



UNIVERSIDAD INTERNACIONAL DEL ECUADOR

Facultad de Ciencias Administrativas y Económicas

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE
MAGÍSTER EN ADMINISTRACIÓN DE NEGOCIOS

DIAGNÓSTICO DE FLUIDEZ ÁGIL EN UNA EMPRESA DE
DESARROLLO DE SOFTWARE DEL ECUADOR

AUTOR: Ing. Iván Alejandro Martínez Cueva

DIRECTOR: Mgtr. Galo Santiago Jácome Sandoval

Marzo, 2020

Quito-Ecuador

CERTIFICACIÓN

Yo, IVÁN ALEJANDRO MARTÍNEZ CUEVA, declaro que soy el autor exclusivo de la presente investigación; y, que ésta es original, es auténtica y personal. Para todos los efectos académicos y legales que se desprendan de la presente investigación serán de mi sola y exclusiva responsabilidad.

Cedo mis derechos de propiedad intelectual a la UIDE, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.



Firma del Graduando

Iván Alejandro Martínez Cueva

Yo, GALO SANTIAGO JÁCOME SANDOVAL, declaro que, en lo que yo personalmente conozco, el señor IVÁN ALEJANDRO MARTÍNEZ CUEVA, es el autor exclusivo de la presente investigación y que ésta es original, auténtica y personal.



Firma del Director

Galo Santiago Jácome Sandoval

DEDICATORIA

A mis padres Nelson y Lucita, y mis hermanos Luis, Carmen, María Elena.

AGRADECIMIENTO

La idea de este tema de tesis es el resultado de mi colaboración en Logiciel, Produbanco, Cobiscorp, Tata Consultancy Services, Thoughtworks y Oshyn, empresas en las que he trabajado en roles técnicos vinculados al desarrollo de software (desarrollador, arquitecto y consultor) y en las que he interactuado con roles administrativos, en cada lugar con sus estilos y peculiaridades. Estoy muy agradecido por la experiencia recibida, siendo cada experiencia un aprendizaje.

También agradezco a la UIDE, sus autoridades y maestros, y a mis compañeros de la novena, décimo tercera y décimo cuarta promociones, que han permitido que aprenda de una forma estructurada el conocimiento administrativo que es un complemento a mis raíces técnicas.

Un agradecimiento especial para Santiago Jácome, director de tesis, y para los principales ejecutivos de la empresa objeto de estudio que permitieron y guiaron la realización de esta investigación.

ÍNDICE GENERAL

CAPÍTULO I PLAN DE INVESTIGACIÓN	1
1.1 PLANTEAMIENTO, FORMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA	1
1.1.1 Planteamiento del problema	1
1.1.1.1 Diagnóstico	1
1.1.1.2 Pronóstico	2
1.1.2 Formulación del problema	2
1.1.3 Sistematización	3
1.2 OBJETIVOS	3
1.2.1 Objetivo general	3
1.2.2 Objetivos específicos	4
1.3 JUSTIFICACIÓN	4
1.3.1 Justificación Teórica	4
1.3.2 Justificación Práctica	5
 CAPÍTULO II MARCO TEÓRICO	 6
2.1 MÉTODOS DE DESARROLLO DE SOFTWARE	6
2.1.1 El Modelo en Cascada	7
2.1.2 Métodos Ágiles	7
2.1.2.1 Manifiesto de Desarrollo de Software Ágil	8
2.1.2.2 Mentalidad Ágil y mentalidad burocrática	10
2.1.2.3 Situación de la adopción de Agile	10
2.1.2.4 Principales retos en la adopción de Agile	11
2.1.3 Pensamiento sistémico	12

2.1.3.1	Implicaciones para los programas de mejora	13
2.1.4	Herramientas para cada situación	13
2.2	EL MODELO DE FLUIDEZ ÁGIL	14
2.2.1	Equipos Enfocados	15
2.2.1.1	Necesidades del negocio	15
2.2.1.2	Trabajo efectivo como equipo	16
2.2.1.3	Maestría del equipo	17
2.2.2	Equipos que Entregan	17
2.2.2.1	Necesidades del negocio	17
2.2.2.2	Trabajo efectivo como equipo	18
2.2.2.3	Excelencia técnica	18
2.2.3	Equipos Optimizados	19
2.2.3.1	Necesidades del negocio	19
2.2.3.2	Trabajo efectivo como equipo	19
2.2.3.3	Excelencia de producto	20
2.2.4	Equipos Fortalecidos	21
2.2.5	Zonas según los requerimientos de cada organización	21
2.3	MÉTRICAS ESENCIALES	22
2.3.1	Prioridades y progreso incremental	23
2.3.2	Trabajo reflexivo y estimaciones útiles	23
2.3.3	Dirección del producto	25
2.3.4	Mejora organizacional	26
CAPÍTULO III OBJETO DE ESTUDIO		27
3.1	CONTEXTO ORGANIZACIONAL	27
3.1.1	Desempeño económico fiscal del sector	28
3.2	LA ESTRATEGIA	28
3.3	LA NUEVA METODOLOGÍA	32
3.3.1	Wave	33
3.4	EL PROYECTO	34
3.4.1	Revisión de forma de trabajo	35

3.4.2	Iteraciones	37
3.4.3	Burn-down	38
CAPÍTULO IV DIAGNÓSTICO DE FLUIDEZ ÁGIL		41
4.1	DISEÑO DE INSTRUMENTOS	41
4.1.1	Entrevistas	41
4.1.2	Observación	44
4.1.2.1	Equipos Enfocados	44
4.1.2.2	Equipos que Entregan	45
4.1.2.3	Equipos Optimizados	46
4.1.2.4	Equipos Fortalecidos	47
4.1.3	Índices	47
4.2	RESULTADOS	48
4.2.1	Entrevistas	48
4.2.1.1	Metodología de desarrollo utilizada	48
4.2.1.2	Certificaciones en la metodología utilizada	57
4.2.1.3	Utilidad de la metodología	57
4.2.1.4	Reconocimiento de logros	57
4.2.1.5	Problema más importante	58
4.2.1.6	Seguimiento de avance	59
4.2.2	Observación	59
4.2.2.1	Equipos Enfocados	59
4.2.2.2	Equipos que Entregan	62
4.2.2.3	Equipos Optimizados	65
4.2.2.4	Equipos Fortalecidos	66
4.2.3	Índices	68
4.2.3.1	Equipo Enfocados	68
4.2.3.2	Equipos que Entregan	68
4.2.3.3	Equipos Optimizados	71
4.3	ANÁLISIS-SÍNTESIS	77
4.3.1	Equipos Enfocados	78

4.3.1.1	Necesidades del negocio	78
4.3.1.2	Trabajo efectivo como equipo	80
4.3.1.3	Maestría del equipo	82
4.3.1.4	Diagnóstico	84
4.3.2	Equipos que Entregan	85
4.3.2.1	Necesidades del negocio	85
4.3.2.2	Trabajo efectivo como equipo	89
4.3.2.3	Excelencia técnica	91
4.3.2.4	Diagnóstico	93
4.3.3	Equipos Optimizados	94
4.3.3.1	Necesidades del negocio	94
4.3.3.2	Trabajo efectivo como equipo	95
4.3.3.3	Excelencia de producto	97
4.3.3.4	Diagnóstico	98
4.3.4	Equipos Fortalecidos	99
4.3.5	Diagnóstico por dimensiones	100
4.3.6	Zona de acuerdo a competencias	102
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		104
5.1	CONCLUSIONES	104
5.2	RECOMENDACIONES	106
	BIBLIOGRAFÍA	108

ÍNDICE DE FIGURAS

3.1	Índice de nivel de actividad registrada 2019	29
3.2	Índice de nivel de actividad registrada 2018	29
3.3	Índice de nivel de actividad registrada 2017	30
3.4	Cuadrante mágico de Gartner 2018 para Plataformas de Experiencia Digital	31
3.5	Sitecore Market Share	31
3.6	Principales países que usan Sitecore	32
3.7	Dos equipos de desarrollo	36
3.8	Iteraciones	39
3.9	Burn-down	40
3.10	Burn-down - Tendencia lineal	40
4.1	Funcionalidades Priorizadas por Iteración - Equipos Enfocados	68
4.2	Funcionalidades Finalizadas por Iteración - Equipos Enfocados	69
4.3	Velocidad, velocidad promedio - Equipos Enfocados	69
4.4	Esfuerzo pendiente - Equipos Enfocados	70
4.5	Esfuerzo pendiente porcentual - Equipos Enfocados	70
4.6	Nuevas funcionalidades - Equipos que Entregan	71
4.7	Corrección de errores - Equipos que Entregan	72
4.8	Tipo de tarea - Equipos que Entregan	72
4.9	Lead Time - Equipos que Entregan	73
4.10	Mantenibilidad - Equipos que Entregan	73
4.11	Complejidad ciclomática - Equipos que Entregan	74
4.12	Acoplamiento - Equipos que Entregan	74
4.13	Niveles de herencia - Equipos que Entregan	75
4.14	Builds - Equipos que Entregan	75

4.15 Pruebas - Equipos que Entregan	76
4.16 Cobertura de código - Equipos que Entregan	76
4.17 Despliegue a producción - Equipos que Entregan	77
4.18 Diagnóstico Equipos Enfocados	85
4.19 Diagnóstico Equipos que Entregan	94
4.20 Diagnóstico Equipos Optimizados	98
4.21 Diagnóstico por dimensiones	102

ÍNDICE DE TABLAS

3.1	Herramientas del proyecto	36
4.1	Fuentes de información - Equipos Enfocados	42
4.2	Fuentes de información - Equipos que Entregan	43
4.3	Fuentes de información - Equipos Optimizados	43
4.4	Fuentes de información - Equipos Fortalecidos	44
4.5	KPIs - Prioridades y progreso incremental	49
4.6	KPIs - Trabajo reflexivo y estimaciones útiles	50
4.7	KPIs - Calidad de código	51
4.8	Índice de mantenibilidad	52
4.9	Índice de complejidad ciclomática	52
4.10	Índice de acoplamiento	52
4.11	Índice de niveles de herencia	52
4.12	KPIs - Integración Continua	53
4.13	KPIs - Despliegue	54
4.14	KPIs - Dirección del producto	55
4.15	KPIs - Análisis de cohorte	56
4.16	Lista de chequeo - Equipos Enfocados	60
4.17	Lista de chequeo - Equipos que Entregan	62
4.18	Lista de chequeo - Equipos Optimizados	65
4.19	Lista de chequeo - Equipos Fortalecidos	66
4.20	Diagnóstico - Equipos Enfocados - Necesidades del negocio	80
4.21	Diagnóstico - Equipos Enfocados - Trabajo efectivo como equipo	82
4.22	Diagnóstico - Equipos Enfocados - Maestría del equipo	84
4.23	Builds exitosos diarios y por desarrollador	86

4.24	Diagnóstico - Equipos que Entregan - Necesidades del negocio	89
4.25	Diagnóstico - Equipos que Entregan - Trabajo efectivo como equipo . . .	91
4.26	Diagnóstico - Equipos que Entregan - Excelencia técnica	93
4.27	Diagnóstico - Equipos Optimizados - Necesidades del negocio	95
4.28	Diagnóstico - Equipos Optimizados - Trabajo efectivo como equipo . . .	96
4.29	Diagnóstico - Equipos Optimizados - Excelencia de producto	98
4.30	Diagnóstico - Equipos Fortalecidos	99
4.31	Diagnóstico por dimensiones - Negocio	100
4.32	Diagnóstico por dimensiones - Equipo	101
4.33	Diagnóstico por dimensiones - Mejora	101

RESUMEN

El software se está comiendo el mundo. En la actualidad cualquier empresa, sin importar la industria a la que pertenezca, es una empresa de software.

Este trabajo trata acerca de la relación entre las partes administrativa y técnica: el tipo de administración requerida en las empresas de desarrollo de software, considerando sus complejidades.

El desarrollo de software es una actividad compleja porque no se conocen desde el inicio todos los elementos y sobre todo sus relaciones: recursos tecnológicos, conocimiento técnico, personas involucradas, ambiente y necesidades cambiantes.

Las Metodologías Ágiles de desarrollo de software tratan la gestión de los proyectos de desarrollo de software considerando el pensamiento complejo requerido. Es un camino que se asemeja mucho a la vida real. No existen recetas únicas y predefinidas. Se toman decisiones y se cambian los planes considerando la realidad, no se busca cambiar la realidad para que coincida con nuestros planes. Ésto último es la gestión por milagro, que no ha sido rara en los proyectos de desarrollo de software.

El marco de trabajo escogido es el Modelo de Fluidez Ágil que describe las características requeridas en los equipos de desarrollo de software y los principales involucrados. Se usa como caso práctico la realidad de una empresa del mercado local que inicia su camino en la adopción de Metodologías Ágiles.

Al final se realizan recomendaciones, principalmente a la administración de la empresa, que es donde se han encontrado las principales dificultades para una correcta adopción. Los cambios principales son de carácter cultural. Esfuerzos individuales no pueden trascender si no existe un auténtico compromiso y un cambio de mentalidad de parte de la alta gerencia.

Palabras claves: diagnóstico de fluidez; metodologías ágiles; pensamiento complejo.

ABSTRACT

The software is eating the world. Today, any company, regardless of the industry to which it belongs, is a software company.

This work is about the relationship between the administrative and technical parties: the type of administration required in software development companies, considering their complexities.

Software development is a complex activity because all the elements and especially their relationships are not known from the beginning: technological resources, technical knowledge, people involved, environment and changing needs.

Agile software development methodologies address the management of software development projects considering the complex thinking required. It is a path that closely resembles real life. There are no unique and predefined recipes. Decisions are made and plans are changed considering reality, we do not seek to change reality to match our plans. The latter is miracle management, which has not been rare in software development projects.

The framework chosen is the Agile Fluency Model that describes the characteristics required in the software development teams and the main stakeholders. It is used as a case study the reality of a company in the local market that begins its path in the adoption of Agile Methodologies.

In the end, recommendations are made, mainly to the administration of the company, which is where the main difficulties for a correct adoption have been found. The main changes are cultural. Individual efforts cannot transcend if there is no genuine commitment and change of mentality on the part of senior management.

Key words: fluency diagnosis; agile methodologies; complex thinking.

CAPÍTULO I

PLAN DE INVESTIGACIÓN

1.1 PLANTEAMIENTO, FORMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA

1.1.1 Planteamiento del problema

1.1.1.1 *Diagnóstico*

El Buen Vivir es un concepto de bienestar que va más allá del desarrollo económico, se complementa por el cumplimiento de los derechos humanos, sociales, culturales y ambientales. Dentro de este marco se han detectado problemas como la matriz productiva primario exportadora y la ausencia de las capacidades necesarias en la ciudadanía para lograr la transformación de dicha matriz a una que se base en el conocimiento (Senplades, 2013).

La aplicación del paradigma Agile en las empresas de desarrollo de software buscan que éstas se adapten al entorno global cambiante, y de esta forma sean más competitivas, generen valor, y puedan perdurar. Esto se enmarca en el Objetivo 10 del Plan Nacional para el Buen Vivir: Impulsar la transformación de la matriz productiva.

Para esto se requiere un cambio cultural y re-pensar la forma de trabajar dentro de las empresas, por lo que esta investigación estudia la transferencia de este tipo de innovación al mercado nacional y mide la capacidad doméstica de asimilación, enmarcados en el Objetivo 4 del Plan Nacional para el Buen Vivir: Fortalecer las capacidades y potencialidades de la ciudadanía.

Las empresas de desarrollo de software se encuentran inmersas en un entorno global cambiante: necesidades variables y no descubiertas de los usuarios, condiciones volátiles

del mercado, nuevas tecnologías. En consecuencia deben tener la capacidad de adaptarse rápidamente para afrontar dichos cambios (Humble, Molesky, y O'Reilly, 2014).

Para este estudio se ha escogido una empresa de desarrollo de software bi-nacional, a la que llamaremos Hyna por razones de confidencialidad, con oficina central en Long Beach, California, Estados Unidos de Norteamérica, y sucursal en Quito, Ecuador. Durante los últimos 15 años ha construido su reputación creando soluciones de marketing digital para sus clientes. Se ha asociado con agencias de marketing reconocidas, a las que provee servicios, tecnología y plataformas que mejoran sus ofertas. En su trayectoria la empresa ha aplicado un modelo de trabajo tradicional con roles definidos, actividades establecidas y equipos independientes.

En 2018 Hyna inició una transformación orientada a la adopción de Metodologías Ágiles. Emplean prácticas y herramientas que se usan en equipos ágiles, como por ejemplo tablero Kanban, reuniones de pie, reuniones de retrospectiva. Como parte de su cultura de mejora, busca que estas prácticas sean evaluadas, que se apliquen de una forma más efectiva los principios del Paradigma Agile, que se tenga clara su utilidad dentro de un contexto que busca resolver los problemas del cliente en una forma alineada con la estrategia de la organización.

Los resultados de la empresa no siempre han sido los que se esperan en cuanto a la satisfacción del cliente, tanto por la calidad de los productos y servicios como por los plazos de entrega.

1.1.1.2 Pronóstico

Si la empresa no es capaz de adaptarse a las condiciones cambiantes del mercado corre el riesgo de verse relegada por empresas nacionales o internacionales que puedan proveer servicios de mejor calidad, entregando valor desde el inicio y en el tiempo que requiere el cliente.

1.1.2 Formulación del problema

¿Qué es lo que debe hacer una empresa de software que desea aplicar el Paradigma Agile para poder adaptarse a las condiciones cambiantes del mercado?

1.1.3 Sistematización

- ¿Qué es el Paradigma Agile?
- ¿Qué condiciones están cambiando actualmente en el mercado?
- ¿Cómo la aplicación del Paradigma Agile permite que una empresa de software pueda adaptarse al cambio?
- ¿Qué es un modelo de madurez de agilidad?

¿Qué factores dentro de una organización determinan el nivel de madurez en cuanto a agilidad?

A nivel operativo

A nivel estratégico

A nivel de liderazgo

A nivel de innovación

¿Cómo se aplica un modelo de madurez en una empresa de software?

- ¿Cuál es la situación actual de Hyna?
 - ¿Cómo ha implementado Hyna el paradigma agile?
- ¿Cuál es el nivel de madurez de Hyna en cuanto a la implementación del Paradigma Agile?
- ¿Qué significa este nivel para Hyna?

1.2 OBJETIVOS

1.2.1 Objetivo general

Determinar las características que debe desarrollar una empresa de software para aplicar el Paradigma Agile.

1.2.2 Objetivos específicos

- a) Explicar en qué consiste el Paradigma Agile.
- b) Indicar qué condiciones cambian continuamente en la industria del software.
- c) Identificar los beneficios que trae la aplicación del Paradigma Agile en una empresa que busca adaptarse a las condiciones cambiantes del mercado.
- d) Describir un modelo de madurez de agilidad y los factores que lo determinan.
- e) Describir la situación actual de Hyna.
- f) Aplicar un modelo de madurez de agilidad en Hyna.
- g) Determinar el nivel de madurez de agilidad de Hyna.
- h) Analizar el nivel de madurez alcanzado por Hyna.

1.3 JUSTIFICACIÓN

1.3.1 Justificación Teórica

Los conceptos de madurez y agilidad pueden considerarse como opuestos (Schweigert, Nevalainen, Vohwinkel, Korsaa, y Biro, 2012). Desde el punto de vista de procesos, la madurez es el nivel que alcanza una organización de acuerdo al grado de cumplimiento de procesos y prácticas predefinidas, como es el caso del modelo CMMI (Capability Maturity Model Integration) que describe procesos y buenas prácticas para el desarrollo de software (Schweigert et al., 2012). Desde el punto de vista ágil se valora más a los individuos y sus interacciones que a los procesos predefinidos. (Beck et al., 2001).

Lo anterior no implica la ausencia de procesos en Agile, si no que se los valora en menor medida. En este contexto, un modelo de madurez de agilidad constituye una herramienta que puede ser usada dentro de las organizaciones para comprender el grado de aplicación de las prácticas de desarrollo de software, procesos organizacionales y de gestión de proyectos que siguen los principios ágiles. (Schweigert, Vohwinkel, Korsaa, Nevalainen, y Biro, 2013).

El modelo escogido para realizar este estudio es el modelo de Fluidez Ágil (Larsen y Shore, 2019). Es un marco de trabajo usado en varias empresas, entre ellas una de las consultoras más reconocidas en el mundo: Thoughtworks (van Haaster, 2020). Esta empresa se la reconoce como la pionera en los métodos de desarrollo de software ágiles (Thoughtworks, 2020). De entre los creadores del Manifiesto Ágil (Beck et al., 2001), la mayoría de sus integrantes está o ha estado relacionado con esta empresa.

1.3.2 Justificación Práctica

En un estudio sobre innovación institucional realizado por Deloitte University Press se menciona que el promedio actual de vida de las 500 empresas S&P en Estados Unidos es de menos de 15 años, en comparación con un promedio de 75 años de vida en 1930, lo que constituye un 80 % de reducción del promedio de vida. El estudio resalta que la medida más básica de desempeño de una empresa es sobrevivir (Hagel y Brown, 2013).

La aplicación de un modelo de madurez en Hyna busca que la empresa mejore su capacidad de adaptación para que pueda perdurar en un entorno cambiante. Esto lo logrará mediante la determinación del nivel de madurez de agilidad en que se encuentra y en base a esto trazar planes de mejora.

Si la empresa mejora su habilidad de adaptación estará en la capacidad de proveer productos y servicios de valor a sus clientes desde el inicio del proyecto e implementar eficazmente cambios cuando lo requiera el cliente.

Los clientes de Hyna se beneficiarán porque su software constituirá una ventaja competitiva. Sus clientes finales estarán satisfechos con productos que satisfagan sus necesidades.

Finalmente si la empresa logra mantenerse en el tiempo y crecer beneficiará al entorno laboral manteniendo y generando fuentes de trabajo de calidad.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se realizará una breve revisión de conceptos relacionados con las metodologías de desarrollo de software. Se determinará la importancia de los Métodos Ágiles en este contexto y el Modelo de Fluidez Ágil como marco de evaluación de adopción de estos métodos. Luego se determinarán los principales indicadores derivados del Modelo de Fluidez Ágil.

El capítulo finalizará con una revisión del pensamiento sistémico y su implicación en los cambios organizacionales.

2.1 MÉTODOS DE DESARROLLO DE SOFTWARE

En esta sección se revisarán los principales Métodos de Desarrollo de Software. También se incluye una sección que trata sobre el pensamiento sistémico por su implicación en los programas de mejora, en este caso el que se aplica en las organizaciones que buscan acoger nuevas formas de trabajo y organización como Agile.

Los Métodos de Desarrollo de Software denotan una colección de prescripciones interconectadas para construir sistemas de una forma efectiva y connotan enfoques ordenados y predecibles que se aplican en entornos diversos (Ralph, 2018).

Estos métodos se dividen principalmente en dos grupos (Ralph, 2018):

- Métodos Ágiles, que incluyen SCRUM, Extreme Programming, Lean, entre otros.
- Métodos de peso pesado, que incluyen Rational Unified Process, PRINCE2, el Modelo en Cascada, entre otros.

2.1.1 El Modelo en Cascada

Este es uno de los modelos más tradicionales. Organiza las actividades de desarrollo típicamente en: requerimientos, análisis, diseño, codificación y pruebas (Ralph, 2018). Su nombre es una metáfora que compara su proceso con el de las cascadas: el trabajo inicia con la primera etapa y continúa secuencialmente con las siguientes, antes de empezar una etapa se debe haber terminado completamente la anterior y no se puede regresar a etapas anteriores.

Este modelo se caracteriza por su estructura de proceso riguroso. La razón para esto es que se derivó de los procesos de las industrias de manufactura y construcción en los que cambios en la implementación del proceso resultan en costos muy elevados (Muslihat, 2018).

El modelo en Cascada se lo considera dentro los Paradigmas Racionales. Según estos paradigmas, el problema es bien conocido desde el inicio y el objetivo del sistema es claro (Ralph, 2018).

Ciertamente en otras industrias esto funciona bien e incluso dentro de la industria de software también funciona. La premisa es que existan requerimientos de ingeniería rigurosos, como por ejemplo en proyectos que se ejecutan en la NASA (Zwinkau, 2018).

En el contexto de desarrollo de software, estos paradigmas consideran a los desarrolladores como ejecutores de planes. La investigación empírica por otro lado muestra que los planes no son el mejor recurso para tomar decisiones y acciones informadas en entornos altamente cambiantes, en los que además los requerimientos no son elaborados de una forma técnicamente rigurosa (Ralph, 2018).

2.1.2 Métodos Ágiles

Los Métodos Ágiles parecen ser más consistentes con los Paradigmas Empíricos. Su narrativa es que no existe un problema conocido, sino una situación que es percibida como problemática desde diferentes puntos de vista que corresponden a cada una de las personas involucradas o que tienen interés en él. Los analistas del negocio trabajan con todos los interesados, cada uno con su objetivo, para construir ideas y preferencias para el posible sistema. La comprensión de la situación y los posibles candidatos de diseño evolucionan en

conjunto. Cada proyecto presenta una única secuencia de eventos que no necesariamente se asemeja a modelos de procesos o métodos conocidos, los eventos no esperados son comunes, las personas involucradas deben improvisar (Ralph, 2018).

En entornos cambiantes los requerimientos y soluciones evolucionan. Los Métodos Ágiles abrazan el cambio y consideran una ventaja competitiva la capacidad de adaptación al cambio (Beck et al., 2001).

La esencia de los Métodos Ágiles descansa sobre dos ideas que contrastan con el proceso orientado a planes tradicional, de acuerdo a la Guía de Desarrollo de Software Ágil realizada por Fowler (2019):

- Es adaptativo en lugar de predictivo.
- Está orientado a las personas en lugar de orientado al proceso.

En los Métodos Ágiles, los planes son tan cuidadosamente elaborados como en métodos tradicionales, pero se diferencian en que son constantemente revisados para reflejar lo que se va aprendiendo durante el proceso (Fowler, 2019).

Los Métodos Ágiles ven al desarrollo de software como una actividad principalmente humana, donde las personas involucradas, colaborando como un equipo, son el elemento principal para alcanzar el éxito. Los procesos y las herramientas mejoran la efectividad, pero influyen en menor grado (Fowler, 2019).

2.1.2.1 Manifiesto de Desarrollo de Software Ágil

El Manifiesto de Desarrollo de Software Ágil sintetiza las características comunes de los Métodos Ágiles en sus 4 postulados (Beck et al., 2001):

- Valorar a los individuos y sus interacciones más que a los procesos y herramientas.
- Valorar al software funcionando más que a la documentación extensiva.
- Valorar la colaboración con el cliente más que a la negociación de contratos.
- Valorar la respuesta ante el cambio más que al seguimiento de un plan.

