



MAESTRÍA EN GERENCIA DE SISTEMAS Y TECNOLOGÍAS DE LA
INFORMACIÓN

FORMULACIÓN DE UN MARCO DE TRABAJO PARA GESTIÓN DE
PROYECTOS DE SOFTWARE CON EQUIPOS GEOGRÁFICAMENTE
DISPERSOS.

Trabajo de Titulación presentado en conformidad a los requisitos establecidos
para optar por el título de Magister en Gerencia de Sistemas y Tecnologías de
la Información

Profesor guía

Giovanni Rafael Roldan Crespo

Autores

Dannys Mireya Luna Quelal
Omar Fernando Díaz Miranda

Año

2014

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con los estudiantes, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Giovanni Roldan

C.I.: 1708717887

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaramos que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Dannys Luna Quelal

C.I.: 0401118674

Omar Díaz

C.I.: 1711706554

AGRADECIMIENTOS

Agradecemos a Giovanni Roldan por sus buenos consejos durante la realización de esta tesis.

A nuestros padres por su apoyo incondicional

Los Autores

DEDICATORIA

A nuestro hijo Dylan

Los Autores

RESUMEN

La evolución de las tecnologías de información, sobre todo en el campo de las comunicaciones, la externalización y la globalización de los servicios, ha favorecido a la conformación de equipos multidisciplinarios geográficamente dispersos, los cuales a través de los años se vuelven más numerosos. Estos proyectos que inician con retos adicionales de gestión inherentes a su naturaleza, requieren de prácticas específicas para asegurar su éxito.

Cuando hablamos de gestión de proyectos nos vienen a la mente múltiples métodos y técnicas probadas y reconocidas por la industria. Pero cuando tenemos que aplicarlas en la gestión de proyectos de desarrollo de software con equipos geográficamente dispersos nos encontramos con que son insuficientes o muy superficiales y nos vemos obligados a recurrir a la intuición o experiencia para suplir estos vacíos.

El propósito del presente trabajo es formular un marco de trabajo que facilite la gestión de proyectos de software cuando los integrantes del equipo están geográficamente dispersos, aportando con una serie de prácticas específicas para cada una de las fases de la gestión de un proyecto de desarrollo de software bajo un enfoque ágil, iterativo e incremental en donde las personas juegan un papel fundamental y las tecnologías de información son importantes habilitadores y facilitadores de las actividades y prácticas diarias del proyecto.

ABSTRACT

The evolution of information technologies, especially in the field of communications, globalization and outsourcing services, has encouraged the creation of multidisciplinary teams geographically distributed, which through the years become more numerous. These projects start with additional management challenges inherent in their nature, require specific practices to ensure success.

When we talk about project management come to mind multiple methods and proven techniques and recognized by industry. But when we have to apply them in managing software development projects with geographically dispersed teams, we find that (those are), insufficient or very shallow and we are forced to rely on intuition or experience to fill out these gaps.

The purpose of this paper is to formulate a framework that facilitates the management of software projects when team members are geographically dispersed, providing a number of specific practices for each of the phases of managing a development software project in an agile, iterative and incremental approach where people play a crucial role and information technologies are important enablers and facilitators of everyday activities and practices of the project.

ÍNDICE

1	Capítulo I: Estado del Arte	3
1.1	El Teletrabajo.....	3
1.1.1	Definición.....	3
1.1.2	Importancia.....	5
1.1.3	Ventajas y desventajas	6
1.2	Equipos distribuidos.....	8
1.2.1	Introducción	8
1.2.2	Conceptos	9
1.2.3	Ventajas y Desventajas	11
1.2.4	Organización.....	12
1.2.5	Comunicación	13
1.2.6	Tecnología.....	21
1.3	Metodologías de desarrollo de software	24
1.3.1	Metodologías formales.....	24
1.3.2	Metodologías ágiles	30
1.4	Trazabilidad en proyectos de desarrollo de software	42
1.4.1	Definición de trazabilidad.....	42
1.4.2	Ventajas y desventajas de la trazabilidad de requerimientos	43
1.5	Gestión de proyectos.....	43
1.5.1	Definición.....	43
1.5.2	Gestión de proyectos de software.....	45
1.5.3	Gestión de proyectos formales	45
1.5.4	Gestión de proyectos ágiles.....	48
2	Capítulo II: Desarrollo del Marco de Trabajo	61
2.1	Por qué un marco de trabajo para gestión de proyectos de software con equipos geográficamente dispersos.....	61
2.1.1	Por qué es necesaria una gestión ágil de proyectos de software	62

2.2	Gestión de proyectos de software.....	63
2.2.1	Descripción.....	63
2.2.2	Comparación de metodologías	64
2.2.3	Síntesis de los modelos de gestión	66
2.3	Casos de estudio	68
2.3.1	Caso de Estudio 1.....	68
2.3.2	Caso de Estudio 2.....	69
3	Capítulo III: Marco de trabajo propuesto.....	72
3.1	Visión general	72
3.2	Fases de la gestión de proyectos de desarrollo de software del marco de trabajo propuesto	73
3.2.1	Conceptualización.....	74
3.2.2	Especulación	74
3.2.3	Exploración.....	75
3.2.4	Revisión.....	75
3.2.5	Cierre.....	76
3.3	Dominios del marco propuesto.....	76
3.3.1	Conformación de equipos	77
3.3.2	Comunicación	87
3.3.3	Tecnologías de información	93
3.3.4	Trazabilidad del proyecto.....	104
3.4	Resumen de prácticas por dominio.....	107
3.4.1	Dominios por fase.....	110
4	Capítulo IV: ¿Como una empresa tradicional puede aplicar el marco propuesto?	111
4.1	Visión General	111
4.2	Transición a una organización ágil.....	111
4.2.1	Alta dirección y liderazgo	113
4.2.2	Estructura organizacional.....	113

4.2.3	La Gente	114
4.2.4	Tecnología	114
4.3	Transición de una gestión tradicional a una gestión ágil... ..	115
4.3.1	Equivalencia de las fases del proyecto	115
4.3.2	Gestión de la integración del Proyecto.....	115
4.3.3	Gestión del alcance y tiempo del proyecto.....	116
4.3.4	Gestión de costos del proyecto	117
4.3.5	Gestión de calidad del proyecto	117
4.3.6	Gestión de RRHH del proyecto	118
4.3.7	Gestión de las comunicaciones del proyecto	119
4.3.8	Gestión de riesgos del proyecto.....	119
4.3.9	Gestión de stakeholders del proyecto	119
4.4	Resumen del marco de trabajo	119
4.4.1	Transicionar a una organización ágil.....	120
4.4.2	Transicionar a un marco de gestión ágil.....	121
4.5	Futuras líneas de investigación.....	124
5	Capítulo V: Conclusiones y Recomendaciones	125
5.1	Conclusiones	12
5.2	Recomendaciones.....	126
	REFERENCIAS.....	126
	ANEXOS.....	129

ÍNDICE DE FIGURAS

Figura 1. Elementos de la comunicación	14
Figura 2. Tipos de comunicación.....	15
Figura 3. Porcentajes del proceso comunicativo	16
Figura 4. Proceso de desarrollo FDD	39
Figura 5. Prácticas efectivas para modelado y documentación	40
Figura 6. Interacción entre las capas de Scrum	52
Figura 7. Clasificación de proyectos por color según Crystal	56
Figura 8. Ciclo de vida ASD	60
Figura 9. Fases de la gestión de proyectos	74
Figura 10. Líneas de comunicación.....	88
Figura 11. Horario común.....	93
Figura 12. Horario con igual afectación para los equipos	93
Figura 13. Videoconferencia H.323	98
Figura 14. Modelo de Trazabilidad basado en el requerimiento	106
Figura 15. Transición de una gestión tradicional a una gestión ágil.....	112
Figura 16. Equivalencia de fases tradicionales vs ágiles	115

ÍNDICE DE TABLAS

Tabla 1. Beneficios y dificultades del uso de la tecnología en equipos distribuidos.....	22
Tabla 2. Comparación entre metodologías ágiles y tradicionales	64
Tabla 3. Comparación entre metodologías tradicionales	65
Tabla 4. Comparación entre metodologías ágiles	65
Tabla 5. Comparación entre la gestión de proyectos formal y ágil.....	66
Tabla 6. Capacidad requerida de hardware y software según calidad esperada	96
Tabla 7. Resumen del dominio: Conformar equipos distribuidos	107
Tabla 8. Resumen del dominio: Comunicación	108
Tabla 9. Resumen del dominio: Tecnologías de la información.....	108
Tabla 10. Resumen del dominio: Trazabilidad.....	109
Tabla 11. Matriz de dominios por fase	110

INTRODUCCIÓN

Durante los últimos años el sector informático ha crecido. Según el Servicio de Rentas Internas (SRI) en el año 2011 alcanzó ingresos totales de USD 260 millones. De acuerdo a un estudio de mercado realizado por la Asociación Ecuatoriana de Software (AESOFT), la Superintendencia de Compañías registró 633 empresas, a nivel nacional, en el sector “Programación Informática, Consultoría de Informática y actividades conexas”.(AESOFT, 2011)

La globalización ha fomentado la externalización de actividades o servicios con la contratación de proveedores o especialistas que ofrecen servicios a costos reducidos y con amplia experiencia; lo cual provoca un cambio en la forma de trabajo ya que los equipos terminan localizados en diferentes lugares y estos proyectos se realizan de forma distribuida. Otra modalidad que ha surgido es el teletrabajo que según la consultora estadounidense International Data Corporation (IDC) para el 2015 el 37.2% de la población mundial trabajará a distancia. Mediante estos cambios las empresas logran una mejora competitiva, reducción de costos, tendiendo a estructuras planas, descentralizadas y de naturaleza global. (IDC, 2013)

Debido a la evolución de las Tecnologías de la Información y de la Comunicación (TICs), sobre todo en el campo de las comunicaciones y a la externalización y globalización de los servicios antes mencionada, las empresas de desarrollo de software tienen la oportunidad de formar equipos multidisciplinarios que estén geográficamente dispersos, en algunos casos hasta separados por diferentes usos horarios, que aprovechan las fortalezas y conocimientos de sus integrantes, así como las tecnologías existentes, para lograr la creación de software de calidad a menores costos. Estos equipos enfrentan múltiples retos para conseguir agilidad durante el proyecto, coordinación entre los miembros del equipo, un correcto entendimiento de los requerimientos y un adecuado aprovechamiento de los medios de comunicación y herramientas disponibles.

Para lograr lo antes mencionado es necesario desarrollar un marco de trabajo, que si bien puede estar basado en los marcos y metodologías existentes, permita coordinar, controlar y promueva un correcto funcionamiento de estos equipos de trabajo con la finalidad de alcanzar los objetivos planteados en los proyectos de desarrollo de software.

El marco propuesto en esta tesis está basado principalmente (aunque no exclusivamente) en metodologías de gestión ágiles, por lo que recurrentemente se observaran comparaciones y equivalencias con metodologías tradicionales

1 Capítulo I: Estado del Arte

1.1 El Teletrabajo

1.1.1 Definición

Refiriéndonos a la interpretación etimológica de Teletrabajo podríamos decir que se refiere al trabajo a distancia (tele proviene del griego “telos” – a distancia). El Teletrabajo permite realizar un trabajo no presencial, es decir fuera del lugar habitual de trabajo, ya sea desde el hogar o cualquier otro sitio; por lo que, no hay que confundirlo con trabajo a domicilio, ya que presenta grandes diferencias, especialmente que para la realización del trabajo es necesario el uso de las tecnologías de información y comunicación (TIC)

El término Teletrabajo aparece en Estados Unidos en la década de los años 70 por parte del físico Jack Nilles. Debido a la crisis del petróleo se buscaba la manera de reducir los desplazamientos, por lo que Nilles tuvo la idea de llevar el trabajo al trabajador y no el trabajador al trabajo y creó el concepto de “telecommuting”(JALA International, 2013), pero sería posteriormente con el avance tecnológico, el avance de la velocidad en las redes de comunicación, la reducción de costos y el Internet que se tendrían las herramientas necesarias para que se pueda difundir y globalizar el Teletrabajo.

“Existen muchas formas de designar el trabajo, fuera del espacio habitual de la organización. Entre las más comunes están: Teletrabajo, telenetworking o telework, en Europa o telecommuting y Mobile Worker, en los EEUU”(Foro de empresas efr, 2012)

Esta gran variedad de términos provocó que no se pueda normar el concepto ya que por ejemplo para Nilles un trabajo telefónico es telecommuting, no así para la definición Europea. Además que el avance de las tecnologías también obliga al concepto a ir evolucionando.

En este punto consideramos importante revisar algunas definiciones:

“Según el Acuerdo marco Europeo define el Teletrabajo como un tipo de prestación, que utiliza las tecnologías de la información y la comunicación online con el empleador y/o con el cliente, y se realiza desde lugares remotos o alejados de la empresa u organización, con la que se tiene vínculos contractuales. El contenido de este acuerdo gira en torno a tres ejes fundamentales:

- El objetivo de mantener un justo equilibrio entre la flexibilidad y la seguridad de la organización del trabajo
- La idea de compatibilizar, la vida familiar con la vida laboral de los trabajadores
- La mejora de la productividad y competitividad de las empresas”(Foro de empresas efr, 2012)

Antonio Padilla Meléndez, profesor Titular de Dirección de Empresas de la Universidad de Málaga y Fundador y moderador de la lista de correo "Teletrabajo y Organización Virtual" de la RedIRIS (2003) declara que el Teletrabajo no debe únicamente ser visto como una cuestión del lugar físico de trabajo y propone el término e-work para de esa manera evitar connotaciones negativas que se asocian al Teletrabajo o hablar de trabajo flexible. “Quizás debiéramos hablar más bien de trabajo flexible, de mejora de la calidad de vida del empleado, de cambio organizativo global en la empresa, y no solo de cambio en la organización del trabajo, de algo normal, cotidiano, que nuestro vecino, familiar o nosotros mismos podemos estar haciendo sin ni siquiera denominarlo así” (Padilla, 2013)

“Según un estudio de ENTER (iebusinessschool), entiende el trabajo virtual, como una forma de trabajar, en la que algunos miembros de la organización, trabajan de forma no presencial, bien a través del teléfono, email, redes sociales, pero no necesariamente desde el hogar”(Foro de empresas efr, 2012)

La Asociación Española de Teletrabajo define el término e-work como “una oficina real desarrollada en el mundo virtual; colaboración en equipo con una

interrelación verdaderamente sinérgica y productiva. Los distintos componentes se encuentran diseminados geográficamente pero unidos en el proyecto común componiendo una organización distribuida”(Sánchez, 2012)

Unificando estas definiciones podemos decir que el Teletrabajo tiene tres componentes esenciales:(Belzunagui, 2001)

1. El lugar del trabajo debe realizarse en una ubicación diferente a la empresa, pudiendo ser el domicilio del trabajador o cualquier otro
2. Para la realización de las actividades laborales es necesario e imprescindible el uso de las TIC. Esta característica excluye a todo tipo de trabajo manual o a destajo, ya que estos no deben considerarse Teletrabajo
3. El trabajo realizado a distancia es de forma habitual y no ocasional por parte del trabajador

1.1.2 Importancia

El Teletrabajo ha sufrido muchos cambios desde su primera definición en 1973 por parte de Jack Nilles, donde se buscaba llevar el trabajo al trabajador. Ahora con los crecientes avances en las TIC este concepto ya no solo se limita a realizar trabajo en el domicilio, sino que este puede realizárselo en cualquier lugar, como es el caso del Teletrabajo móvil. Desde el siglo pasado estos cambios constituyen una verdadera revolución científica y tecnológica en el campo de la telemática.

El Teletrabajo presenta múltiples ventajas significativas tanto para el empleador como para el empleado y para la sociedad en general, sin embargo tampoco son menores las debilidades del mismo. Como en la actualidad somos parte de una globalización donde predomina la importancia y uso de las TIC, es imprescindible hacer el mejor uso de las mismas para lograr una ventaja competitiva, mediante el ahorro de costos de operativos, disminución de tiempos de desplazamiento, aumento de productividad, con flexibilidad horaria y con inclusión laboral incluso de personas discapacitadas y de diferentes localizaciones geográficas.

Por ejemplo, la European Commission revela un déficit de 700,000 profesionales TIC en Europa (European Commission, 2013), mientras que en países como España preocupan los niveles de desempleo y en cambio Latinoamérica está lista para ofrecer buenos profesionales al mundo. Por lo que un adecuado uso del Teletrabajo ofrece la posibilidad de solventar estas y otras necesidades.

Y es importante destacar la parte final de “¿Por qué el Teletrabajo? Del libro blanco del Teletrabajo de España, que dice:

“Asimismo, el Teletrabajo constituye un paso inicial hacia la introducción de políticas de Conciliación y trabajo Flexible (nuevas formas de trabajo), en las Organizaciones en las que se practica” (Foro de empresas efr, 2012)

Sin embargo, es importante considerar que para que esta modalidad de trabajo pueda ser aprovechada ventajosamente requiere de la voluntad política que debe normar los marcos legales de forma adecuada y aunque no es el único modelo de relación laboral si representa una oportunidad, que con ventajas y desventajas, es capaz de aprovechar la tecnología para apoyar a las estrategias empresariales.

1.1.3 Ventajas y desventajas

La adopción del Teletrabajo en la actividad laboral trae consigo ventajas y desventajas, las cuales deben ser analizadas desde 3 puntos de vista: la del trabajador, la de la empresa y la de la sociedad

A continuación se listan las ventajas y desventajas más representativas

1.1.3.1 Para el trabajador

1.1.3.1.1 Ventajas

- Flexibilidad horaria, autonomía
- Optimización en tiempos de desplazamiento
- Ahorro
- Mejor balance vida/trabajo

- Mayor oportunidad laboral
- Entorno de trabajo a elección del trabajador
- Reducción del estrés

1.1.3.1.2 Desventajas

- El ambiente de trabajo puede no ser idóneo
- Puede provocar sedentarismo
- Pérdida de relaciones personales con otros trabajadores
- El trabajo puede ser realizado en horarios no limitados
- Puede provocar sentimiento de aislamiento

1.1.3.2 Para la empresa

1.1.3.2.1 Ventajas

- Reducción de costos operativos y de bienes raíces
- Optimización del tiempo laboral
- Menor deserción
- Mejor cumplimiento de metas al implantar cumplimiento por objetivos
- Menores tiempos de respuesta
- Mayor facilidad de acceso a profesionales de alto nivel
- Mejor aprovechamiento de los puestos de trabajo ya que incluso pueden ser compartidos
- Menor absentismo debido a enfermedad
- Mayor facilidad para el cumplimiento de leyes de inclusión laboral a minusválidos

1.1.3.2.2 Desventajas

- Pueden existir problemas de lealtad e identificación del empleado con la empresa
- Menor participación del empleado debido al aislamiento físico.

