sido poco abordado.
- Este modelo es una buena manera de entrar al mercado laboral.
- La tendencia de ser *Freelance* va a seguir creciendo fuertemente.
- Los aspectos de negocios, como procesos de mercadeo y de estrategia de negocios son elementos críticos en la carrera del *Freelance*.
- El capital humano²⁹, es uno de los factores determinantes en el éxito de un *Freelance*, y él como tal, se da cuenta de la necesidad de mejorar en este aspecto constantemente buscando así, la manera de mejorar y estar actualizado a las necesidades del entorno.
- El capital social³⁰ es importante también, pues es la mejor manera de conseguir proyectos fácilmente y mejores remunerados. Muchos *Freelance*, desarrollan una fuerte relación con empresas recurrentes en busca de nuevos proyectos.

3.4. Perfil del Freelance

Con la información obtenida se procedió a caracterizar al *Freelance* estableciendo sus capacidades y, sobre todo, sus necesidades con el fin de encontrar posibilidades de mejora y de apoyo desde la perspectiva de un proceso de software. Esta caracterización se dividió en aspectos personales, que indican atributos los cuales se consideró que en general, debe tener un *Freelance* para una carrera exitosa y aspectos de su contexto que indican la forma en que trabajan, como se desenvuelven y demás.

3.4.1. Características personales

Las siguientes son características identificadas en un *Freelance*.

²⁹ Se entiende como un conglomerado de conocimientos, talentos, habilidades, hábitos, atributos sociales y de personalidad y que a través del trabajo producen un valor económico.

³⁰ Se entiende como la colaboración entre diferentes grupos de colectivo humano y el uso de las oportunidades que se pueden generar de dicha colaboración.

- **Empresario:** Tiene que pensar en progresar, en mejorar procesos, en ahorrar costos, en ganar *good will*³¹.
- **Emprendedor:** Revisa constantemente ideas y proyectos propios para desarrollar por su cuenta.
- **Proactivo y retrospectivo:** Constantemente está pensando reflexionando sobre su trabajo y cada vez intenta mejorar su desempeño.
- **Ordenado:** Se toma en serio su carácter de empresa unipersonal, como tal y desea que se maneje un orden, una planificación y estrategias definidas.
- **Adaptable:** Los proyectos de un *Freelance* suelen ser relativamente pequeños y de corto tiempo, requiere adaptarse a nuevos contextos con rapidez.
- **Autodidacta:** Debe afrontar grandes retos, debido a su situación individual, de adquirir conocimiento sin mayor tutoría.
- **Recursivo:** Si no conoce lo que se debe hacer, tiene la capacidad de improvisar, de inventar soluciones prácticas.
- **Calmado y enfocado:** Mantiene la calma con proyectos aparentemente confusos e inalcanzables. Tiene que tener un nivel de tolerancia alto a la incertidumbre.
- **Simple:** Busca alternativas simples y prácticas .Evita caer en el perfeccionismo.
- **Responsable:** No se tiene un jefe, no hay supervisión ni supervisión estricta. Es su responsabilidad dedicarle tiempo al trabajo y corregir constantemente lo que está mal.
- **Carismático:** Tiene buena relaciones con sus clientes, los intenta entender y tolerar. Siempre busca en hacer buenas relaciones, los proyectos están en donde menos se piensa.

3.4.2. Características del contexto

Estas características definen el contexto del modelo de trabajo *Freelance*.

- Es usual que trabaje desde la casa principalmente, pero debe moverse frecuentemente a reuniones con el cliente.
- Debe trabajar con varios clientes así que debe adecuarse correctamente a cada uno de ellos.
- Sus proyectos son variables, cortos y simultáneos.
- Normalmente sus clientes son empresas.
- Más de un cliente a la vez.
- Usualmente recibe pagados por la totalidad de los proyectos.

3.4.3. Necesidades

De acuerdo a lo anterior, se establecieron algunas necesidades que podrían transformarse en posibilidades de mejora para aportar en el trabajo profesional del

³¹ Hace referencia a la reputación de una empresa, un servicio o producto y que puede afectar en mercado.

Freelance. Esta oportunidad de mejora se quiere abordar desde el punto de vista de un proceso software que lo forme y lo guíe. A continuación se listan las necesidades junto a un análisis y una propuesta de solución.

a. Seguimiento al esfuerzo y costo de proyectos

Al estar lejos del cliente, hay costos en los proyectos que llegan a ser importantes, como la cantidad de tiempo y dinero invertido en desplazamientos y reuniones, situación que es diferente con los empleados pues cuando por alguna razón se necesita que se desplacen a un lugar distinto a su lugar de trabajo habitual, son pagados con viáticos. Además según se evidencio anteriormente, los *Freelance* prestan poca importancia que se le da a tres aspectos considerados claves a la hora de realizar un proyecto, el presupuesto, las fechas de entrega y el alcance. La poca importancia a estos aspectos disminuye drásticamente la probabilidad de éxito de un proyecto.

Para mejorar esto se requiere de procesos que permitan hacer seguimiento y controlar los proyectos adecuadamente. Además que le permita al *Freelance* una mejor conciencia de lo que está sucediendo con su trabajo y no se enfoque solamente en el desarrollo de un producto o servicio software. Aparentemente el *Freelance* es eficaz pero no eficiente, de esta manera, se requiere de un proceso que lo haga cuestionar para que constantemente reflexione sobre su trabajo y piense ¿Cómo puedo hacerlo mejor?, ¿Cómo puedo ganar más dinero con menor esfuerzo?, ¿Cómo puedo ser más eficiente?, ¿Qué me hace falta para lograrlo?

b. Procesos de desarrollo flexible y ligero.

Al trabajar en distintos tipos de proyecto, que son relativamente cortos (5-6 meses), simultáneos, y con clientes diferentes la mayor parte del tiempo; si se quiere pretender que el *Freelance* use el proceso propuesto este debe ser poco intervencionista, flexible y ligero. Que le permitan adaptarse a su ritmo de trabajo, a las distintas características (técnicas) de los proyectos, que no consuman mucho tiempo y que de verdad lo retroalimenten.

c. Gestionar el capital humano

El capital humano es importante para que el *Freelance* pueda mantener su carácter de empresa y ser competitivo. Su problema es que al ser autónomo, no cuenta con tutores ni directrices para hacer su debido trabajo. Este es un factor importante y que muchas veces puede ser decisivo en el éxito de un proyecto. Por esta razón, el *Freelance* debe ser apoyado con la implementación de procesos de gestión de conocimiento, que lo motiven a enriquecer su capital humano.

d. Herramienta de gestión

Como se dijo anteriormente, se requiere de un proceso ligero y flexible, que no consuma mucho tiempo en ser ejecutado y que presente beneficios reales. De esta manera, el proceso propuesto debe tener una herramienta que apoye al *Freelance* en sus necesidades, logrando que tenga más conciencia de los

proyectos, y así mejorar su productividad y su eficiencia. Esta herramienta debe permitirle trabajar tener su trabajo ordenado y centralizado para que él pueda usarla desde donde quiera que esté.

e. Mercadeo y promoción de su trabajo

El *Freelance* también requiere de procesos que permitan realizar un buen mercadeo y promoción de los servicios y productos que desarrollan, pues como se vio anteriormente, son mejores los resultados cuando se buscan clientes más agresivamente y son claves para una carrera exitosa como profesional *Freelance*.

Estas necesidades son entendibles, si se tiene en cuenta que la mayoría de procesos y metodologías se enfocan más en el desarrollo como tal de un software que a su gestión. El profesional del software es educado para tener un perfil muy técnico y hacer muy bien su labor, pero al seguir modelos como el *Freelance* se requiere que cuente con habilidades complementarias. Convencionalmente, una empresa puede salir adelante porque entre sus trabajadores se tiene un conjunto de habilidades que permiten llenar cualquier exigencia técnica del mercado. Particularmente para un *Freelance* se debe potenciar las diferentes áreas de conocimiento para poder compensar las diferentes habilidades que pueden lograr un equipo de personas.

CAPÍTULO 4. DISEÑO DEL PROCESO AUDACIOUS FREELANCE

4.1. Meta proceso de formalización

Para cumplir el segundo objetivo específico y realizar una correcta formalización de un proceso software que permita solucionar las necesidades del *Freelance* se siguió, en la medida de lo posible, el meta proceso definido por Ruiz et al [18]. Este es un meta proceso ligero, de pocas actividades, que ayudan y guía en la formalización de procesos software en pequeñas compañías. Además usa el estándar de notación SPEM 2.0 [34] mediante la herramienta software EPF³², de esta manera, el proceso puede seguir siendo diseñado y especificado en el futuro. Al tratarse de un meta proceso para formalización de procesos software para pequeñas empresas, fue necesario hacer una adaptación para poder diseñar el proceso software para un *Freelance*, sin embargo, la esencia del meta proceso se mantuvo.

El meta proceso es compuesto por tres fases: *planeación*, *ejecución* y *entrega*. A continuación se mostrará brevemente cada una de estas tres fases.

4.1.1. Fase de planeación

La fase de planeación (Figura 9) del meta proceso, consiste en entender las necesidades de la empresa a la cual se le quiere diseñar un proceso software (o mejorar el que ya tienen). Para esto es necesario conocer el contexto en el que se desempeñan, sus objetivos de negocios y los *stakeholders*, para que posteriormente, se realice un plan de actividades en el cual se establecerían el alcance y la estructura que tendría el proceso formalizado. Por ultimo un entrenamiento de *stakeholders*, en cuanto a las tecnologías y conocimientos necesarios (SPEM y EPF) para la ejecución del meta proceso.

Para el caso de este proyecto, esta fase debió adaptarse a las necesidades de este estudio particular, de esta manera, la “compañía” era el modelo de trabajo *Freelance* y los *stakeholders* serían los *Freelance* como tal.

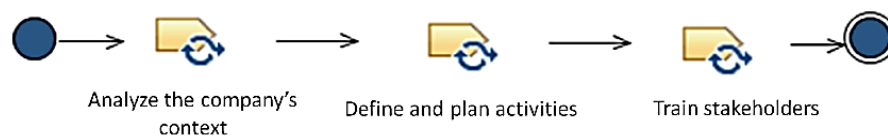


Figura 9: Fase de planeación del meta proceso

Tomado de Ruiz et al.

³² Eclipse Process Framework Componer - <http://www.eclipse.org/epf/>

Esta fase se realizó en el momento en que se analizó y caracterizó al *Freelance*, conociendo sus dinámicas frecuentes, sus necesidades y el medio en el que trabajan, pudiendo establecer oportunidades de mejora por medio de la inclusión de procesos de *gestión de proyectos*, de *gestión de conocimiento* y de desarrollo de software, entendiendo este último, como los pasos para desarrollar un producto software, o *ciclo de vida del desarrollo de software* (SDLC). De acuerdo a esto, se estableció que el proceso a diseñar, debía incluir las áreas mencionadas anteriormente.

La última actividad de esta fase, *entrenamiento de stakeholders*, no era necesaria realizarla debido a que, si bien, para propósitos académicos el proceso estaría especificado en EPF, según lo analizado, un factor importante para el éxito de este estudio, sería facilitar al *Freelance* a la adopción de este proceso, por eso, como se verá más adelante, a ellos se les mostraría una “vista” simplificada de *Audacious Freelance*.

4.1.2. Fase de ejecución

La fase de ejecución del meta proceso consta de seis actividades (Figura 10). Esta fase es la más importante y es la que contiene las actividades de obtención de información, donde se analiza y diseña el proceso, y posteriormente se especifica y se valida. La validación es iterativa, el proceso debe pasar por revisiones con el fin de establecer si realmente suple las necesidades planteadas inicialmente.

Para este proyecto, la actividad de *obtener información*, no se pudo realizar debidamente, pues en esta actividad se obtiene información sobre las tareas, artefactos, lineamientos, prácticas, y demás, que la compañía objetivo usa actualmente. Al tratarse de los *Freelance* como un grupo de profesionales heterogéneos, cada uno utiliza sus propios procesos empíricos, sin embargo, según el trabajo de caracterización se puso establecer que los métodos de desarrollo de software que mejor conocen son *Scrum* y *XP*, mostrando un indicio de qué métodos se podrían usar como referencia para construir *Audacious Freelance*. Las demás actividades fueron hechas siguiendo los lineamientos de Ruiz.

En primer lugar, se estableció los criterios que debería tener el proceso *Audacious Freelance*, de acuerdo a las necesidades establecida anteriormente, usando como referentes *XP* y *Scrum*. En segundo lugar, se diseñó un primer borrador del proceso en papel, el cual fue refinado en tres iteraciones que lo llevo hasta su versión estable. Luego *Audacious Freelance* fue especificado en la herramienta EPF, y posteriormente validado parcialmente por inspección en tres iteraciones más, hasta llevarlo a su versión final. También, es en esta fase en donde se considera y se planea una tres “vistas” de *Audacious Freelance*, con el objetivo de lograr que el proceso fuera aceptado y adoptado fácilmente por los *Freelance*.

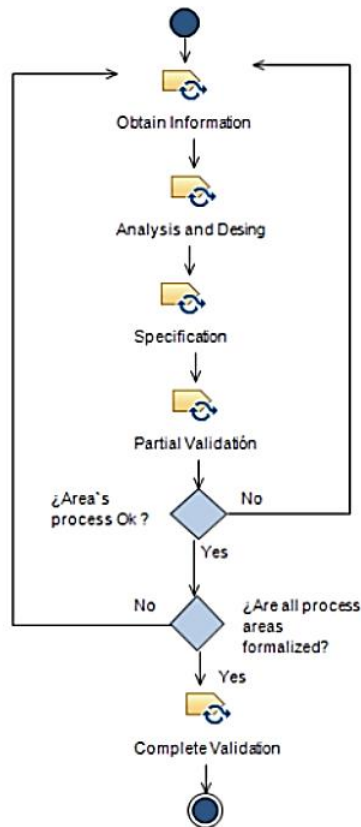


Figura 10: Fase de ejecución del meta proceso

Tomado de Ruiz et al.

La primera “vista”, es una web que contiene la especificación completa del proceso, la cual es generada automáticamente por EPF. La segunda, un documento llamativo que permite conocer los aspectos del proceso con el fin de conseguir el acercamiento y la aceptación por parte de los *Freelance*. Por último, una tercera vista, que es una herramienta software, desarrollada con la intención de que facilite la ejecución del proceso. Estas tres vistas, en conjunto, conforman los *instrumentos de soporte* de *Audacious Freelance* que más adelante, se explican de detalladamente.

4.1.3. Fase de entrega

Esta fase tiene una sola actividad (Figura 11) llamada *despliegue del proceso*. En la esta actividad, el proceso en EPF es entregado a la compañía e instalado en sus computadores para tenerlo como referencia y que dado el caso, lo sigan mejorando y adaptando a sus necesidades.

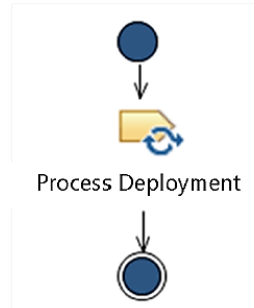


Figura 11: Fase de entrega del meta proceso

Tomado de Ruiz et al.

En este proyecto, esta fase consistió en especificar el proceso *Audacious Freelance* en EPF para asuntos académicas, pues es un activo que se pretendía entregar. Por otro lado, se entregaron *los instrumentos de soporte* al *Freelance* que realizó la evaluación del proceso en el caso de estudio descrito en el CAPÍTULO 6.

4.2. Criterios para el diseño del proceso

Contar con un proceso de software rigurosamente definido, brinda una serie de beneficios como por ejemplo ser capaz de: mejorar la gestión de proyectos y reusar el conocimiento adquirido acerca de los proyectos software ya realizados y ser más eficiente en el desarrollo de software [18].

De esta manera, y sabiendo las necesidades del *Freelance* identificadas en el capítulo anterior, se tomaron dichas necesidades y se definió la forma en que éstas serían mitigadas desde la perspectiva de un proceso software como se muestra a en la Figura 12.

De acuerdo a este análisis, se determinó que el proceso *Audacious Freelance* que se quiere diseñar, debe contar con áreas de gestión del negocio, gestión de procesos y desarrollo de software, que se denominarían las dimensiones de *Audacious Freelance*, y que sean justos para aportar a las necesidades del *Freelance*, sin consumir tiempo y esfuerzo.