Esto no implica que los procesos, herramientas, documentación, negociación de contratos y planes no tengan valor, sino que los individuos, el software funcionando, la interacción con el cliente y la respuesta al cambio, tienen mayor valor.

El Manifiesto Ágil se basa en los siguientes principios (Beck et al., 2001).

- Nuestra más alta prioridad es satisfacer al cliente a través de una entrega temprana y continua de software de valor.
- Los requerimientos cambiantes son bienvenidos, aun tarde en el proceso de desarrollo. Los procesos ágiles aprovechan el cambio para dar una ventaja competitiva al cliente.
- Entregar software funcionando de forma frecuente, desde un par de semanas a un par de meses, con preferencia al intervalo de tiempo más corto.
- Representantes del negocio y desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
- Construir proyectos alrededor de individuos motivados. Entregarles el ambiente y soporte que ellos necesitan, y confiar en ellos para que cumplan su trabajo.
- El método más eficiente y efectivo para transmitir información hacia y dentro de un equipo de desarrollo es la comunicación cara a cara.
- El software que funciona es la medida primaria de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían mantener un ritmo constante indefinidamente.
- La atención continua a la excelencia técnica y buen diseño mejora la agilidad.
- Simplicidad —el arte de maximizar la cantidad de trabajo no hecho— es esencial.
- Las mejores arquitecturas, requerimientos y diseños surgen de equipos auto-organizados.
- En intervalos regulares, el equipo reflexiona sobre cómo ser más efectivos, luego ajusta su comportamiento en consecuencia.

2.1.2.2 *Mentalidad Ágil y mentalidad burocrática*

Denning (2019) hace una distinción entre las mentalidades Ágil y burocrática desde 3 áreas: cuál es el objetivo, cómo se ejecuta el trabajo y cómo es la estructura organizacional de cada una. A continuación se detallan éstas.

Según Denning (2019) el objetivo en la mentalidad Ágil está regido por la ley del cliente: entrega continua y temprana de valor al cliente. La mentalidad burocrática se rige por la ley del accionista: el objetivo es hacer dinero para la empresa y maximizar el valor para los accionistas.

Si consideramos la forma de trabajar, la guía de la mentalidad Ágil es la ley de los equipos pequeños: todo el trabajo se lleva a cabo por equipos pequeños auto-organizados, que trabajan en iteraciones cortas, y se enfocan en entregar valor al cliente. La ley de la burocracia dictamina la mentalidad burocrática: los individuos reportan a sus jefes, quienes definen los roles y reglas del trabajo y criterios de desempeño.

La estructura organizacional en la mentalidad Ágil se caracteriza por la ley del trabajo en red: la empresa opera como una red de equipos que interactúan. En contraste la mentalidad burocrática considera que la organización opera mediante una jerarquía de arriba hacia abajo, con múltiples capas y divisiones.

2.1.2.3 *Situación de la adopción de Agile*

Estadísticas recabadas de una variedad de estudios durante los últimos años demuestran un incremento en la adopción de Agile.

Al ser consultados los representativos de 4455 organizaciones alrededor del mundo sobre el tipo de enfoque utilizado en proyectos completados en su organización durante los últimos 12 meses, el 23 % respondieron que usan Agile, 23 % enfoques híbridos Agile-predictivo y 47 % enfoques predictivos (PMI, 2018).

Si bien los enfoques predictivos son mayoría, los enfoques que consideran nuevas formas de trabajo como Agile, ya sea de forma pura o híbrida, prácticamente los han igualado. Los ejecutivos consultados pertenecen a un gran rango de industrias. Cabe resaltar que entre estas industrias no solamente está incluída tecnologías de la información (IT), además se encuentran representadas las siguientes industrias: telecomunicaciones, energía,

gobierno, manufactura, salud, construcción.

De acuerdo al mismo estudio realizado por PMI (2018), el 30 % de las compañías de más alto desempeño completaron los proyectos usando Agile y el 23 % con enfoques híbridos Agile-predictivo. Los enfoques predictivos son el 44 %.

En este caso si bien Agile no es el primero, por sí solo representa casi la cuarta parte en todas las industrias. Si se lo considera en conjunto con los enfoques híbridos, sobrepasa al enfoque predictivo puro.

Otra fuente muy importante de información, en este caso relacionada exclusivamente a desarrollo de software, es StackOverflow. Es un sitio de preguntas y respuestas para programadores profesionales o aficionados. En su encuesta anual de 2018 realizada a más de 100000 desarrolladores de todo el mundo, 85 % respondieron que utilizan Agile como metodología de desarrollo (StackOverflow, 2018).

Según los datos obtenidos por ForbesInsights y PMI (2017) en una encuesta realizada a más de 500 ejecutivos senior en todo el mundo que abarcan un variado tipo de industrias: tecnología, manufactura, banca, productos al consumidor y al por mayor, energía y gestión de activos e inversión, el 92 % cree que la agilidad organizacional o la habilidad para responder rápidamente a las condiciones del mercado y factores externos es crítica para alcanzar el éxito en los negocios.

En una investigación realizada por HewlettPackard (2017) en base a encuestas a más de 600 desarrolladores de software y profesionales de IT, la basta mayoría reportan que utilizan Métodos Ágiles: dos terceras partes consideran a su compañía como Agile pura o que tiende hacia Agile.

2.1.2.4 Principales retos en la adopción de Agile

Los tres principales retos o barreras en la adopción y escalamiento de prácticas Ágiles son, de acuerdo a una investigación realizada por Collabnet (2019), las siguientes:

- Cultura organizacional en desacuerdo con los principios ágiles.
- Organización en general resistente al cambio.
- Soporte de la administración y patrocinio inadecuado.

El estudio consistió en encuestas realizadas a más de 1300 profesionales de la comunidad de desarrolladores de software a nivel mundial.

En ese mismo estudio se remarca las principales respuestas de los consultados respecto a lo que creen más les ha ayudado en la adopción de Agile:

- Agile Coaches internos
- Patrocinio de los ejecutivos
- Programas de entrenamiento provistos por la compañía
- Prácticas consistentes en todos los equipos.

2.1.3 Pensamiento sistémico

De acuerdo a Ackoff (1999) un sistema es un todo formado por partes individuales, cada una de las cuales pueden afectar el comportamiento o propiedades del todo. Estas partes son inter-dependientes.

Las propiedades esenciales que definen a cualquier sistema son propiedades del todo, ninguna de sus partes individuales las tiene (Ackoff, 1999).

Ackoff (1999) refuerza lo anterior al añadir: cuando un sistema es dividido pierde sus propiedades esenciales; un sistema no es la suma de sus partes, es el producto de las interacciones entre ellas.

Respecto al desempeño de un sistema, Ackoff (1999) subraya que éste depende de cómo las partes se ajustan, no de cómo ellas funcionan separadamente.

En lo referente a la calidad, Ackoff (1999) anota que remover defectos es una actividad relevante, pero al deshacerse de algo que está mal no significa necesariamente que se obtenga lo que en realidad se busca, es decir, no es una forma de mejorar el desempeño de un sistema.

Para Ackoff (1999), esta mejora en el desempeño debe basarse en el concepto correcto de calidad, que debe contener la noción de valor, más allá de la simple eficiencia. El autor considera que la diferencia entre eficiencia y eficacia es la diferencia entre conocimiento y sabiduría.

2.1.3.1 Implicaciones para los programas de mejora

Los administradores deben considerar la naturaleza sistémica de sus organizaciones, para evitar que sus esfuerzos para mejorar su desempeño fracasen.

Antes de iniciar cualquier iniciativa de mejora en un sistema complejo, en este caso una organización, es necesario entender su propósito. Esto no se puede lograr si solo se pone atención a partes específicas.

De lo anterior se desprende que las estrategias que buscan implementar cambios u optimizaciones enfocadas en componentes individuales en una organización pueden tener poco o ningún efecto. Técnicas como dividir y conquistar están condenadas al fracaso si se trata de este tipo de sistemas. No se puede entender o predecir el comportamiento futuro de una parte si se la observa aislada del resto del sistema.

En el área de desarrollo de software a menudo existe una obsesión por la eficiencia: un bajo número de defectos, pago de deuda técnica, código limpio, cobertura de código. Esta obsesión sobrepasa el objetivo primordial, es decir la eficacia: entregar valor al cliente. Éste debe ser el objetivo de cualquier programa de mejora en organizaciones de desarrollo de software.

Según Ackoff (1999), los programas de mejora deben dirigirse hacia lo que se desea, no a lo que no se desea.

2.1.4 Herramientas para cada situación

Hasta este punto hemos visto que existen tipos de proyectos que son más apropiados para ejecutarse con métodos de peso pesado, como el modelo Waterfall. Estos proyectos se caracterizan en que el problema está bien definido desde el inicio, no van a existir grandes cambios durante el desarrollo del mismo porque existe un trabajo de ingeniería o científico riguroso para definir completamente los requerimientos y el plan a seguir. El trabajo de los desarrolladores es seguir fielmente el plan.

En contraste existen otros tipos de sistemas, complejos por su naturaleza debido a los diferentes puntos de vista del negocio y a su entorno cambiante, que son más apropiados para que se los afronte con Métodos Ágiles. Para poder afrontar problemas complejos se requiere de una mentalidad ágil en las organizaciones. Ejemplos de este tipo de proyec-

tos son los sistemas de información: financieros, ventas, gestión de clientes, gestión de contenidos, gestión de cada elemento de la cadena de valor de las empresas.

En el presente trabajo se realizará la valoración de la aplicación de Métodos Ágiles en un equipo de una empresa de desarrollo de software del mercado ecuatoriano. Para este fin se requiere contar con un modelo que provea el marco conceptual a partir del cual se desarrolla la evaluación. En la siguiente sección se revisará el modelo seleccionado para este propósito.

2.2 EL MODELO DE FLUIDEZ ÁGIL

El Modelo de Fluidez Ágil constituye una forma efectiva de evaluar las competencias requeridas por parte de los equipos de desarrollo de software en cuanto a agilidad (Larsen y Shore, 2019). Usa la metáfora de fluidez en el manejo de idiomas para explicar cómo debe ser el ejercicio de estas competencias, el cual debe ser fluido, natural. El nivel de fluidez se logra no solo con el conocimiento sino fundamentalmente a través de la práctica continua.

Este modelo de acuerdo a sus autores presenta un conjunto de 4 zonas de las cuales una organización puede escoger la o las que mejor se adapten a sus necesidades de acuerdo a sus objetivos. A diferencia de los modelos de madurez en los cuales cada etapa es un paso hacia la madurez y donde los niveles iniciales son considerados como inmaduros, este modelo afirma que cada zona es una opción madura en sí misma. Esto no descarta que los equipos de desarrollo de software, conforme mejoran sus competencias, puedan progresar a través de estas zonas (Larsen y Shore, 2019).

Cada zona se caracteriza por tener un conjunto de competencias. Las competencias de las zonas finales tienden a requerir más tiempo que las zonas iniciales. Estas competencias ágiles también se soportan mutuamente, es decir, para que un equipo sea considerado que cumple con las competencias de una zona, debe haber logrado las competencias de las zonas previas. Las zonas por las que un equipo de desarrollo de software transita mientras aprende son las siguientes (Larsen y Shore, 2019):

1. Equipos Enfocados. Para ubicarse en esta zona se requiere una cultura de equipo.
2. Equipos que Entregan. Esta zona requiere que los equipos mejoren sus habilidades

técnicas.

3. Equipos Optimizados. Se necesita cambios en la estructura organizacional de la empresa.
4. Equipos Fortalecidos. Requiere cambios en la cultura organizacional de la empresa.

2.2.1 Equipos Enfocados

La zona de Equipos Enfocados representa la aplicación de los fundamentos ágiles. Los equipos de esta zona trabajan de forma cohesionada con objetivos comunes y producen valor para el negocio. Esto en contraste con equipos que están apenas iniciando su viaje aplicando Métodos Ágiles, que tienden a pensar la mayor parte del tiempo únicamente en términos técnicos y que a menudo trabajan en base a contribuidores individuales que ejecutan tareas asignadas individualmente (Larsen y Shore, 2019).

A continuación se listan las competencias que se requieren en esta zona desde el punto de vista de las necesidades del negocio, trabajo efectivo como equipo y maestría del equipo (Larsen y Shore, 2019).

2.2.1.1 Necesidades del negocio

Para poder responder a las necesidades del negocio el equipo debe desarrollar las siguientes competencias:

- Clientes involucrados. El equipo trabaja con un representante del negocio que provee la perspectiva y expectativas.
- Fijación de prioridades. Los dueños del negocio pueden contar con que el equipo trabaje en las funcionalidades que dan más valor conforme a la visión del representante negocio.
- Software que funciona. El equipo planifica su trabajo y demuestra progreso en incrementos pequeños. El representante del negocio comprende y valora estos incrementos.

- Clientes empoderados. El representante del negocio puede cambiar la dirección del equipo al menos una vez al mes.
- Fijación de expectativas. La administración da soporte al equipo para que éste pueda trabajar a un ritmo que le permita responder a las necesidades del negocio indefinidamente.

2.2.1.2 Trabajo efectivo como equipo

Las siguientes competencias ayudan a que el equipo trabaje efectivamente como tal:

- Equipos auto-organizados. El equipo, basado en las necesidades del negocio, planifica y crea sus tareas.
- Trabajo en equipo. Los integrantes del equipo consideran y ejecutan su plan como un trabajo en equipo, no de forma individual.
- Responsabilidad de equipo. Los integrantes del equipo comparten la responsabilidad para cumplir con su plan.
- Soporte de la administración respecto a la cultura de equipo. La administración toma en cuenta que el plan se ejecuta como un trabajo en equipo y no como una asignación individual de responsabilidades.

En el libro «Time, Talent, Energy» sus autores, Mankins y Garton (2017), señalan en la investigación realizada para su firma consultora en liderazgo Bain & Company, que empresas de alto rendimiento como Apple y Google se enfocan en lo colectivo en lugar de lo individual.

Un ejemplo de esto es Apple y Microsoft a inicios de los 2000, de acuerdo a Mankins y Garton (2017). Apple desarrolló su sistema operativo iOS 10 con 600 ingenieros en menos de dos años. En contraste, a Microsoft le tomó más de cinco años desarrollar y finalmente retirar Vista, con 10000 ingenieros.

Mankins y Garton (2017) encontraron que en Apple los reconocimientos se hacían al desempeño del equipo; ninguna persona en el equipo recibe una evaluación de desempeño

excepcional a menos que el equipo entero lo haya recibido. A diferencia de Apple, en Microsoft se usaba una clasificación en niveles en la que el 20 % de los integrantes de cada equipo recibe una revisión excepcional y una compensación basada enteramente en el desempeño individual. Microsoft eventualmente eliminó este tipo de clasificaciones.

2.2.1.3 Maestría del equipo

El equipo construye su maestría si desarrolla las siguientes competencias.

- Mejora de trabajo en equipo. El equipo acoge y mejora continuamente su enfoque de trabajo en conjunto.
- Mejora de relaciones en el equipo. El equipo es consciente de cómo las relaciones entre sus integrantes afecta su capacidad para tener éxito y proactivamente intenta mejorarlas.
- Mejora de entorno físico. El equipo es consciente de cómo su entorno físico de trabajo afecta su capacidad para tener éxito y proactivamente intenta mejorarlo.

2.2.2 Equipos que Entregan

De acuerdo a Larsen y Shore (2019), los equipos de esta zona, a más de enfocarse en el valor del negocio, son capaces de entregar o poner en producción su trabajo con costos bajos y riesgos mínimos cuando el negocio lo requiera. Representa la sustentabilidad ágil.

A continuación se explican las competencias requeridas para este zona, de acuerdo a los mismos autores, considerando el punto de vista del negocio, del equipo y la excelencia técnica.

2.2.2.1 Necesidades del negocio

Desde el punto de vista de la respuesta a las necesidades del negocio los equipos de esta zona tienen las siguientes competencias:

- Integración continua. El equipo desarrolla código cuya calidad tiene el nivel de ambiente de producción y todo su último trabajo se entrega a un ambiente equivalente a producción al menos diariamente.

- Disponibilidad para liberar a producción. El representante del negocio puede liberar a producción el último trabajo del equipo cuando lo requiera.
- Estimaciones de entrega realistas. El equipo provee rangos estimados útiles de entrega al representante del negocio cuando lo requiera.
- Soporte de la administración. El equipo coordina con las partes interesadas del negocio para desarrollar su código y otros artefactos de una manera que le permita que el código sea mantenible de una forma indefinida y de bajo costo. La administración necesita asegurar con las partes interesadas que se establezca una expectativa respecto a que el trabajo reflexivo se valora sobre la conveniencia.

2.2.2.2 Trabajo efectivo como equipo

Desde el punto de vista de trabajo efectivo como equipo:

- Propiedad y responsabilidad colectiva. Los programadores consideran al código y artefactos similares que pertenecen al equipo, no a individuos, y comparten responsabilidad de cambiarlo y mejorarlo.
- Equipos auto-suficientes y multifuncionales. Todas las habilidades necesarias para que el equipo pueda diseñar, desarrollar, entregar, monitorear, mantener, etc., son accesibles inmediatamente. No se refiere únicamente a cursos de entrenamiento, sino a entrenadores con estas habilidades que se involucren en sus proyectos.

2.2.2.3 Excelencia técnica

Finalmente las competencias para alcanzar la excelencia técnica:

- Mejora continua, simplicidad. Los miembros del equipo cuando trabajan con el código o artefactos similares deben mejorar su calidad al menos ligeramente.
- Entrega continua, builds automáticos. Las entregas a producción están automatizadas y no toman más de 10 minutos de esfuerzo manual.
- Cambios incrementales, trabajo de calidad, pruebas automáticas. El equipo produce código a nivel de producción sin requerir una fase de pruebas manuales.

- **Mentalidad de aprendizaje continuo.** Todos los miembros del equipo son conscientes de cómo sus destrezas técnicas y profesionales afectan su habilidad para cumplir los objetivos del equipo y disminuir los costos de mantenimiento. Ellos proactivamente buscan mejorar estas destrezas.

2.2.3 Equipos Optimizados

Los Equipos Optimizados entienden qué es lo que quieren sus mercados, qué necesita su negocio y cómo satisfacer esas necesidades. A diferencia de los Equipos que Entregan, no solo tienen la habilidad para entregar al mercado, también conocen qué entregar al mercado. Los Equipos Optimizados toman sus propias decisiones acerca de qué financiar y dónde enfocar sus esfuerzos. Requiere cambios organizacionales para permitir que los expertos del negocio y otros roles involucrados, contribuyan en el equipo a tiempo completo (Larsen y Shore, 2019).

Las competencias que deben demostrar los equipos en esta zona son las que se indican a continuación, clasificadas por la respuesta a las necesidades del negocio, trabajo en equipo, perseguir la grandeza del producto (Larsen y Shore, 2019).

2.2.3.1 Necesidades del negocio

En primer lugar, en función de las necesidades del negocio:

- **Retroalimentación cuantitativa del proceso en términos del negocio.** El equipo describe sus planes y progreso en términos de los resultados de las métricas del negocio identificadas en conjunto con la administración.
- **Retroalimentación del mercado.** El equipo colabora con las partes interesadas internas y externas a la empresa, para determinar cuándo y si los pronósticos de liberación a producción tienen el mejor retorno sobre la inversión.

2.2.3.2 Trabajo efectivo como equipo

Las competencias que debe tener el equipo para que trabaje de forma autónoma y confiable son:

- Responsabilidades del negocio. El equipo coordina con la administración para entender y refinar su rol para alcanzar la estrategia del negocio.
- Empoderamiento sobre la dirección del producto. El equipo asume la responsabilidad para alcanzar los resultados del negocio identificados con la administración.
- Delegación de parte de la administración. La administración provee al equipo de los recursos y la autoridad que ellos necesitan para que, de forma autónoma, pueda lograr los resultados del negocio.
- Soporte de la administración respecto a la incorporación de expertos del negocio al equipo. La administración asegura que todas las destrezas que el equipo necesita para entender su mercado y lograr los resultados del negocio están presentes en miembros a tiempo completo del equipo.

Parte de la optimización es una adecuada distribución entre empleados y administradores. En el estudio realizado por Hamel y Zanini (2016) se afirma que el arrastre organizacional debido al exceso de administración cuesta USD 3 trillones por año en EE.UU. Analizan casos de empresas que sugieren que es posible hacer funcionar negocios complejos con menos de la mitad de la carga administrativa que típicamente se encuentra en las empresas.

Según Hamel y Zanini (2016) es posible doblar la relación empleados administradores de 4.7:1 a 10:1 para liberar individuos a otro trabajo más productivo y creativo.

2.2.3.3 Excelencia de producto

Y por último, para que se consiga la distinción, relevancia, grandeza del producto, el equipo debe demostrar las siguientes competencias:

- Análisis del mercado. El equipo se involucra con sus clientes y mercados para comprender las necesidades y oportunidades del producto.
- Aplicación del método científico. El equipo crea hipótesis acerca de las oportunidades del negocio y conduce experimentos para probarlas.

- Lean Software Development. El equipo planifica y desarrolla su trabajo en una forma que le permite cambiar en gran medida sus planes, eliminando el desperdicio, dado un anuncio de corto-mediano plazo (menos de un mes).

2.2.4 Equipos Fortalecidos

Los Equipos Fortalecidos además de tener la habilidad para comprender y satisfacer las necesidades de sus mercados, también juegan un rol más grande en su organización (Larsen y Shore, 2019).

De acuerdo a los autores del Modelo de Fluidez Ágil, Larsen y Shore (2019), los Equipos Fortalecidos contribuyen en sus organizaciones de tres formas. Primero, comprenden cómo su propósito se alinea con el de otros equipos para cumplir la estrategia más grande de la empresa. Trabajan activamente para que la estrategia sea más exitosa.

Segundo, ellos propagan su know-how en la organización. Buscan oportunidades para contribuir con sus habilidades en otros equipos y aprender de otros equipos.

Tercero, la organización delega decisiones organizacionales a los equipos. Los líderes diseñan estructuras para extraer los aspectos más importantes de las ideas colectivas y para incorporarlas dentro de las mejoras organizacionales.

2.2.5 Zonas según los requerimientos de cada organización

La zona apropiada depende de la organización. La zona de Equipos que Entregan o Equipos Optimizados son a menudo las mejores opciones, pero Equipos Enfocados y Fortalecidos también pueden ser buenas alternativas (Larsen y Shore, 2019).

Aunque la zona Enfocados no incluye las prácticas tecnológicas de sostenibilidad, es decir mantener un ritmo de aplicación de mejoras de forma indefinida, constituyen una buena manera de dar valor al negocio, demostrar visibilidad del trabajo realizado y un punto de partida para futuras inversiones en conocimiento. Es una buena opción para empresas que no necesitan realizar constantes modificaciones en su software en el largo plazo (Larsen y Shore, 2019).

En contraste, para aquellas organizaciones que sí necesitan modificar y mejorar su software frecuentemente en el corto-mediano plazo, la zona Equipos que Entregan es una

mejor opción. Estos equipos tienen la habilidad para responder a las solicitudes del negocio (Larsen y Shore, 2019).

Las organizaciones que buscan los beneficios de la agilidad del negocio innovador deben enfocarse en la zona de Equipos Optimizados. Provee grandes beneficios, pero también requiere de cambios disruptivos en la estructura organizacional. Esto normalmente es más fácil de hacer en organizaciones pequeñas (Larsen y Shore, 2019).

Finalmente, la zona Fortalecidos representa el probable futuro Ágil. Es particularmente ideal para organizaciones de tamaño pequeño a mediano. Requiere investigación en las teorías de administración de vanguardia e inventar nuevas formas de trabajar (Larsen y Shore, 2019).

En el mercado ecuatoriano probablemente sea factible encontrar empresas en la zona de Equipos Enfocados y Equipos que Entregan. Es menos probable el caso de Equipos Optimizados y Equipos Fortalecidos.

Hasta este punto hemos descrito las competencias requeridas en cada zona del Modelo de Fluidez Ágil. En el siguiente apartado se presenta la forma de valorar estas competencias.

2.3 MÉTRICAS ESENCIALES

En este apartado se van a describir las métricas esenciales para cada zona del Modelo de Fluidez Ágil (Larsen y Shore, 2019).

Se ha identificado que para los Equipos Enfocados las métricas base se relacionan con la priorización de tareas y demostración de progreso en esas tareas.

Los Equipos que Entregan requieren métricas que provean información relacionada a un trabajo más reflexivo de parte de los equipos y estimaciones útiles. Se busca determinar si dan mayor valor a la calidad que a la conveniencia de terminar a tiempo funcionalidades que luego no son aceptadas por el cliente. Una mayor calidad en el trabajo también influye en estimaciones más acertadas.

Para el caso de los equipos Optimizados se necesitan métricas que provean información acerca de la capacidad del equipo para dirigir la evolución de un producto o cambiar a otro producto que dé más valor al negocio.

Por último los Equipos Fortalecidos buscan mejorar la organización. Las métricas deben proveer información sobre esta capacidad.

2.3.1 Prioridades y progreso incremental

Las métricas de los Equipos Enfocados se relacionan con el priorización y el progreso incremental del trabajo. Estos equipos trabajan en las prioridades que le son dadas desde la perspectiva de valor para el negocio y realizan progresos incrementales sobre las prioridades del negocio a intervalos regulares, por ejemplo cada mes (Larsen y Shore, 2019).

La tabla 4.5 en la página 49, sección 4.1.3 Diseño de Instrumentos - Índices, contiene las métricas para determinar si las tareas están siendo priorizadas y si el equipo está progresando en el cumplimiento de las mismas: funcionalidades priorizadas, funcionalidades finalizadas, velocidad, velocidad promedio, esfuerzo pendiente, esfuerzo pendiente porcentual. La frecuencia de medición es por iteración.

La métrica Velocidad se mide en «puntos». El sistema basado en puntos para estimar y hacer seguimiento de avance de tareas es una técnica muy utilizada en Metodologías Ágiles. Lo que busca es capturar el tamaño relativo de una tarea respecto a otras mediante la asignación de un número de puntos. El equipo de desarrollo se pone de acuerdo en la forma en que va a estimar las tareas, por ejemplo las tareas grandes equivalen a 5 puntos, las medianas a 3 puntos y las pequeñas a 1 punto. La ventaja de esta técnica es que mide el tamaño de una tarea independientemente del tiempo que tomaría completarla.

El esfuerzo pendiente se refiere a la cantidad de puntos pendientes para finalizar el proyecto, correspondientes a la sumatoria de puntos de todas las funcionalidades que restan en el proyecto.

La fuente principal de datos para estas métricas es el Sistema de Gestión de Proyectos.

