



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



AGILE Roadmap: diagnóstico y evaluación de prácticas ágiles para ser implementadas en equipos de trabajo

Fausto I. Nelson Amancio

Director: Dr. Patricio Letelier Torres

Julio del 2013

Trabajo Final de Máster presentado para optar por el título de Máster en Ingeniería del Software, Métodos Formales y Sistemas de Información, de la Universidad Politécnica de Valencia, Julio 2013.

Agradecimientos

Primeramente y antes que nada doy gracias a Dios por haberme dado la sabiduría para realizar este trabajo y por poner gente de buena fe que aportaron en el mismo aconsejándome, dándome ánimos para seguir y diciéndome que si puedo hacerlo.

Agradezco a mi familia por sus oraciones y por preocuparse por mí en todo tiempo y siempre estar pendientes de cualquier necesidad que pudiera tener sin importar la índole de dicha necesidad.

También agradezco de forma incondicional a mi novia Aury Charles por ser esa persona especial que siempre me decía “eres muy inteligente” y “estoy muy orgullosa de ti” esas frases me motivaban día tras día a seguir y hacer lo mejor que pudiera el trabajo que se me asignaba.

Quiero agradecer a mi asesor y director de tesis Patricio, quien ha jugado un papel importante en la elaboración de este trabajo, no solo en la corrección y aportación de ideas sino también con su paciencia en algunas etapas de este proceso.

Por último doy gracias a mis compañeros y amigos de la iglesia y compañeros de piso, principalmente José y Alexander, quienes de una manera u otra me ayudaron a retomar el rumbo en algún que otro momento que lo había perdido.

Muchísimas gracias a cada uno de los que aquí no menciono pero que aportaron al desarrollo de este trabajo.

Dedicatoria

A mi familia (*mis padres y hermanas*), ellos me han enseñado el verdadero valor de la vida y de tener gente a mi lado que se preocupan por mí. Gracias porque a pesar de mis defectos siempre han creído que puedo lograr todo lo que me proponga y porque siempre he contado con su apoyo en todos los sentidos.

A mi novia Aury, por hacerse cargo de mí cuando estuve débil de salud y cuando necesité un abrazo para levantar mis ánimos y por amarme como me ha demostrado durante todo este proceso.

Este triunfo no es solo mío, es de ustedes también. ¡Gracias!

Contenido

Capítulo 1. Introducción.....	7
➤ Motivación.....	8
➤ Objetivo.....	9
Capítulo 2. Métodos Ágiles.....	9
Introducción.....	9
➤ Métodos considerados.....	12
Capítulo 3. Estado del Arte.....	29
¿Adopción o Transformación ágil?.....	29
➤ Estrategias para implantar prácticas ágiles.....	30
➤ Herramientas para diagnóstico e implantación.....	33
Capítulo 4. Modelo de evaluación de Prácticas.....	35
➤ Objetivos de Mejora.....	37
➤ Prácticas Ágiles.....	39
Capítulo 5. AGILE Roadmap.....	59
➤ Concepto teórico.....	59
➤ AGILE Roadmap.....	60
➤ Formula implementada en AGILE Roadmap.....	72
Capítulo 6. Estadísticas y explotación de datos.....	74
Capítulo 7. Conclusiones.....	97
Referencias.....	98
Anexo 1 – Ejemplo del Informe generado por AGILE Roadmap.....	100

Figuras

Ilustración 1 - Ejemplo diversidad de prácticas dependiendo de las metodologías evaluadas.	7
Ilustración 2 - Ejemplo flujo de trabajo con tarjetas en cada etapa.	13
Ilustración 3 - kanban más específico / mayor observabilidad.....	14
Ilustración 4 - Ejemplo ficticio de un cuello de botella.....	15
Ilustración 5 - Ejemplo flujo de trabajo y elementos de Scrum.	21
Ilustración 6 - Características que debe tener un equipo ágil.	30
Ilustración 7 - Agility Calculator de Versionone.....	34
Ilustración 8 - Base de Conocimiento AGILE Roadmap.	36
Ilustración 9 - Diagrama de la base de datos de AGILE Roadmap.	61
Ilustración 10 - Pagina principal de AGILE Roadmap.....	66
Ilustración 11 –Paso 1 del proceso de evaluacion en la aplicacion AGILE Roadmap.....	67
Ilustración 12 - Paso 2 del proceso de evaluacion en la aplicacion AGILE Roadmap	68
Ilustración 13 - Paso 3 del proceso de evaluacion en la aplicacion AGILE Roadmap.	69
Ilustración 14 - Hoja de resultado del proceso de evaluacion en la aplicacion AGILE Roadmap.....	71

Tablas

Tabla 1 - Tamaño de Equipos Evaluados.	74
Tabla 2 - Ambito de equipo y evaluaciones realizadas.	75
Tabla 3 - Categorías de Organización del trabajo.	76
Tabla 4 - Países desde donde se realizaron las evaluaciones.	77
Tabla 5- Importancia promedio y frecuencia de aparición de los objetivos.....	79
Tabla 6 - Aplicación promedio y frecuencia de aparición de las prácticas.	83
Tabla 7 - Dificultad promedio y frecuencia de aparición de los desafíos.	87
Tabla 8 - Dificultad promedio de las prácticas.....	91
Tabla 9 - Evaluación promedio y frecuencia de recomendación de las prácticas.	94

Graficas

Grafica I - Tamaño de equipo más frecuente.	74
Grafica II - Porcentaje de evaluaciones por ámbito.	75
Grafica III - Forma de organización más común entre los equipos evaluados.....	76
Grafica IV - Valor porcentual de evaluaciones por países.	78
Grafica V - Promedio de Importancia obtenido en las evaluaciones.	80
Grafica VI - Frecuencia de Aparición de los Objetivos (Cantidad de evaluaciones en las que aparece).....	80
Grafica VII - Aplicación promedio obtenida en las evaluaciones realizadas.....	83
Grafica VIII - Frecuencia de Aparición de las prácticas (Cantidad de evaluaciones en las que aparece).....	84
Grafica IX - Promedio dificultad de los desafíos.	88
Grafica X - Frecuencia de aparición de las dificultades.	88
Grafica XI - Promedio de dificultad presentado por las prácticas evaluadas.	91
Grafica XII - Promedio de evaluación de las prácticas.	95
Grafica XIII - Frecuencia con que las prácticas son recomendadas (Cantidad de Agile Roadmap en la práctica ha sido recomendada).	96

Capítulo 1. Introducción.

Para la realización de este trabajo se han tomado como enfoque principal cuatro métodos ágiles de la gran variedad existente en el mundo del agilísimo debido a su popularidad. Estos métodos son: Scrum, Kamban, Lean Development y Xtreme Programming (XP). Prácticas Ágiles.

Enfocándonos en estos cuatro métodos, la cantidad de prácticas ágiles que propone cada uno y los diferentes contextos en los que puede ser aplicada una u otra práctica, es muy complejo determinar qué conjunto de prácticas es más adecuado para implantar en el equipo.

Un ejemplo de lo complejo que puede llegar a ser la obtención de una lista de mejoras adecuada para un equipo de desarrollo, es la cantidad de métodos que quieran tomar en consideración y las prácticas que ofrece cada uno. Si se consideran cuatro métodos ágiles y cada uno ofrece una gran cantidad de prácticas, se debe evaluar según las características y metas del equipo cuales prácticas aportan a los objetivos y en qué nivel aporta cada una para poder establecer un orden ya que es casi imposible implantar todas las prácticas a la vez, (*ver ilustración 1*).

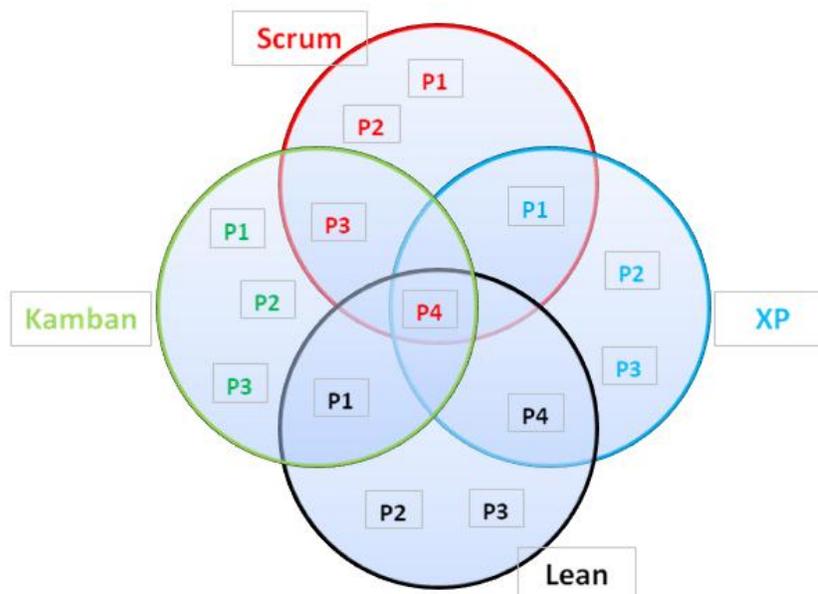


Ilustración 1 - Ejemplo diversidad de prácticas dependiendo de las metodologías evaluadas.

➤ **Motivación.**

La implantación de prácticas ágiles en los últimos años se ha convertido en un gran desafío para los equipos de trabajo en todos los ámbitos, desde el desarrollo de software hasta líneas de producción de artículos de manufactura.

Dependiendo en el contexto que opere el equipo de trabajo, el proceso de implantación de prácticas ágiles podría tener más o menos obstáculos, al pasar del tiempo este proceso se ha visto afectado por la gran variedad de metodologías y prácticas existentes en el mundo del Desarrollo Ágil. A pesar de las diferentes técnicas o recomendaciones para facilitar la implantación de metodologías ágiles sigue siendo un desafío bastante complicado para los equipos de trabajo.

Mientras que muchas de las técnicas de desarrollo, métodos y procesos de implantación han tenido éxito en la mejora de la calidad y el costo de los productos software, todavía hay una gran necesidad de que el desarrollo de software sea más eficaz y eficiente. La Implantación de metodologías/prácticas ágiles está atrayendo la atención de la industria con el objetivo de aumentar la eficacia y la eficiencia. [1]

La gran variedad de métodos y prácticas existentes añaden un alto nivel de dificultad al proceso de selección o adaptación para un equipo al agilismo.

A pesar de las técnicas desarrolladas para la implantación de prácticas ágiles, no existe un cuerpo de conocimiento consensuado que exponga la manera más adecuada de introducir este cambio al equipo debido a que para cada equipo de trabajo/desarrollo se dan diversas situaciones que pueden variar la aplicabilidad de una técnica entre uno y otro, lo que justifica un poco la inexistencia de una base de conocimiento con dicha información.

Los problemas y/o obstáculos existentes en el camino del agilísimo para un equipo precisan de procesos que puedan facilitar la búsqueda de información para hacer más fácil la obtención de una mejor lista de mejoras (*Improvement Backlog*), que abarque todas las áreas de interés para el equipo de trabajo y no solo áreas contempladas por una metodología en específico. No se trata de seleccionar

métodos, más bien es la selección de prácticas que puedan facilitar al equipo la búsqueda de la información necesaria para ser un equipo ágil.

➤ **Objetivo.**

El objetivo de este trabajo es desarrollar un sitio/aplicación web para facilitar a los equipos de trabajo la selección de una lista de prácticas especialmente adaptadas a las necesidades y características del equipo, las cuales pueden ser muy variadas dependiendo el contexto del equipo o producto.

Capítulo 2. Métodos Ágiles.

Introducción.

A principios de la década de los '90, surgió un enfoque que fue bastante revolucionario para esos momento ya que iba en contra de toda creencia de que mediante procesos altamente definidos se iba a lograr obtener software en el tiempo adecuado, a buen costo y con la calidad requerida. Este enfoque fue planteado por primera vez por James Martin [2] y se dio a conocer en la comunidad de Ingeniería de Software con el nombre de RAD o Rapid Application Development. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. En general, se considera que este fue uno de los primeros hitos hacia la agilidad en los procesos de desarrollo. [3]

El desarrollo ágil de productos software, es un conjunto de métodos de ingeniería del software basado en el desarrollo iterativo e incremental, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos auto-organizados y multidisciplinarios.

Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo.

Cada etapa del ciclo de vida de un proyecto incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe añadir demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener el producto o parte del mismo a un nivel en el cual se pueda poner de producción al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, se establece el manifiesto ágil [13]. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue definir los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó *The Agile Alliance*, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil.

En el Manifiesto ágil se valora al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.

La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.
[4]

Según los valores inspirados en el *Manifiesto Ágil*, nacen los doce principios contenidos en el mismo, los cuales son [13]:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.

11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Cabe destacar que los principios mencionados anteriormente no constituyen prácticas directamente aplicables, los métodos son los encargados de esto.

Actualmente existe una gran cantidad de métodos ágiles, Kanban, Lean Software Development, Scrum, Extreme Programming, Feature-Driven Development, Dynamic System Development Method, Crystal Metodologies, pero entre las más usadas están las primeras cuatro, los cuales se explican brevemente a continuación.

➤ **Métodos considerados.**

Entre los cuatro métodos que consideramos existen diferencias y se aplica mejor uno u otro dependiendo las características y necesidades del equipo de trabajo.

Lean proviene de la organización en un intento de crear el contexto que el equipo necesita tener en orden para operar efectivamente. Lean crea un contexto para todo lo que se hace. Este incorpora el trabajo, el liderazgo y la educación.

Scrum intenta conseguir que los equipos trabajen efectivamente y luego mantenerlos juntos. Es un método ágil basado en la administración del proyecto y cada miembro del equipo trabaja individualmente.

XP es un método de desarrollo que está más centrado en la programación o creación del producto donde los miembros del equipo programan en parejas, estos siguen estrictamente el orden de prioridad de las tareas definido por el cliente.

Kanban consiste en limitar el trabajo en progreso (WIP – Work In Progress), utilizando tarjetas como indicadores de que nuevos bloques pueden ser incorporados en un paso del flujo de trabajo. A diferencia de otros métodos este no establece ningún rol, Pero existe libertad para añadir los roles que se consideren necesarios.

I. KANBAN

KANBAN [5] es un sistema de planificación de trabajo que ayuda a determinar que producir, cuando producirlo y qué cantidad producir. KANBAN es una palabra de origen Japonés que significa tablero o tarjeta.

KANBAN se basa en un sistema de producción que dispara trabajo solo cuando existe capacidad para procesarlo. El flujo de trabajo es representado usando tarjetas KANBAN de las cuales se dispone en cantidades limitadas.

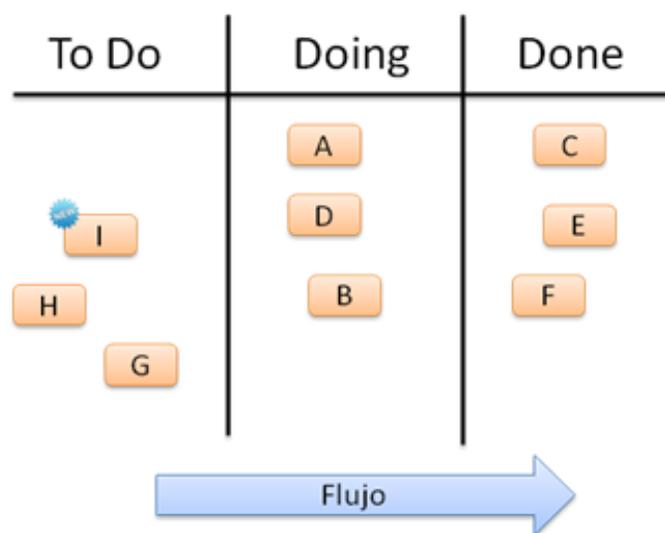


Ilustración 2 - Ejemplo flujo de trabajo con tarjetas en cada etapa. [12]

Cada tarjeta acompaña un ítem de trabajo durante todo el proceso de producción, hasta que éste, es empujado fuera del sistema, liberando una tarjeta. Un nuevo ítem de trabajo, solo podrá ser ingresado/aceptado si se dispone de una tarjeta libre, un ejemplo del flujo de trabajo se puede apreciar en la *ilustración 2*.

Este proceso de producción, donde un trabajo se introduce al sistema solo cuando existe disponibilidad para procesarlo, se denomina pull (tirar) en contrapartida al mecanismo push (empujar), donde el trabajo se introduce en función de la demanda. En la *ilustración 3* se puede apreciar que este sistema puede ser mas específico, hasta el punto de lograr una mejor predicción del estado del trabajo por actividades.

En el desarrollo de software, KANBAN fue introducido por David Anderson en la Unidad de Negocios XIT de Microsoft, en 2004. [5]

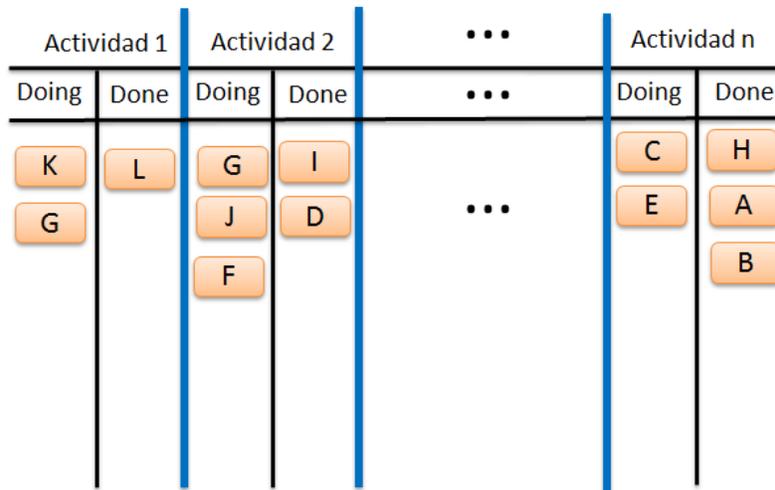


Ilustración 3 - kanban más específico / mayor observabilidad. [12]

Un tablero KANBAN, se divide en columnas las cuales representan un proceso de trabajo. Un ejemplo clásico de columnas para dividir un tablero KANBAN, sería el siguiente:

Cola de entrada | Análisis | Desarrollo | Test | Deploy | Producción

La cantidad y nombre de las columnas, varía de acuerdo a las necesidades de cada equipo y en la mayoría de los casos, estas son subdivididas en dos columnas: cola de espera y en curso.

Las tres reglas de KANBAN

Con tan solo tres simples reglas, KANBAN demuestra ser una de las metodologías adaptativas que menos resistencia al cambio presenta. Dichas reglas son:

Mostrar el proceso

Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero, generalmente, públicamente asequible. El objetivo de mostrar el proceso, consiste en:

- Entender mejor el proceso de trabajo actual.
- Conocer los problemas que puedan surgir y tomar decisiones.
- Mejorar la comunicación entre todos los interesados/participantes del proyecto.
- Hacer los futuros procesos más predecibles.

Limitar el trabajo en curso (WIP)

Los límites del WIP (work in progress - trabajo en curso) consisten en acordar anticipadamente, la cantidad de ítems que pueden abordarse en cada actividad (es decir, en cada columna del tablero). El principal objetivo de establecer estos límites, es el de evitar cuellos de botella.

Los cuellos de botella representan el estancamiento de una actividad determinada.



Ilustración 4 - Ejemplo ficticio de un cuello de botella.

En la *ilustración 4*, podemos visualizar, que en la columna "pruebas" se produce una acumulación de trabajo, el límite WIP se ha alcanzado, mientras que el proceso siguiente (Deploy), está totalmente libre. Esto, marca un problema a resolver en la actividad Pruebas.

La mayoría de las veces, limitar el WIP motiva la colaboración del equipo en las diferentes actividades. Pues mientras existen actividades colapsadas, existen a la vez, actividades libres para aceptar nuevos ítems. El cuello de botella ha generado un estancamiento, y las actividades libres, pueden ayudar a "desestancar" a las colapsadas.

Optimizar el flujo de trabajo

Un objetivo en tres partes, producción estable, continua y previsible. Midiendo el tiempo que demanda el ciclo completo de ejecución del proyecto, se obtiene el CycleTime. (Por ejemplo, cantidad de días desde el inicio del análisis hasta el fin del Deploy según el ejemplo de la *ilustración 5*).

Al dividir, el CycleTime por el WIP, se obtiene el "rendimiento de trabajo", denominado Throughput, es decir, la cantidad de ítems que un equipo puede terminar en un determinado período de tiempo.

Con estos valores, la optimización del flujo de trabajo consistirá en la búsqueda de:

- Minimizar el CycleTime
- Maximizar el Throughput
- Lograr una variabilidad mínima entre CycleTime y Throughput

Estas reglas permiten que KANBAN sea una de las metodologías más fáciles de aplicar en los equipos de desarrollo, debido a su simplicidad en la aplicación de sus técnicas y los esfuerzos requeridos por los integrantes de los equipos.

II. Lean Software Development

Lean [6] es una translación de los principios y prácticas de la manufactura hacia el dominio del desarrollo de software. Adaptado del Sistema de producción Toyota, apoyado por una sub-cultura pro-lean que está surgiendo desde la comunidad ágil.

El término Lean aplicado al de desarrollo de software tiene origen en un libro del mismo nombre, escrito por Mary Poppendieck y Tom Poppendieck [6]. El libro presenta los tradicionales principios Lean en forma modificada, así como un conjunto de 22 instrumentos y herramientas y las comparaciones con otras prácticas ágiles. La participación de Mary y Tom en la comunidad del desarrollo ágil de software, incluyendo charlas en varias conferencias, ha dado lugar a dichos conceptos, que son más ampliamente aceptados en la comunidad de desarrollo ágil.

Principios Lean

El desarrollo Lean puede resumirse en siete principios. [6]

1) *Eliminar los desperdicios*

Todo lo que no añade valor al cliente se considera un desperdicio:

- a. Código y funcionalidades innecesarias
- b. Retraso en el proceso de desarrollo de software
- c. Requisitos poco claros
- d. Burocracia
- e. Comunicación interna lenta

Con el fin de poder eliminar los desperdicios el equipo debería ser capaz de reconocerlos y encontrarlos. Si alguna actividad podría ser excluida o el mismo resultado podría ser logrado sin ella, esta actividad es considerada un desperdicio. Los procesos y funcionalidades extra que no son usados por el cliente son desperdicios. Las esperas ocasionadas por otras actividades, equipos o procesos son desperdicio. Los defectos y la baja calidad son los desperdicios. Los gastos de gestión que no producen valor real son desperdicios. Se utiliza una técnica llamada value stream mapping (o mapa

de flujo de valor) para distinguir y reconocer los desperdicios. El segundo paso consiste en señalar las fuentes de los desperdicios y eliminarlos. Lo mismo debe hacerse iterativamente hasta que incluso los procesos y procedimientos que parecían esenciales sean eliminados.

2) *Ampliar el aprendizaje*

El proceso de aprendizaje es acelerado con el uso de iteraciones cortas, cada una de ellas acompañada de refactorización y sus pruebas de integración.

Incrementando la retroalimentación mediante reuniones cortas con los clientes se ayuda a determinar la fase actual de desarrollo y se ajustan los esfuerzos para introducir mejoras en el futuro. Durante las reuniones, tanto los clientes como el equipo de desarrollo, logran aprender sobre el alcance del problema y buscan posibles soluciones para un mejor desarrollo. Por lo tanto, los clientes comprenden mejor sus necesidades basándose en el resultado de los esfuerzos del desarrollo y los desarrolladores aprenden a satisfacer mejor estas necesidades.

Otra idea para ampliar el aprendizaje es a través de la integración del cliente en el ambiente de desarrollo para concentrar la comunicación en las soluciones futuras y no en las soluciones posibles, promoviendo así el nacimiento de la solución a través del diálogo con el cliente.