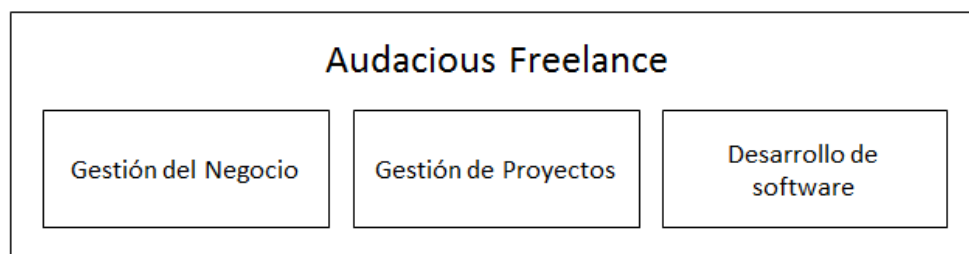


Figura 12: Dimensiones de Audacious Freelance

Tabla 1: Necesidades del Freelance y soluciones desde la perspectiva de procesos software

Necesidad	Solución desde la perspectiva de procesos
Seguimiento al esfuerzo y costo de proyectos	Procesos de gestión de proyectos, que permita realizar un seguimiento de los proyectos de manera rápida, apoyada con una herramienta de gestión.
Procesos de desarrollo flexible y ligero	Procesos de desarrollo de software, entendiéndolo como actividades de programación para construir un software, que sea ligero, poco intervencionista, y que pueda adaptarse a la naturaleza de múltiples proyectos (web, móvil, escritorio, etc.)
Gestionar el capital humano	Procesos de negocio que permitan, establecer estrategias, mejorar el perfil profesional del <i>Freelance</i> .
Herramienta de gestión	Una herramienta software que apoye en la ejecución del <i>Audacious Freelance</i> , retroalimentando al <i>Freelance</i> con información acerca de su desempeño como empresario y gestor de proyectos, concientizando al <i>Freelance</i> de la forma como trabaja.
Mercadeo y promoción de su trabajo	La herramienta software podría incluir funcionalidades que permitan mostrar el nuevo conocimiento adquirido y los proyectos finalizados a través de las redes sociales.

4.3. Modelos de referencia

Se estudió el aporte que podrían brindar los procesos que los *Freelance* afirmaron conocer más, según la encuesta realizada, estos son: *Scrum* y *XP*. Se examinaron las principales características, actividades y artefactos de *Scrum*, con el fin de determinar si la inclusión de éstos podría aportar en la construcción de *Audacious Freelance*. Para *XP*, se tomó en cuenta el trabajo revisado en el estado del arte, *Personal Extreme Programming (PXP - Dzhurov)* el cual realiza una adaptación de *XP* a una sola persona.

Scrum y *XP* son dos métodos ágiles (framework y metodología, respectivamente), que frecuentemente son combinados para desarrollar proyectos software. Esto es posible, debido a que *Scrum* se enfoca en las prácticas de organización y gestión, mientras *XP* se centra en las prácticas de la programación de software y es por eso que, aunque las dos tienen actividades que se solapan, se complementan muy bien y por eso es usual que se ejecuten en conjunto [35].

4.3.1. Aporte de Scrum a Audacious Freelance

Scrum [36] es un framework ágil de desarrollo de software iterativo e incremental creado por Jeff Sutherland y Ken Schwaber en 1995. Entre sus características esta que es ligero, fácil de entender pero difícil de manejar. Algo que se defiende mucho, es que *Scrum* no es un proceso sino una framework en el que se puede integrar varios procesos y técnicas [37].

En la Figura 13 se muestra el ciclo de vida de *Scrum* de manera simplificada.

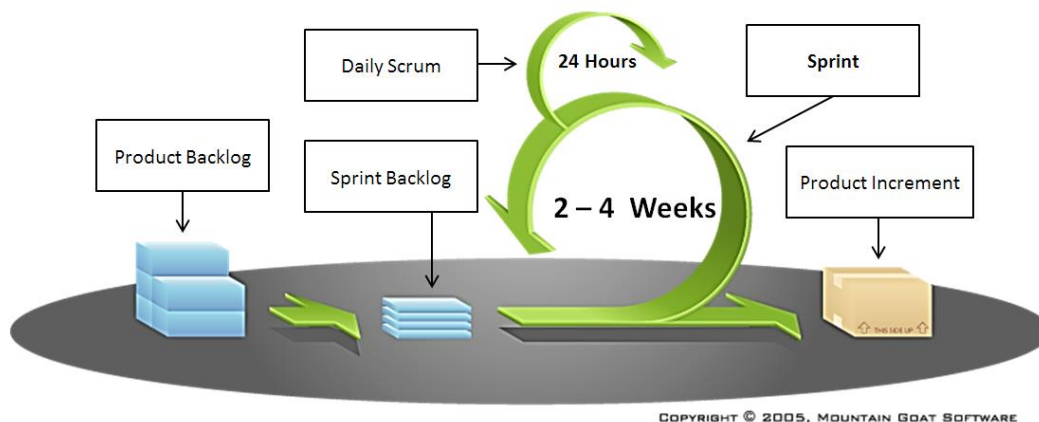


Figura 13: Ciclo de vida de Scrum
Tomado de Mountain Goat Software

Scrum hace uso de tres artefactos simples. Primero y más importante, el *Product Backlog* que es una lista ordenada que contiene todo los requerimientos del producto. El segundo es el *Sprint Backlog*, que es una lista de elementos seleccionados del *Product Backlog*, que han sido planeados con el fin de ser ejecutados en un *Sprint*. El último artefacto es el *Increment*, el que es una suma de todos los elementos del *Product Backlog* que han sido completados en un *Sprint*.

El *Sprint* es, según sus creadores, el corazón de *Scrum*. Es un intervalo de tiempo fijo en el que se desarrolla como tal el producto software. Durante el *Sprint* otros eventos son realizados con el fin de realizar la planeación y gestión del desarrollo como el *Sprint Planning*, *Daily Scrums*, *Sprint Review* y *Sprint Retrospective*.

De acuerdo a esto, se analizó en la Tabla 2 y la Tabla 3 una posible inclusión de estos aspectos podría aportar para *Audacious Freelance* que se quería desarrollar.

Tabla 2: Análisis de los eventos de Scrum

Evento de Scrum	Análisis
Sprint	Este evento es importante y sin dudas se debería incorporar, pero solo adoptando lo esencial, que es desarrollo de los elementos del <i>Sprint Backlog</i> , dejando a un lado características como: intervalo de tiempo fijo, imposibilidad de cambiar requerimientos en el transcurso del <i>Sprint</i> , y la cancelación del curso de un <i>Sprint</i> .
Sprint Planning	Este evento, también se debería incorporar. Es importante debido a que es en donde se planea y construye el <i>Sprint Backlog</i> .
Daily Scrums	Este evento podría aportar, y podría ser usado como una actividad que se ejecute diariamente y que sirva no solo para ver el estado del proyecto, sino el estado en general del <i>Freelance</i> (aspectos de negocio, proyectos y desarrollo)
Sprint Review	Este evento es importante y podría aportar debido a que es donde el <i>Freelance</i> y los <i>stakeholders</i> se encuentran para hablar de lo ocurrido en el <i>Sprint</i> y mostrar resultados.
Sprint Retrospective	Si bien para <i>Scrum</i> es importante este evento al final de cada <i>Sprint</i> , podría usarse más bien, cuando se finalice el proyecto, con el fin de retroalimentarse y así no aumentar la carga del <i>Freelance</i> .

Tabla 3: Análisis de los artefactos de Scrum

Artefactos de Scrum	Análisis
Product Backlog	Estos Artefactos pueden ser combinados en uno solo y así reducir la carga al <i>Freelance</i> . De esta manera se tendría un documento en donde se listen todas las requerimientos del producto, que puedan ser priorizados y planeados y que además, muestre el avance del proyecto.
Sprint Backlog	
Increment	

4.3.2. Aporte de eXtreme Programming a Audacious Freelance

eXtreme Programmig [38] es una metodología liviana de desarrollo de software creada por Kent Beck. Es uno de los procesos de desarrollo más destacado. *XP* está basado en una serie de valores y doce prácticas que propician un aumento en la productividad de software. Son muchas las adaptaciones que se han hecho de *XP* a otros entornos, y uno de ellos es el trabajo realizado por Dzhurov (PXP) [1], en el que propone un procesos de desarrollo de software para una sola persona adoptando seis prácticas de *PSP* para apoyar la gestión de procesos y seis prácticas de *XP* para apoyar al área de desarrollo de software. Las prácticas escogidas por Dzhurov se muestran a continuación:

- **Continuous integration:** Esta práctica consiste en realizar integraciones continuas de funcionalidades al código principal del proyecto con el fin de tener actualizado los cambios realizados para poder realizar pruebas a todo el proyecto y verificar su correcto funcionamiento. Esto incluye actividades de versionamiento, pruebas automatizadas y reportes de defectos automatizados.
- **Simple design:** Esta práctica es justa y necesaria para un desarrollador *Freelance* debido a que está solo, y requiere enfocarse en su trabajo actual sin preocuparse de necesidades futuras.
- **Small Releases:** Esta práctica se podría incorporar a *Audacious Freelance*, para que de esta manera se desarrollen pequeñas funcionalidades del proyecto y constantemente se busque la retroalimentación del proyecto.
- **Refactoring:** Es una técnica de mejoramiento del diseño del código fuente con el fin de refinar y mejorar la seguridad, desempeño, estructura y legibilidad del código.
- **Testing:** Una de los fuertes de *XP* es que el desarrollo es dirigido por pruebas. Antes de que cualquier funcionalidad sea integrada, se debe realizar pruebas con el fin de verificar el software como un todo.
- **Spike Solutions:** Aunque no es una práctica de *XP* como tal, se trata de una práctica en la cual se realizan soluciones rápidas y a manera de pruebas para ver si solucionan o no un problema particular.

Tomando en cuenta el trabajo y la selección de Dzhurov, en donde tomas las prácticas de XP que no tienen que ver con trabajo grupal, se analizó, en la Tabla 4, dichas prácticas para ver que podrían aportar para el diseño de *Audacious Freelance*.

Tabla 4: Análisis de las prácticas de XP

Práctica de XP	Análisis
Continuous integration	Esta práctica podría ser incorporada trabajando de la par con el Sprint. Realizando iteraciones cortas e integraciones constantes.
Simple design	Esta práctica es justa y necesaria para un desarrollador Freelance debido a que está solo, y requiere enfocarse en su trabajo actual sin preocuparse de necesidades futuras.
Small Releases	Esta práctica se podría incorporar a <i>Audacious Freelance</i> , para que de esta manera se desarrollen pequeñas funcionalidades del proyecto y constantemente se busque la retroalimentación del proyecto.
Refactoring	Al realizar desarrollo rápido de funcionalidades con el fin de establecer aprobación del cliente, se dejan a un lado ciertos aspectos de diseño del código que posterior a la aprobación, deberían ser resueltos.
Testing	El hacer pruebas como lo dice XP, con <i>unittest</i> podrían consumir algo de tiempo al Freelance, y no las tendría en cuenta, sin embargo, pruebas con el cliente y de <i>caja negra</i> podrían ser realizadas.
Spike Solutions	El buscar alternativas de manera rápida para solucionar un problema particular, podría ser una buena manera de abordar un problema y se tendrían que incorporar al proceso.

4.4. Diseño iterativo

De acuerdo al meta proceso de Ruiz, se iniciaba una labor iterativa de diseño y es por eso que después de conocer las necesidades del *Freelance* y el aporte que podrían brindar otros métodos como *Scrum* y *XP*, se realizó el diseño de *Audacious Freelance*. Estos diseños pasaron por inspecciones visuales con el fin de mejorar, agregar o eliminar actividades y tareas que fueron refinando el proceso a su versión estable en SPEM con la herramienta EPF en donde se terminó de especificar y refinar.

En esta sección se quiere mostrar a manera de ejemplo, el diseño iterativo que tuvo la evolución del proceso *Audacious Freelance*, particularmente, la vista del proceso desde la perspectiva del diagrama en SPEM junto con unos breves comentarios sobre cada versión.

4.4.1. Audacious Freelance versión 0.5

Esta fue la primera versión del proceso. En esta versión solo se tuvo en cuenta la dimensión de *desarrollo de software* y se detalló hasta el punto de actividades; las tareas no se tuvieron en cuenta. También algunos artefactos fueron definidos.

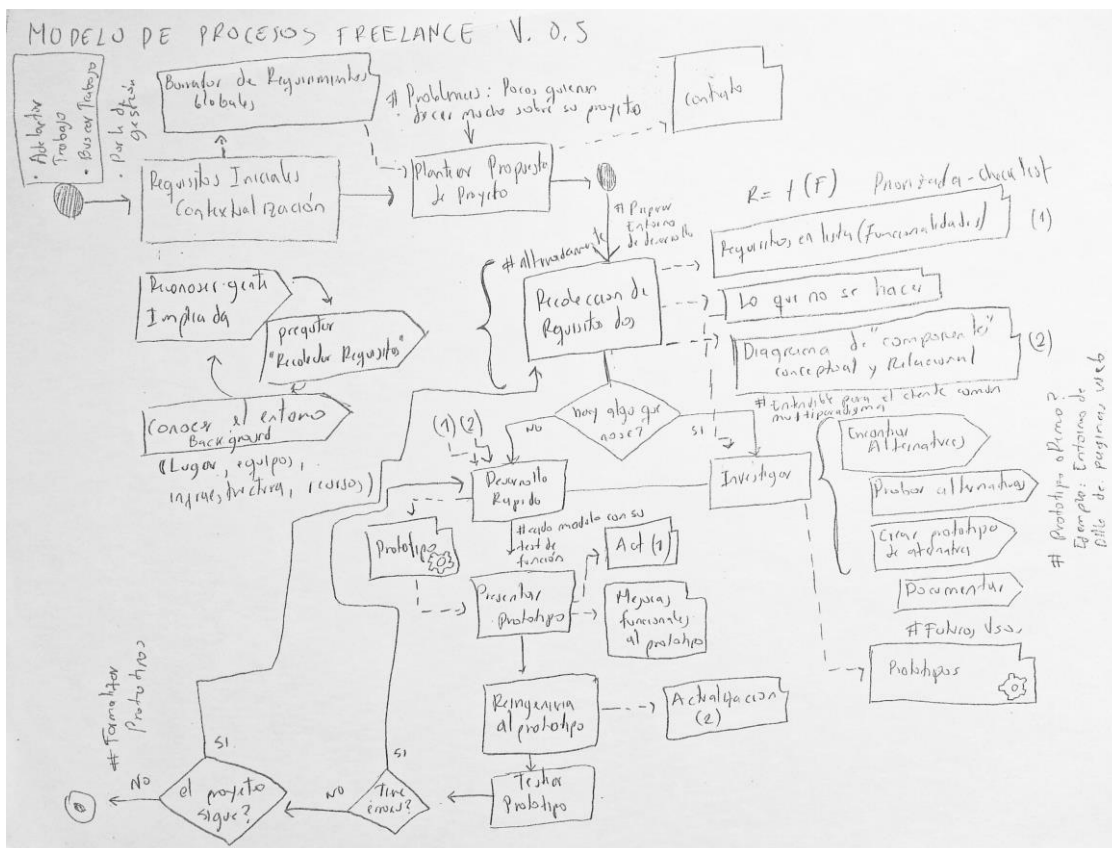


Figura 14: AF v. 0.5 Desarrollo de software

4.4.2. Audacious Freelance versión 0.8.

En la versión 0.8 de *Audacious Freelance*, la dimensión de *gestión de proyectos* fue diseñada. La dimensión de *desarrollo de software* se simplificó y se reordeno lo pasos lógicos del proceso. Algunas actividades se detallaron en tareas y se especificaron.