2.3.2 Trabajo reflexivo y estimaciones útiles

El trabajo más reflexivo y la generación de estimaciones más acertadas, son la métricas base de los Equipos que Entregan. Estos equipos pueden liberar su trabajo más reciente, con costos y riesgos mínimos, cada vez que el negocio lo requiera a la cadencia del mercado (Larsen y Shore, 2019).

La tabla 4.6 en la página 50 lista las métricas principales: puntos estimados y puntos completados tanto para nuevas funcionalidades como para corrección de errores y pago de deuda técnica (mejorar el código para que sea más limpio y mantenible), y lead time.

Estas métricas permiten revelar falencias sistémicas en el ciclo de producción. Un indicador útil para este efecto es lead time o tiempo de espera hasta que una funcionalidad esté lista. El lead time puede dividirse en: tiempo de espera para iniciar el desarrollo, tiempo de desarrollo, tiempo de espera para iniciar pruebas, tiempo de pruebas y tiempo de espera para ser usado por el cliente.

Los Equipos que Entregan deberían tener tasas de defectos bajas. La mayor cantidad de tiempo debería invertirse en mejoras, nuevas funcionalidades, en contraste con el tiempo para corregir errores. Si el código tiene baja deuda técnica los cambios van a ser menos costosos en función del tiempo de entrega.

Fuente de datos típicas para obtener esta información son el Sistemas de Gestión de Proyectos y el Sistema de Integración Continua.

La tabla 4.7 de la página 51 contiene las métricas principales de calidad de código: índice de mantenibilidad, complejidad ciclomática, acoplamiento y niveles de herencia.

Para cada uno de los índices de calidad existen niveles de evaluación (Pattigulla, 2017). El índice de mantenibilidad tiene los rangos indicados en la tabla 4.8. Los niveles de evaluación del índice de complejidad ciclomática son los que se listan en la tabla 4.9. El índice de acoplamiento tiene los niveles para clasificar el riesgo indicados en la tabla 4.10. Por último para el índice de niveles de herencia se tienen los niveles de la tabla 4.11.

Métricas adicionales relacionadas con la calidad del producto desde el punto de vista del Sistema de Integración Continua se encuentran en la tabla 4.12 de la página 53. Entre ellas: builds exitosos y fallidos, pruebas automáticas exitosas y fallidas, cobertura de código.

Build es la acción de compilación del código fuente. Si el código no tiene errores es un build exitoso. Cobertura de código es un índice que mide la cantidad de código que está cubierto por pruebas automáticas.

La tabla 4.13 de la página 54 contiene métricas que se pueden obtener del Sistema de Despliegue a Producción. La frecuencia de medición de estas métricas es mensual. Los

índices son despliegues exitosos y fallidos.

Despliegue es la acción de aplicar los cambios en el ambiente de producción, es decir el que sirve al usuario final.

2.3.3 Dirección del producto

Los Equipos Optimizados describen el lugar donde se posicionan sus productos en el mercado y cómo mejorarán su posición.

El equipo entrega productos que reúnen los objetivos del negocio y las necesidades del mercado. También tiene la capacidad de coordinar de forma temprana con sus líderes para cancelar proyectos de bajo valor.

Para lograr esto el equipo aprende del feedback del mercado para anticipar necesidades del cliente y crear nuevas oportunidades de negocio.

La tabla 4.14 en la página 55 contiene las métricas principales referentes a la dirección del producto: aprobado o rechazado por el usuario. Si el resultado de estas métricas es positivo significa que los cambios que lleva a cabo el equipo son acertados, caso contrario, el equipo debe pivotar basado en esta información.

Otras métricas útiles que permiten conocer objetivamente si los cambios que realiza el equipo proveen valor al usuario se relacionan con el análisis de cohorte o de grupos, como lo explica (Ries, 2011) en su libro *The Lean Startup*. En lugar de usar totales acumulados o número brutos como el número total de clientes o ingresos totales, se busca el desempeño de cada grupo de clientes que interactúa con el producto independientemente. Cada grupo puede representar una etapa en la secuencia del comportamiento de los usuarios del producto (flujo de usuarios), por ejemplo, usuario invitado, usuario de prueba, usuario registrado y que paga por el servicio. La tabla 4.15 en la página 56 tiene una muestra de este tipo de métricas: usuarios por tipo, tasa de conversión de usuarios, uso del producto.

Fuente comunes de datos son: Sistema de Gestión de proyectos, Sistema de Monitoreo de la aplicación, Logs del Sistema/Aplicación.

Otra referencia de utilidad son los Cuadrados Mágicos de Gartner (Gartner, 2019). Posicionan empresas y sus productos en 4 cuadrantes: jugadores de nicho, visionarios, retadores, líderes. Si la empresa figura en uno de estos cuadrantes, puede fijarse la meta

de mantenerse y/o migrar al siguiente nivel.

2.3.4 Mejora organizacional

En la zona de los Equipo Fortalecidos, se busca medir su capacidad para contribuir en la mejora organizacional.

La mejora organizacional se puede medir mediante la capacidad de describir el trabajo en el entorno organizacional, el involucramiento en la solución de problemas organizacionales, la ejecución de actividades multi-equipos, la colaboración entre equipos y las polinizaciones cruzadas.

Con esto culmina el capítulo teórico. Se han expuesto los métodos de desarrollo de software y entre ellos la importancia de los Métodos Ágiles para entornos cambiantes y complejos. Luego se revisó los fundamentos expuestos en el Manifiesto Ágil y el Modelo de Fluidez Ágil que sirve de marco de trabajo para realizar el diagnóstico de competencias de agilidad. También se resaltó la importancia del pensamiento sistémico para comprender las implicaciones de la adopción de Metodologías Ágiles en una organización. El capítulo finalizó con las métricas esenciales de acuerdo a las zonas del Modelo de Fluidez Ágil.

El siguiente capítulo tratará sobre el objeto de estudio, una empresa seleccionada en el mercado ecuatoriano y las razones que tiene para ejecutar un diagnóstico de Fluidez Ágil.

CAPÍTULO III

OBJETO DE ESTUDIO

Por razones de confidencialidad expuestas por la empresa objeto de estudio, la información de este capítulo no necesariamente representa la situación actual de la empresa, cuyo nombre, contexto organizacional y otros datos del proyecto que se van a estudiar, han sido modificados. La información recolectada por entrevistas, observación y métricas obtenidas mediante herramientas no han sido alteradas.

3.1 CONTEXTO ORGANIZACIONAL

La empresa Hyna es una empresa proveedora de software especializada en Sistemas de Gestión de Contenido CMS (Content Management System). Tiene dos filiales, una en Ecuador, y la otra en Estados Unidos. La empresa tiene 15 años de historia con varios altibajos. El número total de empleados entre las dos filiales es de 40.

La filial en Ecuador es la encargada de ejecutar los proyectos. Para esto mantiene equipos de desarrolladores, aseguradores de calidad, administradores de proyectos y personal administrativo de soporte. Los roles técnicos están cubiertos por 30 empleados, mientras que 10 son administradores y directores. La relación empleados productivos versus administradores es 3:1.

La filial en Estados Unidos gestiona los clientes, además mantiene equipos de arquitectos, analistas de negocio y vendedores.

Maneja una cartera de clientes estadounidenses, entre estos: universidades, asociaciones de profesores, inmobiliarias. Unos clientes son más activos que otros, es decir buscan realizar actualizaciones y mejoras en el producto. Entre los más activos, que a la vez generan más proyectos y por lo tanto ingresos para la empresa, se encuentran los del sector de

las universidades. Éstos presionan por más funcionalidades y menores tiempos de entrega. La empresa Hyna busca satisfacer sus necesidades para mantener una relación a largo plazo.

La competencia por mantener una buena cartera de clientes es alta. Existen empresas especializadas en CMS tanto en el mercado ecuatoriano como en EE.UU. que se encuentran bien posicionadas. Su principal competidor es la empresa Vn, cuyos fundadores en el pasado formaron parte de Hyna y compartieron la cartera de clientes. Recientemente varios de los clientes de Hyna se pasaron a Vn debido a que está generando mayor valor para sus clientes desde el inicio mismo de los proyectos y con tiempos de entrega más consistentes.

3.1.1 Desempeño económico fiscal del sector

De acuerdo a la ficha metodológica de operaciones estadísticas, el objetivo del Índice de Nivel de Actividad Registrada es «medir el desempeño económico fiscal de los sectores productivos de la economía nacional a través del tiempo, mediante el comportamiento de las ventas corrientes registradas por los contribuyentes especiales del Impuesto al Valor Agregado (IVA) para un mes calendario, comparadas con las del mismo mes pero del período base (año 2002=100)» (INEC, 2019).

Conforme a este índice, el comportamiento del sector K72 correspondiente a Informática y actividades conexas es inferior al índice general, que representa a todos los sectores del país, durante 2018 y 2017, y ligeramente superior en 2019. Ver figuras 3.1, 3.2, 3.3 de la página 29 a la página 30.

Esto refleja la situación difícil en la que se encuentra el sector que corresponde a la empresa objeto de estudio respecto a la economía en general, que de por sí tampoco es buena.

3.2 LA ESTRATEGIA

La empresa ha decidido enfocarse en un solo sistema de administración de contenidos: Sitecore. Buscan ser expertos en esta herramienta usada por las empresas más grandes a nivel mundial.

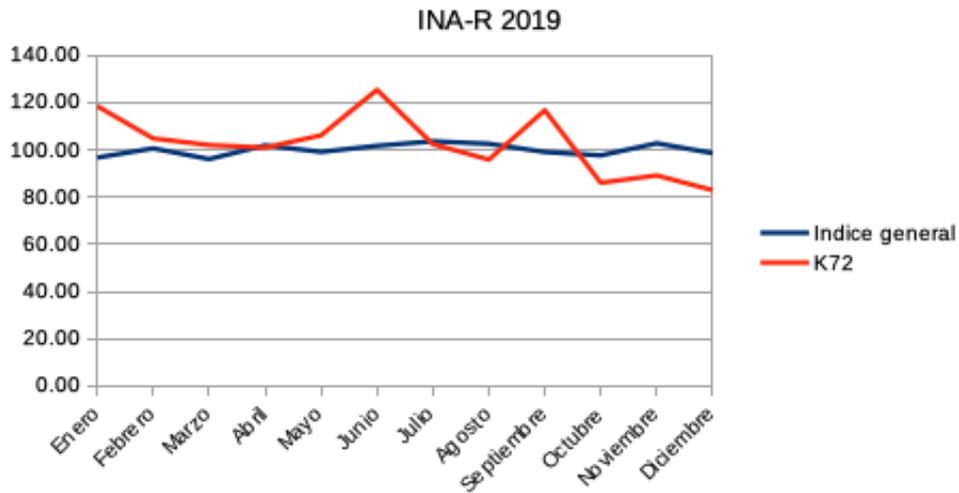


Figura 3.1: Índice de nivel de actividad registrada 2019

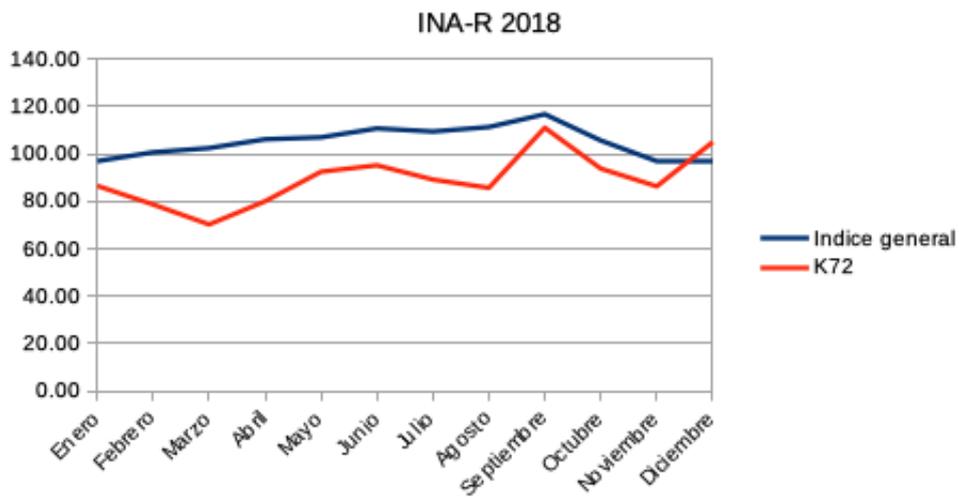


Figura 3.2: Índice de nivel de actividad registrada 2018

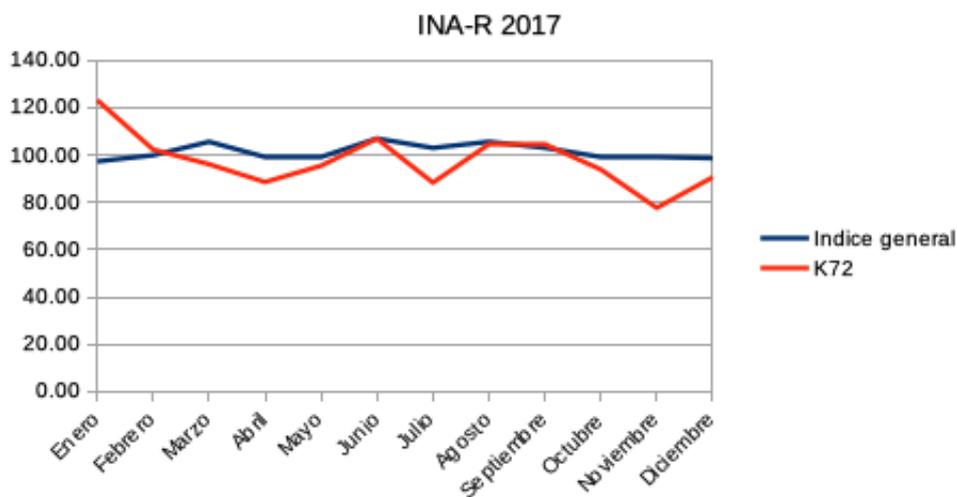


Figura 3.3: Índice de nivel de actividad registrada 2017

En este sentido ha promovido que sus desarrolladores y otros empleados de la empresa se capaciten en esta herramienta. Para ello cuenta con el soporte de Sitecore, de cual es un asociado certificado.

Sitecore es reconocido como un líder de su sector de acuerdo Gartner Research (Tamturk, 2018). Ver figura 3.4 de la página 31.

La nueva generación de CMSs es conocida como la gestión de experiencias, de la que Sitecore es el líder global. Sitecore Experience Platform es la herramienta que combina la administración de contenidos con información de clientes, analítica, marketing automatizado (Sitecore, 2020b).

Entre las empresas líderes que utilizan esta plataforma se encuentran las siguientes: easyJet, L'Oréal, Honda, BMI, Formula E, Volvo (Sitecore, 2020a).

Sitecore tiene una cuota de mercado de alrededor del 0.3 %; a diferencia de Sitecore, el líder en la cuota de mercado es WordPress con el 65.72 % (Enlyft, 2020) (ver figura 3.5 de la página 31).

El enfoque de Hyna es el nicho de clientes que usan Sitecore.

Buscan este nicho porque, de acuerdo a su gerente general, al ser unos clientes exclusivos son los que pueden generar mayores ganancias. No buscan la cantidad de clientes que usan por ejemplo WordPress porque el ingreso por cada cliente individual sería marginal.



Figura 3.4: Cuadrante mágico de Gartner 2018 para Plataformas de Experiencia Digital



Figura 3.5: Sitecore Market Share

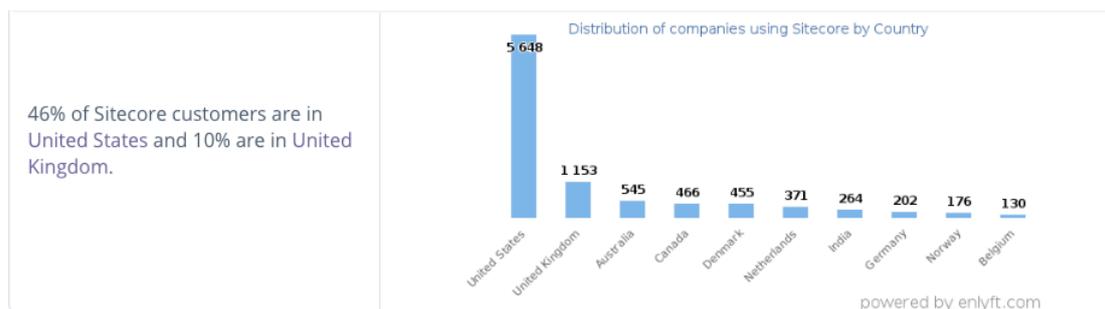


Figura 3.6: Principales países que usan Sitecore

La barrera de entrada a este selecto grupo de clientes que pueden generar mejores ingresos es el costo de sus licencias, mientras que WordPress es software libre.

De acuerdo a este mismo estudio (Enlyft, 2020), la mayor cantidad de clientes que usan Sitecore se encuentran en EE.UU con 5648 empresas, seguido de Reino Unido con 1153. Ver figura 3.6 en la página 32.

Por lo tanto la estrategia se complementa con una oficina y ejecutivos que conocen el entorno de estas empresas en EE.UU., el principal país por el número de empresas que usan Sitecore.

3.3 LA NUEVA METODOLOGÍA

La empresa se encuentra empeñada en modificar su forma de trabajo. Hasta inicios del año anterior venían trabajando de una forma que se asemejaba mucho a una metodología en Cascada.

Existía una etapa inicial en los proyectos donde se definían los requerimientos para el proyecto. Participaban un analista de negocio y un arquitecto de la empresa y un representante del cliente. En base a estos requerimientos el arquitecto y el analista del negocio trabajan en la elaboración de los documentos de diseño.

Una vez elaborados los documentos de diseño se armaba un equipo de desarrollo. En este punto el equipo se enteraba en qué consistía el proyecto en base a la documentación técnica y una breve presentación de parte del analista de negocio. Con estos insumos iniciaba su trabajo el equipo de desarrollo y se esperaba que cumpla con el objetivo en el tiempo estimado por el administrador del proyecto, asesorado por el arquitecto y el ana-

lista del negocio.

Esta metodología estaba provocando problemas dentro del equipo para cumplir con la fecha de entrega y el alcance comprometidos. No era extraño que el cliente solicitara cambios en las funcionalidades sobre las que ya se estaba trabajando, o nuevos requerimientos, lo que implicaba re-trabajos tanto en los documentos de requerimientos, diseños, como en el código fuente mismo.

En los proyectos se cumplía una sola prescripción de la metodología ágil Scrum: el scrum diario. Consistía en una reunión diaria de corta duración en la que cada miembro del equipo debía reportar su trabajo del día anterior y su planificación para el día actual al administrador del proyecto.

El rol de Scrum Master tampoco estaba siendo aplicado y la forma de trabajo no era Scrum. La única reunión de Scrum que se realizaba, el scrum diario, tampoco estaba enfocada correctamente. De acuerdo a Scrum, la reunión diaria es una herramienta que ayuda a que el equipo sincronice su trabajo entre sus miembros, no se supone que sea una reunión de reporte a un superior.

Debido a estos problemas, ahora la empresa está implementando un cambio en su forma de trabajo hacia el uso de Metodologías Ágiles de forma progresiva a partir de inicios del año 2018. Se ha contratado a un experto en Metodologías Ágiles, que ha creado una adaptación de Scrum específica para la empresa. Se han iniciado capacitaciones y se han ido aplicando cambios progresivamente en varios equipos de la empresa. Se ha hecho público en reuniones que ahora el nivel de adopción de la nueva metodología, bautizada como Wave, es prácticamente completo y que se deberían ver cambios en los nuevos proyectos.

3.3.1 Wave

Wave es una metodología de desarrollo de software creada internamente en Hyna y está basada en gran medida en Scrum (Schwaber y Sutherland, 2016). De acuerdo a Wave, un proyecto de desarrollo de software tiene los siguientes roles: administrador de proyectos, analista del negocio, desarrolladores, arquitecto, testers.

Se deben cumplir los siguientes eventos o reuniones:

- **Sprint o iteración.** Es el corazón de la metodología. Está definido en 2 semanas laborales, en las que se espera que se cree un incremento del producto. Contiene la planificación del sprint, los scrum diarios, el trabajo de desarrollo y la retrospectiva del sprint.
- **Planificación del sprint.** Se realiza al inicio de cada sprint. El objetivo es planificar el trabajo a ser ejecutado en el sprint. Dura de 4 horas a 1 día laboral.
- **Scrum diario.** Es una reunión de 15 minutos en la que cada miembro del equipo de desarrollo responde estas preguntas: qué hice ayer, que haré hoy y qué impedimentos tengo para cumplir con mi trabajo.
- **Retrospectiva de sprint.** Es una reunión que se lleva a cabo al finalizar el sprint. El objetivo es inspeccionar el incremento del producto y crear un plan de mejoras a ser ejecutadas durante el siguiente sprint. Dura de 4 horas a 1 día laboral.

Existen los siguiente artefactos:

- **Lista de historias maestra.** Es una lista ordenada de todas las funcionalidades que pueda necesitar el producto. Es la única fuente de requerimientos. Es responsabilidad del analista del negocio.
- **Historias del sprint.** Es un conjunto de historias de la lista de historias maestra que se han seleccionado para el sprint. Es una estimación del trabajo que el equipo puede completar en el sprint. La realiza el equipo del trabajo .

En términos generales, lo descrito en esta sección, constituye Wave.

3.4 EL PROYECTO

La empresa Hyna se encuentra trabajando en el tercer mes de un proyecto de cuatro meses con una universidad en Estados Unidos. El proyecto, nombrado como Partner Portal, consiste en la creación de un portal dinámico para socios del cliente.

El cliente ha amenazado con terminar el contrato debido a una serie de problemas. Esto significaría despedirse de un cliente clave que ha sido considerado importante para las operaciones de la empresa.

Lo que inicialmente se había hecho bien con un piloto entregado a tiempo y un cliente satisfecho, se convirtió en re-trabajos, trabajo fuera de horas de oficina y deuda técnica. Debido al desorden provocado por la presión, no se conoce con seguridad en qué está trabajando el equipo y la administración tampoco está ayudando.

Después de varios ajustes en el alcance del proyecto, el producto final a ser liberado consiste en el mínimo básico que el cliente necesita. No queda nada más para recortar y aun así parece que el equipo no va a terminar en el tiempo restante.

Siempre existe algo positivo, y es que los miembros del equipo de desarrollo son muy comprometidos y el administrador del proyecto tiene actitud positiva. El equipo tiene un fuerte respaldo de la administración y mantiene conversaciones semanales con el cliente. El diálogo no siempre es placentero pero al menos ambas partes conversan regularmente.

3.4.1 Revisión de forma de trabajo

Al revisar la forma en que se está trabajando, se encontró que dos equipos de desarrollo están involucrados en el proyecto (figura 3.7). La parte más importante del trabajo, que corresponde a la base del producto, estaba siendo implementada por el equipo de la empresa Hyna en Ecuador formado por cuatro desarrolladores y un tester, con el respaldo de un arquitecto y un analista de negocio (que también tiene el rol de administrador del proyecto) en Estados Unidos, mientras que el cliente hacía sus adaptaciones con su equipo en Estados Unidos. Los dos administradores de proyectos se reúnen una vez a la semana para revisar las prioridades y el progreso.

El administrador del proyecto y el tester son Scrum Master certificados. Este rol tiene la responsabilidad de asegurar que Scrum es entendido y que todas sus prácticas y reglas son seguidas.

La tabla número 3.1 contiene las herramientas usadas en el proyecto Partner Portal que son de interés para el presente trabajo.

El Sistema de Gestión de Proyectos es donde se definen y asignan las tareas y errores,

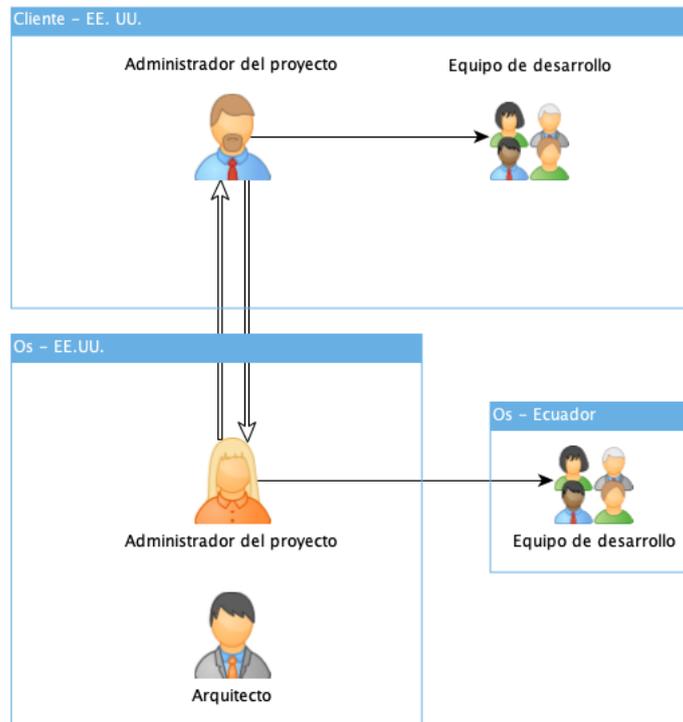


Figura 3.7: Dos equipos de desarrollo

Tabla 3.1: Herramientas del proyecto

Sistema de Gestión de Proyectos	Atlassian Jira, Atlassian Confluence
Sistema de Integración/Entrega Continua	Jenkins
Sistema de Despliegue a Producción	Octopus Deploy
Sistema de Monitoreo de la Aplicación	Google Analytics

se asocian tiempos, estimados y trabajo real.

El Sistema de Integración/Entrega Continua es donde se ejecutan los builds del código fuente, se ejecutan las pruebas automáticas y se organizan todos los artefactos para ser desplegados a ambientes de integración y pruebas.

Cuando todos los cambios se hayan validado y aprobado están listos para ser desplegados al ambiente de producción para que estén disponibles para que puedan ser utilizados por el usuario final. El Sistema Despliegue a Producción es el encargado de esta tarea.

Una vez que el sistema está en producción el Sistema de Monitoreo de la Aplicación muestra qué tan bien la aplicación está sirviendo al usuario final.

3.4.2 Iteraciones

El proyecto ha finalizado la sexta iteración. Cada iteración tiene una duración de 2 semanas.

La primera iteración, que se puede ver en la figura 3.8a (página 39), consistió en 4 funcionalidades, las 4 se completaron con éxito.

Para la segunda iteración se planificaron 4 funcionalidades más. El equipo terminó 3 de ellas (figura 3.8b).

En la tercera iteración (figura 3.8c) se tuvo que planificar la funcionalidad que no se completó en la iteración anterior, y adicionalmente se planificaron 4 funcionalidades más. El equipo pudo completar 3 funcionalidades. Así como ocurrió en la iteración anterior, las tareas que no se pudieron completar deben ser consideradas en la siguiente iteración.