3) *Decidir lo más tarde posible*

El desarrollo de software está siempre asociado con cierto grado de incertidumbre, los mejores resultados se alcanzan con un enfoque basado en opciones por lo que se pueden retrasar las decisiones tanto como sea posible hasta que éstas se basen en hechos y no en suposiciones y pronósticos inciertos.

Cuanto más complejo es un proyecto, más capacidad para el cambio debe incluirse en éste, así que debe permitirse el retraso de los compromisos importantes y cruciales. El enfoque iterativo promueve este principio: la

capacidad de adaptarse a los cambios y corregir los errores, ya que un error podría ser muy costoso si se descubre después de la liberación del sistema.

4) *Reaccionar tan rápido como sea posible*

En la era de la rápida evolución tecnológica, no es el más grande quien sobrevive, sino el más rápido. Cuanto antes se entrega el producto final sin defectos considerables más pronto se pueden recibir comentarios y se incorporan en la siguiente iteración. Cuanto más cortas sean las iteraciones, mejor es el aprendizaje y la comunicación dentro del equipo. Sin velocidad, las decisiones no pueden ser postergadas.

La velocidad asegura el cumplimiento de las necesidades actuales del cliente y no lo que éste requería para ayer. Esto les da la oportunidad de demorarse pensando lo que realmente necesitan, hasta que adquieran un mejor conocimiento. Los clientes valoran la entrega rápida de un producto de calidad.

5) *Potenciar el equipo*

Ha habido una creencia tradicional en la mayoría de las empresas acerca de la toma de decisiones en la organización: los administradores dicen a los trabajadores cómo hacer su propio trabajo. En una técnica llamada Work-Out, hay cambios: a los directivos se les enseña a escuchar a los desarrolladores, de manera que éstos puedan explicar mejor qué acciones podrían tomarse, así como ofrecer sugerencias para mejoras. Los directores de proyecto más experimentados simplemente han declarado la clave de éxito de los proyectos: "Buscar la buena gente y dejarles hacer su propio trabajo".

Los desarrolladores deberían tener acceso a los clientes; el jefe de equipo debe proporcionar apoyo y ayuda en situaciones difíciles, así como asegurarse de que el escepticismo no arruine el espíritu de equipo.

6) *Crear la integridad*

El siempre exigente cliente debe tener una experiencia general del sistema. A esto se le llama percepción de integridad: ¿Qué tan bien encajan las piezas del software? ¿Qué tan intuitivo es su uso? ¿Precio? ¿Qué tan bien resuelve los problemas? Integridad Conceptual significa que los componentes separados del sistema funcionan bien juntos, como en un todo, logrando equilibrio entre la flexibilidad, mantenibilidad, eficiencia y capacidad de respuesta.

La información necesaria es recibida en pequeños lotes, no grandes cantidades y con una preferible comunicación cara a cara, sin documentación por escrito. El flujo de información debe ser constante en ambas direcciones, a partir del cliente a los desarrolladores y viceversa, evitando así la gran y estresante cantidad de información después de un largo periodo de desarrollo en el aislamiento.

7) *Ver todo el conjunto*

Los sistemas de software hoy en día no son simplemente la suma de sus partes, sino también el producto de sus interacciones. Los defectos en el software tienden a acumularse durante el proceso de desarrollo por medio de la descomposición de las grandes tareas en pequeñas.

Las causas reales de los defectos deben ser encontradas y eliminadas. Cuanto más grande sea el sistema, más serán las organizaciones que participan en su desarrollo y más partes son las desarrolladas por diferentes equipos y mayor es la importancia de tener bien definidas las relaciones entre los diferentes proveedores con el fin de producir una buena interacción entre los componentes del sistema.

La manera de pensar ofrecida por Lean tiene que ser bien entendida por todos los miembros de un proyecto antes de aplicarlo de manera concreta en una situación real.

III. Scrum

Scrum [14] es un marco de trabajo estructurado para dar soporte al desarrollo de productos complejos. Scrum consiste en los equipos Scrum y en sus roles, eventos, artefactos y reglas asociadas.

En 1995, Sutherland y Schwaber presentaron conjuntamente un artículo describiendo la metodología Scrum [7], de las metodologías ágiles, es la más utilizada, según una encuesta publicada por VersionOne realizada a 4048 profesionales del Software, la misma, revela que el 54% de los encuestados, utiliza Scrum como metodología para la gestión de proyectos de desarrollo de Software.

Scrum se fundamenta en la teoría empírica de control de procesos, o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea una aproximación iterativa e incremental para optimizar la predictibilidad y controlar el riesgo.

Scrum permite la creación de equipos auto-organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

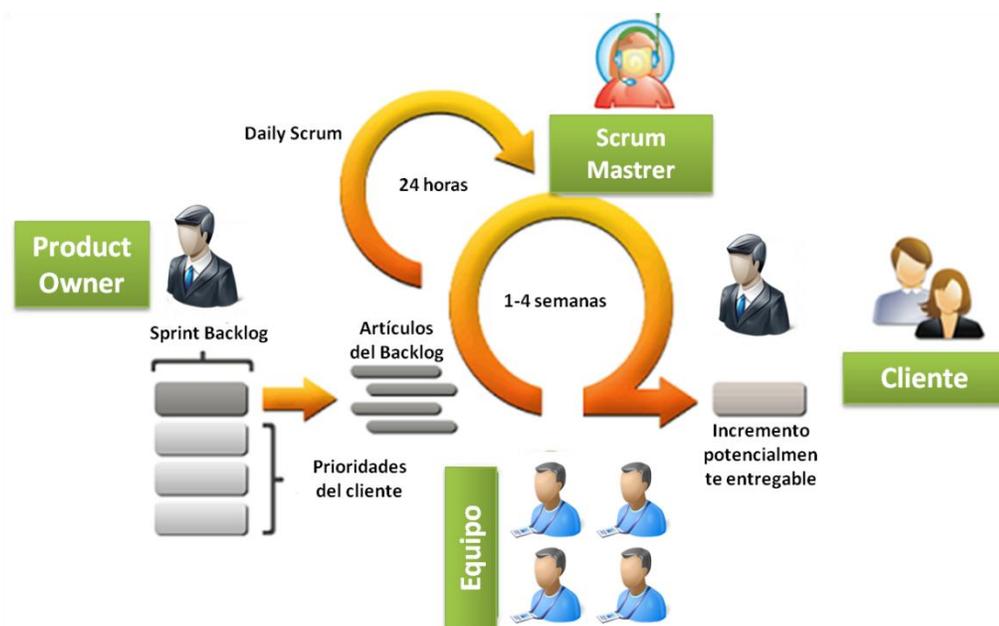


Ilustración 5 - Ejemplo flujo de trabajo y elementos de Scrum. [18]

En la *ilustración 5* se muestra un ejemplo del flujo de trabajo de Scrum y algunos de sus elementos, los cuales se describen a continuación [14]:

➤ **Equipo Scrum**

➤ *Product Owner (Dueño del producto)*

Es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. El cómo se lleva esto a cabo puede variar ampliamente entre distintas organizaciones, equipos Scrum, e individuos.

➤ *ScrumMaster (o Facilitador)*

Es el responsable de asegurar que Scrum es entendido y llevado a cabo. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El Scrum Master es un líder servil, al servicio del Equipo Scrum.

➤ *Equipo de desarrollo (Development Team)*

Consiste en los profesionales que desempeñan el trabajo de entregar un incremento de producto “Hecho”, potencialmente utilizable, al final de cada Sprint.

➤ **Eventos**

➤ *Daily Scrum*

Es una reunión restringida a un bloque de tiempo de 15 minutos, para que el equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una predicción acerca del trabajo que podría ser completado antes del siguiente.

Es mantenido a la misma hora y en el mismo lugar todos los días, para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica: ¿Qué se ha conseguido desde la última reunión?, ¿Qué se hará antes de la próxima reunión? Y ¿Qué obstáculos se encuentran en el camino?

➤ ***Sprint***

El corazón de Scrum es el Sprint, un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable. La duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten en la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

Durante el Sprint:

- No se realizan cambios que afectarían al Objetivo del Sprint (Sprint Goal).
- La composición del Equipo de Desarrollo se mantiene constante.
- Los objetivos de calidad no disminuyen; y
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

➤ ***Reunión de Planificación del Sprint (Sprint Planning Meeting)***

El trabajo a realizar durante el Sprint es planificado en la Reunión de Planificación de Sprint. Este plan es creado mediante el trabajo colaborativo del Equipo Scrum al completo.

La Reunión de Planificación de Sprint está restringida a una duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es proporcionalmente más corto. Por ejemplo, los Sprints de dos semanas tienen una Reunión de Planificación de Sprint de cuatro horas.

➤ ***Revisión de Sprint (Sprint Review)***

Al final del Sprint se lleva a cabo una Revisión de Sprint, para inspeccionar el Incremento y adaptar la Pila de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se ha hecho durante el Sprint. Basándose en eso, y en cualquier cambio a la Pila de Producto hecho durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse. Se trata de una reunión informal, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

➤ ***Retrospectiva del Sprint (Sprint Retrospective)***

La Retrospectiva de Sprint es una oportunidad para el equipo Scrum de inspeccionarse a sí mismo, y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor.

➤ **Artefactos**

➤ ***Product backlog***

Es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requerimientos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable del product backlog, incluyendo su contenido, disponibilidad y ordenación.

➤ ***Sprint backlog***

Es el conjunto de elementos de la product backlog seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. El Sprint backlog es una predicción hecha por el Equipo

de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad.

IV. Extreme Programming

Programación Extrema o XP [8] por sus siglas en inglés, fue concebida y desarrollada por Kent Beck para atender las necesidades específicas de desarrollo de software dirigido por pequeños equipos de cara a los requisitos cambiantes. Este método desafía muchos principios convencionales, incluyendo una vieja suposición de que el costo de cambiar una pieza de software se eleva drásticamente durante el curso del tiempo. XP reconoce que los proyectos tienen que trabajar para lograr esta reducción de costes.

XP es una disciplina de desarrollo de software. Es una disciplina porque hay ciertas cosas que se deben hacer para estar haciendo XP. [8]

Los valores originales de XP son [8]:

➤ **Simplicidad:**

La simplicidad es la base de XP. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando que el código esté autodocumentado.

Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

➤ **Comunicación:**

La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código autodocumentado es más fiable que los comentarios, y debe comentarse sólo aquello que no va a variar, por ejemplo el objetivo de una clase o la funcionalidad de un método.

Los programadores se comunican constantemente gracias a la programación por parejas.

La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.

➤ **Retroalimentación (feedback):**

Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.

➤ **Coraje o valentía:**

Muchas de las prácticas implican valentía. Una de ellas es siempre diseñar y programar para hoy y no para mañana. Esto es un esfuerzo para evitar detener el curso del proyecto en el diseño y requerir demasiado tiempo y trabajo para implementar todo lo demás.

La valentía le permite a los desarrolladores que se sientan cómodos con reconstruir su código cuando sea necesario. Esto significa revisar el sistema existente y modificarlo si con ello los cambios futuros se implementaran más fácilmente.

Valentía significa persistencia: un programador puede permanecer sin avanzar en un problema complejo por un día entero, y luego resolverlo rápidamente al día siguiente, sólo si es persistente.

➤ **Respeto:**

El respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros.

Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, esto crea una mejor autoestima en el equipo y eleva el ritmo de producción.

Un quinto valor, Respeto, fue añadido en la segunda edición del libro original, “Extreme Programming Explained: Embrace Change” [9].

Prácticas de XP [8]

- ***El juego de Planificación:*** Determinar rápidamente el alcance de la próxima publicación combinando las prioridades del negocio y de las estimaciones técnicas.
- ***Versiones pequeñas:*** Poner un sistema simple en producción rápidamente, luego liberar nuevas versiones en un ciclo muy corto.
- ***Metáfora:*** Guía todo el desarrollo con una sencilla historia compartida de cómo funciona todo el sistema.
- ***Diseño simple:*** El sistema debe ser diseñado lo más simple posible en cualquier momento dado. Cualquier complejidad adicional se elimina tan pronto como se descubra.

- **Pruebas:** Los programadores continuamente escriben pruebas unitarias, que se deben ejecutar sin problemas para continuar con el desarrollo. Los clientes escriben las pruebas que demuestran que las características están acabadas.
- **Refactorización:** Los programadores reestructuran el sistema, sin cambiar su comportamiento, para eliminar la duplicación, mejorar la comunicación y simplificar o añadir flexibilidad.
- **Programación en pareja:** Todo el código de producción se escribe con dos programadores en una sola máquina.
- **Propiedad colectiva:** Cualquier persona puede modificar el código en cualquier parte del sistema en cualquier momento.
- **Integración continua:** Integrar y construir el sistema varias veces al día, cada vez que una tarea se ha completado.
- **40 horas a la semana:** No trabajar más de 40 horas a la semana como regla. Nunca trabajar horas extras por segunda semana consecutiva.
- **Cliente en las instalaciones:** Incluir un usuario verdadero y directo en el equipo, disponible a tiempo completo para responder a las preguntas.
- **Normas de codificación:** Los programadores escriben todo el código de acuerdo con las normas que enfatizan la comunicación a través del código.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer.

Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Capítulo 3. Estado del Arte.

¿Adopción o Transformación ágil?

Durante el gran auge que ha tenido agilismo, han surgido grandes controversias. Una de ellas ha sido la Adopción o Transformación ágil.

La adopción es un término que se aplica a un producto o proceso [10]. El agilismo es una forma de pensar y una cultura que no se puede adoptar por sí misma.

La transformación por otro lado, implica un cambio de una forma de ser a otra forma de ser [10]. “Estamos transformando a Ágil.” Es una forma precisa para describir lo que se llevó a cabo en entornos en los que el cambio representa un cambio fundamental en los comportamientos y valores.

La palabra transición significa “movimiento, paso, o el cambio de una posición, estado, tema, concepto, etc.”. El significado de transición podría ser utilizado para describir la aprobación o la transformación. Dado que es ambigua, lo mejor es evitar este término en conjunto.

No es correcto tratar que por imposición y a través de directivas originadas distante de un equipo de desarrollo dichos equipos vivan lo que es el agilismo. Si se trabaja de esta manera, lo más que se podrá conseguir en el equipo será una adopción de prácticas ágiles y no transformar el equipo en un equipo ágil.

El proceso de implantación del agilismo no debe realizarse de la manera que se ha hecho los últimos años, cuando las metodologías que se debían utilizar eran dictadas a un nivel muy superior (nivel corporativo) y alejado de los desarrolladores los cuales son los directamente afectados por las metodologías que deben utilizarse para el desarrollo.

Es más eficaz la transformación ágil que la adopción, lo que no deja de lado que es relativamente más difícil alcanzar una transformación ágil que la adopción ágil. Al lograr una transformación ágil, el equipo logra que cada integrante asimile de forma natural y a un ritmo que puede seguir, cada una de las exigencias del nuevo método.

En la actualidad no se puede encontrar en el mundo del agilismo una herramienta que facilite la selección de prácticas ágiles para su aplicación en un equipo de desarrollo sin tener que “casarse” con una metodología en específico, lo que causa que los equipos prefieran adoptar ciertas prácticas ágiles directamente de una metodología antes que realizar un exhaustivo análisis de todas las metodologías y obtener las prácticas que mas puedan aportar a dicho equipo.

Provocar una revolución ágil significaría aplicar muchas prácticas ágiles al mismo tiempo y en gran profundidad. Es posible pero sería conveniente contar con condiciones favorables (quizás demasiadas), *ver ilustración 6*.



Ilustración 6 - Características que debe tener un equipo ágil. [12]

Si el equipo consigue algunas condiciones favorables en un cierto contexto, difícilmente podrá tener las mismas o todas las condiciones en otro contexto del equipo. [12]

➤ Estrategias para implantar prácticas ágiles

Hay muchas cosas que se deben tener presentes al momento de la implantación de prácticas ágiles en un equipo de desarrollo. En este apartado se describen algunas técnicas o estrategias recomendadas para lograr una exitosa implantación de las mismas con la menor repercusión negativa posible. A continuación se presenta un resumen extraído desde [15].

➤ **La Implantación debe ser a nivel de equipo o a nivel de cada producto o servicio.**

Una de las claves de una implantación exitosa del agilismo es que debe cultivarse y desarrollarse a partir de los equipos.

Como ya se ha mencionado anteriormente en este trabajo es importante favorecer la transformación ágil del equipo, lo cual es posible haciendo participe al equipo en la toma de decisiones para los cambios de la nueva metodología o forma de trabajo.

La afirmación anterior también nos lleva indirectamente a formar equipos, es decir, grupos reducidos de personas que trabajan en común en un producto o servicio, lo que también nos lleva a tener en cuenta que los equipos formados deben poder mantenerse a lo largo del tiempo, es decir, que al final un proyecto el equipo pueda seguir junto en otros proyectos para poder potenciar las ventajas del agilismo.

Dando por hecho que los equipos formados son estables y que el foco de la implantación esta puesto en las características de cada equipo, se estaría listo para definir una estrategia de implantación. Sin embargo, hay que tener en cuenta que un equipo no trabaja solo con un producto o servicio, en los casos donde se presenta esta peculiaridad, la implantación del agilismo debería considerar las particularidades que pudiera tener el equipo para cada uno de los productos o servicios en los que trabaja.

Si no se establecen claramente las diferencias de contexto del equipo y de cada producto o servicio en el que trabaja se estaría cometiendo nuevamente el típico error de sobre-proceso o la falta del mismo.

Sin importar el nivel de preparación de un equipo para poder hacer una revolución ágil en su manera de trabajar, es muy probable que no sea posible realizarla de un día a otro debido a que requiere tiempo y dedicación. Lo

normal es ir incorporando de forma incremental las prácticas al equipo para ir evolucionando la forma de trabajo.

➤ **Tener un líder de implantación en el equipo.**

En un contexto ideal, todo el equipo estaría de acuerdo con los cambios necesarios para la implantación del agilismo, aceptando las ventajas que dicha implantación puede aportar al desarrollo de su trabajo. Desafortunadamente esta situación no es nada frecuente en los equipos de desarrollo dada la diversidad de conceptos de trabajo que tiene cada uno de los miembros, pudiendo darse el caso que unos acepten el cambio y otros no.

Es de gran importancia que exista una figura reconocida en el equipo que haga el papel de líder de implantación como se encuentra en la metodología Scrum el cual se denomina Scrum Máster. Sería perfecto que del equipo surja el Scrum Máster, pero esta idea no es realista. Existen dos características con las que debe cumplir el Scrum Máster, conocimiento en agilismo y liderazgo. La persona seleccionada para ejercer como líder de la implantación debe ser capaz de dedicar todo el tiempo necesario a la implantación de las prácticas ágiles en el equipo, esta persona podría ser un jefe tradicional el cual correría con la ventaja de tener experiencia y habilidades de liderazgo ya desarrolladas o podría ser un miembro que dejara sus actividades comunes para lidiar con la implantación de las prácticas ágiles en el equipo.

Lo más importante a tener en cuenta es que la persona que vaya a desempeñar el rol de Scrum Máster sea un líder efectivo y que sea capaz de no mezclar su labor de apoyo a la implantación con los trabajos asignados al equipo.

➤ **Tener a disposición un coach (Entrenador) que apoye al equipo en la implantación.**

Teniendo en cuenta la dificultad de disponer con una persona en un equipo de desarrollo con conocimiento y experiencia en implantación de prácticas ágiles,

quien juegue el rol de coach complementara al líder de la implantación y ambos serán la figura ideal del Scrum Máster.

El coach debe ser una figura que apoye al líder de la implantación de forma transitoria y a medida que el líder vaya adquiriendo experiencia este debería ir desapareciendo hasta llegado el momento en que dicho líder pueda desempeñar el rol de Scrum Máster por cuenta propia.

Llegados al punto de que el líder está lo suficientemente preparado para desempeñarse como Scrum Máster, este podría desarrollar dicho rol en varios equipos en vez de trabajar con un equipo exclusivamente. El coach podría ser un Scrum Máster ya preparado, de lo contrario debería ser algún consultor externo capaz de orientar al líder en la preparación de la implantación como en los primeros Sprints de mejora prestándole la ayuda necesaria para evaluar las prácticas implantadas y buscar soluciones a los desafíos que pudieran aparecer.

➤ **Herramientas para diagnóstico e implantación.**

Para el seguimiento del desarrollo de un producto software existe una gran cantidad de herramientas que facilitan controlar todo el proceso. Esto es todo lo contrario cuando nos referimos al diagnóstico e implantación de prácticas que mejoren el estado ágil del equipo.

Luego de realizar varias búsquedas no hemos encontrado una herramienta que facilite la elaboración de una lista de prácticas adaptadas a las necesidades y características del equipo de trabajo o para el diagnóstico de dichas prácticas.

Algunas herramientas relacionadas con el diagnóstico e implantación, son aportes de la compañía Versionone, un Survey (*Cuestionario*) anual sobre el estado ágil de los equipos de trabajo que es muy genérico con respecto a brindar información útil para los equipos diagnosticar o implementar prácticas ya que solo especifica información como cuales son los métodos más utilizados en general y otras informaciones. También ofrecen una pequeña herramienta y muy simple desarrollada en Excel llamada Agility Calculator que se puede apreciar en la

Ilustración 7, esta herramienta brinda una valor porcentual de que tan ágil es el equipo respondiendo un pequeño cuestionario.



Agility Calculator

How Agile Are You?

While the business value of agile development methods is clear, the path to agile may not be. Built on a set of simple concepts, agility represents a fundamentally different way of doing business. High levels of agility require a substantial commitment to education and training, new tools and techniques, patient transition, and cultural change. This commitment must flow from management downward, and from the team upward.

Such change is not for everybody. Organizations that don't promote and value open communication, trust, and teamwork may not provide the most fertile ground for agility to succeed. This does not mean that agile cannot work for such organizations. Agility is a spectrum and some teams are more agile than others. Frequently, teams start with modest increases in agility, and gradually become more agile as they gain experience and comfort with agile processes.

Ultimately each organization is responsible for its own agility. With any process or methodology implementation, practices will need to be adapted and localized. The key is not the specific processes, but the mindset and manner in which they are applied. In Ron Jeffries' words, "The practices are not agility. They are a way to find agility."

Test Your Agility

Although by no means scientific, you can test your team's agility by rating it from 1 to 5 on the following 10 agility factors. 5 indicates the highest level of agility and 1 indicates the lowest. Virtually no organization will score the maximum on each criteria, but the best ones will continue to push themselves further and further down the agile spectrum as opportunities arise.

Criteria:

1. **Team Communication**

1
 2
 3
 4
 5

[1 = minimal, written, knowledge is power] [5 = open, trusting, face-to-face]
2. **User Accessibility**

1
 2
 3
 4
 5

[1 = limited, off-site] [5 = constant, on-site]
3. **Team Location**

1
 2
 3
 4
 5

[1 = highly distributed] [5 = colocated]
4. **Team Structure**

1
 2
 3
 4
 5

[1 = departmental, top-down, large teams] [5 = cross-functional, self-organizing, small teams]
5. **Delivery Frequency (Shippable)**

1
 2
 3
 4
 5

[1 = infrequent, 3+ months] [5 = frequent, 1 - 2 weeks]
6. **Measurement of Progress**

1
 2
 3
 4
 5

[1 = phases, tasks, documents] [5 = features / business value, working software]
7. **Ability to Change Direction**

1
 2
 3
 4
 5

[1 = low, prevented] [5 = high, embraced]
8. **Testing**

1
 2
 3
 4
 5

[1 = manual, post coding] [5 = integrated, automated, test driven]
9. **Planning Approach**

1
 2
 3
 4
 5

[1 = upfront, detailed, activity-based] [5 = just enough, adaptive, continuous]
10. **Process Philosophy**

1
 2
 3
 4
 5

[1 = static, audited, "my way"] [5 = analyze, adapt, improve]

Your score: 0
means:

Ilustración 7 - Agility Calculator de Versionone.

