

FRAMEWORK PARA SOPORTAR LA EVALUACIÓN DE LA AGILIDAD DE LOS  
PROCESOS SOFTWARE DE LAS ORGANIZACIONES



WILSON ALFREDO ORTEGA ORDOÑEZ

Tesis de Maestría en Computación

Director:

PhD. César Jesús Pardo Calvache

Codirector:

PhD. Francisco José Pino Correa

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Línea de Investigación: Ingeniería de Software

Popayán, noviembre de 2019



WILSON ALFREDO ORTEGA ORDOÑEZ

FRAMEWORK PARA SOPORTAR LA EVALUACIÓN DE  
LA AGILIDAD DE LOS PROCESOS SOFTWARE DE LAS  
ORGANIZACIONES

Tesis presentada a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de

Magíster en  
Computación

Director: PhD. César Jesús Pardo Calvache  
Codirector: PhD. Francisco José Pino Correa

Popayán  
2019



Página de aceptación



*A Dios por brindarme  
tantas oportunidades de crecimiento  
personal y profesional, y por  
concederme sabiduría y paciencia en los  
momentos difíciles.*

*A mi abuelita que está en el cielo  
y que siempre iluminó mis días  
con su mirada llena de amor  
y de apoyo incondicional.*

*A mi madre, mi hermana,  
mi esposa y mi hijo Martín  
por todo su apoyo, paciencia y amor.  
Por ser el motor de mi vida.*





## **Agradecimientos**

Al doctor César Jesús Pardo Calvache, por su excelente dirección, por compartirme su experiencia y conocimiento durante todo el proyecto, y por su apoyo, paciencia y motivación.

Al doctor Francisco José Pino Correa, por su colaboración en la codirección de este trabajo de grado, aportando su gran experiencia en el área de ingeniería de software.

A la doctora Elizabeth Suescún Monsalve, por su gran colaboración y hospitalidad durante mi estancia de investigación en la Universidad EAFIT.

A los excelentes docentes de la Maestría en Computación, por aportar desde su conocimiento y experiencia a mejorar este trabajo de grado.

A mi familia, por su paciencia, apoyo y amor incondicional.



## Resumen Estructurado

Antecedentes: Los enfoques ágiles para el desarrollo de software, se basan principalmente en los valores y principios definidos en el manifiesto ágil. Actualmente, existe un amplio portafolio de enfoques ágiles, sin embargo, en muchos casos su implementación se realiza de manera informal y definiendo procesos de desarrollo de software que no están alineados con los principios y valores ágiles.

Objetivo: Definir un framework para llevar a cabo la evaluación de la agilidad de los procesos software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles.

Métodos: Se utilizaron los siguientes métodos de investigación: i) el método Investigación-Acción, ii) el grupo focal, y iii) el estudio de caso.

Resultados: i) un mapeo sistemático de la literatura relacionada con la evaluación de la agilidad, ii) una ontología para el desarrollo ágil de software, iii) un modelo de referencia, un método de evaluación y un prototipo de una aplicación web para soportar la evaluación de la agilidad de los procesos de desarrollo de software, v) una monografía con los resultados de la investigación, y vi) dos artículos publicados en revistas indexadas.

Conclusiones: La solución propuesta, soporta la evaluación de la agilidad de los procesos de desarrollo de software y fue evaluada con éxito mediante un estudio de caso en una organización desarrolladora de software. Como trabajo futuro se plantea la actualización del mapeo sistemático y de la ontología, y la automatización del proceso de evaluación.

**Palabras clave:** evaluación de agilidad, ontología, procesos software, desarrollo ágil.



## Structured Abstract

Background: Agile approaches to software development are mainly based on the values and principles defined in the agile manifesto. Currently, there is a broad portfolio of agile approaches, however, in many cases its implementation is done informally and defining software development processes that are not aligned with agile principles and values.

Aim: To define a framework to carry out the agility assessment of the software processes of organizations by analyzing compliance with agile principles and values.

Methods: The following research methods were used: i) the Research-Action method, ii) the focus group, and iii) the case study.

Results: i) a systematic mapping of the literature related to agility assessment, ii) an ontology for agile software development, iii) a reference model, an evaluation method and a prototype of a web application to support the agility assessment of software development processes, v) a monograph with research results, and vi) two articles published in indexed journals.

Conclusions: The proposed solution supports the agility assessment of software development processes and was successfully evaluated through a case study in a software development organization. As future work, the updating of the systematic mapping and the ontology, and the automation of the assessment process are proposed.

**Keywords**: agility assessment, ontology, software processes, agile development.



# Contenido

Lista de Figuras.....	ix
Lista de Tablas.....	xi
Capítulo 1.....	13
Introducción.....	13
1.1. Definición del Problema.....	13
1.2. Objetivos.....	18
1.2.1. Objetivo General - OG.....	18
1.2.2. Objetivos específicos - OE.....	18
1.3. Metodología.....	19
1.3.1. Ciclo conceptual. Análisis conceptual.....	19
1.3.2. Ciclo metodológico. Definición de la solución.....	20
1.3.3. Ciclo de evaluación. Evaluación de la solución.....	20
1.3.4. Ciclo de documentación y socialización.....	21
1.4. Resultados Obtenidos.....	21
1.5. Organización del documento.....	21
Capítulo 2.....	23
Marco teórico y estado del arte.....	23
2.1. Marco teórico.....	23
2.1.1. Proceso software.....	23
2.1.2. Evaluación del proceso software.....	24
2.1.3. Modelo de referencia.....	24
2.1.4. Ontología.....	24
2.1.5. Metodologías para la implementación de ontologías.....	25
2.1.6. Integración de ontologías.....	25

2.1.7	Goal Question Metric – GQM .....	26
2.1.8	Manifiesto ágil.....	26
2.2.	Estado del arte .....	26
2.2.1.	Estado del arte acerca de la evaluación de la agilidad en las organizaciones .....	27
2.2.2.	Estado del arte acerca de la ontologías sobre la evaluación de la agilidad de los procesos software.....	28
Capítulo 3.....		31
AgilityPRO: Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones .....		31
3.1.	Componentes de AgilityPRO .....	31
3.1.1.	Capa conceptual.....	32
3.1.2.	Capa metodológica.....	32
3.1.3.	Capa tecnológica.....	32
3.2.	OntoAgile: Una ontología para procesos de desarrollo ágil de software .....	33
3.2.1.	Metodología.....	34
3.2.2.	Propósito .....	34
3.2.3.	Preguntas de competencia .....	35
3.2.4.	Integración de OntoAgile con otras ontologías .....	35
3.2.5.	Conceptos y relaciones de OntoAgile.....	36
3.2.6.	Discusión de algunos conceptos sus relaciones .....	41
3.2.7.	Representación gráfica.....	36
3.2.8.	Evaluación de OntoAgile .....	41
3.3.	Modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityRef .....	46
3.3.1.	Valores ágiles .....	47
3.3.2.	Principios ágiles.....	48
3.3.3.	Aspectos de agilidad.....	50
3.4.	Método de evaluación de la agilidad de los procesos de desarrollo de software – AgilityPval 54	
3.4.1.	Propósito de AgilityPval .....	54
3.4.2.	Objetivos específicos de AgilityPval .....	55
3.4.3.	Componentes de AgilityPval.....	55
3.5.	Prototipo Web - AgilityProTool.....	63



3.5.1. Requisitos funcionales.....	63
3.5.2. Arquitectura de la aplicación.....	64
3.5.3. Diagrama de secuencia general .....	66
3.5.4. Enfoque de desarrollo .....	67
3.5.5. Resultados obtenidos .....	68
Capítulo 4.....	71
Evaluación de AgilityPRO .....	71
4.1. Evaluación de AgilityRef usando un grupo focal.....	71
4.1.1. Planeamiento de la investigación .....	72
4.1.2. Diseño del grupo de discusión (Reclutamiento) .....	74
4.1.3. Conducción de la sesión de debate.....	76
4.1.4. Captura de información .....	76
4.1.5. Análisis de la información y reporte de resultados .....	78
4.1.6. Acciones de mejora .....	82
4.2. Estudio de caso .....	87
4.2.1. Diseño del estudio de caso.....	87
4.2.2. Sujeto de estudio.....	88
4.2.3. Procedimiento de campo y recolección de datos .....	88
4.2.4. Análisis de resultados del estudio de caso.....	95
4.2.5. Análisis de validez .....	98
4.2.6. Limitaciones del estudio de caso.....	99
Capítulo 5.....	101
Conclusiones y trabajo futuro.....	101
5.1. Análisis de los objetivos de investigación.....	101
5.1.1. Objetivos específicos - OE .....	101
5.1.2. Objetivo general - OG.....	103
5.2. Publicaciones .....	104
5.3. Conclusiones.....	104
5.4. Trabajo futuro .....	107
Bibliografía .....	109
Anexo A.....	115
Mapeo sistemático sobre la evaluación de la agilidad en organizaciones de desarrollo de software	115

A.1	Trabajos relacionados .....	115
A.2	Metodología .....	117
A.3	Resultados.....	119
A.4	Brechas existentes .....	129
Anexo B.....		131
Modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityRef – Versión 1.0.....		
B.1	Valores ágiles .....	132
B.2	Principios ágiles.....	134
B.3	Indicadores de agilidad.....	136
Anexo C .....		143
Cuestionarios de evaluación del grupo focal .....		
Anexo D .....		157
Documentación del estudio de caso.....		
D.1	Aceptación de participación en el estudio de caso .....	157
D.2	Instrumento de evaluación diligenciado.....	159

## Lista de Figuras

Figura 3.1 Vista general de los componentes de AgilityPRO. ....	33
Figura 3.2 Representación gráfica de OntoAgile usando UML. ....	36
Figura 3.3 Extracto de instanciación de Scrum en OntoAgile. ....	43
Figura 3.4 Instanciación de OntoAgile para soportar la evaluación de la agilidad. ....	44
Figura 3.5 Ejemplo de una aplicación web que realiza consultas a OntoAgile. ....	44
Figura 3.6 Componentes de AgilityRef. ....	47
Figura 3.7 Relaciones entre principios y valores ágiles en AgilityRef. ....	50
Figura 3.8 Diagrama de despliegue de AgilityProTool. ....	65
Figura 3.9 Diagrama de clases de AgilityProTool. ....	65
Figura 3.10 Diagrama de secuencia general de AgilityProTool. ....	66
Figura 3.11 Página del cuestionario de evaluación en AgilityProTool. ....	69
Figura 3.12 Página de listado de evaluaciones en AgilityProTool. ....	69
Figura 4.1 Consolidado de respuestas a preguntas P1-P8. ....	78
Figura 4.2 Gráfico de resultados del grado de cumplimiento de cada aspecto de agilidad. ....	91
Figura 4.3 Gráfico de resultados del grado de cumplimiento de cada principio ágil. ....	92
Figura 4.4 Gráfico de resultados del grado de cumplimiento de cada valor ágil. ....	93



## Lista de Tablas

Tabla 3.1. Preguntas de competencia definidas para OntoAgile.....	35
Tabla 3.2. Glosario de conceptos de OntoAgile. ....	39
Tabla 3.3. Relaciones de OntoAgile. ....	40
Tabla 3.4. Preguntas de competencia expresadas en SPARQL.....	45
Tabla 3.5. Comparación de ontologías existentes para enfoques ágiles. ....	46
Tabla 3.6. Interpretación de los valores ágiles en AgilityRef. ....	48
Tabla 3.7. Principios ágiles en AgilityRef. ....	49
Tabla 3.8. Aspectos de agilidad en AgilityRef. ....	53
Tabla 3.9. Relación entre principios ágiles y aspectos de agilidad. ....	54
Tabla 3.10. Indicadores de agilidad de AgilityPval. ....	58
Tabla 3.11. Métricas definidas en AgilityPval. ....	59
Tabla 3.12. Escala de calificación definida en AgilityPval. ....	60
Tabla 3.13. Roles definidos en AgilityPval. ....	62
Tabla 3.14. Historias épicas definidas para AgilityProTool.....	64
Tabla 3.15. Iteraciones definidas para el desarrollo de AgilityProTool. ....	67
Tabla 4.1. Protocolo definido para el grupo focal. ....	73
Tabla 4.2. Elementos para la realización del grupo focal. ....	73
Tabla 4.3. Perfil profesional de los participantes en el grupo focal. ....	75
Tabla 4.4. Organización del grupo focal. ....	76
Tabla 4.5. Cuestionario de evaluación usado en el grupo focal.....	77
Tabla 4.6. Conteo de respuestas a preguntas P1-P8.....	78
Tabla 4.7. Respuestas a las preguntas abiertas .....	82
Tabla 4.8. Acciones de mejora definidas para AgilityRef .....	86
Tabla 4.9. Resultados grado de cumplimiento de los aspectos de agilidad en el estudio de caso. ....	90
Tabla 4.10. Resultados grado de cumplimiento de los principios ágiles en el estudio de caso.....	92
Tabla 4.11. Resultados grado de cumplimiento de los valores ágiles en el estudio de caso. ....	92
Tabla 4.12. Oportunidades de mejora encontradas en el estudio de caso.....	95
Tabla 4.13. Cuestionario de evaluación de AgilityPval.....	98



# Capítulo 1

## Introducción

En este capítulo se brinda una descripción detallada del problema de investigación abordado, los objetivos de la investigación, la metodología utilizada y los resultados obtenidos.

### 1.1. Definición del Problema

Los frameworks, métodos y metodologías ágiles (en adelante enfoques ágiles) para el desarrollo de software brindan grandes beneficios a las organizaciones software superando inconvenientes relacionados con el cambio constante de requerimientos y con productos que no satisfacen al cliente [1]. Los enfoques ágiles enfocan su atención principalmente sobre los siguientes valores o pilares fundamentales: (i) los individuos e interacciones sobre los procesos y herramientas, (ii) el software funcionando sobre la documentación extensiva, (iii) la colaboración con el cliente sobre la negociación contractual y (iv) la respuesta ante el cambio sobre seguir un plan [2].

A partir de los valores mencionados anteriormente, se derivan doce principios ágiles [2]: (i) satisfacer al cliente con entregas tempranas y continuas, (ii) aprovechar el cambio en los requisitos en cualquier etapa del proyecto para proporcionar ventajas competitivas al cliente, (iii) entregar con frecuencia software que funciona, (iv) promover el trabajo colaborativo entre los responsables del negocio y los desarrolladores, (v) motivar el equipo procurando un entorno apropiado de trabajo, apoyo y confianza, (vi) el mejor método de comunicar información al equipo y entre sus miembros es la conversación cara a cara, (vii) el software funcionando es la

medida principal de progreso, (viii) mantener un ritmo constante de trabajo de forma indefinida, (ix) buscar continuamente la excelencia técnica y el buen diseño, (x) maximizar la cantidad de trabajo no realizado, (xi) equipos auto-organizados, y (xii) llevar a cabo retrospectivas o reflexiones del equipo a intervalos regulares sobre la forma de ser más efectivo y que les permita ajustar su comportamiento en consecuencia.

Actualmente, existe un amplio portafolio de enfoques ágiles utilizados frecuentemente en la industria de software mundial, entre los más destacados se encuentran: Scrum [3], Extreme Programming (XP) [4], Crystal Clear [5], Lean Software Development (LSD) [6], Adaptive Software Development (ASD) [7], Dynamic Systems Development Method (DSDM) [8], Feature-Driven Development (FDD) [9], Agile Unified Process (AgileUP) [10], Kanban [11], entre otros. En comparación con los enfoques tradicionales, se puede decir que los enfoque ágiles son menos exhaustivos respecto a la documentación y burocracia, más efectivos y flexibles en los entornos característicos de las MiPyMEs y agregan valor integrando al cliente en el proceso de desarrollo del producto software [12]. En la actualidad, las organizaciones están utilizando este tipo de enfoques porque les permiten reducir sus tiempos de entrega del producto final (time to market), mejorar la calidad y productividad, mejorar la alineación entre los objetivos de negocio y las tecnologías de la información, gestionar los cambios, y la reducción de los riesgos y costos en los proyectos [13], [14].

Asimismo, con respecto al uso de enfoques ágiles en la industria, en el reporte de resultados de la encuesta mundial anual acerca del estado de la agilidad “*state-of-agile*” del 2018 [13], se muestra una tendencia creciente de adopción de este tipo de enfoques por parte de las empresas de software en los último años. Además, se encontró que el 97% de las empresas encuestadas en Asia, América, Europa, África y Oceanía utilizan prácticas ágiles en sus proyectos. Asimismo, es posible observar que la madurez de las empresas en cuanto a sus competencias en el uso de enfoques ágiles está representada de la siguiente manera: el 4% no tienen iniciativas para lograr la agilidad empresarial, otro 5% están considerando una iniciativa ágil, un 21% están experimentando con prácticas ágiles, el 53% usan prácticas ágiles, pero aún están en proceso de madurez y el 6% restante, tienen un alto nivel de competencia en el uso de este tipo de prácticas.



Con relación a los enfoques ágiles más utilizados por las empresas encuestadas, el primer lugar lo ocupa Scrum con un 54%, el segundo lugar corresponde al uso de enfoques híbridos con un 14%, el tercer lugar corresponde a un híbrido entre Scrum y XP con un 10%, en un 8% utilizan la integración de Scrum y Kanban denominando esta integración como Scrumban, un 5% usan sólo Kanban, un 3% utilizan un enfoque iterativo, un 3% no saben, 2% utilizan Lean Startup, y el 1% aplican XP.

Para aprovechar las ventajas que brindan los enfoques ágiles, las organizaciones deben afrontar un proceso de transformación socio-técnico que genere cambios en las prácticas y técnicas a nivel de gestión, personas, procesos, técnicas, tecnologías y herramientas utilizadas [15], [16]. El estilo de gestión de la organización debe eliminar los elementos de autoritarismo y control, e integrar elementos de colaboración con los equipos [16]. Los miembros de los equipos deben interiorizar los principios y valores ágiles y establecer un canal de comunicación y cooperación con todos los interesados en el proyecto. Otro cambio importante se debe dar en los procesos de la organización, pues se debe invertir una gran cantidad de tiempo y esfuerzo en pasar de prácticas centradas en los procesos a prácticas centradas en las personas. Esto puede ser un desafío importante para las organizaciones que llevan un tiempo considerable trabajando con procesos rígidos y complejos [17] a través de la implementación de enfoques tradicionales. En lo que respecta a los retos tecnológicos, las organizaciones deben invertir en tecnología y capacitación relacionada con herramientas para la gestión ágil de sus proyectos, integración y entrega continua, control de versiones, entre otras [16].

Teniendo en cuenta lo anterior, es posible pensar que la implementación de enfoques ágiles no se debe tomar a la ligera, y que dicho proceso es altamente complejo y requiere de una gran cantidad de esfuerzo. Incluso al igual que un proyecto de mejora de procesos con enfoques tradicionales, en la mejora de procesos basados en enfoques ágiles se deben enfrentar algunos desafíos inclusive en mayor medida que en los enfoques tradicionales, por ejemplo: (i) desacuerdo entre la filosofía o cultura de la empresa y los valores ágiles, (ii) la falta de experiencia con enfoques ágiles, (iii) entrenamiento insuficiente, (iv) presencia dominante de prácticas de desarrollo tradicional, (v) incertidumbre y desconocimiento en prácticas detalladas y claras sobre cómo adoptar el enfoque ágil en la

organización, y la (vi) informalidad e indisciplina en la adopción de prácticas ágiles, entre otros [1], [14], [18]–[21].

Por otro lado, el amplio abanico de enfoques ágiles junto con los enfoques tradicionales, permite que las organizaciones software dispongan de diferentes marcos de referencia con los que pueden llevar a cabo la confección de procesos, lo cual involucra la creación, modificación o adaptación de la descripción de procesos para un fin en particular dentro de una organización [21]. Este amplio espectro de marcos permite que las organizaciones puedan elegir entre uno o varios referentes para dar soporte a necesidades específicas de la organización, la utilización de varios marcos se puede llevar a cabo a través de su armonización [22], lo que resulta en la creación de soluciones híbridas. Algunas experiencias sugieren que la efectividad de la implementación de un enfoque ágil o tradicional puede depender en gran medida del ajuste cuidadoso que se hace en el proceso definido [23], [24], sin embargo, es importante resaltar que diseñar procesos, afinarlos y que además sean ágiles resulta particularmente difícil, esto, debido a que la definición de procesos ágiles carece de soporte para muchas de las técnicas definidas para el análisis, modelado y rediseño de procesos; técnicas que se han utilizado principalmente para procesos más estructurados basados en enfoques tradicionales [25].

A partir de la realización de un mapeo sistemático de la literatura [26], el cual tuvo como objetivo conocer el estado del arte acerca de la evaluación de la agilidad en las organizaciones, ha sido posible identificar algunos trabajos que describen soluciones relacionadas con la evaluación de la agilidad de los procesos de desarrollo de software, proyectos, enfoques de desarrollo, equipos de desarrollo y organizaciones, sin embargo, no existe consenso en cuanto a los atributos que deben tener los procesos para ser considerados ágiles. Si se tiene en cuenta que los principios y valores ágiles constituyen las características que debe tener el desarrollo de software para que sea considerado ágil, los enfoques de evaluación deberían tener en cuenta cada uno de esos principios y valores para realizar la evaluación. Por otra parte, se puede observar que la mayoría de enfoques de evaluación definen sus propias características y no cubren completamente los principios y valores expresados en el manifiesto ágil, incluso, no se aprecia que sean considerados como un complemento o extensión para facilitar su comprensión y adopción. También se puede observar que ninguna de las propuestas encontradas presenta de forma clara y detallada la

relación entre los elementos de proceso y los principios y valores ágiles, razón por la cual no es posible determinar el grado de cumplimiento del manifiesto ágil a partir de la evaluación del proceso. Además, no se proporciona una descripción clara de la terminología relacionada con la evaluación de la agilidad de los procesos de desarrollo de software, lo cual dificulta la aplicación de la evaluación por parte de las organizaciones debido a problemas de ambigüedad en los términos empleados en distintos enfoques ágiles y tradicionales. Asimismo, no hay evidencia de la existencia de una herramienta web de evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones. Las herramientas existentes están enfocadas en la evaluación de la agilidad de la organización, del equipo o del proyecto.

De acuerdo a lo anterior, surgió la siguiente pregunta de investigación: ¿Cómo determinar que un proceso de desarrollo de software es ágil de acuerdo a un referente internacional como los valores ágiles y principios del manifiesto ágil? Responder a esta pregunta no es sencillo debido a razones como: i) la falta de consenso en la definición de agilidad en el proceso de desarrollo de software, ii) la dificultad para entender las relaciones entre los principios, valores ágiles y los elementos (actividades, roles, productos y herramientas) de los procesos de desarrollo de software, iii) la influencia del contexto en la agilidad del proceso, lo anterior debido a que un proceso que se considera ágil en un contexto, puede no serlo al ser aplicado en un contexto distinto, y iv) la adopción de prácticas de varios enfoques ágiles y tradicionales en un mismo proceso de desarrollo de software.

En este sentido, el objetivo de esta tesis de maestría es proponer un framework para soportar la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles. El framework definido consta de 4 elementos organizados en 3 capas: i) una capa conceptual, la cual está constituida por una ontología del desarrollo ágil de software y un modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software, ii) una capa metodológica que consiste en un método de evaluación basado en el modelo de referencia de la capa conceptual, y iii) una capa tecnológica que proporciona un prototipo de una aplicación web para soportar la evaluación.

El framework propuesto, representa un aporte significativo para que las organizaciones a nivel local, nacional e internacional puedan identificar oportunidades de mejora relacionadas con la agilidad de sus procesos de desarrollo de software y aprovechen en mayor medida las ventajas que ofrecen los enfoques ágiles. Además, mediante esta tesis se aporta al conocimiento en el área de ingeniería del software, específicamente en la línea de metodologías ágiles, proporcionando un estado del arte actualizado acerca de los trabajos relacionados con la evaluación de la agilidad en las organizaciones software. Además, se aporta una ontología que organiza el conocimiento relacionado con la implementación de los enfoques ágiles en los procesos software de las organizaciones de una manera genérica, permitiendo que el conocimiento pueda ser reutilizado y compartido por los interesados en el área.

## **1.2. Objetivos**

A continuación, se presentan los objetivos de la tesis tal como fueron aprobados por el Consejo de la Facultad de Ingeniería Electrónica y Telecomunicaciones.

### **1.2.1. Objetivo General - OG**

**OG:** Definir un framework para llevar a cabo la evaluación de la agilidad de los procesos software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles.

### **1.2.2. Objetivos específicos - OE**

**OE1:** Definir una ontología de dominio específico que permita entender y organizar el conocimiento relacionado con los términos y relaciones alrededor de los procesos software y los valores y principios del manifiesto ágil.

**OE2:** Definir un modelo de referencia que establezca los elementos necesarios para llevar a cabo la evaluación de la agilidad de los procesos software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles.

**OE3:** Desarrollar un prototipo de una aplicación web que permita a las organizaciones evaluar la agilidad de sus procesos software con base en el modelo de referencia propuesto.

**OE4:** Evaluar el modelo propuesto a través de su aplicación en una organización desarrolladora de software mediante un estudio de caso.

## 1.3. Metodología

Para llevar a cabo la ejecución del proyecto, se utilizó el método Investigación-Acción con múltiples ciclos de forma lineal [27] y el estudio de caso [28]. A continuación se describen los cuatro ciclos de investigación llevados a cabo para la realización del proyecto:

### 1.3.1. Ciclo conceptual. Análisis conceptual

En esta fase se llevó a cabo la investigación del estado del arte acerca de la evaluación de la agilidad en las organizaciones. Además, se organizaron los términos y relaciones alrededor de la evaluación de la agilidad de los procesos software mediante una ontología. En esta fase se realizaron las siguientes actividades:

- **Análisis de la literatura:** Se realizó un mapeo sistemático de la literatura acerca de los trabajos relacionados con la evaluación de la agilidad en las organizaciones. Mediante la realización de esta actividad, se identificaron las propuestas y soluciones existentes, las brechas en el área de evaluación de la agilidad de los procesos software, y los elementos sensibles a tener en cuenta para la definición de la solución.
- **Definición de la ontología:** Se definió una ontología de dominio específico acerca de los procesos de desarrollo ágil de software. Esta ontología, permitió identificar los conceptos y relaciones alrededor de los procesos de desarrollo de software y su relación con los principios y valores ágiles.

### 1.3.2. Ciclo metodológico. Definición de la solución

Con base en los resultados del ciclo anterior, se desarrolló cada componente del framework para soportar la evaluación de la agilidad de los procesos software. En esta fase se llevaron a cabo las siguientes actividades:

- **Definición de la arquitectura:** Se definió una arquitectura de tres capas para el framework: i) capa conceptual, ii) capa metodológica, y iii) capa tecnológica.
- **Definición de la capa conceptual:** En esta capa se desarrolló un modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones. Este modelo fue desarrollado a partir de los conceptos identificados en la ontología desarrollada en la fase anterior.
- **Evaluación del modelo de referencia:** Una vez obtenida la primera versión del modelo de referencia, se realizó un grupo focal con expertos en el área de ingeniería de software con el fin de evaluar la propuesta y encontrar oportunidades de mejora. Con base en los resultados del grupo focal se generó una segunda versión del modelo de referencia.
- **Definición de la capa metodológica:** En esta capa se definió un método de evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones. El método propuesto incluye el instrumento de evaluación y el proceso a seguir para aplicarlo en la organización.
- **Definición de la capa tecnológica:** Se desarrolló un prototipo de una aplicación web para soportar la evaluación de la agilidad de los procesos de desarrollo de software en las organizaciones.

### 1.3.3. Ciclo de evaluación. Evaluación de la solución

La evaluación de la solución propuesta se llevó a cabo mediante la realización de un estudio de caso al interior de una organización desarrolladora de software. Para la realización del estudio de caso se llevaron a cabo las siguientes actividades:

- **Diseño del estudio:** Se definió la pregunta de investigación del estudio, se establecieron los criterios de selección de la organización y se definió el procedimiento de campo y de recolección de datos.
- **Realización del estudio:** Se ejecutó el estudio de caso en la organización teniendo en cuenta el diseño planteado en la actividad anterior.
- **Análisis y conclusiones:** Se generó un reporte como resultado del análisis de los datos obtenidos durante la ejecución del estudio de caso.

### 1.3.4. Ciclo de documentación y socialización

Este ciclo se llevó a cabo de manera transversal al proyecto con el fin de organizar toda la documentación obtenida durante su ejecución y socializar los resultados obtenidos. Esta fase incluyó las siguientes actividades:

- **Publicación de artículos:** Se elaboraron y publicaron dos artículos en revistas indexadas con los resultados del mapeo sistemático y del diseño de la ontología.
- **Estancia de investigación:** Se realizó una pasantía en el grupo de investigación I+D+I en Tecnologías de la Información y las Comunicaciones - GIDITIC de la Universidad EAFIT en la ciudad de Medellín – Colombia bajo la tutoría de la PhD. Elizabeth Suescún. Esta pasantía permitió socializar los resultados parciales del proyecto y recibir retroalimentación sobre el proyecto con los demás integrantes del grupo de investigación.
- **Elaboración de la monografía:** Se elaboró el documento final tipo monografía con los resultados obtenidos durante el proyecto.
- **Sustentación del proyecto:** Se presentaron y sustentaron los resultados obtenidos durante el desarrollo del proyecto.

## 1.4. Resultados Obtenidos

Los resultados obtenidos con la realización de este trabajo de maestría fueron:

- Una monografía del trabajo de grado, cuya estructura del documento se detalla en la siguiente sección.
- Prototipo de una aplicación web para soportar la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones.
- Artículos publicados en revistas indexadas (ver sección 5.2)

## 1.5. Organización del documento

Los demás capítulos de este documento están organizados de la siguiente manera:

- **Capítulo 2.** Presenta el marco teórico, el cual incluye los conceptos básicos relacionados con la temática del proyecto, y el estado del arte donde se identificaron las propuestas existentes y las brechas en el área de evaluación de la agilidad de los procesos software.

- **Capítulo 3.** Presenta la descripción detallada de la solución propuesta, denominada: “Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones - AgilityPRO”.
- **Capítulo 4.** Presenta la evaluación de la solución propuesta a través de la realización de un grupo focal y un estudio de caso. En primer lugar, se describe el grupo focal realizado con el fin de evaluar el modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software. Luego, se presenta el estudio de caso realizado para aplicar el método de evaluación de la agilidad de los procesos de desarrollo de software en una organización.
- **Capítulo 5.** Presenta las conclusiones y trabajo futuro.
- **Bibliografía**
- **Anexos:** Se dividen en:
  - **Anexo A:** Presenta en detalle el mapeo sistemático realizado
  - **Anexo B:** Presenta la primera versión del modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software
  - **Anexo C:** Presenta los cuestionarios de evaluación diligenciados por los participantes del grupo focal
  - **Anexo D:** Presenta documentación del estudio de caso: i) la carta de aceptación de participación, y ii) el instrumento de evaluación diligenciado.



## **Capítulo 2**

### **Marco teórico y estado del arte**

Este capítulo presenta los conceptos más relevantes utilizados durante la realización de esta tesis y que se consideran importantes presentar. Asimismo, se presenta un mapeo sistemático sobre la evaluación de la agilidad en organizaciones de desarrollo de software, y el estado del arte con relación a las ontologías sobre la evaluación de la agilidad de los procesos software. De acuerdo con los objetivos generales de los mapeos sistemáticos, el objetivo fue proporcionar un estado del arte actualizado que sintetizara el trabajo en esta área del conocimiento y que pudiera ser utilizado para identificar las lagunas a partir de las cuales formular actividades de investigación innovadoras. Los trabajos encontrados se clasificaron y analizaron teniendo en cuenta las tendencias de publicación, los modelos utilizados y los métodos y técnicas propuestos. Además, se determinaron los factores que influyen en la evaluación de la agilidad en las organizaciones. Estos temas fueron tenidos en cuenta para definir el framework propuesto en esta tesis de maestría.

#### **2.1. Marco teórico**

A continuación se relacionan algunos conceptos que consideramos importantes para facilitar la lectura del documento.

##### **2.1.1 Proceso software**

En general, un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. En el contexto de la ingeniería de software, un proceso software es un enfoque adaptable que permite que el equipo de desarrollo busque y elija el conjunto apropiado de acciones y tareas

para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán [29].

### **2.1.2 Evaluación del proceso software**

Este tipo de evaluación permite comprobar si un proceso software cumple con ciertos criterios de proceso básicos que se haya demostrado que son esenciales para el éxito de la ingeniería de software [29]. Entre los enfoques de evaluación de procesos más usados se encuentran: Método de evaluación del estándar CMMI para el proceso de mejora (SCAMPI, por sus siglas en inglés) [30], ISO/IEC 9003:2004 (ISO 9001:2000 para software) [31], y SPICE (ISO/IEC 15504) [32].

### **2.1.3 Modelo de referencia**

Un modelo de referencia se puede definir como: “Un modelo conceptual genérico que formaliza prácticas recomendadas para cierto dominio” [33]. Las prácticas recomendadas, también llamadas mejores prácticas, expresan la mejor manera para tratar un problema particular y pueden ser replicadas y adaptadas por una organización en una situación similar [34]. Un modelo de referencia incluye un conjunto mínimo de conceptos, axiomas y relaciones propios de un dominio particular de problema, y es independiente de estándares específicos, tecnologías, implementaciones, o de cualquier otro detalle concreto [35].

### **2.1.4 Ontología**

Gruber [36] define una ontología como la especificación explícita de una conceptualización. En esta definición, el término conceptualización hace referencia a los objetos, conceptos y otras entidades cuya existencia se asume en un área de interés y las relaciones entre ellas. Las ontologías permiten clarificar la estructura del conocimiento en un dominio, ayudan a reducir la ambigüedad de conceptos y términos, y permiten a distintos agentes (personas o sistemas) compartir el conocimiento acerca de un dominio [37]. Según la clasificación propuesta por [37], en el área de la ingeniería del software, las ontologías pueden ser agrupadas en dos categorías: i) ontologías de dominio, aquellas que describen el conocimiento de un dominio particular y ii) ontologías como artefactos software, las cuales sirven como artefactos durante el proceso de desarrollo de software.

### **2.1.5 Metodologías para la implementación de ontologías**

Existen varias metodologías que permiten sistematizar la implementación de ontologías, por ejemplo: gestión del conocimiento basada en ontologías [38], Methontology [39], un enfoque de traducción para especificaciones ontológicas portables [40], formalismo de representación para ontologías de ingeniería de software - REFSENO [41], entre otras.

Para la implementación de la ontología propuesta en este trabajo, se optó por usar la metodología REFSENO, la cual fue diseñada como una especialización de la metodología Methontology para el desarrollo de ontologías en el área de ingeniería del software. REFSENO permite representar ontologías semi-formales mediante tablas, texto y opcionalmente diagramas. En esta metodología, el conocimiento se puede representar en dos niveles de abstracción: nivel de conocimiento y nivel de símbolo. El nivel de símbolo se define mediante la implementación mientras que los niveles de conocimiento son independientes de la implementación [41]. En resumen, un nivel de conocimiento describe "qué" representar, mientras que el nivel de símbolo describe "cómo" representarlo. Según [32], REFSENO propone tres niveles de conocimiento: el nivel epistemológico, el nivel conceptual y el nivel lingüístico. El nivel epistemológico describe conceptos, atributos y relaciones. El nivel conceptual describe el vocabulario estándar, y el nivel lingüístico describe instancias concretas de las construcciones definidas en el nivel anterior. Las etapas sugeridas para la construcción de una ontología en el nivel conceptual usando REFSENO son: i) Construcción, ii) Evolución y iii) Validación.

En la etapa de Construcción, se llevan a cabo las siguientes actividades: i) establecer la especificación de los requisitos de la ontología, ii) definir el glosario de términos, iii) identificar las relaciones semánticas entre los conceptos, iv) identificar los atributos terminales para todos los conceptos, v) verificar la integridad de la tabla de atributos de concepto y definir instancias basadas en la especificación de requisitos.

### **2.1.6 Integración de ontologías**

En el contexto de las ontologías, es posible que la palabra integración tenga varios significados [42], el uso abusivo de esta palabra hace difícil organizar las diferentes ontologías definidas que resultan de una integración. Puede parecer increíble, pero incluso en el área de conocimiento relacionado con el diseño y creación de

ontologías es posible notar que existen ciertas ambigüedades. El término integración puede ser definido a través de diferentes significados [43]: (i) matching: referido al proceso de encontrar relaciones o correspondencias entre las entidades de ontologías diferentes, (ii) alignment: referido al resultado obtenido luego de identificar el conjunto de correspondencias entre dos o más ontologías, (iii) mapping: la versión orientada o dirigida de una alineación, éste mapea las entidades de una ontología en al menos una entidad de otra ontología, (iv) merging: la creación de una nueva ontología a partir de dos ontologías de origen, posiblemente superpuestas, e (v) integration: la inclusión de una ontología en otra ontología junto a las aserciones que las relacionan.