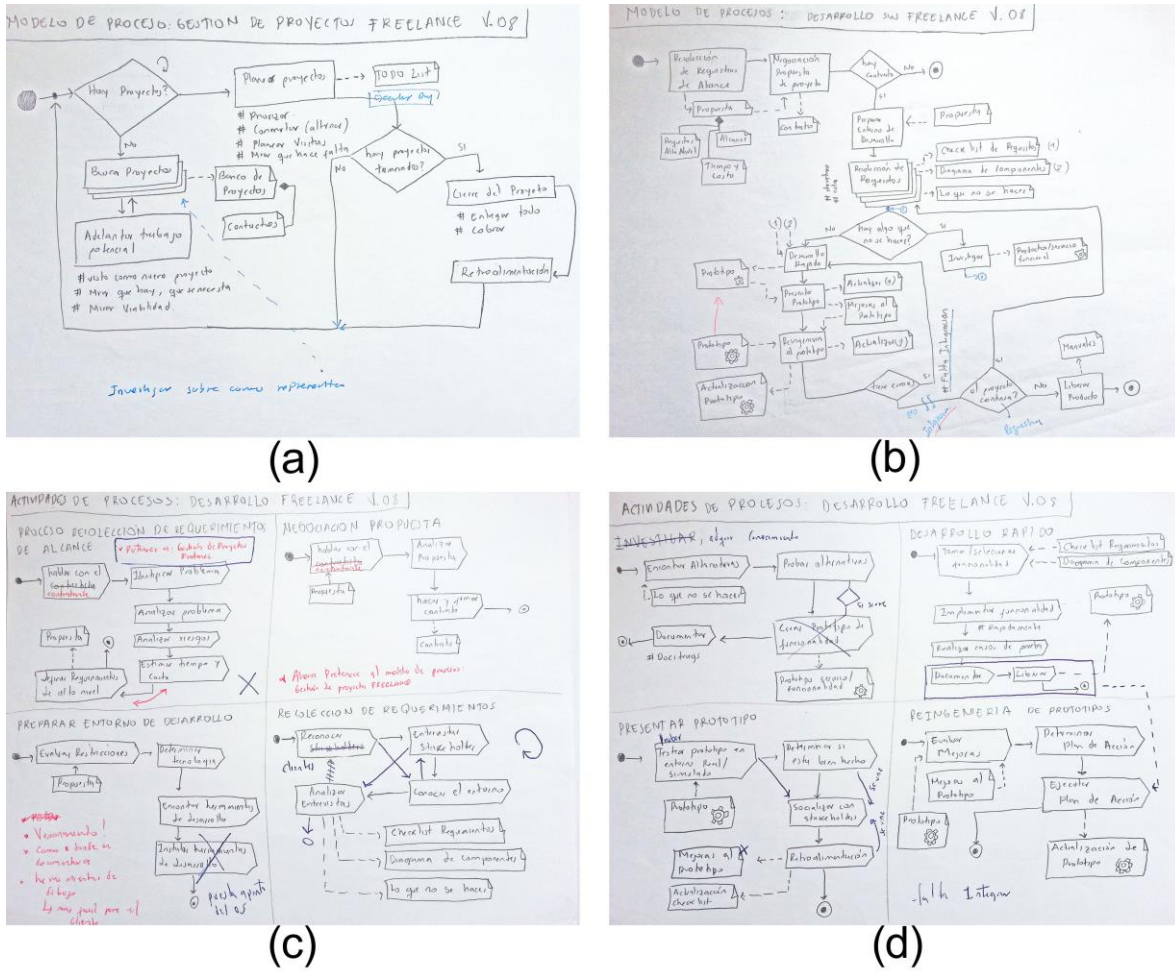


Figura 15: AF v 0.8
(a) Gestión de proyectos, (b) Desarrollo de software, (c) Tareas parte 1, (d) Tarea parte 2

4.4.3. Audacious Freelance versión 1.0

En esta versión, se realizaron cambios en cuanto a flujo lógico del proceso. Algunas actividades cambiaron de dimensión como: *recolección de requisitos de alcance* y *negociación de la propuesta*. Se crearon algunas actividades como: *iniciar proyecto* y *ejecutar proyecto*. Se terminaron de definir y especificar las tareas que hacían falta.

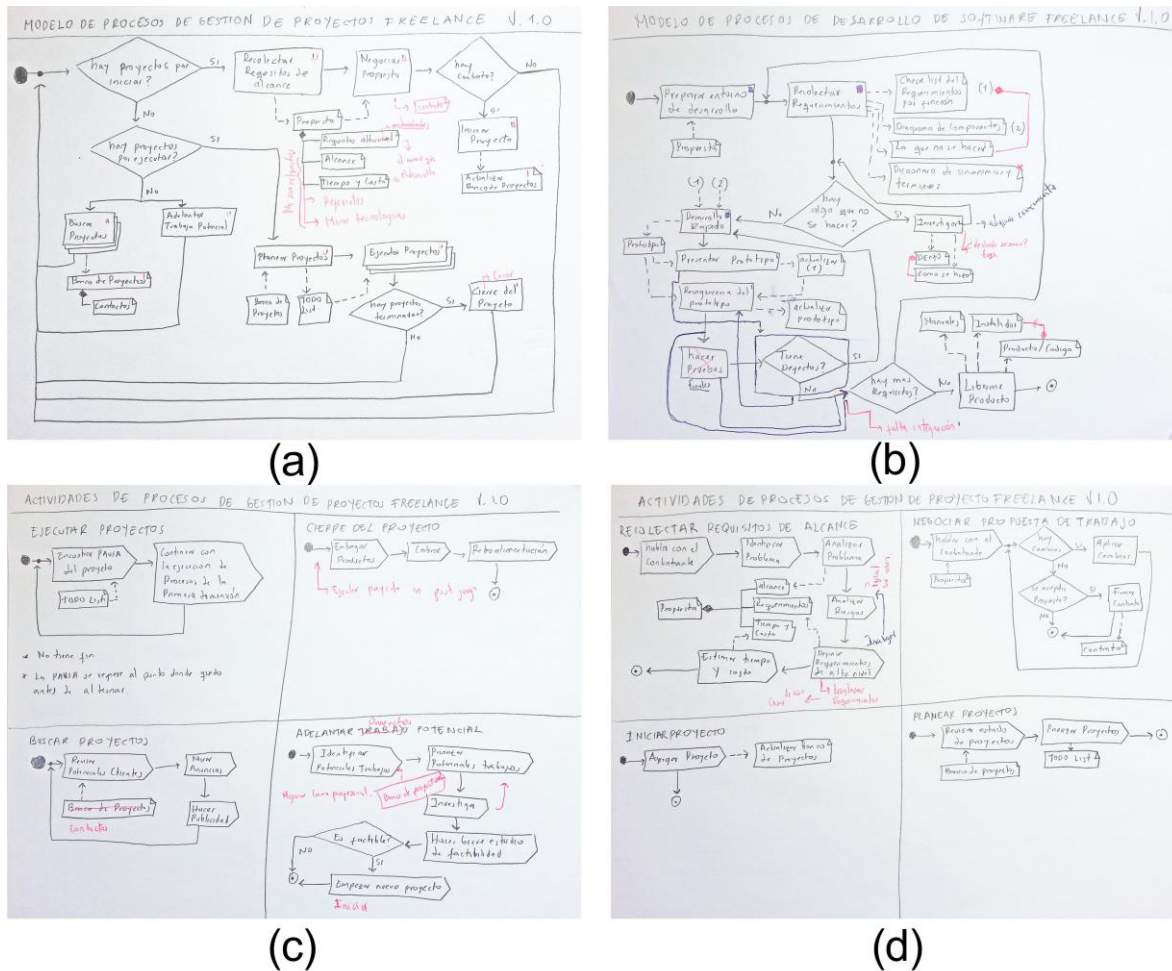


Figura 16: AF v 1.0.
 (a) Gestión de proyectos, (b) Desarrollo de software, (c) Tareas parte 1, (d) Tarea parte 2

4.4.4. Audacious Freelance versión 2.0

Para esta versión, el proceso *Audacious Freelance* fue modelado en la herramienta Eclipse Process Framework Composer - EPF. Para esta versión, se diseñó y se creó la dimensión de *gestión del negocio*. Además se refinaron todos los artefactos y se especificaron los roles. En la dimensión de *desarrollo de software* se dividió en tres fases para un mejor entendimiento.

Junto con esto, se crearon los *instrumentos de soporte* de *Audacious Freelance*, los cuales también estarían enlazados desde EPF.

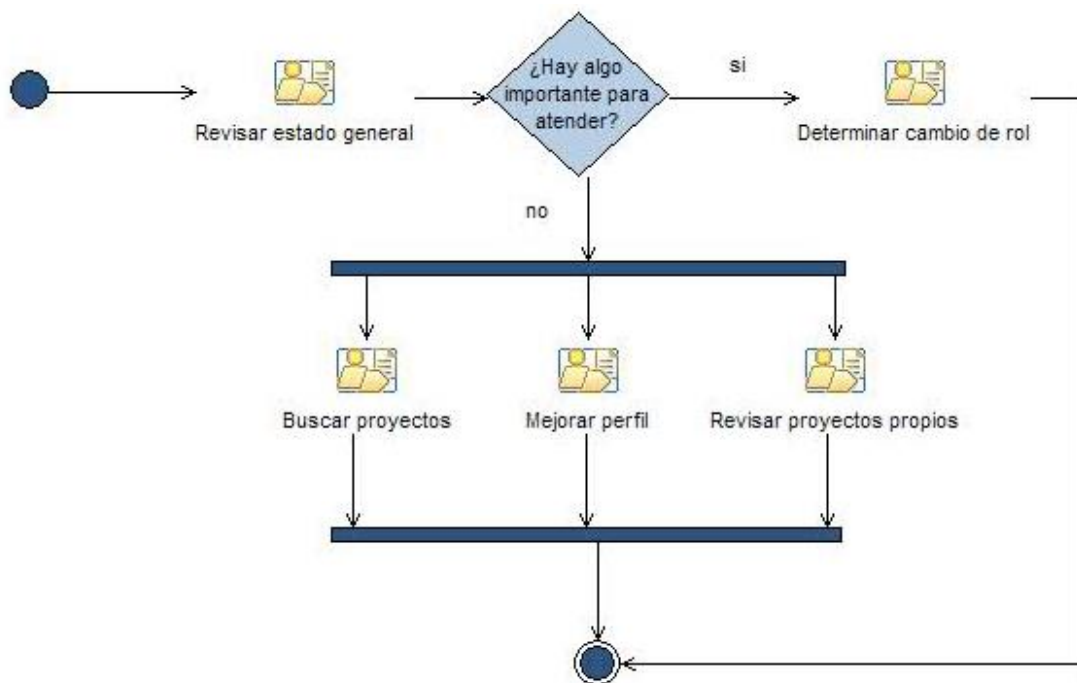


Figura 17: AF v. 2.0 Gestión del negocio

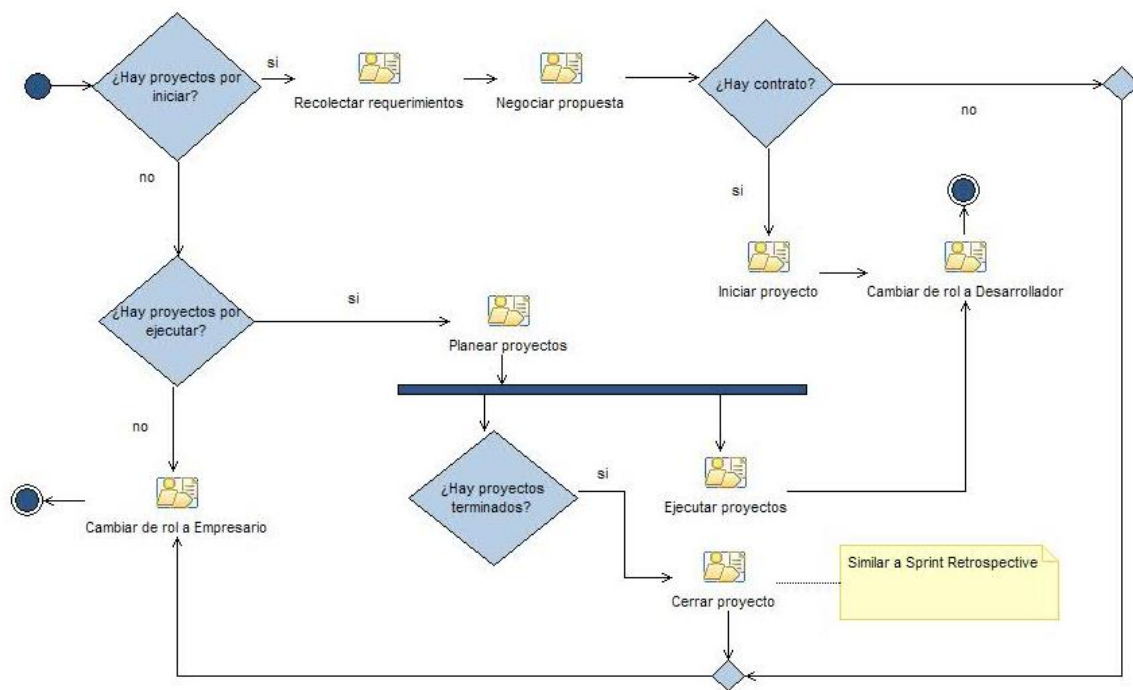


Figura 18: AF v. 2.0 Gestión de proyectos



Figura 19: AF v. 2.0 Desarrollo de software - Fase pre-juego

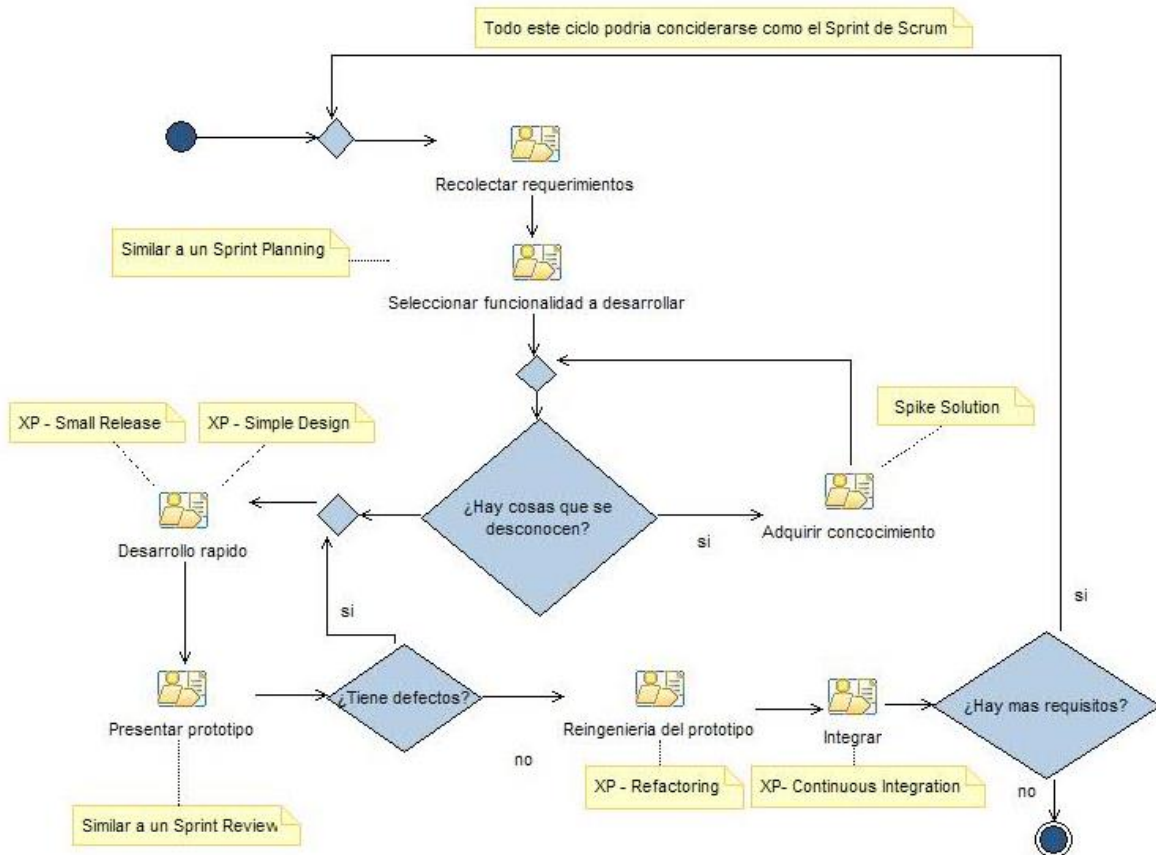


Figura 20: AF v. 2.0 Desarrollo de software – Fase juego



Figura 21: AF v. 2.0 Desarrollo de software - Fase post-juego

CAPÍTULO 5. AUDACIOUS FREELANCE – AF

En esta sección, se detalla la última versión (2.0) del proceso *Audacious Freelance*, mostrando aspectos de éste.

5.1. Introducción

Audacious Freelance (AF) es un proceso de desarrollo de software ligero, diseñado para que desarrollador *Freelance* lo ejecute diariamente. El nombre *Audacious Freelance*, que en español significa el *Freelance Audaz*, fue escogido debido a que se considera que este adjetivo resume las características que tiene que tener un *Freelance* para ejercer su trabajo; inteligente, estratega, reflexivo, una persona que busca abrirse camino en un mundo competitivo y que necesita de todo su potencial.

La intención de AF es ayudar al *Freelance* enfrentar y mitigar problemas y necesidades propias de su modelo de negocio y direccionarlo a una mejora continua. AF es ligero y altamente flexible; se diseñó para ser aprendido rápidamente y ser ejecutado de tal manera que no interfiera en las habilidades naturales del *Freelance*, sino más bien, para complementar sus capacidades en aspectos y áreas del desarrollo de software, la gestión de proyectos y la gestión del negocio.

AF se divide en tres dimensiones, cada una de ellas se enfoca en un ámbito de su modelo de negocios. Estas dimensiones fueron diseñadas para ser poco intervencionista, es decir, para evitar inmiscuir en la manera como un *Freelance* ha venido trabajado, más bien, lo que AF intenta es ayudar a mejorar su forma actual de hacer las cosas, potenciando al *Freelance* para ayudarlo a hacer más competitivo.