Capítulo 4. Modelo de evaluación de Prácticas.

La estrategia presentada en este trabajo abarca tres tópicos:

➤ **Diagnostico de equipo-producto/servicio**

El diagnostico consiste en la evaluación del equipo-producto/servicio, dependiendo de su contexto, ámbito de trabajo y cantidad de miembros involucrados en el mismo. Dependiendo de factores como los ya mencionados, hay prácticas que son NO aplicables debido a sus requisitos.

➤ **Evaluación de prácticas**

Dependiendo de los objetivos del equipo-producto/servicio se pueden obtener listas de mejoras muy diferentes una de otra debido a los diferentes aportes que pueda hacer una práctica a un objetivo o a vario de ellos.

Un listado de prácticas de una metodología puede contener prácticas no aplicables al equipo-producto/servicio, por lo cual es imprescindible evaluar cada una de las prácticas ofrecida por las metodologías seleccionadas para así ir descartando prácticas que no aportan o no son de interés. Cada práctica puede tener un nivel de aplicación diferente y un conjunto de desafíos que superar, donde cada desafío tiene un nivel de dificultad que puede variar entre uno y otro, también cabe tener en cuenta que cada práctica podría estar relacionada a otras prácticas lo cual hace que sea necesario evaluar cada una de las prácticas.

➤ **Obtención de la lista de objetivos.**

Un Roadmap, es una representación estructurada de objetivos en un periodo de tiempo que muestra un plan de desarrollo y es utilizado para la planificación estratégica y comunicaciones.

Hemos automatizado la obtención de este documento mediante el desarrollo de una aplicación web la cual explicaremos más adelante en este trabajo.



Ilustración 8 - Base de Conocimiento AGILE Roadmap.[18]

En el diagrama mostrado en la *ilustración 8* se muestra la base de conocimiento sobre la que funciona AGILE Roadmap, cada objetivo tiene un nivel de importancia asignado por el usuario estos objetivos están directamente relacionados con una o varias prácticas ágiles y recibiendo de la misma un nivel de contribución que puede variar dependiendo de la práctica.

Cada práctica tiene asociado un esfuerzo de preparación y un nivel de aplicación, el usuario puede especificar el nivel de aplicación de cada una de las prácticas candidatas mostradas en el paso 2. Las prácticas se relacionan entre sí debido a que algunas aportan valor a otras.

Los desafíos van asociados a las prácticas, siendo que cada práctica puede estar asociada a varios desafíos y un desafío a varias prácticas. Cada desafío tiene un nivel de dificultad a superar por parte del equipo de trabajo el cual es especificado en el paso 3.

➤ **Objetivos de Mejora**

En este apartado se describen un conjunto de objetivos de mejora extraídos a partir de información en [12].

Cabe destacar que un objetivo de mejora se puede definir como la finalidad de una acción de perfeccionamiento, en palabras más claras es la finalidad de hacer que algo funcione de una manera más adecuada.

A continuación la lista de objetivos de mejoras:

No.	Objetivo de Mejora
1	Evitar o reducir los retrasos en las entregas
2	Aumentar la productividad del equipo
3	Gestionar eficazmente los cambios, tanto en los trabajos como en sus prioridades
4	Reducir las horas extras o demanda no prevista de recursos humanos adicionales
5	Reducir defectos en el trabajo entregado al cliente
6	Mejorar el grado de satisfacción del cliente con el producto o servicio entregado
7	Hacer más visible el estado del trabajo del equipo
8	Mejorar la supervisión del trabajo del equipo
9	Mejorar la gestión de recursos humanos en el equipo
10	Mejorar la comunicación dentro del equipo y con el cliente
11	Mejorar la alineación del trabajo del equipo con los objetivos de la empresa
12	Involucrar en mayor medida al cliente en la planificación, definición y validación del trabajo
13	Mejorar la motivación y compromiso del equipo
14	Evitar costos asociados a la realización de tareas prescindibles o dudosamente rentables
15	Mejorar la sistematización del trabajo
16	Hacer outsourcing total o parcial de ciertas actividades que realiza el equipo
17	Conseguir una certificación (CMMI, ISO 9000, ITIL, u otra)

18	Poder cumplir con la normativa de proyectos establecida en la empresa
19	Promover la mejora continua del proceso empleado por el equipo
20	Reducir el re-trabajo debido a trabajo defectuoso o incompleto detectado por el equipo o por el cliente
21	Reducir el tiempo de entrega al cliente, acelerar el "time to market"

Cada uno de estos objetivos puede tener una importancia diferente para cada equipo, desde no interesarle un objetivo en específico hasta ser su objetivo principal, lo cual puede ir delimitando las prácticas necesarias para lograr cada objetivo en específico.

Sobre los objetivos no hay mucho que decir debido a que estos pueden llegar a ser muy específicos para un equipo de desarrollo, lo que hemos intentado es crear un conjunto generalizado de objetivos que pueda ser útil tanto para equipos esencialmente de desarrollo y/o mantenimiento de productos como para equipos de tramitación de documentos y más.

➤ Prácticas Ágiles.

A continuación se describe el conjunto de prácticas ágiles recopiladas de varios métodos ágiles. Este conjunto está compuesto por un total de 42 prácticas, las cuales están enfocadas a seguir una evolución metodológica en lugar de una revolución. Extraído desde [16].

La lista presentada a continuación ha sido extraída de las metodologías ágiles más populares, como son Kanban, Lean Development, Scrum y Extreme Programming.

Cabe destacar que las prácticas aquí descritas tienen una descripción genérica, es decir, no centrada solamente en el ámbito software. Se le ha dado este enfoque para facilitar la comprensión de las prácticas por parte de gente cuyo trabajo no está relacionado con el desarrollo o mantenimiento de software sino con otros tipos de productos o servicios, ya que el interés por el uso de métodos ágiles se ha extendido a otros ámbitos aunque hay que tener presente que algunas de estas prácticas son exclusivamente para el ámbito de productos, específicamente productos software.

Es importante aclarar que la siguiente lista de prácticas, aunque están numeradas, no tienen un criterio de orden determinado. Además, entre paréntesis se indica el método desde el cual proviene la práctica.

- 1. Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente, (*Extreme Programming, Lean*).**

Hasta que el producto no se comience a utilizar no se tendrá una apreciación precisa del nivel de uso de las características del producto y de la utilidad que ellas ofrecen. Para evitar desperdicio de esfuerzo en desarrollo de características poco utilizadas o subutilizadas respecto de toda su funcionalidad implementada es preferible limitarse a trabajar con el diseño más sencillo que resuelve la necesidad del cliente y con el mínimo conjunto de características que pueden constituir un producto útil para el cliente.

Esta práctica es mencionada como "Diseño simple" en Extreme Programming, y en Lean Development esta práctica está alineada con el principio de eliminación de desperdicios, en este caso referido a trabajo que es prescindible y/o que no es valorado por el cliente. Un término muy en sintonía con esta práctica es el de *Mínimum Viable Product (MVP)*, u otro también similar llamado *Mínimum Marketable Features (MMF)*, los cuales se refiere al conjunto más pequeño posible de características del producto o servicio que por sí mismas podrían aportar valor, constituyendo por ejemplo, una primera entrega de un producto o una primera versión de una nueva funcionalidad de cierta envergadura. En una estrategia similar encontramos el Principio KISS (*"Keep it simple, stupid"*) o el acrónimo YAGNI (*"You aren't gonna need it"*), el primero insiste en no sofisticar las soluciones y el otro en contener la ambición de añadir en el momento actual elementos no imprescindibles, y que incluso en el futuro podrían no ser necesarios.

2. Abordar y ofrecer trabajo terminado de forma incremental, (*Scrum, Extreme Programming*).

No trabajar de forma incremental entraña un alto riesgo, especialmente si el alcance es grande, es decir, cuando solo después de realizar una gran inversión de tiempo o recursos se producirá la entrega que el cliente podrá explotar. Dicho riesgo se asocia a que lo definido inicialmente o con demasiada anticipación cambie antes de la entrega y obligue a hacer re-trabajo para alinear el resultados a los cambios. Un enfoque incremental requiere una definición preliminar de cada unidad de trabajo, pero solo es suficiente para su priorización, dejando la definición completa para el momento justo antes de realizar el trabajo.

3. Realizar entregas frecuentes de unidades de trabajo terminadas, (*Kanban, Lean, Scrum, Extreme Programming*).

Esta práctica es uno de los fundamentos del agilismo. Cuando se trata de un **servicio**, la idea es que continuamente se estén entregando al cliente el

resultado del trabajo, si es posible directamente en la medida que el trabajo se va terminando, excepto en los casos que existan dependencias o divisiones del trabajo que obliguen a entregar el trabajo terminado de forma agrupada por no tener valor para el cliente el contar con un resultado parcial. Cuando se trata de la construcción de un **producto** de cierta envergadura es normal que todo lo que espera el cliente no pueda entregarse en un plazo muy corto.

Esta práctica obligará al cliente a priorizar y organizar el trabajo para ir recibiendo entregas parciales, pero que a la vez le puedan ser útiles. Gracias a esta práctica el cliente y el equipo pueden ir confirmando que se avanza en la correcta dirección con la certeza de que el resultado está siendo ya utilizado, incluso podrían detectar anticipadamente que no vale la pena continuar el trabajo y cancelarlo.

En Extreme Programming se sugiere que las entregas al cliente no deberían tardar más de 3 meses. Es importante destacar la diferencia entre una Entrega (*release*) y un Sprint (*iteración*); una entrega de tres meses podría, por ejemplo, realizarse en base a tres sprints de un mes cada uno. Al final del sprint el cliente puede evaluar un resultado parcial pero sin poder aún explotarlo, en cambio una entrega conlleva el hecho que el cliente puede explotar el resultado. Cuando se está en una situación de mantenimiento de un producto, si las modificaciones no son de gran envergadura los sprints podrían coincidir con entregas, es decir, el resultado del sprint podría ser explotado por el cliente.

4. Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses), (*Scrum, Extreme Programming*).

En un contexto ágil se asume que la planificación puede cambiar para adecuarse a los cambios que se presenten, con lo cual el plan aunque conlleva compromisos de fechas y recursos, debería permitir cambios en cuanto a alcance, para adaptarse a cambios de prioridades y cambios en la

definición del trabajo. En proyectos con muchos cambios durante su realización conviene planificar a corto plazo en detalle y a largo plazo hacerlo de forma global lo suficiente para gestionar el alcance.

5. Acotar el trabajo previsto para un período en base a su estimación y correspondiente coherencia con la capacidad del equipo, (*Scrum, Extreme Programming*).

Si el equipo no tiene una noción fiable de su capacidad (cantidad de trabajo que puede ser abordado con éxito en un período), cada vez que tenga que hacer una previsión o compromiso de trabajo correrá el riesgo de tener diferencias significativas con los datos reales del trabajo una vez efectuado.

Es importante destacar que dependiendo del contexto la estimación del trabajo puede ser realizada con menor intensidad o solo de forma parcial, incluso puede prescindirse de ella. Además, la estimación de un trabajo no es una actividad aislada, siempre estará asociada a la definición del trabajo, con lo cual estimar no debería representar un esfuerzo adicional significativo.

La estimación podría considerarse obligatoria solo si la relación con el cliente involucra un contrato o acuerdo rígido respecto del alcance, plazos y recursos.

6. Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista, (*Scrum, Extreme Programming*).

Esta práctica es esencial en Scrum y Extreme Programming, ambos métodos proponen un proceso iterativo. En Scrum se denominan Sprints a las iteraciones. En Scrum un Sprint no debería durar más de cuatro semanas, en Extreme Programming se sugieren que no sean más de tres semanas.

Esta práctica está condicionada a que efectivamente se pueda e interese agrupar el trabajo para prever o comprometer entregas asociadas a grupos de trabajo terminado. Esta situación es natural cuando se está desarrollando una

primera entrega del producto o una entrega asociada a una modificación de gran envergadura, y cuando además dicha entrega podría tardar más de un mes.

Esta práctica no es aplicable en situaciones en las cuales el equipo no puede agrupar el trabajo para acordarlo con el cliente o ni siquiera puede anticipar cuál es el trabajo que hará en el corto plazo. La organización del trabajo en Sprints, además de ayudar al equipo y al cliente a confirmar que el trabajo va bien encaminado, ofrece otras ventajas tales como: establecer una “cadencia” o ritmo continuo de trabajo que favorece la fiabilidad del equipo, o realización de actividades que conviene aplicarlas en conjunto para un grupo de unidades de trabajo en lugar de hacerlas individualmente para cada una de ellas.

7. Evitar adelantar trabajo que no esté comprometido y/o no esté cercano a su entrega, (*Lean*).

Esta práctica está inspirada en uno de los 7 Mudas (*fuentes de desperdicio*) de Lean Manufacturing, definido como: “Sobre-producción: producir mucho o con demasiada antelación”. En procesos de desarrollo de software (o en ciertos servicios) la consecuencia o problema de NO aplicar esta práctica no es tan evidente como en procesos que generan un producto físico, en los cuales se delata claramente la acumulación de unidades de trabajo terminadas y no entregadas (*vendidas*) al cliente, que probablemente provocará niveles de inventario no deseables en algunos puntos de la cadena de producción. Pero además de dicha posible acumulación de trabajo en curso o terminado, más grave aún puede ser el hecho que el trabajo adelantado llegue a quedar obsoleto y tenga que realizarse re-trabajo para actualizarlo, o incluso tenga que desecharse.

8. Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado, (*Kanban*).

Cuando no se puede anticipar cuál es el trabajo que recibirá el equipo y no se quiere acumular trabajo para ser agrupado y establecer un plazo de entrega, o bien cuando se quiere dar una rápida respuesta a las solicitudes del cliente, especialmente a las pequeñas, en estos casos, el equipo se debería centrar en generar un buen flujo de trabajo terminado. Así pues, con esta práctica el trabajo en la medida que se recibe se ordena con respecto del resto de trabajo pendiente, y respetando dicho orden se va abordando.

Esta práctica es se presenta en el método Kanban con la idea de favorecer el flujo continuo de trabajo terminado, utilizando métricas de flujo tales como: unidades terminadas por unidad de tiempo (*Production Rate*), Cycle Time (*tiempo promedio que tarda una unidad de trabajo en ser procesada hasta terminarse*), y Lead Time (*tiempo promedio que tarda una unidad de trabajo desde que se solicita hasta que se entrega*).

9. Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado, (*Kanban, Scrum*).

Gran parte del éxito de un producto o servicio se basa en la adecuada priorización del trabajo de manera que el cliente vea satisfecha sus solicitudes que le aportan mayor valor, o aquellas que tienen un carácter urgente.

La adecuada definición del trabajo que debe realizarse también es crucial para asegurar que el trabajo realizado satisface las expectativas del cliente. Así pues, el trabajo pendiente debe estar continuamente priorizándose de acuerdo con los cambios u oportunidades que se presenten en el negocio, junto con la entrada de nuevos trabajos solicitados por el cliente. Salvo urgencias, el trabajo antes de ser abordado por el equipo debería ser definido por el cliente y con apoyo del equipo.

10. Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad, (*Kanban*).

Esta práctica es uno de los pilares del método Kanban, en el cual, además de promover la utilización de un tablero Kanban, se destaca en la conveniencia de limitar el trabajo en proceso (WIP) en las actividades que puedan constituir un cuello de botella. La idea de limitar el WIP proviene de la TOC en [11], la cual destaca que una línea de producción tiene un ratio de producción global (*unidades de trabajo terminadas por unidad de tiempo*) que está limitado por el punto de la línea que tiene el menor ratio de producción, y no tiene sentido que otros puntos con mayor ratio que el menor aumenten su ratio. Así pues, al limitar el WIP de las actividades se obliga a tomar medidas que evitan la acumulación de unidades de trabajo en ciertas actividades, por ejemplo, parar por períodos el trabajo en ciertas actividades (*por ejemplo desviando recursos a las actividades con acumulación de trabajo*), o implantar mejoras que permitan elevar o prescindir de la limitación del WIP de una actividad. De todas formas, es importante destacar que estas ideas se originaron en líneas de producción de productos físicos, y deben extrapolarse con cautela en contexto de productos software o de servicios.

11. Formar equipos pequeños y procurar que mantengan sus integrantes, (*Scrum, Extreme Programming*).

Las prácticas ágiles intensifican la comunicación y coordinación frecuente en los equipos de trabajo, y se prefiere que se dichas actividades se realicen cara a cara. Esto es difícil de llevar a cabo si el equipo es de gran tamaño. Scrum sugiere que los equipos deberían estar formados por entre 3 y 9 integrantes. Por otra parte, es importante que el equipo en lo posible se mantenga estable en cuanto a sus integrantes puesto que el equipo irá ganando cohesión.

Los equipos requieren un tiempo para llegar a un nivel adecuado de penetración, para abordar inmediatamente un nuevo proyecto puede resultar más efectivo cambiar el ámbito de trabajo de un equipo ya formado que formar un nuevo equipo, especialmente si se quiere una respuesta rápida y fiable. Además, la frecuente rotación de integrantes puede afectar el desempeño de los equipos.

12. Acotar el ámbito de trabajo del equipo.

Esta práctica tiene el propósito de especializar al equipo en un ámbito de acción. Puede verse desde dos perspectivas: la primera envergadura del producto o servicio, y la segunda, diversidad de trabajos. Cuando se trata un producto o servicio de gran envergadura se debe promover la creación de equipos por partes del producto o servicio (*cuidando que esas partes no tengan demasiada dependencia entre ellas*).

Es importante limitar la cantidad de productos, de servicios y/o de proyectos en los que trabaja simultáneamente cada equipo especialmente si se deben realizar actividades muy diferentes en cada caso. En este aspecto esta práctica es similar a la que propone limitar el WIP (trabajo en proceso en una actividad), la diferencia es que el WIP se refiere a la cantidad de unidades de trabajo en una actividad, en cambio en esta práctica se refiere al número de productos, servicios o proyectos encargados a un equipo.

13. Seguimiento continuo (frecuencia de días, no semanas), (*Scrum, Extreme Programming Kanban*).

La intensidad y frecuencia del seguimiento depende del contexto del trabajo, y en particular depende de si la demanda es o no planificable. Cuando la demanda NO es planificable el seguimiento se basará en la visualización del estado del trabajo y en métricas de flujo de trabajo terminado. Si la demanda es en cierta medida planificable, adicionalmente como seguimiento podemos contrastar el progreso del trabajo con el trabajo restante para el período

planificado, comprobando si es posible cumplir los acuerdos respecto de la entrega.

Mientras menos frecuente sea el seguimiento mayor puede ser la desviación respecto de lo esperado, en proyectos en los cuales existen muchos cambios, el seguimiento continuo y frecuente es esencial. Además, es importante que el equipo al completo se interese por el seguimiento, tanto a nivel de las actividades encargadas a cada integrante como a nivel de los compromisos del equipo.

14. Realizar una reunión diaria del equipo al completo, cara a cara y muy breve, (*Scrum, Extreme Programming*).

Esta reunión se llama “Stand-up Meeting” en Extreme Programming y “Daily Scrum” en Scrum, y se refiere a una reunión que realiza el equipo al iniciar su jornada de trabajo, es una reunión de pie en semi-círculo, en lo posible, frente a la visualización de trabajo del equipo, y dura no más de 15 minutos.

El propósito de estas reuniones es que todo el equipo esté enterado de lo que se está haciendo y sus posibles impedimentos, promoviendo la colaboración entre los miembros del equipo y reforzando el compromiso con los objetivos del equipo, por sobre las responsabilidades individuales.

15. Visualización de todo el trabajo pendiente encargado al equipo, (*Kanban*).

La visualización del trabajo debe ser lo más completa posible y mostrar no solo en qué actividad y quién tiene cada trabajo, sino que debería además permitir que los miembros del equipo sepan los trabajos no terminados en los cuales están participando, incluso cuando todavía no tienen que realizar ninguna acción sobre ellos (*por ejemplo, conocer lo que te llegará como trabajo o en lo que ya has participado pero que aún no está terminado*). Además, dicha visualización debería conllevar el principio de transparencia

indicado por Scrum, según el cual cada miembro del equipo debe tener acceso a toda la información que pueda serle útil del contexto de su trabajo (toda la información de los proyectos o productos en los cuales participa). Las visualizaciones más efectivas están basadas en tableros Kanban, los cuales pueden ser físicos (en una pared) o soportados por alguna herramienta software.

16. Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro. (*Kanban, Scrum*).

La gestión integrada del trabajo pendiente y de su priorización es el primer paso para conseguir que los miembros del equipo estén siempre trabajando en lo más oportuno. Si bien no es lo ideal que un equipo o integrante trabaje en varios productos y/o servicios, es una situación bastante usual y difícil de cambiar. Esto provoca que la gestión integrada de todo el trabajo del equipo sea un desafío, pues un simple tablero Kanban de pared probablemente no será suficiente, una opción más adecuada es disponer de una herramienta software que ofrezca soporte para gestionar de forma integrada varios tableros Kanban.

17. Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ (En el mismo lugar) (*Scrum, Extreme Programming*).

El cliente siempre debe estar disponible para aclarar cuestiones respecto del problema que solicitó resolver.

18. Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente. (*Scrum*).

Es importante cuidar al equipo de los desafíos que le puede suponer su exposición directa a las solicitudes de los clientes y al despropósito que conllevaría que tome decisiones de negocio como las prioridades de las

solicitudes. Así pues, es importante establecer una barrera entre el día a día del equipo y la nueva demanda que se va generando, debe existir un solo interlocutor que represente a la parte cliente y que le proporcione al equipo las prioridades del trabajo pendiente.

19. Realizar reuniones de revisión del trabajo entregado, (*Scrum*).

Estas reuniones corresponden a las “Review Meetings” de Scrum. En estas reuniones el equipo se reúne con representantes de la parte cliente para presentar el trabajo realizado.

Estas reuniones deben servir para remarcar el cumplimiento de lo acordado con el cliente, es importante que el equipo tenga esta oportunidad para divulgar el trabajo realizado.

20. El equipo se auto-organiza y toma las decisiones técnicas. (*Scrum*).

Proviene del concepto “self-organizing team” de Scrum. El equipo debe ser capaz de tomar decisiones técnicas en el trabajo de que deben hacer, cuando y como.

21. Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador. (*Extreme Programming, Scrum*).

El planteamiento de Scrum es radical al respecto: no existe el rol de jefe. El rol Product Owner incluye las responsabilidades de: representar a la parte cliente, definir y priorizar el trabajo, organizando qué trabajo debe abordar el equipo. El rol Scrum Master se encarga de apoyar en cuanto a la implantación y correcta aplicación de la metodología de trabajo y de su mejora continua. Ni el Product Owner ni el Scrum Master indican al equipo quién o cómo debe llevar a cabo el trabajo técnico.

En Extreme Programming existe el rol Manager, caracterizado como facilitador, encargado de resolver inconvenientes que pueda tener el equipo para realizar su trabajo. El planteamiento de Scrum, si bien es ideal, es difícil

de implantar, al menos en contextos donde el rol de jefe tradicional está muy arraigado. El planteamiento de Extreme Programming respecto del rol Manager ofrece menos resistencia y podría servir de evolución hacia al escenario de Scrum.

22. Co-localización de miembros del equipo, todo el equipo trabajando en el mismo espacio físico, (*Extreme Programming*).

Si bien contando con las actuales alternativas de comunicación, especialmente videoconferencias, puede compensarse en gran medida los inconvenientes de no tener un contacto cara a cara in-situ, difícilmente se consigue la misma eficacia de comunicación del contacto in-situ, pues los aspectos gestuales o en general la comunicación no verbal no se consigue transmitir completamente.