### **2.1.7 Goal Question Metric – GQM**

GQM es un paradigma para desarrollar y mantener un programa de métricas que plantea como principio básico, que la medición debe ser realizada siempre orientada hacia un objetivo. Este paradigma propone un modelo de medición de tres niveles: (i) Nivel conceptual (Goal), (ii) Nivel operacional (Question), y (iii) Nivel cuantitativo (Metric). GQM define un objetivo, refina este objetivo en preguntas y define métricas que intentan dar información para responder a estas preguntas. [44].

### **2.1.8 Manifiesto ágil**

El manifiesto ágil es un documento redactado en marzo de 2001 durante una reunión de 17 expertos en desarrollo de software. En esta reunión, se acuñó el término “Métodos Ágiles” para referirse a aquellos métodos que estaban surgiendo como alternativa a las metodologías formales. El manifiesto ágil propone 4 valores y 12 principios que guían el desarrollo ágil de software [2].

## **2.2. Estado del arte**

Con el fin de proporcionar una visión más clara del estado del arte relacionado con esta tesis, es decir, la evaluación de la agilidad en las organizaciones de desarrollo de software, en esta sección se presentan los resultados obtenidos a partir de la realización de un mapeo sistemático de la literatura acerca de la evaluación de la agilidad en las organizaciones, el cual fue presentado en [26]. Asimismo, se puede observar el estado del arte con relación a las ontologías sobre la evaluación de la agilidad de los procesos software, el cual fue presentado en [45].

### **2.2.1. Estado del arte acerca de la evaluación de la agilidad en las organizaciones**

Con el objetivo de obtener un estado del arte actualizado que sintetice los trabajos relacionados con la evaluación de la agilidad en las organizaciones, se realizó un mapeo sistemático de la literatura. Como resultado del mapeo, se encontraron 18 estudios, los cuales fueron analizados y comparados teniendo en cuenta el tipo de estudio, la relación de la evaluación con los principios y valores ágiles, los criterios de evaluación definidos, los métodos de validación utilizados, y las herramientas propuestas. Además, se determinaron los factores que influyen en la evaluación de la agilidad en las organizaciones. En el Anexo A, se presenta una descripción detallada del mapeo sistemático teniendo en cuenta: i) los trabajos relacionados, ii) la metodología empleada, iii) los resultados obtenidos, y iv) las brechas encontradas.

A partir del análisis de los resultados del mapeo sistemático, se puede observar que ya existen propuestas relacionadas con la evaluación de la agilidad, sin embargo, no existe consenso en la definición de agilidad en las organizaciones software y por lo tanto no hay un acuerdo en cuanto a los criterios que deben ser tenidos en cuenta como referencia para evaluar la agilidad. La mayoría de los estudios usan como referencia para la evaluación de la agilidad criterios que los mismos autores de los estudios definen lo que puede representar una visión subjetiva de la agilidad. Además, ninguno de los estudios presenta de forma clara y detallada la terminología relacionada con los enfoques ágiles y la evaluación de la agilidad, razón por la cual se dificulta la aplicación de los enfoques de evaluación por parte de las propias organizaciones debido a problemas de ambigüedad en los términos empleados en distintos enfoques ágiles y tradicionales.

También se puede observar que ninguna de las propuestas encontradas presenta de forma clara y detallada la relación entre los principios y valores ágiles y los elementos (actividades, roles y productos) de los procesos software. Debido a lo anterior, no es posible determinar el grado de cumplimiento del manifiesto ágil a partir de los resultados de la evaluación y se dificulta encontrar oportunidades de mejora detalladas a nivel de elementos de proceso que faciliten la interpretación de los resultados de la evaluación por parte de las organizaciones.

Con respecto a las herramientas software propuestas, se puede observar que están enfocadas en la evaluación de la agilidad de la organización, del equipo o del proyecto. No hay evidencia de la existencia de una herramienta software de evaluación de la agilidad de los procesos software de las organizaciones.

### **2.2.2. Estado del arte acerca de la ontologías sobre la evaluación de la agilidad de los procesos software**

En la revisión de la literatura, ha sido posible identificar pocos esfuerzos para desarrollar ontologías formales para apoyar la evaluación de la agilidad de los procesos de software. El trabajo realizado en este sentido se ha centrado principalmente en el desarrollo de ontologías para representar los elementos clave de dominios particulares, por ejemplo, ontologías para enfoques tradicionales y ágiles. Asimismo, ha sido posible encontrar algunos trabajos que proponen algunas taxonomías. A continuación, se presentan las propuestas identificadas:

- **Ontologías para enfoques tradicionales - OET:** Los trabajos relacionados con ontologías para procesos software han centrado su esfuerzo en los enfoques tradicionales. Algunas propuestas que se pueden destacar son: CMMI-SW [46], basada en la ontología sobre el conjunto de conocimientos propuesto en SWEBOK [47], ontologías para apoyar la implementación de CMMI e ISO/IEC 15504, como la ontología Software Process Ontology (SPO), la cual permite expresar procesos de software a nivel conceptual y proporciona una extensión para generar ontologías para modelos de proceso específicos [48], y la ontología para conceptos básicos unificados para modelos de capacidad de proceso [49], entre otras. Teniendo en cuenta que estas ontologías fueron definidas principalmente para representar estructuras de enfoques de calidad basadas en procesos, no tienen en cuenta algunos conceptos importantes en los enfoques ágiles, como: principio, valor, entre otros.
- **Ontologías para enfoques ágiles - OEA:** Con respecto a los enfoques ágiles, fue posible identificar algunas propuestas que desarrollan ontologías con el objetivo de apoyar el análisis de los conceptos y las relaciones en torno a los procesos relacionados con este tipo de enfoques. A continuación, se describen algunas propuestas encontradas: (i) ontología XPO (eXtreme Programming

Ontology) [24], que modela tres conceptos principales: rol organizacional, producto y fase, XPO también permite indexar documentos relevantes, artefactos XP y páginas Wiki, esto, para extraer y analizar procesos ágiles, la actividad de programadores y el contenido de los repositorios, (ii) Ontología basada en el ciclo de vida del enfoque FDD (Feature Driven Development) [50], esta ontología se puede utilizar en el desarrollo de modelos de aplicación y en el diseño e implementación de características en aplicaciones web semánticas, (iii) Ontologías basadas en Scrum, tales como: (a) la ontología K-CRIO [51], la cual es una conceptualización de Scrum que permite la descripción de los procesos de negocio que están dedicados al diseño de un producto a partir de los conceptos clave y las relaciones del proceso de desarrollo de software, además ilustra el uso de la ontología tomando como ejemplo el proceso de desarrollo Scrum, y (b) una ontología que permite generar evaluaciones del proceso Scrum en una organización de desarrollo de software con el fin de ayudar a los ingenieros de software a comprender el proceso de desarrollo que están siguiendo [52]. Finalmente, (iv) una ontología para apoyar el desarrollo ágil de software (OSDAS), basada en siete enfoques ágiles como: Agile Microsoft Solutions Framework (AMSF), Agile UP, Crystal Clear, Dynamic Systems Development Method (DSDM), eXtreme Programming (XP) y Scrum, con la cual el autor intentó representar de manera genérica la variada terminología usada en los enfoques ágiles [53].

- **Taxonomías:** También se encontraron algunas taxonomías, la primera, es una taxonomía de dependencia que permite identificar las dependencias en los proyectos ágiles de desarrollo de software, este trabajo estudia una de las características más importantes en proyectos ágiles, la colaboración y cómo lograrla mediante una coordinación efectiva usando el análisis de la dependencia entre proyectos [54]. Del mismo modo, se encontró un estudio que analiza el comportamiento de los gerentes de proyecto cuando adoptan enfoques ágiles en proyectos de desarrollo de software [55].

Después de analizar los trabajos relacionado, a continuación se presentan algunos aspectos asociados a las limitaciones de estas propuestas:

La mayoría de las soluciones propuestas son de naturaleza preliminar y están limitadas en su aplicación para apoyar un enfoque específico, es decir: las propuestas están enfocadas para ser utilizadas en dominios específicos como enfoques ágiles o enfoques tradicionales y no de manera integrada o híbrida. Es importante tener en cuenta que actualmente el movimiento ágil ha adquirido una gran importancia y ha comenzado a integrarse con otros enfoques heredados, siendo la gran mayoría tradicional, así como: las normas ISO/IEC 15504, ISO/IEC 29110 [56], o modelos de facto como CMMI, entre otros.

Si bien algunas propuestas representan los conceptos y las relaciones de algunos enfoques ágiles, es posible observar la ausencia de conceptos y relaciones de términos importantes que determinan la verdadera esencia de este tipo de enfoques, algunos de estos conceptos son: valores, principios, y prácticas ágiles. Además, las soluciones propuestas corresponden a soluciones conceptuales con términos de dominios específicos, los cuales tienen características muy particulares que hacen imposible soportar la instanciación de múltiples enfoques.

No se evidenció una solución ontológica que aliente y/o facilite la evaluación de la agilidad de los procesos de software. Aunque se han definido soluciones para abordar la evaluación de ciertos enfoques ágiles, estos no son genéricos y tampoco incorporan conceptos tales como valores y principios ágiles, aspecto fundamental a considerar para llevar a cabo la evaluación de la agilidad de los procesos de software de una organización independientemente del enfoque que se esté utilizando como modelo de referencia.

A partir de los resultados obtenidos en el análisis de los trabajos relacionados, ha sido posible identificar que se han desarrollado diferentes propuestas para apoyar diferentes dominios específicos a partir de la conceptualización de los términos/conceptos y las relaciones que los describen. Sin embargo, en los trabajos relacionados se utilizan diferentes términos/conceptos para identificar sus elementos, esto es normal, porque cada propuesta ha sido definida por diferentes autores o grupos de investigación. Por lo tanto, se requiere resolver algunas diferencias y conflictos terminológicos en el entorno de la definición de una solución ágil que permita apoyar la evaluación de agilidad.



## Capítulo 3

# AgilityPRO: Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones

En este capítulo, se presenta la solución propuesta en esta tesis de maestría, denominada: “Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones - AgilityPRO”. La primera sección presenta una descripción general de los componentes del framework y su organización. En las siguientes secciones, se describe en detalle cada uno de los cuatro componentes de AgilityPRO: i) una ontología para procesos de desarrollo ágil de software (sección 3.2), ii) un modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software (sección 3.3), iii) un método de evaluación de la agilidad de los procesos de desarrollo de software (sección 3.4), y iv) un prototipo de una aplicación web para soportar la evaluación (sección 3.5).

### 3.1. Componentes de AgilityPRO

El framework definido consta de 4 elementos organizados en 3 capas: i) una capa conceptual, la cual está constituida por una ontología del desarrollo ágil de software y un modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software, ii) una capa metodológica que consiste en un método de evaluación basado en el modelo de referencia de la capa conceptual, y iii) una capa tecnológica que consiste en un prototipo de una aplicación web para soportar la evaluación (ver Figura 3.1). A continuación se describe de manera general cada una de las capas y sus componentes.

### **3.1.1. Capa conceptual**

Constituye el soporte conceptual necesario para el funcionamiento de las demás capas. La capa conceptual se compone de una ontología y un modelo de referencia:

- Ontología para procesos de desarrollo ágil de software – OntoAgile, la cual es una ontología que permite entender y organizar el conocimiento relacionado con los términos y relaciones alrededor de los procesos software y los valores y principios del manifiesto ágil.
- Modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityRef, el cual define un conjunto de aspectos de agilidad que describen los elementos de proceso (actividades, roles, productos, herramientas) que evidencian la implementación de uno o más principios ágiles en un proceso de desarrollo de software de una organización. Los principios ágiles describen las características que debe tener un proceso ágil de desarrollo de software y soportan el cumplimiento de uno o más valores ágiles. AgilityRef usa los conceptos y relaciones identificadas en la ontología OntoAgile.

### **3.1.2. Capa metodológica**

Esta capa permite a las organizaciones conocer el grado de agilidad de sus procesos de desarrollo de software. Para conseguir este objetivo, esta capa incluye:

- Método de evaluación de la agilidad de los procesos de desarrollo de software – AgilityPval, el cual define un conjunto de indicadores a partir de los aspectos de agilidad definidos en AgilityRef. Este método de evaluación está basado en PvalCOMPETISOFT, el cual es un proceso que describe de manera detallada la forma en que se debe llevar a cabo la evaluación de procesos. Este proceso se adapta a cualquier instrumento de evaluación que requiera evaluar características específicas de los procesos de una organización [57].

### **3.1.3. Capa tecnológica**

Incluye un prototipo de una aplicación web llamada AgilityProTool, la cual soporta la evaluación de la agilidad de los procesos software de las organizaciones a través de una encuesta que incluye las preguntas definidas en AgilityPval. AgilityProTool

permite generar un reporte para conocer el grado de cumplimiento de los aspectos de agilidad definidos en AgilityRef.

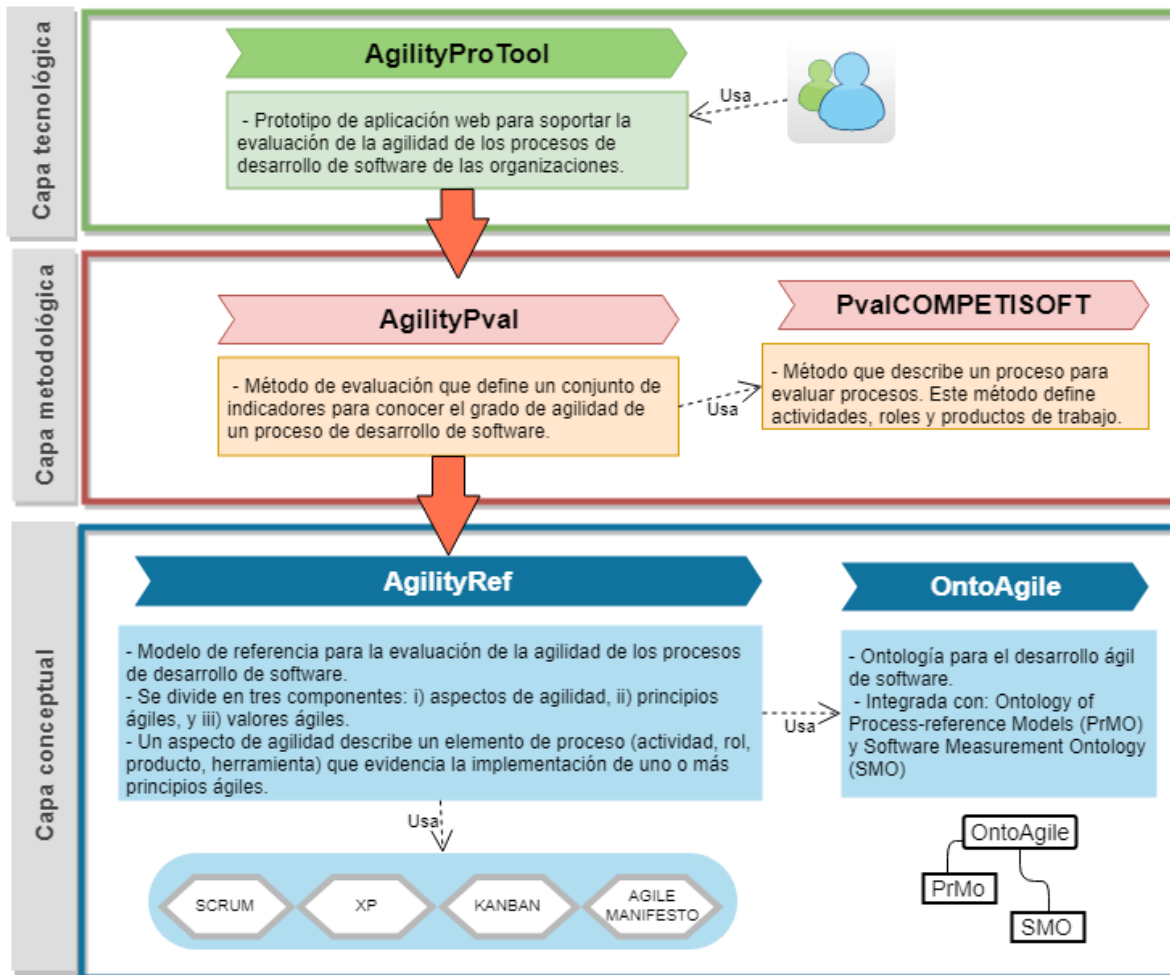


Figura 3.1 Vista general de los componentes de AgilityPRO.

### 3.2. OntoAgile: Una ontología para procesos de desarrollo ágil de software

En esta sección se presenta la ontología denominada OntoAgile, la cual es una ontología de dominio que permite organizar el conocimiento relacionado con la implementación de los enfoques ágiles en los procesos de desarrollo de software de las organizaciones de una manera genérica y formal, permitiendo que el conocimiento pueda ser reutilizado y compartido por los interesados en el área. Asimismo, OntoAgile soporta la evaluación de la agilidad de los procesos de desarrollo de software a partir de la identificación de las relaciones entre los

elementos de los procesos software y los principios y valores ágiles. OntoAgile ha sido publicada en [45].

Teniendo en cuenta el análisis del estado del arte relacionado con las ontologías existentes para soportar la evaluación de la agilidad de los procesos software (ver sección 2.2.2.), se establecieron los siguientes objetivos para la definición de OntoAgile: (i) ubicación e identificación de sinónimos, homónimos, inconsistencias y conflictos terminológicos e (ii) integración de los conceptos encontrados en la literatura analizada. A continuación, se describe OntoAgile en términos de: i) la metodología empleada para su definición, ii) su propósito, iii) las preguntas de competencia, iv) su integración con otras ontologías, v) su representación gráfica, vi) sus conceptos y relaciones, y vii) su evaluación.

### **3.2.1. Metodología**

Después de estudiar diferentes metodologías que pueden usarse para la definición de ontologías, se decidió utilizar la metodología REFSENO para el desarrollo de OntoAgile. Esta elección se basó en los siguientes criterios: (i) REFSENO es una adaptación de Methontology, la cual se utiliza ampliamente para definir ontologías en diferentes contextos, (ii) REFSENO ha sido especialmente diseñada para el desarrollo de ontologías en ingeniería de software a través de constructores que permiten definir conceptos, atributos y relaciones entre ellos, (iii) otras metodologías solo permiten representaciones complejas y menos intuitivas, lo cual dificulta su uso por parte de personas que no están familiarizadas con la lógica de predicados de primer orden o representaciones similares. En la sección 2.1.5 del marco teórico se se detalla un poco más acerca de este formalismo y de otros conceptos relacionados con la definición de ontologías.

### **3.2.2. Propósito**

El propósito de OntoAgile es permitir a los interesados en el desarrollo ágil de software conocer los conceptos y relaciones que deben estar presentes en los procesos ágiles de desarrollo, independientemente del enfoque particular que se esté adoptando. OntoAgile puede ser usada desde las siguientes perspectivas:

- **Académica**, debido a que los interesados en aprender o investigar acerca de este tema pueden usar la ontología para conocer los términos y relaciones que se usan en el desarrollo ágil de software.
- **Industrial**, ya que las organizaciones software pueden usar la ontología para instanciar sus procesos software y conocer en qué medida están presentes los conceptos y relaciones básicas que se dan en los contextos ágiles de desarrollo de software. Asimismo, puede ser utilizada como base para soportar la evaluación de la agilidad de una organización, así como herramienta conceptual para facilitar la armonización de múltiples modelos que deban ser implementados de manera integrada para la definición de soluciones de procesos híbridos.

### 3.2.3. Preguntas de competencia

OntoAgile permite dar respuesta a las preguntas de competencia (PC) presentadas en la Tabla 3.1.

Id.	Preguntas de competencia
PC1	¿Cuál es la relación entre los valores y los principios ágiles?
PC2	¿Qué tipos de enfoques de desarrollo de software están guiados por los valores ágiles?
PC3	¿Cuáles son los principales elementos que componen un proceso software?
PC4	¿Cuál es la relación entre las actividades del proceso y los principios ágiles?
PC5	¿Cuál es la relación entre los recursos usados en el proceso y los principios ágiles?
PC6	¿Cuál es la relación entre los productos generados en el proceso y los principios ágiles?

Acrónimos utilizados: Id: Identificador.

Tabla 3.1. Preguntas de competencia definidas para OntoAgile.

### 3.2.4. Integración de OntoAgile con otras ontologías

Teniendo en cuenta que OntoAgile soporta la evaluación de la agilidad de los procesos software de las organizaciones, fue necesario incluir conceptos relacionados con los procesos software y con la medición del software. En este sentido, se realizó un proceso de integración de OntoAgile con la ontología: Ontology of Process-reference Models (PrMO) [58], la cual establece y clarifica los elementos principales para expresar enfoques basados en procesos, específicamente en enfoques tradicionales, así como: SWEBOK [59], CMMI-DEV [60], ISO 27001 [61], ISO/IEC 15504, entre otros. Asimismo, se han utilizado algunos conceptos de la ontología Software Measurement Ontology (SMO) presentada en

[62]. SMO establece y aclara los elementos clave en la definición de una medida de software y la terminología relacionada con la acción de medir el software.

### 3.2.5. Representación gráfica

En la Figura 3.2 se puede observar la representación gráfica de OntoAgile utilizando UML (lenguaje de modelado unificado). Además, se muestran su relación con las ontologías PrMO y SMO.

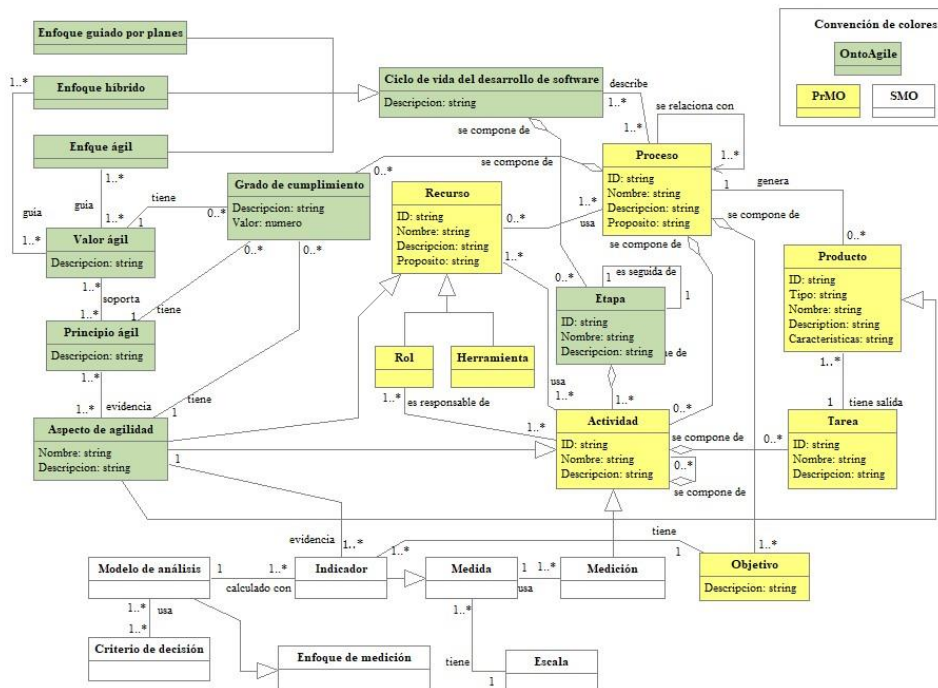


Figura 3.2 Representación gráfica de OntoAgile usando UML.

### 3.2.6. Conceptos y relaciones de OntoAgile

Las definiciones precisas de los conceptos incluidos en la ontología se presentan en la Tabla 3.2, la cual está ordenada alfabéticamente y organizada de la siguiente manera: la primera columna muestra el término que se describe, la segunda columna muestra el súper concepto, luego, la tercera columna muestra la definición del término en OntoAgile. La cuarta columna muestra la fuente usada para adaptar o adoptar el término. La Tabla 3.3, presenta las relaciones entre los conceptos presentados en la Tabla 3.2. Las descripciones presentadas en la cuarta columna de la Tabla 3.2 pueden ser de los siguientes tipos:

- **Tomado de [fuente]**, si el término ha sido definido en otro trabajo y es clave para la definición de OntoAgile. Además, el término no ha sido cambiado o adaptado.
- **Definido de [fuente]**, si el término se ha definido a partir de una fuente que no proporciona una definición particular, es decir, el término ha sido definido sin resaltar, cambiar o complementar un término existente, pero el trabajo presentado en él ha sido clave para establecer una definición
- **Citado en [fuente]**, si el término ha sido citado por una fuente y no es la fuente original. El término no ha sido modificado.
- **Fuente propia**, si el término ha sido definido en esta tesis o se le ha dado una definición particular para esta ontología.

Concepto	Súper-concepto	Definición	Fuente
Actividad	Concepto	Comprende un conjunto de tareas o acciones usadas para producir y mantener dispositivos, así como para lograr los objetivos del proceso. La actividad incluye los procedimientos, estándares, políticas y objetivos para crear y modificar un conjunto de productos de trabajo.	Tomado de [58].
Aspecto de agilidad	Recurso Producto Actividad	Corresponde a una actividad, producto o recurso que evidencia el cumplimiento de uno o más principios ágiles en un proceso de desarrollo de software.	Fuente propia.
Ciclo de vida del desarrollo de software	Concepto	Un ciclo de vida de desarrollo de software se refiere al marco que se utiliza para planificar, administrar y controlar el proceso de desarrollo de un sistema de información.	Citado en [63].
Criterio de decisión	Concepto	Umbrales, objetivos o patrones utilizados para determinar la necesidad de acción o investigación adicional, o para describir el nivel de confianza en un resultado dado.	Tomado de [62].
Enfoque ágil	Ciclo de vida del desarrollo de software	Enfoque de desarrollo de software basado en el desarrollo iterativo, adaptación e inspección frecuente, y entregas incrementales, en el que los requisitos y las soluciones evolucionan a través de la colaboración en equipos multifuncionales y a través de la retroalimentación continua de los participantes.	Definido de [64].
Enfoque de medición	Concepto	Secuencia de operaciones destinadas a determinar el valor de un resultado de medición.	Tomado de [62].
Enfoque híbrido	Ciclo de vida del desarrollo de software	Un enfoque híbrido es cualquier combinación de enfoques ágiles y tradicionales (Plan-Drive) que una unidad organizativa adopta y personaliza según las necesidades de su propio contexto.	Tomado de [65].
Enfoques guiados por	Ciclo de vida del	Enfoque de desarrollo de software que tiene las siguientes características: Los requisitos se especifican al inicio del	Citado en [66].

planes	desarrollo de software	proyecto; se define un plan detallado desde el inicio hasta el final del proyecto; los requisitos se especifican con gran detalle y luego se implementa un proceso de solicitud de cambio riguroso; la especificación de arquitectura y diseño debe estar completa antes de que comience la implementación; el trabajo de programación solo se concentra en la fase de programación; las pruebas se realizan al final del proyecto; la garantía de calidad se maneja de manera formal.	
Escala	Concepto	Un conjunto de valores con propiedades definidas.	Tomado de [62].
Etapas	Concepto	Período dentro del ciclo de vida de una entidad que se relaciona con el estado de su descripción o realización. Las etapas se relacionan con los principales hitos de progreso y logros de la entidad a lo largo de su ciclo de vida.	Tomado de [64].
Grado de agilidad	Concepto	Dato cuantitativo que representa el grado de cumplimiento de un aspecto de agilidad, valor ágil o principio ágil en un proceso de desarrollo de software.	Fuente propia.
Herramienta	Recurso	Las herramientas automatizan la ejecución de ciertas actividades.	Tomado de [58].
Indicador	Medida	Una medida que se deriva de otras medidas utilizando un modelo de análisis como enfoque de medición.	Tomado de [62].
Medición	Concepto	Un conjunto de operaciones que tienen el objeto de determinar el valor de un resultado de medición, para un atributo dado de una entidad, utilizando un enfoque de medición.	Tomado de [62].
Medida	Concepto	El enfoque de medición definido y la escala de medición.	Tomado de [62].
Modelo de análisis	Enfoque de medición	Algoritmo o cálculo que combina una o más medidas con criterios de decisión asociados.	Tomado de [62].
Objetivo	Concepto	El fin que se ha definido para el proceso de desarrollo de software. Debe ser alcanzable y cuantificable.	Fuente propia.
Principio ágil	Concepto	Lineamiento que se considera deseable en el desarrollo ágil y que guía las acciones del equipo de desarrollo. Cada principio ágil se deriva de uno o varios valores ágiles.	Definido de [2].
Proceso	Concepto	Conjunto coherente de políticas, estructuras organizativas, tecnologías; procedimientos, propósitos, objetivos y productos de trabajo necesarios para diseñar, desarrollar, implementar y mantener un producto de software.	Tomado de [58].
Producto	Concepto	El conjunto de artefactos que se desarrollarán, entregarán y mantendrán en un proyecto se llama producto. Los productos pueden ser de entrada o salida; obligatorio u opcional En la mayoría de los casos, los productos son	Tomado de [58].



		artefactos tangibles consumidos, producidos o modificados por tareas.	
Recurso	Concepto	Un recurso es un activo que una empresa necesita. En el campo de la ingeniería de software, hay dos recursos de principal importancia: los desarrolladores y las herramientas.	Tomado de [58].
Rol	Concepto	Describe un conjunto o grupo de responsabilidades, deberes y habilidades requeridas para realizar una actividad específica.	Tomado de [58].
Tarea	Concepto	Elemento de proceso que define el trabajo realizado por roles. Una tarea está asociada con los productos de entrada y salida.	Tomado de [58].
Unidad de medición	Concepto	Cantidad particular, definida y adoptada por convención, con la cual se comparan otras cantidades del mismo tipo para expresar su magnitud en relación con esa cantidad.	Tomado de [62].
Valor ágil	Concepto	Valor que debe estar presente en los integrantes de un equipo ágil de desarrollo. El manifiesto ágil define cuatro valores ágiles.	Definido de [2].

Tabla 3.2. Glosario de conceptos de OntoAgile.

Nombre	Conceptos	Descripción
calculado con	Indicador - Modelo de análisis	Cada indicador se calcula con un modelo de análisis. Cada modelo de análisis puede definir uno o más indicadores.
Describe	Ciclo de vida de desarrollo de software - Proceso	Un ciclo de vida de desarrollo de software describe uno o más procesos. Un proceso es descrito por uno o más ciclos de vida de desarrollo de software.
es responsable de	Rol - Actividad	Un rol es responsable de una o muchas actividades. Una actividad tiene al menos un responsable.
es seguida de	Etapas - Etapas	Una etapa puede ser seguida por otra etapa.
evidencia	Indicador - Aspecto de agilidad	Un indicador evidencia el cumplimiento de un aspecto de agilidad. La evidencia del cumplimiento de un aspecto de agilidad se da a través de uno o varios indicadores.
evidencia	Aspecto de agilidad - Principio ágil	Un aspecto de agilidad evidencia el cumplimiento de uno o varios principios ágiles. La evidencia del cumplimiento de un principio ágil se da a través del cumplimiento de uno o varios aspectos de agilidad.
Genera	Proceso - Producto	Un proceso puede generar cero o muchos productos. Un producto es generado por un proceso.
Guía	Valor ágil - enfoque ágil	Un valor ágil guía uno o más enfoques ágiles. Un enfoque ágil se guía por uno o más valores ágiles.
Guía	Valor ágil - enfoque híbrido	Un valor ágil guía uno o más enfoques híbridos. Un enfoque híbrido se guía por uno o más valores ágiles.
se compone de	Ciclo de vida de	Un ciclo de vida de desarrollo de software se compone de

	desarrollo de software - Etapa	etapas. Una etapa es parte de un ciclo de vida de desarrollo de software.
se compone de	Etapa - Actividad	Una etapa se compone de una o más actividades. Una actividad es parte de una etapa.
se compone de	Proceso - Actividad	Un proceso se compone de una o más actividades. Una actividad es parte de un proceso.
se compone de	Actividad - Tarea	Una actividad está compuesta de cero o muchas tareas. Una tarea es parte de una actividad.
se compone de	Actividad - Actividad	Una actividad está compuesta por cero o muchas actividades.
se compone de	Proceso – Grado de cumplimiento	Un proceso se compone de cero o muchos grados de cumplimiento. Un grado de cumplimiento es parte de un proceso.
se compone de	Proceso – Objetivo	Un proceso se compone de uno o muchos objetivos. Un objetivo es parte de un proceso.
se relaciona con	Proceso - Proceso	Un proceso puede estar relacionado con otro proceso.
Soporta	Principio ágil - valor ágil	Un principio ágil soporta uno o más valores ágiles. Un valor ágil está soportado por uno o más principios ágiles.
Tiene	Medida – Escala	Cada medida tiene una escala. Una escala puede servir para definir más de una medida.
Tiene	Objetivo - Indicador	Un objetivo tiene uno o muchos indicadores. Un indicador pertenece a un objetivo.
Tiene	Aspecto de agilidad – Grado de cumplimiento	Un aspecto de agilidad tiene cero o muchos grados de cumplimiento. Un grado de cumplimiento pertenece a un aspecto de agilidad.
Tiene	Principio ágil – Grado de cumplimiento	Un principio ágil tiene cero o muchos grados de cumplimiento. Un grado de cumplimiento pertenece a un principio ágil.
Tiene	Valor ágil – Grado de cumplimiento	Un valor ágil tiene cero o muchos grados de cumplimiento. Un grado de cumplimiento pertenece a un valor ágil.
tiene como salida	Tarea - Producto	Una tarea tiene como salida uno o muchos productos. Un producto es una salida de una tarea.
Usa	Proceso - Recurso	Un proceso usa cero o muchos recursos. Un recurso es utilizado por uno o más procesos.
Usa	Actividad - Recurso	Una actividad usa uno o más recursos. Un recurso es utilizado por una o más actividades.
Usa	Modelo de análisis - Criterio de decisión	Un modelo de análisis utiliza uno o más criterios de decisión. Cada criterio de decisión se utiliza en uno o más modelos de análisis.
Usa	Medición – Medida	Cada medición usa una medida. Una medida puede usarse en varias mediciones.

Tabla 3.3. Relaciones de OntoAgile.

### 3.2.7. Discusión de algunos conceptos sus relaciones

Algunos de los términos utilizados en OntoAgile merecen una discusión especial, a continuación, se amplía la definición y el análisis de algunos términos como: ciclo de vida de desarrollo de software, valor ágil e indicador.

- Un ciclo de vida de desarrollo de software (CVDS) describe un enfoque para diseñar, construir y mantener software. Existe una amplia variedad de CVDS como cascada, espiral, incremental, proceso unificado de Rational (RUP), desarrollo de software ágil, entre otros. Sin embargo, en términos generales, todos los CVDS consisten en una secuencia de pasos que los desarrolladores de software deben seguir para obtener y entregar un producto de software [63]. Los CVDS se pueden clasificar en enfoques guiados en planes, enfoques híbridos y enfoques ágiles.
- Los valores ágiles se definieron en el manifiesto ágil para resumir los valores que deberían estar presentes en los equipos ágiles de desarrollo de software. Los cuatro valores ágiles son: i) individuos e interacciones sobre procesos y herramientas, ii) software funcionando sobre documentación extensiva, iii) colaboración con el cliente sobre negociación contractual, y iv) respuesta al cambio sobre seguir un plan. Los cuatro valores están soportados por doce principios ágiles que se describen y pueden consultarse en [2].
- Un indicador es una medida derivada de una o más medidas que permite conocer el grado de cumplimiento de un objetivo. En el caso de la evaluación de la agilidad, se pueden definir indicadores relacionados con cada uno de los principios ágiles y, de este modo, conocer el grado de agilidad del proceso de desarrollo de software basado en el análisis del cumplimiento de cada uno de los principios ágiles. Los indicadores pueden verse afectados por los elementos del proceso (productos, roles, actividades y herramientas).