La primera etiqueta dentro de la funcionalidad indica que la tarea se completó en la iteración, la segunda etiqueta significa que la tarea se heredó de la iteración anterior. Similar comportamiento se puede ver en las iteraciones 4 y 5.

Para la sexta iteración (figura 3.8f) se han completado 18 funcionalidades. El número total de funcionalidades que se han levantado hasta el momento es de 30, lo que constituye el alcance del proyecto. El número de funcionalidades que deben completarse en las dos últimas iteraciones del proyecto son 12.

Si bien cada funcionalidad tiene diferente complejidad, en términos generales se com-
pensan. Si a la sexta iteración se han completado 18 funcionalidades, a razón de 3 por

iteración, resulta complicado pensar que en las restantes 2 iteraciones se puedan completar 12 funcionalidades.

3.4.3 Burn-down

Una herramienta típica para medir el avance de proyectos de desarrollo de software que usan Metodologías Ágiles es el gráfico conocido como burn-down. Este gráfico muestra qué tan rápido el equipo está completando las funcionalidades del cliente y cuándo se espera terminar. El eje Y corresponde a la cantidad de trabajo pendiente (esfuerzo en puntos) y el eje X es el tiempo en iteraciones. Simplemente se registra la cantidad de trabajo restante en cada iteración y se lo traza en el gráfico. La velocidad del equipo es la cantidad promedio de trabajo completado en cada iteración.

Los puntos son una medida arbitraria que se utiliza para estimar el esfuerzo requerido para completar tareas, separándolas de la noción del tiempo. Cada iteración dura dos semanas.

La figura 3.9 en la página 40 demuestra un patrón preocupante. Al tercer mes del proyecto, al finalizar la sexta iteración (I6), el trabajo pendiente es 70 puntos, mientras que la estimación indicaba 30 puntos. Si sigue la misma tendencia (figura 3.10), en la última iteración, aun estarán pendientes entre 50 y 60 puntos. Este tipo de gráfico debería ser suficiente para provocar una reacción.

La línea roja corresponde al trabajo pendiente real, mientras que la línea azul es la estimación original.



(a) Iteración 1



(b) Iteración 2



(c) Iteración 3



(d) Iteración 4



(e) Iteración 5



(f) Iteración 6

Figura 3.8: Iteraciones

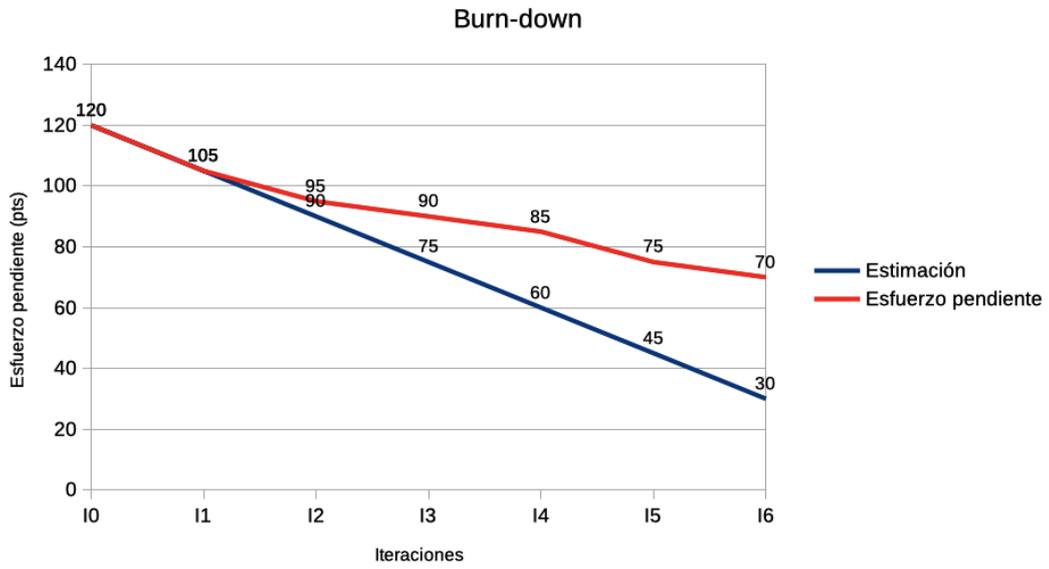


Figura 3.9: Burn-down

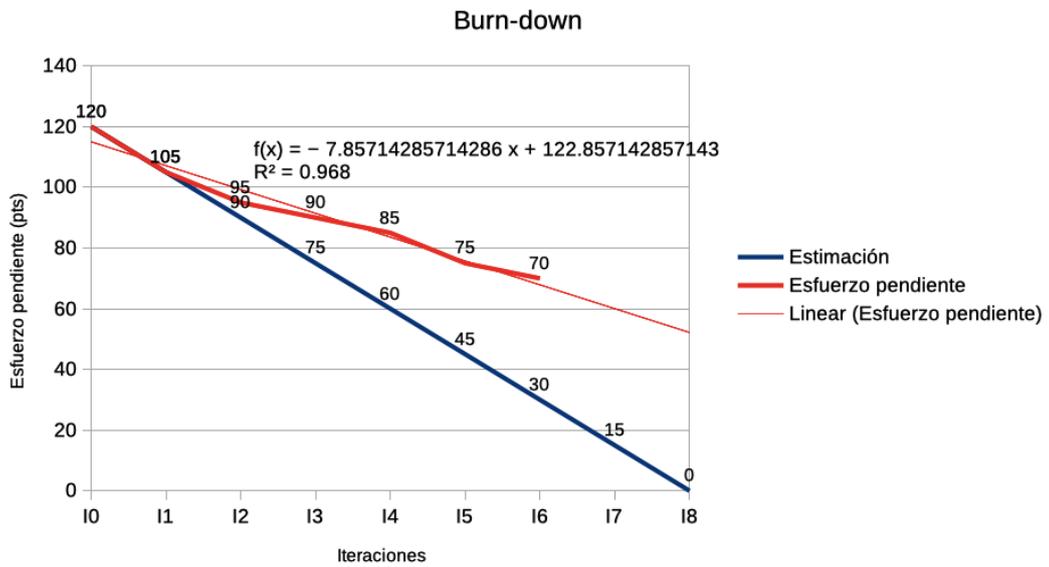


Figura 3.10: Burn-down - Tendencia lineal

CAPÍTULO IV

DIAGNÓSTICO DE FLUIDEZ ÁGIL

En este capítulo se definirán las necesidades de información, es decir las competencias que se requiere evaluar. A continuación se presentan el diseño de los instrumentos para recabar la información requerida y los resultados de la aplicación de estos instrumentos. El capítulo finalizará con el análisis-síntesis de los resultados obtenidos y la determinación de la zona de acuerdo a competencias ágiles para el proyecto objeto de estudio.

4.1 DISEÑO DE INSTRUMENTOS

Las competencias descritas en la sección 2.2 Modelo de Fluidez Ágil son comportamientos observables. En consecuencia se utilizarán métodos empíricos como la entrevista y la observación para levantar la información requerida (indicadores cualitativos) para este estudio descriptivo. Junto a esto se usarán los datos (indicadores cuantitativos) que proveen los sistemas de gestión de proyectos, integración continua, despliegue a producción y, de ser el caso, monitoreo de la aplicación.

Las tablas 4.1 (página 42), 4.2 (página 43), 4.3 (página 43) y 4.4 (página 44) contienen el detalle de las necesidades de información y las fuentes de información para la zonas de Equipos Enfocados, Equipos que Entregan, Equipos Optimizados y Equipos Fortalecidos, respectivamente.

4.1.1 Entrevistas

El objetivo de las entrevistas es recabar información con preguntas abiertas dirigidas a los miembros del equipo de desarrollo y al cliente. Con esto se trata de conocer de fuente de los involucrados lo que piensan y conocen sobre la realidad del proyecto considerando

Tabla 4.1: Fuentes de información - Equipos Enfocados

NECESIDAD DE INFORMACIÓN	FUENTE DE INFORMACIÓN
Clientes involucrados y empoderados	Observación
Fijación de prioridades	Observación, Sistema
Software que funciona y es valorado	Observación, Sistema
Fijación de expectativas sobre trabajo sostenible	Observación
Equipos auto-organizados	Observación
Trabajo y responsabilidad en equipo	Observación
Soporte de la administración respecto a la cultura de equipo	Observación
Mejora de trabajo en equipo	Observación
Mejora de relaciones en el equipo	Observación
Mejora de entorno físico	Observación

todos los puntos de vista.

Las siguientes preguntas están dirigidas tanto al administrador del proyecto como al equipo de desarrollo.

- ¿Qué metodología de desarrollo ágil utilizan?
- ¿Están certificados en esa metodología o en otra metodología de desarrollo ágil?
- ¿Cree que esa metodología está siendo de utilidad?
- ¿Cómo reconoce la empresa logros en el trabajo?

Las siguientes son dirigidas al administrador del proyecto, equipo de desarrollo y adicionalmente al cliente

- ¿Cuál es el problema más importante que identifica en el proyecto?
- ¿Cómo se hace el seguimiento de avance del proyecto?

Tabla 4.2: Fuentes de información - Equipos que Entregan

NECESIDAD DE INFORMACIÓN	FUENTE DE INFORMACIÓN
Integración continua	Sistema
Disponibilidad para liberar a producción	Sistema
Estimaciones de entrega realistas	Sistema
Soporte de la administración sobre trabajo reflexivo	Observación
Propiedad y responsabilidad colectiva	Observación
Equipos auto-suficientes y multifuncionales	Observación
Mejora continua, simplicidad	Sistema
Entrega continua, builds automáticos	Sistema, Observación
Cambios incrementales, trabajo de calidad, pruebas automáticas	Sistema, Observación
Mentalidad de aprendizaje continuo	Observación

Tabla 4.3: Fuentes de información - Equipos Optimizados

NECESIDAD DE INFORMACIÓN	FUENTE DE INFORMACIÓN
Responsabilidades del negocio	Observación
Retro-alimentación del mercado	Observación
Empoderamiento sobre la dirección del producto	Observación
Delegación de parte de la administración	Observación
Soporte de la administración respecto a la incorporación de expertos del negocio al equipo	Observación
Ejecución de experimentos	Observación
Dirección del producto	Sistema
Análisis de cohorte	Sistema

Tabla 4.4: Fuentes de información - Equipos Fortalecidos

NECESIDAD DE INFORMACIÓN	FUENTE DE INFORMACIÓN
Descripción de trabajo en el entorno	Observación
Involucramiento temprano	Observación
Actividades multi-equipos	Observación
Contribución entre equipos	Observación
Polinizaciones cruzadas	Observación

4.1.2 Observación

El proceso de observación consiste en acompañar el equipo de desarrollo en su trabajo diario y durante las reuniones prescritas por la metodología durante un mes. Se registrará y analizará la información y reacciones de los involucrados. El objetivo es comprobar las competencias del equipo.

A continuación se listarán las consideraciones que servirán de guía durante la observación de campo de acuerdo a las zonas del modelo de Fluidez Ágil.

4.1.2.1 Equipos Enfocados

Consideraciones para la zona de Equipos Enfocados:

- Cliente involucrado y empoderado. Se validará si el cliente tiene un representante que acompaña al equipo de desarrollo de forma permanente o periódica.
- Fijación de prioridades de parte del cliente. El cliente es quien tiene la última palabra sobre las siguientes funcionalidades en las que trabajará el equipo.
- Software que funciona. Se validará si el cliente valora el trabajo incremental del equipo.
- Fijación de expectativas sobre ritmo de trabajo sostenible por parte de administración de la empresa. El objetivo es averiguar si la administración permite que el equipo pueda trabajar a un ritmo sostenible de forma indefinida. La administración debe establecer estas expectativas con todos los involucrados.

- Equipo auto-organizado. El equipo define sus tareas y se organiza para cumplir con su plan. Nadie le indica cómo debe hacerlo.
- Trabajo y responsabilidad en equipo. Para cumplir con su objetivo los integrantes del equipo trabajan en conjunto, apoyándose entre sí. Entienden que no se valora cumplimientos individuales si no se ha alcanzado el objetivo.
- Soporte de administración respecto a cultura de equipo. La administración entiende y soporta ejecución del plan como un trabajo en equipo, no como una asignación individual de responsabilidades.
- Mejora de trabajo en equipo. El equipo reflexiona regularmente sobre su forma de trabajar, encuentra oportunidades de mejora y crea un plan para aplicarlas.
- Mejora de relaciones en el equipo. Se validará cómo son las relaciones interpersonales entre los miembros del equipo.
- Mejora de entorno físico. Se validará si el equipo está co-localizado, y cómo se realiza la comunicación con miembros que se encuentran en diferentes localidades.

4.1.2.2 Equipos que Entregan

A continuación las consideraciones para la zona de Equipos que Entregan:

- Soporte de la administración que valora al trabajo reflexivo sobre la conveniencia. La administración debe explicar y fijar expectativas a todos los involucrados sobre la importancia de contar con código de buena calidad para favorecer un trabajo sostenible.
- Propiedad y responsabilidad colectiva. No existe propiedad individual por ciertas partes del código, sino responsabilidad colectiva de todo el código. Todos tienen la responsabilidad de mejorarlo. No debe existir sub-equipos o silos dentro del equipo que creen productos intermedios.

- Equipos auto-suficientes y multifuncionales. El equipo posee todo el conocimiento y habilidad para completar sus tareas a través de todo el ciclo de desarrollo y mantenimiento. No se debe depender continuamente de personas que no forman parte del equipo.
- Entrega continua, builds automáticos. Se debe validar que los despliegues no tomen más de 10 minutos de esfuerzo manual.
- Cambios incrementales, trabajo de calidad, pruebas automáticas. Se debe validar que no se requiere ejecutar pruebas manuales para garantizar la calidad del producto. Las pruebas automáticas son un indicador suficiente de que el producto tiene la calidad requerida para ser desplegado a producción.
- Mentalidad de aprendizaje continuo. Se validará si los integrantes del equipo son conscientes que su destreza técnica es crucial para el equipo y proactivamente buscan mejorarlas. Comportamientos favorables que se deben buscar son cursos y certificaciones, jornadas de difusión técnica en la comunidad,

4.1.2.3 Equipos Optimizados

Las siguientes son las consideraciones para la zona de los Equipos Optimizados.

- Responsabilidades del negocio. Se debe validar si el equipo describe y hace seguimiento de su plan en términos de métricas del negocio.
- El equipo recibe retro-alimentación del mercado para la toma de decisiones sobre su producto.
- Empoderamiento sobre el producto. La condición necesaria para exigir responsabilidad para que el equipo alcance resultados del negocio es empoderarlo sobre la dirección de producto.
- Delegación de parte de la administración para lograr los resultados del negocio.
- Soporte de la administración para incorporar expertos del negocio al equipo de forma permanente. Esto requiere cambios en la estructura organizacional.

- El equipo ejecuta experimentos para probar hipótesis del negocio.

4.1.2.4 Equipos Fortalecidos

Las consideraciones para validar si el equipo se encuentra en la zona de Equipos Fortalecidos se listan a continuación.

- La descripción de trabajo en el entorno se refiere a la capacidad del equipo para describir su trabajo en el contexto de otras iniciativas del negocio, permitiendo que los productos se equilibren entre sí.
- El involucramiento temprano se logra cuando el equipo levanta y ayuda a abordar impedimentos organizacionales y problemas de forma temprana.
- También se busca que el equipo participe en actividades multi-equipos que optimizan el flujo de valor organizacional.
- El equipo reconoce cuándo puede contribuir en el trabajo de otro equipo. En el caso de que el trabajo de otro equipo sea más importante, el equipo redirige sus esfuerzos para ayudarlo.
- El equipo realiza polinizaciones cruzadas acerca de puntos de vista, contextos, aprendizaje, innovaciones con otros equipos y otras partes de la organización.

4.1.3 Índices

Se emplearán los índices cuantitativos indicados en las siguientes tablas.

Algunas de estas métricas fueron inspiradas y/o adaptadas de los trabajos de Skarin (2015), Davis (2015) y de Rasmusson (2010).

- Tabla 4.5: Prioridades y progreso incremental (página 49).
- Tabla 4.6: Trabajo reflexivo y estimaciones útiles (página 50).
- Tabla 4.7: Calidad de código (página 51).
- Tabla 4.8: Índice de mantenibilidad (página 52).

- Tabla 4.9: Índice de complejidad ciclomática (página 52).
- Tabla 4.10: Índice de acoplamiento (página 52).
- Tabla 4.11: Índice de niveles de herencia (página 52).
- Tabla 4.12: Integración continua (página 53).
- Tabla 4.13: Despliegue (página 54).
- Tabla 4.14: Dirección del producto (página 55).
- Tabla 4.15: Análisis de cohorte (página 56).

4.2 RESULTADOS

En esta sección se presentan los resultados de la aplicación de los instrumentos, diseñados en la sección anterior, al equipo del proyecto Partner Portal.

4.2.1 Entrevistas

A continuación se resume la información recabada mediante las entrevistas. Se presentan las respuestas dadas por los integrantes del equipo del proyecto Partner Portal, consolidadas por cada pregunta.

4.2.1.1 Metodología de desarrollo utilizada

El administrador del proyecto asevera que se está usando la metodología desarrollada internamente, conocida como Wave.

Dentro del equipo de desarrollo no existe un consenso respecto a la metodología que se está usando. La mayor parte indica que es la nueva metodología de desarrollo (Wave).

Unos pocos creen que en realidad se está aplicando Scrum de forma parcial. Por ejemplo mencionan que no existe el rol de Scrum Master, cuya función es velar por el equipo y el cumplimiento de la metodología. Indican que ese rol es asumido, en teoría, por el administrador del proyecto, que a la vez es el analista del negocio.

Tabla 4.5: KPIs - Prioridades y progreso incremental

Indicador	Definición	Fórmula
Funcionalidades priorizadas por iteración	Porcentaje de funcionalidades priorizadas en la iteración	$\text{No. de funcionalidades priorizadas en la iteración} / \text{No. total de funcionalidades de la iteración} * 100$
Funcionalidades finalizadas por iteración	Porcentaje de funcionalidades finalizadas en la iteración	$\text{No. de funcionalidades finalizadas en la iteración} / \text{No. total de funcionalidades de la iteración} * 100$
Velocidad	Puntos completados por iteración	Sumatoria de puntos de todas las funcionalidades finalizadas en la iteración
Velocidad promedio	Puntos completados en promedio en todas la iteraciones	Sumatoria de velocidad de todas las iteraciones / No. de iteraciones
Esfuerzo pendiente	Puntos pendientes para finalizar el proyecto	Sumatoria de puntos de todas la funcionalidades - Sumatoria de puntos de funcionalidades finalizadas
Esfuerzo pendiente porcentual	Porcentaje de puntos pendientes	$\text{Puntos pendientes} / \text{Sumatoria de puntos de todas las funcionalidades} * 100$

Tabla 4.6: KPIs - Trabajo reflexivo y estimaciones útiles

Indicador	Definición	Fórmula
Puntos estimados	Esfuerzo total estimado para la iteración en puntos. Objetivo de la iteración	Sumatoria de puntos estimados de todas la funcionalidades de la iteración
Puntos completados	Esfuerzo total cumplido en la iteración en puntos	Sumatoria de puntos de funcionalidades completadas en la iteración
Puntos estimados nuevas funcionalidades	Esfuerzo total estimado para nuevas funcionalidades para la iteración en puntos	Sumatoria de puntos estimados para nuevas funcionalidades de la iteración
Puntos completados nuevas funcionalidades	Esfuerzo total en nuevas funcionalidades en la iteración en puntos	Sumatoria de puntos de nuevas funcionalidades completadas en la iteración
Puntos estimados corrección de errores/mantenimiento	Esfuerzo total estimado para corrección de errores y mantenimiento para la iteración en puntos	Sumatoria de puntos estimados para corrección de errores y mantenimiento de la iteración
Puntos completados corrección de errores/mantenimiento	Esfuerzo total en corrección de errores y mantenimiento en la iteración en puntos	Sumatoria de puntos de corrección de errores y mantenimiento completados en la iteración
Puntos estimados pago deuda técnica	Esfuerzo total estimado para pago deuda técnica para la iteración en puntos	Sumatoria de puntos estimados para pago deuda técnica de la iteración
Puntos completados pago deuda técnica	Esfuerzo total en pago deuda técnica en la iteración en puntos	Sumatoria de puntos de pago deuda técnica completados en la iteración
Lead time	Tiempo en días que tarda una funcionalidad desde que se inicia a trabajar en ella hasta que está lista para ser usada en producción	Medida entre fecha en que la funcionalidad está lista para ser usada en producción y fecha en que se inicia el trabajo en esa funcionalidad

Tabla 4.7: KPIs - Calidad de código

Indicador	Definición	Fórmula
Índice de mantenibilidad	Indica qué tan fácil se puede entender y modificar el código fuente	Calculado por herramienta de desarrollo
Complejidad ciclomática	Mide el número de caminos linealmente independientes a través del código fuente de un programa	Calculado por herramienta de desarrollo
Acoplamiento	Mide el grado de interdependencia entre componentes de software	Calculado por herramienta de desarrollo
Niveles de herencia	Mide el número de niveles de herencia en lenguajes de programación orientados a objetos	Calculado por herramienta de desarrollo

Tabla 4.8: Índice de mantenibilidad

Rango	Evaluación
0 a 9	Baja mantenibilidad
10 a 19	Moderada mantenibilidad
20 a 100	Buena mantenibilidad

Tabla 4.9: Índice de complejidad ciclomática

Rango	Evaluación
>50	Código de muy alto riesgo
21 a 50	Código de alto riesgo
11 a 20	Riesgo moderado
1 a 10	Riesgo bajo

Tabla 4.10: Índice de acoplamiento

Rango	Evaluación
>80	Las dependencias son críticas
10 a 80	Las dependencias son aun manejables
0 a 9	Las dependencias son adecuadas

Tabla 4.11: Índice de niveles de herencia

Rango	Evaluación
>4	Crítico
3 a 4	Manejable
1 a 2	Adecuado

Tabla 4.12: KPIs - Integración Continua

Indicador	Definición	Fórmula
Builds ejecutados	Número de builds de código ejecutados automáticamente por iteración	Sumatoria de builds ejecutados en la iteración
Builds exitosos	Número de builds de código automáticos sin errores de compilación por iteración	Sumatoria de builds exitosos en la iteración
Builds fallidos	Número de builds de código automático con errores de compilación por iteración	Sumatoria de builds fallidos en la iteración
Pruebas ejecutadas	Número total de pruebas automáticas ejecutadas por iteración	Sumatoria de pruebas ejecutadas en la iteración
Pruebas exitosas	Número de pruebas automáticas exitosas por iteración	Sumatoria de pruebas exitosas en la iteración
Pruebas fallidas	Número de pruebas automáticas fallidas por iteración	Sumatoria de pruebas fallidas en la iteración
Cobertura de código	Porcentaje de líneas de código cubiertas por pruebas por iteración	Líneas de código cubiertas por pruebas / Líneas de código totales * 100

Tabla 4.13: KPIs - Despliegue

Indicador	Definición	Fórmula
Despliegues exitosos mensuales	Despliegues que se mantienen en producción luego de su aplicación	Sumatoria de despliegues exitosos mensuales
Despliegues fallidos mensuales	Despliegues rechazados por el cliente (deben revertirse) o por el propio equipo de desarrollo (no se desplegó a producción o se tuvo que reversar)	Sumatoria de despliegues fallidos mensuales

Tabla 4.14: KPIs - Dirección del producto

Indicador	Definición	Fórmula/modo de calcular
Aprobado por el usuario	El producto/nueva funcionalidad provee valor al usuario	Tests A/B, entrevistas, encuestas, que demuestran un cambio positivo en el comportamiento del usuario
Rechazado por el usuario	El producto/nueva funcionalidad no provee valor al usuario o no lo utiliza	Test A/B, entrevistas, encuestas, que no demuestran un cambio positivo en el comportamiento del usuario

Tabla 4.15: KPIs - Análisis de cohorte

Indicador	Definición	Fórmula/modo de calcular
Usuarios por tipo	Porcentaje de usuarios por tipo cada mes	Sumatorio de usuarios por tipo / Total de usuarios * 100
Tasas de conversión de usuarios	Porcentaje de usuarios que se suscriben al servicio por mes	No. usuarios suscritos / Total usuarios * 100
Uso de producto/funcionalidad	Tiempo promedio que un tipo de usuario interactúa con el producto o una funcionalidad en particular a la semana	Sumatoria de tiempo de uso por tipo de usuario / Tiempo total de uso * 100

4.2.1.2 *Certificaciones en la metodología utilizada*

El administrador del proyecto, que también tiene el rol de analista de negocio, y el tester son Scrum Master certificados. Aprovecharon una capacitación que auspició la empresa hace un par de años y se certificaron como Scrum Masters. Creen que el conocimiento en Scrum sería de mucha utilidad porque Wave se basa en gran medida en Scrum.

Los demás integrantes del equipo no tienen capacitaciones en metodologías de desarrollo.

El equipo de desarrollo menciona que pese a que existen personas certificadas como Scrum Master, no existe ese rol en el equipo. Creen que es importante el rol de Scrum Master porque éste trae el balance requerido en este tipo de proyectos, es decir alguien que respalde al equipo. Saben que la función del administrador del proyecto es sacarlo adelante, pero en varias ocasiones ha dejado de considerar el bienestar del equipo.

4.2.1.3 *Utilidad de la metodología*

Tanto el administrador del proyecto como el equipo de desarrollo creen que Wave no está siendo de utilidad.

Hay varios signos de dificultad, por ejemplo, el equipo trabaja demasiadas horas extra, a menudo hasta la 1 de la mañana (hasta 80 horas/hombre semanales). El equipo reconoce que este esfuerzo no ha sido recompensado con resultados. Difícilmente cumplirá con el plazo y el alcance del proyecto en el último mes de trabajo.

4.2.1.4 *Reconocimiento de logros*

Cuando se preguntó a los miembros del equipo, incluido el administrador del proyecto, cómo la empresa reconoce logros en el trabajo, todos respondieron que la empresa otorga premios y reconocimientos individuales.

Informaron que existe una clasificación de las personas en función de los ingresos que aportan para la empresa.

Consideran que este tipo de tratamiento es problemático porque no importa qué tan buen trabajo hagan individualmente en un proyecto si éste representa menores ingresos comparados con otros clientes de la empresa. No es algo que puedan controlar ya que la

asignación es una decisión de la administración de la empresa. Concluyen que se los está evaluando por algo que ellos no pueden controlar.

4.2.1.5 Problema más importante

Al preguntar a cada miembro del equipo que identifique el problema más importante, todos tienen su punto de vista sobre la situación, no existe una perspectiva en común.