1.1.3.3 Para la sociedad

1.1.3.3.1 Ventajas

- Generación de empleo
- Inclusión laboral de personas con discapacidad y madres cabeza de familia
- Movilidad más eficiente y reducida
- Reducción de gases contaminantes
- Aumento de inserción laboral a personas residentes en sectores rurales

1.1.3.3.2 Desventajas

- No todos los trabajadores tienen acceso o manejan adecuadamente las tecnologías de la información
- Las condiciones y medio ambiente de trabajo puede terminar no siendo adecuado, sin tener cómo controlarlo

1.2 Equipos distribuidos

1.2.1 Introducción

El oír hablar de que un mismo proyecto está siendo desarrollado en varios puntos del mundo, ya no es inconcebible. Ahora es la realidad de muchas empresas que aprovechan y sacan ventaja de la globalización y el avance en las comunicaciones y el Internet. Así como también se saca provecho del trabajo en equipo, ya que el espíritu de grupo es lo que a muchas empresas les da una gran ventaja competitiva.

Al desarrollar software en estas condiciones, las personas se enfrentan a un cambio de paradigma, estar de un espacio físico y presencial a interactuar con personal en distintas partes del mundo; cambia la dinámica, la comunicación e interacción, así como las tácticas para mantener un buen ambiente laboral y el poder llevar al éxito los proyectos planteados.

Manejar un proyecto global y mantener la comunicación del equipo, han sido aspectos que se han venido manejando de manera manual, apoyándose en documentos y reuniones. Para solventar esto, habría que analizar el uso de herramientas de apoyo, así como revisar la metodología.

Los equipos distribuidos que trabajan conectados, se están convirtiendo en un elemento clave en la estrategia de las organizaciones, el desarrollo del trabajo y la gestión de las organizaciones.

1.2.2 Conceptos

Al hablar de equipos distribuidos, y de acuerdo a las necesidades que se han ido presentando en las empresas se los podría clasificar y podrían ser: Trabajo a distancia, Outsourcing, equipos distribuidos – personas no dispersas, equipos distribuidos – personas dispersas.

Trabajo a distancia: Sería cuando se habilita el poder trabajar pocas horas desde casa, considerando como herramientas, una laptop y una conexión que abastecerá este periodo.

Outsourcing: Es subcontratar el desarrollo para que sea realizado por terceros.

Equipos Distribuidos, personas no dispersas: las personas trabajan en distintas empresas y asignadas a un proyecto local, ya las responsabilidades se encuentran limitadas en la estructuración del proyecto global, con esto se minimizaría la dependencia.

Equipos Distribuidos, personas dispersas: personas en distintos puntos del mundo que trabajan en un mismo proyecto. En este caso el contacto no existe, y hay que tener cuidado de cómo se avanzará con los desarrollos.

Equipos virtuales, se define a la reunión de varias personas trabajando en un mismo proyecto ubicadas en distintos puntos geográficamente distantes y que aprovechan de las TICs para interactuar.

Desvirtualización, término que aparece para poder definir la situación cuando las personas que interactúan virtualmente, se materializan, o se conocen en el mundo real.

1.2.2.1 7 Hábitos de los equipos virtuales altamente efectivos(Levit, 2012)

1. Los equipos virtuales altamente efectivos se componen de empleados con las “Tres A” (en inglés): asertividad, responsabilidad y capacidad para trabajar de forma independiente.

Son personas que tienen claro el trabajar dispersos, ya que asumen sus responsabilidades y trabajan autogestionando las tareas asignadas.

2. Los equipos virtuales altamente efectivos comprenden las expectativas. El feedback se comunica claramente. No hay supuestos.

3. Los equipos virtuales altamente efectivos están imbuidos de la tecnología pertinente.

Conocen y hacen uso de las herramientas sofisticadas de comunicación que tienen a su alcance.

4. Los equipos virtuales altamente efectivos están familiarizados con la dinámica en persona.

Son personas sociables, que con una reunión en persona logran hacer lazos con el equipo.

5. Los equipos virtuales altamente efectivos tienen un director visible.

No requieren altos niveles de supervisión, se mantienen comprometidos, productivos y menos estresados, con el hecho de conocer que hay un

jefe que también está trabajando como ellos. Conocen como sus actividades impactan al proyecto.

6. Los equipos virtuales altamente efectivos construyen y mantienen relaciones sólidas.

Saben lo importante que es hablar a tiempo de los conflictos y aprenden a conocer a sus compañeros. Comprenden que el compañerismo es útil para poder crear lazos.

7. Los equipos virtuales altamente efectivos dirigen grandes reuniones.

Son organizados, puntuales y respetuosos del tiempo. Los debates se realizan para aportar y llegar a consensos. Mantienen las reglas para que las reuniones sean productivas.

1.2.3 Ventajas y Desventajas

Al formar los equipos distribuidos podemos llegar a aprovechar todo el potencial que se está constituyendo al reunir personas que pueden aportar mucho a los proyectos, sin dejar de revisar los problemas que podrían presentarse, pero que manteniéndolos bajo supervisión no deberían convertirse en problemas.

- Entre las ventajas que se podrían mencionar están:
- Diversidad de conocimientos
- Ahorros para las empresas
- Diferentes habilidades
- Diferentes experiencias
- Nuevos conocimientos de mercados
- No hay limitaciones horarias

Así también se debe considerar las desventajas, para ponerles más atención y que no sean un tope para el desempeño del equipo:

- Distintos idiomas

- Diferencias culturales
- Coordinar horas en común
- No hay interacción cara a cara
- Aumenta el grado de dificultad al querer realizar acuerdos.

1.2.4 Organización

Tanto para las personas, como para la organización el cambio de paradigma conlleva a revisar desde su estructura, sus normativas y políticas, para conseguir mantener un ambiente de trabajo saludable, y fortalecer la formación de los equipos virtuales efectivos.

Hay varios aspectos a considerar:

La empresa ha de revisar el progreso de adaptación de los equipos.

Concientizar acerca de las relaciones que se provocarán de este cambio, cuidándolas y afianzándolas, para sacar el mejor provecho de esta fusión multicultural.

Observar la nueva interacción como equipo distribuido.

Fortalecer los lazos, para generar unión en el equipo, con el objetivo de hacer más agradable el ambiente de trabajo.

Facultar a los integrantes del equipo, para ayudarles a que puedan realizar un trabajo autogestionado, eficiente y su nivel de reporte sea óptimo.

Trabajar en estrategias de liderazgo para obtener un equipo comprometido con los objetivos globales.

Definición clara de los roles y responsabilidades.

Considerar las habilidades sociales de los integrantes del equipo que apoyará de mejorar manera en este nuevo proceso.

Así también las empresas deben analizar los pros y los contras de este proceso, para trabajar en las debilidades para que no se conviertan en un tope en el proceso.

Las ventajas que este cambio provoca, podrían ser el poder elegir a los mejores talentos, aprovechar el tiempo, comunicación sincrónica y asincrónica, costes más bajos, reducción de desplazamientos y más ecológico.

Además habría que analizar cómo manejar el tema del idioma, metodología de trabajo, responsabilidades, multiculturalidad, balance personal y profesional, aislamiento y costumbres.

Las empresas además deben cambiar sus lineamientos de evaluación del personal, ya que al no ver a las personas, lo único a evaluar es su trabajo.

Los indicadores se vuelven útiles e indispensables, para poder llevar el monitoreo y control.

1.2.5 Comunicación

La principal diferencia entre los equipos tradicionales y los virtuales de trabajo o de proyecto, radica no sólo en una cuestión de ubicación física (donde están alocados los recursos) sino que radica en cómo esos individuos se van a comunicar y colaborar entre sí para alcanzar los objetivos y metas de la organización o de un proyecto determinado. La imposibilidad de verse, de conocerse con más profundidad, de verse a diario, de reunirse y desayunar juntos, de realizar actividades por fuera del horario de trabajo, etc, son algunas de las limitaciones que los equipos virtuales tienen, y muchas veces estas limitaciones afectan al desarrollo del espíritu de trabajo.

Como se analizará posteriormente, los gerentes de proyectos que deban trabajar con equipos virtuales de trabajo, deberán prestar especial atención a las cuestiones relativas a la comunicación y colaboración entre los integrantes del equipo de proyecto, y sus respectivas cuestiones culturales que seguramente tendrán impacto en la forma de conducirse.

Por este motivo se dice que las actividades de gestión de equipos virtuales es más compleja que la gestión de equipos tradicionales, debido a que la tecnología mal aplicada podría incurrir en problemas serios de comunicación,

el desconocimiento de la cultura y la falta de comunicación correcta en el momento adecuado podría incurrir en una falta grave para el proyecto.

La mayor parte de nuestras vidas interactuamos con nuestro alrededor, esto se refiere a tener contacto con otras personas. Al trabajar con equipos dispersos esta parte social, se verá interrumpida y se deberá apoyar de los artefactos que permitan un canal de comunicación amigable.

1.2.5.1 Elementos de la Comunicación

En la comunicación intervienen diversos elementos que pueden facilitar o dificultar el proceso. Estos son:

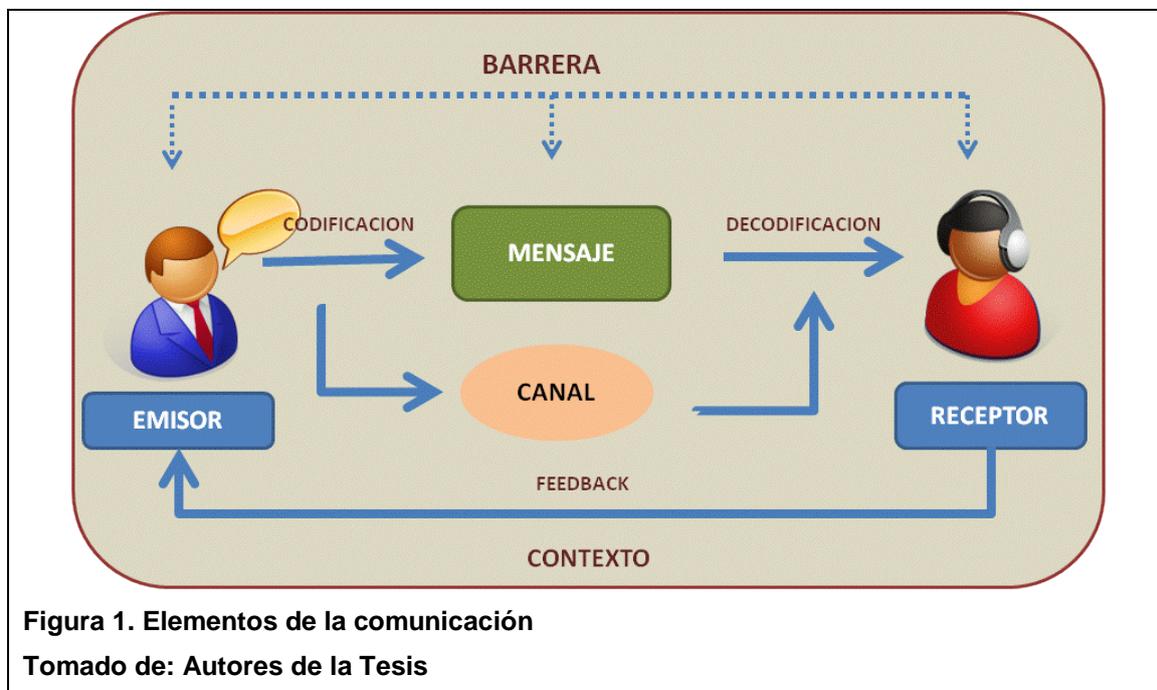


Figura 1. Elementos de la comunicación

Tomado de: Autores de la Tesis

Emisor: La persona (o personas) que emite un mensaje.

Receptor: La persona (o personas) que recibe el mensaje.

Mensaje: Contenido de la información que se envía.

Canal: Medio por el que se envía el mensaje.

Código: Signos y reglas empleadas para enviar el mensaje.

Contexto: Situación en la que se produce la comunicación.

La comunicación eficaz entre dos personas se produce cuando el receptor interpreta el mensaje en el sentido que pretende el emisor.

Cuando las personas se comunican, el mensaje que transmite el emisor tendrá cierto impacto sobre el receptor.

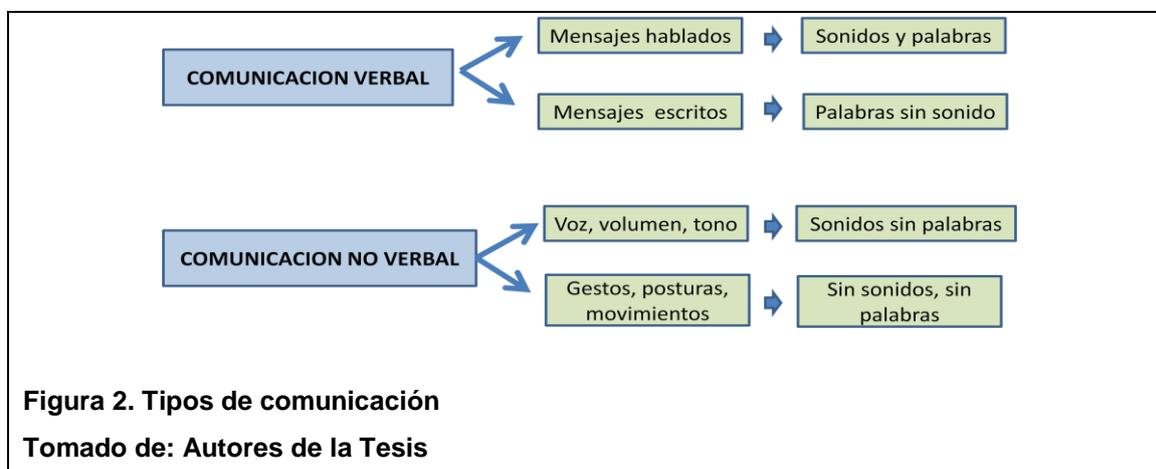
1.2.5.2 Tipos de Comunicación

Se distinguen los tipos de comunicación: Comunicación Verbal y Comunicación No Verbal, Síncrona y Asíncrona, Escritas y Orales.

Comunicación Verbal, se refiere a las palabras que utilizamos y a las inflexiones de nuestra voz (tono de voz).

Comunicación No Verbal, es el paralenguaje (aspectos no semánticos del lenguaje con los cuales se comunica el significado expresivo: voz, timbre, volumen, tono, potencia, silencios, pausas...) hace referencia a un gran número de canales, entre los que se podrían citar como los más importantes el contacto visual, los gestos faciales, los movimientos de brazos y manos o la postura y la distancia corporal, de ahí que el peso específico de la comunicación recae en la no verbal.

Para los equipos distribuidos que asumimos que estarán muy distantes se debería tener especial atención en la comunicación No Verbal para minimizar los efectos, ya que es la parte de la comunicación que no será visible.



En el año 1971 el antropólogo e investigador Albert Mehrabian, luego de haber realizado varios estudios sobre las relaciones humanas y la comunicación, estableció unos porcentajes que definen la proporción en que interviene cada una de las tres “uves” en todo proceso comunicativo:

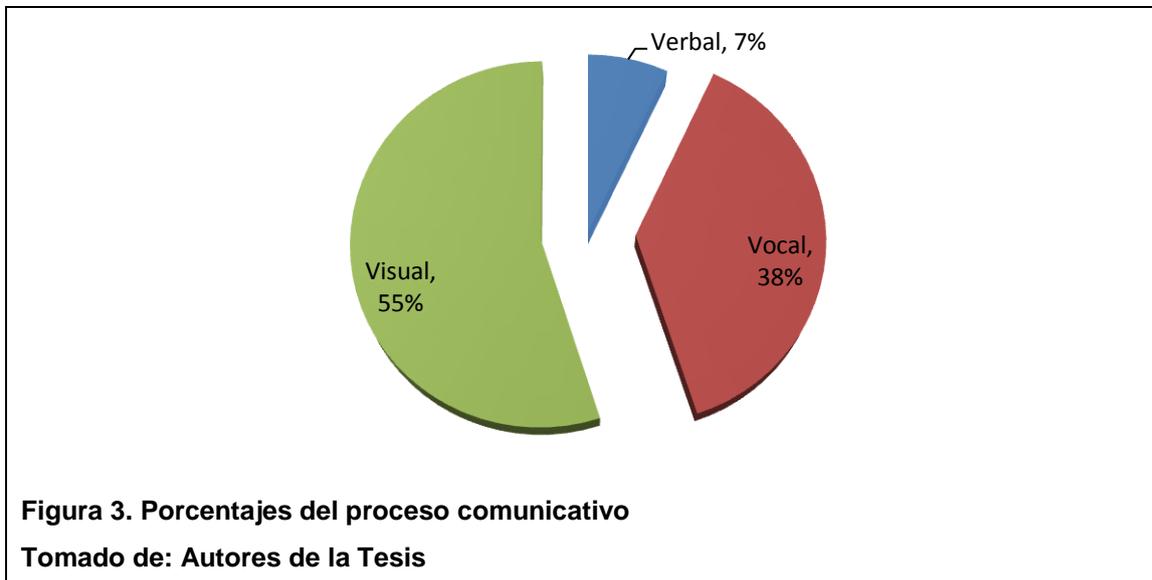
“Verbal” las palabras que se pronuncian,

“Vocal” la manera en que se pronuncian las palabras y,

“Visual” que hace referencia al aspecto, los movimientos, las posturas, etc.

Los porcentajes obtenidos fueron los siguientes:

- Verbal 7% (palabras)
- Vocal 38% (paralenguaje)
- Visual 55% (gestos, posturas, etc.)



Observando el gráfico podría deducirse que en el caso de los equipos virtuales de trabajo, sobre todo los que no tienen contacto visual, es posible que surja algún inconveniente adicional al momento de comunicarse. Por tal razón, es importante considerar con atención los canales de comunicación que se

establecerán, las formas de comunicación y las herramientas de tecnología que podrían ayudar a eliminar las barreras comunicacionales.

Comunicación Asincrónica, el receptor recibe el mensaje y su respuesta puede no ser inmediata.

Algunos ejemplos de estas herramientas son:

- Email
- Foros
- *Blogs*
- Intercambio de archivos
- *Streaming*
- Tablón de anuncios
- Calendario

Comunicación Sincrónica, el emisor y receptor interactúan en el mismo instante.