23. Contar con un espacio físico de trabajo que promueva y favorezca la interacción entre los miembros del equipo, (*Extreme Programming, Lean*).

Si bien no es una de las 12 prácticas de Extreme Programming, allí donde se pone énfasis en el espacio de trabajo del equipo, un espacio abierto (*sin módulos*), sillas con ruedas, pizarras en las paredes, un área para hablar por teléfono, un área para comer y conversar informalmente, etc. En el contexto Lean también el espacio de trabajo, referido como el “Gemba”, tiene cierto protagonismo.

24. Establecer y comunicar al equipo la visión del producto o servicio, y reforzarla regularmente.

La Visión de un producto o servicio es básicamente su información respecto de: propósito y motivación, principales características, tipos de usuarios, productos o servicios competidores, fortalezas y debilidades, amenazas y oportunidades. Si bien en un enfoque tradicional la recolección y especificación de esta información podría tomar un tiempo considerable y

generar un documento voluminoso. Desde una perspectiva más ágil la idea sería conseguir un documento muy simplificado, con lo esencial, o bien podría bastar con que el equipo esté en conocimiento de dicha información aunque no esté explícitamente escrita.

En Extreme Programming la práctica Metáfora tiene un propósito similar a la Visión, y consiste en que el equipo conozca y sea capaz de relatar lo esencial del producto que se está desarrollando. En Scrum el concepto de Goal del sprint también tiene algo en común con la Visión, y permite destacar dentro de todo el trabajo de un sprint aquello que es lo esencial, representado por algunos de los ítems del sprint.

Conocer la Visión del producto le proporciona al equipo un contexto para su trabajo, lo cual debería influir positivamente en su motivación y compromiso.

25. Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo, (*Scrum*).

Esta práctica corresponde al concepto de equipo cross-functional planteado en Scrum. Es indudable que la concentración de ciertas funciones en ciertas unidades en una empresa ofrece las ventajas de la especialización y permite realizar economías de escala, pero como contraparte puede constituir un cuello de botella para terminar trabajo que depende en parte de la participación de dichas unidades, y suele ralentizar el flujo de trabajo por el necesario protocolo de interacción con dichas unidades.

26. Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades, aunque puedan ser especialistas en alguna(s) de ellas.

Los métodos ágiles promueven los roles más genéricos para acotar lo menos posible las actividades que puede desempeñar un integrante del equipo. En Scrum en el equipo sólo existe el rol "Desarrollador" el cual se encarga de

todo el trabajo técnico. En Extreme Programming respecto del trabajo técnico se reconoce el rol Programador (el cual desde la perspectiva tradicional sería un analista, programador y tester en los niveles unitario, de integración y de sistema), y el rol tester (encargado de las pruebas de aceptación).

27. Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente, (*Extreme Programming*).

Aunque aparentemente pareciera que es suficiente que una especificación del trabajo incluya una combinación de texto descriptivo y algunos modelos u otros complementos, si no se establece con precisión el criterio de aceptación desde la perspectiva del cliente siempre habrá lugar para dichas situaciones de no conformidad. Ese criterio de aceptación no es más ni menos que un conjunto de pruebas que el cliente aplicaría para considerarse satisfecho con la entrega. Con lo cual en el mismo momento de la definición del trabajo deben quedar establecidas y acordadas con el cliente las pruebas de aceptación.

28. Documentar, pero solo lo estrictamente necesario, que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla, (*Lean, Scrum, Extreme Programming*).

Lean Development insiste en eliminar toda forma de desperdicio (“*waste*”), es decir, trabajo que no aporta valor para el cliente. La documentación suele ser una fuente importante de desperdicio de esfuerzo. Cuando el trabajo sufre constantes cambios sobre la marcha, pretender mantener actualizada la documentación puede ser un gran desperdicio. La mínima especificación necesaria para llevar a cabo un trabajo es conocer el criterio de aceptación que aplicará el cliente, cualquier otra especificación requerida para la entrega podría generarse de forma detallada y completa una vez que el resultado adquiera cierta estabilidad, evitando o reduciendo dicho desperdicio.

29. Establecer pautas para gestionar convenientemente el re-trabajo.

El re-trabajo corresponde al esfuerzo gastado por volver a repetir alguna actividad asociada a un trabajo por haberse detectado defectos durante el proceso. Evidentemente el re-trabajo tiene una connotación negativa y constituye una medida importante para evaluar el desempeño de un proceso. Si bien, y dependiendo del contexto de trabajo, el re-trabajo suele ser inevitable, el objetivo es reducirlo a su mínima expresión.

Esta práctica no proporciona la solución sino que más bien enfatiza el hecho que el equipo debe tener claramente establecidas pautas para actuar frente a situaciones de re-trabajo. Cuestiones tales como la priorización del trabajo asociado a corrección de fallos, o cualquier otro tratamiento específico para el re-trabajo debe estar consensuado por todo el equipo. Estas pautas deben evaluarse y mejorarse continuamente.

30. Que exista un líder de mejora de proceso disponible para el equipo, (*Scrum, Extreme Programming*).

Esta práctica corresponde a la implantación del rol “Scrum Máster” promovido por Scrum o el similar llamado “Coach” en Extreme Programming.

No es fácil implantar un sistema de mejora continua, se requiere motivación y compromiso de todos los miembros del equipo, pero además, un complemento imprescindible es el apoyo institucional materializado en un líder de mejora que pueda asesorar al equipo en la mejora de sus métodos de trabajo. Este líder debe estar en estrecho contacto con el equipo.

31. Establecimiento de estándares para el trabajo técnico del equipo,
(Extreme Programming).

Esta práctica se propone en Extreme Programming como “Estándares de Programación”. La estandarización es fundamental para independizar el trabajo de la persona que puede realizarlo y a su vez garantizar un nivel de calidad asociado al proceso estandarizado.

32. Reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora Continua, *(Scrum).*

Estas reuniones corresponden a las “Retrospective Meetings” de Scrum. Debe existir un mecanismo establecido y regular para promover la auto-crítica respecto del desempeño del equipo y de su método de trabajo, esto ayuda a evitar que imperfecciones o ineficiencias conocidas pero no graves se mantengan por largo tiempo.

33. Acordar y definir qué se entiende por trabajo terminado, tanto para las actividades realizadas por el equipo como respecto de las entregas al cliente, *(Scrum).*

Esta práctica corresponde a lo que Scrum denomina “Definition of DONE”, refiriéndose al hecho que el equipo y el cliente deben tener acordado qué se entiende por trabajo terminado. Esto es importante pues para cada contexto de cliente-equipo-producto/servicio, el acuerdo respecto a qué se entiende por trabajo terminado puede variar, afectando a aspectos tales como la documentación que debe acompañar a la entrega, niveles de pruebas superados, o cualquier protocolo o actividades asociadas a la terminación de un trabajo.

34. Trabajo o actividades realizadas en conjunto por dos o más integrantes.
(Extreme Programming).

Proviene de la práctica “Programación en Parejas” de Extreme Programming.

Muchas veces cuando un trabajo es de cierta envergadura y/o complejidad para poder abordarlo entre varias personas se intenta a priori dividirlo, lo cual no suele ser sencillo y conlleva desafíos de coordinación. Una alternativa interesante y más sencilla en cuanto a coordinación es considerar que en algunas actividades trabaje más de una persona. Los encargados de realizar la actividad, al momento de llevarla a cabo podrían repartirse el trabajo si les resulta conveniente, pero con una coordinación resuelta en dicho momento y entre ellos.

35. No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo, (*Extreme Programming*).

Esta práctica corresponde a lo que aparece en el principio del Manifiesto Ágil referente a “desarrollo sostenible” y “paz constante”. Extreme Programming destaca este aspecto en su práctica “Semana de 40 horas”.

36. Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo.

Deben existir unas pautas básicas para la utilización de los diversos medios de comunicación (*cara a cara, mensajería instantánea, teléfono, videoconferencia, email o reuniones*) para así conseguir un mejor desempeño global. La clave está en, por un lado, consensuar aproximadamente qué debe considerarse tan urgente como para justificar estas interrupciones sin optar por otra alternativa, y por otro lado, cuándo una interrupción no es tal, es decir, el integrante interrumpido realmente no se ve perjudicado por la interrupción ya sea por estar disponible o porque la interrupción es muy breve.

37. Establecer una disciplina de aprovechamiento de las reuniones.

La introducción de prácticas ágiles conlleva intrínsecamente mayor comunicación entre los miembros del equipo y con el cliente. El entusiasmo de los miembros del equipo ante lo que puede ser una oportunidad nueva de apertura a su participación y su aportación debe ser adecuadamente conducido. Gran parte de dicha comunicación y participación se canaliza a través de reuniones. Pero estas reuniones pueden llegar a ser muy improductivas si no existe una buena disciplina para llevarlas a cabo.

38. Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios, (*Extreme Programming*).

Las actividades de pruebas son en general muy atractivas para ser automatizadas pues el esfuerzo invertido en ellas puede rentabilizarse en la ejecución frecuente de pruebas de regresión.

Cuando un producto está mejorándose continuamente y se hacen entregas frecuentes al cliente, todo este esfuerzo puede incluso llegar a ser valorado negativamente por el cliente si se estropean funcionalidades ya disponibles antes de la última entrega. La automatización está en bastante sintonía con el agilismo, sin embargo, conseguir la infraestructura de soporte para automatizar procesos (herramientas y otros recursos), debe ser evaluado en términos de la inversión y tiempo que se requerirá para ponerla en aplicación y del ahorro o mejora que se conseguirá.

La automatización de pruebas es una de las prácticas que requiere mayor preparación y su rentabilización se consigue a mediano y largo plazo. El testing es un área muy amplia, existiendo diversidad de tipos de pruebas (*unitarias, de integración, de aceptación, de rendimiento, etc.*), multitud de herramientas y muchísimas estrategias para implantar el testeo automatizado en un determinado contexto.

39. Postergar hasta último momento la asignación del encargado de realizar una actividad.

Es importante tener en cuenta la asignación de trabajo a los miembros del equipo debido a que una asignación no adecuada podría generar desniveles en la carga de trabajo de cada integrante.

El trabajo del equipo, dependiendo del contexto, puede ser altamente dinámico, las unidades de trabajo pueden sufrir esperas no previstas y cambios que alteran su estimación, los integrantes del equipo pueden ser interrumpidos o tener que cambiar a otros trabajos por cambios de prioridades o por verse obligados a distribuir su capacidad en varios trabajos asignados. La pre-asignación demasiado anticipada conlleva un riesgo importante en cuando a ineficiencias en el aprovechamiento de los recursos humanos.

40. Integrar de forma continua en el producto o servicio el trabajo terminado, (*Extreme Programming*).

Proviene de la práctica “Integración Continua” de Extreme Programming. Cuando varias personas colaboran en un trabajo solapándose en cuanto al resultado que debe obtenerse, y particularmente cuando se realizan cambios en paralelos en partes compartidas, es necesario gestionar los cambios y versiones de los artefactos generados, y además, tener una disciplina de integración continua para que todo el equipo disponga de la información más actualizada cuando está realizando su trabajo. Esto no solo es aplicable al ámbito del código fuente en un equipo de desarrollo de software sino también al trabajo colaborativo sobre cualquier tipo de documentos, en equipos de cualquier ámbito.

41. Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo, (*Extreme Programming*).

Se refiere a la práctica “Propiedad Colectiva” de Extreme Programming. Existe consenso en los métodos ágiles respecto de que los equipos deben ser pequeños (a modo orientativo: menos de 10 integrantes). En estos casos lo recomendado sería organizar equipos pequeños coordinados para abordar partes del trabajo en el producto o servicio. Asociado a esto, en el ámbito de Scrum se habla de hacer “Scrum de Scrums”. Al margen de los desafíos asociados a la coordinación y centrándonos en la práctica que estamos comentando, hay que destacar que el alcance de “propiedad colectiva” que se promueve está limitado por el alcance del ámbito de responsabilidad encargada al equipo sobre el producto o servicio.

42. Mejorar continuamente la organización interna del producto para facilitar su mantenimiento, (*Extreme Programming*).

Corresponde a la práctica “Refactoring” de Extreme Programming, que consiste en mejorar la arquitectura del producto sin cambiar su funcionalidad.

Una buena arquitectura interna del producto es clave para facilitar el mantenimiento, las pruebas y la reutilización, con lo cual debería ser un objetivo de cualquier metodología, sin embargo, las estrategias para conseguirlo pueden ser muy diferentes.

La estrategia tradicional es invertir un importante esfuerzo al inicio del desarrollo del producto (especificando y modelando intensamente), antes de comenzar a construirlo. Por contraparte, la estrategia ágil es iniciar rápidamente la construcción del producto para poder cuanto antes ir consiguiendo la confirmación del cliente, al menos de lo construido hasta el momento.

Capítulo 5. AGILE Roadmap

➤ Concepto teórico.

Una Agile Roadmap es una guía para la implantación de prácticas ágiles en un equipo de trabajo, lo que también equivaldría al Improvement Backlog de un equipo y es lo mismo que una lista ordenada de mejoras de proceso que se quiere implantar en el equipo a lo largo del tiempo.

Antes de poder elaborar correctamente un Agile Roadmap es necesario conocer las prácticas ofrecidas por las diferentes metodologías ágiles. En un Agile Roadmap se reconoce indirectamente que es difícil y no recomendable la mayoría de veces implantar una gran cantidad de prácticas a la vez.

Cabe destacar que un Agile Roadmap no intenta “casar” al equipo con una metodología en particular, sino más bien centra el foco en la aplicación de las prácticas sin importar la metodología de donde proceda.

Para que un Agile Roadmap sea lo más eficiente posible, este debe ser el resultado del diagnóstico y evaluación del contexto del equipo de trabajo al cual será aplicado para la implantación del agilismo.

A continuación se muestra un conjunto de pasos recomendados para el diagnóstico y evaluación del equipo para generar su Agile Roadmap. [17]

- 1- Seleccionar el equipo de trabajo en el cual se aplicarán las prácticas ágiles.
- 2- Estudiar los productos, servicios y/o proyectos encargados al equipo seleccionado.
- 3- Evaluar globalmente el tipo de trabajo encargado al equipo y su posible diversidad.
- 4- Establecer los objetivos que pretende la iniciativa de mejora de proceso.
- 5- Acotar las prácticas candidatas.
- 6- Establecer el nivel de aplicación de cada práctica.
- 7- Establecer el nivel de dificultad que tendrían los desafíos de implantación de cada práctica.

8- Evaluar la aplicabilidad de cada práctica.

Como una ayuda adicional para la elaboración del Agile Roadmap del equipo, hemos desarrollado una aplicación web también llamada Agile Roadmap, la cual cumple con la automatización parcial de los pasos anteriormente mencionados y permite conseguir un Agile Roadmap.

➤ **AGILE Roadmap.**

Debido a la dificultad del proceso de selección de prácticas ágiles y las pocas herramientas existentes en este aspecto, hemos desarrollado un sitio web que permite a los equipos de trabajo elaborar dicho documento con mayor facilidad.

Luego de realizar diversas búsquedas en la web para encontrar herramientas que facilitaran el proceso de selección de prácticas ágiles para su posterior implantación, no he encontrado una herramienta que ofrezca de manera fácil y sencilla la elaboración de un Agile Roadmap.

La aplicación AGILE Roadmap está desarrollada de una manera simple y fácilmente entendible a cualquier equipo de trabajo, siendo del ámbito software o no. Está estructurada en tres pasos.

- 1) Selección de los objetivos de interés para el equipo o producto/servicio.
- 2) Especificar el nivel de aplicación que tiene cada práctica en el equipo, es decir, que tan aplicada esta esa práctica actualmente o si descartan aplicarla por algún motivo.
- 3) Clasificación y evaluación de los desafíos que puede presentar cada práctica.

Luego de completados estos pasos y una previa especificación de datos sobre el equipo, se completa el proceso tras la obtención del Agile Roadmap.

En la *ilustración 9* se presenta el diagrama de la base de datos que utiliza el sitio web AGILE Roadmap.

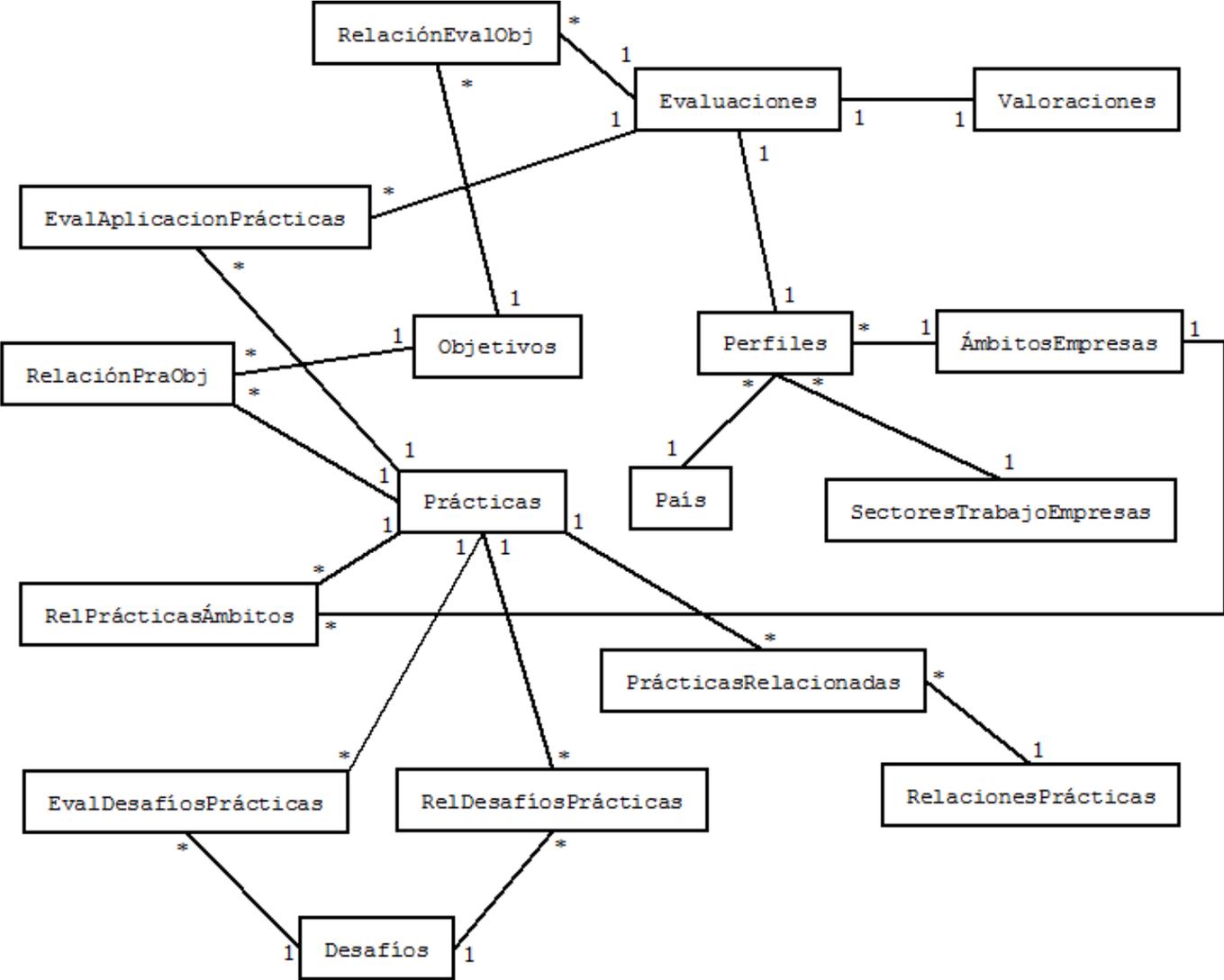


Ilustración 9 - Diagrama de la base de datos de AGILE Roadmap.

Descripción de tablas y atributos

Perfiles

- IdPerfil: *integer*; es el identificador y clave primaria de la tabla.
- Nombre: *nvarchar*; almacena el nombre del producto/servicio evaluado.
- FechaCreacion: *datetime*; fecha en la que se crea el perfil.
- IdAmbito: *integer*; identificador del ámbito en el que trabaja el equipo.
- Correo: *nvarchar*; almacena el correo electrónico.
- NumeroMiembros: *integer*; almacena la cantidad de miembros del equipo.
- IdSector: *integer*; identificador del sector al que pertenece el producto/servicio.
- IdPaisIP: *integer*; identificador del país desde donde se realiza la evaluación.
- IP: *nvarchar*; dirección IP de la máquina desde donde se realiza la evaluación.
- Planificable: *bit*; indica si el equipo planifica el trabajo según capacidad o no.

Evaluaciones

- IdEvaluacion: *integer*; identificador y clave primaria de la tabla.
- IdPerfil: *integer*; identificador del perfil asociado a la evaluación.
- Fecha: *datetime*; fecha en la que se inicia la evaluación en el paso 1.
- Completa: *bit*; indica si la evaluación se completó o no.
- FechaPaso2: *datetime*; almacena la fecha en la que se completa el paso 2.
- FechaPaso3: *datetime*; almacena la fecha en la que se completa el paso 3.

Prácticas

- IdPractica: *integer*; identificador y clave primaria de la tabla.
- Posicion: *integer*; almacena un número usado para ordenar las prácticas.
- Origen: *nvarchar*; almacena los métodos de donde proviene.
- Nombre: *nvarchar*; almacena el nombre de la práctica.
- EsfuerzoPrep: *integer*; almacena el valor de esfuerzo de preparación de la práctica.
- Descripcion: *nvarchar*; almacena la descripción de la práctica.
- Activa: *bit*; indica si la práctica esta activa o no.
- SoloAplicableAProducto: *bit*; indica si la práctica es aplicable a productos.
- SoloAplicableSiPlanificable: *bit*; indica si la práctica es planificable o no.

Objetivos

IdObjetivo: *integer*; identificador numérico y clave primaria de la tabla.

Posicion: *integer*; almacena un número usado para ordenar los objetivos.

Nombre: *nvarchar*; almacena el nombre del objetivo.

Descripcion: *nvarchar*; almacena la descripción del objetivo.

Activo: *bit*; indica si el objetivo está activo o no.

Desafíos

IdDesafio: *integer*; identificador numérico y clave primaria de la tabla.

Posicion: *integer*; almacena un número usado para ordenar los desafíos.

Nombre: *nvarchar*; almacena el nombre del desafío.

Descripcion: *nvarchar*; almacena la descripción del objetivo.

DependienteDeLaPractica: *bit*; indica si es dependiente de la práctica o no.

Activo: *bit*; indica si el desafío está activo o no.

ÁmbitosEmpresas

IdAmbito: *integer*; identificador numérico y clave primaria de la tabla.

Posicion: *integer*; almacena un número usado para ordenar los ámbitos.

Descripcion: *nvarchar*; almacena la descripción del ámbito.

SectoresTrabajoEmpresa

IdSector: *integer*; identificador numérico y clave primaria de la tabla.

Descripcion: *nvarchar*; almacena la descripción del sector de trabajo.

Posicion: *integer*; almacena un número usado para ordenar los sectores.

Valoraciones

IdValoracion: *integer*; identificador numérico y clave primaria de la tabla.

IdEval: *integer*; identificador de la evaluación asociada a la valoración del sitio.

Valoracion: *integer*; almacena el puntaje asignado por el equipo.

Comentario: *nvarchar*; almacena un comentario que quiera añadir el usuario.

País

IdPaís: *integer*; identificador numérico y clave primaria de la tabla.

ISONUM: *integer*; almacena el número ISO asignado al país.

ISO2: *nvarchar*; almacena el código ISO de dos caracteres.

ISO3: *nvarchar*; almacena el código ISO de tres caracteres.

NombreES: *nvarchar*; almacena el nombre del país.