### 3.2.8. Evaluación de OntoAgile

Según [67], la evaluación de ontologías debe centrarse principalmente en la evaluación de tres elementos relacionados con el rendimiento y la correcta definición: consistencia, integridad y concisión. Para la evaluación inicial de OntoAgile, se realizaron algunas entrevistas con expertos en enfoques ágiles y especialistas en ingeniería de software, esto, con el fin de conocer su opinión acerca de los conceptos definidos en la ontología. Como resultado de las entrevistas, surgieron algunas

sugerencias que permitieron actualizar los conceptos, atributos y relaciones. Una vez revisados y actualizados los conceptos y relaciones de la ontología, se implementó OntoAgile en el lenguaje de ontologías OWL [68] utilizando Protégé [69]. Se decidió usar el idioma inglés para implementar los conceptos y relaciones con el fin de facilitar su utilización por parte de la comunidad académica e industrial. Con OntoAgile implementada en OWL (disponible en: <http://bit.ly/owl-ontoagile>), se usó el razonador Hermit [70] para evaluar la consistencia de la estructura de la ontología. Una vez que se completó el análisis, fue posible verificar que OntoAgile no presenta inconsistencias en su estructura. Posteriormente, OntoAgile se utilizó con éxito en tres casos de implementación. El primer caso de implementación consistió en la instanciación de OntoAgile con los elementos principales del enfoque ágil de Scrum. En el segundo caso, la ontología se utilizó para soportar la evaluación de la agilidad basada en el cumplimiento de los principios ágiles. El tercer caso de implementación correspondió al desarrollo de una aplicación web para realizar consultas a OntoAgile utilizando el Protocolo y lenguaje de consulta RDF (SPARQL) para visualizar la relación entre valores y principios ágiles. Además, se evaluó la capacidad de OntoAgile para responder las preguntas de competencia utilizando SPARQL. Finalmente, se comparó la ontología propuesta con las ontologías existentes para enfoques ágiles con el fin de analizar su contribución a la comunidad académica e industrial.

- **Soportando la instanciación de Scrum:** El primer caso de implementación consistió en la instanciación de los elementos principales de Scrum como roles, reuniones, artefactos, entre otros. Para la instanciación de estos elementos, se utilizaron los siguientes conceptos definidos en OntoAgile: Proceso, Etapa, Actividad, Tarea, Herramienta, Rol y Producto. La Figura 3.3 muestra un extracto de la instanciación de los principales elementos de Scrum usando OntoAgile, donde se puede observar que un proceso de software guiado por Scrum se compone de las etapas: Inicio, Sprints y Release. A su vez, la etapa de Sprints se compone de las etapas: Estimación del plan, Implementación y retrospectiva. Las etapas están formadas por actividades, como en el caso de la actividad Crear historias de usuarios que forma parte de la etapa Estimación del plan. En esta actividad, la herramienta Entrevistas con el cliente se utiliza para conocer las necesidades del cliente para llevar a cabo las tareas Escribir historias de usuario

y Escribir criterios de aceptación. Como resultado de la tarea Escribir historias de usuario, obtiene el producto Historias de usuario, que es responsabilidad del rol Product Owner.

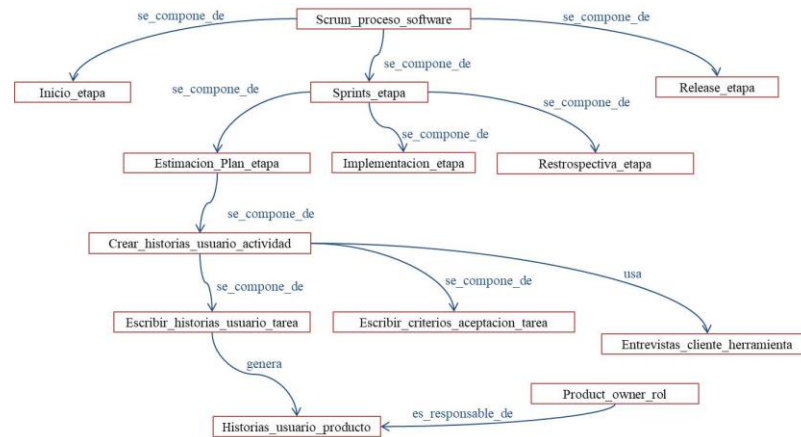


Figura 3.3 Extracto de instancia de Scrum en OntoAgile.

- Soportando la evaluación de la agilidad de los procesos software:** En OntoAgile se pudo establecer que los elementos de proceso (actividades, productos, roles y herramientas) permiten implementar los principios ágiles y a su vez dar soporte al cumplimiento de los valores ágiles. Esta relación entre elementos de proceso y principios ágiles se ve reflejada a través de indicadores. La Figura 3.4 presenta la instancia en OntoAgile de algunas relaciones y conceptos alrededor de la evaluación de la agilidad de los procesos software como: indicadores, principios ágiles y valores ágiles. Para realizar la instancia, se tomó como ejemplo un indicador denominado “Reuniones diarias”, el cual representa la reunión que debe llevar a cabo un equipo para evaluar su progreso con relación al objetivo de una iteración. Se espera que al realizar una reunión diaria de seguimiento, se influya positivamente en el cumplimiento de los principios ágiles seis y doce, los cuales están relacionados con la conversación cara a cara como el método más eficiente para comunicar información al equipo, y la reflexión a intervalos regulares con el fin de realizar ajustes para aumentar la efectividad. A su vez, esos dos principios ágiles brindan soporte al cumplimiento del primer valor ágil, en el cual se pone de manifiesto que en un proceso ágil de desarrollo de software son más importantes los individuos y sus interacciones que los procesos y herramientas. En los siguientes capítulos se describe con más detalle el proceso de evaluación de la agilidad de los procesos

de desarrollo de software basado en los conceptos y relaciones descritas en OntoAgile.

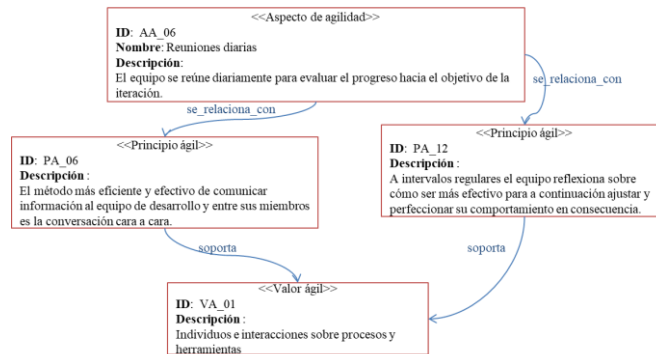


Figura 3.4 Instanciación de OntoAgile para soportar la evaluación de la agilidad.

- **Soportando la consulta a OntoAgile a través de una aplicación web:** El tercer caso de implementación está relacionado con el desarrollo de una aplicación web que envía consultas SPARQL a OntoAgile con el fin de visualizar los valores y principios ágiles. En la Figura 3.5 se puede ver que la página inicial de la aplicación muestra los cuatro valores ágiles en forma de enlaces que permiten navegar a una página que muestra los principios ágiles que sirven de soporte para cada valor. El desarrollo de esta aplicación web permitió establecer una arquitectura que sirve como punto de partida para agregar nuevas consultas a OntoAgile, como visualizar los indicadores de agilidad asociados con cada principio para guiar la evaluación de la agilidad de los procesos de software.

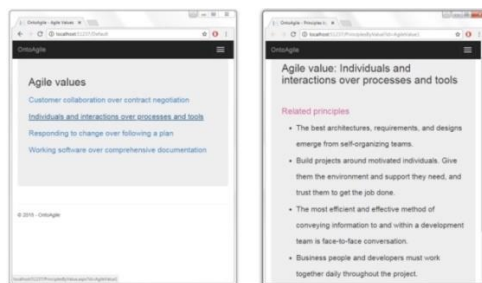


Figura 3.5 Ejemplo de una aplicación web que realiza consultas a OntoAgile.

- **Respondiendo las preguntas de competencia usando SPARQL:** Para evaluar el cumplimiento del propósito de la ontología, se realizó la traducción de cada una de las preguntas de competencia expresadas en lenguaje natural a consultas

SPARQL. Como resultado de la ejecución de las consultas, se comprobó que OntoAgile puede responder a todas las preguntas de competencia propuestas en la etapa inicial de su desarrollo. En la Tabla 3.4 se pueden observar las preguntas de competencia expresadas en SPARQL y el resultado obtenido.

PC	Consulta SPARQL	Resultado
PC1	SELECT ?e1 ?rel WHERE { ?rel rdfs:domain ?e1 . ?rel rdfs:range ontoagile:Agile_principle . FILTER (?e1 IN (ontoagile:Agile_value )) }	Agile_value isSupportedByPrinciple
PC2	SELECT ?Approach WHERE { ?r rdfs:domain ontoagile:Agile_value . ?r rdfs:range ?Approach . ?Approach rdfs:subClassOf ontoagile:Software_development_life_cycle }	Hybrid_approach Agile_approach
PC3	SELECT ?ProcessElement WHERE { ?r rdfs:domain ontoagile:Process . ?r rdfs:range ?ProcessElement }	Resource - Activity Product – Stage
PC4	SELECT ?e1 ?rel1 ?e2 ?rel2 WHERE { ?rel1 rdfs:domain ?e1 . ?rel1 rdfs:range ?e2 . ?rel2 rdfs:domain ?e2 . ?rel2 rdfs:range ontoagile:Activity . FILTER (?e1 IN (ontoagile:Agile_principle )) }	Agile_principle isRelatedToIndicator Indicator isAffectedByActivity
PC5	SELECT ?e1 ?rel1 ?e2 ?rel2 WHERE { ?rel1 rdfs:domain ?e1 . ?rel1 rdfs:range ?e2 . ?rel2 rdfs:domain ?e2 . ?rel2 rdfs:range ontoagile:Resource . FILTER (?e1 IN (ontoagile:Agile_principle )) }	Agile_principle isRelatedToIndicator Indicator isAffectedByResource
PC6	SELECT ?e1 ?rel1 ?e2 ?rel2 WHERE { ?rel1 rdfs:domain ?e1 . ?rel1 rdfs:range ?e2 . ?rel2 rdfs:domain ?e2 . ?rel2 rdfs:range ontoagile:Product . FILTER (?e1 IN (ontoagile:Agile_principle )) }	Agile_principle isRelatedToIndicator Indicator isAffectedByProduct

Tabla 3.4. Preguntas de competencia expresadas en SPARQL.

- Comparación de ontologías para enfoques ágiles:** En la Tabla 3.5, se presenta la comparación entre OntoAgile y las ontologías existentes para enfoques ágiles encontradas en el estado del arte. Para llevar a cabo la comparación, se utilizaron los siguientes criterios: C1) la ontología es independiente de un enfoque ágil particular, C2) se incluyen los conceptos del manifiesto ágil, C3) integración con las ontologías existentes, C4) se definen las preguntas de competencia, C5) la ontología está integrada con herramientas software, C6) se usa un lenguaje formal para representar la ontología y C7) la ontología está disponible en un repositorio público.

Según la comparación, se puede observar que, aunque existen ontologías que permiten instanciar procesos ágiles de desarrollo de software, no hay evidencia de una evaluación formal que garantice el cumplimiento de su propósito

académico o industrial. Además, las ontologías existentes no están disponibles en repositorios públicos que faciliten su uso y evaluación.

Ontología	C1	C2	C3	C4	C5	C6	C7
XPO [24]						✓	
FDD [50]						✓	
K-CRIO [51]	✓		✓			✓	
OSDAS [53]	✓	✓					
OntoAgile [45]	✓	✓	✓	✓	✓	✓	✓

Tabla 3.5. Comparación de ontologías existentes para enfoques ágiles.

### 3.3. Modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityRef

En esta sección, se presenta AgilityRef, el cual es un modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones, el cual está dividido en tres componentes principales: i) aspectos de agilidad, ii) principios ágiles, y iii) valores ágiles. Un aspecto de agilidad describe un elemento de proceso (actividad, rol, producto, herramienta) que evidencia la implementación de uno o más principios ágiles. Por su parte, los principios ágiles describen las características que debe tener un proceso ágil de desarrollo de software y soportan el cumplimiento de uno o más valores ágiles en el proceso de desarrollo (ver Figura 3.6). En AgilityRef se considera que la agilidad de un proceso de desarrollo de software consiste en la implementación de elementos de proceso que aporten al cumplimiento de los principios ágiles.

Es importante resaltar que en este documento se presenta la segunda versión de AgilityRef, la cual se obtuvo luego de la realización de acciones de mejora identificadas en el grupo focal realizado para la evaluar la primera versión de la propuesta. Los resultados del grupo focal se pueden consultar en la sección 4.1, y en el Anexo B se puede observar la primera versión de AgilityRef.



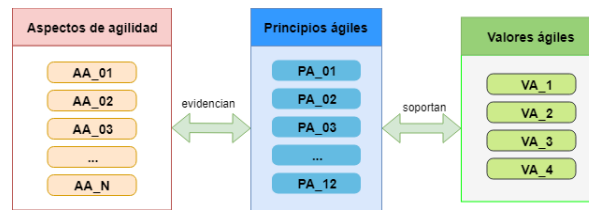


Figura 3.6 Componentes de AgilityRef.

Los componentes y relaciones de AgilityRef surgieron a partir de la creación de la ontología para procesos de desarrollo ágil de software – OntoAgile, la cual proporciona conceptos, definiciones y relaciones alrededor de los enfoques ágiles y los procesos de desarrollo de software. Como se mencionó en la sección 3.2.8, en la definición de OntoAgile se pudo establecer que para evaluar la agilidad de un proceso de desarrollo de software basándose en el cumplimiento de los principios y los valores ágiles, es necesario definir un conjunto de indicadores que permitan conocer de qué manera los elementos de proceso (actividades, productos, roles y herramientas) permiten implementar los principios ágiles y a su vez dar soporte al cumplimiento de los valores ágiles. Como un paso anterior a la definición de los indicadores, en AgilityRef se define un conjunto de aspectos de agilidad que describen de manera general los elementos de procesos que deben estar presentes en un proceso ágil de desarrollo. A partir de los aspectos de agilidad, se definieron los indicadores en el método de evaluación (capa metodológica). La separación entre aspectos de agilidad e indicadores tiene como intención hacer que el framework sea más flexible, permitiendo adaptar o crear indicadores de acuerdo a las características y necesidades de las organizaciones. En las siguientes secciones, se describirán en detalle los valores ágiles, principios ágiles y aspectos de agilidad.

### 3.3.1. Valores ágiles

Los valores ágiles representan los principales atributos que debe tener un proceso de desarrollo de software para ser considerado ágil según lo declarado en el manifiesto ágil [2]. En la Tabla 3.6 se pueden observar cada uno de los valores ágiles, un identificador para facilitar su referenciación en este documento y la interpretación de dichos valores en AgilityRef, la cual resulta del análisis de los valores descritos en el manifiesto ágil.

Id.	Valor ágil	Interpretación en AgilityRef
VA_1	Individuos e	Los procesos ágiles deben estar centrados en las personas y no en

	interacciones sobre procesos y herramientas.	los procesos o herramientas. No se puede negar la importancia de tener procesos que guíen el trabajo y herramientas que mejoren su eficiencia, pero es mucho más importante contar con un equipo de personas motivadas, con excelentes habilidades técnicas, comunicativas y de autoorganización que les permitan reaccionar rápidamente al cambio, interactuar efectivamente con el cliente y sus compañeros de equipo, y generar productos que entreguen valor al cliente. Los procesos y herramientas deben adaptarse al equipo y no al contrario.
VA_2	Software funcionando sobre documentación extensiva.	Aunque la documentación generada durante un proceso de desarrollo de software es importante, el principal objetivo del proceso debe ser entregar al cliente de manera iterativa, incrementos del producto que funcionen y aporten valor al negocio. Cada una de las entregas de incrementos del producto, debe ir acompañada de la documentación mínima necesaria y además debe superar un conjunto de criterios mínimos de aceptación por parte del cliente.
VA_3	Colaboración con el cliente sobre negociación contractual.	Un proceso de desarrollo ágil requiere un cliente que colabore frecuentemente con el equipo durante el desarrollo del producto. Esta colaboración permitirá priorizar las funcionalidades según el valor que aportarán al negocio y detectar riesgos de manera temprana. De ser posible, se debe contar con el cliente en el sitio de desarrollo del producto para que la comunicación sea más efectiva, pero en caso de que esto no sea posible, se debe contar con un representante del cliente que tenga poder de decisión.
VA_4	Respuesta ante el cambio sobre seguir un plan.	A diferencia de los enfoques tradicionales de desarrollo donde el objetivo está centrado en tener un plan detallado desde el comienzo, en un proceso de desarrollo ágil se planifica solamente lo necesario para iniciar el desarrollo del producto. A medida que se avanza en cada iteración, los cambios detectados a partir de la comunicación con el cliente, se ven como una oportunidad para entregar un producto que aportará más valor.

Acrónimos utilizados: Id: Identificador, VA: Valor ágil.

Tabla 3.6. Interpretación de los valores ágiles en AgilityRef.

### 3.3.2. Principios ágiles

Los doce principios ágiles fueron definidos en el manifiesto ágil [2] y aportan lineamientos generales para que un proceso de desarrollo de software esté alineado con los valores ágiles. A continuación, en la Tabla 3.7 se presentan los doce principios ágiles.

Id.	Principio ágil
PA_01	Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
PA_02	Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
PA_03	Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
PA_04	Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
PA_05	Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
PA_06	El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
PA_07	El software funcionando es la medida principal de progreso.
PA_08	Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
PA_09	La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
PA_10	La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
PA_11	Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
PA_12	A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Acrónimos utilizados: Id: Identificador.

Tabla 3.7. Principios ágiles en AgilityRef.

La relación entre principios y valores ágiles propuesta en AgilityRef se muestra en la Figura 3.7, en la cual se puede observar que la mayoría de principios ágiles están relacionados con el valor VA\_01, con un total de 7 principios, confirmando que un proceso de desarrollo ágil está centrado en las personas. El segundo valor con más principios relacionados es el VA\_02, con un total de 3 principios, lo cual refleja la importancia que tiene para un proceso ágil obtener como salida un producto que aporte valor al cliente. Por último, los valores VA\_3 y VA\_4 tienen una relación con un principio cada uno, relacionados con PA\_04 y PA\_02, respectivamente.

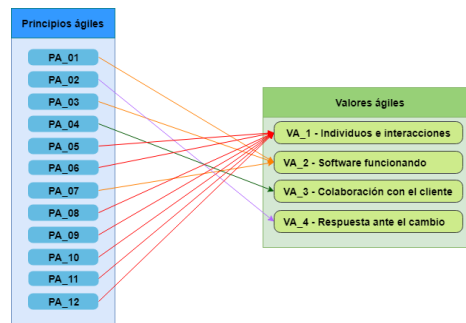


Figura 3.7 Relaciones entre principios y valores ágiles en AgilityRef.

### 3.3.3. Aspectos de agilidad

Los aspectos de agilidad corresponden a un conjunto de actividades, roles, productos y herramientas que evidencian el cumplimiento de los principios ágiles en un proceso de desarrollo de software. Para la definición de los aspectos, se analizaron actividades, roles, productos y herramientas propuestas en enfoques ágiles como: Scrum [71], XP [23] y Kanban [72], y que se relacionan directamente con uno o más principios ágiles. La selección de estos enfoques se realizó teniendo en cuenta que son los más usados en la industria de software actual según los resultados de la encuesta anual sobre agilidad “State of Agile” [73]. En los resultados de la encuesta se destaca que Scrum es el enfoque más usado por las organizaciones consultadas, esto se refleja con un uso del 56% de las personas encuestadas, seguido de la combinación de Scrum con Kanban (ScrumBan) con 8% y la combinación de Scrum y XP con 6%. Es importante destacar que los aspectos describen el “que” y no el “cómo” debido a que las organizaciones que implementan enfoques ágiles deben definir sus procesos de acuerdo a su contexto particular, pero teniendo como pilares los principios y valores ágiles. En la Tabla 3.8 se presentan los aspectos de agilidad que se proponen en AgilityRef, su respectivo identificador, nombre, descripción, fuente, tipo de elemento de proceso que representa, y principios ágiles con los cuales se relaciona. En la Tabla 3.9 se pueden observar los aspectos de agilidad y su relación con los principios ágiles.

<b>Id.</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Fuente</b>	<b>Tipo</b>	<b>PA.s.</b>
AA_01	Iterativo e incremental	El producto se crea de forma iterativa e incremental. Los intervalos tienen una duración máxima de un mes. Además, cada nueva iteración comienza inmediatamente después de la finalización de la iteración anterior.	Scrum, XP, Kanban	Actividad	PA_01 PA_03
AA_02	Incrementos con valor	Cada iteración genera una versión potencialmente útil y funcional del producto.	Scrum	Producto	PA_01 PA_03 PA_07
AA_03	Alcance negociable	El alcance de cada iteración puede clarificarse y renegociarse entre el equipo y el cliente.	Scrum, XP	Actividad	PA_02 PA_04
AA_04	Reunión diaria	El equipo se reúne diariamente para evaluar el progreso hacia el objetivo de la iteración. La duración de la reunión diaria se limita a un tiempo que permita a cada participante compartir de manera concisa lo que hizo el día anterior, lo que tiene planeado hacer ese día y los impedimentos para lograr el objetivo de la iteración.	Scrum, Kanban	Actividad	PA_06 PA_12
AA_05	Refinamiento de requisitos	La lista de requisitos del producto se actualiza constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Además, los requisitos pueden ser actualizados incluyendo más detalle, estimaciones y prioridades.	Scrum	Actividad	PA_01 PA_02 PA_10
AA_06	Autoorganización del equipo	Cada integrante del equipo puede elegir la mejor forma de llevar a cabo las tareas que tenga bajo su responsabilidad. El equipo tiene la posibilidad de dar mayor prioridad a los requisitos del producto que presenten más riesgo.	Scrum, XP	Rol	PA_05 PA_11
AA_07	Responsabilidad compartida	Los miembros del equipo pueden tener habilidades especializadas en áreas en las que estén más enfocados, pero la responsabilidad del producto es de todo el equipo. Cualquier miembro del equipo podría realizar cambios en cualquiera de los componentes del producto.	Scrum, XP	Rol	PA_05

AA_08	Ritmo constante	El tiempo de duración de las iteraciones es consistente durante todo el desarrollo del producto.	Scrum	Actividad	PA_08
AA_09	Iteraciones negociables	Una iteración se puede cancelar cuando se determina que su objetivo ha quedado obsoleto. Por ejemplo, si las condiciones del mercado o de la tecnología cambian.	Scrum	Actividad	PA_02
AA_10	Retrospectivas	Al finalizar cada iteración, el equipo y los interesados se reúnen para revisar los resultados con respecto a las personas, procesos y herramientas. Se identifican y ordenan los elementos más importantes que salieron bien y las posibles mejoras, se crea un plan para implementar las mejoras, y se adapta la lista de requisitos del producto en caso de ser necesario.	Scrum	Actividad	PA_04 PA_06 PA_09 PA_12
AA_11	Eliminación de impedimentos	Se tiene por práctica la asignación de un rol responsable de ayudar al equipo y a los interesados a ejecutar las actividades de la mejor manera y eliminar los impedimentos que se presenten.	Scrum, XP	Rol	PA_05
AA_12	Participación del cliente	Los clientes pueden decidir sobre qué partes del problema se deben resolver con mayor prioridad para que el equipo pueda enfocarse en aquello que entregue valor. Los clientes participan en la creación de las pruebas funcionales del producto y en la elección de la tecnología a emplear para su desarrollo. Un cliente real está disponible en la misma ubicación del equipo para responder preguntas, resolver discusiones y fijar prioridades a pequeña escala. En caso de que no sea posible contar con el cliente real, se debe contar con un representante del cliente y de los interesados en el producto con capacidad para tomar decisiones.	Scrum, XP	Rol	PA_01 PA_04 PA_07
AA_13	Estimación en equipo	El equipo es el encargado de estimar el tiempo que se empleará para implementar una característica del producto	XP	Rol	PA_05 PA_11
AA_14	Retroalimentación al cliente	Cuando el equipo tome decisiones técnicas que puedan tener impacto a nivel de negocio, debe informar a los interesados.	XP	Rol	PA_04
AA_15	Visión	Se define un vocabulario y una visión del producto a desarrollar que se	XP	Actividad	PA_04

	compartida	comparte por el equipo y los clientes.			
AA_16	Atención al buen diseño	Durante el diseño del producto se tiene en cuenta que no se dupliquen componentes, se incluyen todos los aspectos que el equipo considera importantes y se incluyen solamente lo que se requiere en el momento. Cada vez que se agrega una nueva característica al producto, se verifica si se puede modificar su estructura interna para hacerla más sencilla, en caso de ser posible, se realizan los cambios y se verifica que todas las pruebas funcionen.	XP	Actividad	PA_09 PA_10
AA_17	Trabajo colaborativo	Cuando un miembro del equipo va a implementar una nueva característica o va a realizar un cambio en el producto, puede pedir la colaboración a otro miembro del equipo para trabajar en conjunto.	XP	Actividad	PA_11
AA_18	Integración continua	Todos los cambios que se hacen al producto se prueban e integran después de unas pocas horas.	XP	Actividad	PA_01 PA_03 PA_07
AA_19	Reglas técnicas	Se acuerda un conjunto de reglas que todo el equipo debe cumplir al momento de agregar funcionalidades al producto o realizar modificaciones.	XP	Actividad	PA_09
AA_20	Uso de métricas	Se usan métricas que aporten información relevante sobre el desempeño del equipo.	XP, Kanban	Actividad	PA_08 PA_12
AA_21	Herramientas de integración continua	Se usan herramientas que permitan dar soporte a la integración continua de las nuevas funcionalidades y cambios realizados en el producto	XP	Herramienta	PA_01 PA_03 PA_07 PA_10
AA_22	Pruebas automáticas	Se usan herramientas que permitan dar soporte a la ejecución de pruebas automáticas	XP	Herramienta	PA_01 PA_03 PA_07 PA_10

Acrónimos utilizados: Id: Identificador, PAs: Principios Ágiles relacionados

Tabla 3.8. Aspectos de agilidad en AgilityRef.

		Aspectos de agilidad																						
		AA_01	AA_02	AA_03	AA_04	AA_05	AA_06	AA_07	AA_08	AA_09	AA_10	AA_11	AA_12	AA_13	AA_14	AA_15	AA_16	AA_17	AA_18	AA_19	AA_20	AA_21	AA_22	
Principios ágiles	PA_01	✓	✓			✓						✓						✓			✓	✓		
	PA_02			✓		✓				✓														
	PA_03	✓	✓																✓			✓	✓	
	PA_04			✓							✓		✓		✓	✓								
	PA_05						✓	✓				✓		✓										
	PA_06				✓						✓													
	PA_07		✓																✓			✓	✓	
	PA_08								✓													✓		
	PA_09										✓						✓			✓				
	PA_10					✓											✓					✓	✓	
	PA_11						✓							✓					✓					
	PA_12				✓						✓											✓		

Tabla 3.9. Relación entre principios ágiles y aspectos de agilidad.

### 3.4. Método de evaluación de la agilidad de los procesos de desarrollo de software – AgilityPval

En esta sección se presenta en detalle el método de evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones - AgilityPval describiendo: i) propósito, ii) objetivos específicos, iii) componentes, iv) roles, y v) proceso de evaluación. Dicho proceso está basado en PvalCOMPETISOFT, el cual es un proceso que describe de manera detallada la forma en que se debe llevar a cabo la evaluación de procesos y se adapta a cualquier instrumento de evaluación que requiera evaluar características específicas de los procesos de una organización [57].

#### 3.4.1. Propósito de AgilityPval

AgilityPval tiene como propósito evaluar la agilidad de los procesos de desarrollo de software de las organizaciones. La evaluación se realiza mediante un conjunto de preguntas que permiten conocer en qué medida se cumplen los indicadores de agilidad del proceso de desarrollo de software. Dichos indicadores fueron obtenidos a partir del análisis de los aspectos de agilidad descritos en AgilityRef. El reporte de evaluación de AgilityPval permite a las organizaciones encontrar oportunidades de mejora relacionadas con la agilidad de su proceso software.



### 3.4.2. Objetivos específicos de AgilityPval

AgilityPval fue desarrollado con los siguientes objetivos:

- Proveer los elementos necesarios para llevar a cabo la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones independientemente del enfoque de desarrollo que hayan adoptado.
- Proveer los elementos necesarios para identificar oportunidades de mejora relacionadas con la agilidad en los procesos de desarrollo de software de las organizaciones.
- Permitir la evaluación continua de la agilidad de los procesos de desarrollo de software en las organizaciones.

### 3.4.3. Componentes de AgilityPval

Para la definición de los componentes de AgilityPval, se tuvieron en cuenta los pasos descritos en el paradigma Goal, Question, Metric – GQM [44]. Este paradigma propone un modelo de medición de tres niveles: (i) Nivel conceptual (Goal), (ii) Nivel operacional (Question), y (iii) Nivel cuantitativo (Metric). En el nivel conceptual, se definió como meta la implementación en el proceso de desarrollo de los aspectos de agilidad definidos en el modelo de referencia AgilityRef. En el nivel operacional, se definió un conjunto de indicadores de agilidad asociados a cada aspecto de agilidad y un instrumento de evaluación tipo cuestionario. Por último, en el nivel cuantitativo, se definió un conjunto de métricas que a partir de las respuestas a las preguntas del instrumento de evaluación, permiten conocer el grado de agilidad del proceso de desarrollo teniendo en cuenta el grado de cumplimiento de los aspectos de agilidad, principios ágiles y valores ágiles. A continuación, se describen en detalle los indicadores de agilidad, el instrumento de evaluación y las métricas.

- **Indicadores de agilidad:** AgilityPval define un conjunto de indicadores de agilidad obtenidos a partir del análisis de los aspectos de agilidad descritos en el modelo de referencia AgilityRef. Dichos indicadores se definieron usando una escala nominal con posibles valores entre: “SI” o “NO”. En la Tabla 3.10 se puede observar la lista de indicadores de agilidad incluyendo su identificador y descripción. Además, cada indicador está asociado a un aspecto de agilidad de AgilityRef.

IdIND.	Aspecto de agilidad	Indicadores asociados al aspecto de agilidad	
		IdASP.	Descripción del indicador
AA_01	Iterativo e incremental	IA_01	El proceso de desarrollo del producto es iterativo e incremental.
		IA_02	Las iteraciones tienen una duración máxima de un mes.
		IA_03	Una iteración solamente inicia cuando ha terminado la iteración anterior
AA_02	Incrementos con valor	IA_04	Una iteración se planifica para que tenga como resultado una versión funcional del producto que sea útil para el cliente.
AA_03	Alcance negociable	IA_05	El alcance de cada iteración puede clarificarse y renegociarse entre el equipo y el cliente.
AA_04	Reunión diaria	IA_06	El equipo se reúne diariamente para evaluar el progreso hacia el objetivo de la iteración.
		IA_07	La duración de la reunión diaria se limita a un tiempo que permita a cada participante compartir de manera concisa lo que hizo el día anterior, lo que tiene planeado hacer ese día y los impedimentos para lograr el objetivo de la iteración.
AA_05	Refinamiento de requisitos	IA_08	La lista de requisitos del producto se actualiza constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.
		IA_09	Los requisitos pueden ser actualizados en cualquier momento para incluir más detalles, estimaciones y prioridades.
AA_06	Autoorganización del equipo	IA_10	Cada integrante del equipo puede elegir la mejor forma de llevar a cabo las tareas que tenga bajo su responsabilidad.
		IA_11	El equipo tiene la posibilidad de dar mayor prioridad a los requisitos del producto que presenten más riesgo.
AA_07	Responsabilidad compartida	IA_12	Los miembros del equipo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad del producto es de todo el equipo.
		IA_13	Cualquier miembro del equipo podría realizar cambios en cualquiera de los componentes del producto.

AA_08	Ritmo constante	IA_14	El tiempo de duración de las iteraciones es consistente durante todo el desarrollo del producto.
AA_09	Iteraciones negociables	IA_15	Una iteración se puede cancelar cuando se determina que su objetivo ha quedado obsoleto.
AA_10	Retrospectivas	IA_16	Al finalizar cada iteración, el equipo y los interesados se reúnen para revisar los resultados con respecto a las personas, procesos y herramientas.
		IA_17	En la reunión llevada a cabo al finalizar cada iteración, se identifican y ordenan los elementos más importantes que salieron bien y las posibles mejoras, se crea un plan para implementar las mejoras, y se adapta la lista de requisitos del producto en caso de ser necesario.
AA_11	Eliminación de impedimentos	IA_18	Existe un rol responsable de ayudar al equipo y a los interesados a ejecutar las actividades de la mejor manera y eliminar los impedimentos que se presenten.
AA_12	Participación del cliente	IA_19	El cliente o un representante suyo con poder de decisión, participa en la priorización de los requisitos que serán implementados en cada iteración.
		IA_20	El cliente o un representante suyo con poder de decisión, participa en el diseño de las pruebas funcionales del producto.
		IA_21	El cliente o un representante suyo con poder de decisión, participa en la elección de la tecnología a usar para implementar el producto.
		IA_22	El cliente o un representante suyo con poder de decisión, está disponible en la misma ubicación del equipo de desarrollo del producto para resolver inquietudes con respecto a los requisitos y prioridades para el negocio.
AA_13	Estimación en equipo	IA_23	El equipo es el encargado de estimar el tiempo que se empleará para implementar una característica del producto.
AA_14	Retroalimentación al cliente	IA_24	Cuando el equipo tome decisiones técnicas que puedan tener impacto a nivel de negocio, debe informar a los interesados.
AA_15	Visión compartida	IA_25	Se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los

			clientes.
AA_16	Atención al buen diseño	IA_26	Durante el diseño del producto se tiene en cuenta que no se dupliquen componentes, se incluyen todos los aspectos que el equipo considera importantes y se incluyen solamente lo que se requiere en el momento.
		IA_27	Cada vez que se agrega una nueva característica al producto, se verifica si se puede modificar su estructura interna para hacerla más sencilla, en caso de ser posible, se realizan los cambios y se verifica que todas las pruebas funcionen.
AA_17	Trabajo colaborativo	IA_28	Cuando un miembro del equipo va a implementar una nueva característica o va a realizar un cambio en el producto, puede pedir la colaboración a otro miembro del equipo para trabajar en conjunto.
AA_18	Integración continua	IA_29	Después de unas pocas horas todos los cambios que se hacen al producto se prueban e integran.
AA_19	Reglas técnicas	IA_30	Se acuerda un conjunto de reglas que todo el equipo debe cumplir al momento de agregar funcionalidades al producto o realizar modificaciones.
AA_20	Uso de métricas	IA_31	Se usan métricas que aporten información relevante sobre el desempeño del equipo.
AA_21	Herramientas de integración continua	IA_32	Se usan herramientas que permitan dar soporte a la integración continua de las nuevas funcionalidades y cambios realizados en el producto.
AA_22	Pruebas automáticas	IA_33	Se usan herramientas que permitan dar soporte a la ejecución de pruebas automáticas.

Acrónimos utilizados: IdIND: Identificador del indicador, IdASP: identificador del aspecto.

Tabla 3.10. Indicadores de agilidad de AgilityPval.

- Instrumento de evaluación:** A partir de los indicadores de agilidad descritos anteriormente, se definió un instrumento de evaluación tipo cuestionario donde el evaluador asigna un valor de "SI" o "NO" a cada indicador de agilidad de acuerdo a la evidencia que encuentre de su cumplimiento en el proceso de desarrollo. Esta evidencia puede ser recolectada a través de la observación directa de la ejecución del proceso de desarrollo y la recopilación de las opiniones de los responsables del proceso y el personal involucrado en su ejecución.

Es importante tener en cuenta que para asignar un valor de “SI” a un indicador de agilidad, se debe evidenciar que se cumple completamente en el proceso de desarrollo. En caso de que dicho indicador no se cumpla o se cumpla parcialmente, se debe marcar con un valor de “NO” y usar el campo “Comentarios” para describir el estado actual de implementación del indicador en el proceso de desarrollo. Por ejemplo, considerando el indicador IA\_06, si el evaluador determina que el equipo se reúne solamente una vez a la semana para evaluar el progreso hacia el objetivo de la iteración, el valor del indicador será “NO” y en el campo “Comentarios” deberá indicarse que aunque se efectúa la reunión con el propósito adecuado, esta se hace semanalmente. En el anexo D, es posible observar el instrumento de evaluación.

- **Métricas:** AgilityPval define cuatro métricas que permiten conocer de manera cuantitativa el grado de agilidad de los procesos de desarrollo de software a partir del cumplimiento de los indicadores de agilidad. En la Tabla 3.11, es posible observar las métricas definidas en AgilityPval las cuales permiten conocer: i) el grado de cumplimiento de los aspectos de agilidad, ii) el grado de cumplimiento de los principios ágiles, iii) el grado de cumplimiento de los valores ágiles, y iv) el grado de agilidad del proceso de desarrollo de software. Todas las métricas se derivan de las respuestas obtenidas en el instrumento de evaluación.

Id.	Nombre	Descripción	Unidad de medida
<i>gAA</i>	Grado de cumplimiento del aspecto de agilidad.	Representa el grado de cumplimiento de un aspecto de agilidad en el proceso de desarrollo.	Porcentaje
<i>gPA</i>	Grado de cumplimiento del principio ágil.	Representa el grado de cumplimiento de un principio ágil en el proceso de desarrollo.	Porcentaje
<i>gVA</i>	Grado de cumplimiento del valor ágil.	Representa el grado de cumplimiento de un valor ágil en el proceso de desarrollo.	Porcentaje
<i>gAP</i>	Grado de agilidad del proceso de desarrollo de software.	Representa el grado de agilidad del proceso de desarrollo de software.	Porcentaje

Acronimos utilizados: Id: Identificador.