Ejemplos de herramientas sincrónicas son:

- Mensajería instantánea.
- Audioconferencia.
- Videoconferencias
- Entornos colaborativos avanzados
- *Webinars*
- *Webcast*

Comunicación Escrita, La interacción entre el emisor y el receptor no es inmediata e incluso puede llegar a no producirse nunca, aunque aquello escrito perdure eternamente.

- **Formal**
 - Plan de proyecto
 - Informe de avance
 - Revisión de calidad

- **Informal**
 - Memos

Comunicación Oral, es aquella que se establece entre dos o más personas, tiene como medio de transmisión el aire y como código un idioma.

- **Formal**
 - Presentaciones
 - Exposiciones
- **Informal**
 - Conversaciones

1.2.5.3 Barreras de comunicación

"El problema más grande en la comunicación es la ilusión de que ha tenido lugar." - George Bernard Shaw

Las barreras de comunicación son elementos que producen fallas en la comunicación y afectan a las personas y al proyecto. Las que se podrían mencionar para tomar acciones y son las siguientes:

- Falta de conocimiento
- Diferencias culturales
- Clima Organizacional
- Canales de comunicación poco claros
- Comunicaciones indirectas
- Mensajes poco claros
- Dificultades con el lenguaje
- No escuchar
- Sentido equivocado del mensaje
- Mala interpretación.

1.2.5.4 Características de la comunicación para equipos distribuidos

Para equipos distribuidos se distinguen dos características para los casos extremos de ubicación de sus integrantes.

- Comunicación asincrónica, puede provocar que los tiempos efectivos de comunicación se aumenten.
- Imposibilidad de comunicación no verbal, esta característica es altamente importante, debido a lo antes mencionado de la cantidad de información visual que se genera con las reuniones presenciales.

Gestionar proyectos eficientemente implica en gran medida, el establecer canales de comunicación apropiados, tanto con el equipo de proyecto como con el resto de las personas interesadas, para comunicar y recibir las comunicaciones relevantes al proyecto. Los errores más frecuentes en las comunicaciones, es dar por entendido que el receptor sabe o entiende lo que el emisor transmite. En muchas ocasiones es necesario asegurarse que el emisor comprende correctamente el mensaje enviado.

“Una comunicación eficaz crea un puente entre los distintos interesados (stakeholders) involucrados en el proyecto, conectando diferentes entornos culturales y organizacionales, diferentes niveles de experiencia, y perspectivas e intereses diversos en la ejecución o resultado del proyecto” (Extraído de PMBok 4ta edición, pág. 243.).

1.2.5.5 Normas de etiqueta en el uso de Internet

También llamado Netiqueta o Netiquette, es una palabra derivada del francés etiquette y del inglés net (red) o network. Y es la etiqueta que se utiliza para comunicarse en la red, se podría decir también la etiqueta usada en el ciberespacio, conformado por una serie de reglas para comportarse de manera adecuada en la red.

A continuación se exponen 10 reglas básicas (Shea, 2012):

Regla No. 1: Recuerde lo humano.

No hagas a otros lo que no quieres que te hagan a ti, ponerse en los zapatos del otro. Defenderse pero sin herir los sentimientos de los otros.

REGLA No. 2: Siga en la Red los mismos estándares de comportamiento que utiliza en la vida real

Mantenerse dentro de los mismos parámetros de ética y moral aceptables.

REGLA No. 3: Sepa en qué lugar del ciberespacio está

Observe antes de saltar. Lo permitido en algunos sitios puede ser una falta grave en otros. Al participar en los chats formarse la idea de cómo actúan las personas que ya están participando y luego actuar.

REGLA No. 4: Respete el tiempo y el ancho de banda de los demás

Asegúrese que lo que Ud. está aportando o enviando será de utilidad para el equipo, no malgaste el ancho de banda, ni el tiempo de los demás.

REGLA No. 5: Ponga de su parte, véase muy bien en línea

Será usted juzgado eso sí por la calidad de su escritura. Para la mayoría de las personas que escogen comunicarse en línea esto es una ventaja; si no disfrutaran utilizando la palabra escrita, no estarían allí. Esto quiere decir que la redacción y la gramática cuentan.

REGLA No.6: Comparta el conocimiento de los expertos

No tenga temor de compartir con otros lo que usted sabe.

Compartir sus conocimientos es satisfactorio. Ha sido una tradición de la Red y convierte el mundo en un sitio mejor.

REGLA No.7: Ayude a que las controversias se mantengan bajo control

Se denomina "apasionamiento" cuando la gente expresa su opinión con mucha fuerza sin ponerle freno a sus emociones.

REGLA No.8: Respeto por la privacidad de los demás

No leer la correspondencia ajena. Podría llevarles incluso a perder su trabajo.

REGLA No.9: No abuse de las ventajas que pueda usted tener

Algunas personas tienen en el ciberespacio más influencia que otras. Existen expertos en toda clase de juegos de realidad virtual, expertos en todo tipo de software de oficina y expertos en administrar toda clase de sistemas.

Saber más que los otros, o tener un mayor conocimiento de cómo funcionan los distintos sistemas, no le da a usted el derecho de aprovecharse de los demás. Por ejemplo, los administradores de los sistemas no deben leer nunca el correo de otros.

REGLA No. 10: Excuse los errores de otros

Si alguien comete errores de ortografía, haga un comentario fuerte, haga una pregunta tonta o de una respuesta innecesariamente larga, sea paciente y sepa corregir de manera amable.

1.2.6 Tecnología

Centrándonos en el objetivo de que la distancia ya no será un limitante para el desarrollo de un proyecto, la tecnología juega un papel primordial en este proceso. Los tiempos de respuesta deben permitir que las tareas fluyan sin inconvenientes.

La diversidad de herramientas ya concebidas con la visión de distribuidas, nos facilitarán la interacción, así es como a través de la videoconferencia ya acercamos a los grupos de trabajo. Y las redes tanto internas como externas nos permiten el interactuar y retroalimentar al proyecto.

Cobra mayor importancia la gestión de la documentación y del conocimiento, y habrá que determinar qué tipo de información se enviará y recibirá en cada fase del proyecto.

La información deberá estar centralizada, compartida y las herramientas de colaboración serán claves.

Los factores de éxito que se pueden identificar son las herramientas tecnológicas y las metodologías de gestión de equipos dispersos.

1.2.6.1 Beneficios y dificultades con el uso de la tecnología en los equipos distribuidos

Existen múltiples beneficios que la tecnología puede aportar en los equipos distribuidos, así como también algunas dificultades que superar (ver Tabla 1).

Tabla 1. Beneficios y dificultades del uso de la tecnología en equipos distribuidos

Beneficio	Dificultad
Fácil acceso a la información del proyecto	Las diferencias de conocimiento en el uso de la tecnología dentro del equipo pueden ocasionar dificultades
Facilita la comunicación	Se requiere de tiempo y esfuerzo para incluir adecuadamente el componente tecnológico en un proyecto
Reduce costos del proyecto como por ejemplo los de transporte	Todo proyecto y las personas que lo conforman son diferentes por lo que el éxito de una tecnología en un proyecto no necesariamente es igual en otro
Permite compartir información, experiencias, opiniones y formar grupos afines	Las dinámicas del equipo pueden variar en el tiempo y la tecnología puede no adaptarse a la estrategia
Facilita la planificación y coordinación de los proyectos	Se requiere tiempo para dominar las herramientas tecnológicas a utilizarse

Tomado de: Autores de la Tesis

Como se puede observar son grandes los beneficios del uso de la tecnología pero es importante no perder de vista las posibles dificultades que pueden presentarse para resolverlas de la manera adecuada y al justo tiempo.

1.2.6.2 Factores de éxito en la adopción de tecnología

En la actualidad existe gran variedad de tecnología que puede utilizarse y su utilización cada vez es más sencilla. Existen opciones para reducir la necesidad de movilidad de las personas, de manera que estas se sientan más cercanas y aumentando su capacidad de comunicación ya sea en tiempo real o en modo asincrónico facilitando temas como la disponibilidad de las personas. Muchas

veces la tecnología más avanzada no es la más adecuada, es mejor buscar opciones que destaquen por ser sencillas de utilizar.

Como se menciona en el libro blanco del Teletrabajo (Foro de empresas efr, 2012) existen diferentes roles a considerar en la adopción de tecnología, los cuales que se describen a continuación.

1.2.6.2.1 La alta dirección

- Integrar tecnología para aumentar la productividad
- Establecer una comunicación bidireccional que sea transversal a la empresa entre la Dirección, los directivos y empleados. Esta comunicación debe estar representada en el plan de Comunicación de la compañía y debe estar acompañada de la capacitación necesaria para que todos estén en capacidad de hacer uso de ella
- Establecer una relación de confianza con el departamento de Tecnologías de la Información (TI) que permita establecer una clara hoja de ruta
- Fomentar una cultura tecnológica en los empleados
- Poner a disposición de toda la organización soporte tecnológico para que las incidencias sean atendidas adecuadamente en tiempos adecuados y la experiencia sea agradable para los usuarios.

1.2.6.2.2 Supervisores

- Transmitir a sus empleados la idea de que la tecnología permite hacer el trabajo más productivo y fomenta un mejor modelo de vida
- Comunicar a todos los empleados las capacidades tecnológicas y de soporte de la empresa
- Recoger los comentarios y sugerencias del día a día para trasladárselo al departamento de TI
- Liderar con ejemplo integrando a todos los empleados ya sea que estos estén presenciales o desempeñen trabajo virtual.

1.2.6.2.3 Empleado

- Aprender las nuevas tecnologías para ser más productivo
- Tomar las decisiones adecuadas sobre el correcto uso de la tecnología
- Colaborar con los demás en el uso de la tecnología
- Proveer de comentarios y sugerencias que ayuden a mejorar a la tecnología existente
- Hacer un buen uso del soporte tecnológico de la empresa

1.3 Metodologías de desarrollo de software

Las metodologías tienen su importancia ya que es a través de su correcta aplicación que se consiguen resultados satisfactorios en un proyecto, una metodología nos ayuda a realizar un adecuado trabajo en equipo de manera que cada integrante tenga claro cuáles son sus tareas y como estas se integran con el resto del equipo. Las metodologías de desarrollo de software son un conjunto ordenado de procedimientos y técnicas que nos guían paso a paso entre todas las actividades necesarias para conseguir el producto de software deseado.

Si bien el uso de una metodología de desarrollo de software no asegura el éxito de un proyecto de desarrollo de software su correcta aplicación puede incidir en gran manera para aumentar la probabilidad de éxito.

1.3.1 Metodologías formales

Cuando se inició con el desarrollo de software se podría catalogar que era artesanal en su totalidad, había que mejorar el proceso y llevar los proyectos a un fin deseado. Tuvo que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software. Esta nueva etapa contenía el desarrollo dividido en etapas de manera secuencial para mejorar el desarrollo.

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.). Entre las metodologías tradicionales podemos nombrar a las siguientes:

- RUP – Rational Unified Process
- MSF – Microsoft Solution Framework
- Modelo en cascada
- Modelo en espiral
- Modelo basado en prototipos

1.3.1.1 Rational Unified Process (RUP)

Es un proceso formal. Se basa en un acercamiento disciplinado para asignar tareas y responsabilidades. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. (Customización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

Fases

Las cuatro fases del ciclo de vida son:

- Concepción
- Elaboración
- Construcción
- Transición

Ventajas

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

- Seguimiento detallado en cada una de las fases.

Desventajas

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Se coloca al cliente en una situación que puede ser muy incómoda para él.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

1.3.1.2 Microsoft Solution Framework (MSF)(Microsoft)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Se compone de cinco fases principales:

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.

MSF se centra en:

- Alinear objetivos empresariales y tecnológicos
- Establecer objetivos, roles y responsabilidades claros para el proyecto
- Implementar un proceso iterativo, basado en hitos/puntos de control
- Administrar riesgos de forma proactiva
- Respuestas efectivas a los cambios

1.3.1.3 Modelo en cascada

Este modelo utiliza tramos como puntos de transición y de carga. Al usar el modelo de cascada, se necesitaría completar un conjunto de tareas en forma de fase para después continuar con la fase próxima. El modelo en cascada trabaja perfectamente para los proyectos en los cuales los requisitos del proyecto se encuentran definidos claramente y no son obligados a futuras modificaciones. Ya que este modelo está compuesto por puntos de transición entre fases, se puede monitorear fácilmente ya que asigna responsabilidades definidas.

Las fases que considera son:

- Requerimientos
- Diseño
- Codificación y prueba unitaria
- Integración del sistema
- Operación y mantenimiento

Ventajas

- Modelo y planificación fácil y sencillos.
- Sus fases son conocidas por los desarrolladores.
- Los usuarios lo pueden comprender fácilmente.

Desventajas

- Los proyectos raras veces siguen una evolución secuencial
- No todos los requisitos son expuestos al principio de forma explícita como requiere este modelo.
- El cliente debe tener paciencia, ya que la aplicación solo estará disponible en un estado muy avanzado del proyecto.

1.3.1.4 Modelo en espiral

Este modelo se basa en la necesidad continua de refinar los requerimientos para un determinado proyecto. El modelo espiral es eficaz cuando se utiliza para el rápido desarrollo de proyectos muy pequeños. Esta logra consigo el acercamiento entre el equipo de desarrollo y el cliente porque el cliente es implicado en todas las etapas proporcionando la regeneración de proyecto y la aprobación del mismo. De cualquier forma, el modelo en espiral no incorpora puntos de comprobación claros. Por lo tanto, el proceso de desarrollo puede llegar a ser caótico.

En este modelo se considera las siguientes etapas:

- Planificación del proyecto
- Análisis de riesgos
- Toma de decisiones
- Construcción y adaptación
- Evaluación del cliente
- Comunicación con el cliente

Ventajas

- Puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- Es un enfoque realista del desarrollo de sistemas y de software a gran escala.
- Como el software evoluciona, a medida que progresa el proceso el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- Utiliza la construcción de prototipos como mecanismo de reducción de riesgos.
- Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

- Mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora al marco de trabajo iterativo que refleja de forma más realista el mundo real.
- Demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Desventajas

- Requiere comunicación permanente con el cliente
- El cliente debe ser gran conocedor del sistema.
- Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable.
- Requiere una considerable habilidad para la evaluación del riesgo.
- No se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos

1.3.1.5 Modelo basado en prototipos

Los prototipos no poseen la funcionalidad total del sistema pero si tiene la idea total del proyecto. Hay un alto grado de participación del usuario en cada paso del desarrollo.

Considera las siguientes etapas:

- Obtención de requisitos
- Diseño global
- Construcción de prototipo
- Desarrollo de Prototipo
- Refinamiento Prototipo
- Sistema Terminado

Ventajas

- La clave del éxito es definir bien desde el inicio, las reglas del juego.
- Alto grado de participación del usuario.

Desventajas

- El cliente puede pensar que el prototipo es una versión acabada
- Podría llegar a pasarse por alto la calidad del software o el mantenimiento a largo plazo.
- Las herramientas elegidas pueden ser inadecuadas.

1.3.2 Metodologías ágiles

Las metodologías tradicionales tienen una rigurosa y detallada definición de los roles, artefactos y actividades, que se deben realizar y utilizar durante el modelado y la documentación. Estas metodologías se las consideran muy adecuadas para proyectos de desarrollo grandes en tiempo y recursos y con condiciones estables; pero, en la actualidad es fácilmente observable como el entorno y las condiciones del proyecto están en constante cambio y adaptación y constantemente se necesita reducir los tiempos de desarrollo sin sacrificar su calidad.

Las metodologías ágiles aparecen como una alternativa para solventar estas necesidades de cambio constante y las restricciones de tiempo y aunque estaban principalmente orientadas para proyectos pequeños, hoy en día podemos verlas exitosamente utilizadas en proyectos de mayor duración y tamaño.

Sirven principalmente en el desarrollo de productos de software donde es difícil desde un inicio definir claramente los requerimientos, o en proyectos donde lo que se busca es una constante retroalimentación del cliente para conseguir un producto de calidad o en casos donde los requerimientos son muy variables en el tiempo.

En la actualidad podemos hablar de metodologías ágiles como:

- AD - Agile Database Techniques
- AM - Agile Modeling
- ASD - Adaptive Software Development
- AUP - Agile Unified Process
- Crystal
- DSDM - Dynamic Systems Development Method
- FDD - Feature Driven Development
- Lean Software Development
- Scrum
- TDD - Test-Driven Design
- XP - Extreme Programming

De estas algunas se enfocan en la gestión, por lo que serán abordadas en la sección correspondiente a gestión ágil de proyectos de software

Las metodologías ágiles están basadas en lo que se conoce como el Manifiesto Ágil

1.3.2.1 Manifiesto Ágil

El manifiesto está compuesto de 4 valores y basados en estos se redactaron 12 principios (Beck, y otros, 2001)

1.3.2.1.1 Valores

1) Individuos e interacciones sobre procesos y herramientas

Las personas son lo más importante; y aunque, los procesos son importantes es mejor formar un buen equipo, dándole importancia a las personas y sus talentos y dejar que estos construyan su entorno basado en sus necesidades. Los procesos deben ser una ayuda, una guía que debe adaptarse a las personas, a la organización y no a la inversa. Pensar lo contrario es esperar buenos resultados con personas

mediocres que siguen procesos, lo cual no es posible cuando las actividades a realizar necesitan creatividad, conocimiento e innovación

2) Software funcionando sobre documentación extensiva

Si bien un documento puede ser importante y en ocasiones necesario por cuestiones legales o normativas, estos no deben ser cortos, solo con el contenido necesario y únicamente cuando son necesarios. Mediante su uso pueden transmitir conocimiento pero nunca pueden ser más importantes que la comunicación directa entre los miembros de un equipo y nunca podrán decir más que un prototipo o una parte del software ya terminada, donde se pueden validar y refinar los requisitos que muchas veces no se los tiene en cuenta desde el primer momento

3) Colaboración con el cliente sobre negociación contractual

Para el desarrollo ágil el cliente es parte del equipo y el software resultante es el producto de una relación constante del mismo con el resto del equipo de desarrollo. El éxito del proyecto se da por llevar exitosamente esta relación y no por un contrato, que en muchas ocasiones puede provocar división entre estas partes al tratar de separar sus responsabilidades

4) Respuesta ante el cambio sobre seguir un plan

Partiendo de la idea de que un proyecto está sujeto a constantes cambios (requisitos, miembros del equipo, tecnologías, etc.) se torna mucho más importante la anticipación y tener capacidad de respuesta y adaptación a estos eventos que seguir rígidamente un plan.