El administrador del proyecto ve al equipo con dificultades para entregar productos de calidad, continuamente subestimando el nivel de esfuerzo y tiempos requeridos, y estresándose demasiado. Las tareas tardan hasta cinco veces más en ser completadas respecto al tiempo estimado originalmente.

Los desarrolladores indican que no pueden mantener un flujo de trabajo continuo en la jornada laboral diaria y durante cada iteración. Existen interrupciones: otras reuniones a más de las que se indican en la metodología, entorno físico que distrae.

El flujo de trabajo se ve interrumpido además por la asignación de otras tareas urgentes. Los desarrolladores creen que hay muchos cambios entre tareas, es decir, pasan de una tarea a otra sin haberla terminado antes, haciendo varias cosas al mismo tiempo.

Indican que no existe confianza de parte del administrador del proyecto en el equipo de desarrollo. Esto se refleja en el hecho de que pese a que el equipo se debe organizar a sí mismo, según la metodología, el administrador del proyecto toma las riendas asignando tareas e indicando cómo deben hacerlas.

También indican que existen re-trabajos en tareas en las que ya habían comenzado a trabajar, que luego fueron descartadas del sprint, pero posteriormente aparecían un sprint posterior. Debido a que el contexto había cambiado, era necesario re-hacer el trabajo.

Creen que es necesario un espacio en el que puedan discutir los problemas que existen, pero no siempre se hace debido a la falta de tiempo para llevar a cabo la reunión que supuestamente debe contemplar estos temas, que es la reunión de retrospectiva.

Otra causa relevante manifestada por el equipo de desarrollo es la re-asignación de miembros del equipo. La administración de la empresa regularmente asigna miembros de un equipo de desarrollo a otro equipo, especialmente desarrolladores de «front end».

Por otro lado, el cliente cree que el problema principal del proyecto es que los miem-

bros del equipo de desarrollo no tienen las competencias necesarias para cumplir con su trabajo, es decir, entregarle valor desde el inicio y continuamente. Este es el resultado de entregas fallidas o incompletas al finalizar cada iteración.

4.2.1.6 Seguimiento de avance

Al preguntar sobre la forma en que se hace el seguimiento del avance del proyecto se encontró que cada parte hace el seguimiento de cosas diferentes.

Mientras el administrador del proyecto controla qué actividades están siendo completadas y hace un seguimiento de la planificación global del proyecto, el equipo de desarrollo controla el progreso únicamente de la iteración actual.

El equipo de desarrollo informa que trabaja en tareas urgentes asignadas por el administrador del proyecto, que no siempre son visibles en el tablero Kanban.

El cliente por su parte controla el avance en función de la información que le provee el administrador del proyecto. Su principal preocupación es el cumplimiento del objetivo fijado al inicio de cada iteración.

4.2.2 Observación

Los resultados de la técnica de observación se presentan en las tablas 4.16, 4.17, 4.18, 4.19 a manera de listas de chequeo. Si cumple completamente con la descripción, se usa el símbolo , caso contrario . Cuando no se haya podido validar se usa .

4.2.2.1 Equipos Enfocados

En referencia al listado de la zona de Equipos Enfocados (tabla 4.16), se pudo determinar que cumplen con las competencias «Cliente involucrado y empoderado», «Fijación de prioridades por parte del cliente» y «Mejora de relaciones en el equipo».

- **Cliente involucrado y empoderado**

En el primer caso si bien el cliente no está presente de forma permanente, sí está disponible cuando se lo requiera previa solicitud y coordinación de reunión de acuerdo a la planificación de la iteración.

Tabla 4.16: Lista de chequeo - Equipos Enfocados

<input checked="" type="checkbox"/>	Cliente involucrado y empoderado
<input checked="" type="checkbox"/>	Fijación de prioridades por el cliente
<input checked="" type="checkbox"/>	Software que funciona y es valorado
<input checked="" type="checkbox"/>	Fijación de expectativas por la administración sobre ritmo de trabajo sostenible
<input checked="" type="checkbox"/>	Equipo auto-organizado
<input checked="" type="checkbox"/>	Trabajo y responsabilidad en equipo
<input checked="" type="checkbox"/>	Soporte de administración respecto a cultura de equipo
<input checked="" type="checkbox"/>	Mejora de trabajo en equipo
<input checked="" type="checkbox"/>	Mejora de relaciones en el equipo
<input checked="" type="checkbox"/>	Mejora de entorno físico

- **Fijación de prioridades por parte del cliente**

El cliente está empoderado en la dirección que debe tomar el equipo, ya que al inicio de cada sprint o iteración puede modificar las prioridades sobre las que trabajará el equipo.

- **Mejora de relaciones en el equipo**

Se pudo percibir que las relaciones interpersonales son buenas, existe un ambiente cordial y de respeto entre todos.

Las competencias que se listan a continuación no se cumplen.

- **Software que funciona y es valorado**

Se pudo validar que los incrementos del producto no son de completa utilidad para el cliente. En las reuniones de retrospectiva se realizaron demos del sistema que no contemplaban todo el alcance planificado para la iteración.

- **Fijación de expectativas por la administración sobre ritmo de trabajo sostenible**

El hecho de que el equipo deba trabajar horarios extendidos para poder cumplir con las metas de proyecto hace que el trabajo no sea sostenible. El equipo de desarrollo

sabe que debe cumplir con las fechas y se esfuerza para ello; sin embargo, no se logran las metas.

La administración ha fallado en cumplir con la fijación de expectativas respecto a un trabajo sostenible con el cliente y con el propio equipo de desarrollo.

- **Equipo auto-organizado**

Si bien en la fase de planificación de la iteración el equipo de desarrollo crea un plan de tareas para resolver las funcionalidades en las que debe trabajar, nadie externo al equipo, incluido el administrador del proyecto, le indica cómo debe hacerlo. Pero durante la iteración, debido a los retrasos, el administrador de proyectos toma el mando del equipo y distribuye las tareas que él considera se deben hacer, muchas veces provocando que los desarrolladores deban trabajar en paralelo en varias tareas o dejen tareas inconclusas que en iteraciones posteriores son retomadas.

- **Trabajo y responsabilidad en equipo**

El equipo de desarrollo no demuestra un verdadero trabajo en equipo ni responsabilidad común. Existe soporte entre ellos, pero únicamente entre miembros del mismo rol. El trabajo se subdivide en silos y cada rol es responsable únicamente de una parte de esa responsabilidad general.

- **Soporte de administración respecto a cultura de equipo**

Si bien la administración no asigna responsabilidades individualmente, sí lo promueve indirectamente a nivel de roles. Se definen claramente los roles y sus responsabilidades.

Otro rasgo que se pudo observar es el sistema de evaluaciones de desempeño y reconocimientos individuales en reuniones trimestrales de toda la empresa en base a lo que cada individuo produjo económicamente a la empresa. Si alguien pertenece a un equipo que trabaja para un cliente cuyo rendimiento financiero es inferior al de otros proyectos, no importa qué tan bien se desempeña esa persona, de forma individual o como contribución hacia el equipo o a la empresa en general, es imposible que su trabajo sea reconocido y premiado.

Tabla 4.17: Lista de chequeo - Equipos que Entregan

<input checked="" type="checkbox"/>	Soporte de la administración respecto a la valoración del trabajo reflexivo
<input checked="" type="checkbox"/>	Propiedad y responsabilidad colectiva
<input checked="" type="checkbox"/>	Equipos auto-suficientes y multifuncionales
<input checked="" type="checkbox"/>	Entrega continua, builds automáticos
<input checked="" type="checkbox"/>	Cambios incrementales, trabajo de calidad, pruebas automáticas
<input checked="" type="checkbox"/>	Mentalidad de aprendizaje continuo

Por lo tanto no existe el soporte a una verdadera cultura de equipo con responsabilidad común: que el software esté listo para ser usado por el usuario.

■ Mejora de trabajo en equipo

Las reuniones de retrospectiva de sprint no se las hace regularmente al finalizar cada sprint. Cuando tienen lugar, las ideas de mejora no se las concretan en planes o éstos no se ejecutan en el siguiente sprint. Por lo tanto no se observa competencia en la mejora de trabajo en equipo.

■ Mejora de entorno físico

No se ha podido observar iniciativas para mejorar el entorno físico. El entorno físico no brinda las condiciones necesarias para que el equipo se pueda enfocar en sus tareas. La oficina tiene un estilo abierto, sin separadores o cubículos individuales. Este tipo de oficinas es ideal para fomentar las interacciones entre las personas; sin embargo, el equipo se encuentra rodeado por otro equipo desarrollo. La interacciones de dos equipos cercanos, incluso el uno rodeando al otro, provocan distracciones continuas.

4.2.2.2 *Equipos que Entregan*

En la siguiente lista de chequeo, relacionada con la zona de Equipos que Entregan (tabla 4.17), se puede ver que el equipo no cumple con ninguna de las competencias de esta zona.

■ **Soporte de la administración respecto a la valoración del trabajo reflexivo**

Se pudo constatar que la administración busca cumplir con los plazos comprometidos, y no fija expectativas respecto a la importancia de que el equipo produzca código de calidad y mantenible, que generalmente requiere un trabajo más reflexivo y toma más tiempo.

La dirección de la empresa considera que el cliente no les está pagando por producir código de calidad, si no que es suficiente con que el producto funcione relativamente bien.

Sin embargo, tres meses atrás habían perdido a su cliente más importante, de la industria de la salud, debido a los continuos despliegues a producción fallidos, que tomaban días para ser desplegados y estabilizados. Ese proyecto arrastraba una deuda técnica enorme que hacía cada vez más difícil incluir modificaciones.

■ **Propiedad y responsabilidad colectiva**

No existe una propiedad colectiva de todo el código. Existen dueños de ciertas partes del mismo, que corresponden a las funcionalidades que desarrollaron. Si alguien no trabajó en una parte del código, no lo modifica.

■ **Equipos auto-suficientes y multifuncionales**

Debido a que el proyecto Partner Portal es una adición a un proyecto más grande que construyó el sitio de la universidad, varias implementaciones con las que se debe integrar este proyecto solo las conocen otros desarrolladores que no forman parte del proyecto. Por lo tanto se requiere consultar o solicitar apoyo a otros miembros de la empresa, lo que implica que el equipo no sea auto-suficiente.

El equipo cuenta con una base fija de desarrolladores «back-end», 3 de los 4 desarrolladores del equipo. El cuarto desarrollador, que es el desarrollador «front-end», alterna su trabajo entre varios equipos de la empresa. Cuando se lo requiere de acuerdo a la planificación de la iteración se coordina su participación, pero no siempre se lo consigue a tiempo para que apoye al equipo, lo que provoca retrasos. Esto implica que el equipo no sea multifuncional.

- **Entrega continua, builds automáticos**

Los despliegues a los ambientes de pruebas y producción se encuentran automatizados. El esfuerzo manual en el proceso de despliegue en sí mismo es menor a 10 minutos. Sin embargo, se ha incluido la elaboración manual de un documento que no agrega valor al proceso y que toma de una a dos horas del trabajo de todos los miembros del equipo. El documento es una formalización que se entrega al cliente y contiene el listado de los componentes y las funcionalidades que se despliegan al ambiente.

Debido que el proceso requiere intervención manual para la elaboración del documento de entrega, que puede tomar hasta un par de horas de todo el equipo, no se cumple con esta competencia.

- **Cambios incrementales, trabajo de calidad, pruebas automáticas**

Si bien se realizan cambios incrementales sobre el producto y existen pruebas automáticas, éstas no tienen la cobertura requerida para garantizar por sí solas la calidad del producto. Es necesario realizar adicionalmente pruebas manuales para validar la funcionalidad. Estas pruebas son ejecutadas por el tester del equipo.

El tester del equipo no conoce completamente los cambios que ha realizado el equipo, por lo que requiere el soporte de los miembros del equipo para explicarle las últimas modificaciones en funcionalidad.

Al realizar las pruebas manuales el tester encuentra continuamente errores que son reportados y agregados a la lista de tareas que se deben completar en la iteración. Esto provoca re-trabajos en funcionalidades que se suponían listas y quita tiempo para trabajar en las demás funcionalidades que se habían planificado para la iteración.

- **Mentalidad de aprendizaje continuo**

Existen esfuerzos individuales para mejorar las capacidades técnicas dentro del equipo de desarrollo, pero no se pudo observar este comportamiento de forma extendida en todo el equipo.

Se ha implementado un programa de desarrollo de capacidades técnicas mediante

Tabla 4.18: Lista de chequeo - Equipos Optimizados

☒	Plan en términos de métricas del negocio
☒	Toma de decisiones en función de la retro-alimentación del mercado
☒	Empoderamiento sobre el producto
☒	Soporte de administración para incorporar expertos del negocio
☒	Ejecución de experimentos para probar hipótesis del negocio

un plan de carrera. Los arquitectos de la empresa son los coaches, a los que se les ha asignado un grupo de desarrolladores. El objetivo es trazar objetivos anuales y un cronograma con actividades de aprendizaje con la coordinación de los coaches. Se pudo observar que este programa no está dando los resultados deseados: el seguimiento de cumplimiento es trimestral, cerca de la fecha de evaluación trimestral los desarrolladores buscan completar superficialmente sus objetivos.

Por esta razón se puede concluir que no existe una cultura de aprendizaje continuo en el equipo.

4.2.2.3 Equipos Optimizados

En la lista de chequeo de Equipos Optimizados (tabla 4.18) se observa que el equipo no cumple con ninguna de las competencias de esta zona.

- **Plan en términos de métricas del negocio**

El equipo no utiliza métricas del negocio, como por ejemplo número de usuarios que usan el sitio, tiempo que permanecen en el sitio, usuarios que regresan al sitio, tasa de conversión de usuarios, incremento de ROI. Las métricas que utiliza son las del avance del proyecto.

- **Toma de decisiones en función de la retro-alimentación del mercado**

Esta función es llevada a cabo por el cliente, quien toma las decisiones sobre la dirección de su producto y se las comunica a Hyna para que las implemente. El ámbito de la toma de decisiones se restringe a lo técnico.

Tabla 4.19: Lista de chequeo - Equipos Fortalecidos

-
- ☒ Descripción de trabajo en el entorno
 - ☒ Involucramiento temprano
 - ☒ Actividades multi-equipos
 - ☒ Contribución entre equipos
 - ☒ Polinizaciones cruzadas
-

- **Empoderamiento sobre el producto**

El contrato del proyecto no incluye esta atribución, la empresa provee el equipo de desarrollo de software, el cliente no es considerado como un socio con el que se trabaja en conjunto sobre la dirección del producto. El equipo trata de seguir fielmente la guía acerca del producto que le indica el cliente.

- **Soporte de administración para incorporar expertos del negocio**

La administración de Hyna no considera incorporar expertos del negocio al equipo. Se asume que el cliente es quien conoce el negocio y la empresa sigue el programa de proyectos que indica el cliente, sin recibir retro-alimentación de Hyna.

El analista del negocio del proyecto únicamente da soporte al cliente para facilitar la interpretación de sus necesidades, las que luego son transmitidas al equipo de desarrollo.

- **Ejecución de experimentos para probar hipótesis del negocio**

El equipo tampoco lleva a cabo experimentos para probar hipótesis del producto. Todas estas son responsabilidades del cliente.

Debido a que durante el proyecto los despliegues a producción no se han habilitado a los usuarios, el cliente tampoco los ha ejecutado.

4.2.2.4 *Equipos Fortalecidos*

Finalmente, la tabla 4.19 resume que tampoco cumple con ninguna de sus competencias de los Equipos Fortalecidos.

■ **Descripción de trabajo en el entorno**

Al ser una empresa proveedora de tecnología que sigue los lineamientos del negocio que le da el cliente, el equipo no considera entre sus tareas la definición de su trabajo considerando el contexto de otras iniciativas del cliente.

El equipo tampoco trabaja activamente para alinear su propósito dentro de la estrategia del cliente.

Desde la perspectiva de la empresa, es decir Hyna, el equipo no describe su trabajo en el contexto estratégico de otros productos o servicios que provea la empresa. Esto es trabajo de la administración de Hyna.

■ **Involucramiento temprano**

El equipo no ha identificado impedimentos organizacionales ni se ha involucrado de forma temprana para evitarlos. No ha tenido una posición crítica o propositiva respecto a problemas con la estructura y cultura organizacional.

Se pudo observar cierta retro-alimentación sobre la disposición física de los equipos de trabajo en la oficina, que tiene un formato de oficina abierta, pero el responsable de los últimos cambios de ubicación no ha actuado en favor del equipo. La administración no delega este tipo de decisiones ni ha tomado en cuenta las molestias que puedan causar.

■ **Actividades multi-equipos**

El equipo del proyecto Partner Portal y otros equipos de desarrollo de la empresa trabajan concentrados en su proyecto. No existe participación en actividades multi-equipos para optimizar el trabajo de la empresa como un todo.

■ **Contribución entre equipos**

Tampoco se pudo observar que el equipo tuviera tiempo para dar soporte a otros equipos que tengan mayor prioridad o que otro equipo lo haga para dar soporte al equipo de Partner Portal. Se pudo notar el soporte individual de desarrolladores de otros equipos de manera informal cuando su carga de trabajo lo permitía.

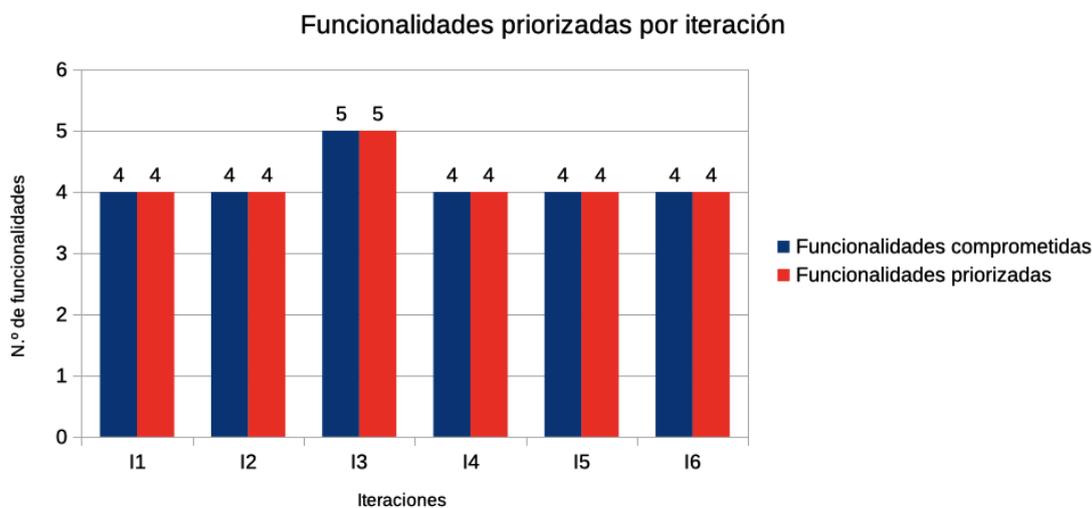


Figura 4.1: Funcionalidades Priorizadas por Iteración - Equipos Enfocados

■ Polinizaciones cruzadas

El equipo no comparte su conocimiento ni busca aprender de otros equipos de forma activa. La presión por los tiempos entrega hace que se concentre exclusivamente en la implementación de las funcionalidades del proyecto Partner Portal.

4.2.3 Índices

En esta sección se presentan los índices recopilados de las herramientas usadas por el equipo de trabajo.

4.2.3.1 Equipo Enfocados

Se incluyen los índices de la tabla 4.5, página 49, correspondientes a Equipos Enfocados, figura 4.1: funcionalidades priorizadas por iteración, figura 4.2: funcionalidades finalizadas por iteración, figura 4.3: velocidad por iteración, figura 4.4: burn-down o esfuerzo pendiente y figura 4.5: esfuerzo pendiente porcentual.

4.2.3.2 Equipos que Entregan

Seguidamente se incluyen los índices de la tabla 4.6, página 50, correspondientes a Equipos que Entregan. Las métricas de puntos estimados y puntos completados pueden

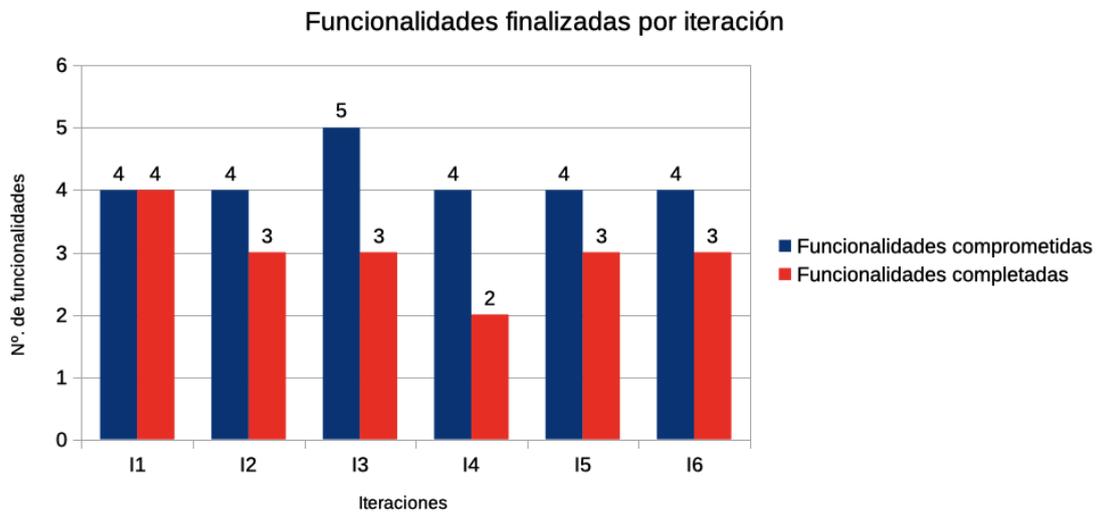


Figura 4.2: Funcionalidades Finalizadas por Iteración - Equipos Enfocados

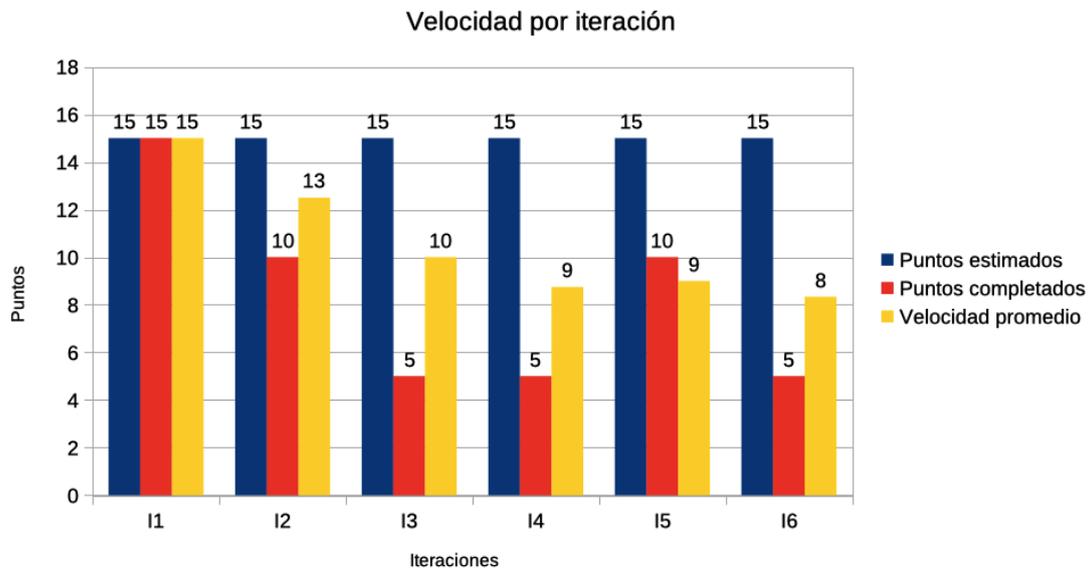


Figura 4.3: Velocidad, velocidad promedio - Equipos Enfocados

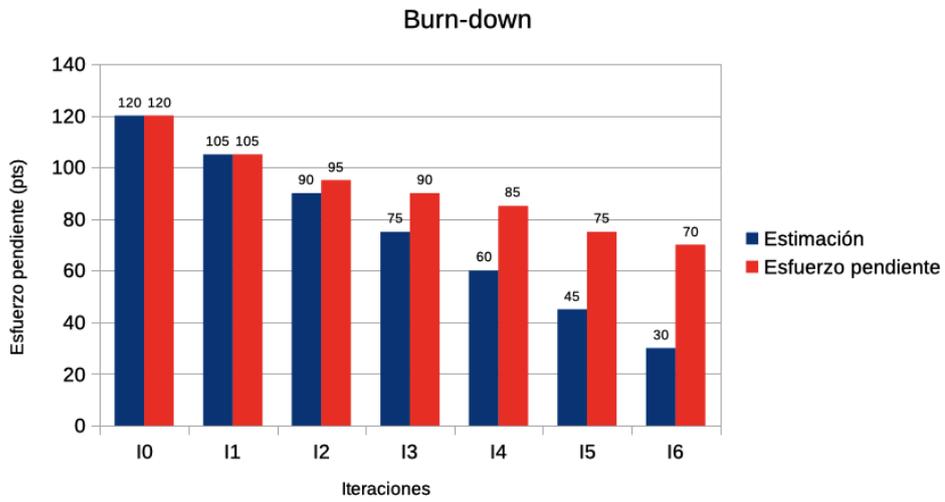


Figura 4.4: Esfuerzo pendiente - Equipos Enfocados

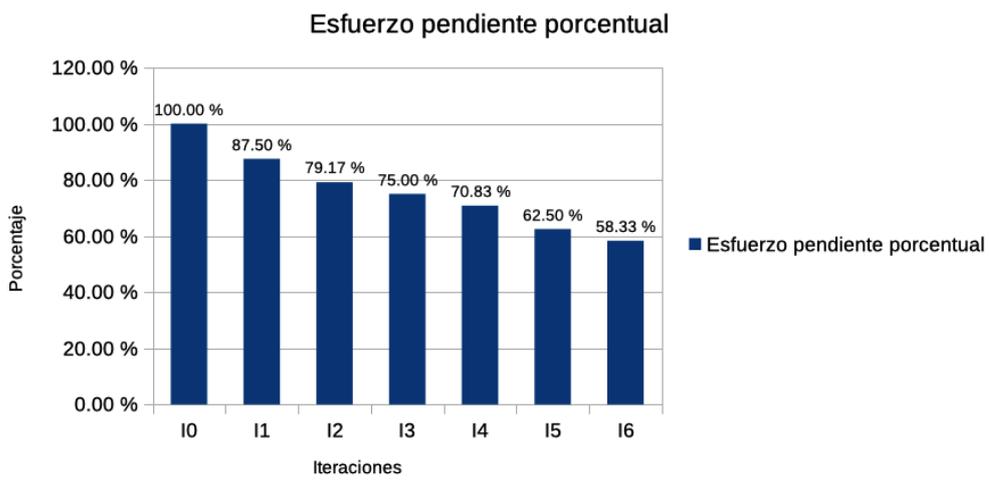


Figura 4.5: Esfuerzo pendiente porcentual - Equipos Enfocados

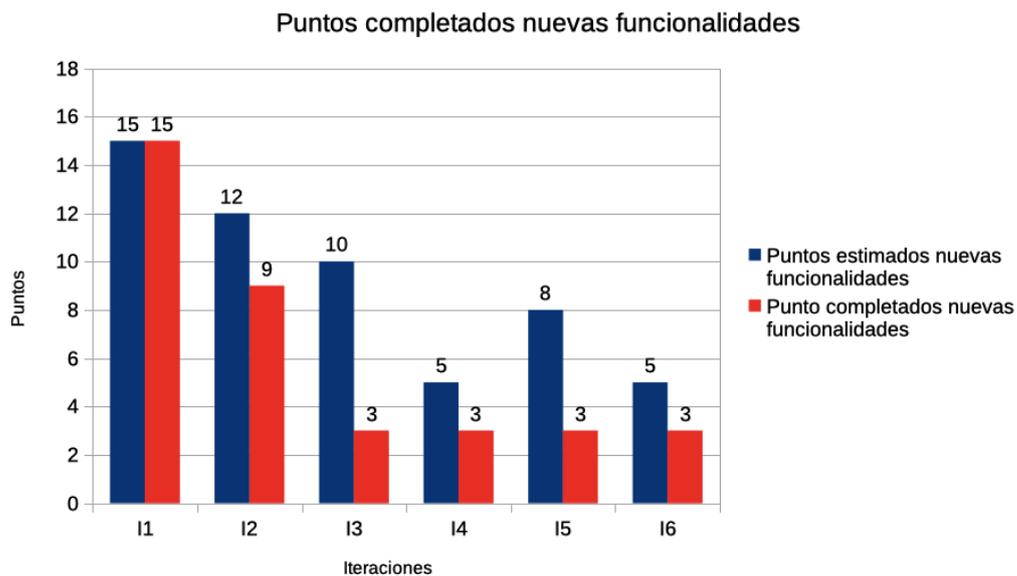


Figura 4.6: Nuevas funcionalidades - Equipos que Entregan

verse en la figura 4.3: velocidad. Las métricas restantes se encuentran en la figura 4.6: nuevas funcionalidades, figura 4.7: corrección de errores, figura 4.8: tipo de tarea y figura 4.9: lead time.