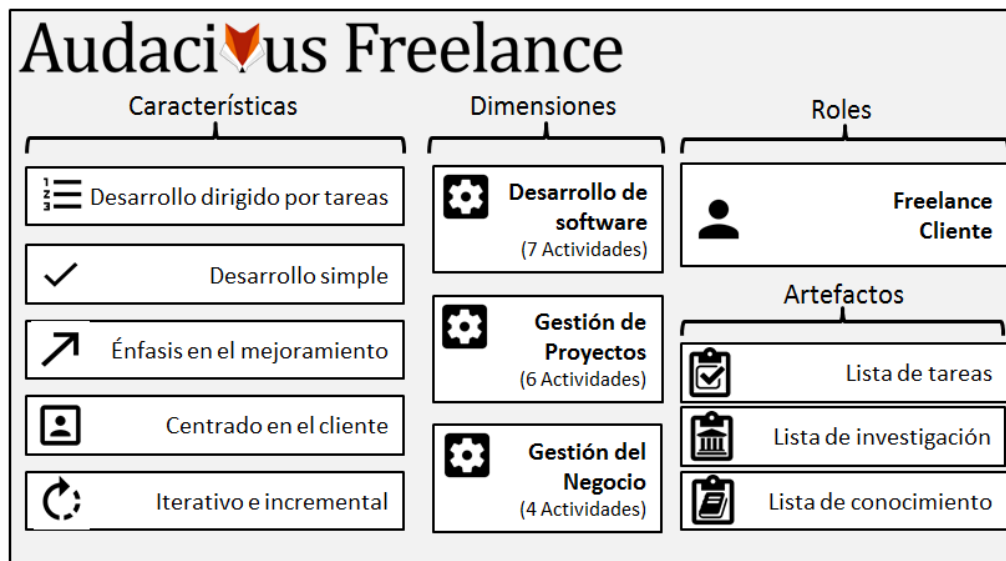


Figura 22: Elementos de Audacious Freelance

5.2. Enfoque

Audacious Freelance sigue un enfoque Ágil, es decir, se rige bajo los principios del Manifiesto por el Desarrollo Ágil del Software, debido a esto es necesario que el *Freelance* que quiera adoptar AF, conozca sobre estos principios.

5.3. Propósito

El propósito de *Audacious Freelance* es lograr que el desarrollador tome conciencia sobre el estado de sus proyectos y su desempeño para de esta manera mejorar su trabajo permitiéndole incrementar su productividad y su competitividad. Para esto, AF se enfoca en:

- Hacer seguimiento de los proyectos y medir su desempeño.
- Permitir tomar buenas decisiones en base a conocimiento.
- Controlar proyectos simultáneos, ayudando al *Freelance* en la gestión de éstos.
- Motivar al *Freelance* en una constante mejora de su perfil como profesional.
- Estandarizar y reflexionar sobre el cómo se están haciendo las cosas.
- Formar al *Freelance* para ser un buen profesional.

5.4. Características

- **Énfasis en el mejoramiento del proceso:** Cada actividad que se realice se debe reflexionar en busca de mejorarla.
- **Centrado en el cliente:** Cada decisión tomada debe ser basada en las necesidades del cliente, sus objetivos, expectativas y motivaciones.
- **Iterativo e incremental:** El problema se desglosa en partes, y se va construyendo poco a poco el producto final.
- **Desarrollo dirigido por tareas:** La unidad básica de medición del desempeño en un proyecto son las tareas. Cada funcionalidad se desglosa hasta ser un listado de tareas.
- **Desarrollo simple:** Las soluciones a los problemas responden siempre a la simplicidad de su ejecución.

5.5. Filosofía

La filosofía de AF se puede explicar con la expresión “*work smarter not harder*”³³, que básicamente significa que se debe trabajar inteligentemente y no fuertemente, esto es, en el ambiente del desarrollo de software, que al momento de enfrentarse a un nuevo proyecto, en lugar de poner todo el entusiasmo desmesuradamente en la codificación, primero se debe pensar y planear las mejores alternativas y realizar buenas decisiones,

³³ Daniel H. Pink

con el fin de aumentar la productividad al máximo. Hay momentos que se piensa que entre más fuerte se trabaje, mejor serán los resultados, pero para aumentar aún más la posibilidad de buenos resultados, más que trabajar fuertemente, se necesita trabajar inteligentemente.

Trabajar inteligentemente ayuda a ser más eficiente y eficaz, a optimizar al máximo el esfuerzo y de esta manera poder realizar más cantidad de trabajo.

5.6. Las tres dimensiones

AF cuenta con tres dimensiones que se ejecutan diariamente, total o parcialmente. Cada dimensión está dispuesta para cubrir un ámbito del trabajo del *Freelance* para hacerlo más competitivo. Las dimensiones son:

a. Desarrollo de software

Tiene siete actividades con el objetivo es desarrollar eficientemente software, desglosando el proyecto en tareas y gestionándolas y junto a un conjunto ordenado de prácticas, intentar mejorar la calidad del producto desarrollado.

b. Gestión de proyectos

Tiene 6 actividades. Su objetivo incrementar la productividad por medio de un control eficiente de proyectos para planificar, iniciar, ejecutar y terminarlos.

c. Gestión del negocio

Tiene 4 actividades. Su objetivo mejorar la capacidad del *Freelance* en cuanto a conocimiento técnico y manejo de herramientas. También se revisan la viabilidad de proyectos propios. Estas actividades propuestas para aumentar el capital humano del *Freelance* y a auto promoverse con el fin de hacerlo más competitivo.

5.7. Artefactos

Los artefactos que *Audacious Freelance* usa, son herramientas (documentos) que ayudan a consignar conocimiento e información perteneciente a cada una de las tres dimensiones. Son pocos artefactos y simples para aligerar la carga que se le pueda ocasionar al *Freelance*. Hay tres artefactos importantes en *Audacious Freelance* y que no pueden ser omitidos, estos son: *lista de tareas*, *lista de investigación* y *lista de conocimiento*. A continuación se muestran los detalles de estos y los productos de trabajo.

a. Lista de tareas

Es el artefacto principal de AF. Es una *checklist* de tareas asociadas a requerimientos o funcionalidades. Cada proyecto tiene una lista de tareas particular. La intención de este artefacto, es medir el esfuerzo y el progreso del proyecto con el fin de incrementar la conciencia del *Freelance* acerca de su desempeño y de esta manera tomar buenas decisiones para planificar y priorizar proyectos. También, llevar seguimiento a el estado actual del proyecto para que

así, al estar alternando en múltiples proyectos, se tenga un listado de lo que se quiere y debe hacerse y no se olvide ninguna funcionalidad a desarrollar.

b. Lista de investigación

Este artefacto permite llevar un listado de tecnologías o conocimientos que se pretenden aprender con el fin de mejorar el perfil del *Freelance*. Con esto, se quiere que el *Freelance* se motive a estar constantemente mejorando sus conocimientos volviéndolo más competitivo.

c. Lista de conocimiento

Este artefacto es un consolidado de conocimiento que el *Freelance* ya ha ido refinando de acuerdo a su experiencia. Su propósito es incrementar la productividad al sintetizar conocimiento y mostrarlo a manera de resumen. De esta manera, al empezar nuevos proyectos, el *Freelance* podrá recurrir a su libro de conocimiento en donde encontrar un tutor (ya sea escrito por el o por un tercero) que le ayudará a realizar rápidamente su labor.

d. Productos de trabajo

Hay productos de trabajo (*Work products*) propios de la naturaleza de la ejecución de contratos y de desarrollo de software que si bien están especificados en la herramienta EPF, no se considera necesario mencionarlos detalladamente. Estos artefactos son: *Lista de Contactos, Demo, Contrato, Manuales, Prototipos, Propuesta, documento de requerimientos, diagrama de componentes*. El detalle de los productos de trabajo se encuentra disponible en la descripción formal del proceso en EPF.

5.8. Roles

Audacious Freelance tiene tres roles que el *Freelance* debe desempeñar y uno para el cliente. Cada rol está asociado a una dimensión y en marca el ámbito en el cual se va a desarrollar.

a. Freelance Desarrollador

Se desempeña en la dimensión de desarrollo de software. Realiza actividades asociadas al desarrollo como tal del producto/servicio software como: analista, programador, *tester*.

b. Freelance Administrador de proyectos

Se desempeña en la dimensión de gestión de proyectos. Realiza actividades asociadas a la gestión de proyectos, planeándolos y priorizándolos de acuerdo a la información recolectada en la lista de tareas.

c. *Freelance Empresario*

Se desempeña en la dimensión de gestión del negocio. Realiza actividades dedicadas a mejorar el perfil y a encontrar nuevos y posibles proyectos. También toma decisiones diariamente sobre las acciones a tomar a todas las situaciones.

d. *Cliente*

Para Audacious Freelance, el cliente será cualquier *stakeholder* que represente al cliente.

5.9. Ciclo de vida

Audacious Freelance es un proceso destinado a ser ejecutado diariamente. Las actividades a ejecutar están a criterio del *Freelance* sin embargo, algunas de ellas se consideran obligatorias. A continuación, en la Figura 23 se muestra el ciclo de vida de *Audacious Freelance*, en una versión resumida del proceso global.

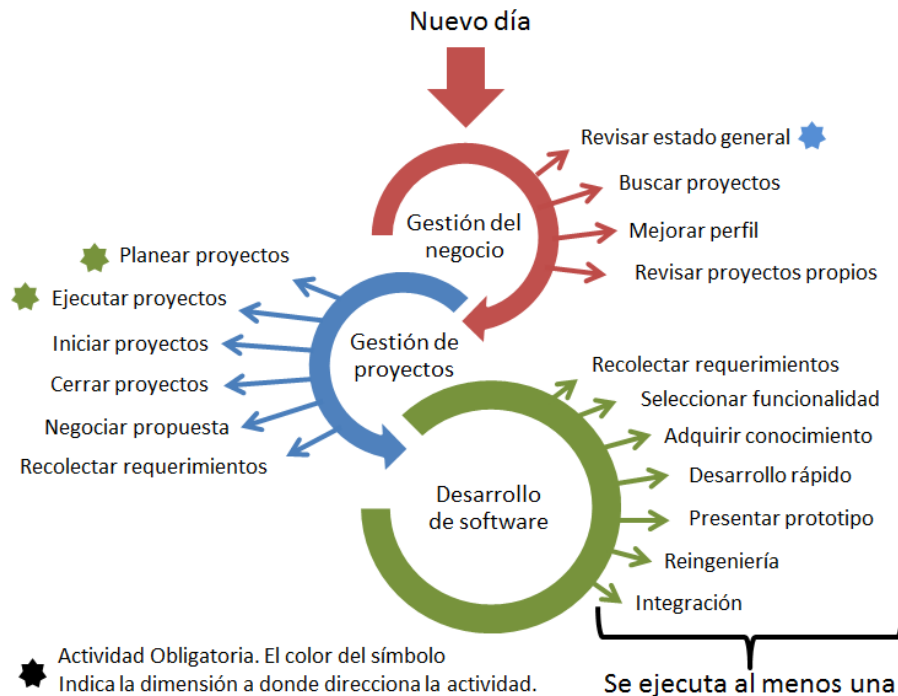


Figura 23: Ciclo de vida de Audacious Freelance

El proceso se ejecuta diariamente comenzando cada nuevo día desde la dimensión de *gestión de negocio*, en la que empieza a tomar decisiones con respecto a los proyectos a actuales de acuerdo a los reportes. Posteriormente va avanzando a la dimensión de *gestión de proyectos* en la que también realiza decisiones y finalmente terminando en la dimensión de *desarrollo de software*.

5.10. Especificación del proceso – EPF

Como se mencionó anteriormente, el proceso Audacious Freelance se diseñó y especificó en la herramienta Eclipse Process Framework Composer - EPF, usando SPEM 2.0 que es un estándar de la OMG³⁴ para la especificación y formalización de procesos.

La estructura global de EPF es la *Library*, esta contiene a su vez, *Plugins* y *Configurations*. Los *Plugins* tienen *Method Contents*, que son la estructura principal (ver Figura 24) de EPF para agrupar información y ordenar la información sobre el proceso (nombres, descripciones, tareas, guías, roles, productos de trabajo, etc.). Los *Plugins* además tienen *Processes* en donde se definen las actividades establecidas en los *Method Content* y se describe el *workflow* y el diagrama en SPREM. Las *Configurations* sirven para tomar la información establecida en los *Method Contents* y poderlas adaptar según las necesidades y el entorno en donde se piense ejecutar el proceso.

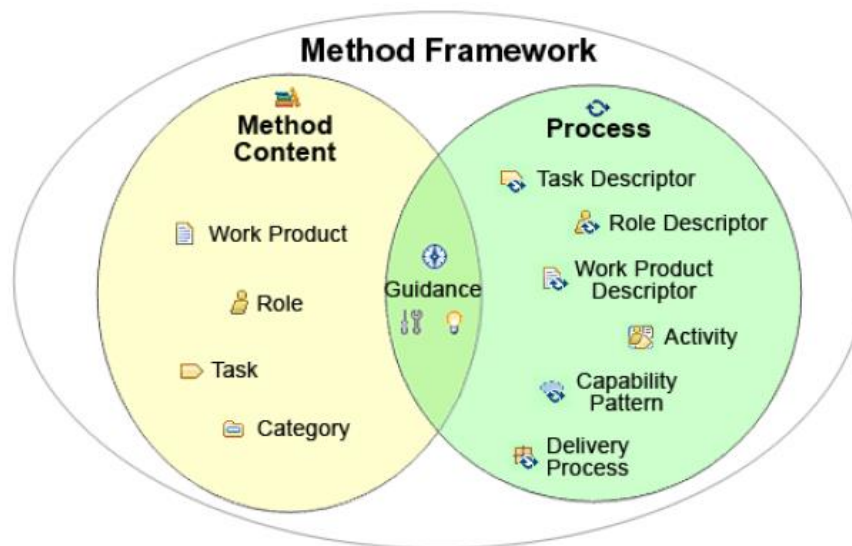


Figura 24: EPF Method Framework

Tomado de EPF Tutorial User Manual

Para *Audacious Freelance* se creó una sola *Library*, un solo *Plugin* y una sola *Configuration*; varios *Content Method* fueron creados. Agrupándolos en las tres dimensiones. Un *Content Method* para cada una de las actividades. Se crearon tres *Capability Patterns* que es una por dimensión y básicamente muestran el *workflow* y el diagrama en SPEM con las actividades, tareas, roles y workproducts asociadas. En la

³⁴ **Object Management Group**

Figura 25 se muestra esta arquitectura. Hay que especificar que dentro de los *Method Contents* de cada dimensión, están especificadas las actividades respectivas de las dimensiones en forma de *Method Contents*, y es en estas donde están especificadas las tareas de *Audacious Freelance*. En el *Anexo B* se encuentra la *Library* que se generó con EPF.

5.11. Diseño de instrumentos de soporte

Los *instrumentos de soporte* de *Audacious Freelance* con la intención de que el *Freelance* pueda conocer y potencialmente, adoptar el proceso. Esta decisión se tomó porque se considera que una de las claves del éxito de este proyecto sería la aceptación, entendimiento y adopción del proceso. Estos instrumentos fueron usados en el momento de la evaluación, es decir, al *Freelance* le fue transparente la especificación realizada en EPF.