1.3.2.1.2 Principios

- 1) La prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- 2) Se acepta que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- 3) Se entrega software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

- 4) Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
- 5) Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- 6) El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- 7) El software funcionando es la medida principal de progreso.
- 8) Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- 9) La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- 10) La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- 11) Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- 12) A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

1.3.2.2 Lean Development (LD)

Tiene su origen en el modelo de gestión que se denomina Manufactura Ajustada (Lean manufacturing), el cual es un modelo de gestión enfocado en crear el flujo que permita entregar el máximo valor a los clientes con el mínimo uso de recursos; la cual se originó en el sistema de producción de Toyota.

Fue definida por Bob Charette's basándose principalmente en su experiencia en la industria automotriz y luego adaptado por Mary y Tom Poppendieck (Poppendieck & Poppendieck, 2006)

Esta metodología se basa en 7 principios que son:

1) Eliminar los desperdicios

Lo que no añade valor al cliente debe considerarse un desperdicio

2) Ampliar el aprendizaje

El desarrollo de software es un proceso de aprendizaje continuo

3) Decidir lo más tarde posible

Debido a la incertidumbre inherente a los proyectos de software es mejor retrasar toda decisión tanto como sea posible para que estas estén basadas en hechos y no suposiciones

4) Reaccionar tan rápido como sea posible

Si no se tiene velocidad de reacción las decisiones no pueden ser postergadas. Todo cliente valora una entrega rápida de un producto de buena calidad

5) Potenciar el equipo

Es una idea errónea considerar a las personas como recursos; el limitarse a asignar a las personas tareas y procurar que estas no cambien hasta ser terminadas no asegura su correcta ejecución; las personas necesitan motivación, saber que su trabajo persigue un propósito superior. Como dijo alguna vez Steve Job: "No contrato gente inteligente para decirle qué hacer. Contrato gente inteligente para que nos digan qué tenemos que hacer".

6) Crear la integridad

Cada uno de los componentes del sistema debe funcionar bien juntos y proporcionar al cliente una experiencia general del sistema. Esto se consigue principalmente con un flujo constante de comunicación bidireccional entre el cliente y los desarrolladores

7) Véase todo el conjunto

Un producto de software no es solo la suma de sus partes, sino principalmente es el resultado de sus iteraciones. Un desarrollo donde se descompone en pequeñas piezas y por ende en pequeñas tareas tiende a acumular defectos.

Una de las principales características de esta metodología es considerar que si bien los cambios son riesgos para el proyecto, un rápido y adecuado manejo de los mismos puede transformarlos en oportunidades; por lo que, propone un mecanismo para enfrentar estos cambios

1.3.2.3 Extreme Programming (XP)

Esta metodología de desarrollo de software fue desarrollada por Kent Beck. Tiene como objetivo mejorar la calidad del software y la capacidad de respuesta a los cambiantes requerimientos del cliente, procura liberaciones cortas y frecuentes, de modo que puedan establecerse puntos de control más frecuentes en los que pueden adoptarse los nuevos requerimientos del cliente.

Como características tiene la programación en pares, pruebas unitarias de todo el código, revisiones de código, frecuente comunicación entre el cliente y los programadores; buscando siempre la reducción de errores y una mejora continua del código que se está escribiendo.

Su nombre radica en que lleva las buenas prácticas, como la revisión de código, al extremo

La metodología Extreme Programming reconoce algunas actividades en el proceso de desarrollo de software:

1) Codificar

El producto del proceso de desarrollo es el código, el cual debe ser claro y conciso

2) Probar

Mientras más pruebas se realicen, más defectos serán eliminados

3) Escuchar

Es importante escuchar a los clientes, una adecuada comunicación entre el cliente y el programador produce un mejor producto

4) Diseñar

El diseño es importante para evitar que el sistema se torne demasiado complejo y evitará dependencias innecesarias, lo cual es importante ya que evitará que los cambios en ciertas partes afecten a otras

Además XP describe cinco valores:

1) Comunicación

La comunicación cliente y programador es importante para entender correctamente los requerimientos del sistema. Una adecuada comunicación entre los miembros del equipo ayuda a que se comparta mejor el conocimiento y que todos compartan una sola visión de lo que se está construyendo

2) Simplicidad

Extreme Programming incentiva a encontrar la solución más sencilla buscando siempre las soluciones para los problemas de hoy, procurando no caer en buscar soluciones para problemas que quizá no ocurran tiempo después.

3) Retroalimentación

La retroalimentación se la representa de diversas maneras, como lo son: pruebas unitarias que permiten retroalimentarse de posibles errores de programación, los comentarios del cliente que ayudan a entender el estado real del sistema y retroalimentación entre los integrantes del equipo que ayuda a buscar las mejores soluciones a los requerimientos del cliente

4) Coraje (valentía)

Para seguir algunas premisas de esta metodología se necesita coraje. Por ejemplo, cuando se habla de solo resolver los problemas de hoy y no los de mañana, valentía para refactorizar el código tantas veces como sea necesario siempre buscando su mejoramiento y sencillez aunque esto implique eliminar código que pudo representar un gran esfuerzo y persistir siempre hasta encontrar las soluciones a los problemas

5) Respeto

Respetar a los demás y a uno mismo, lo cual significa por ejemplo siempre buscar la calidad en lo que se hace, nunca subir cambios que provoquen que el proyecto no compile, esforzarse por formar un equipo de trabajo unido en el que nadie se sienta menos que otro.

Los roles propuestos por Beck son:

- **Programador**
Crea el código y las pruebas unitarias del sistema
- **Cliente**
Escribe las historias de usuario, las pruebas funcionales y prioriza las historias de usuario para decidir cuáles se implementan en cada iteración buscando dar el mayor valor al negocio
- **Tester**
Ayuda al cliente a escribir las pruebas funcionales y las ejecuta periódicamente
- **Tracker**
Realiza el seguimiento del progreso en cada iteración y retroalimenta al equipo con la finalidad de lograr mejores estimaciones
- **Coach**
Es responsable de que se siga el proceso, quien proporciona indicaciones al equipo
- **Consultor**
Es un miembro externo con un conocimiento especializado y específico en algún tema necesario para el proyecto
- **Gestor (*Big boss*)**
Se encarga de la coordinación entre el cliente y los programadores, actúa como facilitador

El proceso puede resumirse en los siguientes pasos:

- 1) El cliente define lo que se debe implementar
- 2) El programador estima el esfuerzo necesario para implementar

- 3) El cliente escoge lo que debe construirse en la iteración según sus prioridades y la restricción de tiempo
- 4) El programador construye
- 5) Se regresa al paso 1

Siempre hay una retroalimentación del equipo para buscar mejora continua, se trata de no sobrecargar al programador para que logre un trabajo de calidad y se cumplan los plazos y el cliente siempre debe buscar que el producto entregue el mayor valor posible al negocio.

1.3.2.4 Feature Driven Development (FDD)

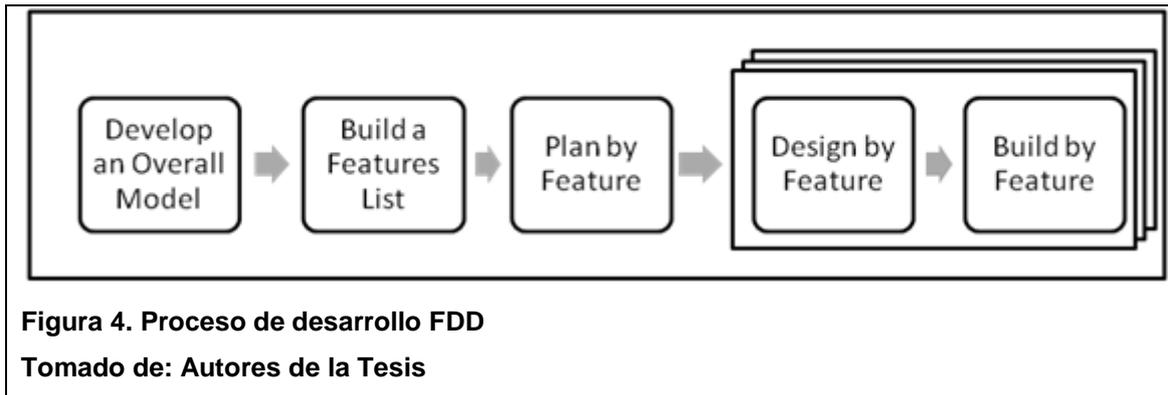
Fue desarrollada por Jeff De Luca y Peter Coad a mediados de los años 90. Se basa en iteraciones cortas para obtener entregables del producto en periodos de dos semanas. Esta metodología se enfoca en las fases de diseño y construcción más que en la obtención de los requerimientos.

Entre sus ventajas podemos decir que se evita el uso innecesario de recursos y mejora la calidad del producto por la retroalimentación de sus constantes entregas. Sin embargo tiene algunas debilidades como la falta de documentación y pueden darse problemas de comunicación ya que puede tener ambigüedades

1.3.2.4.1 Proceso

FDD tiene cinco procesos secuenciales que son:

- 1) Desarrollar un Modelo
- 2) Construir una Lista de funcionalidades
- 3) Planear por funcionalidad
- 4) Diseñar por funcionalidad
- 5) Construir por funcionalidad



Diseño y construcción se las realiza de forma iterativa con desarrollo ágil

1.3.2.4.2 Roles

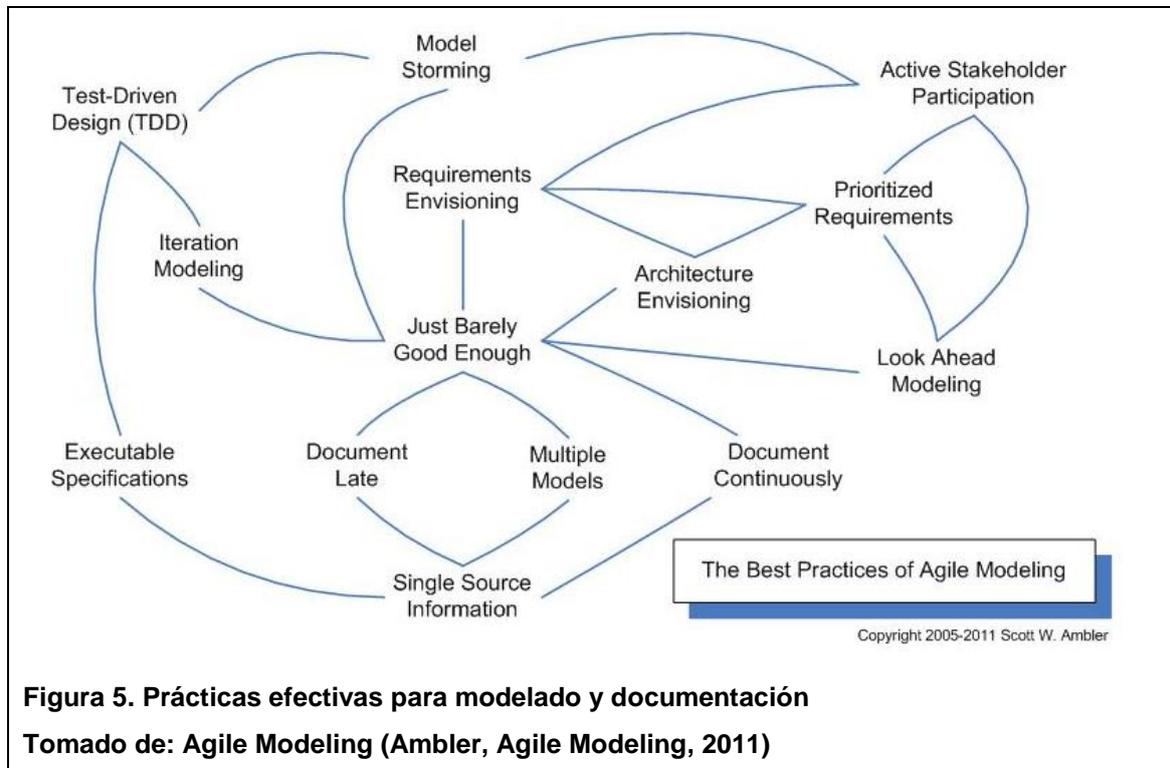
Los equipos se estructuran en jerarquía con los siguientes roles:

- **Arquitecto jefe**
Encargado del Diseño global del sistema
- **Director de desarrollo**
Lleva diariamente las actividades de desarrollo, obtiene los recursos para el proyecto y resuelve conflictos en el equipo
- **Programador Jefe**
Realiza análisis, diseño y selecciona las funcionalidades a desarrollar
- **Propietario de clases**
Es el responsable del desarrollo de las clases asignadas
- **Expertos de dominio**
Es el usuario o la persona que se encarga de definir los requerimientos del sistema

1.3.2.5 Agile Modeling (AM)

Es una metodología basada en la práctica para modelado efectivo de sistemas de software. La metodología AM es una colección de prácticas, guiadas por principios y valores que pueden ser aplicados por profesionales de software en el día a día. AM no es un proceso prescriptivo, ni define procedimientos

detallados de cómo crear un tipo de modelo dado. (Ambler, Agile Modeling, 2002)



Posee las siguientes características:

- 1) Satisface su propósito.
- 2) Es inteligible.
- 3) Es suficientemente preciso.
- 4) Es suficientemente consistente.
- 5) Es suficientemente detallado.
- 6) Aporta valor positivo.
- 7) Es lo más simple posible.

1.3.2.5.1 Que es y no es AM

Las siguientes definiciones son tomadas de Agile Modeling (Ambler, Agile Modeling, 2002)

- 1) AM es una actitud, no un proceso prescriptivo.

- 2) AM es un complemento a los métodos existentes, no es una metodología completa.
- 3) AM es una manera efectiva de trabajar en conjunto para alcanzar las necesidades de las partes interesadas en el proyecto.
- 4) AM es efectivo, y se trata sobre ser efectivo.
- 5) AM es algo que funciona en la práctica, no es una teoría académica.
- 6) AM no es una bala de plata.
- 7) AM es para el desarrollador promedio, más no es un sustituto de la gente competente.
- 8) AM no es un ataque a la documentación, de hecho AM aconseja la creación de documentos que tengan valor.
- 9) AM no es un ataque a las herramientas CASE.
- 10) AM no es para todos.

1.3.2.5.2 Objetivos

Según Scott W. Ambler (Ambler, Agile Modeling, 2002) los objetivos de AM son:

- 1) Definir y mostrar cómo poner en práctica una colección de valores, principios y prácticas que conlleven a un modelado ligero efectivo.
- 2) Explorar la aplicación de técnicas de modelado en proyectos de software a través de un enfoque ágil, tal como XP, DSDM o SCRUM.
- 3) Explorar el cómo mejorar el modelado bajo procesos prescriptivos, tales como el Proceso Unificado de Rational (RUP) o el Proceso Unificado Empresarial (EUP)

1.3.2.6 Agile Unified Process (AUP)

Es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

Fases del ciclo de desarrollo

Incepción: identifica el alcance y dimensión del proyecto, propuesta de la arquitectura y del presupuesto del cliente.

Elaboración: Avala la arquitectura.

Construcción: Desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados.

Transición: Validación e implantación del sistema.

Principios en los que se basa AUP

Simplicidad, Todo se describe concisamente utilizando poca documentación, no miles de ellas.

Agilidad, El ajuste a los valores y principios de La Alianza Ágil.

Centrarse en actividades de alto valor, La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.

Herramienta de la independencia, Se puede usar cualquier conjunto de herramientas que se desee con el AUP. Se sugiere utilizar las herramientas más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

1.4 Trazabilidad en proyectos de desarrollo de software

1.4.1 Definición de trazabilidad

Gotel y Filkestein definen a la trazabilidad de requisitos como: la posibilidad de describir y seguir un requisito hacia adelante y hacia atrás, desde su origen a través del desarrollo hasta el subsecuente empleo, además a través de los ciclos de refinamiento (Gotel & Finkelstein, 1994)

En definitiva la trazabilidad de requisitos permite seguir un requisito durante todo el proceso de desarrollo de software hasta ver cómo fue implementado; o a la inversa, saber a partir de qué requerimiento se desarrolló una

funcionalidad. Todos los documentos, elementos de diseño, código, guiones de pruebas deben poder determinar su punto de origen

1.4.2 Ventajas y desventajas de la trazabilidad de requerimientos

Si bien la trazabilidad de requerimientos permite obtener un sistema lo más cercano a las necesidades expresadas por el cliente, ésta presenta ventajas y desventajas. Entre las principales ventajas del uso de trazabilidad tenemos:

- Determinar si todos los requerimientos fueron implementados adecuadamente y verificar que todas las funcionalidades partieron de un requerimiento
- Ayuda a la gestión de los cambios ya que permite evaluar el impacto en el sistema al efectuar un cambio sobre un requerimiento
- Mejora la comprensión del sistema, sobre todo a los nuevos integrantes del equipo les facilita entender lo que se está construyendo
- Aumento de la calidad del producto final

Sin embargo, el realizar trazabilidad de requerimientos es un proceso bastante exigente que debe adaptarse a las características de cada proyecto, del cual no está claramente definido qué información es la que debe recolectarse y aunque actualmente hay herramientas para realizarlo estas no necesariamente se adaptan completamente a la metodología empleada o no proveen los mecanismos adecuados.

1.5 Gestión de proyectos

1.5.1 Definición

La gestión de proyectos es la disciplina mediante la cual se planifica, organiza y controlan las actividades de otros para alcanzar un objetivo o cumplir con los requerimientos de un proyecto, para lo cual es necesaria la aplicación de conocimientos, habilidades, herramientas y técnicas.

Las actividades que comúnmente forman parte de la gestión son:

- Planificación
Definir un curso de acción
- Organización
Definir las relaciones entre las unidades de trabajo y asignar responsables
- Selección (*Staffing*)
Selección del personal
- Dirección
Creación del ambiente adecuado para motivar a las personas a alcanzar los objetivos
- Control
Medición y evaluación del desempeño y avances de las actividades según los objetivos planteados

Los proyectos por su naturaleza necesitan ser ejecutados bajo ciertas restricciones, las cuales suelen ser representadas en alcance, tiempo y costo (triángulo de administración de proyectos). Cada una de estas restricciones no puede ser modificada sin afectar uno o más de las otras restricciones.

La restricción tiempo hace referencia al tiempo que se dispone para completar el proyecto

La restricción costo se refiere al presupuesto asignado para el cumplimiento del proyecto

La restricción alcance hace referencia a lo que debe tener el resultado final del proyecto para darlo por completo y terminado

Por ejemplo un aumento en el alcance podría significar un aumento en el tiempo y un aumento en el costo del proyecto.