EvalDesafíosPrácticas

IdPractica: *integer*; almacena el identificador de la práctica relacionada.

IdDesafio: *integer*; almacena el identificador del desafío relacionado.

IdEvaluacion: *integer*; almacena el identificador de la evaluación relacionada.

Dificultad: *integer*; valor de dificultad asignado por el usuario al desafío.

EvalAplicaciónPrácticas

idEvaluacion: *integer*; almacena el identificador de la evaluación relacionada.

idPractica: *integer*; almacena el identificador de la práctica relacionada.

NivelAplicacion: *integer*; valor asignado como nivel de aplicación de la práctica.

RelDesafíosPrácticas

IdDesafio: *integer*; identificador del desafío asociado.

IdPractica: *integer*; almacena el identificador de la práctica relacionada.

RelPrácticasÁmbitos

IdPractica: *integer*; almacena el identificador de la práctica relacionada.

IdAmbito: *integer*; almacena el identificador del ámbito relacionado.

RelaciónPraObj

IdRelacion: *integer*; identificador y clave primaria de la tabla.

IdPractica: *integer*; almacena el identificador de la práctica relacionada.

IdObjetivo: *integer*; almacena el identificador del objetivo relacionado.

Contribucion: *integer*; valor de la contribución de la práctica al objetivo.

RelaciónEvalObj

IdRelacion: *integer*; identificador y clave primaria de la tabla.

IdEvaluacion: *integer*; almacena el identificador de la evaluación relacionada.

IdObjetivo: *integer*; almacena el identificador del objetivo relacionado.

Relevancia: *integer*; valor de la relevancia asignada al objetivo por el usuario.

RelacionesPrácticas

IdRelacionPracticas: *integer*; identificador y clave primaria de la tabla.

Nombre: *nvarchar*; nombre de la relación.

NombreRelacionInversa: *nvarchar*; relación inversa.

PrácticasRelacionadas

IdPracticaDesde: *integer*; identificador de la práctica desde donde se relaciona.

IdPracticaHacia: *integer*; identificador de la práctica hacia donde se relaciona.

IdRelacionPracticas: *integer*; identificador de la relación asociada.

Pasos para realizar una evaluación utilizando AGILE Roadmap.

➤ Paso 0 o Inicio

Solo por ponerle un nombre y no clasificarlo como un paso más. Es la pantalla principal de la aplicación donde se solicita al usuario un conjunto reducido de datos relativos al equipo (*Producto o Servicio, Número de miembros, Ámbito, Situación de Planificación y sector de la Empresa*) para iniciar con la elaboración de su AGILE Roadmap como se puede apreciar en la *ilustración 10*.

La información recolectada en esta etapa influye al listado de prácticas recomendadas para el equipo al finalizar la evaluación, esto es debido a que existen prácticas que dependiendo la cantidad de miembro o si el trabajo se planifica o no, estas pueden incluirse en el AGILE Roadmap o no.



AGILE ROADMAP

Una Iniciativa de [Tune-up Process](#) [Créditos](#)

AGILE Roadmap te permite en 3 simples pasos obtener una lista ordenada de prácticas ágiles recomendadas para implantar en un equipo de trabajo (ver [un ejemplo](#) del informe completo obtenido al finalizar). Te recomendamos primero recorrer los pasos y obtener una evaluación borrador en 2 minutos, para que así te hagas una idea de la información que debes proporcionar. Para una evaluación con mayor detenimiento y comentada con otros colegas te sugerimos reservar al menos una hora de reunión :-). El conocimiento tras AGILE Roadmap proviene de nuestra experiencia en la aplicación de la metodología y herramienta [Tune-up Process](#) y de los métodos ágiles Kanban, Lean Development, Scrum y Extreme Programming. Hemos intencionalmente planteado estas prácticas de forma genérica para que también puedan ser entendidas y aprovechadas por equipos que no trabajan en desarrollo de software. Puedes consultar nuestra estrategia de implantación en artículos disponibles en [blog.tuneupprocess.com](#). Si te interesa contar con nuestro apoyo para realizar tu diagnóstico y evaluación de prácticas ágiles, y para asesorarte cómo comenzar su implantación [contáctanos](#).

Producto o Servicio *

Número de Miembros *

Ámbito

Situación de Planificación

Sector de Empresa

Iniciar

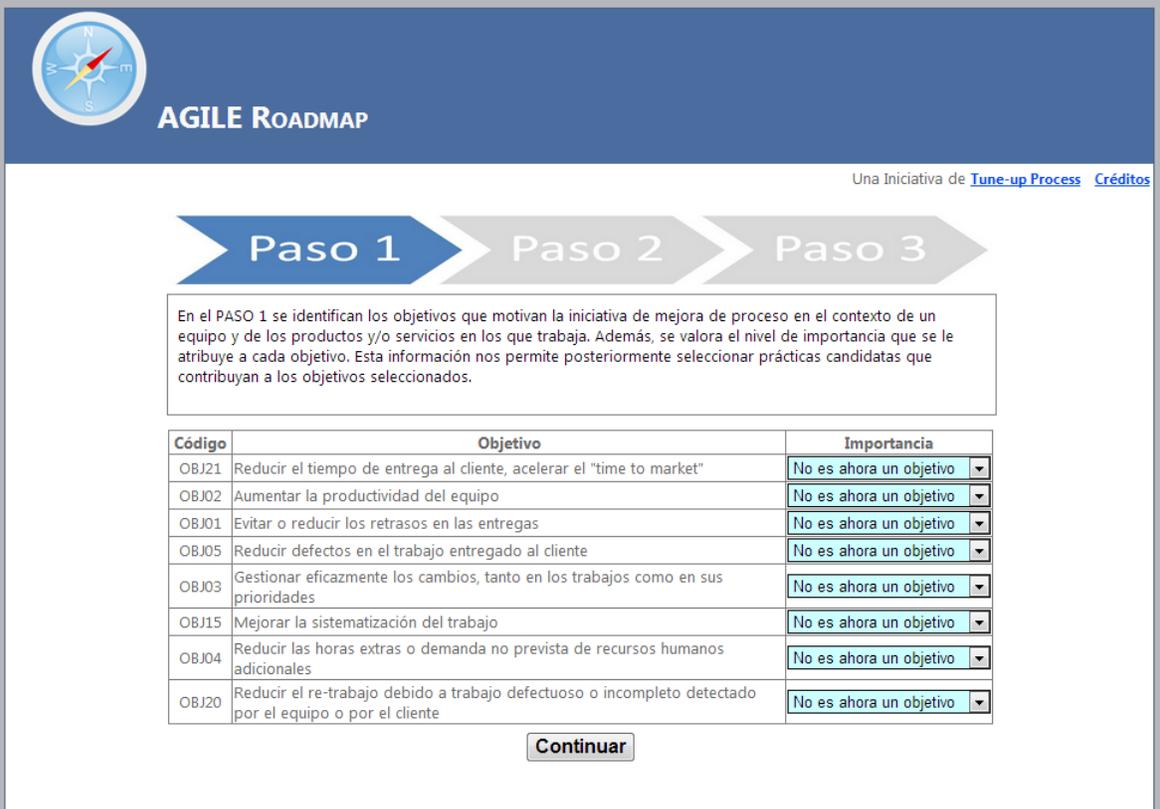
 Agile Roadmap by [Patricio Letelier](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#). Creado a partir de la obra en [blog.tuneupprocess.com](#).

Ilustración 10 - Pagina principal de AGILE Roadmap

➤ Paso 1

En este paso el equipo puede seleccionar los objetivos en los cuales desea enfocarse asignando a cada uno un nivel de importancia (*No es ahora un objetivo, Podría interesarnos, Queremos conseguirlo, Debemos conseguirlo o Es vital conseguirlo*), lo cual influye de manera directa en las prácticas recomendadas al final de la evaluación ya que estas serán valoradas según los intereses indicados por el equipo en cada uno de los pasos durante todo el proceso de evaluación.

Como se puede apreciar en la *ilustración 11*, se muestra al usuario el conjunto de objetivos disponibles en la base de datos de la aplicación para su valoración y evaluación.



Una Iniciativa de [Tune-up Process](#) [Créditos](#)

Paso 1 Paso 2 Paso 3

En el PASO 1 se identifican los objetivos que motivan la iniciativa de mejora de proceso en el contexto de un equipo y de los productos y/o servicios en los que trabaja. Además, se valora el nivel de importancia que se le atribuye a cada objetivo. Esta información nos permite posteriormente seleccionar prácticas candidatas que contribuyan a los objetivos seleccionados.

Código	Objetivo	Importancia
OBJ21	Reducir el tiempo de entrega al cliente, acelerar el "time to market"	No es ahora un objetivo ▼
OBJ02	Aumentar la productividad del equipo	No es ahora un objetivo ▼
OBJ01	Evitar o reducir los retrasos en las entregas	No es ahora un objetivo ▼
OBJ05	Reducir defectos en el trabajo entregado al cliente	No es ahora un objetivo ▼
OBJ03	Gestionar eficazmente los cambios, tanto en los trabajos como en sus prioridades	No es ahora un objetivo ▼
OBJ15	Mejorar la sistematización del trabajo	No es ahora un objetivo ▼
OBJ04	Reducir las horas extras o demanda no prevista de recursos humanos adicionales	No es ahora un objetivo ▼
OBJ20	Reducir el re-trabajo debido a trabajo defectuoso o incompleto detectado por el equipo o por el cliente	No es ahora un objetivo ▼

Continuar

Agile Roadmap by [Patricio Letelier](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#).
Creado a partir de la obra en blog.tuneupprocess.com.

Ilustración 11 – Paso 1 del proceso de evaluación en la aplicación AGILE Roadmap

➤ Paso 2

En este paso el sistema presenta un listado de prácticas ágiles seleccionadas de la base de datos de manera automatizada dependiendo de su contribución a los objetivos especificados por el equipo en el paso anterior. La *ilustración 12* muestra una captura de pantalla de este paso.

Para cada práctica el equipo debe especificar el nivel de aplicación que tendría cada una de las mostradas en este paso (*Descartamos aplicarla, No se está aplicando, Se aplica parcialmente o Completamente aplicada*), cuando se refiere al nivel de aplicación, es a qué nivel se está aplicando en el equipo actualmente dicha práctica.

También para facilitar la navegación y aclarar dudas con las prácticas, está disponible una descripción detallada de cada práctica mostrada al equipo, haciendo clic en el icono con el signo de agregado en la primera columna a la izquierda (🔍) y haciendo clic en el icono con la diana (🎯) podrá ver los objetivos a los que contribuye.

Una Iniciativa de [Tune-up Process](#) [Créditos](#)

Paso 1 Paso 2 Paso 3

El PASO 2 presenta una lista de prácticas que contribuyen a los objetivos seleccionados. Para cada práctica se debe identificar su nivel actual de aplicación. Las prácticas que no están siendo aplicadas o que se aplican de forma parcial ofrecen un margen de mejora que será valorado en el resultado.

	Código	Práctica	Nivel de Aplicación
🔍 🎯	PRA24	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	No se está aplicando
🔍 🎯	PRA08	Visualización de todo el trabajo pendiente encargado al equipo	No se está aplicando
🔍 🎯	PRA30	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	No se está aplicando
🔍 🎯	PRA47	Establecer y comunicar al equipo la visión del producto o servicio, y reforzarla regularmente	No se está aplicando
🔍 🎯	PRA23	Realizar entregas frecuentes de unidades de trabajo terminadas	No se está aplicando
🔍 🎯	PRA40	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente	No se está aplicando

Continuar

Agile Roadmap by [Patricio Letelier](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#)
Creado a partir de la obra en [blog.tuneupprocess.com](#)

Ilustración 12 - Paso 2 del proceso de evaluación en la aplicación AGILE Roadmap

➤ Paso 3

Aquí se presentan los desafíos que pudiera enfrentar el equipo al momento de la implantación de algunas de las prácticas recomendadas, cada práctica es presentada en asociación con los posibles desafíos que puede enfrentar al momento de la implementación de la misma. La *ilustración 13* muestra una captura de pantalla de este paso.

En esta etapa la evaluación de los desafíos sigue la misma manera de evaluación de los pasos anteriores. El equipo debe especificar el nivel de dificultad de cada uno de los desafíos de las prácticas (“*Ninguna, No existe este desafío*”, “*Fácilmente superable*”, “*Tendría algunos impedimentos*”, “*Sería difícil lograrlo*” o “*imposible de conseguir*”), lo cual es variable debido a que unas prácticas pueden presentar más desafíos que otras.

Una Iniciativa de [Tune-up Process](#) [Créditos](#)

Paso 1 Paso 2 Paso 3

El PASO 3 presenta los desafíos de las prácticas candidatas. Con esta información se favorecerá la implantación de prácticas que si bien no contribuyen mucho a ciertos objetivos, podrían ser fáciles de implantar. Así también, se relegarán prácticas que presenten demasiados inconvenientes en cuanto a los desafíos para implantarlas. OJO! muchos desafíos son comunes a varias prácticas compartiendo su valor de dificultad, con lo cual cambiando dicho valor en una práctica se cambiará también en las otras.

Código	Prácticas y sus desafíos	Desafíos	Dificultad
PRA24	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado		
CHA30		Tener un representante de la parte cliente que ofrezca alta disponibilidad para que el equipo interactúe con él	Tendría algunos impedimentos
CHA41		Conseguir un único y buen representante de la parte cliente	Tendría algunos impedimentos
PRA05	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico		
CHA16		Conseguir reunir en el mismo espacio físico a todos los que intervienen en las actividades necesarias para realizar el trabajo	Tendría algunos impedimentos
CHA22		Disponer de un espacio acondicionado para el trabajo en equipo	Tendría algunos impedimentos
CHA50		Conseguir el apoyo de la dirección para implantar esta práctica	Tendría algunos impedimentos

Continuar

Agile Roadmap by [Patricio Letelier](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#).
Creado a partir de la obra en [blog.tuneupprocess.com](#).

Ilustración 13 - Paso 3 del proceso de evaluación en la aplicación AGILE Roadmap.

➤ **Resultado**

En esta etapa de la evaluación, se muestra al equipo el listado de prácticas recomendadas que compone lo que es su Agile roadmap, el resultado como he ido explicando en cada uno de los pasos está relacionado con los datos que el equipo ha proporcionado a la aplicación y la base de conocimiento con la que consta dicha aplicación.

En este resultado se muestra al equipo las prácticas recomendadas ordenadas de mayor a menor por el porcentaje de evaluación que obtuvo cada una el cual es obtenido con una fórmula que explicare más adelante. También se muestran las prácticas recomendadas que el equipo decidió descartar y las ya aplicadas completamente.

Por último y no menos importante, se muestra la correlación que tienen las prácticas entre sí, es decir, si una práctica apoya o requiere de otra y viceversa. Esto es importante para que el equipo tenga en consideración que puede haber descartado una práctica que apoya o es requerida por otras prácticas lo que podría perjudicar su implantación.

Cabe destacar que en la aplicación se da la posibilidad al equipo de obtener su “Roadmap completo” por correo, el cual está compuesto de todos los datos introducidos durante el proceso de evaluación, lo que es muy útil al momento de hacer comparaciones, dicha posibilidad está sujeta a la valoración por parte del equipo de la aplicación AGILE Roadmap.

La *ilustración 14* muestra un ejemplo ficticio de cómo se muestra la información al usuario/equipo.



Obtener Roadmap completo por email

Datos del perfil Agile Roadmap	
	Tuesday, June 11, 2013 - 1:05 PM
Producto o Servicio	tune-up
Número de Miembros	2
Ámbito	Desarrollo y/o mantenimiento de productos
Organización del trabajo	Se planifica considerando capacidad
Sector de Empresa	Tecnología y Software

Evaluación	Aplicación	Código	Nombre de la Práctica	Relaciones
1.24	RECOMENDADA	PRA26	Realizar reuniones de revisión del trabajo entregado	es apoyada por PRA05, PRA09, PRA30, PRA45; requiere de PRA33
1.01	RECOMENDADA	PRA33	Acordar y definir qué se entiende por trabajo terminado, tanto para las actividades realizadas por el equipo como respecto de las entregas al cliente	es apoyada por PRA02, es requerida por PRA26
0.79	RECOMENDADA	PRA02	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	apoya a PRA33
0.28	RECOMENDADA	PRA23	Realizar entregas frecuentes de unidades de trabajo terminadas	es apoyada por PRA03
0.18	APLICADA	PRA37	Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	es apoyada por PRA18, es requerida por PRA13

La columna "Evaluación" muestra un valor según el cual se ordenan las prácticas recomendadas. Dicho valor se calcula considerando los siguientes aspectos: la contribución de la práctica a los objetivos seleccionados, la importancia de cada objetivo, el nivel de aplicación actual de la práctica, el esfuerzo que llevaría la preparación de la práctica y los desafíos para la aplicación de la práctica. La lista de prácticas ágiles recomendadas tiene limitaciones obvias, para el diagnóstico solo hemos recogido información muy general y estructurada. La idea es que este resultado sirva como Roadmap preliminar, el cual debería ser completado y adaptado considerando más detalles del contexto del equipo y de su trabajo en ciertos productos y/o servicios.



Agile Roadmap by [Patricio Letelier](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#).
Creado a partir de la obra en [blog.tuneupprocess.com](#).

Ilustración 14 - Hoja de resultado del proceso de evaluación en la aplicación AGILE Roadmap

Para los equipos de desarrollo es muy importante contar con un Agile roadmap para la implantación del agilismo debido a que este facilita el trabajo de obtención de las prácticas adecuadas para dicho equipo o para un producto/servicio en específico en el cual este trabajando.

En el *anexo 1* se encuentra un informe generado por la aplicación AGILE Roadmap, este informe tiene la finalidad de facilitar al equipo saber los valores especificados en cada uno de los pasos realizados durante el proceso de elaboración de su Agile Roadmap.

➤ **Formula implementada en AGILE Roadmap.**

El proceso de elaboración de un agile roadmap lo más preciso posible está sujeto a los intereses del equipo interpretados de manera correcta y manejados adecuadamente. En cada uno de los pasos de la aplicación AGILE Roadmap se captura información que es procesada y evaluada según una formula específica para determinar la importancia de cada objetivo y que tan importante y/o factible resultaría implementar una práctica u otra.

Para la evaluación hemos utilizado una formula simple pero precisa que está dando los resultados esperados. La formula es la siguiente:

Evaluación

= *Aplicación*

$$* \left(\frac{\text{Promedio}(\text{Importancia del objetivo} * \text{Contribución Práctica al Objetivo})}{16} \right) \\ * \left(\frac{6 - \text{Promedio Dificultades}}{4} \right) * \left(\frac{7.5 - \text{Esfuerzo Preparación}}{5} \right)$$

Las variables implicadas en la formula anterior se describen a continuación:

Aplicación: Esto es referente a las prácticas, cada práctica puede encontrarse a un nivel de aplicación diferente y este valor es importante tenerlo en cuenta al momento de la evaluación, es decir, saber si la práctica esta 100% aplicada o si descartan aplicarla. Este valor se captura en el paso 2 e internamente tiene valores de 1 a 4 siendo 1 que descartan aplicar dicha práctica y 4 que está completamente aplicada. Este factor ha sido tomado en cuenta debido a su importancia de cara a las prácticas que el equipo puede aplicar o ya está aplicando.

Importancia: Esto es referente a los objetivos, cada objetivo puede llegar a tener una importancia diferente para el equipo. Este valor se captura en el paso 1 de la aplicación e internamente obtiene valores que van de 0 a 4 siendo 0 que no le interesa ese objetivo para ese producto/servicio que se está evaluando. Este factor es considerado para determinar las prácticas más relevantes con respecto a que tan importante es dicho objetivo para el producto/servicio evaluado.

Contribución: Esto es referente a los objetivos y prácticas, cada práctica contribuye en porcentaje diferente a los objetivos, por lo cual la aplicación cuenta con una base de conocimiento predefinido que indica dicha contribución. Estos valores van de 1 a 5 siendo 1 Muy Baja contribución y 5 Muy Alta contribución. Este factor es utilizado internamente para especificar cuánto contribuye una práctica a un objetivo debido a que cada práctica puede contribuir más o menos que otras.

Promedio dificultades: Esto es referente a los desafíos de las prácticas, cada práctica tiene una cantidad variable de desafíos donde cada uno de esos desafíos puede conllevar una dificultad diferente. Se calcula el promedio de las dificultades de los desafíos de cada práctica, los cuales van de 0 a 4, significando 0 que no tiene ninguna dificultad (*no existe este desafío*) y 4 que sería imposible conseguir superar ese desafío. Este factor es considerado para poder calificar las prácticas, una práctica con un alto promedio de dificultad sería menos probable lograr implantarla.

Esfuerzo preparación: Esto es referente a las prácticas, cada práctica conlleva un esfuerzo de preparación, la aplicación consta también con una base de conocimiento con respecto al esfuerzo que podría conllevar esa práctica al momento de implantación. Estos valores se manejan internamente y van de 1 a 5 significando 1 que conlleva muy poco esfuerzo y 5 mucho esfuerzo. Este factor es considerado para evaluar el esfuerzo requerido por una práctica antes de la implantación.

El resultado de esta fórmula es el valor presentado en la columna evaluación en la etapa final de la aplicación y varía dependiendo los objetivos, estado de aplicación de la práctica y dificultades especificadas en los pasos correspondientes.

Capítulo 6. Estadísticas y explotación de datos.

La aplicación AGILE Roadmap fue puesta en marcha a principios del mes de abril del presente año 2013 con fines de captar la atención de equipos interesados en realizar un agile roadmap de manera automatizada. Hasta Mayo 2013 se han realizado alrededor de 182 evaluaciones a equipos o productos/servicios de diferentes países.

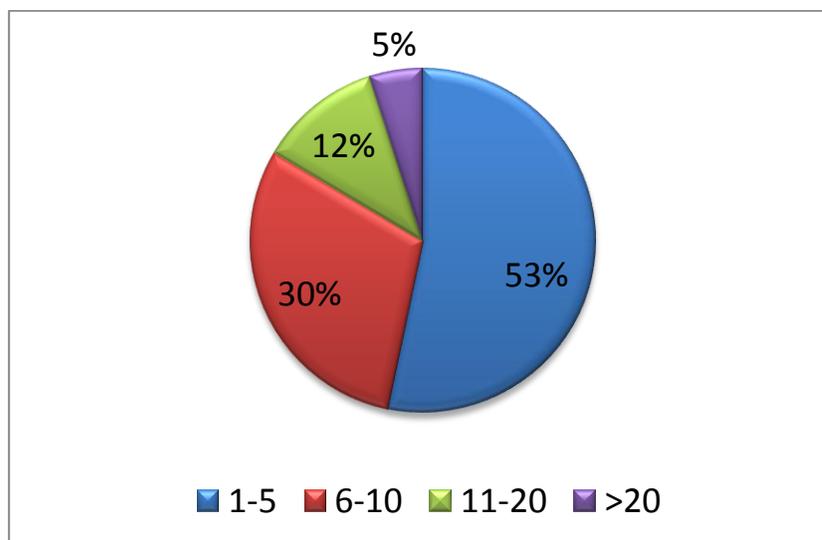
A continuación muestro un conjunto de graficas describiendo los datos evaluados en la aplicación AGILE Roadmap.

Tamaño de equipos evaluados:

En la *Tabla 1* y la *Grafica I*, se muestra el rango de miembros que componen los equipos que han realizado su agile roadmap usando la.

Rango Miembros	Evaluaciones
1-5	97
6-10	55
11-20	21
>20	9
Total	182

Tabla 1 - Tamaño de Equipos Evaluados.