Con el fin de establecer un estilo estético que fuera usado en los tres instrumentos para darle formalismo e identidad, se diseñó un logo que sería usado en cada *instrumento de soporte* (ver Figura 26)

Audacious Freelance Library

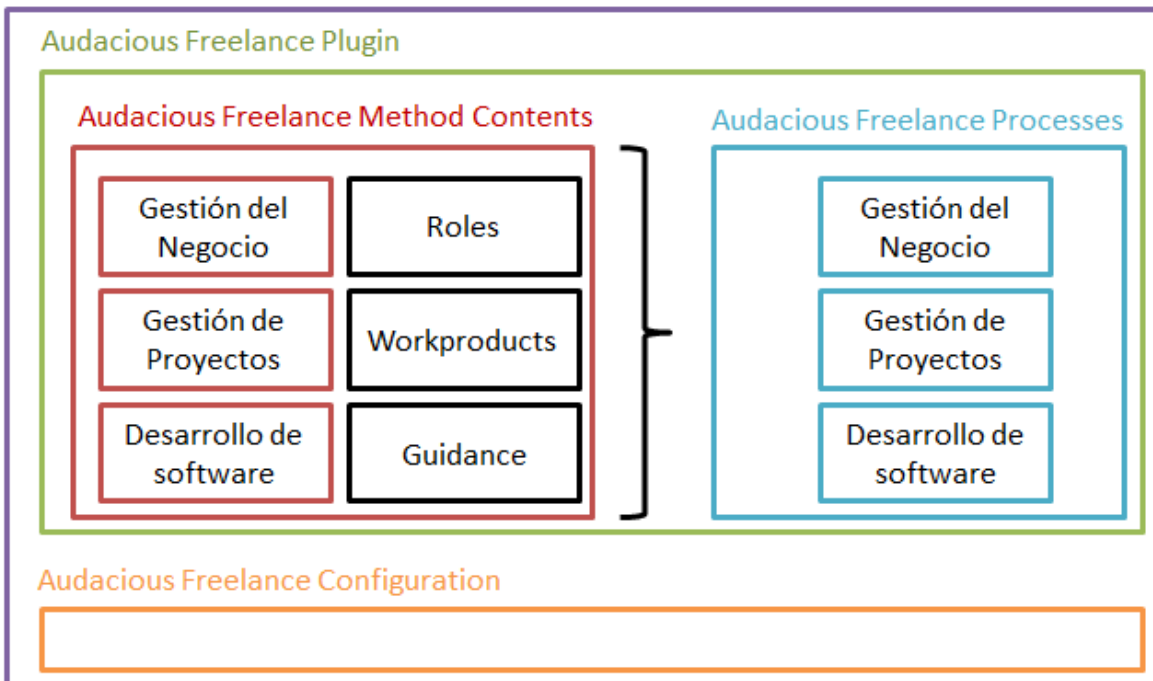


Figura 25: Arquitectura de Audacious Freelance en EPF

Audacious Freelance

Figura 26: Logo de Audacious Freelance

5.11.1. Documentación web

Este instrumento, es una aplicación web autogenerada por EPF. Básicamente EPF toma la información que se decide publicar, asociándola a una estructura llamada *Custom Category*. Para *Audacious Freelance* se creó una sola *Custom Category* en la que se asociaron algunos objetos de *Guidance*, los *workproducts*, roles y los tres *Capability Patterns*, uno por cada dimensión. En la Figura 27 se muestra la página inicial de este instrumento. La web generada se encuentra en el Anexo C.

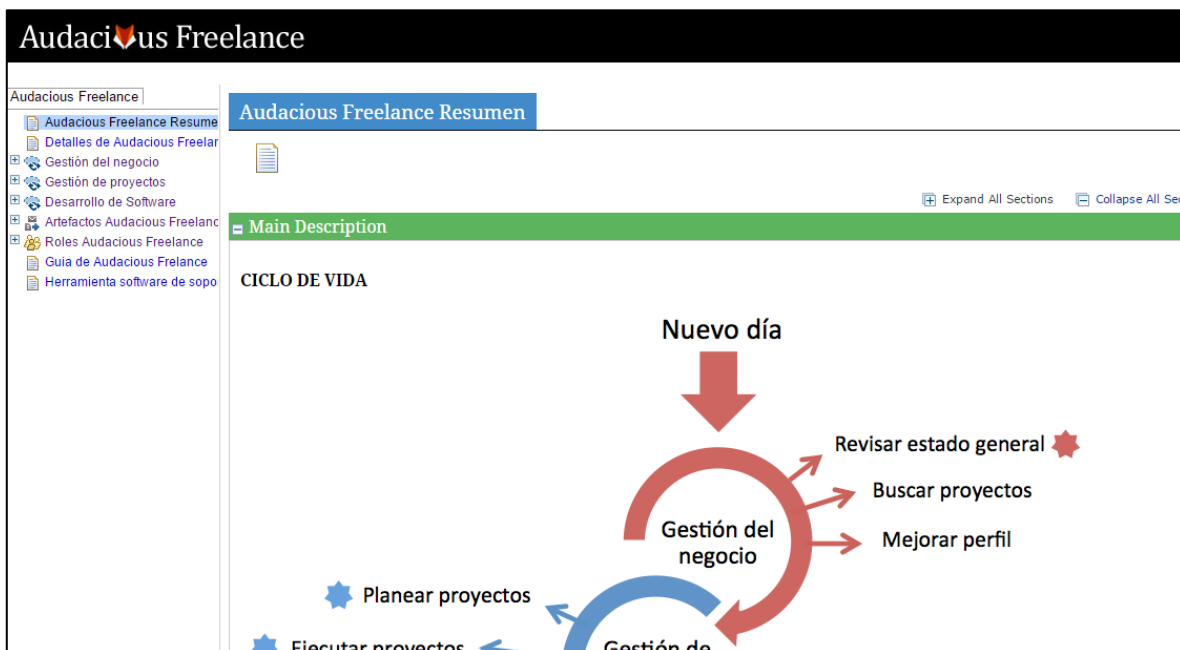


Figura 27: Instrumento de soporte de Audacious Freelance - Documentación web

5.11.2. Guía de Audacious Freelance

Como instrumento para acercarse al *Freelance* se diseñó una guía que comprendía la especificación de *Audacious Freelance*, en versión resumida y de manera más básica. A este documento se le trabajó fuertemente en la parte estética, pensando que de esa manera, se podría conseguir que la persona le tomara más interés. Esa guía se puede ver en el Anexo D.

5.11.3. Audacious Freelance Dashboard - AFD

Este instrumento es el más importante de todos. Es un software Web prototipo de gestión de *Audacious Freelance*. Permite gestionar los tres artefactos principales (*Lista de tareas*, *Lista de conocimiento* y *Lista de Investigación*) con el fin de automatizar los reportes que estos puedan generarse de acuerdo a la información suministrada. AFD, es un software que se desarrolló para ser lo simple de usar y para que el *Freelance* pudiera adoptar el proceso *Audacious Freelance* más fácilmente, para esto, se puso mucho empeño y dedicación en hacer la aplicación estáticamente limpia y usable. Si bien la herramienta maneja solamente los artefactos de AF, es una buena guía para que se conozca el proceso y además, siendo AF un proceso flexible, lo ejecuten a su criterio. Para más detalles ver el *Anexo F*.

Esta herramienta fue la principal al ser evaluada por los *Freelance* como se observa en el capítulo siguiente. Esto es debido a que es la simplificación del proceso a solo tres artefactos, que prometen resolver la pregunta de investigación de este proyecto. El código fuente de esta herramienta se encuentra en el *Anexo E*.

Las características de la AFD son:

- Gestión del artefacto *Lista de tareas* por cada proyecto existente. Esta gestión permite conocer información que será usada para mostrarle al *Freelance* en cuanto a su desempeño y productividad. La herramienta también permite tomar el tiempo que requiere el desarrollo de una tarea.

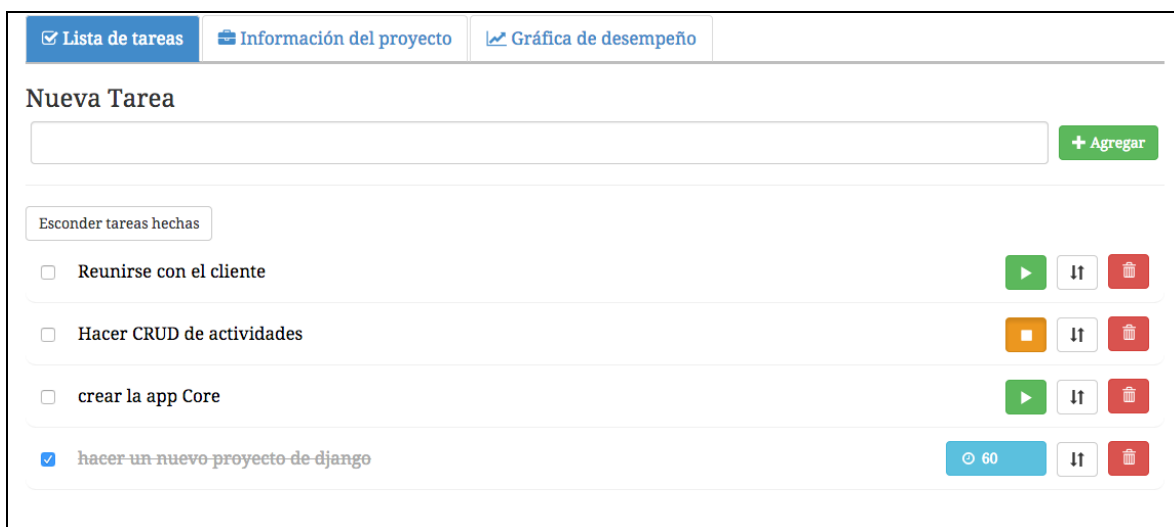


Figura 28: Audacious Freelance Dashboard - Gestión Lista de tareas

- Gestión del artefacto *Lista de investigación*. Con esta gestión se pretende motivar al *Freelance* a que constantemente este aprendiendo cosas nuevas mejorando así su perfil y volviéndolo más competitivo.



Figura 29: Audacious Freelance Dashboard - Gestión Lista de investigación

- Gestión del artefacto *Lista de conocimiento*. Con este artefacto se pretende mejorar la productividad del *Freelance* al poder agrupar conocimiento, para poderlo usar cuando se requiera.

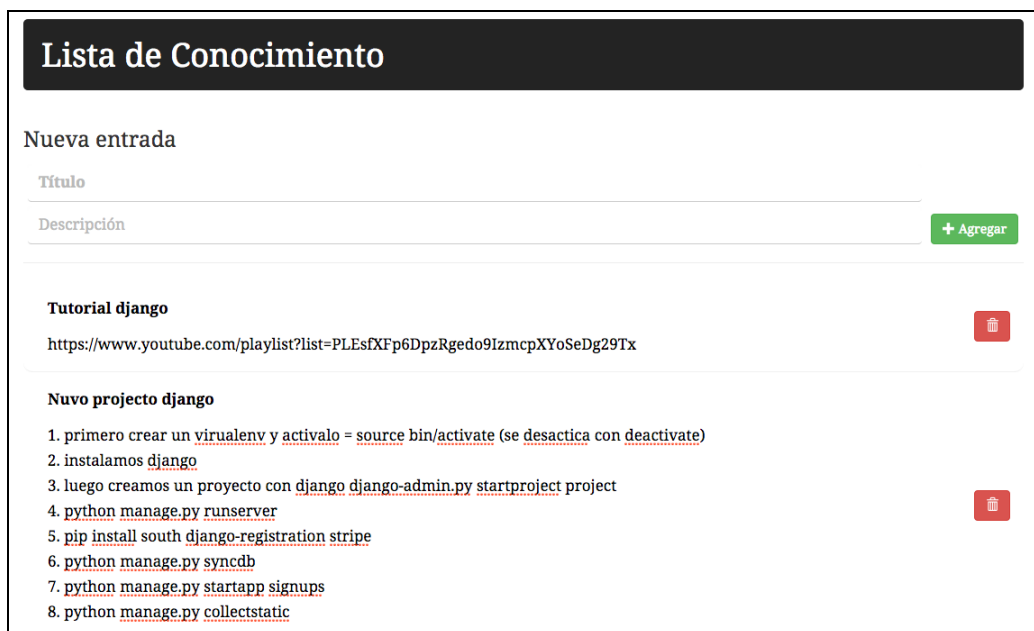


Figura 30: Audacious Freelance Dashboard - Gestión Lista de conocimiento

- Compartir en redes sociales sobre nuevos conocimientos aprendidos y proyectos terminados con el fin de promover el portafolio y hacer algo de mercadeo.



Figura 31: Audacious Freelance Dashboard - Compartir en redes sociales

- Reportar el estado actual de los proyectos para que el *Freelance* tome conciencia del estado de su trabajo y así pueda, bajo su criterio, planear y dar prioridad. La herramienta hace *proyecciones* de cómo ha sido el trabajo realizado en un proyecto particular y con esta información le informa al *Freelance* si debe incrementar su esfuerzo o si el proyecto ha superado el presupuesto. Estas *proyecciones* fueron calculadas de manera básica para mostrar el aporte que daría esta herramienta prototipo. Se trata del ponderado de las actividades que se han hecho diariamente y se comparan con el ponderado de actividades diarias que faltan por hacer y así se determina cuanto si el esfuerzo que se ha llevado realizando alcanza para terminar el proyecto, de lo contrario, se estima cuando esfuerzo de más se necesitaría para acabarlo.

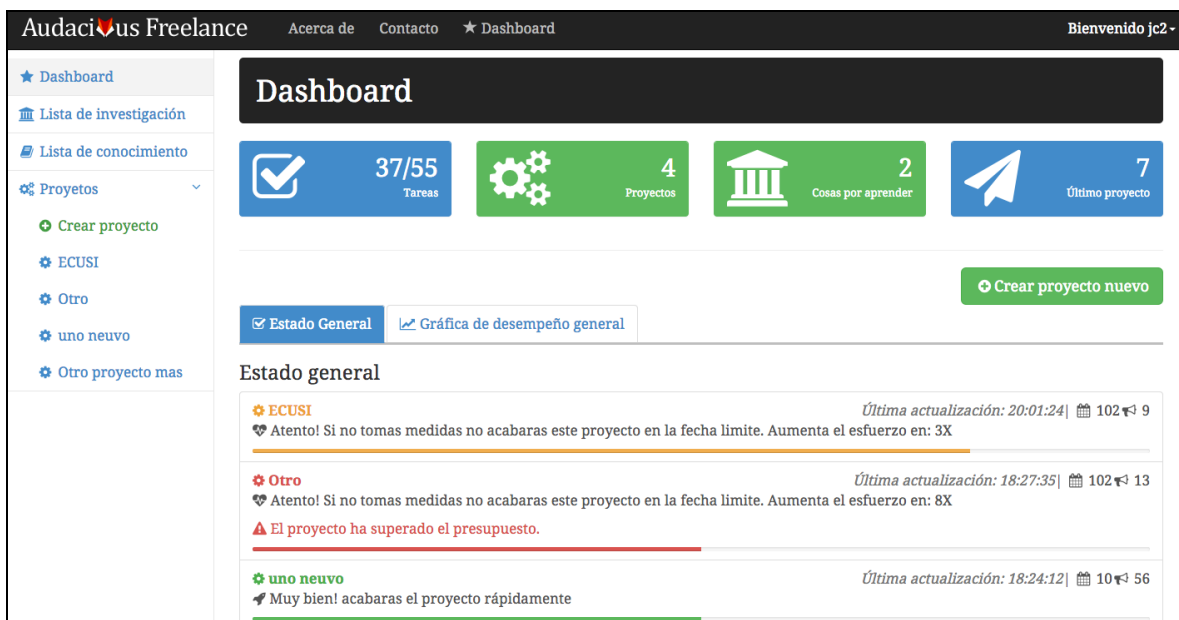


Figura 32: Audacious Freelance Dashboard - Inicio



Figura 33: Audacious Freelance Dashboard - Reportes de proyecto

- Reportes gráficos de desempeño para realizar una reflexión sobre el trabajo realizado.

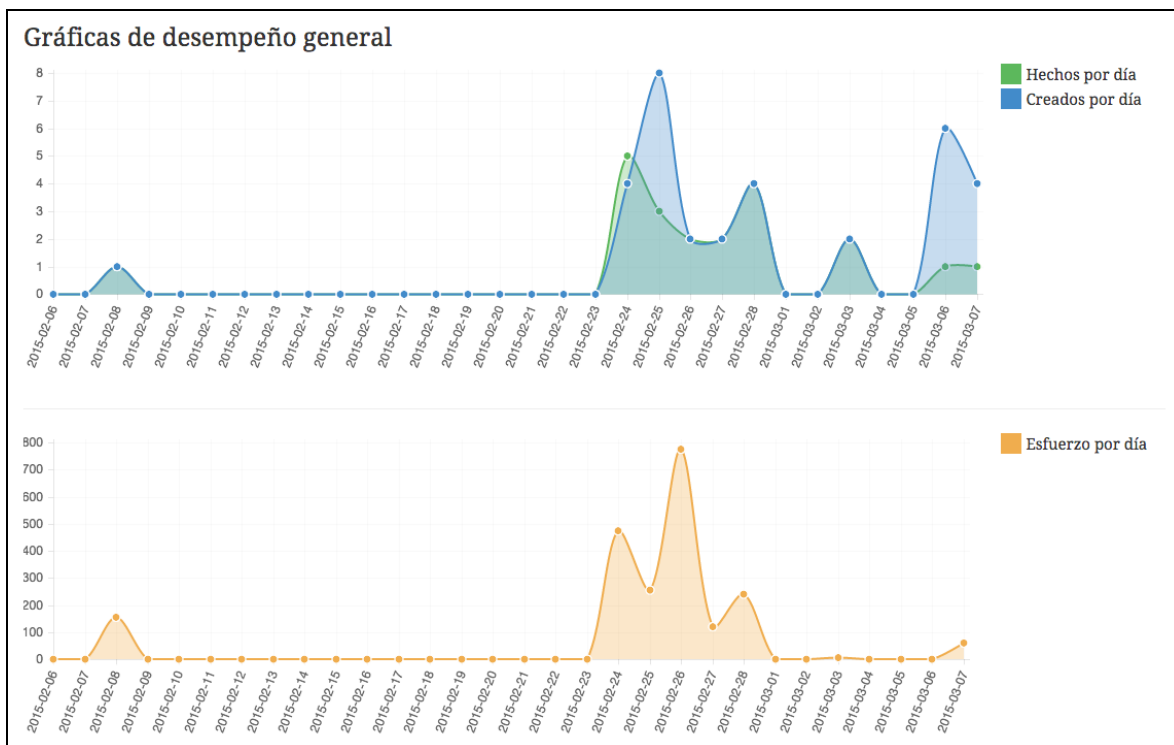


Figura 34: Audacious Freelance Dashboard - Reporte grafico de desempeño

CAPÍTULO 6. EVALUACIÓN DEL PROCESO *AUDACIOUS FREELANCE*

6.1. Introducción

Para dar cumplimiento al último objetivo específico, se realizó un estudio de caso con el fin de evaluar el proceso *Audacious Freelance* diseñado para cubrir las necesidades de un *Freelance* desarrollador de software contrastando la ejecución de proyectos *ad-hoc* con la del nuevo proceso para determinar si de alguna manera *Audacious Freelance* eleva la calidad, productividad y competitividad del *Freelance*.