1.5.2 Gestión de proyectos de software

Existen múltiples desafíos en el desarrollo de software de calidad, entre los principales está el lograr un correcto entendimiento de las necesidades del cliente y el poder adaptarse fácilmente a los diversos cambios que pueden surgir en un proyecto (modificaciones en requerimientos, alteraciones en las restricciones del proyecto costo, tiempo, alcance, etc.). La gestión de proyectos de desarrollo software permite encontrar, revisar y evaluar las múltiples opciones de solución a estos desafíos para seleccionar la solución más adecuada que se enmarque en los parámetros de calidad y eficiencia y que permita alcanzar los objetivos establecidos, es decir la gestión de proyectos de software nos permite hacer uso de habilidades, herramientas y técnicas para llevar al proyecto a un final satisfactorio.

Existen diferentes metodologías para gestión de proyectos que han sido propuestas por diferentes organizaciones, tales como: PMI (Project Management Institute), IPMA (International Project Management Association), SEI (Software Engineering Institute), y diversas metodologías como Prince2 (Projects IN Controlled Environment), MSP (Managing Successful Programmes), MoP (Management of Portfolios), P3O (Portfolio, Programme and Project Offices Pocketbook) y M_o_R (Management of Risk) que ofrecen herramientas para realizar una buena gestión.

1.5.3 Gestión de proyectos formales

La gestión formal de proyectos también se la conoce como clásica o tradicional y tiene un enfoque predictivo, el cual se rige bajo ciertas premisas como:

- Los proyectos tienen características similares y el entorno en que se desarrollan es predecible o los riesgos se pueden anticipar y por tanto ser controlables
- Un buen análisis de inicio puede definir el resultado que se quiere obtener, por lo que es importante cuidar los procesos para asegurar que una vez definido el alcance se cumpla la planificación del proyecto en tiempo y costos

En la gestión formal de proyectos si se tiene un plan general basado en la estrategia y objetivos a alcanzar y se disminuye la incertidumbre y riesgos es más probable que el proyecto llegue a un final exitoso. Por ello es importante conocer a un buen nivel de detalle, y desde un inicio, los requerimientos del proyecto. Organizaciones como PMI e IPMA han desarrollado múltiples estándares, métodos y modelos que tienen su enfoque en este concepto; aunque es importante mencionar que en los últimos años han incorporado conceptos de ágil en sus propuestas.

1.5.3.1 Project Management Body of Knowledge (PMBok)

El Project Management Institute (PMI) es una organización internacional sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos. La guía del PMBOK, la cual es de su autoría, contiene los fundamentos de la gestión de proyectos que esta organización reconoce como buenas prácticas.

La quinta edición del PMBOK reconoce 5 grupos de procesos y 10 áreas de conocimiento que son comunes a casi todos los proyectos.

1.5.3.1.1 Grupos básicos de procesos

- 1) Iniciación
- 2) Planificación
- 3) Ejecución
- 4) Seguimiento y Control
- 5) Cierre

1.5.3.1.2 Áreas de conocimiento

- 1) Gestión de la Integración del Proyecto
- 2) Gestión del Alcance del Proyecto
- 3) Gestión del Tiempo del Proyecto
- 4) Gestión de los Costos del Proyecto
- 5) Gestión de la Calidad del Proyecto
- 6) Gestión de los Recursos Humanos del Proyecto

- 7) Gestión de las Comunicaciones del Proyecto
- 8) Gestión de los Riesgos del Proyecto
- 9) Gestión de las Adquisiciones del Proyecto
- 10) Gestión de los Interesados del Proyecto

1.5.3.2 Projects in a Controlled Environment (Prince2)

Prince2 es un método estructurado para gestión de proyectos, se centra en la organización, gestión y el control.

Fue desarrollado originalmente por la Agencia de la Computadora Central y Telecomunicaciones (CCTA), en el Reino Unido como un estándar para la administración de proyectos de tecnología de la comunicación y actualmente es ampliamente usada. Se podría anotar como características que:

- Se basa en buenas prácticas y en conseguir los productos y no en realizar las tareas o actividades.
- Aplicables a proyectos de cualquier tamaño y sector.
- Proveen de un vocabulario común.
- Está orientado a toda la organización que interviene en el proyecto
- Define con más detalle los roles y responsabilidades dentro del proyecto y la organización de gestión y toma de decisiones.
- Se basa en una gestión por excepción, donde la organización de proyecto permite que cada uno sepa lo que tiene que hacer y pueda tomar las decisiones que le corresponden.
- Está orientado a la finalización del Business Case.

1.5.3.3 SEI

Software Engineering Institute (SEI) es un instituto federal estadounidense de investigación y desarrollo en ingeniería de software patrocinado por el Departamento de Defensa de EE.UU. y operado por la Universidad Carnegie Mellon en Pittsburg, Pa. El SEI trabaja en estrecha colaboración con las organizaciones de defensa y del gobierno, la industria y el mundo académico para mejorar continuamente los sistemas de software. Su objetivo principal es

ayudar a las organizaciones a mejorar sus capacidades de ingeniería de desarrollo de software o en la adquisición del software adecuado, libre de defectos, dentro del presupuesto y a tiempo. (SEI, 2014)

El SEI se convirtió en referente en Ingeniería de Software al haber desarrollado el modelo SW-CMM, el cual ha sido el punto de partida como concepto de capacidad y madurez, hasta llegar al actual CMMI (Capability Marurity Model Integration)

1.5.4 Gestión de proyectos ágiles

La gestión de proyectos ágil es informal y busca únicamente cubrir el ciclo sobre el cual se está trabajando. Se puede incluso ver en algunas de éstas referencias a evitar el uso de diagramas como el Gantt. Entre las metodologías que se consideran ágiles podemos nombrar, entre otras, a: RUP (Rational Unified Process), MSF (Microsoft Solutions Framework) y Scrum. RUP y MSF se consideran metodologías ágiles debido a que su ciclo de vida es iterativo e incremental, incluso teóricamente podrían respetar el manifiesto ágil; pero, su alto énfasis en el proceso y su gran número de artefactos hace que su respuesta al cambio no sea verdaderamente ágil. (ver Anexo 4)

Una gestión ágil de proyectos busca garantizar las siguientes necesidades de un proyecto:

- Ayudar a dar el mayor valor posible a un producto que está en desarrollo en entornos de constante cambio
- Reducir el tiempo de desarrollo para mejorar la ventaja competitiva
- Agilidad para adaptarse a cambios en los requisitos o a circunstancias del entorno
- Fiabilidad en los resultados, es decir cumplimiento de los plazos, en los costes estimados y que estos resultados sean repetibles en los diferentes proyectos

Las metodologías ágiles antes mencionadas (XP, FDD, etc.) cubren puntos específicos del desarrollo de software, pero también existen otras metodologías

que se centran en la gestión de proyectos. A continuación se mencionan las principales metodologías ágiles que se enfocan en la gestión de proyectos

1.5.4.1 Scrum

Scrum fue desarrollado por Ken Schwaber y Jeff Sutherland y se lo define como un marco de trabajo para la gestión y desarrollo de software de forma iterativa e incremental, está enfocado en la gestión de procesos de desarrollo de software de forma empírica. Es empírica ya que define que todo conocimiento viene de la experiencia y las decisiones se toman en base a lo que se conoce. Su enfoque iterativo e incremental ayuda a minimizar el riesgo de manera que los resultados sean más predecibles.

Hoy en día es la principal metodología para gestión de proyectos ágiles. Su objetivo es mejorar la productividad de un equipo reduciendo al máximo las actividades no relacionadas a la producción de software, buscando obtener resultados en cortos periodos de tiempo (máximo 30 días). Se enfoca en las prácticas de gestión sin entrar en las prácticas de desarrollo (puede usarse XP por ejemplo), delegando la responsabilidad de decidir la mejor forma de trabajar al equipo, basándose en teorías de auto-organización.

Scrum se define como un marco de trabajo de procesos para gestionar el desarrollo de productos complejos. No es un proceso o una técnica, es un marco de trabajo dentro del cual se emplean varias técnicas y procesos.

“El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos” (Schwaber & Sutherland, 2013)

Scrum es iterativo e incremental. Inicia con reuniones con los stakeholders para la elaboración de Product Backlog, el cual consiste en un listado de todas las funcionalidades que se desea formen parte del sistema a construirse y estarán

especificados de la manera que se considere conveniente (casos de uso, diagramas de flujo, etc.). A partir de esto inician las iteraciones o Sprints, donde cada una de estas iteraciones tendrá su propio Sprint Backlog, el cual es un subconjunto del product Backlog, es decir es un listado de los requerimientos que se construirán durante el Sprint. La recomendación es que un sprint no dure más de un mes.

Dentro del Sprint, diariamente se realizará el Scrum Daily Meeting, la cual es una reunión de revisión de avance diaria que no debe durar más de 15 minutos y que sirve para que los integrantes del equipo y el Scrum Master (de cierta manera es equivalente a un líder de proyecto) conozcan las tareas en curso, los recursos necesarios y los obstáculos que tengan que ser solventados.

Al final de cada Sprint se realiza el Sprint Review que sirve para revisar los artefactos construidos, adaptar el Product Backlog si es necesario y se determinan acciones para optimizar el valor. Esta reunión ayuda a obtener retroalimentación de la información y a fomentar la colaboración.

De esta manera se obtiene una gran cantidad de retroalimentación del proceso de desarrollo que permite corregir problemas y mitigar riesgos tempranamente.

Scrum no propone el uso de ninguna metodología de desarrollo, sin embargo suele utilizárselo como un framework ágil para administración de proyectos y se lo complementa con cualquiera de las metodologías de desarrollo antes mencionadas.

1.5.4.1.1 Pilares de Scrum

El control de procesos empírico de Scrum es soportado por tres pilares(Schwaber & Sutherland, 2013):

1) Transparencia

Los aspectos significativos del proceso deben ser visibles para las personas responsables de los resultados.

La transparencia exige que los integrantes del equipo hablen un lenguaje común cuando se refieran al proceso, que los aspectos deben

estar definidos en un solo estándar y tanto los que desarrollan como los que aceptan el producto deben entender lo mismo como trabajo terminado.

2) Inspección

Los artefactos de Scrum deben ser continuamente inspeccionados para verificar su avance al objetivo o para detectar desviaciones, siempre procurando que estas inspecciones no sean tan frecuentes que interrumpen el trabajo.

3) Adaptación

Si durante la inspección se determina que un artefacto se ha desviado más allá de límites aceptables y por lo tanto el producto resultante no estará en límites aceptables, el proceso deberá ser reajustado lo antes posible.

Cada Sprint define cuatro eventos donde se realiza la inspección y adaptación y estos son:

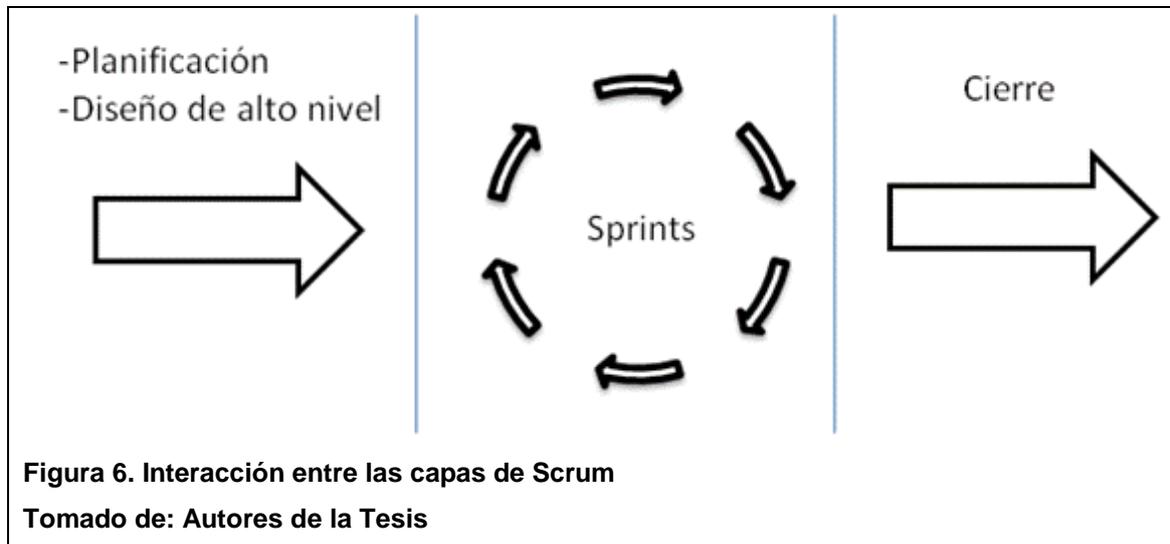
- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

1.5.4.1.2 Proceso

El proceso de Scrum se divide en tres etapas:

- Pre-Juego
Se definen las funcionalidades que son parte del Product Backlog y de estas se escoge las que forman parte del Sprint Backlog del siguiente Sprint. También se diseña una arquitectura de alto nivel.
- Juego
Se desarrollan las tareas del Sprint Backlog
- Post-juego
Se evalúan los resultados del Sprint en cumplimiento de tiempos, avances y de la implementación. Si es una iteración del proyecto se

regresa al Pre-Juego, caso contrario se realiza la documentación para finalizar el proyecto.



1.5.4.1.3 Equipo Scrum (Scrum Team)

Los equipos en Scrum son multifuncionales y se auto organizan, es decir que buscan la mejor forma de llevar a cabo su trabajo y no requieren ser dirigidos por personas externas, por lo que es importante que tengan las competencias necesarias para llevar a cabo el trabajo. Scrum busca incentivar la creatividad, optimizar la flexibilidad y mejorar la productividad de los equipos.

Los roles definidos en Scrum son:

1.5.4.1.3.1 Dueño de Producto (Product Owner)

Es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. Es responsable de gestionar la Lista del Producto (Product Backlog).

Son sus funciones:

- Mantener la Lista del Producto clara, entendible y accesible para todos los miembros del equipo
- La Lista del Producto debe estar ordenada según los objetivos

- Asegurarse que los miembros del equipo conocen y entienden la lista de producto

El dueño del producto es una sola persona y si alguien desea cambiar la prioridad deberá pedírselo, ya que todos deben respetar sus decisiones. Nadie del equipo debe actuar en base a requerimientos de otra persona.

Si bien los miembros de un equipo de desarrollo deben tener habilidades y conocimiento especializado para tareas específicas que realizan, la responsabilidad de cumplir las tareas y objetivos es de todo el equipo

1.5.4.1.3.2 Equipo de Desarrollo (Development Team)

Son todas las personas del equipo encargadas de construir el producto resultante de cada Sprint.

Los equipos deben actuar con autonomía y deben ser empoderados para decidir cómo debe realizarse su propio trabajo, de esta manera se logra una mejor sinergia que produce mejores resultados.

Los equipos deben ser pequeños para poder ser ágiles, pero deben tener los miembros suficientes que permitan completar el trabajo. Menos de tres miembros puede dificultar el trabajo debido a una posible falta de conocimiento o habilidades en los miembros que sean necesarias en un Sprint; en cambio, un equipo de más de nueve miembros puede tener problemas de coordinación. En el tamaño del equipo no se cuenta ni al Dueño del Producto ni al Scrum Master.

1.5.4.1.3.3 Scrum Master

Es el líder del equipo que actúa como facilitador ayudando a mantener al equipo dentro de las prácticas de Scrum, facilitando los medios para los eventos de Scrum, eliminando cualquier impedimento para que el equipo pueda avanzar en desarrollo del producto y actuando como un vínculo con las personas externas al equipo y como un filtro para evitar interrupciones.

El Scrum Master interactúa con el Dueño del Producto con la finalidad de entender la planificación del producto y ayudar al Dueño del producto con el conocimiento necesario para ordenar la Lista de Producto de modo que maximice su valor

1.5.4.1.4 Eventos del Scrum

- **Sprint**

Es un plazo de tiempo de un mes o menos durante el cual se construye un producto utilizable. Los Sprints se suceden uno a continuación del otro y se procura que duren el mismo tiempo. No deben hacerse cambios que afecten el objetivo del Sprint (Sprint Goal)

La idea de tener periodos cortos de desarrollo permite ser más adaptables a los cambios ya que se es capaz más prontamente de adaptarse a los cambios y en ciertos casos el Dueño del Producto (y solo él) puede cancelar un Sprint, cuando el objetivo haya dejado de ser necesario por cambios de la empresa

- **Reunión de Planificación de Sprint (Sprint Planning Meeting)**

Durante esta reunión, de la cual participan todos los miembros del equipo, se planifica qué se realizará durante el Sprint.

Debe durar máximo 8 horas y es el Scrum Master el encargado de hacer que este tiempo se cumpla y la reunión cumpla sus objetivos, que son: definir cuál es el entregable del Sprint y que hay que hacer para conseguirlo

- **Objetivo del Sprint (Sprint Goal)**

Es la meta a alcanzar con la realización del Sprint y que está basada en la construcción de lo que consta en la Lista del Producto. Se la define durante la reunión de Planificación del Sprint. Su importancia radica en que si todo el equipo tiene claro el Objetivo del Sprint podrán coordinar y trabajar juntos de mejor manera

- **Scrum Diario (Daily Scrum)**

Es una reunión del equipo que no debe durar más de 15 minutos y permite que las tareas se coordinen de la manera más adecuada, solo

los miembros del equipo pueden participar de la reunión. Se basa en la contestación de 3 preguntas:

- Que hice ayer para ayudar a conseguir el objetivo del Sprint
- Que haré hoy para ayudar a conseguir el objetivo del Sprint
- Tengo algún impedimento que impide alcanzar los objetivos

De esta manera se puede evaluar el avance que se va realizando, el equipo se integra, se eliminan impedimentos, se toman decisiones más rápidamente y se coordina de mejor manera y se pueden detectar temas que sean necesarios de detallarlos luego de la reunión Revisión de Sprint (Sprint Review)

- **Revisión de Sprint (Sprint Review)**

Al finalizar un Sprint es necesario realizar una reunión donde se revisan los avances para determinar si es necesario realizar modificaciones o mejoras a la Lista de Producto. Es una reunión informal y no de seguimiento que no debe durar más de cuatro horas

- **Retrospectiva de Sprint (Sprint Retrospective)**

En esta reunión el equipo hace una revisión de las personas, sus relaciones, las herramientas y los procesos del Sprint en busca de oportunidades de mejora para el próximo Sprint. Esta reunión no debe durar más de tres horas

1.5.4.1.5 Artefactos

- **Lista de Producto (Product Backlog)**

Es una lista ordenada de todo lo que podría conformar el producto final (características, requisitos, funcionalidades, mejoras) y se convierte en el único documento de requisitos. Esta lista no se espera que este completa, ya que durante el proyecto se irán detectando nuevos elementos o cambios que son necesarios de incorporarse

- **Lista de Pendientes del Sprint (Sprint Backlog)**

Es un plan que se forma de un subconjunto de la Lista de Producto que define los objetivos de un Sprint. Esta lista debe siempre estar visible al

equipo ya que les permite tener claro que forma parte del Sprint en curso. Se debe procurar tener un nivel de detalle lo suficientemente bueno como para poder determinar los avances en el Scrum diario

1.5.4.2 Método Crystal

Crystal es un conjunto de metodologías para desarrollo de software que se basan en las personas, la comunicación y en reducir al máximo el número de artefactos producidos. Fueron desarrolladas por Alistair Cockburn. El nombre Crystal es debido a la clasificación de los proyectos según 2 dimensiones, tamaño y complejidad.