Tabla 3.11. Métricas definidas en AgilityPval.

Para la calificación del grado de cumplimiento de los aspectos, principios y valores, así como para la calificación del grado de agilidad del proceso de desarrollo, se utilizó como referente la escala establecida en la norma ISO/IEC 15504 debido a que es usada para calificar los atributos de un proceso que está siendo evaluado. A continuación, en la Tabla 3.12 se describe la escala de calificación.

Acrónimo	Rango	Grado de cumplimiento
NA	El grado de cumplimiento o grado de agilidad está entre el 0% y el 15%.	No alcanzado
PA	El grado de cumplimiento o grado de agilidad está entre el 16% y el 50%.	Parcialmente alcanzado
AA	El grado de cumplimiento o grado de agilidad está entre el 51% y el 85%.	Ampliamente alcanzado
CA	El grado de cumplimiento o grado de agilidad está entre el 86% y el 100%.	Completamente alcanzado

Tabla 3.12. Escala de calificación definida en AgilityPval.

A continuación, se describe en detalle cada una de las métricas:

- **Grado de cumplimiento del aspecto de agilidad –  $gAA$**

Esta métrica permite conocer en qué porcentaje se está cumpliendo con un determinado aspecto de agilidad en el proceso de desarrollo de software. Se calcula usando la siguiente ecuación:

$$gAA = \frac{nIAc}{nIA} \times 100$$

Dónde:

$nIAc$  = Número de indicadores de agilidad asociados al aspecto de agilidad con un valor de "SI"

$nIA$  = Número total de indicadores de agilidad asociados al aspecto de agilidad

Por ejemplo, el aspecto de agilidad AA\_01 (iterativo e incremental) tiene asociados tres indicadores de agilidad (IA\_01, IA\_02, IA\_03). En este caso, suponiendo que se cumplieran solamente dos de ellos en el proceso de desarrollo, se tendría que el grado de cumplimiento del aspecto sería de  $(2/3) \times 100 = 66,7\%$  - Ampliamente alcanzado.

- **Grado de cumplimiento del principio ágil –gPA**

Esta métrica permite conocer en qué porcentaje se está cumpliendo con un determinado principio ágil en el proceso de desarrollo de software. Su cálculo corresponde al promedio del grado de cumplimiento de los aspectos de agilidad asociados a ese principio ágil:

$$gPA = \frac{\sum gAAp}{nAAp}$$

Dónde:

*gAAp* = Grado de cumplimiento de cada aspecto de agilidad asociado al principio ágil

*nAAp* = Número total de aspectos de agilidad asociados al principio ágil

Por ejemplo, el principio ágil PA\_05 tiene asociados cuatro aspectos de agilidad (AA\_06, AA\_07, AA\_11, AA\_13). En este caso, suponiendo un grado de cumplimiento de cada aspecto correspondiente a: AA\_06: 50%, AA\_07: 50%, AA\_11: 100% y AA\_13: 100%, se tendría un grado de cumplimiento del principio ágil PA\_05 de 75% - Ampliamente alcanzado.

- **Grado de cumplimiento del valor ágil –gVA**

Esta métrica permite conocer en qué porcentaje se está cumpliendo con un determinado valor ágil en el proceso de desarrollo de software. Su cálculo corresponde al promedio del grado de cumplimiento de los principios ágiles asociados a ese valor ágil:

$$gVA = \frac{\sum gPAv}{nPAv}$$

Dónde:

*gPAv* = Grado de cumplimiento de cada principio ágil asociado al valor ágil

*nPAv* = Número total de principios ágiles asociados al valor ágil

Por ejemplo, el valor ágil VA\_2 tiene asociados tres principios ágiles (PA\_01, PA\_03, PA\_07). En este caso, suponiendo un grado de cumplimiento de cada principio ágil correspondiente a: PA\_01: 62%, PA\_03: 47% y PA\_07: 50%, se tendría un grado de cumplimiento del valor ágil VA\_2 de 53% - Ampliamente alcanzado.

- **Grado de agilidad del proceso de desarrollo de software –gAP**

Esta métrica permite conocer el porcentaje de agilidad del proceso de desarrollo de software. Su cálculo corresponde al promedio del grado de cumplimiento de los valores ágiles:

$$gAP = \frac{\sum gVA}{nVA}$$

Dónde:

*gVA = Grado de cumplimiento de cada valor ágil*

*nVA = Número total de valores ágiles*

Por ejemplo, suponiendo un grado de cumplimiento de cada valor ágil correspondiente a: VA\_1: 57%, VA\_2: 31%, VA\_3: 50% y VA\_4: 50%, se tendría un porcentaje de agilidad del proceso de desarrollo de 47% - Parcialmente alcanzado.

- **Roles:** En la Tabla 3.13 se pueden observar los roles definidos en AgilityPval. Es importante considerar que un rol puede ser asumido por varias personas y una persona puede asumir varios roles.

Id. rol	Nombre	Características
EV	Evaluador	Conocimiento del método de evaluación, capacidad para aplicar el instrumento de evaluación y analizar los datos recolectados.
RP	Responsable del proceso	Conocimiento del proceso de desarrollo de software de la organización.

Acrónimos utilizados: Id: Identificador.

Tabla 3.13. Roles definidos en AgilityPval.

- **Proceso de evaluación:** El proceso de evaluación consta de las cuatro actividades que se describen a continuación:
  - **Planificación de la evaluación:** El evaluador describe la organización donde se realizará la evaluación, la fecha de realización, las actividades y el tiempo a emplear.
  - **Ejecución de la evaluación:** El evaluador recolecta la información necesaria para diligenciar el instrumento de evaluación. Dicha información se recolecta mediante observación directa de la ejecución del proceso de desarrollo en la



organización y mediante la recolección de las opiniones de los responsables del proceso y el personal involucrado en su ejecución.

- **Generación de resultados:** Con el instrumento de evaluación diligenciado, el evaluador aplica las métricas descritas anteriormente con el fin de obtener el grado de agilidad del proceso de desarrollo de software de la organización y en más detalle el grado de cumplimiento de aspectos, principios y valores ágiles.
- **Reporte de resultados:** Con los resultados de las métricas, el evaluador presenta un reporte de resultados al responsable del proceso. En el reporte, se muestran los resultados cuantitativos (valores numéricos calculados con las métricas), gráficas de grado de cumplimiento de aspectos, principios y valores ágiles, y oportunidades de mejora detectadas.

## 3.5. Prototipo Web - AgilityProTool

En esta sección se presenta el prototipo de una aplicación web para soportar la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones llamada AgilityProTool, la cual está basada en el método de evaluación AgilityPval, descrito en la sección 3.4. AgilityProTool está disponible en <http://bit.ly/AgilityProTool> y permite a las organizaciones registrarse de forma gratuita para realizar la evaluación de la agilidad de su proceso de desarrollo. A continuación se detalla el prototipo desarrollado teniendo en cuenta: i) requisitos funcionales, ii) arquitectura de la solución propuesta, iii) enfoque de desarrollo, y iv) resultados obtenidos.

### 3.5.1. Requisitos funcionales

La especificación inicial de requisitos para el desarrollo de AgilityProTool se realizó mediante la definición de historias épicas, las cuales son una descripción general de la funcionalidad esperada del software tal y como la percibe el usuario [74]. Las historias épicas permiten tener una visión general de las funcionalidades del sistema y pueden ser divididas en varias historias de usuario a medida que se avanza el desarrollo de cada funcionalidad [74]. En la Tabla 3.14 se describen las historias épicas definidas para AgilityProTool.

Id.	Historia épica
H1	Como evaluador, necesito registrarme en la aplicación con la finalidad de iniciar sesión.

H2	Como evaluador, necesito iniciar sesión en la aplicación con la finalidad de iniciar una evaluación o ver los resultados de las evaluaciones realizadas.
H3	Como evaluador, necesito iniciar una evaluación con la finalidad de conocer el grado de agilidad del proceso de desarrollo de software de la organización.
H4	Como evaluador, necesito consultar las evaluaciones realizadas con el fin de ver los resultados obtenidos o continuar realizando la evaluación en caso de que no haya sido terminada.

Acrónimos utilizados: Id: Identificador.

Tabla 3.14. Historias épicas definidas para AgilityProTool.

### 3.5.2. Arquitectura de la aplicación

La arquitectura de una aplicación describe la estructura del sistema, lo que comprende a los componentes software, sus propiedades y las relaciones entre ellos [29]. De acuerdo a lo anterior, para el diseño de AgilityProTool, se eligió el patrón de arquitectura de software Modelo-Vista-Controlador – MVC, el cual se usa a menudo para el diseño de aplicaciones Web debido principalmente a que facilita el mantenimiento y la escalabilidad de las aplicaciones gracias al bajo acoplamiento entre sus componentes [75].

En la Figura 3.8, se puede observar el diagrama de despliegue de la aplicación, el cual permite representar las relaciones de los nodos hardware que componen el sistema y la forma en que los componentes de la aplicación se despliegan en cada nodo y se comunican entre sí. El sistema se compone de tres nodos: el servidor web, el servidor de base de datos, y el cliente. La interacción entre los usuarios y la aplicación se hace mediante un navegador web, y para la persistencia de datos se utiliza una de base de datos relacional. El patrón MVC se ve reflejado en los componentes desplegados en el servidor Web, en el cual se despliegan tres componentes principales: el modelo, la vista, y el controlador.

- **Modelo:** El modelo contiene las clases de lógica de negocio y las operaciones que permiten ejecutar las reglas de negocio y comunicarse con el servidor de base de datos para actualizar o consultar datos. En la Figura 3.9, es posible observar el diagrama de clases de negocio, en el cual se ven reflejados las entidades y relaciones identificadas en el método de evaluación AgilityPval. La información de los indicadores de agilidad, aspectos de agilidad, principios y valores ágiles, es gestionada por las clases PreguntaIndicador, AspectoAgilidad, PrincipioAgil y ValorAgil, respectivamente. La información de la evaluación de la

empresa es gestionada por las clases EvaluacionEmpresa y RespuestaPreguntaInd. Los resultados de la evaluación se gestionan mediante las clases CumplimientoValor, CumplimientoPrincipio y CumplimientoAspecto.

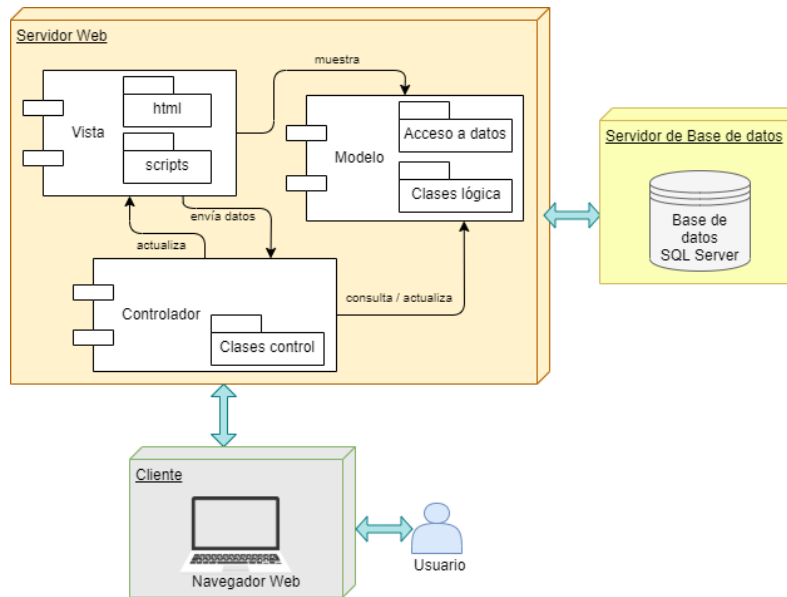


Figura 3.8 Diagrama de despliegue de AgilityProTool.

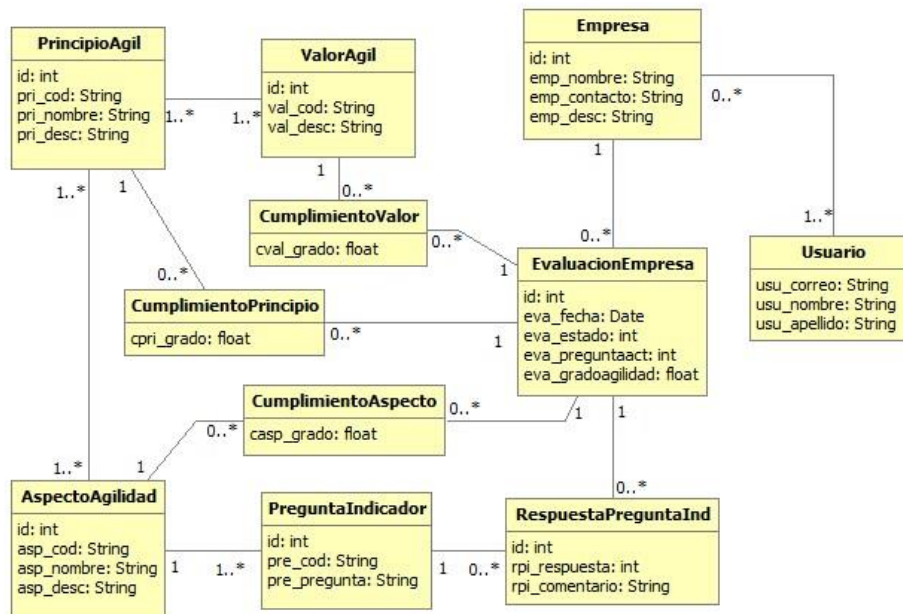


Figura 3.9 Diagrama de clases de AgilityProTool.

- **Vista:** Las vistas contienen los elementos relacionados con la interfaz gráfica, así como la lógica de presentación de los contenidos al usuario. Los principales elementos que contiene este componente son: páginas HTML, archivos JavaScript para realizar validaciones de entradas de usuario y ajustes de la interfaz gráfica según el tipo de dispositivo, hojas de estilo CSS, e imágenes.
- **Controlador:** Contiene las clases controladoras, las cuales atienden las peticiones de los usuarios, realizan las operaciones necesarias a través de las clases del modelo, y seleccionan y las vistas que deben mostrar al usuario según las peticiones realizadas.

### 3.5.3. Diagrama de secuencia general

En la Figura 3.10, es posible observar un diagrama de secuencia que representa de manera general los mensajes enviados entre cada componente del sistema para atender una petición del usuario según la arquitectura definida. A continuación, se describen las interacciones entre los componentes principales:

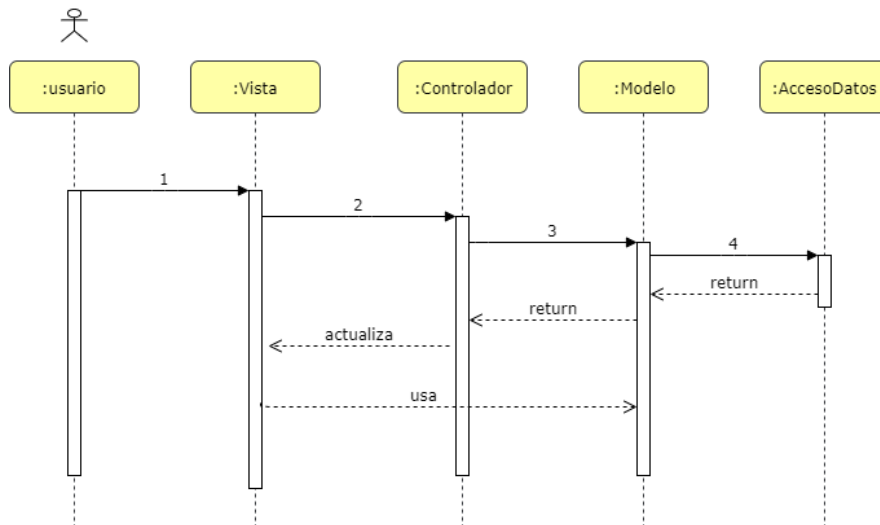


Figura 3.10 Diagrama de secuencia general de AgilityProTool.

1. El usuario interactúa con la vista y genera un evento
2. El evento es atendido por el controlador respectivo
3. El controlador interactúa con el modelo para solicitar datos o actualizarlos.
4. Si es necesario, el modelo interactúa con las clases de acceso a datos para consultar o actualizar información en la base de datos.

5. Una vez el controlador ha procesado la solicitud con la ayuda del modelo, actualiza la vista o realiza una redirección a una nueva vista.
6. En caso de ser necesario, la vista se actualiza con la información almacenada en el modelo.

#### 3.5.4. Enfoque de desarrollo

El desarrollo de la aplicación se realizó usando un enfoque iterativo e incremental donde el rol de desarrollador fue asumido por el Ing. Wilson Alfredo Ortega Ordoñez y el rol de cliente fue asumido por el PhD. César Jesús Pardo Calvache. A continuación se describen las fases de desarrollo de la aplicación:

- **Fase de planificación:** En esta fase, a partir de las historias épicas definidas con el cliente, se realizó una planificación inicial de las iteraciones necesarias para el desarrollo de la aplicación. Para la definición de las iteraciones y su orden de realización, se realizó una reunión entre el desarrollador y el cliente con el fin de definir la prioridad de las funcionalidades desde el punto de vista del cliente. Además, se tuvo en cuenta que la estimación de esfuerzo en horas realizada por el desarrollador para cada iteración no superara las 10 horas. En la Tabla 3.15 se describen las iteraciones definidas.

No. Iteración	Objetivo
1	Seleccionar y configurar del entorno de desarrollo
2	Implementar el registro/inicio de sesión del usuario
3	Implementar la creación/listado de evaluaciones
4	Implementar la visualización de preguntas de la evaluación
5	Implementar el almacenamiento de respuestas a las preguntas
6	Implementar el cálculo de resultados de la evaluación
7	Implementar la visualización de resultados de la evaluación

Tabla 3.15. Iteraciones definidas para el desarrollo de AgilityProTool.

- **Fase de implementación:** En esta fase se realizaron todas las iteraciones planificadas en la fase anterior. En la iteración inicial, como framework de desarrollo, se seleccionó ASP.NET MVC 5 teniendo en cuenta la experiencia previa del desarrollador usando tecnologías .NET y la facilidad que brinda para implementar la arquitectura MVC definida para la aplicación [75]. Como entorno de desarrollo se seleccionó Visual Studio 2017 y como motor de base de datos se

usó SQL Server 2017. Asimismo, se configuró un servidor web para desplegar los resultados de cada iteración con el fin de facilitar la revisión de las funcionalidades por parte del cliente a lo largo del proceso de desarrollo.

En cada una de las siguientes iteraciones, se realizaron los incrementos de acuerdo a los objetivos de la iteración. Es importante resaltar que en cada iteración se realizaron las actividades de análisis, diseño, implementación, pruebas y despliegue. Asimismo, al finalizar cada iteración se realizaron reuniones entre el desarrollador y el cliente para evaluar el progreso y detectar problemas o posibles cambios en el producto.

### **3.5.5. Resultados obtenidos**

Al finalizar las iteraciones de la fase de implementación, se obtuvo el prototipo de la aplicación web de acuerdo a los requisitos planteados al inicio del desarrollo (disponible en: <http://bit.ly/AgilityProTool>). El prototipo está disponible para que cualquier organización pueda registrarse de manera gratuita y realizar la evaluación de la agilidad de su proceso de desarrollo las veces que desee. Dicha evaluación consiste en el diligenciamiento de un cuestionario basado en el instrumento de evaluación AgilityPval (ver Figura 3.11). En caso de no terminar completamente el cuestionario, se almacena el progreso alcanzado y es posible continuar con la evaluación más adelante (ver Figura 3.12). Al finalizar el cuestionario, se calculan automáticamente los grados de cumplimiento de los aspectos de agilidad, principios y valores ágiles, así como el grado de agilidad del proceso. Asimismo, es posible visualizar el listado de evaluaciones realizadas por cada organización con el fin de facilitar el análisis histórico de resultados obtenidos.

The screenshot shows a web interface for a questionnaire. At the top, there is a green header with the logo 'AgilityProTool' and navigation links 'Inicio' and 'Contacto'. On the right side of the header, it says 'Hola waortega' and 'Cerrar sesión'. A dark sidebar on the left contains the menu items 'Empresa', 'Evaluaciones', and 'Resultados'. The main content area is titled 'Evaluación de la agilidad del proceso de desarrollo de software' and includes a 'Ver Evaluaciones' link. A green callout box contains the text: 'Iterativo e incremental. El producto se crea de forma iterativa e incremental. Los intervalos tienen una duración máxima de un mes. Además, cada nueva iteración comienza inmediatamente después de la finalización de la iteración anterior.' Below this, a dark blue box states: 'Las iteraciones tienen una duración máxima de un mes.' There are two radio buttons for 'Si' and 'No', with 'Si' selected. A text input field labeled 'Comentarios' is present. At the bottom, there are three buttons: 'Anterior', 'Pregunta 2', and 'Siguiente'.

Figura 3.11 Página del cuestionario de evaluación en AgilityProTool.

The screenshot shows a web interface for a list of evaluations. At the top, there is a green header with the logo 'AgilityProTool' and navigation links 'Inicio' and 'Contacto'. On the right side of the header, it says 'Hola waortega' and 'Cerrar sesión'. A dark sidebar on the left contains the menu items 'Empresa', 'Evaluaciones', and 'Resultados'. The main content area is titled 'Evaluaciones' and includes a 'Ver Evaluaciones' link. A green button labeled 'Nueva Evaluación +' is visible. Below it is a table with the following data:

Empresa	Estado	Fecha	Acciones
Desarrollo Web SA	1,19%	17/10/2019	<a href="#">Continuar Evaluación</a>

Below the table, there is a blue button labeled '1'.

Figura 3.12 Página de listado de evaluaciones en AgilityProTool.





## Capítulo 4

### Evaluación de AgilityPRO

En este capítulo, se presenta la evaluación de la solución propuesta a través de la realización de un grupo focal y un estudio de caso. En primer lugar, en la sección 4.1, se describe el grupo focal realizado con el fin de evaluar la primera versión del modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software. A partir de los resultados del grupo focal, se encontraron oportunidades de mejora que permitieron realizar ajustes al modelo y obtener una segunda versión, la cual se presentó en la sección 3.3. En segundo lugar, se presenta el estudio de caso realizado en una organización desarrolladora de software, el cual tuvo como objetivo evaluar la utilidad, comprensibilidad, y practicidad del método de evaluación de la agilidad de los procesos de desarrollo de software.

#### 4.1. Evaluación de AgilityRef usando un grupo focal

En esta sección se presentan los resultados del grupo focal realizado con el fin de evaluar la primera versión del modelo de referencia AgilityRef. Mediante la realización del grupo focal se encontraron oportunidades de mejora y se refinó el modelo de referencia propuesto. Para la realización del grupo focal se siguieron las siguientes fases teniendo en cuenta los lineamientos propuestos en [76]:

- **Planeamiento de la investigación:** Se establecen los elementos de contenido y de procedimiento que serán aplicados al debate de los participantes.
- **Diseño del grupo de discusión (Reclutamiento):** Se definen las estrategias de selección de los participantes del grupo focal.

- **Conducción de la sesión de debate (Moderación):** Se ejecutan los procedimientos establecidos en la primera fase con el fin de generar la sesión de debate y capturar la opinión de los participantes.
- **Análisis de la información y reporte de resultados:** Se realiza el análisis de tipo cualitativo y/o cuantitativo utilizando estadística descriptiva o métodos de tipo cuantitativo.

A continuación, se presentan en detalle los resultados de la ejecución de cada una de las fases descritas anteriormente.

#### 4.1.1. Planeamiento de la investigación

Se definió el objetivo del grupo focal y el objetivo de investigación. Adicionalmente, se prepararon los materiales y procedimientos a seguir por parte del grupo investigador.

- **Objetivo del grupo focal:** Conocer la opinión y percepción de profesionales con experiencia en desarrollo ágil de software acerca de la completitud, idoneidad y comprensibilidad de AgilityRef como modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones.
- **Objetivo de investigación:** Realizar la evaluación inicial del modelo de referencia AgilityRef con el fin de encontrar oportunidades de mejora y/o generar una nueva versión del modelo.
- **Preparación de materiales y procedimientos a seguir por parte del grupo investigador:** El objetivo de esta actividad fue definir los elementos, procedimientos y técnicas a emplear en la ejecución del grupo focal, entre los elementos se encuentran: (i) estructura del protocolo del grupo focal, (ii) los instrumentos y métodos que serán empleados, (iii) definición de métodos de captura y registro de información y (iv) la definición de los métodos de análisis de la información obtenida en el debate.
  - **Protocolo del grupo focal:** A continuación en la Tabla 4.1 se presenta el protocolo definido para llevar a cabo el grupo focal.

N°	Elemento	Descripción
1	Agenda de	Documento que contiene las actividades que llevará a cabo cada

	trabajo	participante durante la aplicación del grupo focal.
2	Cuestionario	Documento que contiene las preguntas que permitirán obtener información relevante para la evaluación de la propuesta durante el grupo focal.
3	Estructura de protocolo	Documento que indica el protocolo utilizado para la aplicación del grupo focal.
4	Propuesta a evaluar	Documento que contiene la descripción de la propuesta que será evaluada.

Acrónimos utilizados: N°: Número

Tabla 4.1. Protocolo definido para el grupo focal.

- **Elementos necesarios para llevar a cabo el grupo focal:** En la Tabla 4.2, se presentan los elementos utilizados para la realización del grupo focal.

Elemento	Descripción
Fecha de realización	Fecha en la cual se realizará el grupo focal.
Hora de inicio	Hora exacta en la que dará inicio el grupo focal.
Duración	Duración aproximada del grupo focal
Lugar	Lugar donde se realizará el grupo focal.
Tema a tratar	El tema que tratará el grupo focal.
Moderador	Nombre de la persona que asegura que los participantes realicen aportes acordes con el tema a tratar y verificar que se cumpla con la agenda propuesta.
Supervisor	Persona encargada de recopilar la información relevante.
Relator	Persona encarga de exponer el tema del grupo focal.
Participantes	Personas encargadas de evaluar la propuesta presentada.
Objetivo del grupo focal	Objetivo principal de la realización del grupo focal.
Objetivo de investigación	Objetivos relacionados con las actividades realizadas en el grupo focal.

Tabla 4.2. Elementos para la realización del grupo focal.

- **Métodos de captura y registro de información:** Para llevar a cabo la ejecución del grupo focal se contó con un relator, quien tomó nota de las apreciaciones, comentarios y sugerencias relevantes de cada participante. También, se hizo entrega a cada participante de un cuestionario y un documento con la propuesta. Además, se realizó un registro de audio de todo el grupo focal, esto con el fin de tener material extra de apoyo.

- **Métodos de análisis de la información:** Luego de la realización del grupo focal, se llevó a cabo un análisis estadístico de la información plasmada en los cuestionarios y un análisis cualitativo a partir de las observaciones y oportunidades de mejora registradas.

#### 4.1.2. Diseño del grupo de discusión (Reclutamiento)

En esta sección se definen las estrategias de selección de los participantes del grupo focal. Esta actividad estuvo a cargo del grupo investigador e incluyó las siguientes actividades:

- **Definición del perfil del participante:** Para la selección de expertos, se definieron los siguientes criterios:
  - Estar activo en la industria software y/o el entorno académico ya sea como profesor o como estudiante.
  - Tener conocimientos sobre enfoques ágiles y su aplicación en la industria software.
  - Profesionales con experiencia en la industria de software.
- **Identificación de los participantes:** Teniendo en cuenta los criterios anteriormente definidos, se identificaron los posibles participantes para el grupo focal. El grupo conformado estuvo compuesto por profesionales con experiencia y conocimiento en diferentes áreas de la ingeniería del software, específicamente en enfoques ágiles de desarrollo de software. La Tabla 4.3, presenta la descripción del perfil profesional de los participantes.

Id.	Ocupación	Estudios	Experiencia
PF1	Docente universitario.	Ingeniería de Sistemas. Maestría en Ingeniería de Software.	Desarrollo de software. Docente en el área de programación, bases de datos e ingeniería de software. Experiencia en Scrum tanto para aplicaciones web como empresariales.
PF2	Instructor en el área de ingeniería de software.	Ingeniería de Sistemas. Especialización en desarrollo de soluciones informáticas.	Desarrollador Backend. Desarrollador de aplicaciones móviles. Instructor SENA en el área de desarrollo de software.

		Estudiante de Maestría en Computación. Scrum Master. Scrum Developer.	Todos los proyectos de desarrollo en el SENA usan SCRUM. Líder de grupo de desarrollo.
PF3	Gerente.	Maestría en Computación. Especialización en Sistemas Gerenciales. Especialización en Redes de Comunicación.	Líder de calidad Líder de aseguramiento de calidad Coordinador de calidad Coordinador de desarrollo Líder de proyectos híbridos Implementador de prácticas ágiles
PF4	Arquitecto líder.	Ingeniería de Sistemas. Especialización en Gerencia de Proyectos. Maestría en Ingeniería.	Docencia en Ingeniería de Sistemas. Docencia en Especialización en Desarrollo de Soluciones Informáticas. Arquitecto líder en SIIGO SA. Durante 12 años en SIIGO se adaptaron diferentes enfoques ágiles. Actualmente se trabaja con un proceso híbrido y adaptado a las necesidades de la organización y al tamaño del equipo.
PF5	Ingeniero de desarrollo.	Ingeniería de Sistemas. Scrum Master.	Desarrollador para la plataforma EPIONE. Scrum master del equipo. Proyecto de investigación como tesis de pregrado: Guía para evaluar el nivel de cumplimiento de Scrum. Apoyo en el proceso de capacitación en implementación del marco Scrum en el equipo TIC de la IPS Horisoos.
PF6	Docente universitario.	Ingeniería de Sistemas.	5 años de experiencia en desarrollo de software. 16 años de experiencia en docencia universitaria. Experiencia usando enfoques ágiles en la asignatura Proyecto II del programa de Ingeniería de Sistemas de la Universidad del Cauca. Además, 2 años usando enfoques ágiles en el área de desarrollo de software.

Acrónimos utilizados: Id: Identificador

Tabla 4.3. Perfil profesional de los participantes en el grupo focal.

#### 4.1.3. Conducción de la sesión de debate

La sesión de debate tuvo una duración de dos horas y fue coordinada por un moderador que siguió el orden y la secuencia presentada en la Tabla 4.4.

N°	Descripción
1	Bienvenida a los participantes
2	Presentación del grupo investigador, objetivos del grupo focal y de investigación
3	Presentación de los participantes
4	Presentación de AgilityRef
5	Discusión de la propuesta por parte de los participantes
6	Realización de la encuesta
7	Agradecimiento a los participantes
8	Finalización del grupo Focal

Acrónimos utilizados: N°: Número

Tabla 4.4. Organización del grupo focal.

#### 4.1.4. Captura de información

El proceso de captura de información se realizó teniendo en cuenta las estrategias definidas en la fase de métodos de captura y registro de información. El relator fue la persona encargada de tomar atenta nota de cada observación y comentario que realizaron los participantes.

Además, como apoyo a los comentarios recibidos se pidió a los participantes que respondieran un cuestionario al final de la sesión de discusión de la propuesta. La primera parte del cuestionario permitió recolectar la siguiente información de cada participante: i) nombre, ii) ocupación, iii) estudios, iv) experiencia profesional, y v) experiencia con enfoques ágiles de desarrollo de software.

La segunda parte del cuestionario se enfocó en conocer la opinión de los participantes sobre la propuesta. Las primeras 8 preguntas, se diseñaron para ser respondidas mediante la siguiente escala de Likert de 5 puntos:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Ni de acuerdo ni en desacuerdo
4. De acuerdo
5. Totalmente de acuerdo

Dichas preguntas, permitieron medir la completitud (2 preguntas), comprensibilidad (4 preguntas) e idoneidad (2 preguntas) del modelo propuesto. Por último, se incluyeron 2 preguntas abiertas que permitieron a los participantes proponer ajustes al modelo y realizar comentarios adicionales. A continuación, en la Tabla 4.5, se pueden observar las preguntas sobre la propuesta. En el anexo C, se presentan los cuestionarios diligenciados por los participantes.

Aspecto a evaluar	Id.	Pregunta
Completitud	P1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?
	P2	¿Considera que AgilityRef contiene todas las relaciones posibles entre los principios y los valores ágiles?
Comprensibilidad	P3	¿Considera que la estructura de AgilityRef es de fácil comprensión?
	P4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?
	P5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?
	P6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?
Idoneidad	P7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?
	P8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?
<b>Preguntas abiertas</b>		
Id.	Pregunta	
P9	¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, explique los cambios propuestos.	
P10	¿Tiene algún comentario adicional acerca de AgilityRef?	

Acrónimos utilizados: Id: Identificador

Tabla 4.5. Cuestionario de evaluación usado en el grupo focal

#### 4.1.5. Análisis de la información y reporte de resultados

Una vez realizado el grupo focal, se llevó a cabo el análisis de los aportes realizados por los participantes durante la sesión de discusión de la propuesta y a través de los cuestionarios diligenciados al final de la sesión. A continuación, se presentan las actividades llevadas a cabo para realizar el análisis de la información obtenida en el grupo focal.

- **Análisis de las preguntas cerradas:** Para las preguntas P1 a P8, se realizó el conteo de las respuestas de cada participante. En la Tabla 4.6, se presenta el conteo de respuestas para cada opción de la escala Likert (1, 2, 3, 4, 5) en las preguntas P1 a P8. En la Figura 4.1 se muestra una gráfica con los resultados consolidados.

Id.	Nivel de conformidad				
	1	2	3	4	5
P1	0	0	0	6	0
P2	0	0	1	2	3
P3	0	0	0	2	4
P4	0	0	2	3	1
P5	0	0	0	1	5
P6	0	0	0	4	2
P7	0	0	3	2	1
P8	0	0	0	4	2

Acrónimos utilizados: Id: Identificador

Tabla 4.6. Conteo de respuestas a preguntas P1-P8

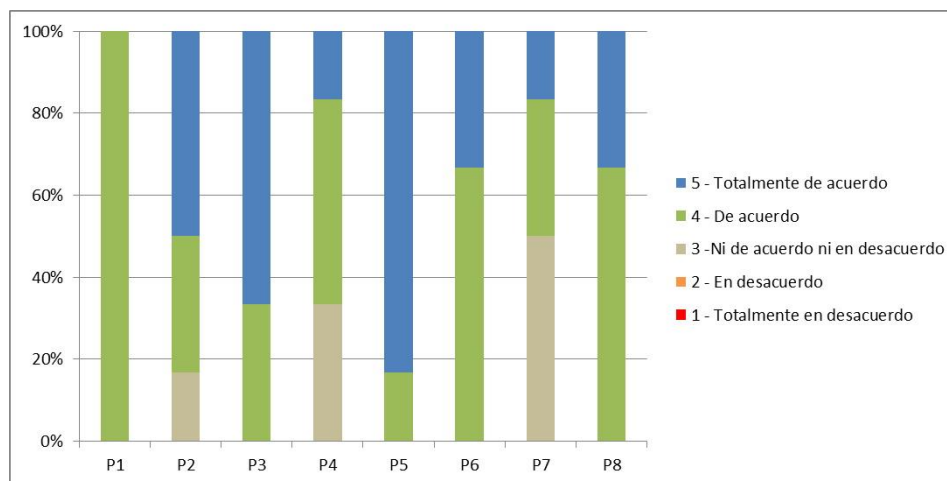


Figura 4.1 Consolidado de respuestas a preguntas P1-P8



A continuación se presenta el análisis de las respuestas a las preguntas agrupadas según el aspecto a evaluar: completitud, comprensibilidad e idoneidad.

○ **Completitud:**

En P1 se preguntó a los participantes si consideraban que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles. Todos los participantes (6) respondieron con la opción “De acuerdo”.

En P2 se preguntó a los participantes si consideraban que AgilityRef contiene todas las relaciones posibles entre los principios y los valores ágiles. Un (1) participante respondió usando la opción “Ni de acuerdo ni en desacuerdo”, dos (2) participantes respondieron con la opción “De acuerdo”, y tres (3) participantes respondieron usando la opción “Totalmente de acuerdo”.

○ **Comprensibilidad:**

En P3 se preguntó a los participantes si consideraban que la estructura de AgilityRef es de fácil comprensión. Dos (2) participantes respondieron con la opción “De acuerdo”, y cuatro (4) participantes respondieron usando la opción “Totalmente de acuerdo”.

En P4 se preguntó a los participantes si consideraban que los indicadores de agilidad propuestos son de fácil comprensión. Dos (2) participantes respondieron usando la opción “Ni de acuerdo ni en desacuerdo”, cuatro (4) participantes respondieron con la opción “De acuerdo”, y un (1) participante respondió con la opción “Totalmente de acuerdo”.