Runeson [19] afirma que estudio de caso es una metodología de investigación adecuada para la ingeniería de software debido a que estudia un fenómeno en su contexto natural y que sería difícil estudiarlo separadamente. Los estudios de caso no siempre general el mismo resultado pero proveen un mejor entendimiento del fenómeno.

Para la ejecución de esta actividad, se usó la plantilla proporcionada por Brereton [39] para casos de estudio en ingeniería de software. Esta plantilla tiene once secciones, sin embargo para este estudio y por cuestiones de redundancia de información se realizaron las siete primeras. Además, se agregó una ficha del estudio de caso a manera de resumen.

6.2. Ficha del estudio de caso

La siguiente tabla muestra un resumen de varios aspectos del estudio de caso.

Tabla 5: Ficha del estudio de caso

Objetivo del Caso	Determinar si Audacious Freelance incrementa la calidad, productividad y competitividad del Freelance
Objeto del caso	El proceso Audacious Freelance
Sujetos de estudio	Freelance desarrolladores de software
Unidad de análisis	Proyecto de desarrollo de software
Teoría	Audacious Freelance
Tipo de caso según el objetivo	Exploratorio
Tipo de análisis según las unidades de análisis	Embebido
Tipo de información	Cualitativo
Proceso	Fijo
Método de colección de Información	Primer grado
Propósito de selección de caso	típico

Pregunta de investigación Principal	Indicadores	Mediciones	Instrumentos	Aplicado a
¿Qué tan eficaz es Audacious Freelance para mejorar el trabajo del Freelance en cuanto a calidad, productividad y competitividad?	Eficacia = (OCa + OPr + OCo + OAd) / 4	OCa - Contraste de opinión entre proyecto ad-hoc y AF sobre calidad	Entrevista	Freelance
		OPr - Contraste de opinión entre proyecto ad-hoc y AF sobre productividad		
		OCo - Opinión sobre la mejora en competitividad que brinda AF al trabajo del Freelance		
		OAd - Opinión sobre la facilidad de adopción de AF		
Preguntas de investigación secundarias	Indicadores	Mediciones	Instrumentos	Aplicado a
¿Qué tan apropiado es AF para mejorar la calidad del desarrollo de software como Freelance?	OCa - Contraste de opinión entre proyecto ad-hoc y AF	Percepción sobre calidad en uso	Encuesta	Freelance
		Percepción sobre calidad externa		
		Percepción sobre calidad interna		
¿Qué tan apropiado es AF para mejorar la productividad del desarrollo de software como Freelance?	OPr - Contraste de opinión entre proyecto ad-hoc y AF	Conciencia del estado del proyecto	Entrevista	Freelance
		Reflexión sobre el trabajo realizado		
¿Qué tan apropiado es AF para mejorar la competitividad del desarrollo de software como Freelance?	OCo - Opinión sobre la mejora en competitividad que brinda AF al trabajo del Freelance	Aumento de la relevancia hacia la calidad	Entrevista	Freelance
		Aumento de la relevancia hacia la productividad		
		Apoyo en el mejoramiento del perfil como profesional Freelance y su autopromoción		
¿Qué tan fácil es adoptar Audacious Freelance?	OAd - Opinión sobre la facilidad de adopción de AF	Satisfacción del uso de la herramienta Web generada por EPF sobre la especificación de AF	Entrevista	Freelance
		Satisfacción del uso de la herramienta AF Dashboard		
		Satisfacción del uso de la guía de socialización de AF		
		Facilidad de entendimiento del proceso	Entrevista	
		Comprensión de conceptos		

6.3. Contexto

En este proyecto se ha buscado diseñar un proceso de desarrollo de software integro que responda a las necesidades del *Freelance*. En primer lugar se realizó una revisión en la literatura científica para poder establecer los trabajos relacionados. No se encontró ningún trabajo relacionado a esta búsqueda particular, sin embargo, se encontraron trabajos sobre procesos de desarrollo de software aplicados para una sola persona, en su mayoría adaptaciones de métodos ágiles. En segundo lugar, se estudió el modelo de trabajo *Freelance*, pudiendo así, establecer dinámicas, características y necesidades del *Freelance*. Por último, y contando con la información recolectada y analizada, se establecieron los criterios que debía tener un proceso de desarrollo de software para un *Freelance*. Se tomó *Scrum* y *XP* como referentes, los cuales aportaron al proceso. Este nuevo proceso se denominó *Audacious Freelance* (AF) y tiene como objetivo elevar la calidad la calidad, productividad y competitividad del *Freelance*. Para poder determinar si AF satisface el objetivo, se decidió realizar este estudio de caso, el cual consiste en evaluar y contrastarlo con respecto a un desarrollo de proyecto software *ad-hoc*.

6.3.1. Pregunta de Investigación

La pregunta de investigación de este caso de estudio, está estrechamente relacionada con la pregunta de investigación del proyecto y es: ¿Qué tan eficaz es *Audacious Freelance* para mejorar el trabajo del *Freelance* en cuanto a calidad, productividad y competitividad? Se espera la respuesta a esta pregunta, responda la pregunta que enmarca este proyecto de grado.

De acuerdo a esta pregunta de investigación, otras preguntas secundarias fueron formuladas, a partir de los tres aspectos que se querían medir, para que así se pueda ayudar a responder a la pregunta principal:

- ¿Qué tan apropiado es AF para mejorar la *calidad* del desarrollo de software como *Freelance*?
- ¿Qué tan apropiado es AF para mejorar la *productividad* del desarrollo de software como *Freelance*?
- ¿Qué tan apropiado es AF para mejorar la *competitividad* del desarrollo de software como *Freelance*?

Por otro lado, y con el fin de corroborar la información y determinar el éxito del proyecto se formuló la siguiente pregunta:

- ¿Qué tan fácil es adoptar AF?

A partir de estas cuatro preguntas, se diseñó el caso de estudio, se seleccionaron los indicadores, medidores e instrumentos que se utilizarían para dar respuesta a cada una de las preguntas, como se muestra en el diseño.

6.4. Diseño

Como el objetivo del estudio es determinar si *Audacious Freelance* incrementa la calidad, productividad y competitividad del *Freelance*, el estudio de caso se ha diseñado de tipo *exploratorio* debido a que AF es un proceso nuevo y único en su clase, sin antecedentes y se requiere formular conocer cómo afecta a los *Freelance* y generar ideas e hipótesis.

Para realizar el objetivo y responder las preguntas de investigación se planteó un estudio *embebido* con *dos unidades de análisis*: un proyecto de desarrollo de software *ad-hoc* y un proyecto de desarrollo de software con AF. De esta manera se podrían contrastar los dos desarrollos y poder evidenciar el verdadero aporte de AF al modelo de trabajo *Freelance*.

Al ser el objeto de estudio el proceso *Audacious Freelance*, se quería conocer la percepción de mejora que tenían *los sujetos del estudio*, es decir, los *Freelance* quienes estaban ejecutando el AF, con respecto a cómo había estado trabajando en proyectos pasados sin tener un proceso formal que los guíe, desarrollando software de manera improvisada. Esto quiere decir que la información que se quería extraer iba a ser de tipo cualitativo y por lo tanto los indicadores y las mediciones tenían que estar acordes. Para lograr esto, se tomaron los tres aspectos que se querían examinar: la *calidad*, la *productividad* y la *competitividad*. Se formularon tres preguntas de investigación, cada una correspondiente a cada aspecto.

De esta manera, para los aspectos de calidad y productividad se quiso conocer la percepción de mejora que podría tener el *Freelance*, al contrastar los dos procesos de desarrollo AF y *ad-hoc*. Para la calidad, la percepción sería medida de acuerdo a las métricas de calidad del producto software de la ISO 25010. La productividad, sería medida de acuerdo a como AF ayudaba a conocer del estado de los proyectos, el esfuerzo y tiempo consumidos en ellos y la reflexión que esta información podría generar. Para el aspecto de competitividad, se quiso conocer la opinión que tenían los *Freelance* con respecto a si a la relevancia que generaba AF de los otros dos aspectos (Calidad y productividad) y demás si el proceso permitía un mejoramiento del perfil profesional y su autopromoción.

Ahora bien, otro aspectos que se quería examinar, fue la facilidad de adopción de AF por parte de los *Freelance*. Para esto se quiso conocer la opinión que ellos tenían sobre los *instrumentos de soporte* de AF y medir la satisfacción del uso de estos instrumentos, y el entendimiento de la información que ellos mostraban.

Con estos cuatro aspectos a examinar, se planeó que en conjunto respondieran a la pregunta principal de investigación y así poder dar cumplimiento al objetivo del estudio de caso.

6.5. Selección del caso

Debido a que se quería determinar si la ejecución AF en un proyecto de software afectaba positivamente al trabajo del *Freelance*, se decidió confrontar la percepción en un desarrollo de software *ad-hoc* y contrastarlo con la percepción en un desarrollo con AF. De esta manera, nuestro objeto sería el proceso *Audacious Freelance*. Se quería conocer como éste, afectaba el trabajo del *Freelance*, quienes serían los sujetos de estudio, pues son los directamente afectados por este proyecto. De ellos se extraería información cualitativa, siendo la única fuente de información. El caso seleccionado sería *típico*, es decir, los dos proyectos tanto *ad-hoc* como el AF serían casos reales de la industria; se analizaría el trabajo real de los *Freelance*, en proyectos que ya estuvieran desarrollando, por consiguiente, solo se podría ejecutar la dimensión de *desarrollo de software* del proceso AF, sin embargo, los *instrumentos de soporte* serían presentados y sustentados a los *Freelance* para con esto, conseguir información que pudiera retroalimentar el proceso.

El proyecto *ad-hoc* sería un proyecto que ya hubieran desarrollado, o que estuviera en desarrollo. El proyecto con AF, sería cualquier proyecto que estuvieran ejecutando en el momento, en el que, por el tiempo de una semana, seguirían la dimensión de *desarrollo de software* de AF haciendo uso de los tres *instrumentos de soporte*.

La selección de los *Freelance* se hizo de acuerdo a la disponibilidad, pues debido a su trabajo diario, no todos podrían servir como sujetos para este caso. Fue así como se escogieron cuatro *Freelance* y se intentó realizar una *triangulación*³⁵ [19] para incrementar la precisión del estudio; los cuatro sujetos escogidos tienen un perfil, un sector del mercado de software y experiencia, diferente. Las características de estos cuatro sujetos eran:

Freelance 1:

- 1 año de experiencia - Colombia
- Estudiante - Último semestre pregrado de Ingeniería de sistemas
- Desarrollador de aplicaciones web en PHP
- 3 Proyectos realizados

Freelance 2:

- 3 años de experiencia - Colombia
- Estudiante - Último semestre pregrado de Ingeniería de sistemas
- Desarrollador de aplicaciones web en PHP
- 6 proyectos realizados

³⁵ La triangulación consiste en tomar diferentes ángulos del objeto o sujeto de estudio y así proporcionar un panorama más amplio. La triangulación puede ser de información, del observador, de la metodología o de la teoría.

Freelance 3:

- 5 años de experiencia - Colombia
- Profesional - Graduado en 2009 de Ingeniería de sistemas
- Desarrollador de aplicaciones móviles para iOS
- 50+ proyectos realizados
- 1 proyecto presentado en Silicon Valley (San Francisco) en 2014

Freelance 4:

- 4 años de experiencia - Colombia
- Profesional - Graduado en 2009 de Ingeniería de sistemas
- Desarrollador de aplicaciones móviles para Android
- 20+ proyectos realizados
- 1 proyecto presentado en Silicon Valley (San Francisco) en 2014

6.6. Procedimiento

Para la ejecución de este caso de estudio, en primer lugar se preparó *los instrumentos de soporte* que serían entregados a los sujetos. Estos eran:

- Una guía básica (en físico) de instrucción a AF
- Una web de documentación del proceso AF
- Una herramienta de gestión de artefactos para AF, Audacious Freelance Dashboard – AFD.

En segundo lugar se preparó una inducción a AF que sería dada de manera verbal de 30 minutos de duración. También se preparó una encuesta sobre métricas de calidad de producto software según la ISO 25010 con 13 preguntas y una entrevista que sería realizada a cada sujeto al finalizar la semana de ejecución de AF por parte de los *Freelance*. La entrevista tenía una duración prevista de 1 hora. Durante la semana de ejecución de AF, no habría intervención de ningún tipo por parte de los investigadores, es decir, el procedimiento del estudio era *fijo*³⁶.

6.7. Recopilación de datos

Como se dijo anteriormente, la información que se quiere recolectar es cuantitativa y se trata de la percepción y la opinión que tienen los *Freelance* con respecto al contraste que encuentran entre el desarrollo de software guiado por un proceso *ad-hoc* y el proceso *Audacious Freelance*. Se decidió por una información de tipo cualitativo en primer lugar debido a la facilidad de recolección de información (comparado con una cuantitativa),

³⁶ Se dice que el procedimiento de un estudio de caso es "fijo" cuando todos los parámetros del estudio son definidos en el comienzo sin modificaciones en el transcurso. Sería "flexible" si los parámetros cambiaron de acuerdo al curso que tome el estudio.

pues el caso de estudio involucraría diferentes instrumentos que serían entregados al *Freelance* y que debían ser analizados y los alcances de este proyecto limitaban agregar esfuerzos a la preparación de dichos instrumentos (por ejemplo, tomar mediciones desde el AFD). En segundo lugar debido a la naturaleza subjetiva de la información que se quería determinar, pues se trataba de la percepción y opinión de una persona. Y por último, debido al poco tiempo de dedicación que los sujetos del caso podrían asignarle al estudio de caso, pues son personas que trabajan profesionalmente y dependen de ello.

La información que será recolectada será de primer grado, es decir, se usaran instrumentos como encuestas y entrevistas para adquirir la información de los sujetos de estudio, que son los cuatro *Freelance* escogidos. La recolección se haría solo al terminar el estudio de caso que tendría una duración de una semana, y se guardaría en notas de audio para, luego, ser analizadas e interpretadas.

6.7.1. Indicadores y mediciones

Para evaluar de manera objetiva y particularmente, para dar respuesta a las preguntas de investigación, se definió un conjunto de indicadores y mediciones. Para definir esto, se tomó cada una de las preguntas que se habían formulado previamente y que están directamente relacionada con los cuatro aspectos a examinar (calidad, productividad, competitividad, facilidad de adopción de AF) y se establecieron, indicadores, seguimiento de las mediciones e instrumentos de recolección de información.

A continuación se muestran las preguntas de investigación junto a sus mediciones, indicadores e instrumentos.

Pregunta de investigación principal:

¿Qué tan eficaz es *Audacious Freelance* para mejorar el trabajo del *Freelance* en cuanto a calidad, productividad y competitividad?

Se estableció que el indicador de la pregunta principal de investigación del estudio de caso sería el promedio de las opiniones de los *Freelance*, que a la vez, son indicadores de las preguntas secundarias de investigación. Se realiza un ponderado cuantitativo, considerando que las opiniones pueden ser positivas o negativas.