Las figuras 4.10: mantenibilidad, 4.11: complejidad ciclomática, 4.12: acoplamiento, 4.13: niveles de herencia, contienen las mediciones en cuanto a calidad de código.

Respecto a los índices de Integración Continua de la tabla 4.12 de la página 53, los resultados se muestran en las figuras 4.14: builds, 4.15: pruebas automáticas, 4.16: cobertura de código.

Los resultados de la última tabla de índices de los Equipos que Entrega, KPIs de despliegue, tabla 4.13 de la página 54, se muestran en la figura 4.17: despliegues a producción.

4.2.3.3 Equipos Optimizados

No existe información referente a los índices de la zona de Equipo Optimizados: tabla 4.14, página 55, sobre dirección de producto y tabla 4.15, página 56, análisis de cohorte. Si bien se han hecho despliegues a producción, el cliente no ha habilitado el producto para que pueda ser usado por el usuario. Piensa hacerlo una vez que se haya terminado el proyecto, que consiste únicamente en el conjunto de funcionalidades mínimas que requiere

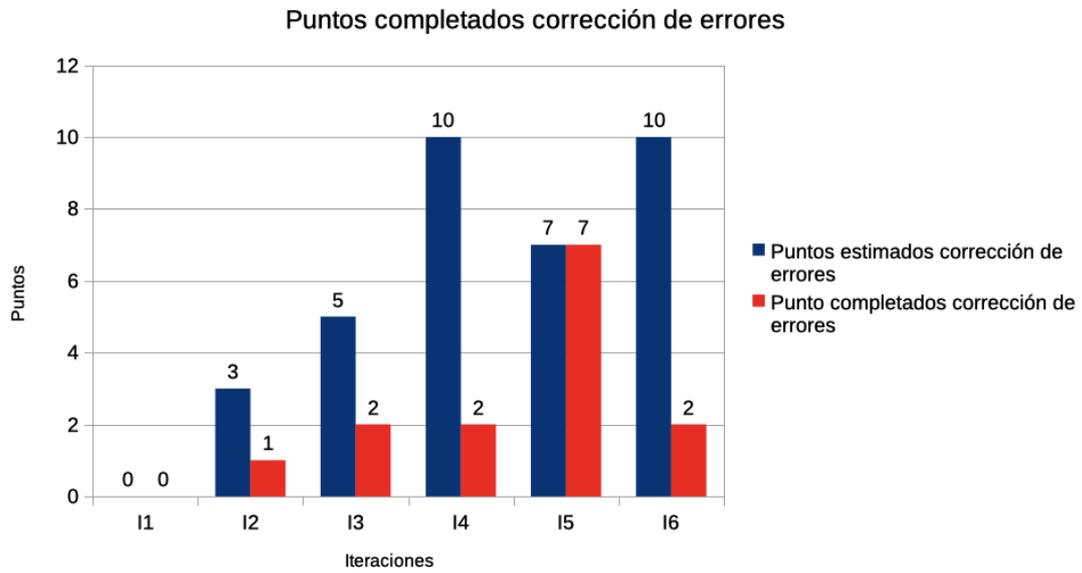


Figura 4.7: Corrección de errores - Equipos que Entregan

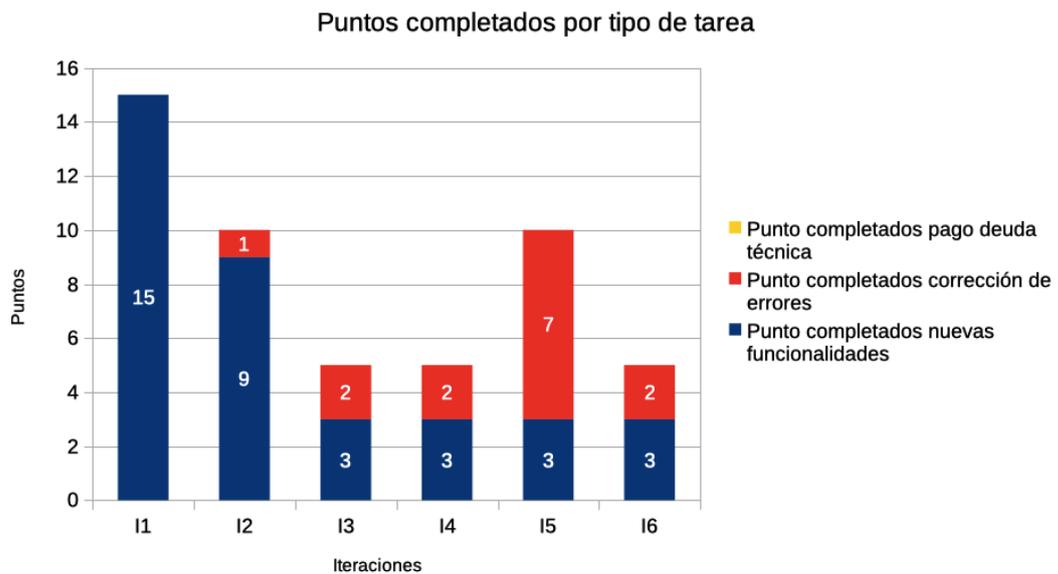


Figura 4.8: Tipo de tarea - Equipos que Entregan

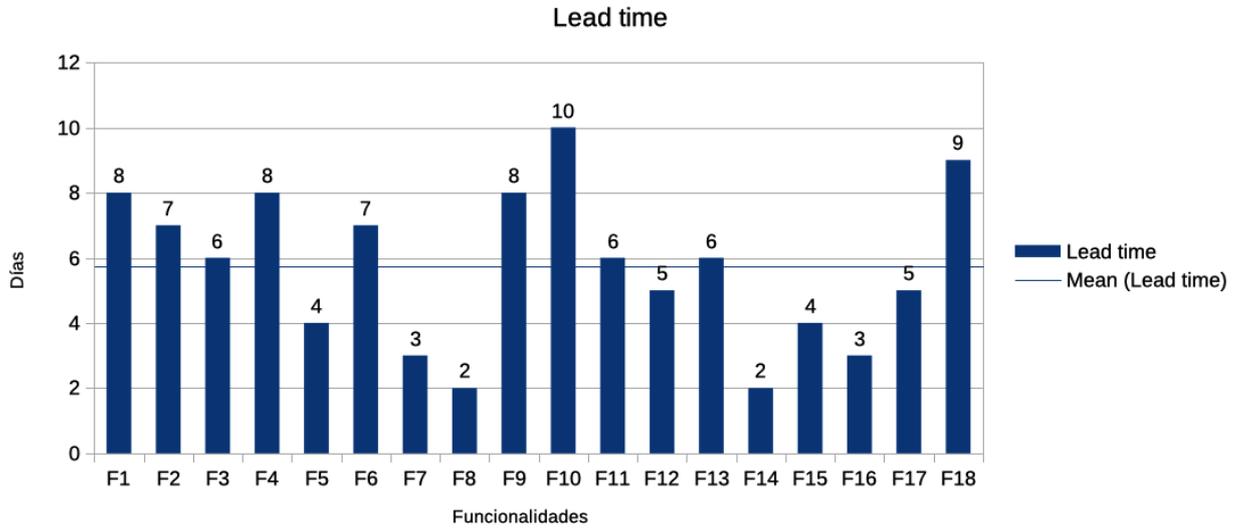


Figura 4.9: Lead Time - Equipos que Entregan

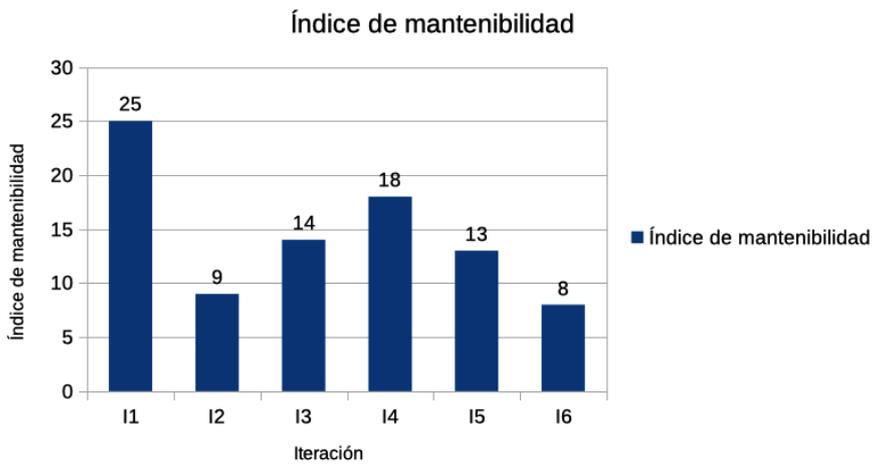


Figura 4.10: Mantenibilidad - Equipos que Entregan

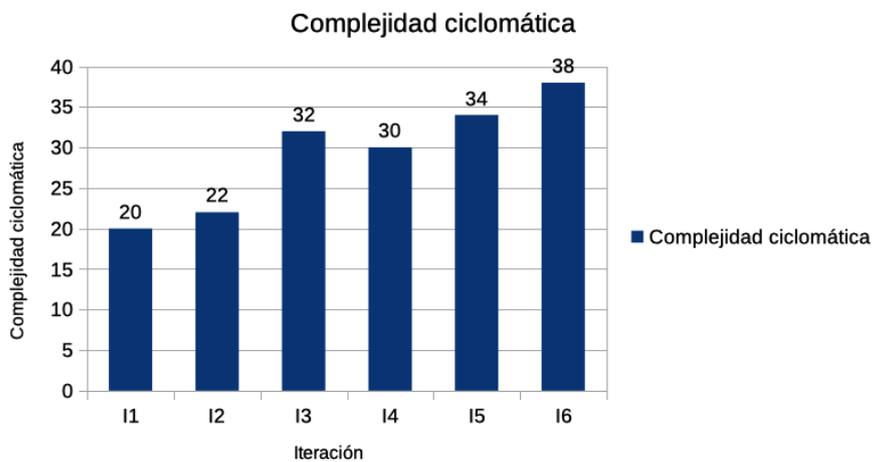


Figura 4.11: Complejidad ciclomática - Equipos que Entregan

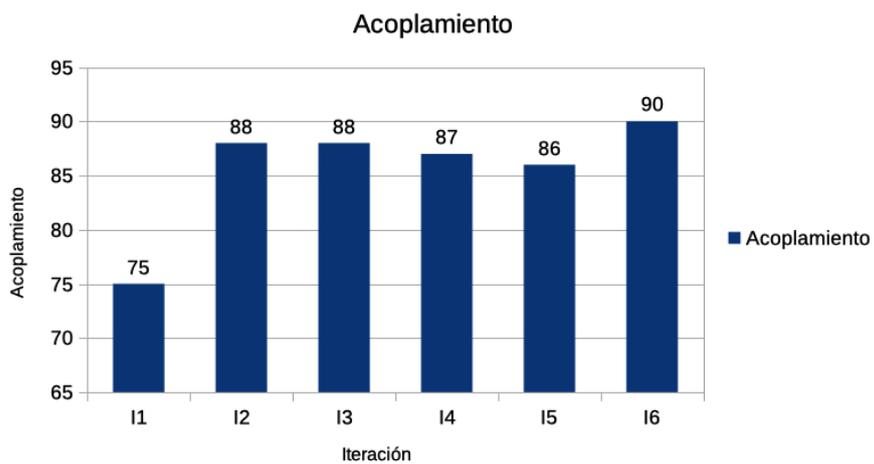


Figura 4.12: Acoplamiento - Equipos que Entregan

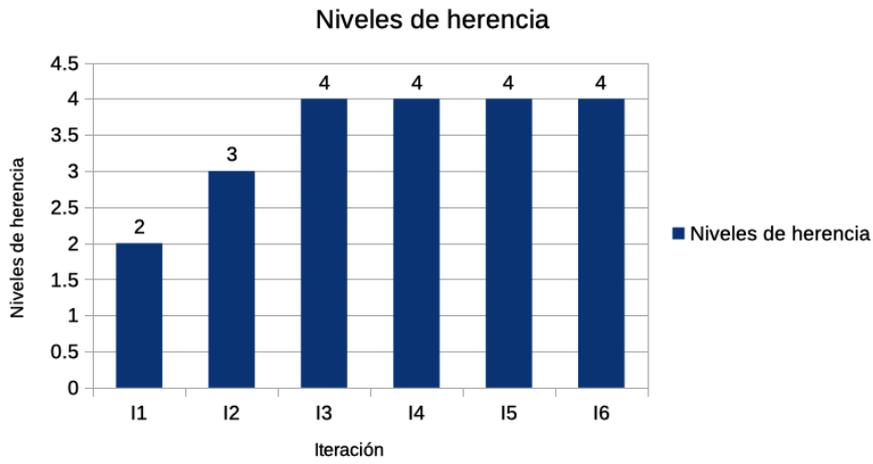


Figura 4.13: Niveles de herencia - Equipos que Entregan

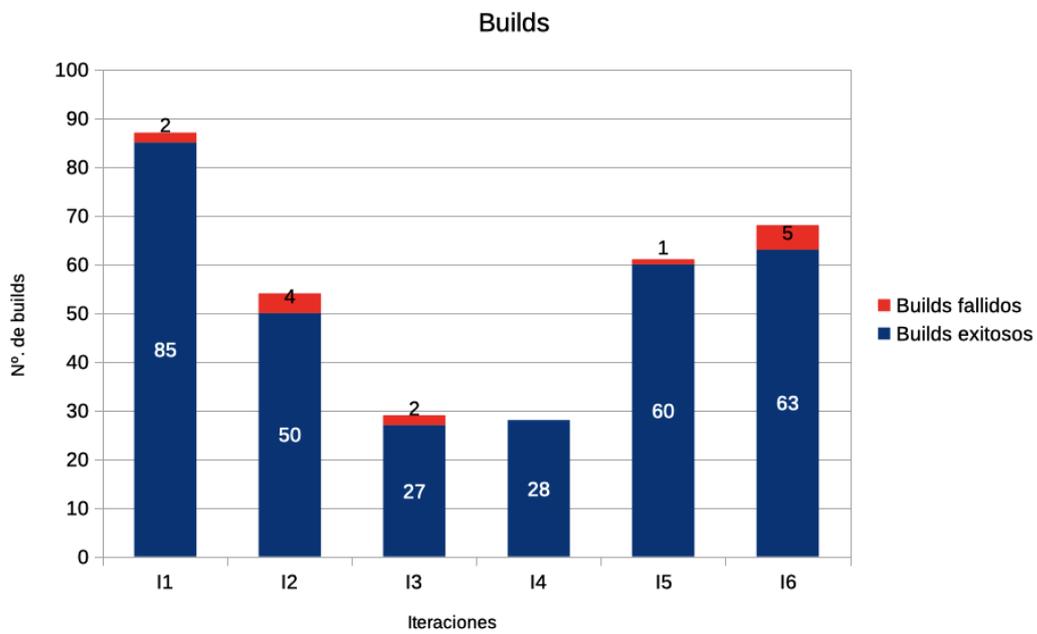


Figura 4.14: Builds - Equipos que Entregan

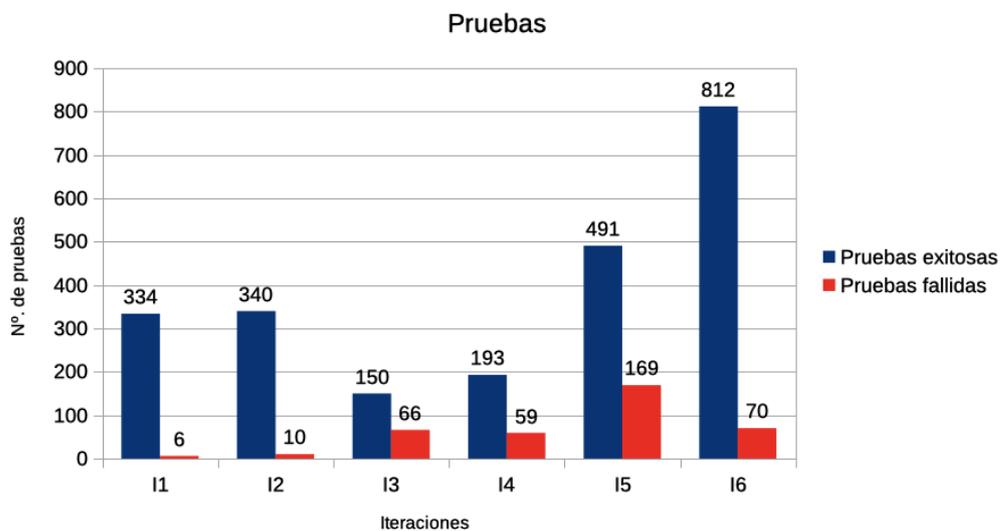


Figura 4.15: Pruebas - Equipos que Entregan

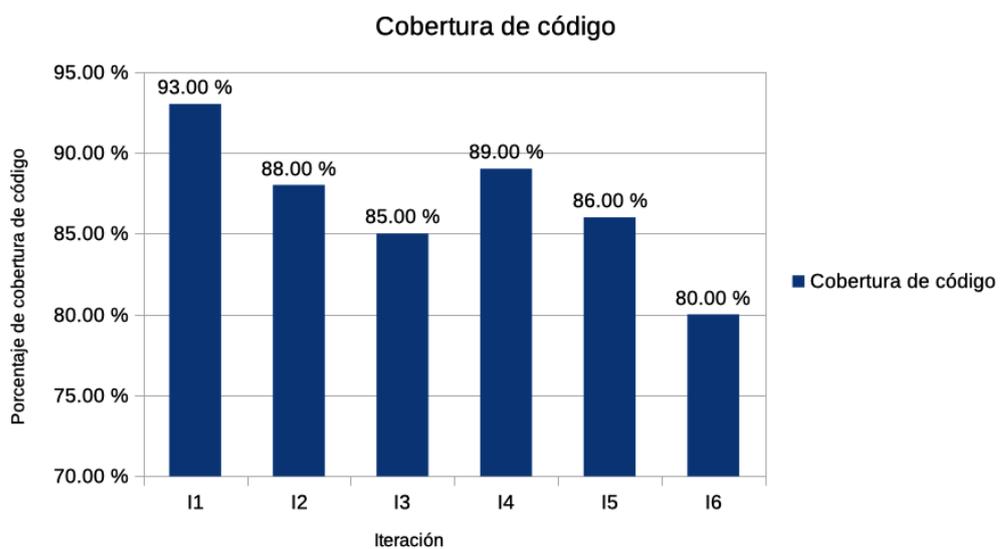


Figura 4.16: Cobertura de código - Equipos que Entregan

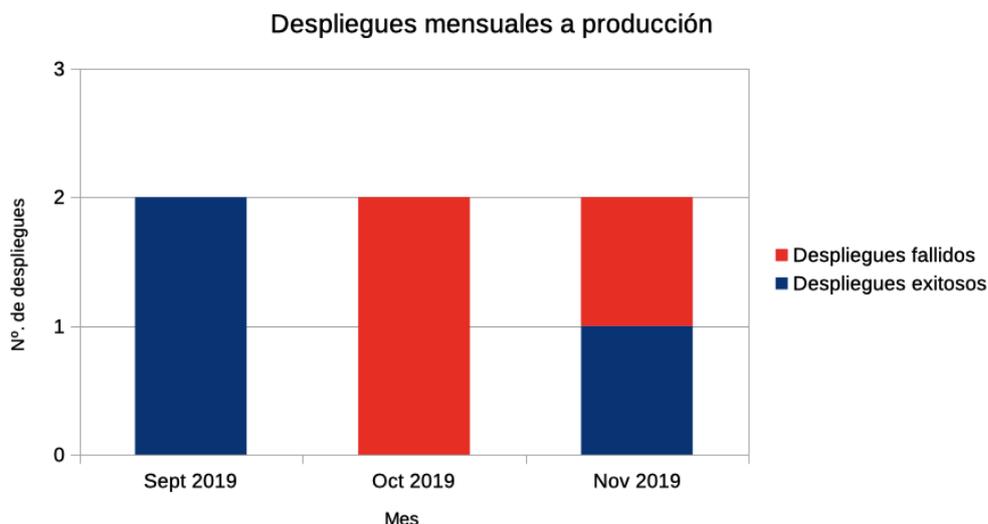


Figura 4.17: Despliegue a producción - Equipos que Entregan

el cliente. Hasta la sexta iteración se han completado 18 de 30 funcionalidades.

Con esto termina la sección de presentación de los resultados obtenidos con las técnicas de entrevistas, observación y mediciones a partir de los datos obtenidos de los sistemas de soporte al proyecto Partner Portal.

La siguiente sección tratará sobre el análisis-síntesis de los mismos.

4.3 ANÁLISIS-SÍNTESIS

Se seguirá el orden prescrito por Larsen y Shore (2019) para analizar los resultados obtenidos en la sección anterior. Es decir, primero se validarán las competencias de los Equipos Enfocados, luego de los Equipos que Entregan, Equipos Optimizados y finalmente Equipos Fortalecidos. La justificación de este orden proviene de la experiencia de los autores. Ellos han observado que los equipos de desarrollo tienden alcanzar fluidez en dicho orden.

Debido a que los equipos pueden lograr competencias de varias zonas en cualquier orden e incluso de forma simultánea, se considera que para que un equipo tenga fluidez en una zona, debe no solo ser fluido en esa zona, sino también en las zonas que la preceden.

Las competencias se pueden clasificar en función de las necesidades del negocio, trabajo en equipo y mejora continua. La mejora continua depende de la zona: para los Equipos

Enfocados se refiere a la mejora del trabajo en equipo, para los Equipos que Entregan es la excelencia técnica y para los Equipos Optimizados trata de la excelencia del producto.

Para realizar el diagnóstico de fluidez por competencias se va a utilizar la siguiente clasificación:

- No fluido. No cumple con la competencia.
- Iniciando camino a la fluidez. Es parcialmente competente, pero apenas inicia.
- Aproximándose a la fluidez. Es parcialmente competente, cerca de ser fluido.
- Fluido. Cumple con la competencia.

4.3.1 Equipos Enfocados

4.3.1.1 Necesidades del negocio

Las cuatro primeras competencias de los Equipos Enfocados evalúan su respuesta a las necesidades del negocio.

- **Cliente involucrado y empoderado**

En este caso, el cliente está trabajando cercanamente en el proyecto y está empoderado para definir los requerimientos y crear un plan en conjunto con el equipo. El cliente está disponible para resolver dudas y especificar la funcionalidad cuando el equipo de desarrollo lo requiere.

Por todo esto se concluye que el equipo es fluido en esta competencia. Existe una buena interacción entre el equipo de desarrollo y el cliente.

- **Fijación de prioridades**

El cliente está haciendo su trabajo respecto al establecimiento de prioridades para definir desde el punto de vista del negocio qué se va a construir y cuándo.

Ésto se pudo comprobar en la reuniones de planificación y en los datos que provee la herramienta de gestión de proyectos, donde se pudo confirmar que el 100 % de funcionalidades o historias de usuario por iteración han sido priorizadas.

Por lo tanto el equipo es fluido en la competencia de fijación de prioridades.

■ **Software que funciona y es valorado**

De parte del equipo se pudo observar que está demostrando progreso en el trabajo, aunque no al ritmo que se esperaba. Solamente en la primera iteración se cumplió con todas las funcionalidades estimadas. En las siguientes iteraciones hubo progreso, pero no al mismo nivel de la primera iteración.

La velocidad promedio del equipo, figura 4.3, página 69 ha ido disminuyendo en cada iteración, lo que significa que está produciendo menor valor conforme avanza el proyecto. Actualmente, en la sexta iteración, la velocidad promedio es 8 puntos por iteración. Hasta la tercera iteración la velocidad promedio era de 10 puntos y en la cuarta y quinta iteraciones era de 9 puntos.

A la sexta iteración se esperaba que el esfuerzo pendiente sea de 30 puntos. La realidad es que están pendientes 70 puntos (figura 3.9, página 40) lo que corresponde al 58 % del esfuerzo total pendiente estimado de todo el proyecto (figura 4.5, página 70). La tendencia indica que en la última iteración aun estarán pendientes entre 50 y 60 puntos si no hay cambios en la forma de trabajo (figura 3.10, página 40).