En P5 se preguntó a los participantes si consideraban que la estructura de AgilityRef es de fácil comprensión. Dos (2) participantes respondieron con la opción “De acuerdo”, y cuatro (4) participantes respondieron usando la opción “Totalmente de acuerdo”.

En P6 se preguntó a los participantes si consideraban que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión. Cuatro (4) participantes respondieron con la opción “De acuerdo”, y dos (2) participantes respondieron mediante la opción “Totalmente de acuerdo”.

○ **Idoneidad:**

En P7 se preguntó a los participantes si consideraban que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software. Tres (3) participantes respondieron usando la opción “Ni de acuerdo ni en desacuerdo”, dos (2) participantes respondieron con la opción “De acuerdo”, y un (1) participante respondió mediante la opción “Totalmente de acuerdo”.

En P8 se preguntó a los participantes si consideraban que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software. Cuatro (4) participantes respondieron con la opción “De acuerdo”, y dos (2) participante respondieron usando la opción “Totalmente de acuerdo”.

Según los resultados anteriores, se puede observar que los participantes tuvieron una opinión favorable acerca de la completitud, comprensibilidad e idoneidad de AgilityRef.

- **Análisis de las preguntas abiertas:** Las preguntas P9 y P10 permitieron a los participantes proponer ajustes al modelo y realizar comentarios adicionales. A continuación, en la Tabla 4.7 se presentan las respuestas de cada uno de los participantes.

Pregunta	PF.	Respuesta
P9: ¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los	PF1	Considero que se deberían definir indicadores más genéricos que puedan dar cabida a las actividades que se realizan en los diferentes enfoques ágiles. Por ejemplo: Reunión incluye: reunión de iniciación, reunión diaria, reunión de seguimiento, reunión de lanzamiento. Gestión del cambio incluye: requisitos, producto, etc.

cambios propuestos.	PF2	Los indicadores deben tener nombres y descripciones más generales y no amarradas a una fuente específica ya que las organizaciones que no usen esas mismas fuentes no se van a sentir identificadas con el identificador propuesto. Por ejemplo, para el indicador reunión diaria, si mi organización no usa Scrum puede no calzar a ese identificador.
	PF3	Agregar relaciones entre principios y valores: PA_01 -> VA_1, PA_02 -> VA_1, PA_08 -> VA_3, PA_09 -> VA_2, PA_11 -> VA_2. Definición de proceso de desarrollo ágil. Adicionar tipo de indicador para las reglas de negocio que en un proceso son condicionales y que siempre se ligan con reglas de negocio que dan valor al usuario. Los tipos de indicador podrían ser entendidos desde varias perspectivas por tanto podrían quitarse.
	PF4	Me gustaría que se pudiera hacer un mapeo entre otras actividades propuestas y las estándar de los modelos ágiles.
	PF5	Consideraría realizar un agrupamiento en la tabla de indicadores. Lo anterior dado que algunas descripciones podrían agrupar más fuentes. Ejemplo: En la tabla 3 en el identificador IA_15: "Se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los clientes" está relacionada con XP, además de esta fuente se podría incluir Scrum dado que al igual que XP propone un lenguaje común para todo el equipo. Lo mismo para otras actividades como: IA_20, IA_21. Aunque estas últimas están relacionadas más bien hacia el cómo y no están planteadas en la guía de Scrum. En tal caso sería interesante diferencias las fuentes.
	PF6	Ninguno
	P10: ¿Tiene algún comentario adicional acerca de AgilityRef?	PF1
PF2		Excelente trabajo.
PF3		¿Qué pasaría con el modelo en caso que una organización cuente con sus propios valores ágiles y con los cuales viene trabajando su concepto de ser ágiles? Los "indicadores de agilidad" podrían considerarse como "aspectos de agilismo a evaluar" que luego sí podrían pasar a ser indicadores cuando trabajen una medida en la siguiente

		etapa del proyecto. Lo anterior bajo la definición de la ontología de la medición)
	PF4	Ninguno.
	PF5	Excelente trabajo.
	PF6	Expresar los indicadores en términos de medidas. Por ejemplo, en lugar de "iterativo e incremental", % de iteraciones que tuvieron entregas en la fecha establecida.

Acrónimos utilizados: PF: Identificador del participante

Tabla 4.7. Respuestas a las preguntas abiertas

- **Información extraída de la relatoría:** Además de las preguntas planteadas en el cuestionario del grupo focal, durante la sesión de debate los participantes expresaron verbalmente sus opiniones y comentarios sobre la propuesta. A continuación, se listan las oportunidades de mejora que pudieron extraerse de la relatoría de la sesión de debate y que no habían sido incluidas en las respuestas a las preguntas abiertas del cuestionario:
  - Dejar clara la definición de agilidad que se está usando.
  - El tipo de indicador puede causar ruido al momento de la evaluación, lo anterior debido a que para algunas organizaciones un indicador específico puede ser considerado como producto y en otra organización el mismo indicador puede ser considerado como actividad. En realidad, el tipo de indicador no tiene una incidencia alta sobre la evaluación así que se podría considerar eliminarlo.
  - Aclarar el alcance de AgilityRef. No se deja explícito si solamente permite evaluar procesos de desarrollo de software. Al usar el término proceso software podría referirse también a la gestión de proceso de desarrollo, etc.

#### 4.1.6. Acciones de mejora

A partir de los resultados del grupo focal, se analizaron las sugerencias realizadas por los participantes con el fin de determinar si se considerarían como oportunidades de mejora para el modelo. A continuación, en la Tabla 4.8, por cada comentario se describe el cambio realizado o la justificación de por qué no se realizó el cambio.

Comentario	Acción de mejora / justificación
<p>Considero que se deberían definir indicadores más genéricos que puedan dar cabida a las actividades que se realizan en los diferentes enfoques ágiles. Por ejemplo:</p> <p>Reunión incluye: reunión de iniciación, reunión diaria, reunión de seguimiento, reunión de lanzamiento. Gestión del cambio incluye: requisitos, producto, etc.</p>	<p>En efecto, tener indicadores más genéricos permitiría dar cabida a las distintas reuniones que pueden implementarse en un proceso de desarrollo de una organización. Sin embargo, en AgilityRef se considera que las dos reuniones más importantes para aportar a la implementación de los principios ágiles en el proceso de desarrollo son la reunión diaria y la reunión de retrospectiva. Por esta razón, se mantienen solamente los aspectos de agilidad: reunión diaria y retrospectiva.</p>
<p>Los indicadores deben tener nombres y descripciones más generales y no amarradas a una fuente específica ya que las organizaciones que no usen esas mismas fuentes no se van a sentir identificadas con el identificador propuesto. Por ejemplo, para el indicador reunión diaria, si mi organización no usa Scrum puede no calzar a ese identificador.</p>	<p>Se considera la reunión diaria como una actividad importante para aportar a la implementación de los principios ágiles. Independientemente de que la organización no use Scrum, es importante que el equipo se reúna diariamente para hacer seguimiento al avance del trabajo. El nombre que la organización le asigne a esa actividad puede ser distinto pero el objetivo de la reunión debe ser el evaluar el progreso hacia el objetivo de la iteración.</p>
<p>Agregar relaciones entre principios y valores: PA_01 -&gt; VA_1, PA_02 -&gt; VA_1, PA_08 -&gt; VA_3, PA_09 -&gt; VA_2, PA_11 -&gt; VA_2.</p>	<p>PA_01 -&gt; VA_1: No se tiene en cuenta porque el principio PA_01 está centrado en la entrega temprana y continua de software, y el valor VA_1 se refiere a las interacciones entre los individuos.</p> <p>PA_02 -&gt; VA_1: No se tiene en cuenta porque el principio PA_02 está centrado en la entrega temprana y continua de software, y el valor VA_1 se refiere a las interacciones entre los individuos.</p> <p>PA_08 -&gt; VA_3: No se tiene en cuenta porque el principio PA_08 hace referencia a mantener un ritmo constante, y el valor VA_3 se refiere a la colaboración con el cliente.</p>

	<p>PA_09 -&gt; VA_2: Se mantiene la relación de PA_09 con VA_1 debido a que PA_09 menciona la atención que se debe poner a la excelencia técnica y al buen diseño por parte del equipo, y aunque se espera que lo anterior incida en la generación de un producto software que funcione (valor VA_02), se considera que tiene una relación más directa con VA_1 debido a que este valor está centrado en la importancia de las habilidades comunicativas, técnicas y de autoorganización del equipo.</p> <p>PA_09: La atención continua a la excelencia técnica y al buen diseño mejora la agilidad VA_2: Software funcionando</p> <p>PA_11 -&gt; VA_2: No se tiene en cuenta porque el principio PA_11 se refiere a la autoorganización de los equipos, y el valor VA_2 está centrado en la importancia del software que funciona.</p>
<p>Adicionar tipo de indicador para las reglas de negocio que en un proceso son condicionales y que siempre se ligan con reglas de negocio que dan valor al usuario.</p>	<p>No se agregan nuevos tipos de indicadores debido a que se estableció que los tipos corresponden a los elementos de proceso identificados en OntoAgile.</p>
<p>Me gustaría que se pudiera hacer un mapeo entre otras actividades propuestas y las estándar de los modelos ágiles.</p>	<p>Es interesante contar con un mapeo entre las actividades que proponen los enfoques ágiles y las actividades del proceso de desarrollo implementadas en una organización pero no hace parte del alcance del proyecto.</p>
<p>Consideraría realizar un agrupamiento en la tabla de indicadores. Lo anterior dado que algunas descripciones podrían agrupar más fuentes. Ejemplo: En la tabla 3 en el identificador IA_15: "Se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los clientes" está relacionada con XP,</p>	<p>La propuesta de agrupar las fuentes es interesante en el sentido de poder analizar los aspectos de agilidad que aporta cada enfoque. Sin embargo, lo anterior no aporta valor al momento de evaluar la agilidad del proceso de desarrollo, por lo queda por fuera del alcance. Por otro lado, en la guía de Scrum se hace referencia al concepto</p>

<p>además de esta fuente se podría incluir Scrum dado que al igual que XP propone un lenguaje común para todo el equipo. Lo mismo para otras actividades como: IA_20, IA_21. Aunque estas últimas están relacionadas más bien hacia el cómo y no están planteadas en la guía de Scrum. En tal caso sería interesante diferencias las fuentes.</p>	<p>compartido por equipo de lo que significa “Terminado”, sin embargo no se hace referencia explícita a una visión compartida entre el equipo y el cliente.</p>
<p>Se debería pensar el término “indicador de agilidad”, se podría cambiar por “elemento de agilidad”.</p>	<p>Se cambia el nombre del componente “Indicador de agilidad” por “Aspecto de agilidad”.</p>
<p>Los indicadores sirven mucho pero deberían ser más genéricos para que puedan definir las métricas.</p>	<p>Se considera que si los aspectos de agilidad son demasiado genéricos será complicado definir indicadores que permitan conocer el grado de cumplimiento de un principio ágil en particular. Por ejemplo, si se define un aspecto de agilidad llamado “Reuniones”, podría considerarse que con el hecho de realizar reuniones durante el proceso de desarrollo ya se está aportando a la agilidad. Mientras que al definir aspectos de agilidad para reuniones diarias de seguimiento y reuniones de retrospectiva se puede especificar mejor qué características deben tener las reuniones del equipo para aportar al cumplimiento de los principios ágiles.</p>
<p>¿Qué pasaría con el modelo en caso que una organización cuente con sus propios valores ágiles y con los cuales viene trabajando su concepto de ser ágiles?</p>	<p>Cada organización tiene características y contextos particulares, por esta razón es válido que cada una de ellas centre su atención en los aspectos que más le interesen. AgilityRef le brinda una visión global de los aspectos que debe tener su proceso de desarrollo para cumplir los principios ágiles, de esta manera la organización puede evaluarse y tomar decisiones desde una perspectiva más global con respecto a la agilidad. Por otro lado, es cierto que los valores pueden ser distintos de acuerdo al enfoque ágil que se implemente en la organización. Por ejemplo, en Scrum se consideran los valores de compromiso, coraje, foco, apertura y respeto. Sin embargo, se observa que estos valores están alineados con los principios y valores del manifiesto ágil.</p>
<p>Los “indicadores de agilidad” podrían considerarse como “aspectos</p>	<p>Se cambia el nombre del componente “Indicador de agilidad” por</p>

de agilismo a evaluar” que luego sí podrían pasar a ser indicadores cuando trabajen una medida en la siguiente etapa del proyecto. Lo anterior bajo la definición de la ontología de la medición)	“Aspecto de agilidad”.
Expresar los indicadores en términos de medidas. Por ejemplo, en lugar de "iterativo e incremental", % de iteraciones que tuvieron entregas en la fecha establecida.	Los indicadores serán definidos en una etapa posterior del proyecto (método de evaluación)
Dejar clara la definición de agilidad que se está usando	Se aclara en el documento que en AgilityRef se considera que la agilidad de un proceso de desarrollo de software consiste en la implementación de elementos de proceso que aporten al cumplimiento de los principios ágiles.
El tipo de indicador puede causar ruido al momento de la evaluación, lo anterior debido a que para algunas organizaciones un indicador específico puede ser considerado como producto y en otra organización el mismo indicador puede ser considerado como actividad. En realidad, el tipo de indicador no tiene una incidencia alta sobre la evaluación así que se podría considerar eliminarlo.	Debido a que para la evaluación de la agilidad no se usará directamente AgilityRef si no el método de evaluación con sus respectivas preguntas, se decide mantener el tipo de indicador (ahora aspecto) para tener trazabilidad del elemento de proceso asociado.
Aclarar el alcance de AgilityRef. No se deja explícito si solamente permite evaluar procesos de desarrollo de software. Al usar el término proceso software podría referirse también a la gestión de proceso de desarrollo, etc.	Se deja explícito que se hace referencia al proceso de desarrollo de software.

Tabla 4.8. Acciones de mejora definidas para AgilityRef



## 4.2. Estudio de caso

En este capítulo se presenta la evaluación de AgilityPval mediante su aplicación en una organización desarrolladora de software como estudio de caso, para esto, se han seguido los lineamientos de estudios de caso propuestos en [28]. A continuación, se muestra en detalle: i) el diseño del estudio de caso, ii) el sujeto de estudio, iii) el procedimiento de campo y de recolección de datos, iv) el análisis de resultados, iv) el análisis de validez, y v) las limitaciones del estudio.

### 4.2.1. Diseño del estudio de caso

Con el fin de guiar la realización del estudio de caso, se definió la siguiente pregunta de investigación principal:

**¿El método de evaluación AgilityPval es comprensible, útil y práctico para llevar a cabo la evaluación de la agilidad de los procesos de desarrollo de software en las organizaciones?**

A partir de esta pregunta, se definieron las siguientes preguntas específicas:

- ¿Los componentes de AgilityPval (instrumento de evaluación, proceso de evaluación y roles) son de fácil comprensión?
- ¿Los resultados obtenidos mediante la aplicación AgilityPval proporcionan información útil a las organizaciones para mejorar la agilidad de sus procesos de desarrollo de software?
- ¿La aplicación de AgilityPval requiere de un nivel de esfuerzo aceptable para las organizaciones?

Teniendo en cuenta la clasificación presentada en [28], el diseño del estudio de caso fue holístico con una única unidad de análisis, la cual correspondió a la aplicación del método de evaluación AgilityPval en una organización desarrolladora de software. Las medidas usadas en el estudio fueron: Las oportunidades de mejora identificadas para AgilityPval, el esfuerzo requerido para aplicar AgilityPval en la organización, y los beneficios percibidos por la organización con relación a la aplicación de AgilityPval.

#### 4.2.2. Sujeto de estudio

A continuación, se describen los criterios que se tuvieron en cuenta para seleccionar la organización donde se llevó a cabo el estudio de caso:

- Es una organización desarrolladora de software.
- La organización está interesada en evaluar la agilidad de su proceso de desarrollo de software.
- La organización ha adoptado prácticas ágiles provenientes preferiblemente de varios enfoques ágiles.

A partir de los criterios descritos anteriormente, se seleccionó una organización desarrolladora de software del ámbito local, con 17 empleados y 8 años en el mercado nacional. Las principales áreas de actividad de la organización son: i) outsourcing en capacidad operativa (capacidad extendida), ii) desarrollo a la medida (fábrica de software), y iii) estrategias ecommerce (Implementación de tiendas en línea). Con respecto a los modelos y/o estándares implementados por la organización se destaca el uso de prácticas ágiles de metodologías y frameworks como: i) Scrum (Gestión de proyecto), ii) Lean Startup (Validación temprana de idea de negocio internas y de clientes), XP (desarrollo de software), Kanban (Seguimiento de proyectos / actividades / asignaciones). Asimismo, la organización está en proceso de implementación de la norma ISO/IEC 29110.

#### 4.2.3. Procedimiento de campo y recolección de datos

El procedimiento de campo y de recolección de datos está ampliamente relacionado con el proceso de evaluación propuesto en AgilityPval, el cual se describió de manera detallada en el capítulo anterior. En las siguientes secciones se describen las actividades ejecutadas en la organización con el fin de aplicar AgilityPval: i) planificación de la evaluación, ii) ejecución de la evaluación, y iii) generación y reporte de resultados.

- **Planificación de la evaluación:** Con el objetivo de planificar la realización del estudio de caso, se estableció un primer contacto telefónico con el gerente de la organización con el fin de exponer el objetivo del estudio, las actividades a realizar, aclarar dudas y establecer una fecha para realizar la reunión de inicio del

estudio. La reunión de inicio tuvo una duración de una hora y se desarrolló con la siguiente agenda:

- Presentación del proyecto y del equipo investigador.
  - Presentación general del modelo de referencia - AgilityRef.
  - Presentación detallada del método de evaluación- AgilityPval.
  - Presentación y firma del documento: Aceptación de participación en el estudio de caso (ver anexo D).
  - Asignación de roles. En este punto de la reunión, el gerente de la organización asumió el rol de responsable del proceso y evaluador debido a su amplio conocimiento en el proceso de desarrollo de la organización y a su interés en el desarrollo de software guiado por principios y valores ágiles.
  - Cronograma de evaluación. Por motivos de disponibilidad del gerente, se acordó que la evaluación se realizaría en una fecha posterior y se harían llegar los resultados al equipo investigador.
  - Entrega del instrumento de evaluación y documentación de soporte.
- **Ejecución de la evaluación:** El evaluador contestó las preguntas del instrumento de evaluación proporcionado en formato de hoja de cálculo siguiendo las instrucciones del método de evaluación AgilityPval. El tiempo aproximado requerido para la evaluación fue de treinta minutos.
  - **Generación y reporte de resultados:** Teniendo en cuenta que el instrumento de evaluación se elaboró en formato de hoja de cálculo, fue posible incluir fórmulas que calcularan automáticamente el grado de cumplimiento de cada aspecto de agilidad, principio ágil y valor ágil, así como el grado de agilidad del proceso de desarrollo. Además, los gráficos de resultados fueron generados automáticamente. El instrumento de evaluación diligenciado puede ser consultado en el anexo D.

A partir de las respuestas del cuestionario de evaluación, se calculó el grado de cumplimiento de los aspectos de agilidad. Los resultados del grado de cumplimiento de cada aspecto de agilidad se muestran en la Tabla 4.9. En la Figura 4.2, se muestran los resultados gráficamente. Los aspectos de agilidad con una calificación de “No alcanzado” fueron: AA\_06 (Autoorganización del

equipo), AA\_08 (Ritmo constante), AA\_18 (Integración continua), AA\_19 (Reglas técnicas), AA\_20 (Uso de métricas), y AA\_22 (Pruebas automáticas).

<b>IdAA.</b>	<b>Grado de cumplimiento</b>	<b>Calificación</b>
AA_01	67%	Ampliamente alcanzado
AA_02	100%	Completamente alcanzado
AA_03	100%	Completamente alcanzado
AA_04	50%	Parcialmente alcanzado
AA_05	100%	Completamente alcanzado
AA_06	0%	No alcanzado
AA_07	50%	Parcialmente alcanzado
AA_08	0%	No alcanzado
AA_09	100%	Completamente alcanzado
AA_10	50%	Parcialmente alcanzado
AA_11	100%	Completamente alcanzado
AA_12	50%	Parcialmente alcanzado
AA_13	100%	Completamente alcanzado
AA_14	100%	Completamente alcanzado
AA_15	100%	Completamente alcanzado
AA_16	50%	Parcialmente alcanzado
AA_17	100%	Completamente alcanzado
AA_18	0%	No alcanzado
AA_19	0%	No alcanzado
AA_20	0%	No alcanzado
AA_21	100%	Completamente alcanzado
AA_22	0%	No alcanzado

Acrónimos utilizados: IdAA: Identificador del Aspecto de agilidad.

Tabla 4.9. Resultados grado de cumplimiento de los aspectos de agilidad en el estudio de caso.

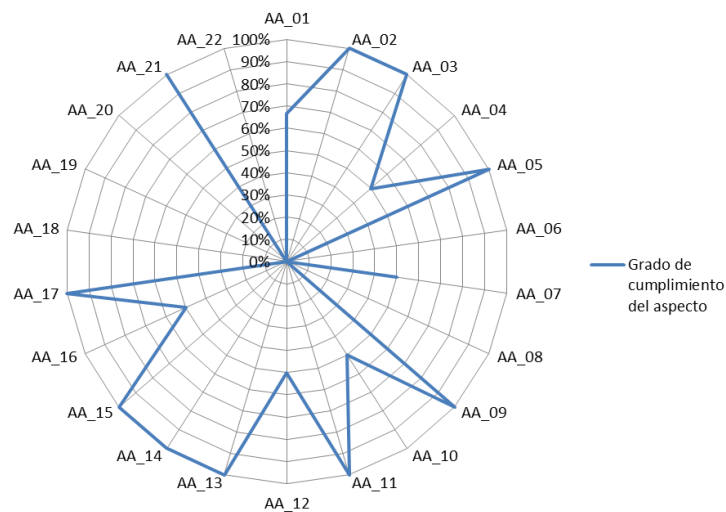


Figura 4.2 Gráfico de resultados del grado de cumplimiento de cada aspecto de agilidad.

Una vez calculado el grado de cumplimiento de cada aspecto de agilidad, se calculó el grado de cumplimiento de los principios ágiles. Los resultados obtenidos se pueden observar en la Tabla 4.10. En la Figura 4.3, se muestra una gráfica del porcentaje de cumplimiento de cada principio ágil.

Al analizar los resultados se destaca que el principio PA\_02 obtuvo una calificación de “Completamente alcanzado” con un 100% de cumplimiento. Por otro lado, el principio PA\_08 obtuvo una calificación de “No alcanzado” con 0% de cumplimiento.

<b>IdPA.</b>	<b>Grado de cumplimiento</b>	<b>Calificación</b>
PA_01	60%	Ampliamente alcanzado
PA_02	100%	Completamente alcanzado
PA_03	53%	Ampliamente alcanzado
PA_04	80%	Ampliamente alcanzado
PA_05	63%	Ampliamente alcanzado
PA_06	50%	Parcialmente alcanzado
PA_07	50%	Parcialmente alcanzado
PA_08	0%	No alcanzado
PA_09	33%	Parcialmente alcanzado
PA_10	63%	Ampliamente alcanzado
PA_11	67%	Ampliamente alcanzado

PA_12	33%	Parcialmente alcanzado
-------	-----	------------------------

Acrónimos utilizados: IdPA: Identificador del Principio ágil.

Tabla 4.10. Resultados grado de cumplimiento de los principios ágiles en el estudio de caso.

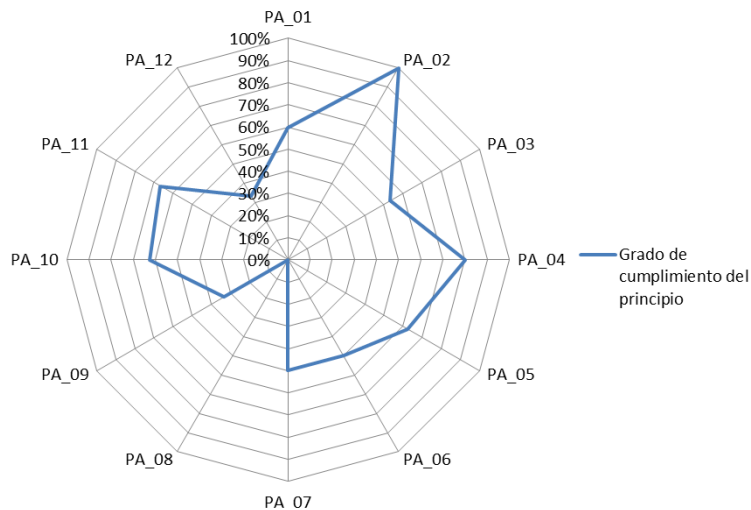


Figura 4.3 Gráfico de resultados del grado de cumplimiento de cada principio ágil.

Luego de calcular el grado de cumplimiento de los principios ágiles, se calculó el grado de cumplimiento de los cuatro valores ágiles. En la Tabla 4.11, se muestra el grado de cumplimiento de cada valor ágil. En la Figura 4.4 se muestran los resultados mediante una gráfica.

En los resultados, se destaca que el valor VA\_4 (respuesta ante el cambio sobre seguir un plan) obtuvo una calificación de “Completamente alcanzado” con el 100%, y el valor VA\_1 (individuos e interacciones sobre procesos y herramientas) obtuvo una calificación de “Parcialmente alcanzado” con un porcentaje de cumplimiento del 44%.

IdVA.	Grado de cumplimiento	Calificación
VA_1	44%	Parcialmente alcanzado
VA_2	54%	Ampliamente alcanzado
VA_3	80%	Ampliamente alcanzado
VA_4	100%	Completamente alcanzado

Acrónimos utilizados: IdVA: Identificador del Valor ágil.

Tabla 4.11. Resultados grado de cumplimiento de los valores ágiles en el estudio de caso.

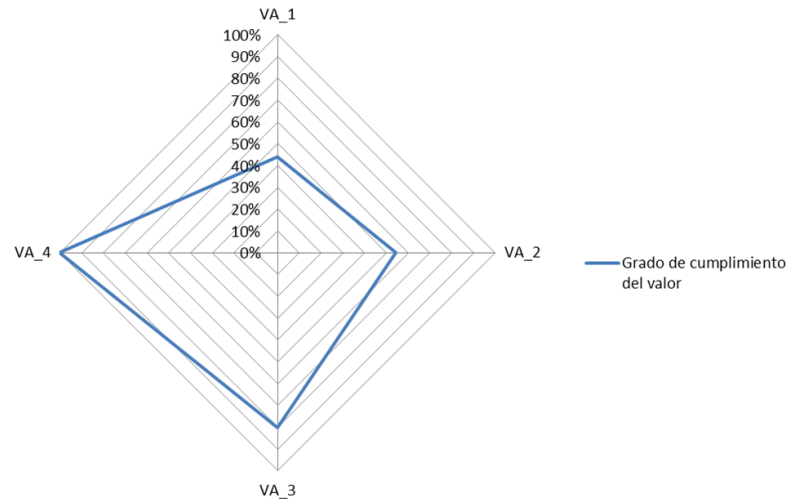


Figura 4.4 Gráfico de resultados del grado de cumplimiento de cada valor ágil.

Una vez calculados los grados de cumplimiento de cada valor ágil, se calculó el grado de agilidad del proceso, obteniendo un resultado de 70%, lo cual corresponde a la calificación de ampliamente alcanzado.

Como se mencionó en la descripción de AgilityPval, los indicadores de agilidad que no se están cumpliendo completamente en el proceso de desarrollo, van acompañados de un comentario que justifica el valor asignado al indicador. Dichos comentarios, son oportunidades de mejora a partir de las cuales se pueden definir acciones de mejora para incrementar la agilidad del proceso de desarrollo de la organización. En Tabla 4.12, se muestran las oportunidades de mejora identificadas.

IdIA	Indicador de agilidad	Oportunidad de mejora
IA_03	Una iteración solamente inicia cuando ha terminado la iteración anterior	No al 100%. En algunos proyectos a la medida cuando el cliente final es otra empresa de software, es frecuente que el cliente tenga ambiente de preproducción y producción. De esta forma, el equipo de desarrollo entrega en el primero de estos ambientes, mientras el cliente hace pruebas en producción. Por efectos de tiempo y costo no nos resulta rentable esperar a

		<p>que termine la totalidad de las pruebas para continuar.</p> <p>En los proyectos internos si finaliza una iteración antes de iniciar la siguiente.</p>
IA_07	<p>La duración de la reunión diaria se limita a un tiempo que permita a cada participante compartir de manera concisa lo que hizo el día anterior, lo que tiene planeado hacer ese día y los impedimentos para lograr el objetivo de la iteración.</p>	<p>Aunque se realizan los dailys de seguimiento, algunas veces se extiende más del tiempo necesario por trabajar regularmente con equipos distribuidos y resultan más temas de los esperados.</p>
IA_10	<p>Cada integrante del equipo puede elegir la mejor forma de llevar a cabo las tareas que tenga bajo su responsabilidad.</p>	<p>No se ha implementado esta opción hasta el momento en la empresa.</p>
IA_11	<p>El equipo tiene la posibilidad de dar mayor prioridad a los requisitos del producto que presenten más riesgo.</p>	<p>En común acuerdo con el cliente, se hace. No de manera unilateral por el equipo de desarrollo.</p>
IA_13	<p>Cualquier miembro del equipo podría realizar cambios en cualquiera de los componentes del producto.</p>	<p>Procuramos que la persona que desarrollo determinado componente, sea quien realice cambios en este.</p>
IA_14	<p>El tiempo de duración de las iteraciones es consistente durante todo el desarrollo del producto.</p>	<p>Ha sido difícil porque aún no tenemos una manera de medir velocidad del equipo y a veces no se cumplen al 100% los tiempos de estimación, viéndose esto reflejado en tiempos de desarrollo de una iteración</p>
IA_17	<p>En la reunión llevada a cabo al finalizar cada iteración, se identifican y ordenan los elementos más importantes que salieron bien y las posibles mejoras, se crea un plan para implementar las mejoras, y se adapta la lista de requisitos del producto en caso de ser necesario.</p>	<p>En los proyectos internos sí. En los proyectos a la medida cuando un cliente es una empresa de desarrollo de software, no hemos logrado realizar el ejercicio hasta el momento.</p>
IA_20	<p>El cliente o un representante suyo con poder de decisión, participa en el diseño de las pruebas funcionales del producto.</p>	<p>Algunas veces, pero es una tarea difícil con los clientes.</p>
IA_22	<p>El cliente o un representante suyo con poder de decisión, está disponible en la misma ubicación del equipo de desarrollo del producto para resolver inquietudes con respecto a los requisitos y prioridades para el negocio.</p>	<p>Por nuestra modalidad de trabajo remoto, no se ha presentado esta oportunidad.</p>



IA_26	Durante el diseño del producto se tiene en cuenta que no se dupliquen componentes, se incluyen todos los aspectos que el equipo considera importantes y se incluyen solamente lo que se requiere en el momento.	Se busca que solo se incluyan los aspectos necesarios, pero al momento de la ejecución pueden faltar o sobrar algunos componentes o artefactos.
IA_29	Después de unas pocas horas todos los cambios que se hacen al producto se prueban e integran.	Este tiempo no ha sido medido para establecer si el término "pocas horas" se cumple o no.
IA_30	Se acuerda un conjunto de reglas que todo el equipo debe cumplir al momento de agregar funcionalidades al producto o realizar modificaciones.	Se tiene el conjunto de reglas y condiciones, pero ha sido un proceso lograrlo con algunos miembros del equipo.
IA_31	Se usan métricas que aporten información relevante sobre el desempeño del equipo.	No se tiene y es una necesidad inmediata para la compañía.
IA_33	Se usan herramientas que permitan dar soporte a la ejecución de pruebas automáticas.	No se cuenta con las herramientas.

Acrónimos utilizados: IdIA: Identificador de Indicador de agilidad relacionado.

Tabla 4.12. Oportunidades de mejora encontradas en el estudio de caso.

#### 4.2.4. Análisis de resultados del estudio de caso

A continuación, se presenta el análisis de los resultados obtenidos mediante la realización del estudio de caso: i) esfuerzo, ii) análisis del método de evaluación AgilityPval, iii) análisis de validez del estudio de caso, y iv) limitaciones del estudio de caso.

- **Esfuerzo:** El esfuerzo requerido para aplicar AgilityPval en la organización correspondió a la suma de los tiempos de planificación y ejecución de la evaluación. La planificación requirió de una reunión de 60 minutos con la participación del gerente de la organización y un representante del equipo investigador, y la ejecución de la evaluación tuvo una duración de 30 minutos y fue realizada solamente por el gerente de la organización quien asumió el rol de evaluador. Gracias a la implementación del instrumento de evaluación en una hoja de cálculo, fue posible obtener de manera automática los resultados de la evaluación una vez el evaluador diligenció el cuestionario. De acuerdo a lo anterior, el tiempo total requerido para realizar la evaluación de la agilidad del proceso de desarrollo en la organización fue de 90 minutos.

Con los resultados obtenidos, es posible considerar que el esfuerzo requerido para aplicar AgilityPval fue razonable teniendo en cuenta las características de la organización involucrada en la evaluación. Asimismo, es posible afirmar que la aplicación de AgilityPval no causó efectos negativos o una sobrecarga considerable en las actividades diarias de los participantes en el estudio.

- **Análisis del método de evaluación AgilityPval:** Una vez finalizada la aplicación de AgilityPval, se realizó una encuesta al participante que asumió el rol de evaluador con el fin de conocer su opinión sobre la comprensibilidad, utilidad y practicidad el método de evaluación.

La primera parte del cuestionario permitió contar con un perfil general de la organización mediante la recolección de los siguientes datos: i) número de empleados, ii) años de existencia, iii) alcance del mercado, iv) actividades desarrolladas, y v) modelos y/o estándares usados.

La segunda parte del cuestionario se enfocó en conocer la opinión del participante sobre AgilityPval. Las primeras 7 preguntas, se diseñaron para ser respondidas mediante la siguiente escala de Likert de 5 puntos: 1) Totalmente en desacuerdo, 2) En desacuerdo, 3) Ni de acuerdo ni en desacuerdo, 4) De acuerdo, y 5) Totalmente de acuerdo.

Dichas preguntas, permitieron medir la comprensibilidad (3 preguntas), practicidad (2 preguntas) y utilidad (2 preguntas) del método propuesto. Por último, se incluyeron 4 preguntas abiertas que permitieron al participante proponer ajustes al método y realizar comentarios adicionales. A continuación, en la Tabla 4.13, se pueden observar las preguntas realizadas y las respuestas proporcionadas por el representante de la organización que asumió el rol de evaluador.

Aspecto a evaluar	Id.	Pregunta	Resp.
Comprensibilidad	P1	¿Considera que el proceso de evaluación de AgilityPval es de fácil comprensión?	5
	P2	¿Considera que el instrumento de evaluación de	4

		AgilityPval es de fácil comprensión?	
	P3	¿Considera que los resultados generados por AgilityPval son de fácil comprensión?	4
Practicidad	P4	¿Considera que AgilityPval puede ser aplicado en una organización independientemente de los modelos/estándares que guíen sus procesos?	4
	P5	¿Considera que el esfuerzo requerido para aplicar AgilityPval es razonable teniendo en cuenta las características de la organización?	5
Utilidad	P6	¿Considera que los resultados obtenidos con la aplicación de AgilityPval reflejan el estado real del proceso de desarrollo de software de la organización?	4
	P7	¿Considera que los resultados obtenidos con la aplicación de AgilityPval sirven de punto de partida para diseñar planes de mejora que permitan incrementar la agilidad de los procesos de desarrollo de software de la organización?	5

#### Preguntas abiertas

Id.	Pregunta
P8	<p>Según su experiencia, ¿Cuál es su percepción acerca del esfuerzo requerido para aplicar AgilityPval en la organización?</p> <p><b>Respuesta:</b> Es un esfuerzo necesario e importante porque la herramienta da un buen punto de partida para mejorar. Se requiere por supuesto de un esfuerzo y compromiso de todo el equipo de trabajo de la empresa.</p>
P9	<p>Según su experiencia, ¿Cuál es su percepción acerca de la utilidad de AgilityPval para las organizaciones que desarrollan software?</p> <p><b>Respuesta:</b> Muy importante. Regularmente las empresas se valoran de una manera muy subjetiva en si son ágiles o no y esta herramienta permite tener un resultado más objetivo y medible a lo que una empresa debe mejorar para convertirse realmente en una compañía que implementa el agilismo como filosofía de vida.</p>
P10	<p>¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityPval? Por favor, describa los cambios propuestos.</p> <p><b>Respuesta:</b> Podría tenerse en cuenta el tema de documentación, que, si bien el agilismo busca en su segundo valor documentar lo necesario, en el instrumento no encontré un indicador de agilidad al respecto.</p>
P11	<p>¿Tiene algún comentario adicional acerca de AgilityPval?</p> <p><b>Respuesta:</b> Es una herramienta fácil de utilizar, muy útil tanto para las empresas que ya se consideran ágiles al encontrar una manera objetiva de medirse, como para las que</p>

	<p>apenas inician el proceso de querer ser ágiles, pues tienen de una manera desglosada y entendible, los aspectos a tener en cuenta que quizá no es muy explícita en los principios y valores. Resultaría interesante tenerla en una herramienta más automatizada que permita al menos unas 3 veces al año medirse y hacer seguimiento de cómo va la empresa en su afán de ser ágil. Que muestre una analítica de resultados y sea más explícita en sus recomendaciones de mejora en pro de alcanzar cada vez más el objetivo.</p>
--	---

Acrónimos utilizados: Id: Identificador, Resp: Respuesta.