Preguntas secundarias:

- **¿Qué tan apropiado es AF para mejorar la *calidad* del desarrollo de software como *Freelance*?** Para esta pregunta, el indicador que se tomó se estableció fue el contraste de opinión sobre el aspecto de calidad en un desarrollo *ad-hoc* y uno con AF. Las mediciones fueron escogidas de acuerdo a las categorías de métricas de calidad del producto software de la ISO 25010. El instrumento para recolección de información sería la encuesta. Esto debido a que la calidad es un tema fuertemente estudiado y hay varias definiciones. Usar una de éstas, elimina un poco la subjetividad y ayuda a la recolección de información. Sin embargo, no se pierde el carácter cuantitativo del caso de estudio, pues se quería recolectar la

opinión de los sujetos, básicamente, si pensaban que el uso de AF mejoraba, empeoraba o era irrelevante en cuanto a cada métrica de calidad.

- **¿Qué tan apropiado es AF para mejorar la *productividad* del desarrollo de software como *Freelance*?** Al igual que en la pregunta anterior, el indicador sería también el contraste de opinión. Las mediciones tenían que ver con abstraer el concepto de productividad más allá de que la productividad es producir más en menos tiempos; no se quería ser tan directo. De esta manera, se preguntó sobre la conciencia del estado de los proyectos y la reflexión del trabajo hecho, con el fin de que el determinar si AF ayudaba a que el Freelance se cuestionara sobre su desempeño y por ende la su productividad. El instrumento usado fue la entrevista debido a que se quería conocer su opinión sin un formato estructurado que permitiera que el sujeto fuera espontáneo.
- **¿Qué tan apropiado es AF para mejorar la *competitividad* del desarrollo de software como *Freelance*?** El indicador establecido para esta pregunta, fue también el contraste de opinión. Para las mediciones, se definió el concepto de competitividad a partir de la calidad y la productividad, y se quería saber si había ocurrido un aumento en la importancia hacia estos dos aspectos. Otros indicadores como el capital humano (entendido como el mejoramiento del perfil profesional), y la autopromoción de su trabajo fueron establecidos. El instrumento de recolección sería la encuesta, por los motivos antes planteados.
- **¿Qué tan fácil es adoptar AF?** Para esta pregunta, el indicador fue también la opinión de los *Freelance* sobre la facilidad de adopción de AF. Tres de las mediciones que se establecieron tenían que ver, cada una, con la satisfacción del uso los tres *instrumentos de soporte* de AF. Otras mediciones tenían que ver con la comprensión de conceptos, coherencia y organización de los *instrumentos de soporte*. De igual manera que las anteriores preguntas, se usó la entrevista como instrumento de recolección.

6.8. Resultados y Análisis

Los datos recolectados fueron recopilados y posteriormente analizados. Estos datos eran cuantitativos, por lo tanto, en esta sección, se presentara el análisis de cada medición según la pregunta de investigación, respondiente al mismo tiempo a dicha pregunta.

Los primeros datos a analizar, serán los de que tienen que ver con primera pregunta secundaria de investigación que tiene que ver con los aspectos de calidad. Primero porque al tratarse de una encuesta los datos son más sencillos de analizar y segundo porque las preguntas secundarias serán las primera en ser abordadas pues son las que dan respuesta a la pregunta principal. Finalmente, luego de analizar las preguntas secundarias, se dará respuesta a la pregunta principal.

6.8.1. Aspecto Calidad

Para este aspecto se realizó una encuesta sobre la percepción del *Freelance* con respecto a 13 métricas de calidad del producto software, comparando cada métrica con un desarrollo *ad-hoc* y con AF. Las respuestas se ven en la Figura 35. Había tres posibles respuestas: *mejora*, *es igual / no nota el cambio*, *empeora*. Como en ninguna métrica, ninguno de los cuatro *Freelance* se escogió la opción que *empeora*, esta opción se eliminó de la gráfica.

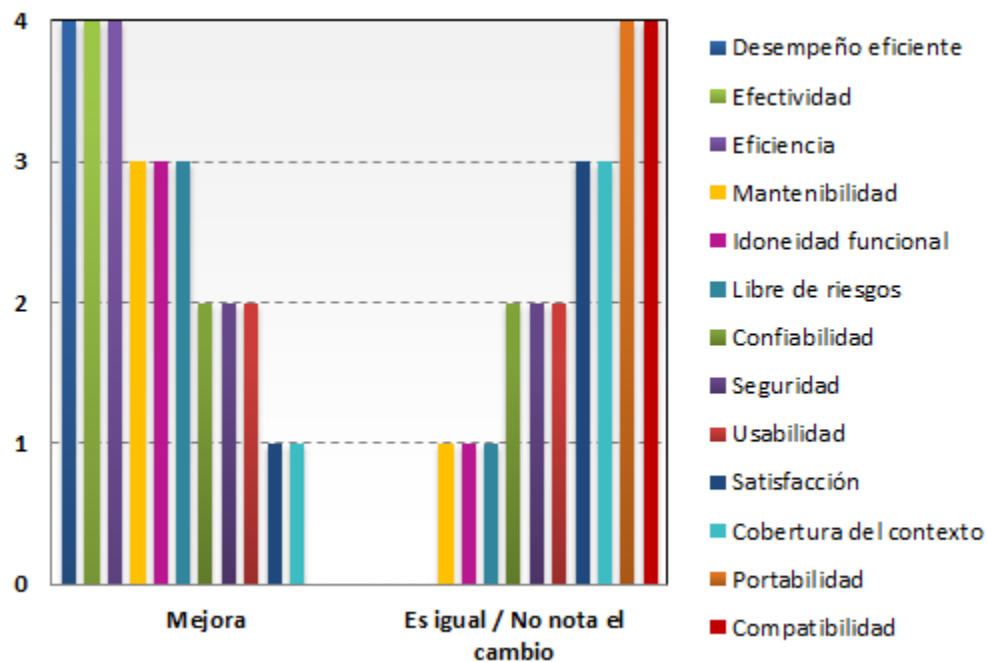


Figura 35: ¿Qué métricas de calidad cree usted que Audacious Freelance mejora?

Análisis: Por unanimidad, las tres métricas que se consideran que se mejoran con AF, son el *desempeño eficiente*, la *efectividad* y la *eficacia*. Se pueden tomar las otras tres métricas que recibieron tres de cuatro votos como positivas también: *mantenibilidad*, *idoneidad funcional* y *libre de riesgos*. Las otras no se tendrán en cuenta y se consideraran como que no aportan al estudio.

La percepción en mejora en el *desempeño eficiente* y *mantenibilidad* se explican con la actividad influenciada por la práctica de XP llamada *refactoring* que, para AF, es una actividad llamada *reingeniería*. Esta práctica mejora los el diseño del código fuente, por ende la calidad interna del producto.

En cuanto a la *efectividad* e *idoneidad funcional*, se pueden explicar por la manera en que AF es dirigido por tareas, es decir, las tareas son siempre anotadas y tenidas en cuenta en el artefacto *Lista de tareas*, gestionado por la herramienta AFD.

La *eficiencia* y la métrica *libre de riesgos*, se podría explicar desde el punto de vista de la productividad, pues por medio de la herramienta AFD, los *Freelance* conocen el estado de sus proyectos en cuanto al presupuesto y esfuerzo invertido, de esta manera el desarrollador tiene una mejor conciencia de que está sucediendo y toma acciones correctivas más rápidamente.

Pregunta: ¿Qué tan apropiado es AF para mejorar la *calidad* del desarrollo de software como *Freelance*?

Respuesta: Si bien 6 de las 13 métricas son positivas, las otras 7 se consideran neutras (porque no hubo percepción de empeoramiento), y de esta manera se puede decir que AF mejora la calidad del producto software tanto interna, externa como en uso.

6.8.2. Aspecto productividad

Para este aspecto, el instrumento de recolección de información usado fue la entrevista, esta entrevista se hizo en conjunto con las demás entrevistadas de una hora de duración y luego de la semana de ejecución de AF. De acuerdo a las mediciones, se expondrá los comentarios unificados que dijeron los cuatro *Freelance* al respecto.

Mediciones:

- **Conciencia del estado del proyecto:** Por unanimidad, los *Freelance* aseguraron encontrar mejoras en cuanto a la conciencia que se genera del proyecto puesto que la herramienta AFD (que es la que proporciona reportes automatizados) mostraba y hacía relevancia al esfuerzo y presupuesto consumido de cada proyecto, y permitía de esta manera planear y priorizar el trabajo diario. Mencionaron además, que en sus proyectos *ad-hoc* no toman en cuenta el seguimiento al esfuerzo y dinero y que muchas veces los clientes alargan los proyectos, convirtiéndolos en un fracaso financiero para los *Freelance*.
- **Reflexión sobre el trabajo realizado:** tres de los cuatro entrevistados afirmó que encontraba una mejora en este aspecto al usar AF, debido a que por la herramienta AFD proporcionaba mensajes de alerta con respecto al estado y las proyecciones del proyecto. El *Freelance* que no estuvo de acuerdo, mencionó que a AFD le faltaban mejores algoritmos de proyección que fueran más certeros.

Análisis: La herramienta AFD se creó con la intención de proporcionar información relevante para el *Freelance*, automatizando la generación de reportes de estado y permitiendo conocer proyecciones de los proyectos para así, establecer planes a seguir. Al tratarse de un prototipo, se usó un algoritmo muy simple de proyección con el fin de mostrar la idea de la funcionalidad. Sin embargo, de acuerdo a las opiniones mencionadas, se puede establecer que hay una opinión generalizada de mejora en cuanto a productividad ejecutando AF.

Pregunta: ¿Qué tan apropiado es AF para mejorar la *productividad* del desarrollo de software como *Freelance*?

Respuesta: AF es apropiado para mejorar la productividad. Los sujetos de estudio mencionan que la herramienta AFD es la clave para eso.

6.8.3. Aspecto Competitividad

De igual manera que en la productividad, en este aspecto se hizo uso de la entrevista como medio de recolección de información. La competitividad se analiza de acuerdo a la calidad y la productividad, el mejoramiento del perfil profesional y la autopromoción.

Mediciones:

- **Aumento de la relevancia hacia la calidad:** Debido a la seguimiento de prácticas que propician la calidad del producto en un proceso establecido, dos de cuatro *Freelance* consideran que su hay relevancia a la calidad. Sin embargo, AFD, que fue el instrumento del que más hablaron, no ayuda explícitamente a este aumento en la relevancia de calidad.
- **Aumento de la relevancia hacia la productividad:** Por unanimidad, y debido al uso de AFD (que se enfoca sobre todo en la productividad) mencionaron que sí aumenta la relevancia con respecto a la productividad, debido a los reportes automatizados que generan *conciencia* de cómo se están haciendo las cosas y sobre todo pusieron relevancia en la forma como esa información es mostrada, pues llama mucho la atención.
- **Apoyo en el mejoramiento del perfil como profesional Freelance y su autopromoción:** Si bien un *Freelance* considero que AF apoya en estos dos aspectos, la opinión generalizada (y de nuevo enfocándose en AFD) es. que a la herramienta aún le faltan trabajo en este punto, aunque indicaron que va por buen camino con funcionalidades como la de compartir conocimientos adquiridos y proyectos finalizados por las redes sociales.

Análisis:

Debido a que los sujetos de estudio, pusieron mucho más interés en la herramienta software AFD, se cree que esto afecto las mediciones en este aspecto. Sin embargo, se considera que un refinamiento en la herramienta, podría mejorar significativamente las opiniones al respecto.

Pregunta: ¿Qué tan apropiado es AF para mejorar la *competitividad* del desarrollo de software como *Freelance*?

Respuesta: Es medianamente apropiado. Se podría mejorar la respuesta a esta pregunta mejorando las funcionalidades de la herramienta AFD.

6.8.4. Aspecto facilidad de adopción de AF

En este aspecto también se realizó una entrevista. Las mediciones hacen referencia a los *instrumentos de soporte* de AF.

Mediciones:

- **Satisfacción del uso de la herramienta Web generada por EPF sobre la especificación de AF:** Debido a que los cuatro *Freelance* desconocían SPEM, esta herramienta no aportó mucho al comienzo, sin embargo, luego de una breve explicación pudieron mostrar algo de satisfacción, sobre todo en cuanto al diagrama pues mencionaron que así podrían reconocer la esencia del proceso y como éste modelaba correctamente el trabajo como *Freelance*.
- **Satisfacción del uso de la guía de socialización de AF:** Mencionaron que era una buena manera de intentar buscar el reconocimiento y la aceptación del proceso pues mostraba de una manera interesante el contenido teórico que es poco llamativo.
- **Satisfacción del uso de la herramienta AF Dashboard:** Si bien la herramienta prototipo aún necesita de trabajo, los cuatro *Freelance* se mostraron muy atraídos por las funcionalidades de la herramienta y como podrían aportar a su trabajo. La facilidad de uso y simpleza fue un aspecto reconocido.
- **Facilidad de entendimiento del proceso:** Por unanimidad, los entrevistados mostraron que el proceso era fácilmente entendible pues comprendía de unas cuantas prácticas muy concisas y que modelaban correctamente el trabajo como *Freelance*.
- **Comprensión de conceptos:** El único concepto que en un principio desconocieron los cuatro entrevistados era SPEM y el diagrama generado por EPF.

Análisis:

Como se mencionó anteriormente, los *Freelance* pusieron mucho interés en la herramienta, y dejaron muchas veces de lado los otros dos instrumentos. Esto no es de extrañarse, al tener perfiles tan prácticos, les llama la atención cualquier instrumento que puedan ayudar a su trabajo automatizando procesos y desechan cualquier cosa que podría consumirles tiempo. Esto es un reto debido a que, si estuvieran en una empresa su jefe podría imponerles un proceso, con todas las actividades que esto conlleva (como, documentarse, seguirlo a pie de la letra, usar todos sus elementos, etc.). Sin embargo, también muestra una oportunidad, que es el instruirlos por medio de una herramienta como AFD que haga completamente transparente el proceso. Desafortunadamente, por alcances de este proyecto, solo se pudo desarrollar un prototipo que tenía en cuenta pocos elementos de AF.

Pregunta: ¿Qué tan fácil es adoptar AF?

Respuesta: Si bien, los *Freelance* se mostraron reacios a usar algunos de los instrumentos, ninguno comentó que el proceso fuera complejo. Hay aspectos que se pueden mejorar, pero se puede responder esta pregunta diciendo que AF es medianamente fácil de adoptar.

6.8.5. Respuesta a la pregunta de investigación

La pregunta de investigación de este estudio de caso fue: **¿Qué tan eficaz es Audacious Freelance para mejorar el trabajo del Freelance en cuanto a calidad, productividad y competitividad?** El indicador para dar solución a esta pregunta fue el promedio de las opiniones de las preguntas secundarias. Aunque es un estudio cuantitativo, es conveniente para este análisis otorgar una puntuación a cada pregunta del -1 a 1 en donde -1 significa que empeora completamente, 0 que no aporta nada y 1 que mejora completamente, podríamos entonces calificar así:

- **Contraste de opinión entre proyecto ad-hoc y AF sobre calidad:** Se calificaría como 0.5, considerando que muchas métricas de calidad no fueron mejoradas; se piensa que es difícil abarcarlas todas para un proceso de una sola persona.
- **Contraste de opinión entre proyecto ad-hoc y AF sobre productividad:** Se calificaría como 0.9, puesto que la herramienta ADF ayuda mucho en la generación de *conciencia* del estado del proyecto y con esto, al aumento de productividad.
- **Opinión sobre la mejora en competitividad que brinda AF al trabajo del Freelance:** Se calificaría como 0.6, puesto que los *Freelance* se mostraron muy interesados por AFD, el cual, no incorporaba los elementos de *Audacious Freelance* en su totalidad y a consecuencia, se vio afectada este indicador.
- **Opinión sobre la facilidad de adopción de AF:** Esta opinión se calificaría con 0.5, debido al accionar de los *Freelance* quienes son personas muy prácticas, es necesario mejorar AFD para que el proceso sea más transparente para ellos .

Teniendo estas puntuaciones se podría decir que la eficacia de AF es: 0.625. De esta manera se podría decir que AF es satisfactoriamente eficaz para mejorar el trabajo del *Freelance* con respecto a la calidad, productividad y competitividad.

6.8.6. Sugerencias, ideas y problemas

En el flujo natural de la entrevista, los sujetos de estudio mencionaron sugerencias, ideas o problemas con es interesante mencionar. A continuación se listan:

- El Freelance 4 mencionó que el proceso AF representa correctamente el modelo del trabajo *Freelance*.
- El Freelance 3 sugirió que para que la herramienta AFD tuviera más sentido y más aceptación, ésta debería generar documentos importantes para el trabajo diario, como propuestas para clientes o cuentas de cobro.
- El Freelance 4 sugirió que la herramienta AFD debería motivar aún más al mejoramiento del perfil profesional por medio de información al respecto de las tecnologías más usadas y demandadas.
- El Freelance 3 mencionó que al proceso le hace falta actividades de pruebas automatizadas como *unittest*.
- El Freelance 2 sugirió que en cuanto a la lista de conocimiento se debería poder subir archivos para así poder guardar módulos previamente codificados.