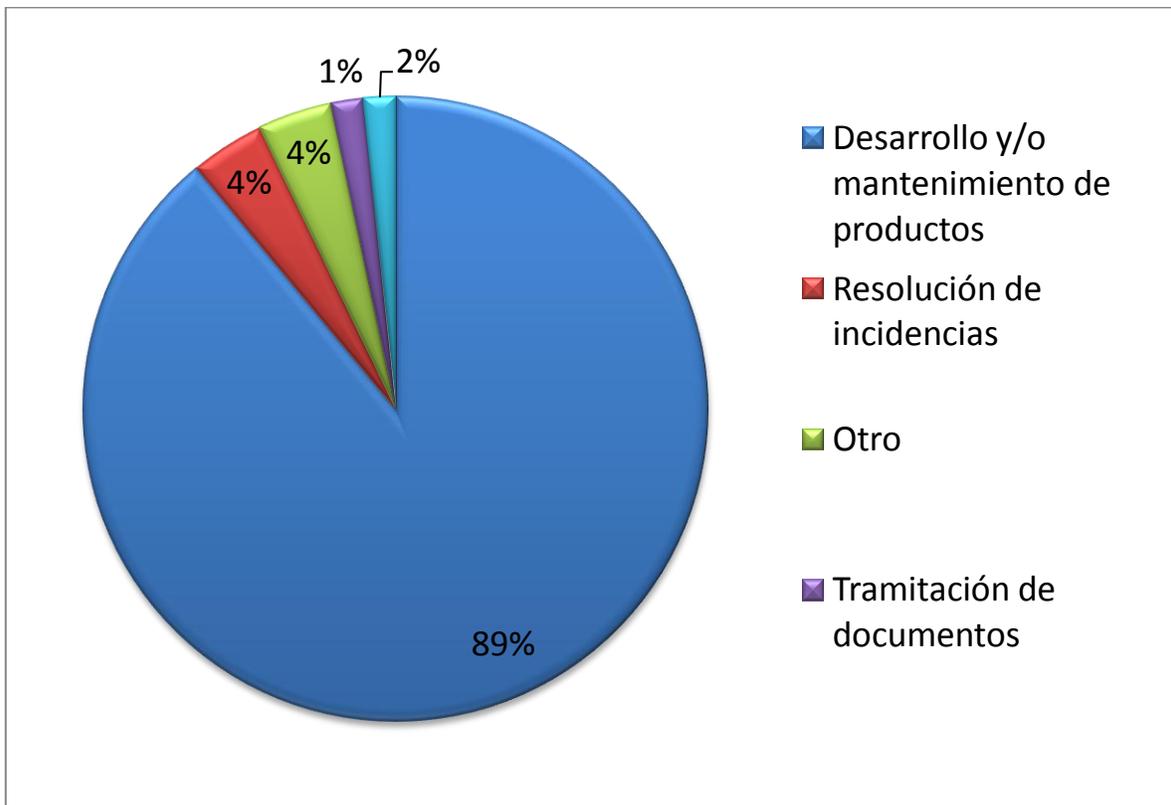
Grafica I - Tamaño de equipo mas frecuente.

Ámbito del equipo:

De los 182 equipos que realizaron sus evaluaciones en la aplicación, el 89% trabajan en Desarrollo y/o mantenimiento de productos según se puede observar en la *Grafica II*. En la *Tabla 2* se muestra la cantidad de evaluaciones que se realizaron en cada uno de los ámbitos.

Ámbito	Evaluaciones
Desarrollo y/o mantenimiento de productos	162
Resolución de incidencias	7
Otro	7
Tramitación de documentos	3
Atención y soporte al cliente	3
Total	182

Tabla 2 - Ambito de equipo y evaluaciones realizadas.



Grafica II - Porcentaje de evaluaciones por ambito.

Organización del trabajo.

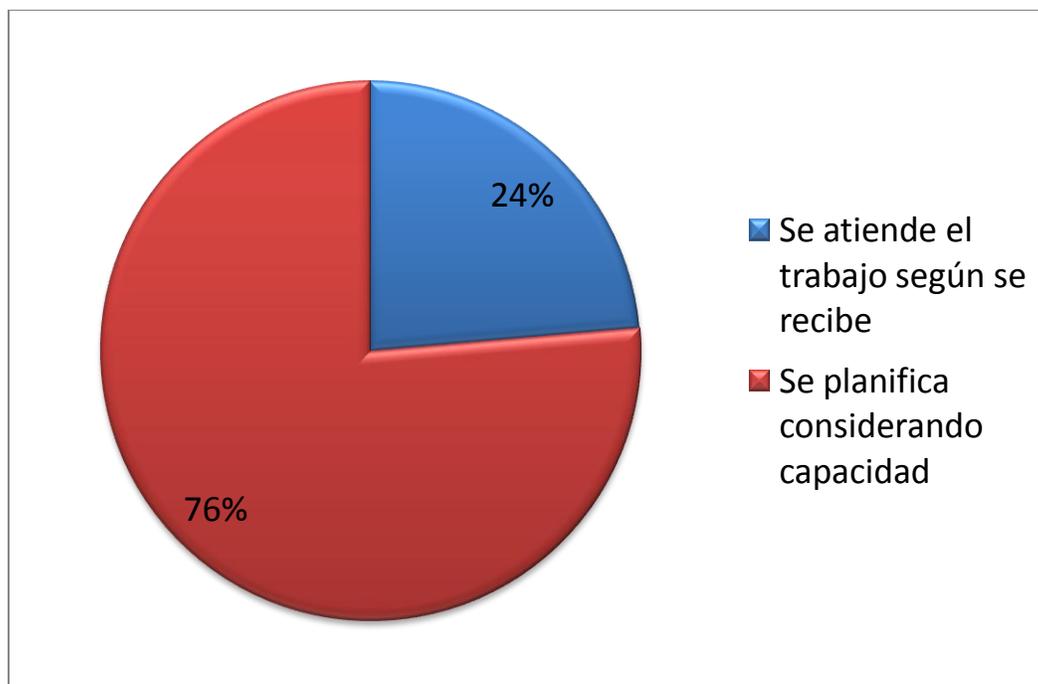
Existen dos formas comunes en las que los equipos llevan la organización del trabajo:

- ✓ *Atendiendo el trabajo según lo reciben*: Según se va asignando trabajo al equipo, este lo va completando, es decir, el trabajo se realiza bajo demanda.
- ✓ *Planificándolo considerando la capacidad*: El equipo organiza el trabajo teniendo en cuenta su capacidad, es decir, el trabajo pendiente se puede organizar por prioridad o algún otro criterio e ir acorde con la capacidad del equipo.

La *Tabla 3* muestra que la mayoría de los equipos que han realizado su agile roadmap en la aplicación. La *Grafica III* muestra el valor porcentual obtenido por cada forma de organización:

Organización del Trabajo	Evaluaciones
Se atiende el trabajo según se recibe	43
Se planifica considerando capacidad	139
Total	182

Tabla 3 - Categorías de Organización del trabajo.



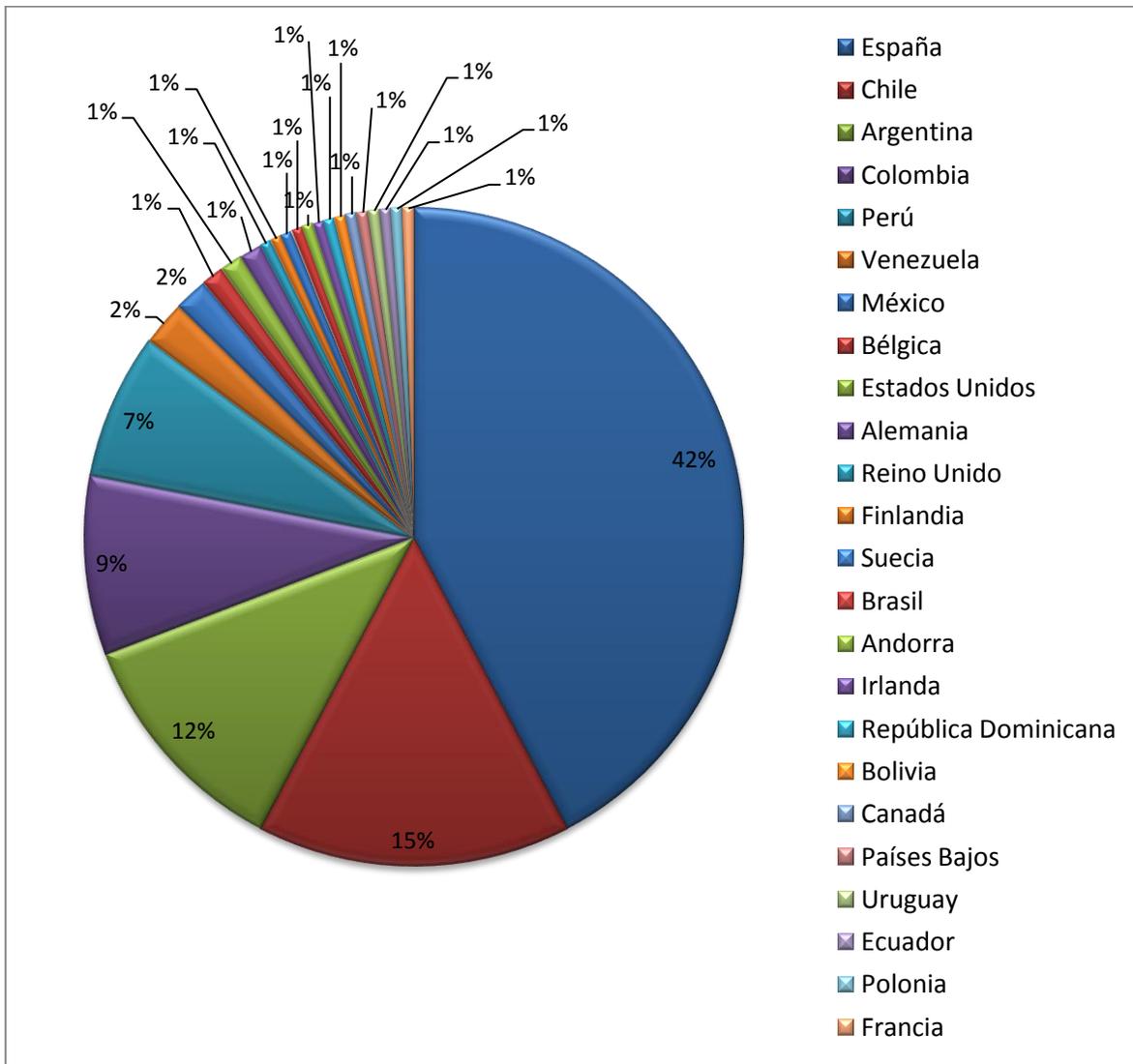
Grafica III - Forma de organización más común entre los equipos evaluados.

Evaluaciones realizadas por país.

En la *Tabla 4* se presenta el listado de países desde donde se han realizado las evaluaciones en la aplicación, en la *Grafica IV* se expresan los valores porcentuales de las evaluaciones realizadas.

<i>País</i>	<i>Evaluaciones</i>
España	77
Chile	28
Argentina	21
Colombia	16
Perú	13
Venezuela	4
México	3
Bélgica	2
Estados Unidos	2
Alemania	2
Reino Unido	1
Finlandia	1
Suecia	1
Brasil	1
Andorra	1
Irlanda	1
República Dominicana	1
Bolivia	1
Canadá	1
Países Bajos	1
Uruguay	1
Ecuador	1
Polonia	1
Francia	1
Total	182

Tabla 4 - Países desde donde se realizaron las evaluaciones.



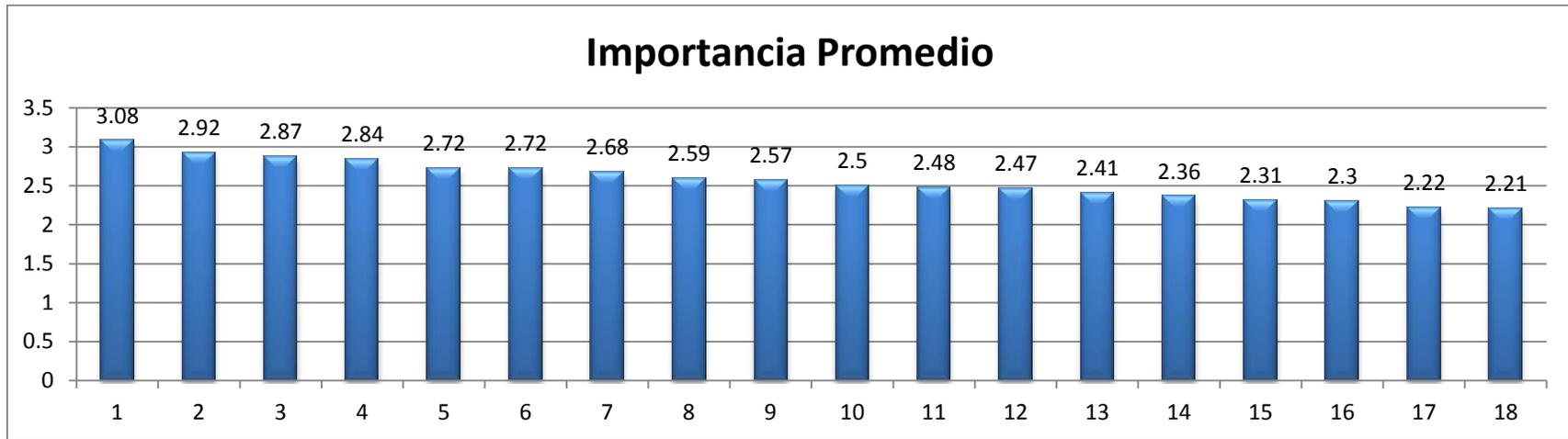
Grafica IV - Valor porcentual de evaluaciones por países.

Objetivos de mejora

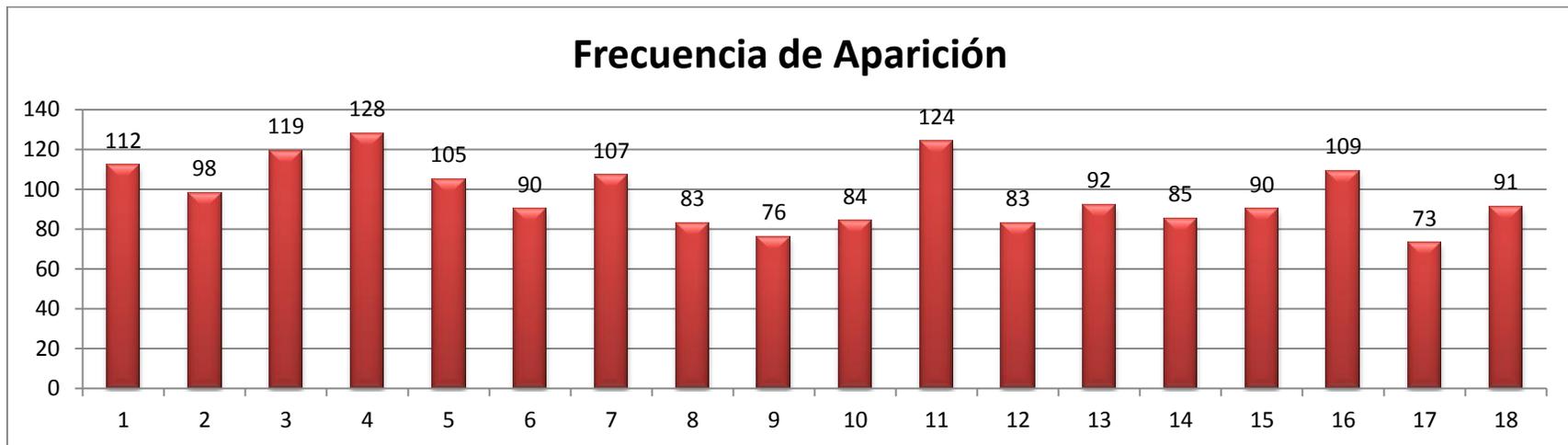
Aquí presentamos el listado de objetivos de mejora que esta incluidos en la aplicación. Cabe destacar que los objetivos que aparecen con mayor frecuencia no tienen por qué ser los más importantes como se puede apreciar en la *Tabla 5*. Las *Graficas V y VI* muestran los valores de importancia promedio y frecuencia de aparición respectivamente.

	Objetivo	Importancia Promedio	Frecuencia de Aparición
1	Reducir defectos en el trabajo entregado al cliente	3.08	112
2	Reducir el re-trabajo debido a trabajo defectuoso o incompleto detectado por el equipo o por el cliente	2.92	98
3	Evitar o reducir los retrasos en las entregas	2.87	119
4	Aumentar la productividad del equipo	2.84	128
5	Mejorar el grado de satisfacción del cliente con el producto o servicio entregado	2.72	105
6	Mejorar la comunicación dentro del equipo y con el cliente	2.72	90
7	Mejorar la motivación y compromiso del equipo	2.68	107
8	Mejorar la alineación del trabajo del equipo con los objetivos de la empresa	2.59	83
9	Reducir las horas extras o demanda no prevista de recursos humanos adicionales	2.57	76
10	Evitar costos asociados a la realización de tareas prescindibles o dudosamente rentables	2.50	84
11	Reducir el tiempo de entrega al cliente, acelerar el "time to market"	2.48	124
12	Involucrar en mayor medida al cliente en la planificación, definición y validación del trabajo	2.47	83
13	Promover la mejora continua del proceso empleado por el equipo	2.41	92
14	Mejorar la supervisión del trabajo del equipo	2.36	85
15	Hacer más visible el estado del trabajo del equipo	2.31	90
16	Gestionar eficazmente los cambios, tanto en los trabajos como en sus prioridades	2.30	109
17	Mejorar la gestión de recursos humanos en el equipo	2.22	73
18	Mejorar la sistematización del trabajo	2.21	91

Tabla 5- Importancia promedio y frecuencia de aparición de los objetivos.



Grafica V - Promedio de Importancia obtenido en las evaluaciones.



Grafica VI - Frecuencia de Aparición de los Objetivos (Cantidad de evaluaciones en las que aparece).

Nivel de aplicación de prácticas.

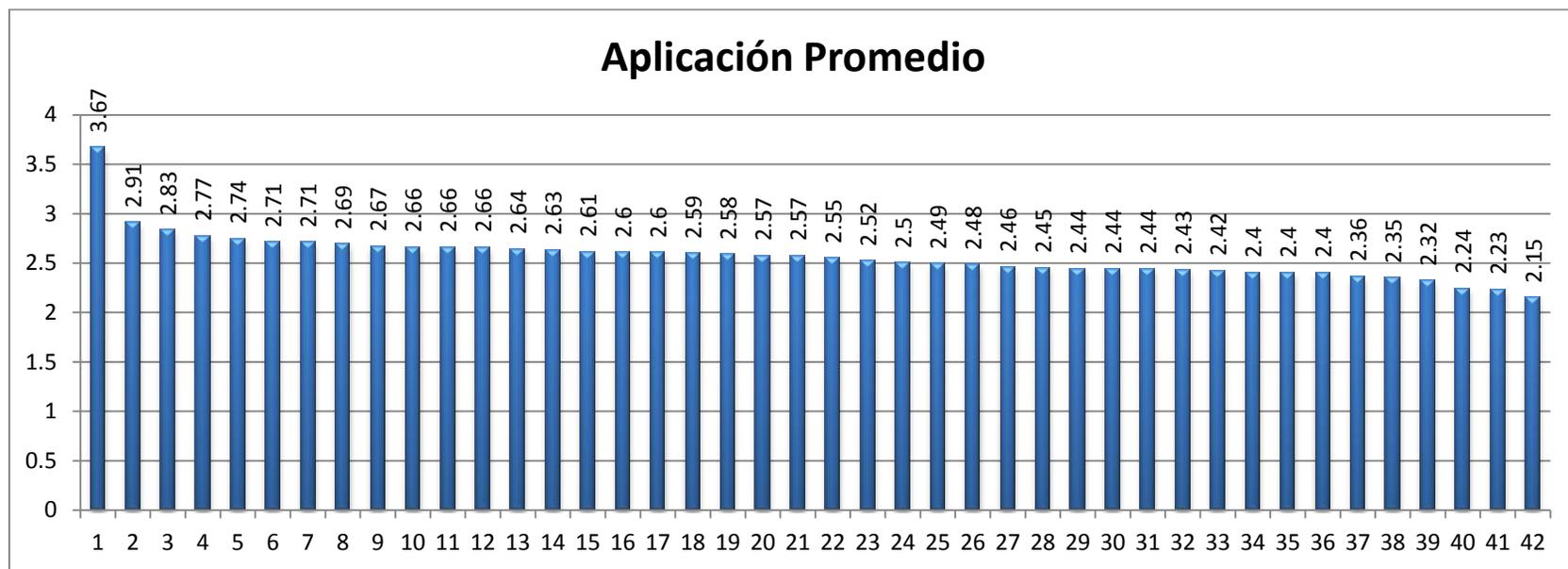
En la *Tabla 6* se muestra en promedio el nivel de aplicación de las prácticas evaluadas por los usuarios, sus respectivos agile roadmaps y su frecuencia de aparición. El nivel de aplicación de las prácticas es un poco indiferente de la frecuencia de aparición siendo que una práctica con una alta frecuencia de aparición puede no ser la que tenga el nivel de aplicación promedio más elevado.