Tabla 4.13. Cuestionario de evaluación de AgilityPval

A partir del análisis de las respuestas del cuestionario de evaluación descrito anteriormente, fue posible obtener las siguientes conclusiones con respecto a AgilityPval:

- Con respecto a la comprensibilidad de AgilityPval, se considera que el proceso de evaluación, el instrumento de evaluación y los resultados generados son de fácil comprensión. Además, el instrumento de evaluación permite a las organizaciones con o sin experiencia en el desarrollo ágil, entender de manera detallada los aspectos que se deben tener en cuenta para ser ágiles.
- AgilityPval se considera práctico para la organización teniendo en cuenta que requiere de un esfuerzo razonable puede ser aplicado y es independiente de los modelos o estándares implementados en la organización.
- Los resultados obtenidos con AgilityPval son útiles para la organización debido a que reflejan el estado real del proceso de desarrollo de manera objetiva y medible, y pueden ser usados como punto de partida para diseñar planes de mejora que permitan incrementar la agilidad del proceso de desarrollo de la organización.
- Como recomendaciones de mejora para AgilityPval se pueden considerar: i) la inclusión de indicadores de agilidad relacionados con la documentación generada durante el proceso de desarrollo, y ii) la automatización de la evaluación con el fin de hacer un seguimiento del progreso de la organización mediante analítica de los resultados y generación de recomendaciones de mejora.

#### 4.2.5. Análisis de validez

Con el fin de analizar la confiabilidad de los resultados obtenidos en el estudio de caso, se realizó un análisis de validez de constructo, validez externa y confiabilidad.

- **Validez de constructo:** Este aspecto de validez refleja el grado en que las medidas operativas del estudio representan los aspectos que se pretende medir y que están reflejados en las preguntas de investigación. Con el fin de mantener la validez de constructo en este estudio, se realizó una reunión de inicio donde se presentó el contexto del proyecto, el método de evaluación, el plan de trabajo y se resolvieron inquietudes. Además, se proporcionó documentación de apoyo al evaluador para resolver inquietudes relacionadas con la aplicación del método de evaluación.
- **Validez externa:** Este aspecto de validez se refiere al grado de generalización de los resultados del estudio. Con respecto a este tipo de validez, no es posible generalizar los resultados obtenidos para todas las organizaciones que desarrollan software, sin embargo, es posible replicar el estudio en organizaciones con un perfil similar a la organización donde se llevó a cabo este estudio siguiendo los criterios de selección y procedimientos de campo y de recolección de datos establecidos en el diseño del estudio.
- **Confiabilidad:** En este aspecto se analiza la independencia que debe existir entre los resultados obtenidos y los investigadores que llevan a cabo el estudio. Con el fin de asegurar la confiabilidad de este estudio, se realizaron las siguientes actividades: i) definición de los criterios de selección de la organización, ii) definición de los procedimientos de campo y de recolección de datos, iii) elaboración de documentación detallada del método de evaluación para ser entregada a la organización, y (iv) soporte permanente a los participantes durante la ejecución del estudio.

#### 4.2.6. Limitaciones del estudio de caso

Las limitaciones consideradas en el estudio de caso son:

- La realización de un solo caso de estudio limita la capacidad de generalización de los resultados obtenidos. Es necesario aplicar AgilityPval en un número de mayor de estudios de caso y en organizaciones con diversas características.
- Sesgo en el estudio de caso con respecto a: i) la posible subjetividad del evaluador durante la aplicación del instrumento de evaluación, y ii) la posible subjetividad en la interpretación de los datos por parte del equipo investigador.



# Capítulo 5

## Conclusiones y trabajo futuro

En este capítulo se presentan y evalúan los resultados finales obtenidos a partir del desarrollo de esta investigación. En primer lugar, se describen las evidencias del trabajo realizado para cumplir los objetivos de la investigación. Luego, se presentan los artículos escritos y publicados a partir de los resultados de las actividades realizadas a lo largo del desarrollo de la investigación. Asimismo, se presentan las conclusiones de la investigación y los trabajos futuros.

### 5.1. Análisis de los objetivos de investigación

A continuación, se describen los objetivos de la investigación y los capítulos de la monografía que evidencian el trabajo realizado para dar cumplimiento a cada uno de ellos.

#### 5.1.1. Objetivos específicos - OE

- **OE1:** Definir una ontología de dominio específico que permita entender y organizar el conocimiento relacionado con los términos y relaciones alrededor de los procesos software y los valores y principios del manifiesto ágil.

Para lograr este objetivo, en primer lugar se realizó un mapeo sistemático de la literatura acerca de los trabajos relacionados con la evaluación de la agilidad en las organizaciones de desarrollo de software. A partir del análisis de los trabajos encontrados, se identificaron los métodos propuestos, los factores clave que influyen en la evaluación de la agilidad, y las brechas existentes. El mapeo sistemático se presenta en la sección 2.2.

En la sección 3.2, se presenta la ontología OntoAgile, la cual es una ontología de dominio que permite organizar el conocimiento relacionado con la implementación de los enfoques ágiles en los procesos software de las organizaciones, permitiendo que el conocimiento pueda ser reutilizado y compartido por los interesados en el área. Asimismo, OntoAgile soporta la evaluación de la agilidad de los procesos software a partir de la identificación de las relaciones entre los elementos de los procesos software y los principios y valores ágiles.

- **OE2:** Definir un modelo de referencia que establezca los elementos necesarios para llevar a cabo la evaluación de la agilidad de los procesos software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles.

En la sección 3.3, se presenta AgilityRef, el cual es un modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones, el cual está dividido en tres componentes principales: i) aspectos de agilidad, ii) principios ágiles, y iii) valores ágiles. Un aspecto de agilidad describe un elemento de proceso (actividad, rol, producto, herramienta) que evidencia la implementación de uno o más principios ágiles. Por su parte, los principios ágiles describen las características que debe tener un proceso ágil de desarrollo de software y soportan el cumplimiento de uno o más valores ágiles en el proceso de desarrollo.

En la sección 4.1, se presentan los resultados del grupo focal realizado con el fin de evaluar la primera versión del modelo de referencia AgilityRef. Mediante la realización del grupo focal se encontraron oportunidades de mejora y se refinó el modelo de referencia propuesto.

Teniendo en cuenta que AgilityRef es un modelo de referencia, no describe el proceso requerido para realizar la evaluación. En este sentido, en el capítulo 6 se presenta el método de evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityPval. Este método define un proceso de evaluación basado en PvalCOMPETISOFT, el cual es un proceso que describe de manera detallada la forma en que se debe llevar a cabo la evaluación de



procesos y se adapta a cualquier instrumento de evaluación que requiera evaluar características específicas de los procesos de una organización.

- **OE3:** Desarrollar un prototipo de una aplicación web que permita a las organizaciones evaluar la agilidad de sus procesos software con base en el modelo de referencia propuesto.

En la sección 3.5, se presenta el prototipo de una aplicación web para soportar la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones llamada AgilityProTool, la cual está basada en el método de evaluación AgilityPval, descrito en la sección 3.4. AgilityProTool permite a las organizaciones realizar la evaluación de la agilidad de su proceso de desarrollo.

- **OE4:** Evaluar el modelo propuesto a través de su aplicación en una organización desarrolladora de software mediante un estudio de caso.

En la sección 4.2, se presenta la evaluación de AgilityPval mediante su aplicación en una organización desarrolladora de software como estudio de caso. Dicho estudio tuvo como objetivo determinar si AgilityPval es comprensible, útil y práctico para llevar a cabo la evaluación de la agilidad de los procesos de desarrollo de software en las organizaciones.

### 5.1.2. Objetivo general - OG

**OG:** Definir un framework para llevar a cabo la evaluación de la agilidad de los procesos software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles.

En el capítulo 3, se describe el “Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones - AgilityPRO”, el cual tiene como objetivo soportar la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones mediante el análisis del cumplimiento de los principios y valores ágiles. AgilityPRO, es el resultado de la integración de los resultados del trabajo realizado para conseguir cada uno de los objetivos específicos de la investigación.

## 5.2. Publicaciones

Los resultados relacionados con el mapeo sistemático de la literatura y con la definición de la ontología para el desarrollo ágil de software, fueron publicados en revistas indexadas.

- Artículo publicado en una revista indexada categoría C según PUBLINDEX de Colciencias con la siguiente referencia: W. A. Ortega-Ordoñez, C. J. Pardo-Calvache, and F. J. Pino-Correa, "Systematic mapping on the agility evaluation in software development organizations," ITECKNE, vol. 16, no. 1, pp. 64–76, 2019.
- Artículo publicado en una revista indexada categoría A2 según PUBLINDEX de Colciencias con la siguiente referencia: W. A. Ortega-Ordoñez, C. J. Pardo-Calvache, and F. J. Pino-Correa, "OntoAgile : an ontology for agile software development processes," DYNA, vol. 86, no. 209, pp. 86–97, 2019.
- Actualmente, se cuenta con un artículo escrito y en espera de ser enviado a revisión en una revista indexada categoría A2 con los resultados del estudio de caso.

## 5.3. Conclusiones

- El mapeo sistemático de la literatura acerca de la evaluación de la agilidad en organizaciones desarrolladora de software, permitió evidenciar: i) la inexistencia de una solución de evaluación de agilidad de los procesos de desarrollo de software que sea ampliamente aceptada y aplicada en la industria, ii) la falta de consenso en la definición de agilidad en los procesos de desarrollo de software y en los criterios necesarios para evaluar la agilidad, iii) problemas de ambigüedad en los términos relacionados con los enfoques ágiles y la evaluación de la agilidad, iv) falta de claridad en cuanto a las relaciones existentes entre los principios y valores ágiles y los elementos (actividades, roles, productos y herramientas) de los procesos software, y v) la inexistencia de una herramienta software que soporte la evaluación de la agilidad de los procesos de desarrollo de software de las organizaciones.

- La definición de la ontología denominada OntoAgile, aportó a la organización del conocimiento relacionado con la implementación de los enfoques ágiles en los procesos software de las organizaciones de una manera genérica. Asimismo, OntoAgile permitió soportar la definición de un modelo de referencia para la evaluación de la agilidad de los procesos de desarrollo de software, al identificar las relaciones entre los elementos de los procesos de desarrollo de software y los principios y valores ágiles. OntoAgile fue implementada usando el lenguaje OWL (disponible en <http://bit.ly/owl-ontoagile>) con el fin de facilitar su uso por parte de la comunidad académica y las organizaciones desarrolladoras de software.
- El uso de la metodología REFSENO para la definición de OntoAgile, permitió obtener una ontología con una representación adecuada para el área de ingeniería de software debido a que los conceptos y relaciones fueron presentados con un formato sencillo basado en tablas y se obtuvo una representación gráfica de la ontología usando el lenguaje UML.
- El uso de distintas estrategias de evaluación de ontologías permitió comprobar que OntoAgile: i) tiene una estructura consistente, permite instanciar los principales elementos del enfoque ágil Scrum, iii) permite soportar la evaluación de la agilidad de los procesos de desarrollo de software basada en el cumplimiento de los principios ágiles, iv) puede ser usada como fuente de datos para una aplicación Web que use el Protocolo y lenguaje de consulta RDF (SPARQL), y v) es capaz de responder a las Preguntas de Competencia definidas durante su etapa de diseño. Sin embargo, la evaluación realizada no incluyó el uso de la ontología para el desarrollo de una aplicación que por medio del uso de algoritmos de inferencia, se utilizará para instanciar los procesos de desarrollo en organizaciones software de distintos tamaños, y que su vez, implementen diferentes tipos de enfoques ágiles. En este sentido, y con el objetivo de profundizar en el desarrollo de un sistema autónomo en la valoración del cumplimiento de la agilidad de las organizaciones software, como trabajo futuro se espera desarrollar un sistema de inferencia que basado en la ontología propuesta permita conocer el estado actual de una organización con relación a su agilidad, este sistema sería de gran ayuda para consultores, organizaciones y la comunidad ágil en general.

- La realización de un grupo focal para evaluar el modelo de referencia propuesto, permitió reunir a expertos en el área de ingeniería de software y enfoques ágiles, quienes desde su conocimiento y experiencia identificaron oportunidades de mejora en la propuesta. La utilización de este método de investigación, aportó formalismo a la evaluación, y propició la sinergia y comparación de experiencias y puntos de vista entre los participantes. Como resultado del grupo focal, se realizaron ajustes al modelo de referencia obteniendo una versión mejorada de la propuesta antes de pasar a la etapa de definición del método de evaluación.
- La realización de un estudio de caso, permitió aplicar el método de evaluación AgilityPval en una organización desarrolladora de software con el fin de evaluar su comprensibilidad, utilidad y practicidad como método de evaluación de la agilidad de los procesos de desarrollo de las organizaciones. De acuerdo a los resultados obtenidos, AgilityPval permitió conocer el grado de agilidad del proceso de desarrollo de la organización sujeto de estudio. Además, fue considerada como una solución de fácil comprensión, práctica y de utilidad para la organización.
- El estudio de caso fue diseñado teniendo en cuenta estrategias para conseguir la validez de constructo, validez interna, y confiabilidad. Sin embargo, es importante tener en cuenta que el estudio de caso realizado presenta las siguientes limitaciones: i) no es posible generalizar los resultados debido a que se debe aplicar AgilityPval en un número mayor de organizaciones con diversas características, por ejemplo, organizaciones con enfoques híbridos, enfoques escalados y equipos distribuidos, y ii) la existencia de sesgo en el estudio de caso con respecto a la posible subjetividad del evaluador durante la aplicación del instrumento de evaluación, y la posible subjetividad en la interpretación de los datos por parte del equipo investigador.
- El prototipo de la aplicación Web desarrollado, permite a las organizaciones evaluar la agilidad de sus procesos de desarrollo usando el método de evaluación AgilityPval. Para el diseño del prototipo, se eligió el patrón de arquitectura de software Modelo-Vista-Controlador – MVC, el cual facilita el mantenimiento y la escalabilidad gracias al bajo acoplamiento entre sus componentes. Como trabajo

futuro, se espera que la aplicación Web pueda ser utilizada por organizaciones, esto permitirá no sólo conocer el estado actual de la agilidad de empresas de la región, sino, también tener una perspectiva de país y por qué no, a nivel mundial. Asimismo, permitirá extender el número de estudios de caso que permitan seguir mejorando la propuesta.

- Como respuesta a la pregunta de investigación formulada al principio de esta investigación: “¿Cómo determinar que un proceso software es ágil de acuerdo a un referente internacional como los valores ágiles y principios del manifiesto ágil?”, se obtuvo como resultado el “Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones - AgilityPRO”, el cual consta de tres capas: i) conceptual, ii) metodológica, y iii) tecnológica. En la capa conceptual, define un modelo de referencia que describe un conjunto de aspectos de agilidad que permite relacionar los elementos de un proceso de desarrollo de software con los principios y valores ágiles. Las relaciones descritas surgieron a partir de la definición de la Ontología para procesos de desarrollo ágil de software – OntoAgile. En la capa metodológica, define un método de evaluación que describe el proceso necesario para evaluar la agilidad en los procesos de desarrollo de software de las organizaciones. En la capa tecnológica, proporciona un prototipo de una aplicación Web para soportar la evaluación basada en el método de evaluación propuesto.
- La definición de AgilityPro se basó en los valores y principios expresados en el manifiesto ágil, el cual sigue siendo el documento de referencia de la comunidad ágil a nivel internacional. Sin embargo, se hace necesario revisar en detalle cada uno de los principios y valores ágiles con el fin de determinar su vigencia, lo anterior, teniendo en cuenta las tendencias en la industria software actual como: los enfoques escalados, los enfoques híbridos, DevOps, equipos distribuidos, integración y despliegue continuo, entre otras.

## 5.4. Trabajo futuro

A continuación, se presentan los trabajos futuros relacionados con esta investigación:

- **Actualización del mapeo sistemático:** Debido a que el mapeo sistemático de la literatura acerca de la evaluación de la agilidad en las organizaciones desarrolladoras de software fue realizada al inicio del proyecto, es necesario realizar una actualización que permita encontrar nuevas propuestas y líneas de investigación en el tema de la evaluación de la agilidad.
- **Actualización de OntoAgile:** La ontología propuesta debe ser evaluada mediante la realización de estudios de caso en organizaciones que usen distintos enfoques ágiles con el fin de instanciar sus procesos de desarrollo de software. De acuerdo a los resultados obtenidos, se debe evaluar la necesidad de actualizar la ontología y extender sus conceptos y relaciones.
- **Realización de nuevos estudios de caso:** Debido a que en esta investigación se realizó solamente un estudio de caso para evaluar la propuesta, es necesario realizar nuevos estudios de caso que permitan aplicar el método de evaluación en organizaciones con diversas características.
- **Automatización de la evaluación:** Actualmente, el método de evaluación es un proceso manual basado en un cuestionario. Una mejora considerable, sería la implementación de algoritmos que permitieran realizar una evaluación automática y periódica de la agilidad del proceso de desarrollo de la organización.

## Bibliografía

- [1] O. Ozcan-Top and O. Demirörs, “A reference model for software agility assessment: AgilityMod,” in *Communications in Computer and Information Science*, 2015, vol. 526, pp. 145–158.
- [2] K. Beck *et al.*, “Manifesto for Agile Software Development,” 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed: 01-Jun-2018].
- [3] H. Takeuchi and I. Nonaka, “The New New Product Development Game,” *Harvard Business Review*, vol. 64, no. 1, pp. 137–146, 1986.
- [4] C. A. K.Beck, *Extreme Programming Explained*. Pearson Education, 1999.
- [5] A. Cockburn, *Crystal Clear: A Human-Powered Methodology for Small Teams*. Pearson Education, 2004.
- [6] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit (The Agile Software Development Series)*. 2003.
- [7] J. A. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, vol. 12. Dorset House Publishing Co Inc., 2000.
- [8] J. Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley Professional, 1997.
- [9] S. R. Palmer and M. Felsing, *A Practical Guide to Feature Driven Development*. Prentice Hall, 2002.
- [10] S. W. Ambler, “Agile Unified Process.” [Online]. Available: <http://www.ambysoft.com/unifiedprocess/agileUP.html>. [Accessed: 01-Jun-2018].
- [11] “KanBan Fundamentals.” [Online]. Available: <http://www.scrumhub.com/kanban-fundamentals/>. [Accessed: 01-Jun-2018].
- [12] C. Larman, *Agile and Iterative Development: A Manager’s Guide*, 01 ed. 2003.
- [13] VersionOne Inc., “The 13th annual State of Agile Report,” 2019. [Online]. Available: <https://www.stateofagile.com>. [Accessed: 01-Oct-2019].
- [14] A. S. Campanelli and F. S. Parreiras, “Agile methods tailoring - A systematic literature review,” *Journal of Systems and Software*, vol. 110, pp. 85–100, 2015.
- [15] M. Karpudewan and C. C. Keong, “Agile transformation: A multi-dimensional process,” *Jurnal Teknologi*, vol. 2, pp. 1–6, 2015.
- [16] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and M. Z. Nafchi, “Obstacles in moving to agile software development methods; At a Glance,” *Journal of Computer Science*, vol. 9, no. 5, pp. 620–625, 2013.
- [17] S. Nerur, R. Mahapatra, and G. Mangalaraj, “Challenges of migrating to agile

- methodologies,” *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.
- [18] C. Tolfo, R. S. Wazlawick, M. G. G. Ferreira, and F. A. Forcellini, “Agile methods and organizational culture: Reflections about cultural levels,” *Journal of Software Maintenance and Evolution*, vol. 23, no. 6, pp. 423–441, 2011.
- [19] A. Sidky, J. Arthur, and S. Bohner, “A disciplined approach to adopting agile practices: The agile adoption framework,” *Innovations in Systems and Software Engineering*, vol. 3, no. 3, pp. 203–216, 2007.
- [20] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner’s Guide to Agile Software Delivery in the Enterprise*. 2012.
- [21] SEI, “Capability Maturity Model for Software - CMMI for Development V1.3,” Pittsburg, Pennsylvania, USA, 2010.
- [22] C. Pardo, F. García, F. Pino, M. Piattini Velthuis, and M. T. Baldassarre, “A reference ontology for harmonizing process-reference models,” *Revista Facultad de Ingeniería, Universidad de Antioquia*, no. 73, pp. 29–42, 2014.
- [23] K. Beck, *Extreme Programming Explained*. Addison-Wesley Professional, 1999.
- [24] P. Ceravolo, E. Damiani, M. Marchesi, S. Pinna, and F. Zavatarelli, “A ontology-based process modelling for XP,” in *Proceedings of the Tenth Asia-Pacific Software Engineering Conference*, 2004, pp. 236–242.
- [25] W. Scacchi, “Understanding software process redesign using modeling, analysis and simulation..,” *Software Process: Improvement and Practice*, vol. 5, no. 2–3, pp. 183–195, 2000.
- [26] W. A. Ortega-Ordoñez, C. J. Pardo-Calvache, and F. J. Pino-Correa, “Systematic mapping on the agility evaluation in software development organizations,” *ITECKNE*, vol. 16, no. 1, pp. 64–76, 2019.
- [27] R. L. Baskerville, “Investigating Information Systems with Action Research,” *Communications of the Association for Information Systems*, vol. 2, no. 3, pp. 1–32, 1999.
- [28] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009.
- [29] R. Pressman, *Ingeniería de software. Un enfoque práctico.*, 7th ed. McGraw-Hill, 2010.
- [30] Software Engineering Institute (SEI), “Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document,” 2001.
- [31] ISO, “ISO/IEC 90003:2004 - Software engineering - Guidelines for the application of ISO 9001:2000 to computer software,” 2004.
- [32] ISO, “ISO/IEC 15504-5:2012 Information technology – Process assessment – Part 5: An exemplar software life cycle process assessment model,” 2012.
- [33] D. Pajk, M. Indihar Štemberger, and A. Kovačič, “Reference Model Design : An Approach and its Application,” *Proceedings of the ITI 2012 34th Int. Conf. on Information Technology Interfaces*, pp. 455–460, 2012.
- [34] H. A. Reijers, “Process Design and Redesign,” in *Process-Aware Information Systems: Bridging People and Software through Process Technology*, M. Dumas, W. M. P. van der Aalst, and A. H. M. ter Hofstede, Eds. John Wiley &



- Sons, Inc., 2005, pp. 207–234.
- [35] “OASIS SOA Reference Model (SOA-RM) TC.” [Online]. Available: <https://www.oasis-open.org/committees/soa-rm/faq.php>. [Accessed: 17-Aug-2017].
- [36] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” *International Journal of Human - Computer Studies*, vol. 43, no. 5–6, pp. 907–928, 1995.
- [37] F. Ruiz and J. Hilera, “Using Ontologies in Software Engineering and Technology,” in *Ontologies for Software Engineering and Software Technology*, C. Calero, F. Ruiz, and M. Piattini, Eds. Springer-Verlag, 2006, pp. 49–102.
- [38] D. Fensel, “Ontology-Based Knowledge Management,” *Computer*, vol. 35, no. 11, pp. 56–59, 2002.
- [39] M. Fernández, A. Gómez-Pérez, and N. Juristo, “METHONTOLOGY: from Ontological Art towards Ontological Engineering,” in *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, 1997, pp. 33–40.
- [40] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
- [41] C. Tautz and C. G. von Wangenheim, “REFSENO: A representation formalism for software engineering ontologies,” Fraunhofer IESE-Report No. 015.98/E V1.1, 1998.
- [42] H. S. Pinto, A. Gomez-Perez, and J. P. Martins, “Some Issues on Ontology Integration,” in *In Proceedings of IJCAI99’s Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999, pp. 7–12.
- [43] J. Euzenat and P. Shvaiko, *Ontology matching*, 1st ed. Springer-Verlag Berlin Heidelberg, 2007.
- [44] V. R. Basili, G. Caldiera, and H. D. Rombach, “Software modeling and measurement: The Goal, metric, and question paradigm,” NASA/GSFC AND AFOSR, 1992.
- [45] W. A. Ortega-Ordoñez, C. J. Pardo-Calvache, and F. J. Pino-Correa, “OntoAgile : an ontology for agile software development processes,” *DYNA*, vol. 86, no. 209, pp. 86–97, 2019.
- [46] S. Gokhan Halit and M. K. Mieczyslaw, “An OWL Ontology for Representing the CMMI-SW Model,” in *The 2nd International Workshop on Semantic Web Enabled Software Engineering*, 2006.
- [47] O. Mendes and A. Abran, “Software engineering ontology: A development methodology,” in *Metrics News*, 2004, vol. 9, pp. 68–76.
- [48] L. Liao, Y. Qu, and H. K. N. Leung, “A Software Process Ontology and Its Application,” in *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, 2005, pp. 1–8.
- [49] C. F. Salviano and A. M. C. M. Figueiredo, “Unified Basic Concepts for Process Capability Models,” in *Proceedings of the 20th International Conference on Software Engineering & Knowledge Engineering (SEKE’08)*, 2008, pp. 173–178.
- [50] F. Siddiqui and M. Afshar Alam, “Ontology based application model for feature driven development,” in *Proceedings of the 5th Indian International Conference*

- on Artificial Intelligence, IICAI 2011*, 2011, pp. 1125–1137.
- [51] Y. Lin, V. Hilaire, N. Gaud, and A. Koukam, “Scrum Conceptualization Using K-CRIO Ontology,” in *Data-Driven Process Discovery and Analysis*, 2012, pp. 189–211.
  - [52] I. A. Zualkernan, “An Ontology-Driven Approach for Generating Assessments for the Scrum Software Process,” in *New Trends in Software Methodologies, Tools and Techniques - Proceedings of the Seventh SoMeT 2008*, 2008.
  - [53] D. Parsons, “Agile software development methodology, an ontological analysis.” 2010.
  - [54] D. E. Strode, “A Dependency Taxonomy for Agile Software Development Projects,” *Information Systems Frontiers*, vol. 18, no. 1, pp. 23–46, Feb. 2016.
  - [55] K. Sutling, Z. Mansor, S. Widyarto, S. Letchmunan, and N. H. Arshad, “Agile project manager behavior: The taxonomy,” in *8th Malaysian Software Engineering Conference MySEC 2014*, 2014, pp. 234–239.
  - [56] ISO/IEC, “ISO/IEC TR 29110-1:2016 - Systems and software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 1: Overview,” 2016.
  - [57] F. J. Pino, C. Pardo, F. García, and M. Piattini, “Assessment methodology for software process improvement in small organizations,” *Information and Software Technology*, vol. 52, no. 10, pp. 1044–1061, 2010.
  - [58] C. Pardo, F. García, M. Piattini, F. J. Pino, and M. T. Baldassarre, “A Reference Ontology for Harmonizing Process-reference Models,” *Revista Facultad de Ingeniería*, vol. In Press, 2014.
  - [59] P. Bourque and R. E. Fairley, *SWEBOK v.3 - Guide to the Software Engineering - Body of Knowledge*. IEEE Computer Society, 2014.
  - [60] C. P. Team, “CMMI for Development, Version 1.3,” Software Engineering Institute, Pittsburgh, Pennsylvania, 2010.
  - [61] ISO, “ISO/IEC 27001:2013 Information technology - Security techniques - Information security management systems - Requirements,” 2013.
  - [62] F. García *et al.*, “Towards a consistent terminology for software measurement,” *Information & Software Technology*, vol. 48, no. 8, pp. 631–644, 2006.
  - [63] Y. Bassil, “A Simulation Model for the Waterfall Software Development Life Cycle,” *International Journal of Engineering & Technology*, vol. 2, no. 5, pp. 2049–3444, 2012.
  - [64] “ISO/IEC/IEEE 12207:2017(E) First edition 2017-11 - Systems and software engineering - Software life cycle processes,” 2017.
  - [65] M. Kuhrmann *et al.*, “Hybrid software and system development in practice: waterfall, scrum, and beyond,” in *Proceedings of the 2017 International Conference on Software and System Process - ICSSP 2017*, 2017, pp. 30–39.
  - [66] K. Petersen and C. Wohlin, “The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study,” *Empirical Software Engineering*, vol. 15, no. 6, pp. 654–693, 2010.
  - [67] R. Saraiva de Almeida, C. J. Pardo Calvache, and M. Mira da Silva, “An Ontology-based model for ITL Process Assessment using TIPA for ITIL,” in *Communications in Computer and Information Science – CCIS 918*, p. In press.
  - [68] “OWL 2 Web Ontology Language,” 2012. [Online]. Available:

- <https://www.w3.org/TR/owl2-overview/>. [Accessed: 27-Aug-2018].
- [69] “Protégé.” [Online]. Available: <https://protege.stanford.edu/>. [Accessed: 27-Aug-2018].
- [70] “HerMiT OWL Reasoner.” [Online]. Available: <http://www.hermit-reasoner.com/>. [Accessed: 26-Aug-2018].
- [71] K. Schwaber and J. Sutherland, “La Guía de Scrum,” 2017. [Online]. Available: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>. [Accessed: 01-Jun-2018].
- [72] D. J. Anderson and A. Carmichael, *Kanban Esencial Condensado*. 2017.
- [73] VersionOne Inc., “The 12th annual State of Agile,” 2018. [Online]. Available: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>. [Accessed: 01-Jun-2018].
- [74] M. Cohn, *User Stories Applied for Agile Software Development*. Boston, MA: Pearson Education, Inc, 2004.
- [75] A. Freeman, *Pro ASP.NET MVC 5*, 5th ed. Apress, 2013.
- [76] M. Mendoza, C. González, and F. J. Pino, “Focus Group Como Proceso En Ingeniería De Software: Una Experiencia Desde La Práctica.,” *DYNA (Colombia)*, vol. 80, no. 1, pp. 51–60, 2013.
- [77] M. Taronirad and R. Ramsin, “An appraisal of existing evaluation frameworks for agile methodologies,” in *Proceedings - Fifteenth IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS 2008*, 2008, pp. 418–427.
- [78] O. Ozcan-Top and O. Demirörs, “Assessment of Agile Maturity Models: A Multiple Case Study,” in *Software Process Improvement and Capability Determination*, 2013, pp. 130–141.
- [79] M. Z. Nafchi, H. Zulzalil, and T. J. Gandomani, “On the current agile assessment methods and approaches,” in *2014 8th Malaysian Software Engineering Conference, MySEC 2014*, 2014, pp. 251–254.
- [80] M. Muñoz, J. Mejia, B. Corona, J. A. Calvo-Manzano, T. San Feliu, and J. Miramontes, “Analysis of Tools for Assessing the Implementation and Use of Agile Methodologies in SMEs,” in *Software Process Improvement and Capability Determination*, 2016, pp. 123–134.
- [81] O. E. Adali, Ö. Özcan-Top, and O. Demirörs, “Evaluation of Agility Assessment Tools: A Multiple Case Study,” in *Software Process Improvement and Capability Determination*, 2016, pp. 135–149.
- [82] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
- [83] F. McCaffery, P. S. Taylor, and G. Coleman, “Adept: A unified assessment method for small software companies,” *IEEE Software*, vol. 24, no. 1, pp. 24–31, 2007.
- [84] B. Shen and D. Ju, “On the Measurement of Agility in Software Process,” in *Software Process Dynamics and Agility*, 2007, pp. 25–36.
- [85] A. Qumer and B. Henderson-Sellers, “An evaluation of the degree of agility in six agile methods and its applicability for method engineering,” *Information and Software Technology*, vol. 50, no. 4, pp. 280–295, 2007.

- [86] M. Tatomirad and R. Ramsin, "CEFAM: Comprehensive evaluation framework for agile methodologies," in *32nd Annual IEEE Software Engineering Workshop, SEW-32 2008*, 2009, pp. 195–204.
- [87] F. Mafakheri, F. Nasiri, and M. Mousavi, "Project agility assessment: an integrated decision analysis approach," *Production Planning & Control*, vol. 19, no. 6, pp. 567–576, 2008.
- [88] M. Ikoma, M. Ooshima, T. Tanida, M. Oba, and S. Sakai, "Using a validation model to measure the agility of software development in a large software development organization," in *2009 31st International Conference on Software Engineering - Companion Volume*, 2009, pp. 91–100.
- [89] D. M. Shawky and A. F. Ali, "A practical measure for the agility of software development processes," in *2010 2nd International Conference on Computer Technology and Development*, 2010, pp. 230–234.
- [90] TongShi, JianbinChen, and J. Zhu, "Study on assessment framework of software process in agile," in *2010 2nd International Conference on Industrial and Information Systems*, 2010, vol. 1, pp. 498–501.
- [91] S. Soundararajan, J. D. Arthur, and O. Balci, "A Methodology for Assessing Agile Software Development Methods," in *2012 Agile Conference*, 2012, pp. 51–54.
- [92] V. Escobar-Sarmiento and M. Linares-Vásquez, "A model for measuring agility in small and medium software development enterprises," in *38th Latin America Conference on Informatics, CLEI 2012 - Conference Proceedings*, 2012, pp. 1–10.
- [93] K. M. Poonacha and S. Bhattacharya, "Towards a framework for assessing agility," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2012, pp. 5329–5338.
- [94] R. Wendler, "Development of the organizational agility maturity model," in *2014 Federated Conference on Computer Science and Information Systems*, 2014, pp. 1197–1206.
- [95] O. E. Adali, O. O. Top, and O. Demirors, "Assessment of agility in software organizations with a web-based agility assessment tool," in *Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017*, 2017, pp. 88–95.
- [96] M. Florêncio, F. Sambinelli, and M. A. Francisco Borges, "ASA: Agile Software Development Self-assessment Method," in *Agile Methods*, 2018, pp. 21–30.
- [97] H. Kniberg, "Squad Health Check model - visualizing what to improve," 2014. [Online]. Available: <https://labs.spotify.com/2014/09/16/squad-health-check-model/>. [Accessed: 07-May-2019].
- [98] "Scrum Level," 2018. [Online]. Available: <https://scrumlevel.com>. [Accessed: 07-May-2019].

## **Anexo A**

### **Mapeo sistemático sobre la evaluación de la agilidad en organizaciones de desarrollo de software**

Con el objetivo de obtener un estado del arte actualizado que sintetice los trabajos relacionados con la evaluación de la agilidad en las organizaciones, se realizó un mapeo sistemático de la literatura. Como resultado del mapeo, se encontraron 18 estudios, los cuales fueron analizados y comparados teniendo en cuenta el tipo de estudio, la relación de la evaluación con los principios y valores ágiles, los criterios de evaluación definidos, los métodos de validación utilizados, y las herramientas propuestas. Además, se determinaron los factores que influyen en la evaluación de la agilidad en las organizaciones. A continuación, se describe el mapeo sistemático teniendo en cuenta: i) los trabajos relacionados, ii) la metodología empleada, iii) los resultados obtenidos, y iv) las brechas encontradas.

#### **A.1 Trabajos relacionados**

Luego de revisar la literatura relevante acerca de la evaluación de la agilidad, no se evidenció la existencia de revisiones sistemáticas de la literatura o mapeos sistemáticos actualizados acerca de enfoques propuestos para evaluar la agilidad en organizaciones software. Sin embargo, se encontraron algunos estudios publicados entre el 2008 y el 2016 que presentan estudios de caso y revisiones de la literatura acerca de frameworks de evaluación de enfoques ágiles, modelos de madurez de agilidad y herramientas de apoyo a la implementación de enfoques ágiles y de evaluación de la agilidad.

En [77], se propone un conjunto de criterios de evaluación, o metacriterios, que son usados para evaluar seis frameworks de evaluación de enfoques ágiles. Los resultados de la evaluación muestran que los frameworks no satisfacen todos los metacriterios. El problema más importante encontrado en los frameworks, es que la mayoría de ellos no han considerado evaluar la agilidad del enfoque. Teniendo en cuenta que la agilidad es la principal característica de los enfoques ágiles, se espera que un framework de evaluación incluya criterios para evaluar su agilidad.

En [78], se presenta un estudio de caso múltiple que tiene por objetivo evaluar las fortalezas y debilidades de los modelos/frameworks de madurez ágil. Este tipo de soluciones se han desarrollado para guiar a las organizaciones en la mejora de procesos ágiles y en la adopción ágil. En el estudio, se seleccionan cinco modelos/frameworks de madurez de la agilidad a partir de una revisión de la literatura, luego las propuestas encontradas son evaluadas mediante su aplicación en una organización software. Como resultado del estudio de caso, se encuentran deficiencias en todas las propuestas, de acuerdo con seis criterios de evaluación definidos (idoneidad, integridad, definición de niveles ágiles, objetividad, exactitud y consistencia).