- Los Freelance 2 y 3 piensan que la herramienta está muy simple en cuanto al manejo de tareas y piensa que podrían tener más características, como marcar el fin de una tarea o asociarla a un requerimiento.
- Por otro lado, el Freelance 1 piensa que la simpleza es uno de los mejores atributos de la herramienta con respecto a otras alternativas del mercado.
- Hubo un problema al intentar contextualizar a los cuatro *Freelance* desconocía el estándar SPEM.
- La idea de mejorar AFD para que cuente con más elementos de AFD, haciendo el proceso más transparente para quien lo ejecute.

6.9. Amenazas de validez

La validez de este estudio de caso se ve amenazada básicamente por el alcance del proyecto, a continuación se menciona los factores que se cree que pueden afectar.

- Debido a disponibilidad de los *Freelance*, el proceso AF solo pudo ser evaluado en cuanto a la dimensión de *desarrollo de software*. Se requiere de más tiempo de ejecución de la evaluación para poder tener retroalimentación de las otras dimensiones. No obstante, debido al uso de AFD y su gestión de los tres artefactos de AF, se evaluaron aspectos de otras dimensiones, actividades de *gestión de proyectos* con la *planeación de proyectos* y *ejecución de proyectos* y actividades de *gestión del negocio* como *mejoramiento del perfil*.
- La cantidad de sujetos de estudio es aún insuficiente y no es nada representativo. Sin embargo, esto se intentó mitigar haciendo una *triangulación* de sujetos, encontrando los perfiles con diferentes sectores del mercado (móvil y web), con diferentes años de experiencia y número de proyectos realizados.
- Al tener un tipo de información cualitativa, los resultados del estudio pueden variar si se ejecuta nuevamente. Este factor es difícil de controlar, pues los *Freelance* son un grupo heterogéneo de personas con sus particulares maneras de hacer las cosas, sin embargo, el proceso se diseñó lo más flexible posible para poder ser adoptado por *Freelance* con múltiples perfiles. Además la herramienta AFD se simplificó para no añadir funcionalidades de más, y para que el *Freelance* a su criterio, le pueda encontrar potencialidades.

CAPÍTULO 7. CONCLUSIONES, LECCIONES APRENDIDAS, LIMITACIONES Y TRABAJOS FUTUROS

El modelo de trabajo *Freelance* no es muy nuevo. Tiene sus orígenes a mediados de los noventa como producto de las tendencias del *outsourcing* y el *offshoring* de tecnologías informáticas. Estas tecnologías han sido potenciadas en los últimos años por el avance de la tecnología, permitiendo usar diferentes tipos de dispositivos con diferentes tipos de software, causando así un enorme crecimiento de aplicaciones, que han hecho de la labor *Freelance* una buena manera de llegar al mercado laboral y competir abiertamente con proyectos de alta calidad. Desafortunadamente, este modelo de negocio no ha tenido la atención formal que se merece por parte de la investigación en la ingeniería de software y no ha sido estudiado profundamente, hasta ahora y hasta donde se sabe.

En este proyecto se ha buscado diseñar un proceso formal de desarrollo de software íntegro que responda a las necesidades, dinámicas y características del *Freelance*. Es por eso que se realizó una encuesta que permitiera identificar los aspectos que enmarcan el modelo de trabajo *Freelance*, y junto con otros estudios similares, se pudo caracterizar este modelo que se desconocían hasta el momento, aportando así, un conocimiento formal a la academia y a la industria. Posteriormente, y con la información recolectada, se realizó un diseño de un proceso que tomó como referente dos métodos ágiles *Scrum*, y *XP*. Este nuevo proceso, denominado *Audacious Freelance*, se diseñó para ser poco intervencionista, fácil de aprender y que pudiera conducir al *Freelance* en una mejora continua, con lineamientos que le permitiera aumentar la calidad del producto desarrollado, su productividad y por ende, su competitividad. *Audacious Freelance*, además de contar con una especificación formal en la herramienta EPF y el estándar SPEM, también contó con *instrumentos de soporte* que buscaban la rápida y fácil aceptación y con esto la adopción del proceso por parte de los *Freelance*. Estos instrumentos eran tres: primero una web con la documentación del proceso, el segundo, una guía llamativa para acercar al *Freelance* y conseguir su aceptación y además una herramienta que permite gestionar los artefactos del proceso y con la información suministrada permite generar reportes sobre el desempeño realizado. Finalmente, y para corroborar el trabajo hecho, se realizó una evolución planteada en un estudio de caso donde cuatro *Freelance*, activos y destacados de la industria, probaron el proceso, junto con los *instrumentos de soporte*, durante una semana permitiendo así, conocer la eficacia de *Audacious Freelance* para mejorar aspectos de calidad del producto, productividad y competitividad. Los resultados concluyeron en que el proceso *Audacious Freelance*, mejoraba significativamente estos factores, y de esta manera, el proceso *Audacious Freelance* se responde la pregunta de investigación planteada en este proyecto.

A continuación se presentan conclusiones, limitaciones y trabajos futuros que el desarrollo de este proyecto aportó.

7.1. Conclusiones

- Debido a la poca atención de la industria al modelo de trabajo *Freelance*, no se encontraron mayores aportes en la literatura científica, haciendo de este proyecto, un estudio preliminar y pionero en el área.
- El aspecto que se consideró crítico fue caracterizar al *Freelance* adecuadamente, pues de lo contrario, si la caracterización no fuera certera, se estaría desarrollando un proceso que no cubriría realmente sus necesidades. Fue difícil hacerlo y se tuvo que omitir mucha información debido a que los sujetos son heterogéneos, sin embargo, se pudo deducir aspectos comunes lo que llevo a lo que consideramos una caracterización adecuada.
- El aspecto que se consideró más complejo a la hora de realizar un proceso era que este tuviera coherencia con todos sus activos y elementos. Además, poder sintetizar la información para diseñar un proceso útil, sencillo y poco complejo, es un reto de ingeniería de procesos.

7.2. Lecciones aprendidas

- La revisión sistemática de Biolchini [21] aporta un protocolo que permite hacer una buena búsqueda formal de referencias en la literatura científica. Al ser simple y conciso, permite al estudiante realizar un buen estado del arte, sin consumir esfuerzos innecesarios. Algo destacable del proceso, que muchas veces es omitido, es el cuestionamiento de qué se quiere buscar y determinar con qué términos será buscado.
- Los lineamientos propuestos por Lumsden [32] para el diseño de cuestionarios online, resume en una serie de pasos fácilmente entendibles, el proceso de realizar una encuesta y difundirla por medio del internet. El hecho de dirigir el proceso de acuerdo a preguntas de investigación permite reflexionar y diseñar una encuesta identificando elementos que se quieren conocer y que de otra manera podrían pasar desapercibidos.
- El uso de la técnica CAWI (Computer Assisted Web Interview) para la recolección de información, presenta ventajas y desventajas. Como ventajas se encontró que esta técnica es económica de realizar, comparado a otras técnicas y permite la fácil y rápida obtención de información y los reportes automatizados reduciendo así el esfuerzo en análisis. Como desventaja se encontró que difundir la encuesta es algo difícil, pues al no tener contacto directo con el individuo que se quiere encuestar, no es fácil sugestionarlo para que lo haga. Además muchas veces la encuesta fue retirada de foros, donde inicialmente se publicó, porque se pensaba que era *spam*. Sin embargo, el hecho de que no todos quisieran contestar la encuesta, sino quienes realmente estaban interesados y tenían la dedicación, era un filtro que permitía saber información más certera. La mejor manera de difundir la encuesta fue por medio de grupos especializados en redes sociales.
- Si bien, el meta proceso de formalización de procesos de Ruiz [18] es especialmente dirigido para pequeñas compañías de software es lo suficientemente ligero y moldeable como para ser adaptado a la formalización del proceso *Audacious Freelance* que es para una sola persona. Se realizaron

algunas adaptaciones menores en las fases de planeación y entrega. En la fase de ejecución, que es la esencia del proyecto no se realizaron mayores cambios.

- El uso de la herramienta EPF para la especificación fue un reto. Al comienzo es algo difícil de entender el paradigma que maneja, sin embargo, el manual oficial provee una muy buena manera de aprender a usarlo. Luego de saberlo usar, EPF es una herramienta clave para la formalización de procesos ya que permite gestionar el contenido que tendrá y ordenarlo de una manera lógica.
- La guía realizada por Runeson [19] para realizar estudio de casos es un conocimiento que sin dudas todo tesista debería conocer. Aporta muy buenos lineamientos y conocimiento sobre conceptos que permite hacer un buen trabajo. Complementariamente, el uso del protocolo de Brereton [39] agiliza las cosas.
- El realizar una ficha, como la mostrada en la sección 5.2 de este documento, antes de comenzar con el estudio de caso formal, permite planear y diseñar un buen estudio de caso que sea coherente con lo que se busca resolver.
- El uso de información cuantitativa aumenta considerablemente el tiempo de análisis, sin embargo, reduce significativamente el tiempo para la obtención de dicha información.
- El *Freelance*, al ser de un perfil tan práctico, no pone mucha atención a asuntos tan teóricos como la especificación del proceso. Sin embargo, se encontró como oportunidad, el mejoramiento de la herramienta para que haga más transparente el proceso.

7.3. Limitaciones

- Al realizar la encuesta con la técnica CAWI y sin contar con una base de datos de trabajadores *Freelance*, solo se consiguió que le encuesta fuera respondida por 66 personas.
- La cantidad de *Freelance* que evaluaron el proceso en el estudio de caso aún es escaso. Se requiere de un proyecto con mayores alcances y presupuesto para poder realizar una evaluación masiva que permita así conocer más certeramente el aporte del *Audacious Freelance* a la industria, con sujetos de estudio con perfiles más diversos.
- Debido al alcance de este proyecto, y porque en sus aportes inicialmente no se contaba con el desarrollo, la herramienta AFD es un prototipo que incluye parcialmente los elementos de *Audacious Freelance* y debido al énfasis que pusieron los sujetos de estudio a la herramienta, la evaluación del proceso se vió afectada.

7.4. Trabajos futuros

- Mejorar la herramienta AFD para que abarque la totalidad los elementos de AF junto con las sugerencias obtenidas en la retroalimentación de la evaluación, y así poder evaluar de mejor manera y completamente el proceso.
- Realizar una mejora del proceso *Audacious Freelance* apuntando a niveles superiores madurez de procesos con el fin de hacerlo más íntegro y que pueda aportar más al *Freelance*.

- Evaluar el *Audacious Freelance* en un ambiente académico, con estudiantes que estén cursando materias de laboratorios en donde se desarrollen productos software, pues se considera que en estas condiciones AF podría ser aplicado sin inconvenientes. Además, la herramienta AFD podría ser evolucionada para permitir trabajar colaborativamente entre desarrolladores.
- Evaluar AF en otras áreas del modelo *Freelance* (escritores, fotógrafos, consultores) que no tengan que ver con realizar un producto software, con previa adaptación.
- Adecuar AF a diferentes características de los metodologías ágiles, entre ellas el ritmo sostenido. Además hacer más evidente la influencia de descrito en el manifiesto ágil.

BIBLIOGRAFÍA

- [1] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming – An Agile Process for Autonomous Developers," 2009. [Online]. Available: <http://www.bg-openaire.eu/handle/10867/28>. [Accessed: 18-Nov-2014].
- [2] A. Nyström, "Defining and Evaluating an Agile Software Development Process for a Single Software Developer," M.S, University of Gothenburg, Gothenburg -- Sweden, 2011.
- [3] S. Bonar, "The Freelance Marketplace Review," *WhichLance.com*, 2010. [Online]. Available: http://www.onlineprnews.com/framework/uploads/files/6bfab485982c38c2b1b8d78fd0b3c310_Freelance%20Market%20Review.pdf. [Accessed: 23-Sep-2014].
- [4] R. Middkekopp, "How to (not) manage a one-person software development project," *Rody Middelkoop*, woensdag, mei-2010. [Online]. Available: <http://rodimiddelkoop.blogspot.com/2010/05/how-to-not-manage-one-person-software.html>. [Accessed: 18-Nov-2014].
- [5] A. van den Born and A. van Witteloostuijn, "Drivers of freelance career success," *J. Organ. Behav.*, vol. 34, no. 1, pp. 24–46, Jan. 2013.
- [6] A. Walter, "Success Factors in Leveraging Freelance Marketplaces in Software Development Projects," M.Sc., University of Ottawa (Canada), Canada, 2013.
- [7] P. Selvaraj, "NEED A NERD – Adapting PCSE (Practitioner Centered Software Engineering) to develop a Web Application," M.S, Auburn University, Alabama, 2012.
- [8] "Fedesoft | Federación Colombiana de la Industria de Software y TI." .
- [9] "La industria del software en Colombia vive una época dorada | Fedesoft | Federación Colombiana de la Industria de Software y TI." .
- [10] "Las 7 claves del software colombiano en 2012 y 2013, según Fedesoft | Fedesoft | Federación Colombiana de la Industria de Software y TI." .

-
- [11] FITI and Fedesoft, “Estudio de salarios y profesionales del sector de software y TI de Colombia,” 2012.
 - [12] C. Dorman, “An experience report of the solo iterative process,” M.S., Wayne State University, United States -- Michigan, 2011.
 - [13] “The Harsh Reality of Indie Game Success.” .
 - [14] “Fez sells 100,000 copies on XBLA.” .
 - [15] “The Insane Success of Minecraft: \$33M and Counting.” .
 - [16] “What Flappy Bird’s success can teach the game industry,” 10-Mar-2015. .
 - [17] Anonymous, “Are We Becoming a Freelance Economy? Definitely,” *Growth Strateg.*, no. 1083, pp. 3–4, Mar. 2014.
 - [18] P. Ruiz, A. Quispe, M. C. Bastarrica, and J. Hurtado Alegría, “Formalizing the software process in small companies,” *8CCC Colomb. August 2013*, 2012.
 - [19] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.
 - [20] I. C. Society, *Guide to the Software Engineering Body of Knowledge (Swebok(r)): Version 3.0*. IEEE Computer Society Press, 2014.
 - [21] J. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos, “Systematic review in software engineering,” *Syst. Eng. Comput. Sci. Dep. COPPEUFRJ Tech. Rep. ES*, vol. 679, no. 05, p. 45, 2005.
 - [22] F. Pino, F. García, and M. Piattini, “Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas,” *Rev. Esp. Innov. Calid. E Ing. Softw.*, vol. 2, no. 1, pp. 6–23, 2006.
 - [23] W. S. Humphrey, *PSP(sm): A Self-Improvement Process for Software Engineers*. Addison-Wesley Professional, 2005.
 - [24] P. Ferguson, W. S. Humphrey, S. Khajenoori, S. Macke, and A. Matvya, “Results of applying the Personal Software Process,” *Computer*, vol. 30, no. 5, pp. 24–31, May 1997.
 - [25] V. Rajlich, “Solo Iterative Process,” in *Software Engineering: The Current Practice*, CRC Press, 2011, pp. 181–192.
 - [26] V. Rajlich and P. Gosavi, “Incremental change in object-oriented programming,” *Softw. IEEE*, vol. 21, no. 4, pp. 62–69, 2004.
 - [27] D. Umphress, “Software process course notes,” Auburn University, Computer Science and Software Engineering Department, Fall 2010.
 - [28] R. Agarwal and D. Umphress, “Extreme programming for a single person team,” in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, 2008, pp. 82–87.

-
- [29] G. Cronin, “eXtreme Solo - A Case Study in Single Developer eXtreme Programming,” University of Auckland, Technical report.
- [30] E. Akpata and K. Riha, “Can Extreme Programming be used by a Lone Programmer?,” Kingston University, Technical report, 2004.
- [31] P. M. Shields and N. Rangarajan, “Chapter 4. Description - Categories,” in *A Playbook for Research Methods: Integrating Conceptual Frameworks and Project Management*, New Forums Press, 2013, pp. 71–108.
- [32] J. Lumsden, “Online-Questionnaire Design Guidelines,” in *Handbook of research on electronic surveys and measurements*, R. A. Reynolds, R. Woods, and J. D. Baker, Eds. Hershey, PA: Idea Group Reference, 2007.
- [33] E. Gandia, “2012 Freelance Industry Report,” Aug. 2012.
- [34] OMG, “Software & Systems Process Engineering Metamodel Specification, v2.0 (Beta 2),” Technical report, 2007.
- [35] H. Kniberg, “How we combine Scrum with XP,” in *Scrum and XP from the Trenches*, EEUU: C4Media, 2007, pp. 81–85.
- [36] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. 2002.
- [37] K. Schwaber and J. Sutherland, “The Scrum Guide.” Jul-2013.
- [38] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2000.
- [39] P. Brereton, B. Kitchenham, D. Budgen, and Z. Li, “Using a protocol template for case study planning,” in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. University of Bari, Italy*, 2008.