Debido a esto se concluye que en la competencia relacionada con «Software que funciona y que es valorado» por el cliente, apenas está iniciando su camino hacia la fluidez.

■ **Fijación de expectativas sobre trabajo sostenible**

Para que el equipo pueda responder a las necesidades del negocio de forma indefinida se requiere que la administración fije expectativas sobre el trabajo sostenible. La administración no está comprometida con esta competencia.

Esto se refleja en el hecho de que la calidad del código ha ido disminuyendo conforme avanza el proyecto. Se privilegia la creación de código que funcione, pero no necesariamente que sea de la calidad requerida para que sea más fácilmente mantenible en el futuro. El código de calidad en general requiere más tiempo, pero es una inversión para cuando se requiera realizar modificaciones futuras, que serán más fáciles. La dirección de la administración de la empresa ha sido la contraria: ahorrar tiempo al inicio creando código que simplemente funcione pero difícil de mantener

Tabla 4.20: Diagnóstico - Equipos Enfocados - Necesidades del negocio

ID	COMPETENCIA	DIAGNÓSTICO
EF1	Clientes involucrados y empoderados	Fluido
EF2	Fijación de prioridades	Fluido
EF3	Software que funciona y es valorado	Iniciando camino a la fluidez
EF4	Fijación de expectativas sobre trabajo sostenible	No fluido

y una vez que el proyecto se encuentra en marcha y se ha creado más código, éste es más difícil de mantener y actualizar, provocando retrasos.

Otro síntoma de esto es que no existe un rol que vele por el bienestar del equipo, es decir un rol similar a Scrum Master en la metodología Wave.

Por lo tanto el equipo no es fluido en esta competencia.

Aunque el equipo es fluido en dos competencias, «clientes involucrados y empoderados» y «fijación de prioridades», y ha empezado a trabajar en la fluidez de la competencia «software que funciona y es valorado», no demuestra fluidez en «fijación de expectativas sobre trabajo sostenible». Éste último comportamiento es el más difícil de superar porque depende de la cultura organizacional comprender la importancia de la calidad para dar valor al cliente desde el inicio y en forma continua. Si no es posible cambiar esta mentalidad de parte de la empresa, es difícil que el equipo pueda mejorar en la competencia «software que funciona y es valorado».

La tabla 4.20 contiene el resumen del diagnóstico de las competencias para cumplir con las necesidades del negocio para los Equipos Enfocados.

4.3.1.2 Trabajo efectivo como equipo

Los Equipos Enfocados deben trabajar de forma efectiva como equipo. Esto se logra con la auto-organización, el trabajo y responsabilidad en equipo y con el soporte de la administración respecto a la cultura de equipo.

- **Equipos auto-organizados**

El equipo se auto-organiza de forma parcial con la planificación de tareas por silos, no como un todo. Dentro del equipo de desarrollo existen sub-equipos definidos por sus roles claramente identificados: desarrolladores front-end y backend, testers, analistas del negocio. Cada uno de estos sub-equipos se organiza internamente para realizar su trabajo, que constituye un subproducto.

Adicionalmente, cuando existen épocas de mayor urgencia, no se respeta esta auto-organización por silos. El administrador del proyecto toma el control del equipo y asigna tareas individualmente para tratar de solventar las urgencias.

Por esto se puede concluir que la competencia en auto-organización se cumple parcialmente, pero apenas están iniciando.

■ **Trabajo y responsabilidad en equipo**

Al encontrarse el trabajo dividido en fases o etapas como en una línea de producción, la responsabilidad también se encuentra fragmentada para ejecutar las tareas específicas de cada rol y generar el subproducto que corresponde a cada uno, desentendiéndose del producto como un todo.

Si alguien requiere soporte, por ejemplo si el tester tiene trabajo pendiente acumulado, los otros roles no lo van a ayudar porque ese no es su trabajo. La responsabilidad del equipo debería ser que el software esté terminado como un todo y listo para ser usado en producción.

Para el equipo del proyecto Partner Portal en lo referente a la competencia trabajo responsabilidad de equipo, se puede concluir que apenas comienzan en el camino de la fluidez.

■ **Soporte de la administración respecto a la cultura de equipo**

La administración ha establecido la expectativa en cuanto a una forma de trabajo por roles. Si a esto sumamos el reconocimiento individual, también promovido por la dirección de la empresa y basado en el ingreso individual generado, todo va en contra de una cultura de trabajo en equipo, que es la herramienta para poder afrontar retos complejos como el desarrollo de software.

Tabla 4.21: Diagnóstico - Equipos Enfocados - Trabajo efectivo como equipo

ID	COMPETENCIA	DIAGNÓSTICO
EF5	Equipos auto-organizados	Iniciando camino a la fluidez
EF6	Trabajo y responsabilidad en equipo	Iniciando camino a la fluidez
EF7	Soporte de la administración respecto a la cultura de equipo	No fluido

Debido a que la administración da soporte exactamente a lo contrario que sugiere el modelo de Fluidez, es decir a una cultura de trabajo individual, se concluye que el equipo no tiene fluidez en esta competencia.

Dado que la empresa no da soporte respecto a una cultura de equipo, es difícil que se pueda mejorar la situación en las otras dos competencias, directamente relacionadas a la primera, en las el equipo demuestra que apenas está iniciando su camino hacia la fluidez, esto es, «Trabajo y responsabilidad en equipo» y «Equipos auto-organizados». La dirección de la empresa no otorga la confianza que el equipo requiere para desarrollarse.

La tabla 4.21 contiene el resumen del diagnóstico de las competencias de trabajo efectivo como equipo para los Equipos Enfocados.

4.3.1.3 Maestría del equipo

Las tres últimas competencias de los Equipos Enfocados propenden la mejora continua del equipo, tanto en la manera que percibe e identifica oportunidades de mejora en la forma en que trabaja, entendiendo cómo las relaciones en el equipo son importantes para facilitar las interacciones en el trabajo y cómo el espacio físico y virtual permiten un trabajo colaborativo.

- **Mejora de trabajo en equipo**

En el proyecto Partner Portal no siempre existe espacio para la reflexión sobre la forma de trabajo en equipo y cuando se crean planes de mejora no siempre se los ejecuta, como consecuencia de la presión para cumplir con los plazos establecidos.

El momento adecuado para esto son las reuniones de retrospectiva al finalizar cada iteración, pero debido a la falta de tiempo se las ha visto como reuniones que no agregan valor. Se prefiere usar ese tiempo en la ejecución de tareas consideradas como más productivas.

En consecuencia el equipo no es fluido en esta competencia.

■ **Mejora de relaciones en el equipo**

Los intentos de mejora de relaciones dentro del equipo tuvieron su espacio al inicio del proyecto, pero se han dejado de hacer, en gran medida debido a la presión existente para culminar el trabajo.

La moral del equipo es baja y no existen iniciativas dentro de la empresa para revertir esta situación. Debido a que el equipo por su cuenta sí demostró esta competencia se diagnostica que el equipo ha iniciado el camino a la fluidez, aunque no ha podido pasar de este punto.

■ **Mejora del entorno físico**

El equipo ha propuesto cambios en el entorno físico y ha intentado mejorarlos pero no cuenta con el respaldo de la organización. El equipo no tiene influencia sobre la administración de la empresa ni su soporte para sugerir e implementar las mejoras que puede identificar, entre ellas re-ubicar al equipo a un espacio en el que tenga mayor separación con otros equipos con los que no requiere interactuar.

Por esto se concluye que el equipo no es fluido en la mejora del entorno físico.

El equipo ha demostrado que puede iniciar su camino a la fluidez en la competencia «Mejora de relaciones en el equipo», que además es la única en la que no tiene injerencia la administración de la empresa. En las otras dos competencias, «Mejora de trabajo en equipo» y «Mejora de entorno físico», no tiene otra alternativa que obedecer las indicaciones del administrador del proyecto, que no cree que las reuniones de retrospectiva o el entorno físico sean importantes para lograr la maestría del equipo. La empresa demuestra una mentalidad burocrática, donde los líderes son los que tienen razón y los subalternos deben obedecer.

Tabla 4.22: Diagnóstico - Equipos Enfocados - Maestría del equipo

ID	COMPETENCIA	DIAGNÓSTICO
EF8	Mejora de trabajo en equipo	No fluido
EF9	Mejora de relaciones en el equipo	Iniciando camino a la fluidez
EF10	Mejora de entorno físico	No fluido

La tabla 4.22 contiene el resumen del diagnóstico de las competencias para alcanzar la maestría del equipo para los Equipos Enfocados.

4.3.1.4 *Diagnóstico*

Si consideramos las 4 primeras competencias, referentes al cumplimiento de las necesidades del negocio, vemos que el equipo ya ha iniciado el camino a la fluidez, pero aun le falta desarrollar estas competencias para considerar que está próximo a la fluidez.

En las siguientes tres competencias, relacionadas con el trabajo efectivo como equipo, apenas inicia su camino hacia la fluidez

Desde la perspectiva de la mejora del trabajo en equipo, correspondiente con las 3 últimas competencias, se puede concluir que el equipo apenas inicia su camino a la fluidez en una de ellas: EF9 - mejora de relaciones en el equipo. En las otras dos no lo ha iniciado.

Considerando en forma global las competencias de los Equipos Enfocados, que se resumen en alcanzar los fundamentos Ágiles, el equipo apenas inicia su camino hacia la fluidez en esta zona.

Esta situación es responsabilidad directa de la administración de la empresa que no ha otorgado confianza ni ha empoderado al equipo para que sea éste el que dirija su destino. Si no existe empoderamiento no se puede exigir responsabilidad al equipo.

La figura 4.18 contiene el diagnóstico para las competencias de Equipos Enfocados.

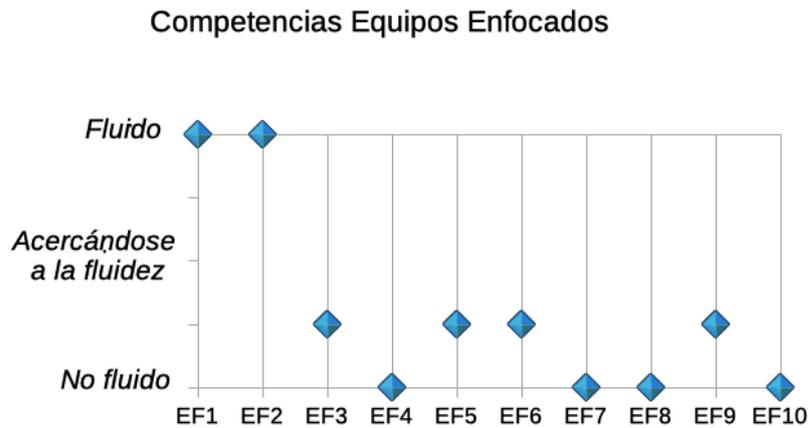


Figura 4.18: Diagnóstico Equipos Enfocados

4.3.2 Equipos que Entregan

4.3.2.1 Necesidades del negocio

Los Equipos que Entregan deben responder a las necesidades del negocio. Esto se refleja en las 4 primeras competencias de los Equipos que Entregan.

■ Integración continua

La primera competencia es la demostración del equipo de su capacidad de integrar continuamente los cambios a un ambiente similar a producción. Se debe valorar los builds, pruebas y cobertura de código.

El número de builds es una muestra de qué tan continuamente el equipo está integrando sus cambios. La integración continua se refiere a que el trabajo del equipo se va completando mediante incrementos pequeños y continuos. La integración de código es mucho más fácil cuando se trabaja con pequeños incrementos. Lo ideal es integrar el código trabajado al menos una vez cada hora.

En la figura 4.14 de la página 75 se puede ver que se realizaron un mínimo de 27 builds exitosos (iteración 3) y un máximo de 85 en la (iteración 1). Debido a que la duración de cada iteración es de 10 días, se hicieron entre 2.7 y 8.5 builds diarios al ambiente de pruebas. En la primera iteración se logró la mayor cantidad de builds exitosos por desarrollador con 2.1 (el equipo tiene 4 desarrolladores). En la tercera y

Tabla 4.23: Builds exitosos diarios y por desarrollador

ITERACIÓN	BUILDS EXITOSOS DIARIOS	BUILDS EXITOSOS DIARIOS POR DESARROLLADOR
I1	8.5	2.1
I2	5.0	1.3
I3	2.7	0.7
I4	2.8	0.7
I5	6.0	1.5
I6	6.3	1.6

cuarta iteraciones se alcanzó el número más bajo 0.7. En las dos últimas iteraciones, la 5 y la 6, el equipo está tratando de lograr el rendimiento que alcanzó en la primera con 1.5 y 1.6 integraciones por desarrollador respectivamente. Ver tabla 4.23.

La proporción de builds fallidos es baja, se puede ver que en las iteraciones 2 y 6 se presentaron el mayor número de builds fallidos, un 7% del total de builds en cada una de esas iteraciones (figura 4.14, página 75). Esto nos indica que el trabajo de integración del código es bueno, el porcentaje de errores en builds es bajo. El objetivo es llegar a cero.

En la figura 4.15, de la página 76 se puede ver un incremento de las pruebas automáticas fallidas; sin embargo, el incremento de pruebas exitosas es mayor. Esto nos indica que conforme se avanza en el proyecto, se han creado más pruebas automáticas para garantizar su calidad. Otra causa del aumento de pruebas fallidas puede ser un mayor acoplamiento en el código, lo que implica que cambios en una parte del código produzcan resultados no deseados en otras funcionalidades. Esta es una característica de un código de baja calidad (ver figura 4.12, página 74).

Existe una tendencia de disminución de cobertura de código (ver figura 4.16, página 76). Ésta ha disminuido desde el 93% que se observa en la primera iteración hasta el 80% en la sexta iteración. Esto indica que conforme avanza el proyecto no se crean las suficientes pruebas para que alcancen a cubrir todo el código que se va creando.

Mientras exista mayor cobertura de código es mejor, incluso algunos autores consideran que se debe llegar al 100 % de cobertura; sin embargo, los rangos hasta ahora son aceptables, pero es necesario revertir la tendencia decreciente.

En el gráfico de puntos completados por tipo de tarea (figura 4.8, página 72), se puede ver que solamente en la primera iteración se trabajó exclusivamente en nuevas funcionalidades (15 puntos). En las siguientes iteraciones se ha trabajado en corrección de errores llegando al extremo de 7 puntos para este tipo de tareas y solo 3 para nuevas funcionalidades en la iteración 4. También se resalta que no se trabajó en ningún tipo de tarea relacionada con pago de deuda técnica.

Considerando que es la primera experiencia de todos los integrantes del equipo en proyectos que usan Metodologías Ágiles, el número de integraciones diarias por desarrollador es aceptable. El porcentaje de builds fallidos es bajo, lo que indica que las integraciones de código se están probando previamente a desplegar los cambios en el ambiente de pruebas. La cobertura de código ha bajado debido a la presión por la terminación en las fechas de entrega, es decir no se están creando las pruebas necesarias para las nuevas funcionalidades. Esto puede traer consecuencias porque el equipo está dejando de contar con el indicador de que los cambios que se están aplicando en el código no afecten las funcionalidades existentes (pruebas de regresión). Esto se demuestra también en el tipo de tareas que se están completando, con un aumento de las tareas relacionadas con la corrección de errores, lo que indica re-trabajos en funcionalidades que han dejado de funcionar. Tampoco se puede ver que se haya trabajado en tareas de pago de deuda técnica, que ayudarían a mejorar la calidad de código para que sea más mantenible.

Con esta información se puede concluir que en la competencia de «Integración Continua» el equipo se está aproximando a la fluidez. Existen despliegues a un ambiente semejante al de producción al menos de forma diaria en el que se integra el trabajo del equipo, pero algunos de ellos han fallado y la cobertura de código ha disminuido.

■ **Disponibilidad para liberar a producción**

Para responder a las necesidades del negocio el representante del cliente debe tener

la opción de liberar al ambiente de producción cuando él lo necesite (figura 4.17, página 77).

La herramienta de despliegue registra que el representante del cliente fue quien ejecutó las liberaciones a producción, lo hizo al final de la iteración, en acuerdo con el equipo. Las dos primeras iteraciones culminaron con una liberación exitosa a producción cada una. Las iteraciones del mes de octubre no tuvieron igual suerte, ambas liberaciones fueron rechazadas por el usuario. En noviembre una liberación fue exitosa y la otra rechazada.

De las liberaciones que se hicieron en común acuerdo con el equipo, en tres de ellas el cliente no estuvo conforme con los últimos cambios, debiendo rechazar el despliegue, y en tres ocasiones el cliente aceptó los cambios. Los rechazos fueron causados por errores detectados por el cliente en sus pruebas de aceptación.

Debido a esto el equipo no demuestra fluidez en esta competencia ya que no ha podido darle la opción al cliente de desplegar los últimos cambios cuando él lo requiera.

■ **Estimaciones de entrega realistas**

Otra característica de los equipos que responden a las necesidades del negocio es realizar estimaciones realistas.

En el caso de nuevas funcionalidades (figura 4.6, página 71) se puede ver que en la tercera iteración se cumplió sólo con el 30 % de la estimación original. Sólo en a primera iteración se cumplió con la estimación. Desde entonces no se pudo cumplir en ninguna iteración. También se puede observar una tendencia de disminución de las tareas relacionadas con nuevas funcionalidades. El promedio de cumplimiento en nuevas funcionalidades en todas las iteraciones es del 60 %.

Si consideramos las estimaciones de tareas de tipo corrección de errores (figura 4.7, página 72), el cumplimiento es aun menor, en dos ocasiones solo es del 20 % de la estimación inicial, siendo el promedio 42 %. Existe una tendencia de aumento de tareas relacionadas con la corrección de errores.

Por lo tanto el equipo no es fluido en esta competencia.

Tabla 4.24: Diagnóstico - Equipos que Entregan - Necesidades del negocio

ID	COMPETENCIA	DIAGNÓSTICO
EE1	Integración continua	Acercándose a la fluidez
EE2	Disponibilidad para liberar a producción	No fluido
EE3	Estimaciones de entrega realistas	No fluido
EE4	Soporte de la administración sobre trabajo reflexivo	No fluido

▪ Soporte de la administración sobre trabajo reflexivo

El equipo requiere contar con el soporte de la administración para que pueda trabajar sosteniblemente.

Para la administración de la empresa la prioridad es el cumplimiento de los plazos de entrega. No cree que el cliente pague lo que se requiera para producir un código de calidad que va tomar más tiempo. Al parecer no está consciente que el requisito principal para poder cumplir con los plazos de entrega es contar con un código de calidad, es decir mantenible, y de esta forma ser capaces de realizar estimaciones realistas.

En conclusión no existe fluidez en esta competencia.

Salvo el caso de la primera competencia, «Integración Continua», en la que el equipo se aproxima a la fluidez, el equipo no ha demostrado fluidez en las competencias necesarias para responder a las necesidades del negocio.

La tabla 4.24 resume el diagnóstico de las competencias que soportan la respuesta a las necesidades del negocio.

4.3.2.2 Trabajo efectivo como equipo

Para que se considere que el equipo realiza un trabajo efectivo como equipo, se requiere que exista propiedad y responsabilidad colectiva sobre el código y que el equipo sea auto-suficiente y multi-funcional.

■ **Propiedad y responsabilidad colectiva**

No solamente existen silos por roles dentro del equipo, sino también una mentalidad de propiedad individual del código. Los desarrolladores dentro de cada silo, por ejemplo «back-end» y «front-end», prefieren no modificar el código que no han desarrollado. Solo lo hacen cuando no hay otra alternativa. Esto puede deberse a una ausencia de confianza entre los desarrolladores sobre la calidad de código que producen.

Al tratarse de una competencia que depende directamente de la voluntad del equipo, se puede trabajar en ella y lograr buenos resultados en el corto plazo, pero es necesario cambiar esa mentalidad con ejemplos prácticos de los beneficios.

Una posible solución puede ser implementar la práctica de XP conocida como «pair programming» en la que dos desarrolladores trabajan en conjunto en una misma tarea. Esta práctica fomenta el trabajo en equipo, se comparten conocimientos tanto técnicos como del negocio. Se busca que las capacidades técnicas y del negocio de todo el equipo mejoren, así como también la calidad del código.

Actualmente el equipo no es fluido en esta competencia.

■ **Equipos auto-suficientes y multi-funcionales**

Para que el equipo sea auto-suficiente debe contar con todas las habilidades o funciones para desarrollar su trabajo. Debido a que la mayoría de los desarrolladores de la empresa se han especializado en funciones específicas, como por ejemplo desarrolladores «front-end», «back-end», «operaciones», se debe contar con al menos uno de cada función. Esto no ha ocurrido en el proyecto Partner Portal, que no ha contado con un desarrollador «front-end» permanente.

Esto se puede solucionar con el fomento y capacitación de desarrolladores «full-stack» que puedan dar soporte a los equipos de la empresa de acuerdo a las necesidades.

El equipo del proyecto Partner Portal no ha demostrado fluidez en esta competencia, pero se puede revertir si se invierte en el desarrollo de capacidades en el equipo.

Tabla 4.25: Diagnóstico - Equipos que Entregan - Trabajo efectivo como equipo

ID	COMPETENCIA	DIAGNÓSTICO
EE5	Propiedad y responsabilidad colectiva	No fluido
EE6	Equipos auto-suficientes y multifuncionales	No fluido

En resumen, el equipo no es fluido en ninguna de las competencias relacionadas con el trabajo efectivo como equipo (tabla 4.25).

Dadas las anteriores razones se puede concluir que el equipo no trabaja como tal de una forma efectiva.

4.3.2.3 *Excelencia técnica*

La última categoría de competencias de los Equipos que Entregan es la excelencia técnica. A continuación se analizan los resultados de las competencias de esta categoría.

- **Mejora continua, simplicidad del código**

Las figuras 4.11: complejidad ciclomática, página 74; 4.12: acoplamiento, página 74 y 4.13: niveles de herencia, página 75 indican que la calidad del código ha ido disminuyendo y la complejidad ha aumentado conforme avanza el proyecto.

Esto se confirma con los valores decrecientes del índice de mantenibilidad, presentado en la figura 4.10, página 73.

Esto se puede revertir mediante la propiedad y responsabilidad colectiva del código, y la capacitación técnica del equipo, ya sea proactivamente o impulsada por la empresa. Pero por ahora el equipo no tiene fluidez en esta competencia.

- **Entrega continua, builds automáticos**

Como parte normal del proceso existen builds fallidos y pruebas fallidas, las que son resueltas antes de que los cambios sean desplegados al ambiente de pruebas o de producción. Sin embargo, debido a que el proceso incluye la elaboración de documentación manual, los despliegues toman alrededor de 2 horas. Para que pueda

considerarse entrega continua los despliegues no deben implicar un trabajo manual que tome más de 10 minutos.

La solución es automatizar la elaboración de los documentos o mejor aun enviar notificaciones con toda la información necesaria requerida por el cliente cuando los despliegues se hayan efectuado.

Aunque los builds y pruebas son automáticos, existe integración de código en rangos aceptables, la entrega no se la hace de una forma continua. Por lo tanto el equipo no ha demostrado fluidez en esta competencia por el tiempo que tardan los despliegues.

■ **Cambios incrementales, trabajo de calidad, pruebas automáticas**

Otra requisito importante de la excelencia técnica es el trabajo incremental de calidad con pruebas automáticas que lo garanticen.

Se pudo validar que se realizan cambios incrementales y pruebas automáticas, pero estas pruebas no alcanzan a cubrir todos los casos y escenarios de las funcionalidades que se han ido desarrollando. Esto se puede ratificar con los despliegues a producción rechazados por el cliente debido a errores encontrados (figura: 4.17, página 77).

Por estas razones el equipo no demuestra fluidez en esta competencia.

■ **Mentalidad de aprendizaje continuo**

El último requisito de excelencia técnica es tener una mentalidad de aprendizaje continuo. No se pudo observar este requisito de forma general en el grupo, solo unos cuantos esfuerzos individuales.

La empresa inició un programa de desarrollo de capacidades pero no ha tenido el éxito esperado, en parte debido a la falta de seguimiento continuo de los acuerdos al inicio del programa de parte de los coaches. Los coaches por su lado no han demostrado la motivación ni la pasión para mejorar sus capacidades.

Se concluye que el equipo no tiene fluidez en aprendizaje continuo.

El equipo no ha demostrado fluidez en todas las competencias de la categoría excelencia técnica. La responsabilidad es tanto de la dirección de la empresa, que debería fomentar

Tabla 4.26: Diagnóstico - Equipos que Entregan - Excelencia técnica

ID	COMPETENCIA	DIAGNÓSTICO
EE7	Mejora continua, simplicidad	No fluido
EE8	Entrega continua, builds automáticos	No fluido
EE9	Cambios incrementales, trabajo de calidad, pruebas automáticas	No fluido
EE10	Mentalidad de aprendizaje continuo	No fluido

de una forma más efectiva una cultura apasionada por la excelencia técnica, como individual de cada miembro del equipo.

La tabla 4.26 resume el análisis realizado en la categoría excelencia técnica.

4.3.2.4 *Diagnóstico*

La ausencia de excelencia técnica y de trabajo en equipo nos indica que el equipo del proyecto Partner Portal no se encuentra en la zona de los Equipos que Entregan. Sin embargo, su primer paso en esta zona se ha dado en la competencia «Integración Continua».

Debido a que el cliente requiere cambios continuos y una respuesta rápida de la empresa, es un requisito del equipo formar parte de esta zona. No es suficiente con que cumpla con los fundamentos ágiles, que aun debe desarrollarlos, también requiere la sustentabilidad Ágil que provee esta zona.

Para poder alcanzar la excelencia técnica se requiere contar con personal apasionado y motivado por el aprendizaje continuo. Si la empresa no es capaz de desarrollar esta competencia, el equipo de desarrollo puede indicarle el camino.

La principal obligación de la empresa si desea atender a las necesidades del cliente debe ser fomentar, comenzando desde ella, una mentalidad de trabajo en equipo. Todo reconocimiento debe hacerse a los equipos de los proyectos en base a sus logros, no rendimientos económicos. Los reconocimientos individuales no hacen otra cosa que fomentar el individualismo.