El principal objetivo de [79] es revisar los métodos de evaluación ágil existentes y proporcionar una breve discusión sobre los inconvenientes de estos métodos. Este trabajo analiza cinco enfoques que permiten medir el grado de agilidad de las organizaciones que se están moviendo hacia un enfoque ágil. Los autores concluyen que aunque el uso de enfoques ágiles está aumentando en las organizaciones software, aún existe una brecha para evaluar el grado de agilidad de este tipo de organizaciones. En general, no existe un modelo perfecto para evaluar el grado de agilidad de las organizaciones o equipos de software que adoptan métodos o prácticas ágiles.

En [80], se hace un análisis de las herramientas software que pretenden ayudar a las PYMES en la implementación de enfoques ágiles. Como parte del estudio, se hace una revisión sistemática de la literatura relacionada con frameworks, métodos y metodologías para evaluar la implementación y uso de enfoques ágiles en las PYMES. Como resultado de la revisión se encuentran veintisiete herramientas software que son evaluadas con base en seis criterios: i) enfoque que evalúa, ii)

enfoque de evaluación (implementación, adopción, ambos), iii) objetivo (generar una discusión o proveer una guía), iv) soporte al usuario, v) aplicación de los principios ágiles para la evaluación, y vi) puntuación de agilidad que provee. Los resultados de la evaluación de las herramientas mostraron que ninguna ofrece un soporte completo para las PYMES incluyendo la evaluación, el análisis de los resultados y las acciones sugeridas de acuerdo con los resultados que cubren tanto la implementación como el uso del enfoque ágil.

En [81], se presenta un estudio de caso múltiple donde se evalúan las herramientas software existentes para la evaluación de la agilidad a partir de nueve criterios de calidad: i) cobertura de principios ágiles, ii) disponibilidad, iii) capacidad de orientación al evaluador, iv) capacidad de almacenamiento de la evaluación, v) reportes automatizados, vi) funcionalidades de comparación de resultados, vii) diferentes modos de uso (individual, grupal, en paralelo), viii) diferentes alcances de evaluación (proyectos, equipos y organizaciones), y ix) extensibilidad del modelo de evaluación usado por la herramienta. El análisis de las herramientas permitió concluir que ninguna cumple con todos los criterios de calidad mencionados anteriormente, además se basan específicamente en determinar si los equipos adoptan un conjunto específico de prácticas ágiles.

## A.2 Metodología

Teniendo en cuenta que un estudio de tipo mapeo sistemático permite encontrar tendencias de investigación en un tema de interés, se realizó un estudio de este tipo siguiendo los lineamientos propuestos en [82]. La pregunta de investigación principal definida para la realización de este trabajo fue: ¿Qué estudios e iniciativas se han llevado a cabo con relación a enfoques para la evaluación de la agilidad en organizaciones software?

La pregunta principal se dividió en seis preguntas más concretas que permitieron analizar las propuestas con más detalle. A continuación, en la Tabla A.1. se pueden observar las preguntas de investigación definidas para el mapeo sistemático.

Número	Pregunta de investigación
RQ1	¿Qué estudios primarios existen sobre evaluación de la agilidad en organizaciones software?

RQ2	¿Qué tipos de soluciones han sido propuestas para evaluar la agilidad en organizaciones software y cuál es su alcance?
RQ3	¿Cuáles son los criterios usados para evaluar la agilidad en organizaciones software?
RQ4	¿Qué herramientas software se han propuesto para apoyar el proceso de evaluación de la agilidad en las organizaciones software?
RQ5	¿Qué estrategias de validación se han usado en cada estudio?
RQ6	¿Cuáles son los factores de éxito o fracaso en la evaluación de la agilidad en las organizaciones software?

Tabla A.1. Preguntas de investigación del mapeo sistemático.

En la Tabla A.2. se muestra la cadena de búsqueda básica formada a partir de un conjunto de palabras clave utilizadas para encontrar una respuesta a las preguntas de investigación.

<b>Cadena de búsqueda básica</b>
(method OR model OR methodology OR framework) AND (evaluation OR assessment OR appraisal OR measurement OR measuring) AND (Agile OR agility) AND Software AND (development OR process)

Tabla A. 2. Cadena de búsqueda básica para el mapeo sistemático.

Las fuentes con las que se llevó a cabo el mapeo sistemático fueron: i) Bases de datos científicas: ScienceDirect, Springer Link, IEEE Xplore Digital Library, Scopus, y ii) Literatura gris: documentos entregados por expertos. Además, La ventana de tiempo establecida para este mapeo fue desde el 2007 a mayo de 2018.

A continuación, se ejecutó la cadena búsqueda en cada una de las bases de datos científicas con el fin de encontrar los estudios relevantes (candidatos potenciales a convertirse en estudios primarios). Para determinar si un estudio era relevante, se realizó un análisis del título, resumen y palabras clave de cada estudio obtenido como resultado de la búsqueda. Dicho análisis se centró en determinar si cumplía con los siguientes criterios de inclusión: i) estudios que proponían una estrategia para llevar a cabo la evaluación de la agilidad en organizaciones software, ii) estudios escritos en inglés y, iii) estudios publicados en revistas internacionales (journals) o en eventos. Como criterios de exclusión se consideraron: i) estudios que no propusieran una estrategia para evaluar la agilidad en las organizaciones, y ii)



estudios duplicados (considerando el estudio más completo y reciente). Con los estudios relevantes seleccionados, nuevamente se aplicaron los criterios de inclusión y exclusión, pero esta vez analizando cada estudio en su totalidad. Los estudios seleccionados luego de este análisis se convirtieron en estudios primarios.

La selección de estudios siguió un procedimiento iterativo incremental. Este procedimiento se implementó buscando, extrayendo y visualizando los resultados de cada fuente de búsqueda de forma iterativa. De esta manera, el informe de revisión creció y evolucionó cada vez más hasta que se completó, obteniendo así el informe final del mapeo sistemático.

### A.3 Resultados

En la Tabla A.3. se muestra el consolidado de los estudios analizados y seleccionados en cada una de las iteraciones de la revisión. En total, se encontraron 1491 estudios, de los cuales 1489 se obtuvieron como resultado de ejecutar la cadena de búsqueda en las bases de datos científicas y 2 estudios correspondieron a literatura gris. Siguiendo la metodología planteada anteriormente, se seleccionaron 35 estudios relevantes y 18 estudios primarios.

<b>No. Iteración – Fuente</b>	<b>Estudios encontrados</b>	<b>Estudios relevantes</b>	<b>Estudios primarios</b>
1- Science Direct	54	6	1
2 - Springer Link	511	8	3
3 - IEEE Xplore	466	15	10
4 – Scopus	458	4	2
5 - Literatura gris	2	2	2
Total	1491	35	18

Tabla A.3. Estudios analizados y seleccionados en el mapeo sistemático.

A continuación se presentan los resultados del análisis de cada uno de los estudios primarios a través de cada una de las preguntas de investigación.

#### A.3.1 Pregunta RQ1. Estudios sobre evaluación de la agilidad

En la Tabla A.4. se muestran los estudios primarios incluyendo: Identificación del estudio, nombre del estudio, año de publicación y referencia. Los estudios E1 al E16

se obtuvieron de las bases de datos científicas. Los estudios E17 y E18 corresponden a literatura gris.

<b>Id.</b>	<b>Nombre del estudio</b>	<b>Año</b>	<b>Ref.</b>
E1	Adept: a unified assessment method for small software companies	2007	[83]
E2	A disciplined approach to adopting agile practices: the agile adoption framework	2007	[19]
E3	On the measurement of agility in software process	2007	[84]
E4	An evaluation of the degree of agility in six agile methods and its applicability for method engineering,	2008	[85]
E5	CEFAM: comprehensive evaluation framework for agile methodologies	2008	[86]
E6	Project agility assessment: an integrated decision analysis approach	2008	[87]
E7	Using a validation model to measure the agility of software development in a large software development organization	2009	[88]
E8	A practical measure for the agility of software development processes	2010	[89]
E9	Study on assessment framework of software process in agile	2010	[90]
E10	A methodology for assessing agile software development methods	2012	[91]
E11	A model for measuring agility in small and medium software development enterprises	2012	[92]
E12	Towards a framework for assessing agility	2012	[93]
E13	Development of the organizational agility maturity model	2014	[94]
E14	A reference model for software agility assessment: Agilitymod	2015	[1]
E15	Assessment of agility in software organizations with a web-based agility assessment tool	2017	[95]
E16	ASA: agile software development self-assessment method	2017	[96]
E17	Squad Health Check model	2014	[97]
E18	Scrum Level	2018	[98]

Acrónimos utilizados: Id: Identificador, Ref: Referencia.

Tabla A.4. Estudios primarios seleccionados en el mapeo sistemático.

### **A.3.2 Pregunta RQ2. Tipos de soluciones propuestas y alcance de la evaluación**

En la Tabla A.5., se clasificaron los estudios de acuerdo al tipo de solución que proponen para realizar la evaluación de la agilidad. Asimismo, en la columna

relacionada con el alcance del enfoque se clasificaron los estudios teniendo en cuenta si proponían soluciones para evaluar la agilidad de: i) los enfoques de desarrollo, ii) la organización, iii) los proyectos software, iv) los procesos software, y v) el equipo de desarrollo.

A partir del análisis de la información de los estudios primarios se puede destacar: El tipo de solución que más se propone corresponde a framework de evaluación (E4, E5, E9, E10, E12), seguido de método de evaluación (E1, E16) y métrica (E7, E8). Otros tipos de solución corresponden a: i) índice de medición, ii) modelo de soporte a la decisión, iii) modelo de evaluación, iv) modelo de madurez, v) modelo de referencia, vi) herramienta web, y vii) técnica de visualización.

<b>Id.</b>	<b>Tipo de solución</b>	<b>Alcance de la evaluación</b>
E1	Método de evaluación	Proceso software
E2	Índice de medición	Proyecto y organización
E3	Método de medición	Proceso software
E4	Framework de evaluación	Enfoque de desarrollo
E5	Framework de evaluación	Enfoque de desarrollo
E6	Modelo de soporte a la decisión	Proyecto
E7	Métrica	Proyecto y organización
E8	Métrica	Proceso de desarrollo
E9	Framework de evaluación	Proceso software
E10	Framework de evaluación	Enfoque de desarrollo
E11	Modelo de evaluación	Organización
E12	Framework de evaluación	Organización
E13	Modelo de madurez	Organización
E14	Modelo de referencia	Proyecto
E15	Herramienta Web	Proyecto
E16	Método de evaluación	Enfoque de desarrollo
E17	Técnica de visualización	Equipo de desarrollo
E18	Modelo de evaluación	Organización

Acrónimos utilizados: Id: Identificador.

Tabla A.5. Tipos de soluciones y alcance de la evaluación.

Con relación al alcance de la evaluación, el 22,2% de los estudios (E1, E3, E8, E9) proponen evaluar el proceso software usado por la organización, otro 22,2%

de los estudios (E4, E5, E10, E16) proponen evaluar un enfoque de desarrollo en particular, por ejemplo Scrum, XP o una adaptación, el 22,2% de los estudios (E11, E12, E13, E18) proponen evaluar la agilidad de una organización, el 16,7% (E6, E14, E15) se enfocan en evaluar la agilidad de un proyecto de desarrollo de software, el 11,1% (E2, E7) han sido propuestos para evaluar la agilidad tanto de organizaciones como de proyectos, y el estudio E17 que corresponde al 5,5% permite evaluar la agilidad de un equipo de desarrollo (ver Figura A.1).

Los distintos tipos de alcance que proponen los estudios, evidencian que para tener una visión global de la agilidad de una organización, es importante contar con enfoques que permitan conocer en detalle el grado de agilidad de sus procesos, proyectos y equipos.

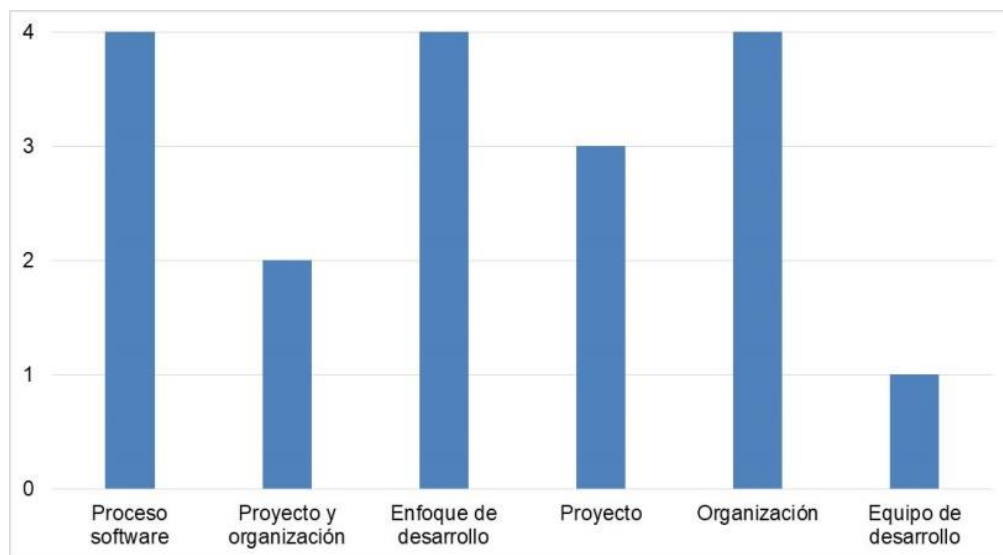


Figura A.1 Alcance de evaluación de los estudios primarios.

### A.3.3 Pregunta RQ3. Criterios usados para evaluar la agilidad

En la Tabla A.6, se pueden observar los estudios clasificados de acuerdo a los criterios que usan para realizar la evaluación de la agilidad. Se han definido cuatro categorías: C1: soluciones que verifican el cumplimiento de los principios del manifiesto ágil. C2: soluciones que verifican el cumplimiento de los valores del manifiesto ágil. C3: soluciones que verifican el cumplimiento de características de agilidad definidas en el mismo estudio. C4: soluciones que verifican el cumplimiento de características de agilidad definidas en otros estudios. Las categorías descritas no

son exclusivas, por ejemplo, el estudio E10 usa los principios y valores ágiles, y el estudio E4 define sus propias características y usa también los valores ágiles.

<b>Categoría</b>	<b>Estudios</b>
C1	E2, E10, E14
C2	E4, E7, E10
C3	E1, E3, E4, E5, E6, E7, E8, E9, E12, E13, E17, E18
C4	E11, E15, E16

Tabla A.6. Clasificación de estudios según los criterios usados para la evaluación.

A continuación, en la Tabla A.7. se describen los elementos que se han tenido en cuenta en cada estudio para evaluar la agilidad.

<b>Id.</b>	<b>Elementos que se evalúan</b>
E1	Combina un método de valoración de procesos compatible con SCAMPI clase C que se ajusta a las características de las organizaciones con poca experiencia y un método que permite determinar en qué grado se deben combinar métodos tradicionales y ágiles en la empresa para disminuir los riesgos y aprovechar las ventajas de los dos enfoques.
E2	Se analizan los doce principios del manifiesto y se resumen en cinco principios: i) aceptar el cambio para entregar el valor del cliente, ii) planificar y entregar software con frecuencia, iii) centrarse en las personas, iv) excelencia técnica y v) colaboración con los clientes. A partir de estos principios, se definen las prácticas ágiles que les dan soporte y que deben ser implementadas para conseguir un nivel de agilidad determinado.
E3	Se tienen en cuenta cuatro perspectivas del desempeño de una organización: i) cliente, ii) financiero, iii) proceso interno, y iv) aprendizaje y crecimiento. Para cada uno de los cuadrantes, la organización define sub-objetivos y se definen métricas asociadas a cada uno de ellos. Se define un conjunto de métrica ágiles comunes que las organizaciones pueden usar: i) ROI: Retorno de la inversión, ii) Productividad: Relación entre la producción de un producto y los recursos utilizados en el desarrollo de software, iii) Calidad: Calidad del producto o servicio, iv) Adaptabilidad: Habilidad de adaptación al cambio, v) Innovación: Habilidad para innovar, mejorar y aprender.
E4	Se considera que un método ágil debe contar con cinco características: flexibilidad, velocidad, delgadez (leanness), capacidad de aprendizaje y capacidad de respuesta. El cumplimiento de los valores ágiles se evalúa mediante el análisis de

	<p>las prácticas que se han definido en el método y que están relacionadas con cada valor. Se proponen dos valores adicionales a los expresados en el manifiesto ágil: i) mantener el proceso ágil, y ii) mantener el proceso rentable.</p>
E5	<p>Los elementos para la evaluación de la agilidad se definieron basándose en el manifiesto ágil y propuestas que presentan características comunes de los enfoques ágiles. Se evalúa: velocidad, sostenibilidad, flexibilidad, aprendizaje, sensibilidad, delgadez (leanness), ligereza y simplicidad, calidad técnica, y colaboración activa del cliente.</p>
E6	<p>Aunque no se hace referencia directa a los valores y principios del manifiesto ágil, se definen seis dimensiones de agilidad que están estrechamente relacionadas con los principios ágiles, esas dimensiones son: i) dinamismo, ii) tamaño del equipo, iii) comunicación, iv) pruebas, v) habilidades y conocimientos de los desarrolladores, y vi) cultura.</p>
E7	<p>Este documento define la agilidad en el desarrollo de software como la minimización del tiempo desde la generación de entregables intermedios hasta la validación, y propone una métrica similar a la utilizada para medir el volumen de negocios en el área de finanzas pero adaptada a las características del desarrollo de software.</p>
E8	<p>Se propone un enfoque para medir la agilidad de un proceso de desarrollo de software mediante el cálculo de la entropía. Se entiende la agilidad como la tasa de cambio de entropía en el tiempo del desarrollo del sistema. Cuanto mayor sea esta tasa, mayor será la agilidad del proceso de desarrollo de software. Para el cálculo de la entropía se analizan los logs que contienen los registros de cambios en el sistema que se está desarrollando.</p>
E9	<p>Se define la agilidad como la capacidad de operar de forma rentable en un entorno competitivo de cambios continuos e impredecibles. A partir de esta definición se proponen los siguiente factores de evaluación: i) cooperación, ii) calidad y capacidad de respuesta del proceso, iii) proceso de cambio, iv), aplicabilidad y rendimiento del proceso, y v) capacitación y bienestar.</p>
E10	<p>A partir de los valores ágiles se identifican cinco objetivos que reflejan la filosofía ágil, nueve principios que resumen los principios ágiles y respaldan el logro de esos objetivos, y 27 prácticas que ayudan a implementar los principios.</p>
E11	<p>Define que para evaluar la agilidad en una organización se deben evaluar los siguientes aspectos: Proyecto, equipo, y espacio de trabajo. Para cada uno de los aspectos mencionados, se establecen cuestionarios adaptados de modelos de evaluación de agilidad existentes.</p>
E12	<p>Se definen 17 parámetros para medir la agilidad de una organización. Dichos</p>

	<p>parámetros se dividen en 4 grupos: i) desgaste, flexibilidad funcional, entrenamiento y conocimiento, toma de decisión descentralizada, fuerza, ii) programación en parejas y pruebas en paralelo, desarrollo iterativo, grado de modularidad, proceso de captura de requisitos, reutilización, mejora continua. iii) participación del cliente en el diseño, equipos fuera de la organización, período de entrenamiento del cliente. iv) costo de cambio de requisitos, proyectos fallidos por incapacidad, calidad del software.</p>
E13	<p>Se propone un modelo de madurez que consta de tres dimensiones: i) Dimensión 1: Requisitos previos de agilidad. Grado en que las personas de una organización comparten valores ágiles y en qué medida la organización establece los prerrequisitos tecnológicos para apoyar la agilidad, ii) Dimensión 2: Agilidad de las personas. Resume todas las capacidades necesarias de los miembros de una organización para traducir los valores ágiles en acciones, y iii) Dimensión 3: Estructuras que apoyan la agilidad. Describe la capacidad de una organización para adoptarse y cambiar de manera flexible en combinación con una cultura organizacional que respalda la colaboración y la cooperación en todos los niveles.</p>
E14	<p>Se evalúa el cumplimiento de los doce principios ágiles mediante lo que denominan atributos de aspecto. Propone un modelo de referencia basado en la norma ISO/IEC 15504 (SPICE) y define cuatro aspectos que abarcan el ciclo de vida de desarrollo de software: exploración, construcción, transición y gestión. Para evaluar la agilidad de un proyecto, se analizan cada uno de los aspectos para determinar en qué nivel cumplen con lo que denominan atributos de aspecto. Dichos atributos se derivan de cada uno de los doce principios del manifiesto ágil y consisten en la implementación de prácticas genéricas que lleven al aspecto a ser iterativo, simple, técnicamente excelente, y que apoyen el aprendizaje.</p>
E15	<p>Realiza la evaluación de la agilidad de un proyecto a través de una herramienta web usando el modelo de referencia propuesto en E14, el cuál fue desarrollado por los mismos autores.</p>
E16	<p>Usa el framework propuesto en E10 para establecer los criterios de evaluación combinando algunas prácticas para disminuir el número de criterios a evaluar. La organización que se evalúa puede elegir las prácticas que considera aplicables según su contexto. Integra la metodología PDCA (Plan, Do, Check, Act) para que una vez se haga la evaluación se generen planes de mejora.</p>
E17	<p>Se define la salud del equipo como la opinión que cada uno de los miembros tiene acerca de distintos factores relacionados con su entorno de trabajo. Entre los factores se incluyen: soporte, equipo de trabajo, proceso, entrega de valor, aprendizaje, velocidad, facilidad de entrega de versiones y diversión.</p>
E18	<p>Plantea que la agilidad no se alcanza mediante el uso de prácticas ágiles sino</p>

	mediante la aplicación de principios y valores. Define un conjunto de principios y valores que deben estar presentes en una organización para ser considerada ágil. Los principios que define son: i) foco en el cliente, ii) mejora continua, iii) asumir la variabilidad, iv) desarrollo incremental, v) ritmo sostenible, vi) calidad técnica, vii) transparencia, y viii) cadencia y sincronización. Define dos valores: i) organización centrada en las personas, y ii) gobernanza descentralizada. Además, considera que el desarrollo de la agilidad necesita la implicación y el apoyo de la dirección.
--	---

Acrónimos utilizados: Id: Identificador.

Tabla A.7. Elementos que se evalúan en cada estudio primario.

Pregunta RQ4. Herramientas software de apoyo a la evaluación de la agilidad Solamente los estudios E15 y E16 proponen herramientas software para apoyar el proceso de evaluación de la agilidad. El estudio E15 propone una herramienta web para realizar el proceso de evaluación de la agilidad de un proyecto. Dicha herramienta está basada en el modelo de referencia propuesto en E14. En el estudio E16, se propone un prototipo de una hoja de cálculo que apoya una parte del método de evaluación propuesto.

#### A.3.4 Pregunta RQ5. Estrategias de validación de los estudios

En la Tabla A.8. se pueden observar las estrategias de validación usadas en cada estudio.

Id.	Estrategia de validación
E1	No se menciona
E2	El framework fue presentado a veintiocho miembros de la comunidad ágil para obtener retroalimentación a través de entrevistas y encuestas.
E3	Se realizó un estudio de caso.
E4	Se usó el framework para evaluar seis enfoques de desarrollo de software: XP, Scrum, FDD, ASD, DSDM y Crystal.
E5	Se usó el framework para evaluar el enfoque XP.
E6	Se realizó un estudio de caso.
E7	Se aplicó la métrica propuesta en una organización de gran tamaño y se compararon resultados de dos grupos de trabajo que trabajaron en más de mil proyectos cada uno. La medición se realizó durante cuatro años.
E8	Se realizaron dos estudios de caso.
E9	Se realizaron encuestas en organizaciones software



E10	Los componentes principales del framework (objetivos, principios y prácticas) se validaron mediante una encuesta online dirigida a profesionales del desarrollo de software. Además, se evaluó la adecuación de tres enfoques de desarrollo de software: XP, FDD, y una adaptación de XP.
E11	No se menciona.
E12	Se validó el modelo usando datos de prueba en MATLAB.
E13	Se evaluó la aplicabilidad del modelo mediante un análisis de cluster sobre los datos de una encuesta realizada en organizaciones software y del sector de servicios de tecnologías de la información.
E14	Se realizó un estudio de caso múltiple.
E15	Se realizó un estudio de caso múltiple.
E16	No se menciona.
E17	No se menciona.
E18	No se menciona.

Acrónimos utilizados: Id: Identificador.

Tabla A.8. Estrategias de validación de los estudios primarios.

### A.3.5 Pregunta RQ6. Factores de éxito o fracaso en la evaluación de la agilidad

A partir de los estudios analizados se ha identificado un conjunto de factores de éxito o fracaso que podrían ser tenidos en cuenta en trabajos de evaluación de la agilidad en organizaciones software. En la Tabla A.9., se muestra el nombre del factor, su descripción y el estudio usado como fuente.

<b>Factor 1</b> Minimizar el costo de la evaluación.	Descripción	El tiempo usado para preparar y realizar la evaluación debe ser mínimo, y el método de evaluación debe permitir que las organizaciones seleccionen las áreas de proceso que sean más relevantes para sus objetivos comerciales.
	Fuente	E1
<b>Factor 2</b> Validez de la solución para la evaluación de la agilidad.	Descripción	Un método ideal para la validación de un modelo de evaluación de la agilidad sería aquel que presente un estudio longitudinal que compare los resultados obtenidos al evaluar la agilidad de los procesos de desarrollo de varias organizaciones, versus los resultados obtenidos con relación a otras organizaciones que no lo usen, esto permitiría obtener suficiente evidencia

		empírica que justifique la validez del modelo. Sin embargo, el desafío con este tipo de estudios es la gran cantidad de tiempo y presupuesto requerido para llevarlo a cabo.
	Fuente	E2
<b>Factor 3</b> Aplicar prácticas ágiles no es sinónimo de ser ágil.	Descripción	Suponer que implementar prácticas ágiles en una organización implica que esta sea más ágil es erróneo, esto debido a que los equipos pueden adoptar las prácticas propuestas en una metodología ágil sin alinearlas con los principios y valores ágiles. El factor de éxito más importante en la transición ágil es adoptar y practicar los principios y valores proporcionados por el manifiesto ágil. Estos principios y valores están redactados de manera general, lo que permite que se interpreten de distintas maneras. Por esta razón, si se desea conocer el grado de agilidad alcanzado por una organización, es necesario evaluar en qué medida ha implementado y adaptado prácticas ágiles según su contexto para cumplir con los principios y valores ágiles.
	Fuente	E3, E11
<b>Factor 4</b> Modelo de referencia para la evaluación de la agilidad.	Descripción	Es necesario que cuando se implementen enfoques de evaluación de agilidad, se identifiquen todos los parámetros que influyen en la agilidad involucrando a expertos de la academia y de la industria. Lo anterior con el fin tener un conjunto de parámetros de agilidad universalmente aceptable. A pesar de la existencia de un gran número de modelo de evaluación de agilidad, no existe uno comúnmente aceptado por la industria.
	Fuente	E12, E14
<b>Factor 5</b> Ambigüedad y multidimensionalidad en	Descripción	Las métricas ágiles son difíciles de definir principalmente debido a la ambigüedad en la terminología que usan los distintos enfoques

los conceptos de enfoques ágiles.		ágiles existentes y al número de dimensiones que influyen en la agilidad. En la Tabla A.10., se muestran algunos ejemplos de términos que tienen interpretaciones distintas en los enfoques ágiles Scrum y XP.
	Fuente	E3

Tabla A.9. Factores a tener en cuenta en la evaluación de la agilidad

<b>Término</b>	<b>Interpretación del término en Scrum</b>	<b>Interpretación del término en XP</b>
Valores	Compromiso, coraje, foco, apertura y respeto	Comunicación, simplicidad, retroalimentación, coraje y respeto
Prácticas	Sprint, planificación del sprint, reunión diaria, revisión del sprint, retrospectiva del sprint	Historias, ciclos semanales, revisiones y planificaciones, holgura, sentarse juntos, equipo completo, puesto de trabajo con información, ritmo sostenible, programación en parejas, diseño incremental o evolutivo, pruebas antes de programas, integración continua, construcción en diez minutos.
Desarrollador	Responsable de entregar un incremento de producto que potencialmente se pueda poner en producción al final de cada sprint.	Todos los miembros del equipo (con excepción del cliente) se consideran desarrolladores. Son responsables de implementar las historias identificadas por el cliente.

Tabla A.10. Ejemplos de ambigüedad en la terminología de enfoques ágiles.

## A.4 Brechas existentes

A partir del análisis de los resultados del mapeo sistemático, se puede observar que ya existen propuestas relacionadas con la evaluación de la agilidad, sin embargo, no existe consenso en la definición de agilidad en las organizaciones software y por lo tanto no hay un acuerdo en cuanto a los criterios que deben ser tenidos en cuenta como referencia para evaluar la agilidad. Solamente el estudio E10 usa como

referencia para la evaluación el cumplimiento de los principios y valores ágiles los cuales son aceptados ampliamente a nivel internacional como indicadores de agilidad en el desarrollo de software, los estudios E2 y E14 usan como referencia solamente los principios ágiles, y los estudios E4 y E7 usan los valores ágiles y definen sus propios criterios de agilidad. La mayoría de los estudios usan como referencia para la evaluación de la agilidad criterios que los mismos autores de los estudios definen lo que puede representar una visión subjetiva de la agilidad. Además, ninguno de los estudios presenta de forma clara y detallada la terminología relacionada con los enfoques ágiles y la evaluación de la agilidad, razón por la cual se dificulta la aplicación de los enfoques de evaluación por parte de las propias organizaciones debido a problemas de ambigüedad en los términos empleados en distintos enfoques ágiles y tradicionales.

También se puede observar que ninguna de las propuestas encontradas presenta de forma clara y detallada la relación entre los principios y valores ágiles y los elementos (actividades, roles y productos) de los procesos software. Debido a lo anterior, no es posible determinar el grado de cumplimiento del manifiesto ágil a partir de los resultados de la evaluación y se dificulta encontrar oportunidades de mejora detalladas a nivel de elementos de proceso que faciliten la interpretación de los resultados de la evaluación por parte de las organizaciones.

Con respecto a las herramientas software propuestas, se puede observar que están enfocadas en la evaluación de la agilidad de la organización, del equipo o del proyecto. No hay evidencia de la existencia de una herramienta software de evaluación de la agilidad de los procesos software de las organizaciones.

## Anexo B

# Modelo de referencia para la evaluación de la agilidad en los procesos de desarrollo de software de las organizaciones – AgilityRef – Versión 1.0

Un modelo de referencia se puede definir como: “Un modelo conceptual genérico que formaliza prácticas recomendadas para cierto dominio” [33]. Las prácticas recomendadas, también llamadas mejores prácticas, expresan la mejor manera para tratar un problema particular y pueden ser replicadas y adaptadas por una organización en una situación similar [34]. En este sentido, se presenta AgilityRef, el cual es un modelo de referencia para la evaluación de la agilidad en los procesos software de las organizaciones, el cual está dividido en tres componentes principales: i) indicadores de agilidad, ii) principios ágiles, y iii) valores ágiles. Un indicador de agilidad describe un elemento de proceso (actividad, rol, producto, herramienta) que evidencia la implementación de uno o más principios ágiles. Por su parte, los principios ágiles describen las características que debe tener un proceso ágil de desarrollo de software y soportan el cumplimiento de uno o más valores ágiles en el proceso de desarrollo (ver Figura B.1).

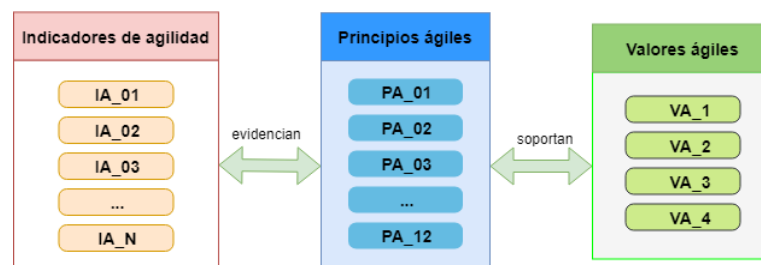


Figura B.1 Componentes de AgilityRef.

Los componentes y relaciones de AgilityRef surgieron a partir de la creación de la ontología para procesos de desarrollo ágil de software – OntoAgile [45] (ver Figura B.2) la cual proporciona conceptos, definiciones y relaciones alrededor de los

enfoques ágiles y los procesos de desarrollo de software. En OntoAgile se pudo establecer que para conocer el grado de agilidad de un proceso software basándose en el cumplimiento de los principios y los valores ágiles, es necesario definir un conjunto de indicadores que describan la manera en que los elementos de proceso (actividades, productos, roles y herramientas) permiten implementar los principios ágiles y a su vez dar soporte al cumplimiento de los valores ágiles.

En la Figura B.3, se ilustra la instanciación de OntoAgile para representar la relación entre indicadores, principios ágiles y valores ágiles. En el ejemplo se muestra el indicador llamado “Reunión diaria”, el cual representa la reunión que debe llevar a cabo un equipo para evaluar su progreso con relación al objetivo de una iteración. Si esta actividad se lleva a cabo de acuerdo a las características descritas en el indicador, se evidenciaría un aporte al cumplimiento de los principios ágiles seis y doce, los cuales están relacionados con la conversación cara a cara como el método más eficiente para comunicar información al equipo, y la reflexión a intervalos regulares con el fin de realizar ajustes para aumentar la efectividad. A su vez, esos dos principios ágiles brindan soporte al cumplimiento del primer valor ágil, en el cual se pone de manifiesto que en un proceso ágil de desarrollo de software son más importantes los individuos y sus interacciones que los procesos y herramientas. Tanto los valores y principios ágiles, como los indicadores de agilidad serán descritos en detalle en las secciones 1.1, 1.2 y 1.3, respectivamente.

## **B.1 Valores ágiles**

Los valores ágiles representan los principales atributos que debe tener un proceso software para ser considerado ágil según lo declarado en el manifiesto ágil [2]. En la Tabla B.1., se pueden observar cada uno de los valores ágiles, un identificador para facilitar su referenciación en este documento y la interpretación de dichos valores en AgilityRef, la cual resulta del análisis de los valores descritos en el manifiesto ágil.

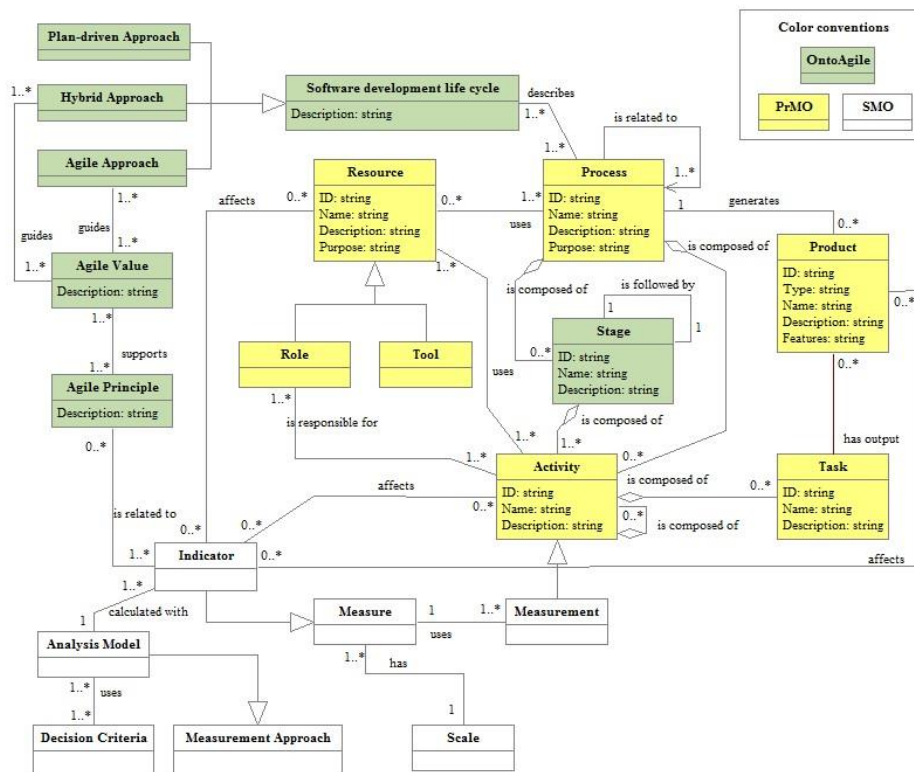


Figura B.2 Representación gráfica de OntoAgile usando UML.