Al igual que otras metodologías ágiles Crystal propone iteraciones cortas con retroalimentación constante por parte del cliente, evitando de esta manera la necesidad de desarrollar productos intermedios. Por ello es necesario contar con un usuario que tenga tiempo para realizar validaciones del sistema y participe de la definición de los requerimientos

Debido a sus pilares se aconseja que el tamaño del equipo sea reducido y se busque una adecuada comunicación, por lo que es preferible que todos estén en el mismo lugar.

Crystal clasifica a los proyectos por colores según el número de integrantes del equipo: Crystal Clear para equipos de 8 o menos integrantes; Amarillo, para equipos de 8 a 20; Naranja, para 20 a 50; Rojo, para 50 a 100 y le siguen Marrón, Azul y Violeta. Crystal Clear (CC) es el método mejor desarrollado y documentado



1.5.4.2.1 Valores

Los siete valores de Crystal Clear son

- Entrega frecuente
- Comunicación osmótica
- Mejora reflexiva
- Seguridad personal
- Foco
- Fácil acceso a usuarios expertos

1.5.4.2.2 Roles

Los roles establecidos son:

- Patrocinador Ejecutivo (Executive Sponsor)
- Jefe de Proyecto (Project Manager)
- Experto en el Dominio (Domain Expert)
- Experto de uso (Usage Expert)
- Programador Diseñador (Designer-Programmer)
- Diseñador UI (UI Designer)
- Realizador de Pruebas (Tester)
- Programador Técnico (Technical)

1.5.4.3 Dynamic Systems Development Method (DSDM)

El método de desarrollo de sistemas dinámicos es un método cuya filosofía dice que todo proyecto debe tener claramente definidos los objetivos estratégicos y enfocarse en realizar entregas tempranas y periódicas de beneficios al negocio.

DSDM fue creado por un consorcio de proveedores y expertos de Reino Unido en los años 90 al combinar las mejores prácticas encontradas. La primera versión se presentó en 1995, la versión actual en uso es la del 2007 que se la denomina Atern(DSDM CONSORTIUM)

La versión anterior a Atern era la DSDM 4.2 y en esta constaba una guía de cómo usar DSDM conjuntamente con Extreme Programming. Otras metodologías comúnmente usadas en conjunto de DSDM es PRINCE2

1.5.4.3.1 Fases y Etapas de DSDM

Las fases y etapas definidas son:

- Pre-proyecto
- Ciclo de vida del proyecto
 - Estudio de viabilidad
 - Estudio de la empresa
 - Iteración del modelo funcional
 - Diseño e iteración de la estructura
 - Implementación
- Post-proyecto

1.5.4.3.2 Principios

Atern tiene los siguientes principios

- Centrarse en las necesidades del negocio
- Entregar a tiempo
- Colaborar
- Nunca comprometer la calidad
- Construir de forma incremental desde bases sólidas
- Desarrollar iterativamente
- Comunicar de forma clara y continua
- Demostrar control

1.5.4.3.3 Prerrequisitos

Hay 9 factores que deben cumplirse para asegurar el éxito de un proyecto con DSDM, de lo contrario se convierten en riesgos que deben ser administrados.

- 1) La aceptación de la filosofía Atern antes de comenzar el trabajo.
- 2) Empoderamiento adecuado del equipo de desarrollo de soluciones.

- 3) Compromiso por parte de la gestión de negocios de alto nivel para proporcionar las personas que los representen y que suministren la información necesaria que ayude a obtener un entendimiento detallado de los requerimientos.
- 4) Entrega incremental
- 5) El acceso por el Equipo de Desarrollo de Soluciones para las funciones de negocio
- 6) Estabilidad del equipo de desarrollo de soluciones.
- 7) Las habilidades necesarias en el equipo de desarrollo de soluciones.
- 8) Tamaño adecuado del equipo de desarrollo de soluciones, se sugiere que sea de siete +/- dos personas
- 9) Una relación comercial favorable entre el cliente y el proveedor.

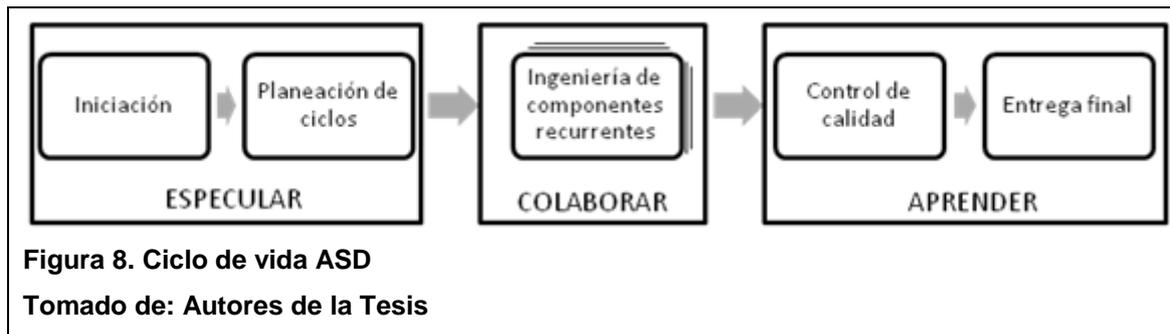
1.5.4.4 Adaptive Software Development (ASD)

Esta metodología se basa en el principio de la adaptación continua, este proceso de adaptación se lo considera normal en el proyecto y no se debe luchar contra él.

Es una metodología iterativa, se orienta a los componentes de software y no a las tareas y es muy tolerante a los cambios

1.5.4.4.1 Ciclo de vida

El ciclo de vida de ASD es iterativo y no es lineal, es decir que un ciclo puede iterar y sufrir modificaciones al mismo tiempo que otro está siendo ejecutado. Su ciclo dinámico se lo denomina Especular-Colaborar-Aprender y requiere de constante comunicación y colaboración entre desarrolladores y clientes para solventar los constantes cambios.



2 Capítulo II: Desarrollo del Marco de Trabajo

2.1 Por qué un marco de trabajo para gestión de proyectos de software con equipos geográficamente dispersos.

Los modelos de gestión tradicional o ágil antes descritos son marcos de referencia para gestionar proyectos, algunos de forma general, otros específicos para proyectos de desarrollo de software; pero cuando el proyecto requiere que los equipos estén geográficamente dispersos solo mencionan generalidades, por lo que se requieren un enfoque más específico en este ámbito.

Standish Group en su reporte Chaos Manifesto 2013 afirma que el 39% de los proyectos fueron exitosos (terminaron a tiempo, en el presupuesto y con las características requeridas); 43% tuvieron resultados cuestionables (atrasos, se pasaron del presupuesto y/o terminaron con menos de las funcionalidades requeridas); y 18% fracasaron (cancelados antes de terminar o completaron pero nunca se utilizaron) (The Standish Group International, 2013). A pesar de lo desalentador que parezca estas estadísticas, en realidad representan una mejora ya que en el 2004 la estadística empezó con un 29% de éxito en los proyectos que ha ido en aumento año tras año. Un dato que resulta interesante es que atribuye este incremento de éxito en las estadísticas a factores como una mejor comprensión del entorno del proyecto, métodos, técnicas, costos, herramientas, decisiones, las influencias internas y externas, y una mejor química del equipo. También resalta el incremento en el uso de metodologías ágiles y un decremento en los proyectos basados en metodologías en cascada.

La encuesta State of Agile de VersionOne del 2013 muestra resultados de que un 83% de los encuestados aseguran que sus empresas han usado prácticas ágiles de desarrollo, el 52% de los encuestados aseguraron usar ágil para administrar la mayoría de sus proyectos y que el número de equipos distribuidos usando ágil se ha duplicado en el año 2013, llegando a un 76% en contraste al 36% del 2012. (VersionOne, 2014).

Estos datos muestran una clara tendencia a buscar métodos ágiles en la gestión y desarrollo de software y más aún con equipos distribuidos. Por lo tanto es también necesario un marco de gestión de proyectos de desarrollo de software para equipos distribuidos geográficamente que se base en conceptos ágiles

El marco ágil para gestión de proyectos de software más utilizado en la actualidad es Scrum, sin embargo este marco de trabajo presenta múltiples debilidades al utilizarlo con equipos geográficamente dispersos ya que su fortaleza está basada principalmente en las personas y en la comunicación que hay entre ellas. Existen algunos estudios de personas con experiencia en la gestión ágil de este tipo de proyectos que nos dan algunas recomendaciones, pero es necesario encontrar las prácticas comunes que permiten asegurar el éxito de un proyecto de estas características. Sin embargo el trabajo y experiencia adquirida por metodologías desarrolladas por PMI, Prince2 o CMMI también es un punto de partida muy valioso, por lo que sus prácticas son consideradas en el presente documento.

2.1.1 Por qué es necesaria una gestión ágil de proyectos de software

“Lo único constante es el cambio”. Esta frase atribuida al filósofo griego Heráclito describe perfectamente lo que acontece en el mundo tecnológico en que vivimos. La tecnología cambia constantemente y cada vez se renueva más rápidamente.

Las empresas que triunfan son las que tienen capacidad de adaptarse a los continuos cambios del mercado y a las exigencias de los clientes de forma rápida y con altas dosis de innovación. Si el negocio cambia tan rápidamente y la tecnología también es lógico suponer que la gestión de proyectos también debe estar lista para asumir estos retos y transformar estos riesgos en oportunidades, por lo que a continuación se describe algunas de las razones de por qué hoy en día es tan necesaria una gestión ágil:

- Las metodologías ágiles permiten obtener productos en corto tiempo, lo que permite entregar resultados al cliente de forma temprana y así

también obtener una valiosa retroalimentación para seguir evolucionando y mejorando

- La flexibilidad permite realizar cambios de manera más rápida y sencilla
- Los equipos son reducidos y con tiempos de entrega ajustados, por lo que el costo de producción es menor
- La organización de un equipo es más sencilla, se reducen los niveles de administración y control, teniendo al cliente más cercano al equipo de desarrollo.
- El equipo de trabajo tiene más clara la visión global y los objetivos, lo que repercute positivamente en los resultados
- El producto final es más cercano a las necesidades del cliente ya que va evolucionando rápida y periódicamente hasta llegar a los resultados deseados
- Evita la generación de grandes cantidades de documentos que muchas veces solo son almacenados y son difíciles de catalogar y administrar
- La gestión ágil brinda una visión clara, rápida y en tiempo real de lo que está sucediendo en cada proyecto

2.2 Gestión de proyectos de software

2.2.1 Descripción

La gestión de proyectos puede ser predictiva o ágil, la elección depende de algunos factores, pero se puede resumir en el hecho mismo de la mayor diferencia de estas dos formas de gestión. La metodología predictiva se concentra y le da más valor a los procesos ya que se considera que los proyectos tienen comportamientos similares en el tiempo, mientras que la gestión ágil le da más valor al resultado o producto final.

En la actualidad una de las mayores problemáticas en la gestión de proyectos de software es la rapidez con la que los productos de software requieren ser desarrollados y en esa premura también deben cambiar o actualizarse para permanecer competitivos en el mercado, por lo que a más de contar con una metodología de desarrollo que facilite realizar estos frecuentes cambios,

también es igualmente importante contar con una metodología para gestión de proyectos de software ágil, que permita mantener un adecuado control y que bajo ninguna circunstancia se convierta en una carga adicional a llevar durante los proyectos.

Un aspecto importante a resaltar es que la gestión ágil tiene por debilidad una alta dependencia de una buena comunicación, por lo que recurrentemente se sugiere que las personas trabajen juntas. Esto principalmente se debe a la falta de documentación, por lo que la mayor parte del conocimiento del proyecto está en la memoria de las personas. Por ello al escoger una metodología ágil, con equipos geográficamente dispersos, se debe minimizar el impacto ocasionado por la dificultad de comunicación, lo cual es parte del objeto de estudio de esta tesis.

2.2.2 Comparación de metodologías

Con el propósito de identificar y determinar adecuadamente los requerimientos de una gestión ágil de proyectos de software que funcione con equipos geográficamente dispersos es necesario identificar las diferencias entre las metodologías tradicionales y ágiles, las diferencias y similitudes entre las metodologías ágiles, para finalmente hacer una comparación entre lo que es una gestión ágil y una formal.

Tabla 2. Comparación entre metodologías ágiles y tradicionales

Metodología Tradicional	Metodología Ágil
Dificultad para adaptarse a los cambios	Diseñada para adaptarse a los cambios
Impuestas externamente	Impuestas internamente (empoderamiento del equipo)
Proceso muy controlado con políticas y normas	Proceso menos controlado con principios
Se basa en un contrato prefijado	El contrato no existe o es más flexible
El cliente no es parte del equipo pero participa mediante reuniones	El cliente es parte del equipo de desarrollo
Puede usarse con grupos grandes de desarrollo y distribuidos	Requiere grupos pequeños y en el mismo sitio

Muchos artefactos	Pocos artefactos
La arquitectura es fundamental y se la documenta extensamente en modelos	Hay menor énfasis en la arquitectura

Tomado de: Autores de la Tesis

Tabla 3. Comparación entre metodologías tradicionales

Nombre	Origen	Tipo	Descripción
PMBok	PMI, 1990	Es una guía. Orientado a gerentes de proyecto	Metodología basada en buenas prácticas y orientado a la finalización del proyecto.
Prince2	CCTA, 1989	Es un Método. Orientado a toda la organización que interviene en el proyecto.	Orientado a la consecución del Business Case
CMMI	SEI, 2012	Son modelos, no son procesos, ni descripciones de procesos.	Colecciones de buenas prácticas para apoyar en mejorar los procesos

Tomado de: Autores de la Tesis

Tabla 4. Comparación entre metodologías ágiles

Nombre	Origen	Tipo	Descripción
LD	Bob Charette 1993	Forma de pensar Modelo Logístico	Metodología basada en procesos productivos
XP	Beck 1999	Se enfoca en el proceso de desarrollo	Método ágil radical
Scrum	Sutherland 1994 Schwaber 1995	Metodología para gestión de proyectos ágiles	Se complementa de otros métodos para desarrollo
Crystal	Cockbum 1998	Familia de metodologías. Cubre la gestión de proyectos	Metodología ágil con énfasis en modelo de ciclos

DSDM	Stapleton 1997	Modelo de Procesos para desarrollo Cubre la gestión de proyectos	Creado por 16 expertos en RAD
FDD	Jeff De Luca y Peter Coad 1998	Metodología para desarrollo	Método ágil de diseño y Construcción
ASD	Highsmith 2000	modelo de implementación de patrones ágiles Cubre la gestión de proyectos	Inspirado en sistemas adaptativos complejos
AM	Ambler 2002	Metodología basada en la Práctica	Suministra modelado ágil a otros métodos

Tomado de: Autores de la Tesis

Tabla 5. Comparación entre la gestión de proyectos formal y ágil

	Formal	Ágil
Contexto	Predictiva. Estable, sin cambios	Adaptativa. Dinámico, cambiante
Prioridad	Cumplimiento del plan	Valor del resultado
Importancia de usar prototipos	Baja	Alta
Nivel profesional sugerido	Según el rol	Sénior
Modelo de desarrollo	Procesos	Personas

Tomado de: Autores de la Tesis

2.2.3 Síntesis de los modelos de gestión

2.2.3.1 Beneficios y debilidades de una gestión tradicional de proyectos

La gestión tradicional utiliza mucha planificación para tratar de prever todos los posibles escenarios, llegando a definiciones con alto nivel de detalle lo que

requiere de mucho tiempo y esfuerzo en definiciones de planes para condiciones que muy probablemente nunca se den durante el proyecto o sucedan de manera diferente e imprevista. Cuando se requiere un cambio se debe hacer uso del control de cambios, lo cual no es más que otra planificación sobre cómo manejar el cambio y su impacto en el proyecto.

Se ha trabajado con el “triángulo de hierro”, que considera alcance, costo y tiempo, donde los proyectos se entregarían a tiempo, por debajo del presupuesto y con máxima calidad, pero para proyectos de desarrollo de software esto rara vez se logra alcanzar.

Su enfoque en los procesos requiere de altas cantidades de documentación con la finalidad de no depender de las personas, de manera que remplazarlas tenga un impacto mínimo en el proyecto. Adicionalmente este enfoque permite que los integrantes del equipo puedan tener niveles profesionales según su rol a cumplir, reduciendo la necesidad de contar solo con gente de amplia experiencia

2.2.3.2 Beneficios y debilidades de una gestión ágil de proyectos

La gestión ágil aplica el proceso empírico, es decir que aprende a medida que avanza, en lugar de planear todo anticipadamente, este proceso estipula “planear para el cambio”, y los cambios se irán presentando a través de la inspección y la adaptación.

Esta gestión considera la transparencia, la inspección y la adaptación, como sus pilares a seguir. Donde transparencia se refiere a comunicación abierta con el equipo, la inspección se la realiza de manera sencilla y en tiempo real apoyado en herramientas como el *Burn down chart* (ver anexo 3) que presenta el avance del trabajo y permite proyectar los resultados en el tiempo, y la adaptación es una tarea repetitiva y periódica que permite realizar modificaciones y mejoras en las tareas que puedan ser optimizadas.

Al apoyarse directamente en las personas las debilidades de este tipo de gestión están directamente relacionadas con las debilidades del trabajo en equipo, por lo que tenemos que:

- Se recomienda que las personas estén físicamente juntas para facilitar la comunicación. Esto es principalmente importante debido a varias razones como el hecho de que la documentación puede no ser muy extensa
- Requiere de gente motivada, lo cual no debería normalmente ser un inconveniente pero en algunas empresas puede terminar siendo un tema de cultura empresarial que debe ser reinventado

2.3 Casos de estudio

Con el propósito de aportar al marco de gestión de este documento la experiencia profesional se describe la situación actual de Cobiscorp y Sonda del Ecuador, donde existe una clara necesidad de un marco de trabajo que permita solventar los inconvenientes a continuación descritos.