A través de la excelencia técnica y de una mentalidad de trabajo en equipo se podrá responder a las necesidades del negocio con la calidad y en el tiempo que requiera el

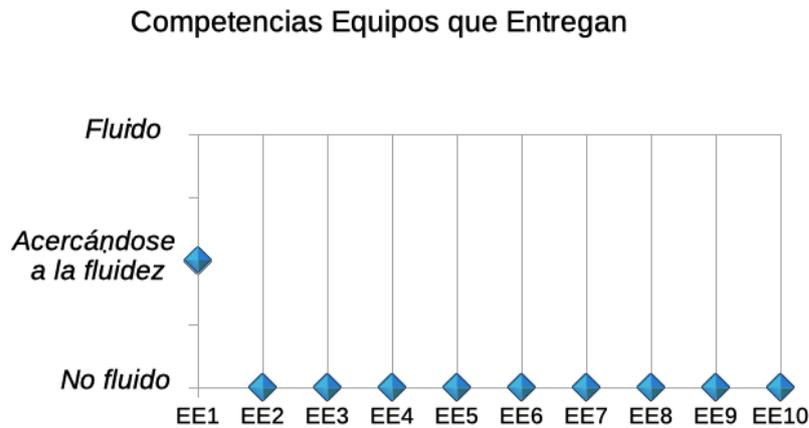


Figura 4.19: Diagnóstico Equipos que Entregan

cliente.

La figura 4.19 contiene el diagnóstico para las competencias de Equipos que Entregan. Se puede ver que únicamente en la primera competencia se acerca a la fluidez, en todas las demás no es fluido.

4.3.3 Equipos Optimizados

4.3.3.1 Necesidades del negocio

Los Equipos Optimizados atienden las necesidades del negocio haciéndose responsables de la dirección del producto. Toman decisiones en base a la retro-alimentación del mercado.

- **Responsabilidades del negocio**

La dirección del negocio es una responsabilidad exclusiva del cliente, por lo que el equipo no tiene responsabilidades directas en este tipo de decisiones. No utiliza métricas del negocio para formular sus planes y medir su progreso.

- **Retro-alimentación del mercado**

El equipo no busca retro-alimentación del mercado para guiar la dirección del producto.

Tabla 4.27: Diagnóstico - Equipos Optimizados - Necesidades del negocio

ID	COMPETENCIA	DIAGNÓSTICO
EO1	Responsabilidades del negocio	No fluido
EO2	Retro-alimentación del mercado	No fluido

El ámbito de trabajo del equipo es exclusivamente técnico. No es el caso de una «start-up» que requiere conocer qué debe entregar al mercado. Su trabajo se enmarca en el «cómo», la definición del «qué» la da el cliente.

Los fundadores de la empresa provienen de una tradición de agencias de publicidad y marketing. Su conocimiento y experiencia podría apalancar nuevas líneas de negocio, por ejemplo de asesoría a agencias de publicidad o a clientes, sobre mejores prácticas o dirección de campañas de publicidad. De esta forma complementarían la línea de soporte técnico para implementaciones de software. Las líneas técnicas y de asesoría en el negocio podrían atraer más clientes a la empresa.

La tabla 4.27 resume el diagnóstico en esta categoría.

4.3.3.2 Trabajo efectivo como equipo

Para que los Equipos Optimizados trabajen de forma efectiva como equipo requieren tener la capacidad de guiar el desarrollo de productos. Deben conocer qué requiere el mercado.

- **Empoderamiento sobre la dirección del producto**

El equipo no está empoderado en esta competencia. La responsabilidad de la dirección del producto es del cliente.

- **Delegación de parte de la administración**

Debido a que el rol del equipo es la implementación de las funcionalidades solicitadas por el cliente, la administración de la empresa no ha delegado la responsabilidad sobre la dirección del producto al equipo.

Tabla 4.28: Diagnóstico - Equipos Optimizados - Trabajo efectivo como equipo

ID	COMPETENCIA	DIAGNÓSTICO
EO3	Empoderamiento sobre la dirección del producto	No fluido
EO4	Delegación de parte de la administración sobre la dirección del producto	No fluido
EO5	Soporte de la administración respecto a la incorporación de expertos del negocio al equipo	No fluido

▪ **Soporte de la administración respecto a la incorporación de expertos del negocio al equipo**

El equipo no cuenta con expertos del negocio. Éstos están del lado del cliente, en este caso expertos en el ámbito de la educación superior. No forman parte del equipo.

Para que los Equipos Optimizados trabajen de forma efectiva como equipo requieren que la administración de la empresa les delegue las decisiones sobre el producto. Para esto necesitan contar con expertos del negocio de forma permanente, lo que requiere cambios en la estructura organizacional de la empresa. Una vez empoderado el equipo debe tomar decisiones sobre la dirección del producto.

Este tipo de competencias están fuera del alcance de este proyecto y de este equipo. El ámbito en el que se desenvuelven actualmente no requiere el desarrollo de las mismas.

Parte de los cambios organizacionales requeridos en esta zona buscan eliminar el lastre causado por una excesiva carga de labor administrativa no productiva. La proporción empleados administradores de 3:1 es alta.

El resumen del diagnóstico de las competencias de esta categoría se puede observar en la tabla 4.28.

4.3.3.3 *Excelencia de producto*

Esta dimensión requiere cambios en la estructura organizacional para incorporar expertos del negocio de forma permanente al equipo para que con ellos se busque la excelencia del producto. Este tipo de cambios en la estructura organizacional no se ha observado en la empresa Hyna.

- **Ejecución de experimentos**

El equipo no ha ejecutado experimentos para probar hipótesis. Los despliegues a producción no han sido habilitados al usuario final, por lo que el cliente no ha podido validarlos en vivo.

- **Dirección del producto**

Esta métrica busca conocer si el producto está siendo aceptado o rechazado por el usuario final. No ha sido posible validar si las funcionalidades son de utilidad para él.

- **Análisis de cohorte**

Esta capacidad busca medir la tasa de conversión de usuarios finales y el uso de las funcionalidades en producción. No se ha ejecutado este tipo de análisis.

Esta dimensión requiere que el producto se haya desplegado y habilitado en producción. El cliente ha tomado la decisión de no hacerlo porque considera que las funcionalidades no se han completado.

En gran medida esto se debe a que el equipo no ha sido capaz de entregar las funcionalidades con la calidad y en los tiempos deseados al cliente, quien considera que el producto no está listo para ser usado por el usuario final.

La tabla 4.29 resume el diagnóstico de las competencias de la dimensión «excelencia de producto». Se puede observar que el equipo de proyecto Partner Portal no ha demostrado fluidez en ninguna de ellas.

Tabla 4.29: Diagnóstico - Equipos Optimizados - Excelencia de producto

ID	COMPETENCIA	DIAGNÓSTICO
EO6	Ejecución de experimentos	No fluido
EO7	Dirección del producto	No fluido
EO8	Análisis de cohorte	No fluido

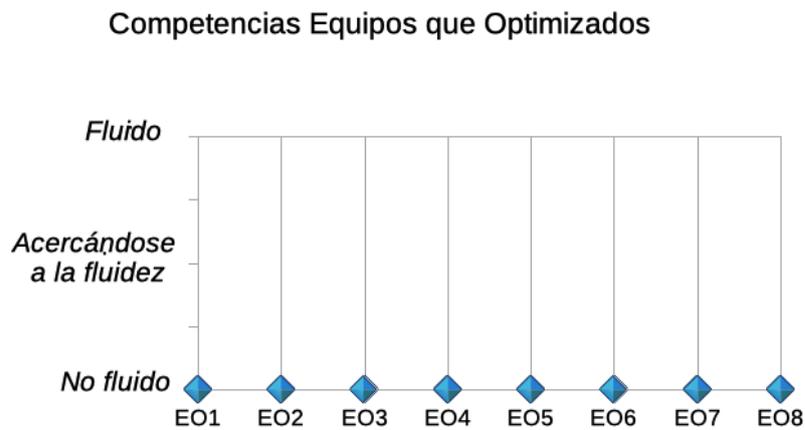


Figura 4.20: Diagnóstico Equipos Optimizados

4.3.3.4 Diagnóstico

El proyecto Partner Portal como ha sido concebido posiblemente no requiere el desarrollo de las competencias de Equipos Optimizados. Su rol es ser ejecutores, no asesores y peor aun responsables de la dirección del producto.

La empresa no debería descartar la creación proactiva de productos y servicios. Su tamaño la podría convertir en una buena candidata para emular los objetivos de una startup. Así podrían dirigir a equipos que sean capaces de conocer qué necesita el mercado. Pero antes de esto, debe desarrollar las competencias de las dos zonas previas: Equipos que Entregan y Equipos Optimizados.

La figura 4.20 contiene el diagnóstico para las competencias de Equipos Optimizados. El equipo no tiene fluidez en ninguna de ellas.

Tabla 4.30: Diagnóstico - Equipos Fortalecidos

COMPETENCIA	DIAGNÓSTICO
Descripción de trabajo en el entorno	No fluido
Involucramiento temprano	No fluido
Actividades multi-equipos	No fluido
Contribución entre equipos	No fluido
Polinizaciones cruzadas	No fluido

4.3.4 Equipos Fortalecidos

Los Equipos Fortalecidos representan el futuro de Agile. No se conoce qué forma tendrá este futuro y por esto sus características no han sido clasificadas de forma similar a como se lo hizo en las zonas previas: Equipos Enfocados, Equipos que Entregan y Equipos Optimizados.

Los Equipos Fortalecidos requieren cambios en la cultura organizacional. Requieren que el equipo conozca cuál es su posición dentro de la estrategia de la empresa, que trabaje proactivamente de forma temprana en las actividades enmarcadas en esta estrategia.

Actualmente estas decisiones son competencia exclusiva del fundador de Hyna y de sus colaboradores más cercanos.

No se ha podido ver que exista contribución entre equipos o polinizaciones cruzadas. Lo único que se pudo observar es el soporte de manera informal e individual de desarrolladores de otros equipos al equipo del proyecto Partner Portal cuando la carga de trabajo lo permitía.

Si la empresa es capaz de impulsar a sus equipos para que alcancen la zona de Equipos Optimizados, por qué no pensar que éstos sean capaces también de ser parte del grupo de las empresas líderes que aplican y crean nuevas formas de trabajo y representan el futuro de Agile.

El resumen del diagnóstico de las características de los Equipos Fortalecidos se encuentra en la tabla 4.30. El equipo no ha demostrado fluidez en ninguna de ellas.

Tabla 4.31: Diagnóstico por dimensiones - Negocio

ID	COMPETENCIA	ZONA	PUNTAJE
EF1	Clientes involucrados y empoderados	E. Enfocados	1.00
EF2	Fijación de prioridades	E. Enfocados	1.00
EF3	Software que funciona y es valorado	E. Enfocados	0.25
EF4	Fijación de expectativas sobre trabajo sostenible	E. Enfocados	0.00
EE1	Integración continua	E. que Entregan	0.50
EE2	Disponibilidad para liberar a producción	E. que Entregan	0.00
EE3	Estimaciones de entrega realistas	E. que Entregan	0.00
EE4	Soporte de la administración sobre trabajo reflexivo	E. que Entregan	0.0
EO1	Responsabilidades del negocio	E. Optimizados	0.00
EO2	Retro-alimentación del mercado	E. Optimizados	0.00
TOTAL			2.75

4.3.5 Diagnóstico por dimensiones

Las tablas 4.31, 4.32 y 4.33 contienen los puntajes alcanzados en cada competencia dentro de las dimensiones negocio, equipo y mejora, respectivamente, a través de todas las zonas: Equipos Enfocados, Equipos que Entregan y Equipos Optimizados.

Se considera el puntaje máximo de 1 si es fluido en la competencia, 0 si no es fluido, 0.25 si está iniciando su camino a la fluidez y 0.5 si se está acercando a la fluidez. Se ha normalizado los puntajes de la dimensión equipos, que tiene solo 8 competencias, para que todas las dimensiones logren el puntaje máximo de 10 en el caso que se alcance la fluidez en todas las competencias de la zona de Equipos Optimizados.

Si se consideran los puntajes de las tres dimensiones que se evalúan transversalmente en las zonas de Equipos Enfocados, Equipos que Entregan y Equipos Optimizados: negocio, equipo y mejora continua, se obtiene el gráfico de radar de la figura 4.21: Diagnóstico por dimensiones, en la página 102.

Tabla 4.32: Diagnóstico por dimensiones - Equipo

ID	COMPETENCIA	ZONA	PUNTAJE
EF5	Equipos auto-organizados	E. Enfocados	0.31
EF6	Trabajo y responsabilidad en equipo	E. Enfocados	0.31
EF7	Soporte de la administración respecto a la cultura de equipo	E. Enfocados	0.00
EE5	Propiedad y responsabilidad colectiva	E. que Entregan	0.00
EE6	Equipos auto-suficientes y multifuncionales	E. que Entregan	0.00
EO3	Empoderamiento sobre la dirección del producto	E. Optimizados	0.00
EO4	Delegación de la administración	E. Optimizados	0.00
EO5	Soporte de la administración sobre incorporación expertos del negocio	E. Optimizados	0.00
TOTAL			0.62

Tabla 4.33: Diagnóstico por dimensiones - Mejora

ID	COMPETENCIA	ZONA	PUNTAJE
EF8	Mejora de trabajo en equipo	E. Enfocados	0.00
EF9	Mejora de relaciones en el equipo	E. Enfocados	0.25
EF10	Mejora de entorno físico	E. Enfocados	0.00
EE7	Mejora continua, simplicidad	E. que Entregan	0.00
EE8	Entrega continua, builds automáticos	E. que Entregan	0.00
EE9	Cambios incrementales, trabajo de calidad, pruebas automáticas	E. que Entregan	0.00
EE10	Mentalidad de aprendizaje continuo	E. que Entregan	0.00
EO6	Ejecución de experimentos	E. Optimizados	0.00
EO7	Dirección del producto	E. Optimizados	0.00
EO8	Análisis de cohorte	E. Optimizados	0.00
TOTAL			0.25

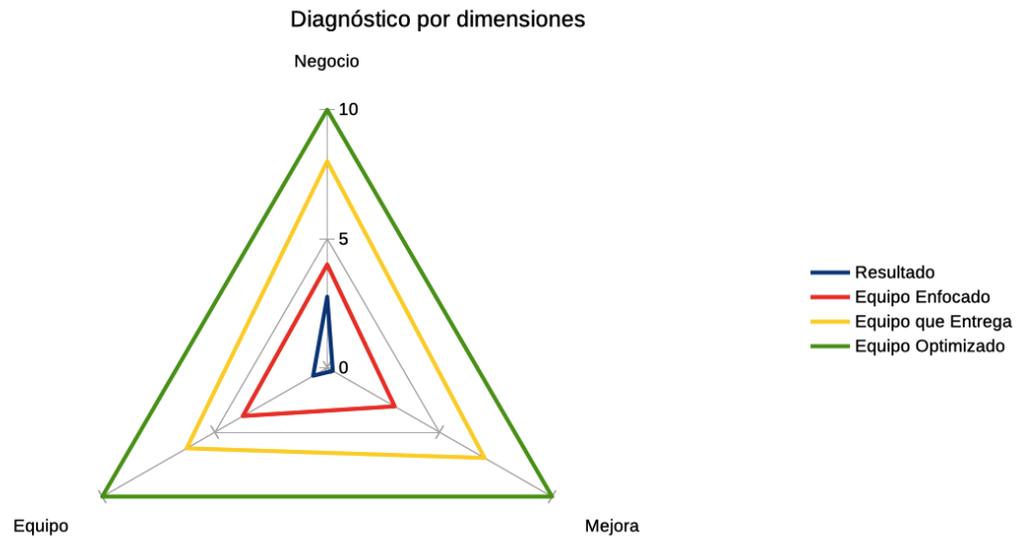


Figura 4.21: Diagnóstico por dimensiones

El gráfico se obtiene de la calificación que se ha dado a cada competencia dentro de cada zona. La línea azul corresponde al resultado de evaluación del equipo del proyecto Partner Portal. La línea roja es el puntaje esperado para los equipos que han alcanzado fluidez en la zona de Equipos Enfocados, la línea amarilla para los Equipos que Entregan y la línea verde para los Equipos Optimizados.

No existe un puntaje para los Equipos Fortalecidos porque es una zona experimental, de vanguardia, de la que no se conoce qué forma tiene o tendrá ya que representa el futuro de Agile.

4.3.6 Zona de acuerdo a competencias

En la figura 4.21: Diagnóstico por dimensiones, se puede observar que el equipo del proyecto Partner Portal está trabajando en las competencias de la zona de Equipos Enfocados.

Tiene un mayor puntaje en la dimensión Negocio: 2.75/10, seguido por la dimensión Equipo con 0.62/10 y por último la dimensión Mejora con 0.25/10.

Para que el equipo alcance la fluidez en la zona de Equipos Enfocados requiere llegar a

un puntaje de 4 en la dimensión Negocio, 3.75 en la dimensión Equipo y 3 en la dimensión Mejora. Se encuentra más cerca de su objetivo en la categoría Negocio.

Todavía debe trabajar mucho para ser considerado fluido en esta zona. Sin embargo, su objetivo real debería ser alcanzar la fluidez en la zona de Equipos que Entregan para así satisfacer las necesidades de su cliente. Para esto su puntaje debería ser 8 en la dimensión Negocio, 6.25 en la dimensión Equipo y 7 en la dimensión Mejora.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El resultado del diagnóstico de Fluidez Ágil indica que el equipo del proyecto Partner Portal de la empresa Hyna está trabajando principalmente en las competencias de la zona de Equipos Enfocados. Aun no ha alcanzado la fluidez en esta zona, que representa los fundamentos Agile.
- El equipo ha alcanzado la fluidez en las competencias individuales «involucramiento y empoderamiento del cliente», y «fijación de prioridades de parte del cliente» de la dimensión «respuesta a las necesidades del negocio». En ambas competencias se evidencia una relación activa entre el equipo y el cliente. Éste ha entendido que su involucramiento es fundamental en el proyecto.
- El equipo inicia su camino a la fluidez en la competencia «software que funciona y es valorado por el cliente» de la dimensión «respuesta a las necesidades del negocio». Esto significa que aunque el equipo se ha enfocado en las prioridades fijadas por el cliente, los progresos incrementales según estas prioridades no han sido los esperados por el cliente.
- El diagnóstico de Fluidez Ágil indica que el equipo está apenas iniciando su camino a la fluidez en las competencias «equipos auto-organizados» y «trabajo y responsabilidad de equipo» de la dimensión «trabajo efectivo como equipo». El equipo no ha demostrado tener una cultura de trabajo en equipo, que es lo que requiere la zona de Equipos Enfocados. Esto se debe en gran parte a la cultura de individualismo promovida por la empresa que reconoce los rendimientos individuales.

- El equipo inicia su camino a la fluidez en la competencia «mejora en la relaciones dentro del equipo» de la dimensión «maestría del equipo». El equipo es consciente de que las buenas relaciones entre los miembros del equipo son importantes para el proyecto, pero los intentos para mejorar estas relaciones no se han mantenido. Esta es la última competencia de la zona de Equipos Enfocados en la que demuestra algún progreso.
- El equipo está acercándose a la fluidez en la competencia «integración continua» de la dimensión «respuesta a las necesidades del negocio». Esta competencia pertenece a la zona de Equipos que Entregan y es la única de esta zona en la que el equipo ha demostrado progreso. Esta competencia, esencialmente técnica, es importante para que el equipo pueda responder al ritmo que requiere el cliente en base a las exigencias del mercado. Sin embargo, el producto que se entrega acarrea deuda técnica que no ha sido atendida, lo que implica que no sea de alta calidad y se invierta más tiempo en la corrección de errores y menos en nuevas funcionalidades.
- Las metodologías o los procesos no fallan, lo hacen las personas que las implementan. Esto nos recuerda que los individuos y sus interacciones son más importantes que los procesos y herramientas.
- No existe una receta que se pueda aplicar a todas las empresas y situaciones. Este trabajo ha buscado dar una visión global de lo que se espera en los proyectos ágiles. Cada empresa debe revisar su forma de trabajo y aplicar las soluciones de acuerdo a su realidad.

5.2 RECOMENDACIONES

- La primera meta del equipo debe ser alcanzar la fluidez en la zona de Equipos Enfocados. Si logra este objetivo habrá dado el paso más difícil porque requerirá abrazar los fundamentos Ágiles y un cambio a una cultura de equipo.
- El cambio principal debe venir de los altos ejecutivos de la empresa. El trabajo de un gerente general debe ser definir y hacer cumplir la cultura y los valores de la organización. Es entonces a este nivel donde se debe fomentar la cultura de trabajo en equipo.
- Se recomienda reconocer los logros de los equipos en función del valor entregado a los clientes. Se deben eliminar las recompensas y rankings individuales basados en el valor entregado a los accionistas.
- Se debe promover una comunicación más efectiva para equipos que trabajan distribuidos en diferentes ubicaciones geográficas. Si no es posible eliminar este impedimento, se deben buscar alternativas. Por ejemplo el empleo de video-llamadas cuando se requiera comunicación cara a cara, salas de conferencias con soporte audiovisual para reuniones del equipo completo, intercambio de miembros del equipo que se encuentren en diferentes ubicaciones geográficas para fomentar la consolidación y el desarrollo de capacidades como equipo.
- No todas las personas son capaces de trabajar en la forma que los proyectos ágiles exigen. Para esto es necesario implementar un proceso de selección de personal que tenga la capacidad y voluntad de trabajo en equipo, y que además sean pro-activas.
- Es indispensable que los miembros del equipo y administrador de proyectos sean capacitados por profesionales con experiencia comprobada en proyectos de desarrollo ágiles, tanto a nivel de competencias técnicas como de administración ágil.
- Se debe crear un espacio físico compartido enfocado en la productividad del equipo, que evite las distracciones del entorno y que promueva la interacción de sus integrantes.

- Alcanzada la fluidez en la zona de Equipos Enfocados, o en paralelo, el siguiente objetivo del equipo debe ser la zona de Equipos que Entregan. Esta zona representa la sustentabilidad ágil que le permitirá responder a las necesidades del negocio con la calidad y en el tiempo que requiere el cliente.
- El requisito para alcanzar la zona de Equipos Enfocados es la excelencia técnica. Por lo tanto la principal inversión debe estar encaminada hacia el desarrollo de prácticas ágiles de carácter técnico, como por ejemplo Extreme Programming (XP), Integración Continua, Test Driven Development (TDD), Refactoring, entre otras. Esto se puede lograr con la incorporación y el compromiso de expertos que guíen y trabajen diariamente junto al equipo.

BIBLIOGRAFÍA

- Ackoff, R. L. (1999). Re-creating the corporation, a design of organizations for the 21st century. Oxford University Press.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifiesto por el desarrollo ágil de software. Recuperado de <http://www.agilemanifesto.org/iso/es/>
- Collabnet. (2019). 13th annual state of agile report. Recuperado de <https://www.stateofagile.com/>
- Davis, C. (2015). Agile metrics in action. Manning Publications Co.
- Denning, S. (2019). Understanding the agile mindset. Recuperado de <https://www.forbes.com/sites/stevedenning/2019/08/13/understanding-the-agile-mindset/>
- Enlyft. (2020). Companies using sitecore. Recuperado de <https://enlyft.com/tech/products/sitecore>
- ForbesInsights, y PMI. (2017). Achieving greater agility. Project Management Institute.
- Fowler, M. (2019, 8). Agile software guide. Recuperado de <https://martinfowler.com/agile.html>
- Gartner. (2019). Magic quadrant research methodology. Recuperado de <https://www.gartner.com/en/research/methodologies/magic-quadrants-research/>
- Hagel, J., y Brown, J. S. (2013, 3). Institutional innovation. Recuperado de <http://dupress.com/articles/institutional-innovation/>
- Hamel, G., y Zanini, M. (2016). Excess management is costing the u.s. 3 trillion per year. Harvard Business Review. Recuperado de <https://hbr.org/2016/09/excess-management-is-costing-the-us-3-trillion-per-year>
- HewlettPackard. (2017). Agile is the new normal. Recuperado de <https://softwaretestinggenius.com/docs/4aa5-7619.pdf>
- Humble, J., Molesky, J., y O'Reilly, B. (2014). Lean enterprise: How high performance organizations innovate at scale. O'Reilly Media. Recuperado de <https://books.google.com.ec/books?id=ivixBQAAQBAJ>

- INEC. (2019). Índice de nivel de actividad registrada. Recuperado de <https://www.ecuadorencifras.gob.ec/indice-de-nivel-de-la-actividad-registrada/>
- Larsen, D., y Shore, J. (2019). The agile fluency model. Recuperado de <https://www.agilefluency.org/>
- Mankins, M., y Garton, E. (2017). Time, talent, energy. overcome organizational drag and unleash your team's productive power. Harvard Business Review Press.
- Muslihat, D. (2018, 3). Agile vs waterfall. Recuperado de <https://zenkit.com/en/blog/agile-vs-waterfall/>
- Pattigulla, M. (2017). Measure your code using metrics. Recuperado de <https://www.c-sharpcorner.com/article/measure-your-code-using-code-metrics/>
- PMI. (2018). Success in disruptive times. Recuperado de <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2018.pdf>
- Ralph, P. (2018, 5). The two paradigms of software development research. Science of Computer Programming.
- Rasmusson, J. (2010). The agile samurai. The Pragmatic Programmers.
- Ries, E. (2011). The lean startup. Crown Business.
- Schwaber, K., y Sutherland, J. (2016). The scrum guide. Scrum.Org.
- Schweigert, T., Nevalainen, R., Vohwinkel, D., Korsaa, M., y Biro, M. (2012, 5). Agile maturity model: Oxymoron or the next level of understanding. Communications in Computer and Information Science(290).
- Schweigert, T., Vohwinkel, D., Korsaa, M., Nevalainen, R., y Biro, M. (2013). Agile maturity model: analysing agile maturity characteristics from the SPICE perspective. Journal of Software: Evolution and Process.
- Senplades. (2013). Plan Nacional de Desarrollo / Plan Nacional para el Buen Vivir 2013-2017. Senplades.
- Sitecore. (2020a). Announcing the 2019 sitecore experience award winners. Recuperado de <https://www.sitecore.com/customers/experience-awards>
- Sitecore. (2020b). Sitecore experience platform. Recuperado de

- <https://www.sitecore.com/products/sitecore-experience-platform>
- Skarin, M. (2015). Real-world kanban. The Pragmatic Programmers.
- StackOverflow. (2018). Developer survey results 2018. Recuperado de <https://insights.stackoverflow.com/survey/2018>
- Tamturk, V. (2018). Gartner magic quadrant for dxp 2018 ... who's in who's out? Recuperado de <https://www.cms-connected.com/News-Archive/January-2018/Gartner-Magic-Quadrant-for-DXP-2018-Who-is-In-Who-is-Out>
- Thoughtworks. (2020). About us. Our story. Recuperado de <https://www.thoughtworks.com/about-us>
- van Haaster, K. (2020). Road-mapping your way to agile fluency. Recuperado de <https://www.thoughtworks.com/insights/blog/road-mapping-your-way-agile-fluency>
- Zwinkau, A. (2018, 12). The "waterfall" methodology was a historic accident and they knew it. Recuperado de <http://bezale1.tuxen.de/waterfall.html>