	Práctica	Aplicación Promedio	Frecuencia de Aparición
1	Formar equipos pequeños y procurar que mantengan sus integrantes	3.67	146
2	Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista	2.91	78
3	Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses)	2.83	81
4	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico	2.77	111
5	Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo	2.74	111
6	El equipo se auto-organiza y toma las decisiones técnicas	2.71	93
7	Evitar invertir esfuerzo en adelantar trabajo que no esté comprometido y/o no esté cercano a su entrega	2.71	17
8	Seguimiento continuo (frecuencia de días, no semanas)	2.69	114
9	Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	2.67	101
10	Realizar entregas frecuentes de unidades de trabajo terminadas	2.66	107
11	Visualización de todo el trabajo pendiente encargado al equipo	2.66	108
12	Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	2.66	93
13	No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo	2.64	96
14	Abordar y entregar trabajo terminado de forma incremental	2.63	131
15	Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en	2.61	111

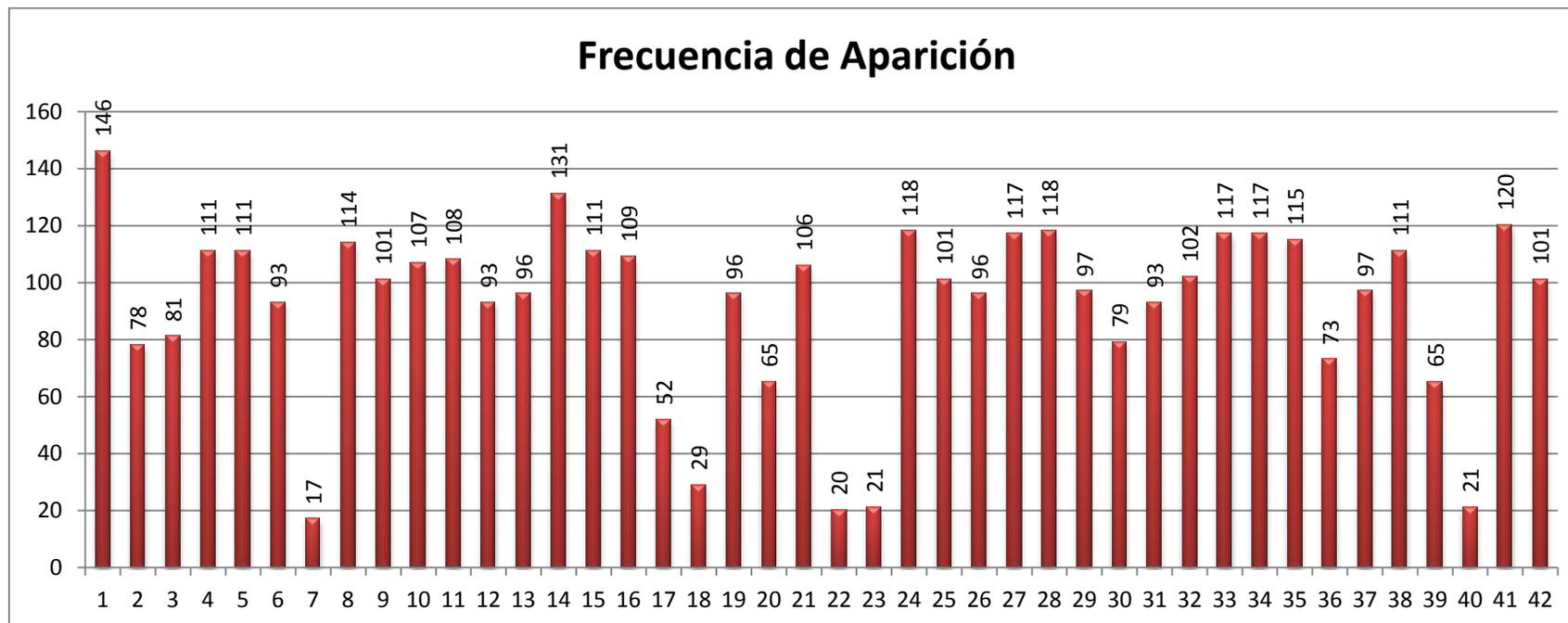
	alguna(s) de ellas		
16	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	2.60	109
17	Integrar de forma continua en el producto el trabajo terminado	2.60	52
18	Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado	2.59	29
19	Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo	2.58	96
20	Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo	2.57	65
21	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	2.57	106
22	Establecer y comunicar al equipo la visión del producto o servicio, y reforzarla regularmente	2.55	20
23	Establecer pautas para gestionar convenientemente el re-trabajo	2.52	21
24	Realizar una reunión diaria del equipo al completo, cara a cara y muy breve	2.50	118
25	Realizar reuniones de revisión del trabajo entregado	2.49	101
26	Acordar y definir qué se entiende por trabajo terminado, tanto para las actividades realizadas por el equipo como respecto de las entregas al cliente	2.48	96
27	Acotar el ámbito de trabajo del equipo	2.46	117
28	Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla	2.45	118
29	Que exista un líder de mejora de proceso disponible para el equipo	2.44	97
30	Establecimiento de estándares para el trabajo técnico del equipo	2.44	79
31	Trabajo o actividades realizadas en conjunto por dos o más integrantes	2.44	93
32	Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	2.43	102
33	Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad	2.42	117
34	Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	2.40	117

35	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente	2.40	115
36	Mejorar continuamente la organización interna del producto para facilitar su mantenimiento	2.40	73
37	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	2.36	97
38	Establecer una disciplina de aprovechamiento de las reuniones	2.35	111
39	Postergar hasta último momento la asignación del encargado de realizar una actividad	2.32	65
40	Contar con un espacio físico de trabajo que favorezca la interacción entre los miembros del equipo	2.24	21
41	Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	2.23	120
42	Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	2.15	101

Tabla 6 - Aplicacion promedio y frecuencia de aparicion de las prácticas.



Grafica VII - Aplicación promedio obtenida en las evaluaciones realizadas.



Grafica VIII - Frecuencia de Aparición de las prácticas (Cantidad de evaluaciones en las que aparece).

La aplicación promedio de las prácticas mostradas en la *Grafica VII*, fueron obtenidas de los datos provistos por los usuarios en la aplicación, específicamente en el paso 2, ahí se pueden apreciar las prácticas que tienen mayor nivel de aplicación. Cabe destacar que el nivel de aplicación de las prácticas es en cierto sentido independiente de la frecuencia de aparición o cantidad de evaluaciones, en la *Grafica VIII* se puede ver que la práctica 7 es la que aparece en menos evaluaciones siendo todo lo contrario a su nivel de aplicación que es de los más altos.

Dificultad de los desafíos.

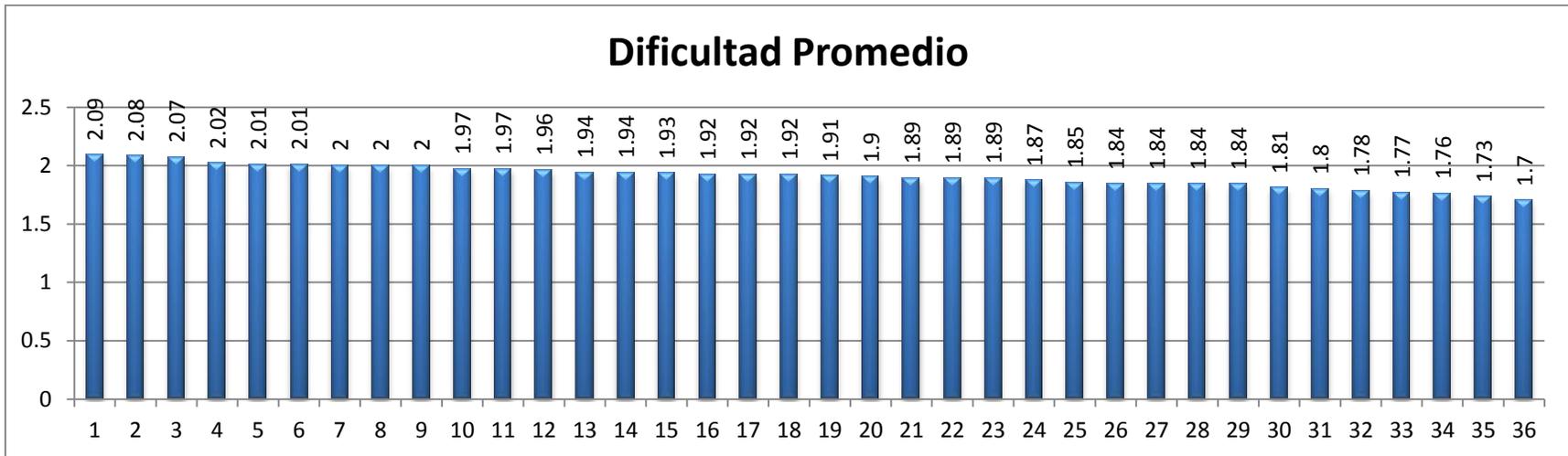
En la *Tabla 7* se muestra el listado de desafíos, su dificultad promedio y su frecuencia de aparición. Cabe destacar que la frecuencia de aparición de un desafío está limitada a la aparición de la práctica a la que está asociada.

	Desafío	Dificultad Promedio	Frecuencia de Aparición
1	Experiencia en automatización de pruebas	2.09	76
2	Contar con un mecanismo de incentivos que valore el desempeño del equipo, no solo el desempeño de cada integrante	2.08	65
3	Contar con una de definición de puestos de trabajo y remuneraciones compatible con la diversidad de actividades que se pretende que puedan llegar a realizar los integrantes del equipo	2.07	81
4	Si existe un contrato con la parte cliente, que sea flexible en cuando a contenido. Que se puedan añadir, quitar o modificar parte del trabajo acordado, pero manteniendo la consistencia entre el esfuerzo previsto y la capacidad del equipo.	2.02	44
5	Evitar las interrupciones a miembros del equipo, en su lugar promover la realización de reuniones programadas, especialmente cuando la interrupción pueda ser de más de 10 minutos (orientativo)	2.01	84
6	Conseguir que no se añada trabajo adicional al acordado para un período planificado, excepto por urgencias y/o cambios de prioridades	2.01	92
7	Que el cliente esté de acuerdo en recibir como entrega lo mínimo que pueda serle útil en un momento determinado	2.00	82
8	Infraestructura para la ejecución de pruebas automatizadas	2.00	76
9	Experiencia en definición y aplicación de pruebas de aceptación	2.00	72
10	Conseguir que representantes de la parte cliente participen en las revisiones de cada entrega	1.97	69
11	Experiencia y capacidad de los miembros del equipo para tomar decisiones técnicas acertadas	1.97	65
12	Disponer de un espacio acondicionado para el trabajo en equipo	1.96	77
13	Conseguir tener en el equipo integrantes que tengan habilidades para abordar todas las actividades necesarias para terminar un trabajo	1.94	72

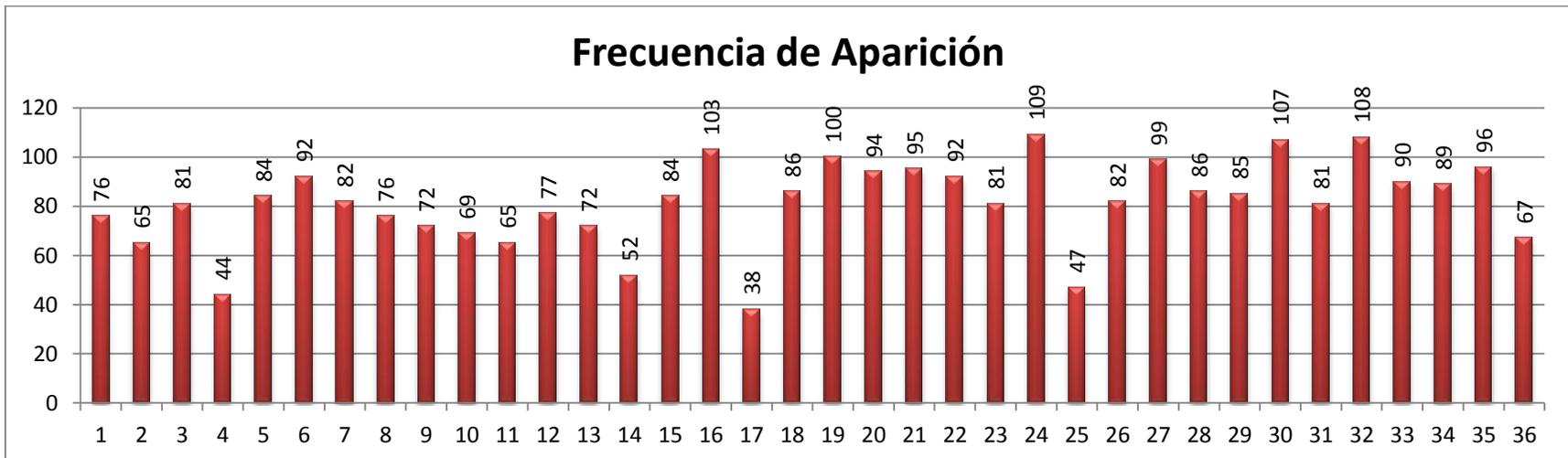
14	Experiencia en refactorización	1.94	52
15	Posibilidad de añadir, modificar o incluso eliminar total o parcialmente la documentación utilizada en el proceso	1.93	84
16	Tener un representante de la parte cliente que ofrezca alta disponibilidad para que el equipo interactúe con él	1.92	103
17	Experiencia, infraestructura y disciplina para integración continua	1.92	38
18	Conseguir que los integrantes del equipo estén dispuestos a realizar actividades que no sean de su especialidad o preferencia	1.92	86
19	Disciplina de registro del progreso del trabajo (si el producto o servicio lo requiere)	1.91	100
20	Experiencia y hábito de realización de estimaciones (si el producto o servicio lo requiere)	1.90	94
21	Hábito de elaboración y aprovechamiento de la documentación durante el proceso, no solo para acompañar a la entrega	1.89	95
22	Pro actividad y habilidad de los miembros del equipo para auto-gestionarse individualmente y/o en equipo	1.89	92
23	Evitar que el equipo esté simultáneamente trabajando en demasiados productos, servicios o proyectos. Priorizar el trabajo de manera que un equipo no se vea obligado a distribuir su capacidad entre contextos de trabajo diferentes	1.89	81
24	Conseguir el apoyo de la dirección para implantar esta práctica	1.87	109
25	Desistir de realizar un balanceo de carga de trabajo de los miembros del equipo pues sólo tendrían asignado el trabajo en el cual están trabajando, no se realizarían asignaciones a futuro, salvo en casos excepcionales.	1.85	47
26	Que el equipo tenga un líder-facilitador en lugar de un jefe autoritario	1.84	82
27	Conseguir un único y buen representante de la parte cliente	1.84	99
28	Tener la posibilidad de racionalizar la documentación	1.84	86
29	Evitar que los miembros del equipo tengan demasiadas unidades de trabajo asignadas. Privilegiar el terminar unidades de trabajo asignadas en lugar de asignar(se) nuevas	1.84	85
30	Disciplina de reuniones efectivas, que incluya: tener un moderador, anticipar el propósito e información relevante, etc.	1.81	107

31	Alinear la planificación con un proceso incremental, en el cual se acuerdan las unidades de trabajo que se entregan en cierto plazo, sin detallar cómo se organiza el equipo para cumplir dicho plazo	1.80	81
32	Medir el progreso del trabajo respecto del grado de avance de las funcionalidades, características o servicios que aprovechará el cliente, no por volumen o avance de documentación u otros artefactos que acompañan al producto o servicio	1.78	108
33	Conseguir reunir en el mismo espacio físico a todos los que intervienen en las actividades necesarias para realizar el trabajo	1.77	90
34	Definir el trabajo en términos de incrementos en las características que ofrece el producto o servicio que se le ofrece al cliente, no en términos de actividades o artefactos necesarios desde la perspectiva técnica	1.76	89
35	Transparencia en cuanto a la información asociada al trabajo y a los encargados de realizarlo	1.73	96
36	Buena actitud de los miembros del equipo para trabajar en conjunto en una misma actividad para determinadas unidades de trabajo	1.70	67

Tabla 7 - Dificultad promedio y frecuencia de aparición de los desafíos.



Grafica IX - Promedio dificultad de los desafíos.



Grafica X - Frecuencia de aparición de las dificultades.

Con los desafíos se da un caso parecido al de las prácticas respecto a la independencia de su dificultad promedio frente a la frecuencia de aparición, es decir que el desafío que presenta mayor promedio de dificultad no tiene que ser obligatoriamente el que es más frecuente como se puede comprobar en las *Graficas IX y X*.

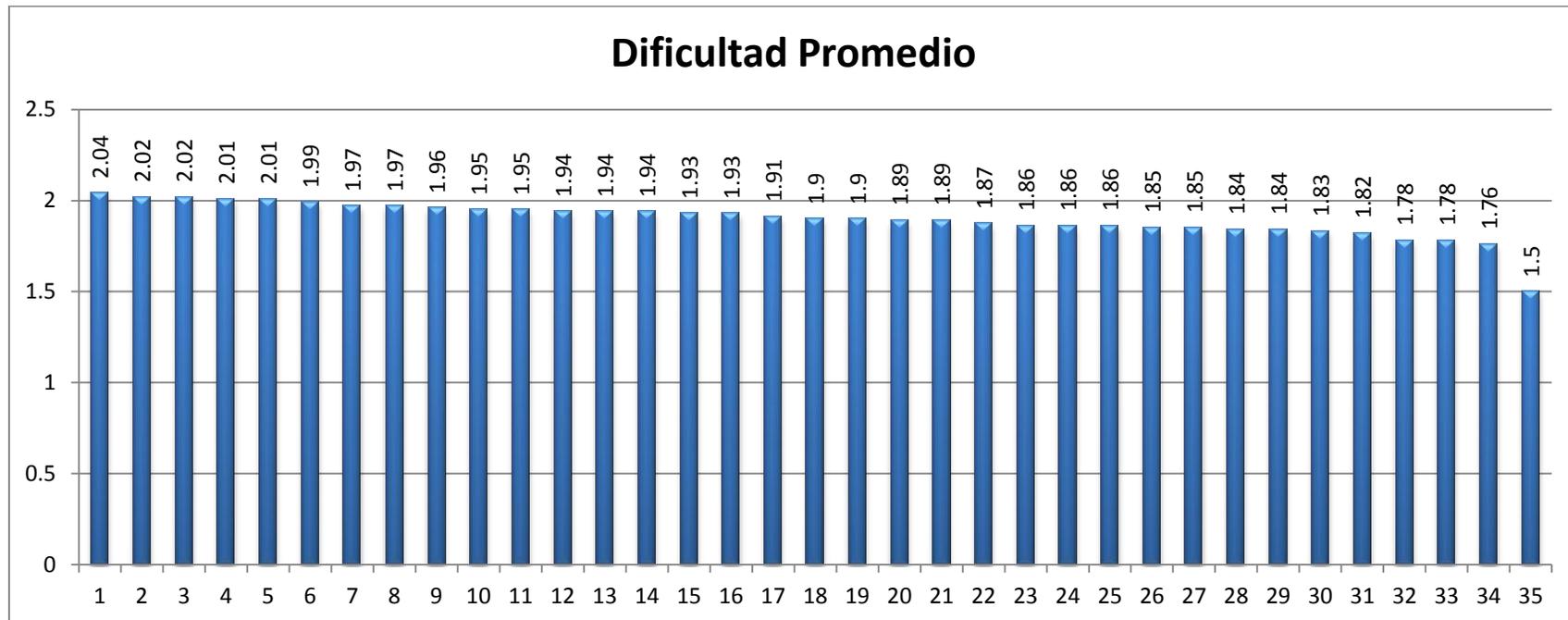
Dificultad de las prácticas con respecto a los desafíos.

Cada práctica con respecto a sus desafíos asociados, tiene un nivel de dificultad específico. En la *Tabla 8* se muestra el listado de prácticas ordenadas por su dificultad promedio con respecto de sus desafíos asociados. Cabe destacar que actualmente existen en nuestra base de conocimientos 7 prácticas las cuales no tienen desafíos asociados por lo que no se evalúan en los resultados siguientes.

	Práctica	Dificultad Promedio
1	Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	2.04
2	Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista	2.02
3	No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo	2.02
4	Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas	2.01
5	Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	2.01
6	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente	1.99
7	Que exista un líder de mejora de proceso disponible para el equipo	1.97
8	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico	1.97
9	Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo	1.96
10	Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo	1.95
11	Mejorar continuamente la organización interna del producto para facilitar su mantenimiento	1.95

12	Seguimiento continuo (frecuencia de días, no semanas)	1.94
13	Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	1.94
14	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	1.94
15	Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	1.93
16	Integrar de forma continua en el producto el trabajo terminado	1.93
17	Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	1.91
18	Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses)	1.90
19	Acotar el ámbito de trabajo del equipo	1.90
20	Trabajo o actividades realizadas en conjunto por dos o más integrantes	1.89
21	Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla	1.89
22	Realizar entregas frecuentes de unidades de trabajo terminadas	1.87
23	Establecer una disciplina de aprovechamiento de las reuniones	1.86
24	Postergar hasta último momento la asignación del encargado de realizar una actividad	1.86
25	El equipo se auto-organiza y toma las decisiones técnicas	1.86
26	Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad	1.85
27	Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo	1.85
28	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	1.84
29	Realizar reuniones de revisión del trabajo entregado	1.84
30	Abordar y entregar trabajo terminado de forma incremental	1.83
31	Realizar una reunión diaria del equipo al completo, cara a cara y muy breve	1.82
32	Visualización de todo el trabajo pendiente encargado al equipo	1.78
33	Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	1.78
34	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	1.76

Tabla 8 - Dificultad promedio de las prácticas



Grafica XI - Promedio de dificultad presentado por las prácticas evaluadas.

La *Grafica XI* muestra la dificultad promedio de cada práctica, la cual esta asociada directamente a los desafíos de las prácticas.

Prácticas recomendadas.

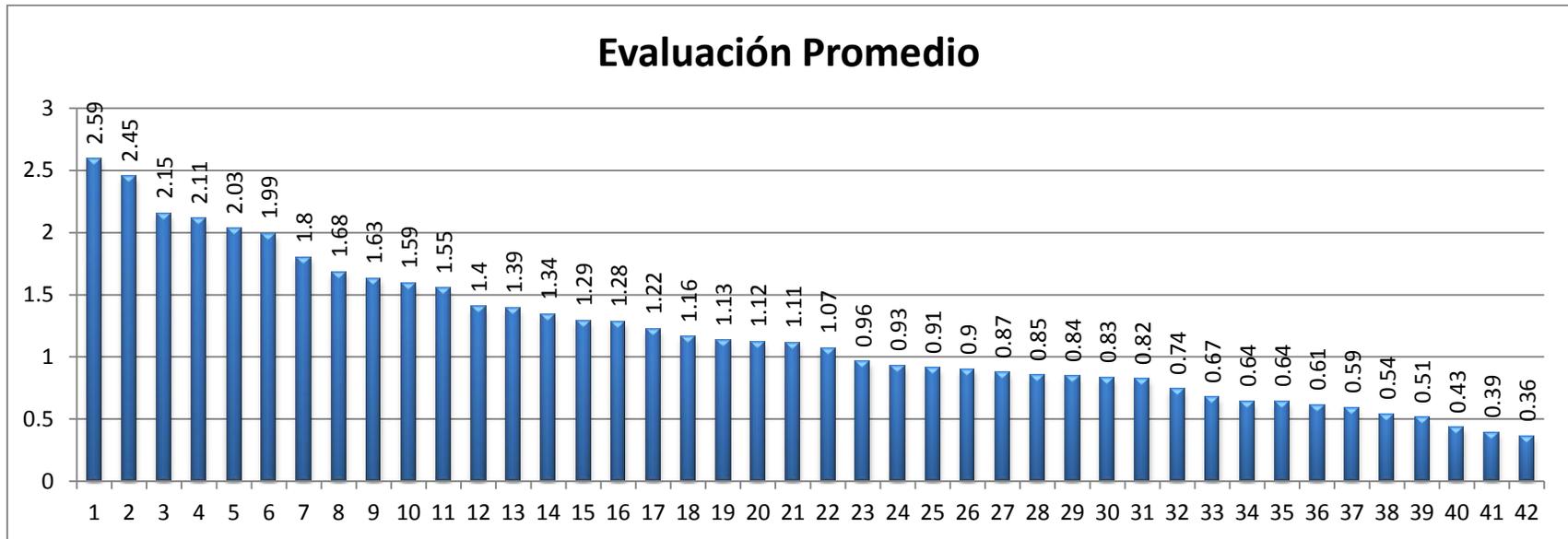
La *Tabla 9* vuelve a mostrar el listado de prácticas, pero esta vez mostrando el promedio de valoración obtenido según los datos proporcionados por los equipos que han realizado su Roadmap en la aplicación y la frecuencia de recomendación.