2.3.1 Caso de Estudio 1

Cobiscorp es una empresa multinacional que actualmente está cambiando de una metodología de gestión y desarrollo de software basada en RUP a una metodología ágil basada en Scrum y Kanban. Esta empresa, que se dedica a la comercialización de soluciones para la Banca, cuenta con alrededor de 700 empleados, comercializa sus productos en Latinoamérica, África y Japón, tiene oficinas en más de 5 países. Las principales sedes de desarrollo de software están en Ecuador y Colombia, las personalizaciones al software en algunas ocasiones se realizaron en el país donde se esté instalando el software y en los últimos años algunos proyectos han recurrido a fábricas de software en Colombia y en la India.

Después de estas experiencias se observaron problemas en algunos ámbitos como:

- Problemas de comunicación

- Diferencia de idioma
- Diferentes usos horarios y diferente calendario laboral
- El levantamiento y especificación de requerimientos dura 3 meses en promedio y está sujeto a múltiples cambios durante el proyecto
- Falta de detalle en los diseños ocasionada por falta de claridad en los documentos de especificación
- Baja calidad en los resultados obtenidos al contratar fábricas de software para acelerar los desarrollos

Por estos problemas la empresa ha decidido adoptar metodologías ágiles y formar equipos multidisciplinarios, y aunque se observan debilidades en estas metodologías que deben ser solventadas, especialmente los problemas ocasionados por el trabajo distribuido, se espera que este cambio pueda generar un ambiente más productivo y lograr resultados que los clientes puedan apreciar en plazos más cortos.

En lo antes descrito se observa claramente la necesidad de un marco de trabajo de gestión de proyectos que funcione con metodologías de desarrollo ágiles y que al mismo tiempo permitan un trabajo con equipos de desarrollo distribuidos.

2.3.2 Caso de Estudio 2

Sonda del Ecuador es una empresa que provee servicios a 10 países de Latinoamérica como son: Argentina, Brasil, Chile, Colombia, Costa Rica, Ecuador, México, Panamá, Perú y Uruguay, su matriz está en Chile.

Es una de las mayores empresas latinoamericanas de Servicios de TI e Integración de Sistemas.

En SONDA, como empresa líder en la industria de TI en Latinoamérica, también se han adoptado “mejores prácticas” para la gestión de TIC, mediante un sistema de gestión de calidad que, además de contar con los conocimientos, credenciales y certificaciones destinados a prestar servicios de calidad estandarizada y homogénea, según las mejores prácticas de la industria TI, tales como ISO9001, ISO 27001, ITIL, PMO ó CMMI.

Dentro de los servicios que presta está el Outsourcing de Aplicaciones, que es el objeto de este caso de estudio; ya que este servicio es el único que tiene proyectos de desarrollo de software, que están dirigidos a un único cliente en Ecuador.

Al momento se está trabajando bajo la metodología MSF misma que combina el concepto de la administración de proyectos tradicional (cascada), con los modelos en espiral (mejora continua). Aprovechando los beneficios de la planeación en cascada que se basa en el alcance de los hitos, con los entregables iterativos e incrementales del modelo en espiral.

El servicio de Outsourcing de Aplicaciones se lo viene prestando por alrededor de 10 años con el mismo cliente, renovando el contrato de forma anual. Debido a las fuertes restricciones de seguridad y control del cliente se ha tenido que dividir el equipo entre las instalaciones de Sonda y el cliente, y dada esta situación se tiene la necesidad de realizar una adecuada gestión de coordinación entre ambos equipos de desarrollo. Adicionalmente el cliente tiene el concepto de trabajo flexible lo que faculta a que se realice la gestión del proyecto incluso desde casa. Este beneficio surge sobre todo por la necesidad de apoyar en tareas que requieren trabajar en horas fuera del horario normal, como es el caso de implementación de ambientes de prueba o puesta en producción de sistemas. En cambio, las tareas que requieren coordinación dentro del horario de trabajo son: coordinación del equipo de desarrollo, levantamiento de requerimientos, alcance y análisis (UML), documentación (documento de requerimientos funcionales, diagramas de casos de uso y diagramas de secuencia), realización de cronogramas, seguimiento de los proyectos, informes de avance y reuniones.

Se cuenta con un equipo de 6 personas, que se encargan de los desarrollos, soporte y corrección de sistemas en producción.

Dentro de este esquema se ha encontrado varias oportunidades de mejora como son:

- Los requerimientos entregados son en algunos casos muy generales, se requiere de varias reuniones presenciales para conseguir más detalle, esto conlleva múltiples demoras y retrasos en las planificaciones iniciales.
- El poder realizar un seguimiento remoto de proyectos más efectivo, que reduzca la necesidad de tener que ir donde el cliente y por consiguiente no se pierda tiempo en traslados.
- Los problemas de comunicación provocan reprocesos y por consiguiente incumplimiento de los cronogramas.
- Empoderar y motivar de mejor manera al equipo remoto para que busquen menos autorización de parte de la coordinación para ejecutar su trabajo.
- Al realizar un control de cambio se tiene desconocimiento de las posibles afectaciones.
- El conocimiento de los sistemas desarrollados, se encuentra en la mayoría de los casos concentrado en un solo especialista.
- Se realizan reuniones de equipo semanales para revisión de estatus de proyectos y detección de trabas que impiden avanzar con las tareas planificadas, pero en algunas ocasiones se observa tareas que están detenidas por días hasta que se realice esta reunión, ocasionando cuellos de botella.

Con lo anteriormente mencionado, se evidencia la necesidad de mejorar la gestión de proyectos para obtener canales de comunicación adecuados y respuestas más ágiles a los requerimientos del cliente, de manera que se mejore el desempeño del equipo, para seguir siendo competitivos y apoyar de manera más eficiente al cliente.

3 Capítulo III: Marco de trabajo propuesto

3.1 Visión general

El marco de trabajo propuesto facilita una gestión ágil de proyectos de desarrollo de software incorporando elementos que permiten una comunicación y coordinación adecuada entre equipos geográficamente dispersos, ya sea dentro de una misma región como en países diferentes, que fácilmente se puede integrar o complementar a otros marcos de gestión.

Primero se describen las fases de la gestión de proyectos de desarrollo de software propuestas, las cuales buscan una gestión ágil sin descuidar aspectos formales necesarios por la naturaleza misma de este tipo de proyectos distribuidos. Estas fases tienen mucha semejanza con las ya propuestas en numerosas metodologías, pero es importante entender el enfoque que este marco le da a cada fase dentro de cada uno de los dominios después descritos.

Luego, Independientemente de la metodología de desarrollo de software utilizada se propone realizar la gestión enfocada en cuatro dominios que son:

- 1) Conformación y coordinación de equipos
- 2) Comunicación
- 3) Tecnologías de información (herramientas)
- 4) Seguimiento y trazabilidad

Mediante estos cuatro dominios se busca solventar los requerimientos para una gestión ágil exitosa, mitigando los riesgos inherentes a la ejecución de proyectos geográficamente distribuidos y potenciando sus ventajas (identificados en capítulos anteriores). De manera que se consiga una excelente comunicación, un equipo integrado y motivado, con las herramientas para potenciar su productividad y con un nivel adecuado de trazabilidad para cada uno de los componentes creados; utilizando una serie de prácticas como las de los equipos virtuales altamente efectivos (Levit, 2012) y otra serie de prácticas recopiladas de la industria para equipos ágiles distribuidos

(Sutherland, 2007), para equipos virtuales (Watkins, 2013), para gestión ágil efectiva (Bourne, 2010) y otras prácticas internacionalmente reconocidas.

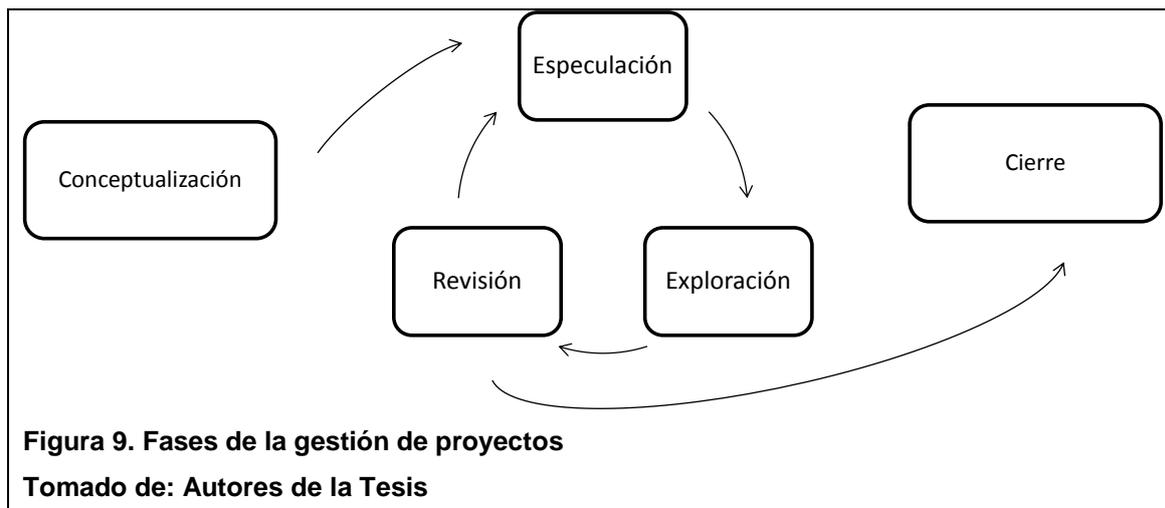
Finalmente se elabora una guía muy práctica, que funciona como el enlace entre las fases y los dominios planteados indicando cómo aplicar el presente marco de trabajo en cada una de las fases de la gestión de proyectos de desarrollo de software.

3.2 Fases de la gestión de proyectos de desarrollo de software del marco de trabajo propuesto

El marco propuesto para la gestión de proyectos incluye agilidad y un enfoque iterativo e incremental, que inicia con un adecuado entendimiento de la visión del producto al cual se va acercando mediante sucesivos incrementos, priorizando siempre lo que da más valor al negocio. Entendiéndose que la información inicial no es ni completa ni perfecta y que el producto ideal final en muchas ocasiones puede ir variando por condiciones del entorno y necesidades del negocio.

Son cinco las fases que se requieren para proyectos de desarrollo:

- 1) Conceptualización
- 2) Especulación
- 3) Exploración
- 4) Revisión
- 5) Cierre



Como se observa en la gráfica anterior las cinco fases están delimitadas por un principio (Conceptualización) y un fin (Cierre) y en la mitad se suceden las diferentes iteraciones que permiten la construcción incremental del producto mediante tres fases que son: especulación, exploración y revisión.

3.2.1 Conceptualización

En esta fase el objetivo es definir la visión del producto y seleccionar el equipo de personas que serán parte del proyecto.

Sin visión las posibilidades de tener éxito en el proyecto son muy bajas, ya que es muy probable que durante el proyecto se termine realizando actividades o tareas que no aportan en el producto final, se pierda la sinergia del equipo al no tener un objetivo claro y los costos y tiempos se terminan incrementando.

La visión no solo debe ser accesible para todos los miembros del equipo, sino que tiene que estar entendida por todos. La visión define el alcance del proyecto.

3.2.2 Especulación

Luego de tener claro lo que se quiere hacer el siguiente paso es realizar su especulación. Esta fase consiste en especular en base a la información disponible y a la visión el primer diseño en alto nivel de lo que será el producto final. En una metodología ágil no se espera a disponer de todos los

requerimientos para iniciar la ejecución del proyecto, al contrario, se reúne la información disponible, se la prioriza en base a los elementos que puedan dar más valor al producto y se planifica una primera aproximación. Aunque el cliente proporcione la mayor cantidad posible de información del producto deseado durante el tiempo, las condiciones del entorno pueden cambiar provocando la necesidad de realizar cambios o incluir nuevas características al sistema.

Esta fase debe repetirse en cada iteración y siempre considerando la visión, alcance del proyecto y la información en ese momento disponible, se deben realizar las siguientes actividades:

- Diseño conceptual del producto
- Lista de funcionalidades requeridas por el sistema
- Elaborar un plan de entrega con fechas de entrega, hitos, Iteraciones y estimaciones de tiempo.
- Plan de gestión de riesgos
- Análisis financiero de los costos y beneficios para el presupuesto general.

3.2.3 Exploración

En esta fase se desarrollan las funcionalidades de la iteración en curso que permiten la obtención de un incremento del producto basado en la planificación realizada. En la exploración está implícita la necesidad de coordinación de los recursos y personas del proyecto.

Al ser un proyecto de desarrollo de software es en esta fase donde más aportan las metodologías de desarrollo de software.

3.2.4 Revisión

Se realiza una revisión de los resultados obtenidos en el producto de la iteración, para determinar si está alineado con lo que espera el cliente y funciona como estaba previsto. También se verifica que el proyecto siga una

dirección adecuada según los objetivos planteados. En esta revisión participa tanto el equipo como los usuarios.

También es importante revisar y recordar todos los inconvenientes que pudieron acontecer durante la ejecución de la última iteración, con el propósito de buscar acciones de mejora inmediata y se constata que la ejecución haya terminado en los tiempos previstos, de lo contrario en base a la experiencia obtenida se realizan ajustes para evitar desviaciones futuras.

3.2.5 Cierre

En el cierre se realiza la entrega del producto final, el cual fue verificado en cada una de las iteraciones.

En algunos casos el cierre puede no significar una terminación real del proyecto, ya que puede ser necesario el inicio de un nuevo proyecto de mantenimiento, el cual permita realizar nuevos ciclos incrementales para la liberación de nuevas versiones para evitar su obsolescencia o para satisfacer necesidades nuevas del entorno.

Durante esta fase se archivan los documentos y se realiza un informe con los resultados obtenidos. En muchos casos esta fase concluye con el cierre del contrato.

3.3 Dominios del marco propuesto

Durante las diferentes fases del proyecto se va a requerir hacer uso extensivo de las prácticas y recomendaciones de los dominios a continuación descritos, con el objetivo de conformar un equipo adecuadamente integrado y coordinado, con excelente comunicación, con las herramientas que ayudan a mejorar su productividad y con la trazabilidad del proyecto que facilite su seguimiento y control.

3.3.1 Conformación de equipos

La conformación de equipos en este marco de trabajo está clasificada según su forma de interacción, ya que cada uno tiene necesidades y comportamientos diferentes de comunicación.

Uno de los motivantes para realizar proyectos con equipos distribuidos suele ser la reducción de costos, ya que en otros países se pueden encontrar empresas con personal calificado a costos muy exequibles. Esto no necesariamente es verdad, porque existen costos ocultos que normalmente no son considerados, como lo son el tiempo y esfuerzo de búsqueda de personal o empresas y su evaluación, el tiempo y esfuerzo adicional en refinar los detalles del proyecto para poder transmitírselo a personal sin conocimiento alguno de las necesidades de la empresa, los esfuerzos adicionales de control y coordinación realizados durante el proyecto, etc. Por lo que el costo no debería ser un motivante para emprender un proyecto con equipos distribuidos. Razones por las que se podría requerir formar un equipo distribuido son: necesidad de personal con conocimientos específicos, proyectos que requieren de personal adicional que localmente ya no puede conseguirse por diferentes razones o necesidades temporales de mayor cantidad de personal, entre otras anteriormente descritas en la sección del teletrabajo. Una vez establecida claramente la necesidad de formar un equipo distribuido es importante escoger el nivel de integración adecuado según la clasificación a continuación descrita.

3.3.1.1 Clasificación de Equipos

Los equipos pueden clasificarse de la siguiente manera:

- Aislados
- Distribuidos
- Totalmente Integrados

3.3.1.1.1 Equipos Aislados

Este tipo de equipos se caracterizan por que trabajan como su nombre lo indica aisladamente del equipo de la empresa matriz.

La mayoría de desarrollos subcontratados usan modelos aislados, los requerimientos son generados en la empresa matriz y desarrollados por un equipo de una empresa diferente (Ej. Subcontratación de empresas en la India). En ocasiones esto resulta en problemas de comunicación, diferencias en la forma de trabajo y metodologías de desarrollo y gestión diferentes. Por lo que escoger este tipo de forma de trabajo debe estar limitado a proyectos con escasa o nula necesidad de coordinación. Una desventaja es que los equipos aislados a menudo tienen impedimentos que pueden ocasionar demoras y errores.

3.3.1.1.2 Equipos Distribuidos

Si el proyecto necesita coordinación es mejor formar equipos distribuidos donde existen equipos independientes, con sus propias responsabilidades, trabajando en el mismo proyecto, aunque comúnmente en módulos o requerimientos diferentes, que están coordinados por un sólo líder de proyecto.

Este tipo de equipos necesita que el dueño del producto se reúna las veces que sea necesario con cada uno de los equipos para aclarar los objetivos y requerimientos, y que los desarrolladores del equipo remoto sientan que existe un adecuado canal de comunicación al que puedan recurrir en caso de dudas.

3.3.1.1.3 Equipos Totalmente Integrados

Un equipo totalmente integrado se caracteriza porque los miembros de este equipo pueden estar geográficamente dispersos, pero trabajan en conjunto sobre los mismos requerimientos, es decir funcionan como su nombre lo indica, totalmente integrados, tal cual si no estuvieran separados geográficamente.

La práctica recomendada es tener la mitad del equipo de forma local y la mitad de forma remota y lograr que trabajen coordinadamente. Por lo cual es importante iniciar el proyecto con todo el equipo junto con el objetivo de que se conozcan y se creen lazos que faciliten la colaboración.

Este tipo de equipos requiere de una coordinación centralizada, por lo que debe haber un solo líder que coordina a ambos equipos y un solo dueño del

producto, aunque ambos podrán designar personas alternas a las cuales recurrir en caso de que no estén disponibles.

La mayoría de las experiencias documentadas recomiendan este tipo de conformación de equipo.(Sutherland, 2007)

Para este tipo de equipos debe considerarse muy seriamente seguir las prácticas de conformación de equipos distribuidos y las actividades para crear sinergia descritas a continuación.

3.3.1.2 Prácticas para conformar equipos distribuidos.

A continuación ponemos a consideración las siguientes prácticas para conformar equipos distribuidos.

1) Fomentar la unidad

En una oficina, la unión del personal por lo general ocurre con los colegas de trabajo al compartir del día a día. Las relaciones se desarrollan y las personas aprenden unas de otras. En la actualidad con tantos equipos de trabajo virtuales está dinámica se ve modificada, convirtiéndose en un reto el poder generar un agradable ambiente de trabajo y unión de grupo, y se dificulta saber por dónde empezar. A continuación se describen varios puntos que podrían servir para apoyar en esta metamorfosis.