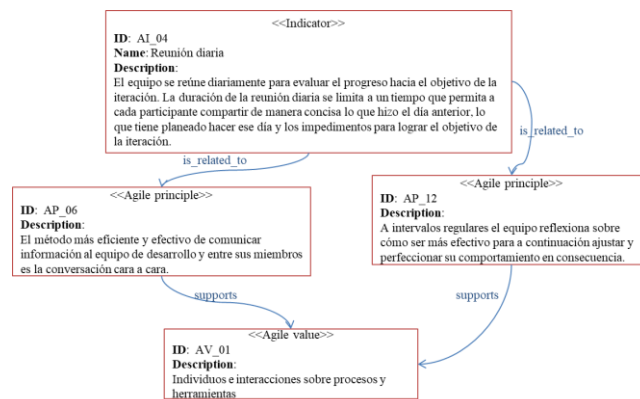


Figura B.3 Ejemplo de instanciación en OntoAgile de un indicador y su relación con principios y valores ágiles.

Id.	Valor ágil	Interpretación en AgilityRef
VA_1	Individuos e interacciones sobre procesos y	Los procesos ágiles deben estar centrados en las personas y no en los procesos o herramientas. No se puede negar la importancia de tener procesos que guíen el trabajo y herramientas que mejoren su eficiencia, pero es mucho más importante contar con un equipo de personas

	herramientas.	motivadas, con excelentes habilidades técnicas, comunicativas y de auto organización que les permitan reaccionar rápidamente al cambio, interactuar efectivamente con el cliente y sus compañeros de equipo, y generar productos que entreguen valor al cliente. Los procesos y herramientas deben adaptarse al equipo y no al contrario.
VA_2	Software funcionando sobre documentación extensiva.	Aunque la documentación generada durante un proceso de desarrollo de software es importante, el principal objetivo del proceso debe ser entregar al cliente de manera iterativa, incrementos del producto que funcionen y aporten valor al negocio. Cada una de las entregas de incrementos del producto, debe ir acompañada de la documentación mínima necesaria y además debe superar un conjunto de criterios mínimos de aceptación por parte del cliente.
VA_3	Colaboración con el cliente sobre negociación contractual.	Un proceso de desarrollo ágil requiere un cliente que colabore frecuentemente con el equipo durante el desarrollo del producto. Esta colaboración permitirá priorizar las funcionalidades según el valor que aportarán al negocio y detectar riesgos de manera. De ser posible, se debe contar con el cliente en el sitio de desarrollo del producto para que la comunicación sea más efectiva, pero en caso de que esto no sea posible, se debe contar con un representante del cliente que tenga poder de decisión.
VA_4	Respuesta ante el cambio sobre seguir un plan.	A diferencia de los enfoques tradicionales de desarrollo donde el objetivo está centrado en tener un plan detallado desde el comienzo, en un proceso de desarrollo ágil se planifica solamente lo necesario para iniciar el desarrollo del producto. A medida que se avanza en cada iteración, los cambios detectados a partir de la comunicación con el cliente, se ven como una oportunidad para entregar un producto que aportará más valor.

Acrónimos utilizados: Id: Identificador, VA: Valor Ágil.

Tabla B.1. Valores ágiles.

## B.2 Principios ágiles

Los doce principios ágiles fueron definidos en el manifiesto ágil [2] y aportan lineamientos generales para que un proceso de desarrollo de software esté alineado con los valores ágiles. A continuación, en la Tabla B.2 se presentan los doce principios ágiles.

Id.	Principio ágil
PA_01	Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
PA_02	Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.



PA_03	Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
PA_04	Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
PA_05	Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
PA_06	El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
PA_07	El software funcionando es la medida principal de progreso.
PA_08	Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
PA_09	La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
PA_10	La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
PA_11	Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
PA_12	A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Acrónimos utilizados: Id: Identificador, PA: Principio Ágil

Tabla B.2. Principios ágiles.

La relación entre principios y valores ágiles propuesta en AgilityRef se muestra en la Figura B.4, en la cual se puede observar que la mayoría de principios ágiles están relacionados con el valor VA\_01, con un total de 7 principios, confirmando que un proceso de desarrollo ágil está centrado en las personas. El segundo valor con más principios relacionados es el VA\_02, con un total de 3 principios, lo cual refleja la importancia que tiene para un proceso ágil obtener como salida un producto que aporte valor al cliente. Por último, los valores VA\_3 y VA\_4 tienen una relación con un principio cada uno, relacionados con PA\_04 y PA\_02, respectivamente.

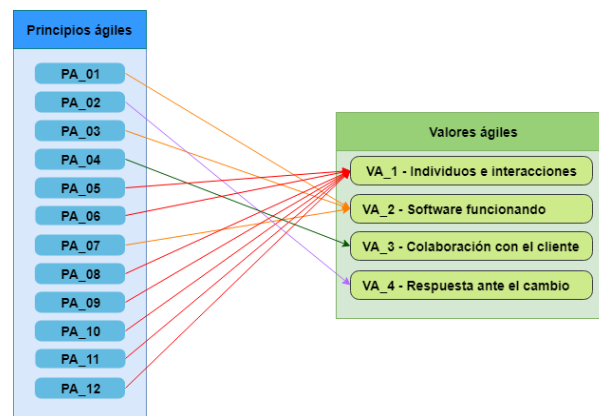


Figura B.4 Relaciones entre principios y valores ágiles en AgilityRef.

### B.3 Indicadores de agilidad

Los indicadores de agilidad corresponden a un conjunto de actividades, roles, productos y herramientas que evidencian el cumplimiento de los principios ágiles en un proceso software. Para la definición de los indicadores, se analizaron actividades, roles, productos y herramientas propuestas en enfoques ágiles como: Scrum [71], XP [23] y Kanban [72], y que se relacionan directamente con uno o más principios ágiles. La selección de estos enfoques se realizó teniendo en cuenta que son los más usados en la industria de software actual según los resultados de la encuesta anual sobre agilidad “State of Agile” [73]. En los resultados de la encuesta se destaca que Scrum es el enfoque más usado por las organizaciones consultadas, esto se refleja con un uso del 56% de las personas encuestadas, seguido de la combinación de Scrum con Kanban (ScrumBan) con 8% y la combinación de Scrum y XP con 6%. Es importante destacar que los indicadores describen el “que” y no el “cómo” debido a que las organizaciones que implementan enfoques ágiles deben definir sus procesos de acuerdo a su contexto particular, pero teniendo como pilares los principios y valores ágiles. En la Tabla B.3., se presentan los indicadores de agilidad que se proponen en AgilityRef, su respectivo identificador, nombre, descripción, fuente, tipo de elemento de proceso que representa, y principios ágiles a los cuales se relaciona.

Id.	Nombre	Descripción	Fuente	Tipo	PA's relacionados
IA_01	Iterativo e incremental	El producto se crea de forma iterativa e incremental. Los intervalos tienen una duración máxima de un mes. Además, cada nueva iteración comienza inmediatamente después de la finalización de la iteración anterior.	Scrum, XP, Kanban	Actividad	PA_01 PA_03
IA_02	Incrementos con valor	Cada iteración genera una versión potencialmente útil y funcional del producto.	Scrum	Producto	PA_01 PA_03 PA_07

IA_03	Alcance negociable	El alcance de cada iteración puede clarificarse y renegociarse entre el equipo y el cliente.	Scrum, XP	Actividad	PA_02 PA_04
IA_04	Reunión diaria	El equipo se reúne diariamente para evaluar el progreso hacia el objetivo de la iteración. La duración de la reunión diaria se limita a un tiempo que permita a cada participante compartir de manera concisa lo que hizo el día anterior, lo que tiene planeado hacer ese día y los impedimentos para lograr el objetivo de la iteración.	Scrum, Kanban	Actividad	PA_06 PA_12
IA_05	Refinamiento de requisitos	La lista de requisitos del producto se actualiza constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Además, los requisitos pueden ser actualizados incluyendo más detalle, estimaciones y prioridades.	Scrum	Actividad	PA_01 PA_02 PA_10
IA_06	Auto organización del equipo	Cada integrante del equipo puede elegir la mejor forma de llevar a cabo las tareas que tenga bajo su responsabilidad. El equipo tiene la	Scrum, XP	Rol	PA_05 PA_11

		posibilidad de dar mayor prioridad a los requisitos del producto que presenten más riesgo.			
IA_07	Responsabilidad compartida	Los miembros del equipo pueden tener habilidades especializadas en áreas en las que estén más enfocados, pero la responsabilidad del producto es de todo el equipo. Cualquier miembro del equipo podría realizar cambios en cualquiera de los componentes del producto.	Scrum, XP	Rol	PA_05
IA_08	Ritmo constante	El tiempo de duración de las iteraciones es consistente durante todo el desarrollo del producto.	Scrum	Actividad	PA_08
IA_09	Iteraciones negociables	Una iteración se puede cancelar cuando se determina que su objetivo ha quedado obsoleto. Por ejemplo, si las condiciones del mercado o de la tecnología cambian.	Scrum	Actividad	PA_02
IA_10	Retrospectivas	Al finalizar cada iteración, el equipo y los interesados se reúnen para revisar los resultados con respecto a las personas, procesos y herramientas. Se identifican y ordenan	Scrum	Actividad	PA_04 PA_06 PA_09 PA_12

		los elementos más importantes que salieron bien y las posibles mejoras, se crea un plan para implementar las mejoras, y se adapta la lista de requisitos del producto en caso de ser necesario.			
IA_11	Eliminación de impedimentos	Se tiene por práctica la asignación de un rol responsable de ayudar al equipo y a los interesados a ejecutar las actividades de la mejor manera y eliminar los impedimentos que se presenten.	Scrum, XP	Rol	PA_05
IA_12	Participación del cliente	Los clientes pueden decidir sobre qué partes del problema se deben resolver con mayor prioridad para que el equipo pueda enfocarse en aquello que entregue valor. Los clientes participan en la creación de las pruebas funcionales del producto y en la elección de la tecnología a emplear para su desarrollo. Un cliente real está disponible en la misma ubicación del equipo para responder preguntas, resolver discusiones y fijar prioridades a pequeña	Scrum, XP	Rol	PA_01 PA_04 PA_07

		escala. En caso de que no sea posible contar con el cliente real, se debe contar con un representante del cliente y de los interesados en el producto con capacidad para tomar decisiones.			
IA_13	Estimación en equipo	El equipo es el encargado de estimar el tiempo que se empleará para implementar una característica del producto	XP	Rol	PA_05 PA_11
IA_14	Retroalimentación al cliente	Cuando el equipo tome decisiones técnicas que puedan tener impacto a nivel de negocio, debe informar a los interesados.	XP	Rol	PA_04
IA_15	Visión compartida	Se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los clientes.	XP	Actividad	PA_04
IA_16	Atención al buen diseño	Durante el diseño del producto se tiene en cuenta que no se dupliquen componentes, se incluyen todos los aspectos que el equipo considera importantes y se incluyen solamente lo que se requiere en el	XP	Actividad	PA_09 PA_10

		momento. Cada vez que se agrega una nueva característica al producto, se verifica si se puede modificar su estructura interna para hacerla más sencilla, en caso de ser posible, se realizan los cambios y se verifica que todas las pruebas funcionen.			
IA_17	Trabajo colaborativo	Cuando un miembro del equipo va a implementar una nueva característica o va a realizar un cambio en el producto, puede pedir la colaboración a otro miembro del equipo para trabajar en conjunto.	XP	Actividad	PA_11
IA_18	Integración continua	Todos los cambios que se hacen al producto se prueban e integran después de unas pocas horas.	XP	Actividad	PA_01 PA_03 PA_07
IA_19	Reglas técnicas	Se acuerda un conjunto de reglas que todo el equipo debe cumplir al momento de agregar funcionalidades al producto o realizar modificaciones.	XP	Actividad	PA_09
IA_20	Uso de métricas	Se usan métricas que aporten información relevante sobre el desempeño del equipo.	XP, Kanban	Actividad	PA_08 PA_12

IA_21	Herramientas de integración continua	Se usan herramientas que permitan dar soporte a la integración continua de las nuevas funcionalidades y cambios realizados en el producto	XP	Herramienta	PA_01 PA_03 PA_07 PA_10
IA_22	Pruebas automáticas	Se usan herramientas que permitan dar soporte a la ejecución de pruebas automáticas	XP	Herramienta	PA_01 PA_03 PA_07 PA_10

Acrónimos utilizados:: Id: Identificador, IA: Indicador de Agilidad, PAs: Principios Ágiles

Tabla B.3. Indicadores de agilidad.

A continuación, en la Tabla B.4 se pueden observar los indicadores de agilidad y su relación con los principio ágiles.

		Indicadores de agilidad																					
		IA_01	IA_02	IA_03	IA_04	IA_05	IA_06	IA_07	IA_08	IA_09	IA_10	IA_11	IA_12	IA_13	IA_14	IA_15	IA_16	IA_17	IA_18	IA_19	IA_20	IA_21	IA_22
Principios ágiles	PA_01	✓	✓			✓							✓						✓			✓	✓
	PA_02			✓		✓				✓													
	PA_03	✓	✓								✓								✓			✓	✓
	PA_04			✓							✓		✓		✓	✓							
	PA_05						✓	✓				✓		✓									
	PA_06				✓						✓												
	PA_07		✓																✓			✓	✓
	PA_08								✓												✓		
	PA_09										✓						✓			✓			
	PA_10					✓										✓						✓	✓
	PA_11						✓							✓				✓					
	PA_12				✓						✓										✓		

Tabla B.4. Relación entre principios ágiles e indicadores de agilidad.



## **Anexo C**

### **Cuestionarios de evaluación del grupo focal**

A continuación se presentan los cuestionarios de evaluación de la propuesta, los cuales fueron diligenciados por cada uno de los participantes en el grupo focal.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

**Nombre de la propuesta:** Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	Jhon Eder Masso Daza
Ocupación	Docente.
Estudios	Ingeniero de stmas, Maestría en Ingeniería de software
<b>Describe su experiencia profesional</b>	
<ul style="list-style-type: none"> <li>- Desarrollo de software.</li> <li>- Docente en el área de programación, bases de datos e Ingeniería de Software.</li> </ul>	
<b>Describe su experiencia con enfoques ágiles de desarrollo de software</b>	
<p>Mi experiencia se centra en el uso de Scrum y enfoques tradicionales de desarrollo, tanto para aplicaciones web como empresariales.</p>	

Preguntas sobre la propuesta						
<p>Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:</p>						
<ol style="list-style-type: none"> <li>1. Totalmente en desacuerdo</li> <li>2. En desacuerdo</li> <li>3. Ni de acuerdo ni en desacuerdo</li> <li>4. De acuerdo</li> <li>5. Totalmente de acuerdo</li> </ol>						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?				X	

Figura C.1 Cuestionario de evaluación del participante PF1 – Página 1.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

3	¿Considera que la estructura de AgilityRef es de fácil comprensión?				X	
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?			X		
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?					X
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?					X
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?			X		
8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?				X	
<b>Preguntas abiertas</b>						
9	¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los cambios propuestos.	<p>Considero que se debieran definir indicadores más genéricos que puedan dar cabida a las actividades que se realizan en los diferentes enfoques ágiles.</p> <p>Ejm: Reunión. { - Reunión de Iniciación - Reunión diaria - Reunión de seguimiento - Reunión de Lanzamiento</p> <p>Gestión del cambio { → Requisitos → Producto, etc.</p>				
10	¿Tiene algún comentario adicional acerca de AgilityRef?	<p>Los indicadores sirven mucho pero deberían ser más genéricos para que puedan definir las métricas.</p> <p>También se debería pensar el término de indicador de agilidad, podría pensar en elemento de agilidad.</p> <p>Es un excelente trabajo. Éxitos.</p>				

Figura C.2 Cuestionario de evaluación del participante PF1 – Página 2.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

**Nombre de la propuesta:** Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	Jonathan Guerrero Ortega
Ocupación	Instructor Atea Software
Estudios	Ingeniero de sistemas, Exp en desarrollo de soluciones informáticas, estudiante Maestro en computación, Scrum Master & Scrum Developer
Describe su experiencia profesional	
Desarrollador backend, aplicaciones móviles Instructor en área de desarrollo de software	
Describe su experiencia con enfoques ágiles de desarrollo de software	
Todos los proyectos desarrollados en el SENA usan metodología SCRUM, soy líder de grupo de desarrollo Certificado Scrum Master & Scrum Developer	

Preguntas sobre la propuesta						
Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:						
1. Totalmente en desacuerdo 2. En desacuerdo 3. Ni de acuerdo ni en desacuerdo <input checked="" type="checkbox"/> 4. De acuerdo 5. Totalmente de acuerdo						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?					X

Figura C.3 Cuestionario de evaluación del participante PF2 – Página 1.



Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

3	¿Considera que la estructura de AgilityRef es de fácil comprensión?					X
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?				X	
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?					X
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?				X	
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?				X	
8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?					X
<b>Preguntas abiertas</b>						
9	<p>¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los cambios propuestos.</p> <p>Los indicadores deben tener nombres y descripciones más generales y no amarrados a una fuente específica ya que las organizaciones que no usen esas mismas fuentes no se van a sentir identificadas con el indicador propuesto. por ejemplo: Reunión diario → si mi organización no usa Scrum puede no calzar a ese indicador</p>					
10	<p>¿Tiene algún comentario adicional acerca de AgilityRef?</p> <p>Excelente trabajo.</p>					

Figura C.4 Cuestionario de evaluación del participante PF2 – Página 2.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

**Nombre de la propuesta:** Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	Sandra Lorena Borrón Ruiz
Ocupación	Gerente
Estudios	Magister en computación Especialista en sistemas generales Especialista en redes de comunicación
Describa su experiencia profesional	
Lider de calidad – Lider de enquadramiento de calidad Coordinador de calidad Coordinador de desarrollo	
Describa su experiencia con enfoques ágiles de desarrollo de software	
Lider de proyectos híbridos Implementador de prácticas ágiles	

Preguntas sobre la propuesta						
Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:						
<ol style="list-style-type: none"> <li>1. Totalmente en desacuerdo</li> <li>2. En desacuerdo</li> <li>3. Ni de acuerdo ni en desacuerdo</li> <li>4. De acuerdo</li> <li>5. Totalmente de acuerdo</li> </ol>						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?			X		

Figura C.5 Cuestionario de evaluación del participante PF3 – Página 1.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

3	¿Considera que la estructura de AgilityRef es de fácil comprensión?					X
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?				X	
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?					X
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?				X	
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?					X
8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?					X
<b>Preguntas abiertas</b>						
9	<p>¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los cambios propuestos.</p> <p>– Adicionar relaciones sugeridas en hoja de comentarios</p> <p>– Adicionar tipo de indicador para las reglas de negocio que en un proceso son condicionales y que siempre se ligan con reglas de negocio, que dan valor al usuario.</p> <p>– Los tipos podrían ser entendidos desde varias perspectivas por tanto podrían quitarse.</p>					
10	<p>¿Tiene algún comentario adicional acerca de AgilityRef?</p> <p>¿Que podría pasar con el modelo, en caso que una organización cuente con sus propios valores ágiles y con los cuales viene trabajando se concepte de ser ágiles?</p> <p>Los indicadores podrían ser mejor aspectos de agilidad a evaluar, que luego si podrían pasar a ser indicadores cuando trabajen una medida en el ciclo o fase 2 del proyecto. (Esto bajo la definición de una la ontología de la medición)</p>					

Figura C.6 Cuestionario de evaluación del participante PF3 – Página 2.



Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

**Nombre de la propuesta:** Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	Jaime Adalberto López
Ocupación	Arquitecto líder
Estudios	Ing. Sistemas Esp. Gerencia de Proyectos. Maestría en Ingeniería
<b>Describa su experiencia profesional</b>	
Docente de Ing. Sistemas, Docente Esp en desarrollo de Soluciones Informáticas, Arquitecto líder de Sigo SA	
<b>Describa su experiencia con enfoques ágiles de desarrollo de software</b>	
Dentro de mi experiencia en Sigo adoptamos en 12 años diferentes enfoques ágiles en la actualidad contamos con un proceso híbrido y adaptado a nuestras necesidades y al tamaño del equipo de desarrollo.	

Preguntas sobre la propuesta						
Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:						
1. Totalmente en desacuerdo 2. En desacuerdo 3. Ni de acuerdo ni en desacuerdo 4. De acuerdo ⑤. Totalmente de acuerdo						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?				X	

Figura C.7 Cuestionario de evaluación del participante PF4 – Página 1.



Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

3	¿Considera que la estructura de AgilityRef es de fácil comprensión?					X
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?					X
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?					X
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?					X
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?				X	
8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?				X	
<b>Preguntas abiertas</b>						
9	¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los cambios propuestos.  De acuerdo a lo comentado me gustaría que se pudiera hacer un mapeo entre otras actividades propuestas y las estándares de los modelos ágiles.					
10	¿Tiene algún comentario adicional acerca de AgilityRef?  Ninguno					

Figura C.8 Cuestionario de evaluación del participante PF4 – Página 2.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

Nombre de la propuesta: Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	Hamilton Daniel Jajen Córdoba
Ocupación	Ingeniero de Sistemas - IPS Horisoes
Estudios	Ingeniero de sistemas Scrum Master
Describa su experiencia profesional	
<ul style="list-style-type: none"> <li>Ingeniero de desarrollo en IPS Horisoes donde trabajo actualmente y me desempeño como desarrollador para la plataforma EPIONE. Como Scrum Master apoyo la implementación del marco scrum en el equipo TIC'S.</li> </ul>	
Describa su experiencia con enfoques ágiles de desarrollo de software	
<ul style="list-style-type: none"> <li>Tesis de pregrado en la cual se diseño una guía para evaluar el nivel de cumplimiento de Scrum.</li> <li>Scrum Master certificado en 2019.</li> <li>Apoyo en el proceso de capacitación e implementación del marco en el equipo TIC'S de IPS Horisoes.</li> </ul>	

Preguntas sobre la propuesta						
Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:						
<ol style="list-style-type: none"> <li>Totalmente en desacuerdo</li> <li>En desacuerdo</li> <li>Ni de acuerdo ni en desacuerdo</li> <li>De acuerdo</li> <li>Totalmente de acuerdo</li> </ol>						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?					X

Figura C.9 Cuestionario de evaluación del participante PF5 – Página 1.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgiltyRef

3	¿Considera que la estructura de AgiltyRef es de fácil comprensión?				X	
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?				X	
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?				X	
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?				X	
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?			X		
8	¿Considera que AgiltyRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?				X	
<b>Preguntas abiertas</b>						
9	<p>¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgiltyRef? Por favor, describa los cambios propuestos.</p> <p>Consideraría realizar un agrupamiento en la tabla de indicadores. Lo anterior dado que algunas descripciones podrían agrupar más fuentes. Ejemplo: en la tabla N°3 en el identificador IA-15: "se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los clientes" está relacionada con xp, además de esta fuente puede incluirse el marco scrum dado que al igual que xp propone un lenguaje común para todo el equipo. Lo mismo para otras actividades como: IA-20, IA-21, Aunque estas últimas están relacionadas más hacia el como y no están planteados en la guía de scrum. En tal caso sería interesante diferenciar las fuentes.</p>					
10	<p>¿Tiene algún comentario adicional acerca de AgiltyRef?</p> <p>Excelente trabajo!</p>					

Figura C.10 Cuestionario de evaluación del participante PF5 – Página 2.



Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

### Cuestionario de evaluación de la propuesta

Nombre de la propuesta: Modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones – AgilityRef

Información del participante en el grupo focal	
Nombre	W. Ricardo Pantoja Y
Ocupación	Docente
Estudios	Ing. Sistemas
Describa su experiencia profesional	
Desarrollador x 5 años docente universitario x 16 años	
Describa su experiencia con enfoques ágiles de desarrollo de software	
- 3 años en asignatura de Proyecto II - 2 años en TICs. en el área de desarrollo de SW	

Preguntas sobre la propuesta						
Las siguientes preguntas permiten conocer su opinión acerca de la completitud, comprensibilidad e idoneidad de AgilityRef. Por favor, responda cada pregunta marcando con una X según su grado de conformidad:						
<ol style="list-style-type: none"> <li>1. Totalmente en desacuerdo</li> <li>2. En desacuerdo</li> <li>3. Ni de acuerdo ni en desacuerdo</li> <li>4. De acuerdo</li> <li>5. Totalmente de acuerdo</li> </ol>						
No	Pregunta	Respuesta				
		1	2	3	4	5
1	¿Considera que los indicadores de agilidad propuestos cubren los elementos de proceso (actividades, roles, productos de trabajo y herramientas) que deben estar presentes en un proceso software que implemente los principios ágiles?				X	
2	¿Considera que AgilityRef tiene en cuenta todas las relaciones posibles entre principios y valores ágiles?					X

Figura C.11 Cuestionario de evaluación del participante PF6 – Página 1.

Grupo focal: Evaluación del modelo de referencia para soportar la evaluación de la agilidad en los procesos software de las organizaciones - AgilityRef

3	¿Considera que la estructura de AgilityRef es de fácil comprensión?					X
4	¿Considera que los indicadores de agilidad propuestos son de fácil comprensión?			X		
5	¿Considera que las relaciones propuestas entre valores y principios ágiles son de fácil comprensión?					X
6	¿Considera que las relaciones propuestas entre indicadores de agilidad y principios ágiles son de fácil comprensión?				X	
7	¿Considera que los indicadores de agilidad propuestos sirven como base para la definición de preguntas y métricas que permitan conocer el grado de implementación de los principios ágiles en un proceso software?			X		
8	¿Considera que AgilityRef puede ser usado para la definición de un método de evaluación que permita conocer el grado de agilidad de los procesos software?				X	
<b>Preguntas abiertas</b>						
9	¿Considera que se deben agregar, eliminar o modificar elementos propuestos en AgilityRef? Por favor, describa los cambios propuestos.					
10	<p>¿Tiene algún comentario adicional acerca de AgilityRef?</p> <p>Expresar los indicadores en términos de medidas. Por ejemplo, en lugar de "iterativo e incremental", % de iteraciones que tuvieron entregas en la fecha establecida.</p>					

Figura C.12 Cuestionario de evaluación del participante PF6 – Página 2.



## **Anexo D**

### **Documentación del estudio de caso**

#### **D.1 Aceptación de participación en el estudio de caso**

A continuación, se presenta el documento de aceptación de participación de la empresa en el estudio de caso realizado. Los datos personales del representante de la empresa se mantienen ocultos por razones de confidencialidad.

### Aceptación de la participación en el estudio de caso

**Nombre de la propuesta e evaluar en el estudio de caso:** Método de evaluación de la agilidad de los procesos de desarrollo de software – AgilityPval.

**Nombre del proyecto:** Framework para soportar la evaluación de la agilidad de los procesos software de las organizaciones.

**Investigadores:** Investigador principal: Esp. Ing. Wilson Alfredo Ortega Ordoñez, Director: PhD. César Jesús Pardo Calvache, Codirector: PhD. Francisco José Pino Correa.

**Objetivo del estudio de caso:** Aplicar el método de evaluación AgilityPval en la organización con el fin evaluar su utilidad y practicidad para llevar a cabo la evaluación de la agilidad de los procesos de desarrollo de software en las organizaciones.

**Beneficios del estudio:** El framework propuesto permitirá a la industria de software disponer de una solución para llevar a cabo la evaluación de la agilidad de los procesos de desarrollo de software que sirva como punto de partida para diseñar planes de mejora de sus procesos.

**Riesgos asociados con el estudio:** La investigación no presenta riesgos para los participantes en el estudio. Los datos recolectados se usarán con fines académicos y se conservará el anonimato del nombre de la organización y de los participantes.

**Compensación:** No habrá ningún tipo de retribución económica por participar en la investigación y los resultados de la misma serán utilizados para el mejoramiento de la propuesta.

**Voluntariedad:** Usted está siendo invitado a participar en este estudio de investigación. Antes de decidir si participa o no, debe conocer y comprender cada uno de los siguientes apartados. Este proceso se conoce como consentimiento informado, siéntase con absoluta libertad para preguntar sobre cualquier aspecto que le ayude a aclarar sus dudas al respecto. Una vez que haya comprendido el estudio y si usted desea participar, entonces se le pedirá que firme esta forma de consentimiento, de la cual se le entregará una copia firmada y fechada.

- No habrá ninguna consecuencia desfavorable en caso de no aceptar la invitación.
- La participación es libre y voluntaria; si decide participar en el estudio, puede retirarse en el momento que lo desee, aun cuando el investigador responsable no se lo solicite, informando las razones de su decisión, la cual será respetada en su integridad.
- En el transcurso del estudio usted podrá solicitar información actualizada sobre el mismo, al investigador responsable.
- Si considera que no hay dudas ni preguntas acerca de su participación, puede, si así lo desea, firmar el consentimiento informado que se anexa a este documento.
- La información obtenida solamente será utilizada para la investigación mencionada en el presente documento y ante cualquier inquietud favor comunicarse con los investigadores: Departamento de Sistemas, Universidad del Cauca, teléfono: +57 2 8209800 ext 2145.



### Aceptación de participación

He leído y comprendido la información anterior y mis preguntas han sido respondidas de manera satisfactoria. He sido informado y entiendo que los datos obtenidos en el estudio pueden ser publicados o difundidos con fines científicos. Convengo en participar en este estudio de investigación.

Nombre

Firma: .

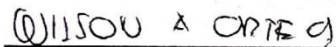
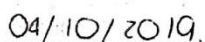
Documento de identificación:

Recibiré una copia firmada y fechada de esta forma de consentimiento.

Esta parte debe ser completada por el Investigador (o su representante):

He explicado al Sr(a). \_\_\_\_\_ la naturaleza y los propósitos de la investigación; le he explicado acerca de los riesgos y beneficios que implica su participación. He contestado a las preguntas en la medida de lo posible y he preguntado si tiene alguna duda. Acepto que he leído y conozco la normatividad correspondiente para realizar investigación con seres humanos y me apego a ella.

Una vez concluida la sesión de preguntas y respuestas, se procederá a firmar el presente documento.

Firma del investigador

Fecha.

Figura D.2 Aceptación de participación en el estudio de caso – Página 2.

## D.2 Instrumento de evaluación diligenciado

En la Tabla D.1., es posible observar el instrumento de evaluación diligenciado por el representante de la organización durante el estudio de caso.

Id.	Indicador de agilidad	¿Se cumple en el proceso de desarrollo? (Marque con una X según corresponda)		Comentarios
		SI	NO	

IA_01	El proceso de desarrollo del producto es iterativo e incremental.	X		
IA_02	Las iteraciones tienen una duración máxima de un mes.	X		
IA_03	Una iteración solamente inicia cuando ha terminado la iteración anterior		X	<p>No al 100%. En algunos proyectos a la medida cuando el cliente final es otra empresa de software, es frecuente que el cliente tenga ambiente de preproducción y producción. De esta forma, el equipo de desarrollo, entrega en el primero de estos ambientes, mientras el cliente hace pruebas en producción. Por efectos de tiempo y costo no nos resulta rentable esperar a que termine la totalidad de las pruebas para continuar.</p> <p>Los proyectos internos si finaliza una iteración antes de iniciar la siguiente.</p>
IA_04	Una iteración se planifica para que tenga como resultado una versión funcional del producto que sea útil para el cliente.	x		
IA_05	El alcance de cada iteración puede clarificarse y renegociarse entre el equipo y el cliente.	x		
IA_06	El equipo se reúne diariamente para evaluar el progreso hacia el objetivo de la iteración.	x		
IA_07	La duración de la reunión diaria se limita a un tiempo que permita a cada participante compartir de manera concisa lo que hizo el día anterior, lo que tiene planeado hacer ese día y los impedimentos para lograr el objetivo de la iteración.		x	<p>Aunque se realizan los dailys de seguimiento, algunas veces se extiende más del tiempo necesario por trabajar regularmente con equipos distribuidos y resultan más temas de los esperados.</p>

IA_08	La lista de requisitos del producto se actualiza constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.	x		Aunque la respuesta es sí, resulta difícil limitar esa opción de actualización al cliente para que esta oportunidad de ingresar nuevas características a la pila de producto o repriorizarla sin que se convierta en inconvenientes para el equipo de desarrollo.
IA_09	Los requisitos pueden ser actualizados en cualquier momento para incluir más detalles, estimaciones y prioridades.	x		Es posible, pero igual que el punto anterior, es difícil manejarlo con el cliente.
IA_10	Cada integrante del equipo puede elegir la mejor forma de llevar a cabo las tareas que tenga bajo su responsabilidad.		x	No se ha implementado esta opción hasta el momento en la empresa.
IA_11	El equipo tiene la posibilidad de dar mayor prioridad a los requisitos del producto que presenten más riesgo.		x	En común acuerdo con el cliente, se hace. No de manera unilateral por el equipo de desarrollo.
IA_12	Los miembros del equipo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad del producto es de todo el equipo.	x		
IA_13	Cualquier miembro del equipo podría realizar cambios en cualquiera de los componentes del producto.		x	Procuramos que la persona que desarrollo determinado componente, sea quien realice cambios en este.
IA_14	El tiempo de duración de las iteraciones es consistente durante todo el desarrollo del producto.		x	Ha sido difícil porque aún no tenemos una manera de medir velocidad del equipo y a veces no se cumplen al 100% los tiempos de estimación, viéndose esto reflejado en tiempos de desarrollo de una iteración
IA_15	Una iteración se puede cancelar cuando se determina que su objetivo ha quedado obsoleto.	x		Se puede hacer, aunque hasta el momento no ha ocurrido en alguno de los proyectos.

IA_16	Al finalizar cada iteración, el equipo y los interesados se reúnen para revisar los resultados con respecto a las personas, procesos y herramientas.	x		
IA_17	En la reunión llevada a cabo al finalizar cada iteración, se identifican y ordenan los elementos más importantes que salieron bien y las posibles mejoras, se crea un plan para implementar las mejoras, y se adapta la lista de requisitos del producto en caso de ser necesario.		x	En los proyectos internos sí. En los proyectos a la medida cuando un cliente es una empresa de desarrollo de software, no hemos logrado realizar el ejercicio hasta el momento.
IA_18	Existe un rol responsable de ayudar al equipo y a los interesados a ejecutar las actividades de la mejor manera y eliminar los impedimentos que se presenten.	x		
IA_19	El cliente o un representante suyo con poder de decisión, participa en la priorización de los requisitos que serán implementados en cada iteración.	x		
IA_20	El cliente o un representante suyo con poder de decisión, participa en el diseño de las pruebas funcionales del producto.		x	Algunas veces, pero es una tarea difícil con los clientes.
IA_21	El cliente o un representante suyo con poder de decisión, participa en la elección de la tecnología a usar para implementar el producto.	x		

IA_22	El cliente o un representante suyo con poder de decisión, está disponible en la misma ubicación del equipo de desarrollo del producto para resolver inquietudes con respecto a los requisitos y prioridades para el negocio.		x	Por nuestra modalidad de trabajo remoto, no se ha presentado esta oportunidad.
IA_23	El equipo es el encargado de estimar el tiempo que se empleará para implementar una característica del producto.	x		
IA_24	Cuando el equipo tome decisiones técnicas que puedan tener impacto a nivel de negocio, debe informar a los interesados.	x		
IA_25	Se define un vocabulario y una visión del producto a desarrollar que se comparte por el equipo y los clientes.	x		
IA_26	Durante el diseño del producto se tiene en cuenta que no se dupliquen componentes, se incluyen todos los aspectos que el equipo considera importantes y se incluyen solamente lo que se requiere en el momento.		x	Se busca que solo se incluyan los aspectos necesarios, pero al momento de la ejecución pueden faltar o sobrar algunos componentes o artefactos.
IA_27	Cada vez que se agrega una nueva característica al producto, se verifica si se puede modificar su estructura interna para hacerla más sencilla, en caso de ser posible, se realizan los cambios y se verifica que todas las pruebas funcionen.	x		

IA_28	Cuando un miembro del equipo va a implementar una nueva característica o va a realizar un cambio en el producto, puede pedir la colaboración a otro miembro del equipo para trabajar en conjunto.	x		
IA_29	Después de unas pocas horas todos los cambios que se hacen al producto se prueban e integran.		x	Este tiempo no ha sido medido para establecer si el término "pocas horas" se cumple o no.
IA_30	Se acuerda un conjunto de reglas que todo el equipo debe cumplir al momento de agregar funcionalidades al producto o realizar modificaciones.		x	Se tiene el conjunto de reglas y condiciones, pero ha sido un proceso lograrlo con algunos miembros del equipo.
IA_31	Se usan métricas que aporten información relevante sobre el desempeño del equipo.		x	No se tiene y es una necesidad inmediata para la compañía.
IA_32	Se usan herramientas que permitan dar soporte a la integración continua de las nuevas funcionalidades y cambios realizados en el producto.	x		
IA_33	Se usan herramientas que permitan dar soporte a la ejecución de pruebas automáticas.		x	No se cuenta con las herramientas.

Tabla D.1. Instrumento de evaluación diligenciado.