	Práctica	Evaluación Promedio	Frecuencia de Recomendación
1	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	2.59	129
2	Formar equipos pequeños y procurar que mantengan sus integrantes	2.45	174
3	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	2.15	138
4	Realizar reuniones de revisión del trabajo entregado	2.11	124
5	Realizar una reunión diaria del equipo al completo, cara a cara y muy breve	2.03	146
6	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico	1.99	140
7	Visualización de todo el trabajo pendiente encargado al equipo	1.80	135
8	Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	1.68	126
9	Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses)	1.63	139
10	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	1.59	140
11	Que exista un líder de mejora de proceso disponible para el equipo	1.55	117
12	Seguimiento continuo (frecuencia de días, no semanas)	1.40	143
13	Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad	1.39	149
14	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente	1.34	162
15	Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	1.29	128

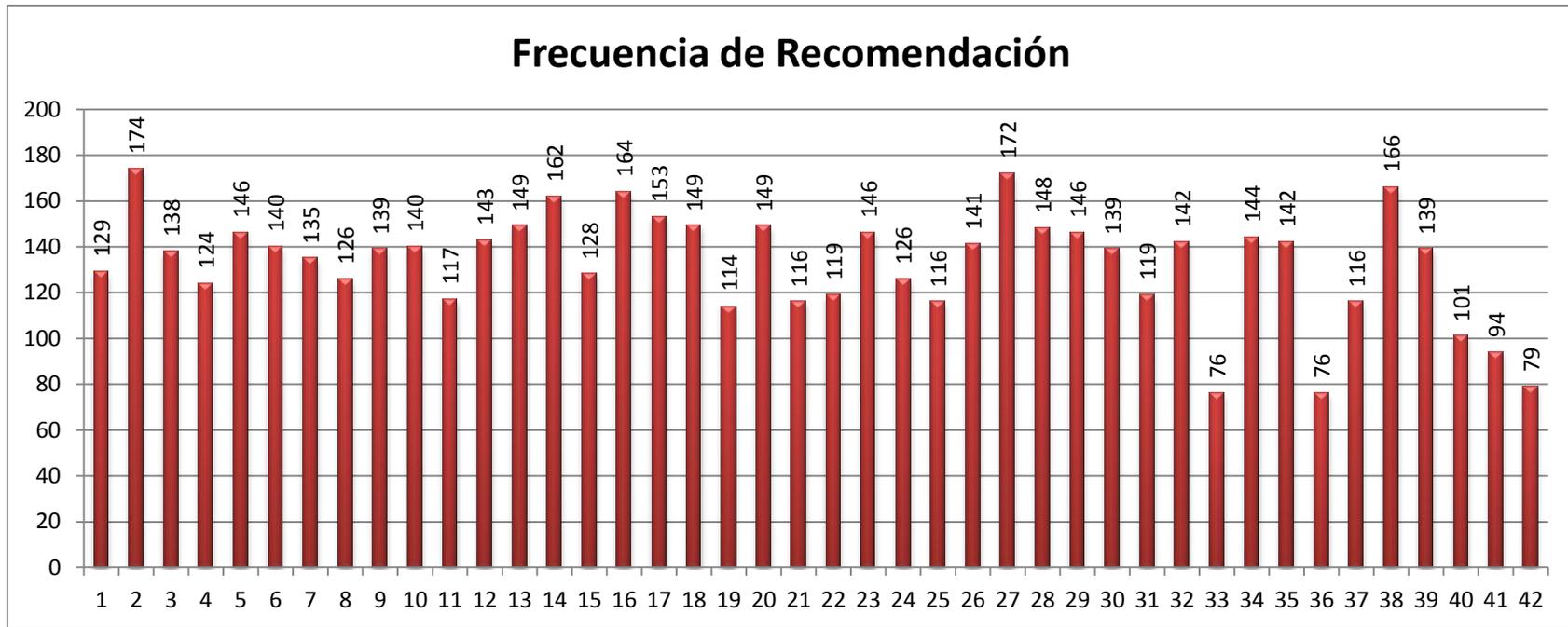
16	Abordar y entregar trabajo terminado de forma incremental.	1.28	164
17	Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	1.22	153
18	Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla	1.16	149
19	Acordar y definir qué se entiende por trabajo terminado, tanto para las actividades realizadas por el equipo como respecto de las entregas al cliente	1.13	114
20	Acotar el ámbito de trabajo del equipo	1.12	149
21	Trabajo o actividades realizadas en conjunto por dos o más integrantes	1.11	116
22	No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo	1.07	119
23	Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado	0.96	146
24	Establecer y comunicar al equipo la visión del producto o servicio, y reforzarla regularmente	0.93	126
25	El equipo se auto-organiza y toma las decisiones técnicas	0.91	116
26	Establecer una disciplina de aprovechamiento de las reuniones	0.90	141
27	Realizar entregas frecuentes de unidades de trabajo terminadas	0.87	172
28	Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	0.85	148
29	Evitar invertir esfuerzo en adelantar trabajo que no esté comprometido y/o no esté cercano a su entrega	0.84	146
30	Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista	0.83	139
31	Contar con un espacio físico de trabajo que favorezca la interacción entre los miembros del equipo	0.82	119
32	Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas	0.74	142
33	Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo	0.67	76

34	Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	0.64	144
35	Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo	0.64	142
36	Postergar hasta último momento la asignación del encargado de realizar una actividad	0.61	76
37	Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	0.59	116
38	Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo	0.54	166
39	Establecer pautas para gestionar convenientemente el re-trabajo	0.51	139
40	Mejorar continuamente la organización interna del producto para facilitar su mantenimiento	0.43	101
41	Establecimiento de estándares para el trabajo técnico del equipo	0.39	94
42	Integrar de forma continua en el producto el trabajo terminado	0.36	79

Tabla 9 - Evaluación promedio y frecuencia de recomendación de las prácticas.



Grafica XII - Promedio de evaluación de las prácticas.



Grafica XIII - Frecuencia con la que las prácticas son recomendadas (Cantidad de Agile Roadmap en la práctica ha sido recomendada).

La *Grafica XII* y *XIII*, muestran el promedio de evaluación y la frecuencia de recomendación que ha recibido cada una de las prácticas, al igual que el nivel de aplicación y la frecuencia de aparición, estos valores son independientes uno de otros. La evaluación promedio de las prácticas es el resultado obtenido luego de aplicada la fórmula de AGILE Roadmap sobre los datos proporcionados por el equipo y la frecuencia de recomendación muestra la cantidad de Roadmaps en los que ha sido recomendada cada práctica.

Capítulo 7. Conclusiones

La realización de este trabajo ha sido motivada por las dificultades que tiene un equipo de trabajo para comenzar a implantar agilísimo debido a los numerosos métodos, sus prácticas se solapan, además de las dificultades e impedimentos de cada contexto. Nuestra contribución ha sido unificar las prácticas y proporcionar un apoyo para la elaboración de un Roadmap preliminar.

Las dificultades existentes en el proceso de selección de prácticas ágiles son un punto clave por el que muchos equipos se quedan en la etapa inicial de su transformación ágil.

Debido a la gran cantidad de prácticas existentes que provienen de diversos métodos, a los equipos de trabajo se les hace difícil seleccionar las prácticas más adecuadas para ser implantadas.

Un Agile Roadmap elaborado correctamente proporciona al equipo una guía que facilita la implantación de prácticas ágiles de una manera continua e incremental.

El objetivo de desarrollar un sitio/aplicación web para la implantación de prácticas ágiles propuesto en este trabajo, se ve alcanzado con la puesta en marcha de AGILE Roadmap y con las evaluaciones realizadas por diversos equipos de trabajos para sus productos o servicios ya almacenadas en su base de datos.

El desarrollo de este trabajo de fin de máster me ha aportado conocimientos muy importantes en lo referente al desarrollo ágil. En lo práctico, ha servido para impulsarme a trabajar a mayores niveles en ambientes web al tener que desarrollar un sitio/aplicación.

Referencias

- [1]. Michael J. O'Donnell and Ita Richardson "Problems Encountered When Implementing Agile Methods in a Very Small Company", Communications in Computer and Information Science Volume 16, 2008, pp 13-24 Limerick, Ireland.
- [2]. Martin, J., Rapid Application Development, Macmillan Inc., New York, 1991.
- [3]. Amaro Calderón, Sarah Dámaris Valverde Rebaza, Jorge Carlos "Metodologías Ágiles", Universidad Nacional de Trujillo, Facultad de Ciencias Físicas y Matemáticas, Escuela de Informática, Trujillo, Perú 2007.
- [4]. José H. Canós, Patricio Letelier yM^a Carmen Penadés "Metodologías Ágiles en el Desarrollo de Software", DSIC -Universidad Politécnica de Valencia, España.
- [5]. David J. Anderson, Kanban, "Successful Evolutionary Change for Your Technology Business".
- [6]. Mary y Tom Poppendieck, "Lean Software Development - An Agile Toolkit", May 2003.
- [7]. Sutherland, Jeffrey Victor; Schwaber, Ken "The Scrum methodology" (1995). Business object design and implementation: OOPSLA '95 workshop proceedings. The University of Michigan. p. 118. ISBN 3-540-76096-2.
- [8]. Kent Beck, "Extreme Programming Explained", ISBN: 0201616416; September 1999.
- [9]. Kent Beck and Cynthia Andres, "Extreme Programming Explained: Embrace Change (2nd Edition)", ISBN: 0321278658; Addison-Wesley Professional ©2004.

- [10]. Michael Sahota, “An Agile Adoption And Transformation Survival Guide”, ISBN: 1105735729 9781105735721, Lulu.com ©2012.
- [11]. Eliyahu M. Goldratt, “Theory of Constraints”, ISBN-10: 0884271668, North River Pr; 1 edition (December 1999).
- [12]. Patricio Letelier, “Tres claves para comenzar a implantar prácticas ágiles con éxito”, agilismoatwork.blogspot.com.es, Marzo 2013.
- [13]. Varios Autores, “Manifiesto por el Desarrollo Ágil de Software”, agilemanifesto.org © 2001.
- [14]. Ken Schwaber y Jeff Sutherland, “La Guía Definitiva de Scrum: Las reglas del juego”, Scrum.org, Octubre 2011.
- [15]. Patricio Letelier, “Tres claves para comenzar a implantar prácticas ágiles con éxito”, agilismoatwork.blogspot.com.es, Marzo 2013.
- [16]. Patricio Letelier, “Carta de Prácticas Ágiles: Arma tu propio menú ágil”, agilismoatwork.blogspot.com.es, Febrero 2013.
- [17]. Patricio Letelier, “Agile Roadmap: el primer paso para la implantación de prácticas ágiles”, agilismoatwork.blogspot.com.es, Abril 2013.
- [18]. Extraído de “Curso Gratis Scrum Día a Día”, disponible en santimacnet.wordpress.com, Noviembre, 2010.

Anexo 1 – Ejemplo del Informe generado por AGILE Roadmap

AGILE Roadmap <http://agileroadmap.tuneupprocess.com/>

Una iniciativa de Tune-up Process www.tuneupprocess.com

Datos del perfil Agile Roadmap	
Thursday, April 11, 2013 - 1:25 PM	
Producto o Servicio	Ejemplo
Número de Miembros	15
Ámbito	Desarrollo y/o mantenimiento de productos
Situación de Planificación	Se atiende el trabajo según se recibe
Sector de Empresa	Tecnología y Software

PASO 1

Objetivo	Importancia
OBJ21 - Reducir el tiempo de entrega al cliente, acelerar el "time to market"	Es vital conseguirlo
OBJ02 - Aumentar la productividad del equipo	Debemos conseguirlo
OBJ01 - Evitar o reducir los retrasos en las entregas	No es ahora un objetivo
OBJ05 - Reducir defectos en el trabajo entregado al cliente	No es ahora un objetivo
OBJ03 - Gestionar eficazmente los cambios, tanto en los trabajos como en sus prioridades	No es ahora un objetivo
OBJ06 - Mejorar el grado de satisfacción del cliente con el producto o servicio entregado	Podría interesarnos
OBJ11 - Mejorar la alineación del trabajo del equipo con los objetivos de la empresa	Queremos conseguirlo
OBJ14 - Evitar costos asociados a la realización de tareas prescindibles o dudosamente rentables	Queremos conseguirlo
OBJ13 - Mejorar la motivación y compromiso del equipo	Queremos conseguirlo
OBJ08 - Mejorar la supervisión del trabajo del equipo	No es ahora un objetivo
OBJ10 - Mejorar la comunicación dentro del equipo y con el cliente	Debemos conseguirlo
OBJ12 - Involucrar en mayor medida al cliente en la planificación, definición y validación del trabajo	Queremos conseguirlo
OBJ07 - Hacer más visible el estado del trabajo del equipo	No es ahora un objetivo
OBJ19 - Promover la mejora continua del proceso empleado por el equipo	No es ahora un objetivo
OBJ09 - Mejorar la gestión de recursos humanos en el equipo	Debemos conseguirlo
OBJ15 - Mejorar la sistematización del trabajo	No es ahora un objetivo
OBJ04 - Reducir las horas extras o demanda no prevista de recursos humanos adicionales	No es ahora un objetivo
OBJ20 - Reducir el re-trabajo debido a trabajo defectuoso o incompleto detectado por el equipo o por el cliente	No es ahora un objetivo

PASO 2

Práctica	Nivel de Aplicación
PRA01 - Abordar y entregar trabajo terminado de forma incremental	Completamente aplicada
PRA44 - Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado	No se está aplicando
PRA45 - Formar equipos pequeños y estables	No se está aplicando
PRA31 - Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad	No se está aplicando
PRA43 - Acotar el ámbito de trabajo del equipo	No se está aplicando
PRA32 - No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo	Descartamos aplicarla
PRA24 - Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	Se aplica parcialmente
PRA28 - Realizar una reunión diaria del equipo al completo, cara a cara y muy breve	Descartamos aplicarla
PRA26 - Realizar reuniones de revisión del trabajo entregado	Descartamos aplicarla
PRA08 - Visualización de todo el trabajo pendiente encargado al equipo	Completamente aplicada
PRA30 - Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	Completamente aplicada
PRA09 - Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	No se está aplicando
PRA18 - Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	Completamente aplicada
PRA12 - Trabajo o actividades realizadas en conjunto por dos o más integrantes	Se aplica parcialmente
PRA13 - El equipo se auto-organiza y toma las decisiones técnicas	Completamente aplicada
PRA37 - Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	Se aplica parcialmente
PRA05 - Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico	Se aplica parcialmente
PRA06 - Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo	No se está aplicando
PRA42 - Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas	No se está aplicando
PRA02 - Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	Completamente aplicada
PRA04 - Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla	Completamente aplicada
PRA14 - Que exista un líder de mejora de proceso disponible para el equipo	Completamente aplicada
PRA27 - Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	No se está aplicando
PRA35 - Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	Completamente aplicada
PRA07 - Establecer una disciplina de aprovechamiento de las reuniones	Completamente aplicada
PRA10 - Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	Se aplica parcialmente
PRA39 - Postergar hasta último momento la asignación del encargado de realizar una actividad. En la medida que los miembros del equipo van terminando trabajo, van cogiendo nuevo trabajo desde el que está pendiente y priorizado	No se está aplicando
PRA15 - Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo	No se está aplicando
PRA40 - Promover la sencillez en todos los aspectos. Centrarse en el mínimo conjunto de características que pueden constituir un producto útil y satisfactorio para el cliente	No se está aplicando

PASO 3

Práctica	Desafío	Dificultad
PRA05 - Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico		
	CHA16 - Conseguir reunir en el mismo espacio físico a todos los que intervienen en las actividades necesarias para realizar el trabajo	Tendríamos algunos impedimentos
	CHA22 - Disponer de un espacio acondicionado para el trabajo en equipo	Tendríamos algunos impedimentos
	CHA50 - Conseguir el apoyo de la dirección para implantar esta práctica	Ninguna, no existe este desafío
PRA06 - Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo		
	CHA23 - Conseguir que los integrantes del equipo estén dispuestos a realizar actividades que no sean de su especialidad o preferencia	Sería difícil lograrlo
	CHA24 - Conseguir tener en el equipo integrantes que tengan habilidades para abordar todas las actividades necesarias para terminar un trabajo	Tendríamos algunos impedimentos
PRA09 - Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ		
	CHA30 - Conseguir tener a un representante de la parte cliente ofreciendo alta disponibilidad para que el equipo interactúe con él	Fácilmente superable
	CHA41 - Conseguir un único y buen representante de la parte cliente	Tendríamos algunos impedimentos
PRA10 - Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios		
	CHA31 - Experiencia en automatización de pruebas	Tendríamos algunos impedimentos
	CHA32 - Infraestructura para la ejecución de pruebas automatizadas	Tendríamos algunos impedimentos
PRA12 - Trabajo o actividades realizadas en conjunto por dos o más integrantes		
	CHA34 - Buena actitud de los miembros del equipo para trabajar en conjunto en una misma actividad para determinadas unidades de trabajo	Tendríamos algunos impedimentos
	CHA54 - Contar con un mecanismo de incentivos que valore el desempeño del equipo, no solo el desempeño de cada integrante	Tendríamos algunos impedimentos
PRA15 - Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo		
	CHA23 - Conseguir que los integrantes del equipo estén dispuestos a realizar actividades que no sean de su especialidad o preferencia	Sería difícil lograrlo
	CHA34 - Buena actitud de los miembros del equipo para trabajar en conjunto en una misma actividad para determinadas unidades de trabajo	Tendríamos algunos impedimentos
PRA24 - Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado		
	CHA30 - Conseguir tener a un representante de la parte cliente ofreciendo alta disponibilidad para que el equipo interactúe con él	Fácilmente superable
	CHA41 - Conseguir un único y buen representante de la parte cliente	Tendríamos algunos impedimentos
PRA27 - Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.		
	CHA25 - Disciplina de reuniones efectivas. Tener siempre un buen moderador, anticipar el propósito e información relevante, registrar acuerdos, etc.	Tendríamos algunos impedimentos
	CHA28 - Transparencia en cuanto a la información asociada al trabajo y a los encargados de realizarlo	Tendríamos algunos impedimentos

	CHA29 - Disciplina de registro del progreso del trabajo	Tendríamos algunos impedimentos
	CHA36 - Proactividad y habilidad de los miembros del equipo para auto-gestionarse individualmente y/o en equipo	Tendríamos algunos impedimentos
PRA31 - Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad		
	CHA48 - Evitar que los miembros del equipo tengan demasiadas unidades de trabajo asignadas. Privilegiar el terminar unidades de trabajo asignadas en lugar de asignar (se) nuevas	Fácilmente superable
PRA37 - Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador		
	CHA35 - Que el equipo tenga un líder-facilitador en lugar de un jefe autoritario	Tendríamos algunos impedimentos
	CHA36 - Proactividad y habilidad de los miembros del equipo para auto-gestionarse individualmente y/o en equipo	Tendríamos algunos impedimentos
	CHA37 - Experiencia y capacidad de los miembros del equipo para tomar decisiones técnicas acertadas	Tendríamos algunos impedimentos
	CHA50 - Conseguir el apoyo de la dirección para implantar esta práctica	Ninguna, no existe este desafío
PRA39 - Postergar hasta último momento la asignación del encargado de realizar una actividad. En la medida que los miembros del equipo van terminando trabajo, van cogiendo nuevo trabajo desde el que está pendiente y priorizado		
	CHA52 - Desistir de realizar un balanceo de carga de trabajo de los miembros del equipo pues sólo tendrían asignado el trabajo en el cual están trabajando, no se realizarían asignaciones a futuro, salvo en casos excepcionales.	Tendríamos algunos impedimentos
PRA40 - Promover la sencillez en todos los aspectos. Centrarse en el mínimo conjunto de características que pueden constituir un producto útil y satisfactorio para el cliente		
	CHA51 - Que el cliente esté de acuerdo en recibir como entrega lo mínimo que pueda serle útil en un momento determinado	Tendríamos algunos impedimentos
PRA42 - Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas		
	CHA23 - Conseguir que los integrantes del equipo estén dispuestos a realizar actividades que no sean de su especialidad o preferencia	Sería difícil lograrlo
	CHA55 - Contar con una definición de puestos de trabajo y remuneraciones compatible con la diversidad de actividades que se pretende que puedan llegar a realizar los integrantes del equipo	Tendríamos algunos impedimentos
PRA43 - Acotar el ámbito de trabajo del equipo		
	CHA47 - Evitar que el equipo esté simultáneamente trabajando en demasiados productos, servicios o proyectos. Priorizar el trabajo de manera que un equipo no se vea obligado a distribuir su capacidad entre contextos de trabajo diferentes	Tendríamos algunos impedimentos
PRA44 - Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado		
	CHA50 - Conseguir el apoyo de la dirección para implantar esta práctica	Ninguna, no existe este desafío

RESULTADO

Código	Práctica Recomendada	Evaluación	Aplicación	Relaciones
PRA44	Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado	1.14	RECOMENDADA	--
PRA40	Promover la sencillez en todos los aspectos. Centrarse en el mínimo conjunto de características que pueden constituir un producto útil y satisfactorio para el cliente	1.01	RECOMENDADA	es apoyada por PRA02
PRA45	Formar equipos pequeños y estables	0.97	RECOMENDADA	apoya a PRA05, PRA13, PRA25, PRA26, PRA27, PRA28
PRA39	Postergar hasta último momento la asignación del encargado de realizar una actividad. En la medida que los miembros del equipo van terminando trabajo, van cogiendo nuevo trabajo desde el que está pendiente y priorizado	0.93	RECOMENDADA	--
PRA05	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico	0.90	RECOMENDADA	es apoyada por PRA45; apoya a PRA25, PRA26, PRA27, PRA28
PRA01	Abordar y entregar trabajo terminado de forma incremental	0.90	APLICADA	apoya a PRA23, PRA24, PRA34
PRA15	Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo	0.89	RECOMENDADA	--
PRA31	Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad	0.88	RECOMENDADA	es apoyada por PRA30, PRA43; apoya a PRA32, PRA35
PRA12	Trabajo o actividades realizadas en conjunto por dos o más integrantes	0.64	RECOMENDADA	--
PRA43	Acotar el ámbito de trabajo del equipo	0.61	RECOMENDADA	apoya a PRA31, PRA35
PRA42	Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas, que puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas	0.60	RECOMENDADA	--
PRA27	Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	0.59	RECOMENDADA	es apoyada por PRA05, PRA14, PRA45; requiere de PRA07
PRA26	Realizar reuniones de revisión del trabajo entregado	0.57	DESCARTADA	es apoyada por PRA05, PRA09, PRA30, PRA45
PRA30	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro	0.55	APLICADA	es requerida por PRA25, PRA28; apoya a PRA26, PRA28, PRA31, PRA32, PRA34, requiere de PRA08
PRA08	Visualización de todo el trabajo pendiente encargado al equipo	0.51	APLICADA	es requerida por PRA24, PRA25, PRA28, PRA30, PRA34
PRA06	Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo	0.49	RECOMENDADA	--
PRA14	Que exista un líder de mejora de proceso disponible para el equipo	0.45	APLICADA	apoya a PRA27
PRA32	No abusar de las horas extras, negociar y re-planificar oportunamente para evitarlo	0.45	DESCARTADA	es apoyada por PRA25, PRA30, PRA31, PRA34

PRA13	El equipo se auto-organiza y toma las decisiones técnicas	0.44	APLICADA	es apoyada por PRA18, PRA45; requiere de PRA37
PRA28	Realizar una reunión diaria del equipo al completo, cara a cara y muy breve	0.41	DESCARTADA	es apoyada por PRA05, PRA24, PRA30, PRA45; requiere de PRA07, PRA08, PRA30
PRA09	Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	0.38	RECOMENDADA	apoya a PRA24, PRA25, PRA26
PRA24	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado	0.37	RECOMENDADA	es apoyada por PRA01, PRA09; apoya a PRA25, apoya a PRA28, requiere de PRA08
PRA37	Cambiar la actitud del jefe desde un jefe autoritario hacia otro más de carácter líder y facilitador	0.37	RECOMENDADA	es apoyada por PRA18, es requerida por PRA13
PRA07	Establecer una disciplina de aprovechamiento de las reuniones	0.37	APLICADA	es requerida por PRA25, PRA27, PRA28
PRA35	Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	0.34	APLICADA	es apoyada por PRA31, PRA43
PRA04	Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla	0.34	APLICADA	es apoyada por PRA02
PRA18	Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	0.29	APLICADA	apoya a PRA13, PRA37
PRA10	Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	0.23	RECOMENDADA	--
PRA02	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente	0.18	APLICADA	apoya a PRA04, PRA40

La columna "Evaluación" muestra un valor según el cual se ordenan las prácticas recomendadas. Dicho valor se calcula considerando los siguientes aspectos: la contribución de la práctica a los objetivos seleccionados, la importancia de cada objetivo, el nivel de aplicación actual de la práctica, el esfuerzo que llevaría la preparación de la práctica y los desafíos para la aplicación de la práctica. La lista de prácticas ágiles recomendadas tiene limitaciones obvias, para el diagnóstico solo hemos recogido información muy general y estructurada. La idea es que este resultado sirva como Roadmap preliminar, el cual debería ser completado y adaptado considerando más detalles del contexto del equipo y de su trabajo en ciertos productos y/o servicios.