- Reunir al equipo físicamente, al inicio del proyecto y reconectarlos de forma periódica dependiendo de la duración del proyecto, esto fomenta las relaciones personales que darán cohesión a los equipos. Como lo describe Jason Christensen durante esta reunión se podrían (Christensen, 2010):
 - Establecer metas y objetivos como grupo
 - Definir las funciones del equipo
 - Establecer relaciones entre los miembros del equipo
 - Construir una identidad de equipo
 - Construir una base para la confianza

Así también en esta cita los miembros podrían darse a conocer, mencionando:

- Quiénes son.
 - De dónde vienen
 - Su familia
 - Sus intereses
 - Las experiencias de vida
 - Metas personales
 - Antecedentes profesionales
 - Qué rol tendrán en el equipo
 - Qué quieren lograr en su carrera
 - Sus expectativas
 - Y por qué no también hablar de sus temores
- Abrir espacios para la camaradería, en estos espacios se podría obtener la mejor información, ya que se produce en un espacio más relajado.
 - Publicar lo tratado en esta primera cita en el espacio virtual de trabajo interno o en las redes sociales en grupos privados para que haya más conocimiento de las personas que se estará tratando.
 - Para que el equipo sea productivo, debe mantenerse feliz. Tratar a todos por igual. Si se espera lo mejor del equipo ellos tratarán de cumplir las expectativas.
 - Motivar a los miembros, dándoles autonomía y removiendo obstáculos y cuellos de botella que impidan alcanzar las metas.
 - El dueño del producto y el equipo deben compartir tiempo de calidad.

2) Construir una identidad de Equipo

“Las personas tienen interés inherente en sentirse parte de algo especial” (Christensen, 2010)

Construir identidad en el equipo le permite trabajar construyendo un distintivo que los identificará, elementos que pueden definirse son:

- Nombre del equipo
- Logo
- Marca
- Avatares del equipo

Usar este distintivo en los documentos del proyecto, presentaciones, sitios, material relacionado. Apoyará a fortalecer el carácter y pertenencia del equipo. Se puede usar sus avatares o fotografías actualizadas en los chats y cuadros de control del proyecto.

3) Establecer las reglas para la interacción con el equipo

Se deben establecer las reglas debido a que los equipos distribuidos requieren de más estructura, gestión y normas para trabajar efectivamente. Con este propósito se deberá:

- Designar un facilitador para el equipo, que apoye en las tareas de comunicación con el equipo remoto y se quede a cargo en los momentos que el líder este con el equipo remoto.
- Planificar para que las iteraciones sean cortas, con una duración de 2 a 3 semanas.
- El equipo creará la lista de características de alto nivel y sus estimaciones. El detalle se lo completará offline y luego se lo comunicará en la próxima reunión o por mail.
- Se realizará una reunión diaria
 - Las reuniones locales se las realizará 2 veces por semana lideradas por el líder o facilitador a cargo. Las reuniones conjuntas se las llevará a cabo los otros 3 días. Para estas reuniones se requiere ponerse de acuerdo en una hora que sea justa tanto para los equipos locales como para los remotos. Al pasar del tiempo se logrará una mejor sinergia en el equipo, por lo que podrá establecerse las reuniones locales en 3 días a la semana y las remotas en 2 días a la semana.

- Cerrar la conversación de manera puntual para evitar sobreesfuerzos.
- Tomar decisiones basadas en lo que se conoce y revisarlas únicamente cuando nueva información se presente.
- Nunca dejar los plazos de decisión abiertos, estos deben durar máximo 1 o 2 días.
- Crear una agenda con horas específicas de temas de discusión.
- Una manera de agradecer la asistencia a las reuniones es acompañarlo de comida y bebida.
- Tener inspección de código de pares y revisiones formales que involucren a todo el equipo. Esto debe ocurrir cada 2 iteraciones y se debe seleccionar a alguien para presentar. Una forma práctica de realizar esta presentación es el *brown bag* (ver anexo 3).
- Empezar eliminando las preconcepciones de lo que se puede hacer localmente y que remotamente.
- Mantener a mano los datos de contacto de las personas de los equipos (Teléfono, Nick, Horas de contacto), poner en el espacio de trabajo compartido.
- Las grabaciones realizadas de las comunicaciones deben tener un determinado tiempo, esto se revisará y acordará con el equipo. Si este tiempo se excede, cualquier miembro del equipo podría alertar para remediar este exceso.
- Realizar videoconferencias periódicas, tanto individuales como grupales.

4) Desarrollar los Objetivos como Equipo

Todo el equipo debe conocer que es lo que se desea conseguir, y entender que basándose en la colaboración es que se logrará que los objetivos se cumplan. El líder deberá asegurarse que su plan de acción es SMART por sus siglas en inglés:

- Específico
- Medible
- Alcanzable
- Relevante
- Programado

“Al permitir que el equipo establezca las metas por sí mismo, les da empoderamiento en el proceso sabiendo que eran parte de la toma de decisiones” (Christensen, 2010)

5) Esclarecer tareas y procesos, además de objetivos y funciones

Con los equipos distribuidos, la coordinación es de por sí más que un desafío, por que las personas no son localizables a simple vista, debido a esto se requiere dar la mayor claridad en el proceso de trabajo, siendo necesario:

- Como principal tarea de líder, debe alinear objetivos, funciones y responsabilidades en los primeros 90 días. Aclarar las funciones y responsabilidad mejora la comprensión del trabajo y aumenta la productividad.
- Poner atención en los detalles de diseño de las tareas y procesos.
- Simplificar las tareas
- Asignarlas a subgrupos de 2 o 3 miembros del equipo
- Asegurarse que está claro el proceso de trabajo con detalles de quién hace qué y cuándo.
- Realizar revisiones posteriores, que evaluarán cómo van las cosas y determinar ajustes en los procesos y necesidades de capacitación.

6) Entrenamiento al equipo

El entrenamiento debería considerarse como una estrategia para poder mejorar las competencias necesarias y obtener el mejor desempeño del equipo.

Así esto será un factor clave para superar los problemas de la comunicación electrónica y de la etiqueta virtual (Shea, 2012), mejorando la cohesión y satisfacción en los procesos de equipo.

Posibles temas de entrenamiento, podrían ser:

- Habilidades de comunicación.
- Cómo manejar sus tiempos y control de avances
- Qué tecnología elegir dependiendo de la tarea
- Cómo resolver conflictos en un equipo virtual
- Cómo establecer normas
- Cómo asignar las funciones al equipo
- Cómo generar confianza
- Cómo crear equipos eficientes para trabajar con las TICs
- Cómo liderar estos equipos
- Cómo reconocer los logros individuales y de equipo.

7) Construir un equipo que marque al mismo ritmo

Cuando uno o todos los integrantes del equipo están separados es fácil quedar desconectado del ritmo normal de trabajo, para minimizar esto, considerar:

- Esforzarse por que se sientan un solo equipo. Considerar seriamente el efecto de la Ley de Conway que dice: “Las organizaciones que diseñan sistemas están limitadas a producir diseños que son copias de las estructuras de comunicación de estas organizaciones.” (Conway, 1968)
- Conectarse con los ritmos normales de la vida laboral
- Ser disciplinado y respetar el ritmo de trabajo en equipo
 - Establecer y compartir la agenda con antelación.
 - Idealmente tener reuniones todos los días y a la misma hora.
 - Tener claros los acuerdos de comunicación.
 - Ser puntual, para la llegada y la salida de la conexión.

- En distintas zonas horarias, rotar regularmente los horarios de reunión para distribuir la carga de manera equitativa.

8) Seguimiento de acuerdos

El equipo multidisciplinario que se ha formado, debe también ser empoderado de sus tareas y compromisos, así se reducirá la desmotivación que puede causar un exceso de supervisión.

- Diseñar y aclarar las tareas
- Revisión corta y regular de estatus
- Crear un tablero de entregables que sea visible para todo el equipo.

9) Fomentar el liderazgo compartido

Resulta crucial el encontrar formas de involucrar a otros en la dirección del equipo. Para fomentarlo se podría realizar lo siguiente:

- Asignar responsabilidades
- Liderazgo compartido, involucrar a otros en la dirección del equipo.
 - Asignar a proyectos especiales como investigar, identificar y compartir las mejores prácticas.
 - Lograr que miembros entrenen a otros en sus áreas de especialización
 - Asignarlos como mentores para miembros nuevos del equipo.
 - Ponerlos a ejecutar alguna actividad virtual para el equipo.

10) Recordar el Uno a Uno

En las interacciones de gestión del rendimiento y de entrenamiento de los líderes es importante considerar el uno a uno con los miembros del equipo ya que son una parte fundamental para hacer cualquier trabajo en equipo. Y considerar:

- En las reuniones de seguimiento no solo verificar estatus, además hacer que se sientan conectados con la visión del proyecto y resaltar que lo están haciendo juntos.
- Con equipos heredados tomar el tiempo de entender como dirigió el antecesor.
- Ayudar a identificar los temas y en los 90 días afinar el equipo.

11) Celebrar los logros obtenidos

Después del trabajo duro, que mejor que esto sea reconocido y publicado. Es importante considerar los logros del equipo.

- El líder debe informar los éxitos para que toda la empresa sea participe y recompensar al equipo.
- Reconocer el sacrificio y agradecer al inicio y fin de cada reunión, podría considerarse el organizarlas con bocadillos.
- Realizar teleconferencias que cuente con la presencia de todo el equipo.
- Enviar presentes tangibles vía encomienda
- Enviar mails de reconocimiento
- Publicar en el muro común, los reconocimientos.

3.3.1.3 Actividades para generar sinergia en el equipo

Las actividades deben de transportarse del mundo real al mundo virtual, el equipo debe encontrar un lugar para interactuar con libertad y poder conocer más a las personas que están detrás de una voz o un texto.

Se podrían analizar los intereses comunes del equipo para poder organizar diferentes eventos que convoquen a todos a participar.

Se podrían mencionar:

- Interactuar en redes sociales formando grupos privados.
- Juegos de integración que pueden estar en una red social empresarial.

- Juegos de estrategia en línea
- Organizar Eventos de Second Life o similares
- Participar en ejercicios de agricultura.
- Realizar torneos de póquer en línea
- Tener espacios para meditación, técnicas de yoga, manejo de estrés.

3.3.2 Comunicación

La falta de comunicación genera dudas, desconfianza, aislamiento y falta de identificación con el equipo.

A continuación se detallan las prácticas que se consideran podrían apoyar a minimizar estas consecuencias.

3.3.2.1 Prácticas para apoyar en el proceso de comunicación

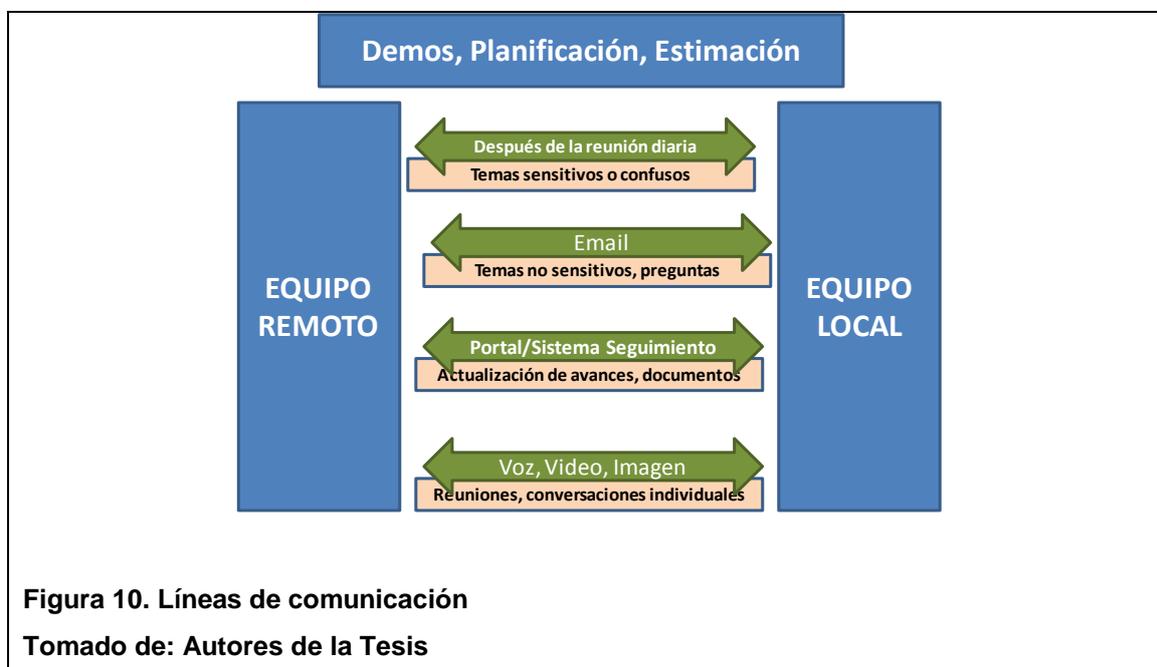
Para garantizar una buena comunicación se podría considerar las siguientes prácticas:

1) Crear un código para facilitar la comunicación

A menudo la comunicación de los equipos virtuales es menos rica porque no hay la interacción cara a cara, para evitar inconvenientes es mejor ser claro y disciplinado sobre como el equipo se va a comunicar y establecer explícitamente un código común de comunicación sobre los elementos más importantes. Para esto se requiere:

- Comprometerse a seguir el código.
- La comunicación la inicia el líder que presenta la visión de la empresa, el equipo y el proyecto.
- Se podría realizar una cartilla donde se establecen las normas de comportamiento al participar en reuniones virtuales.
 - Limitar el ruido de fondo.
 - Concentrarse en la conversación en curso.
 - Hablar con claridad
 - Escuchar con atención y no dominar la conversación.

- Se podría realizar una cartilla donde se establecen las directrices para el uso de los medios de comunicación y en qué circunstancias.
 - Cuándo recurrir al teléfono
 - Cuándo usar el chat
 - Cuándo responder por mail
 - Cuándo crear y compartir un documento
- Establecer líneas de comunicación, al hacer esto se reduce la clásica excusa “estuve esperando por una respuesta”



- Recordar que en las reuniones diarias es el tiempo de los desarrolladores, no del líder.
- Mejorar la comunicación aumentando un poco la documentación, pero solo actualizar los documentos cuando sean útiles.
- Definir técnicas para la resolución de conflictos y para la toma de decisiones.
- Usar lenguaje simple para explicar problemas complejos.
- Democratizar la participación.

- En las presentaciones combinar audio con recursos gráficos.
- Contestar los mails por más obvio que parezcan las respuestas, aquí pueden existir detalles muy importantes.
- Incluir emoticones para poder incluirle a la comunicación un toque diferente, incluso más personalizado.

2) Acordar el lenguaje a usarse

En la interacción que se producirá entre varias culturas e idiomas, que unirán esfuerzos para la generación de ideas y resolución de problemas, no deben existir divergencias en la comunicación ya que esto podría causar algunos inconvenientes. Para evitar estos problemas debemos:

- Revisar las interpretaciones de palabras y frases importantes, para llegar a acuerdos.
- Crear un diccionario de terminología del proyecto. (evita ambigüedades)
- Publicar lo acordado en el espacio de trabajo compartido.
- El líder o facilitador es el encargado de que todos expresen sus opiniones.

3) Crear un espacio para la interacción informal

Al no existir la posibilidad de encontrarse para compartir un café, bromear, hablar de cosas que podrían molestar, tal vez se está perdiendo la oportunidad de tener información valiosa y de afianzar los lazos del equipo. Se podría considerar lo siguiente:

- Comenzar cada sesión en la que cada integrante tome dos minutos para hablar de lo que están haciendo, lo que va bien y lo que es un reto.
- Tomar nota y hacer checking en la próxima reunión de lo que podrían considerarlo que está mal.
- Realizar ejercicios para interactuar virtualmente, dándole poco de diversión para salir de la rutina.

- Tener espacios de trabajo compartidos con características de redes sociales, esto ayudaría a que el equipo se sienta más conectado.

4) Comunicación del Líder

El líder para los equipos virtuales y geográficamente dispersos, al no tener un contacto permanente cara a cara, debe complementarse con competencias adicionales. Se pueden mencionar: producir resultados claramente visibles, ser ejemplo, ser analista de negocios, ser sponsor, ser entrenador y ser facilitador.

- El líder debe tener precaución al momento de enviar los mensajes, sobre todo vía electrónica. Debe siempre asegurarse de que el receptor recibe y entiende el mensaje.
- El líder debe tener en cuenta los potenciales canales de comunicación, como indicador de la complejidad de las comunicaciones en un proyecto determinado.
 - Fórmula: $N \times (N - 1) / 2$ Donde N representa la cantidad de personas
- El líder debe siempre conocer y entender todas las situaciones que sucedan en su proyecto, y debe saber comunicar en tiempo y forma los acontecimientos y/o los hechos que sucedan o que puedan suceder, afrontando las situaciones con responsabilidad, compromiso y sobre todo, siempre con la verdad.
- El líder debe trabajar en la seguridad psicológica del equipo, permitiendo una fluida comunicación. Nadie puede trabajar al nivel más alto de su desempeño si no se siente seguro y sin miedo.

5) Comunicación entre miembros del equipo

Las personas que participen en estos equipos, deben conocer las condiciones, incluso limitaciones que tendrán que sortear para generar un proyecto exitoso.

Estos integrantes deberían ser:

- Seguros de sí mismos
- Independientes
- Proactivos
- Comunicativos, que busquen activamente la comunicación con sus compañeros.
- Dispuestos a interactuar y comunicarse con otras culturas.

6) Comunicación virtual

Los equipos de trabajo tendrán mayor interacción virtual dando como consecuencia la búsqueda de sistemas de intercambio de archivos, más eficientes.

Dependiendo del tamaño de los archivos, se irá cambiando el medio de intercambio, es así que se para archivos pequeños se podría usar el mail, si el tamaño impide el enviar o recibir los mails, se podría usar FTP, aunque la más eficiente vendría a ser Intranet y Extranet creando un entorno de trabajo online. Limitación de esta última solución serían los costos de implementación que muchas empresas no podrían justificar esta inversión.

- Hay muchas posibilidades para aumentar una comunicación virtual:
 - Espacio de trabajo virtual compartido.
 - Foro en línea
 - Redes sociales internas en la organización.
 - Sitio de SharePoint para almacenar los archivos del equipo.
 - Programa de mensajería instantánea.
 - Conferencias semanales.
 - Conferencias Web.
 - Salas de conferencias virtuales.
 - Teléfono.
- Moderar las comunicaciones a través de los foros u otras herramientas de comunicación asíncrona (email, wikis y documentos compartidos), de los chats u otras herramientas de

comunicación síncrona textual (Instant Messaging) y de las multi/videoconferencias.

- Las reuniones son claves para tener una comunicación organizada y estructurada, estas se irán afinando conforme avance el proyecto, alcanzando mayor eficiencia en comunicaciones formales.

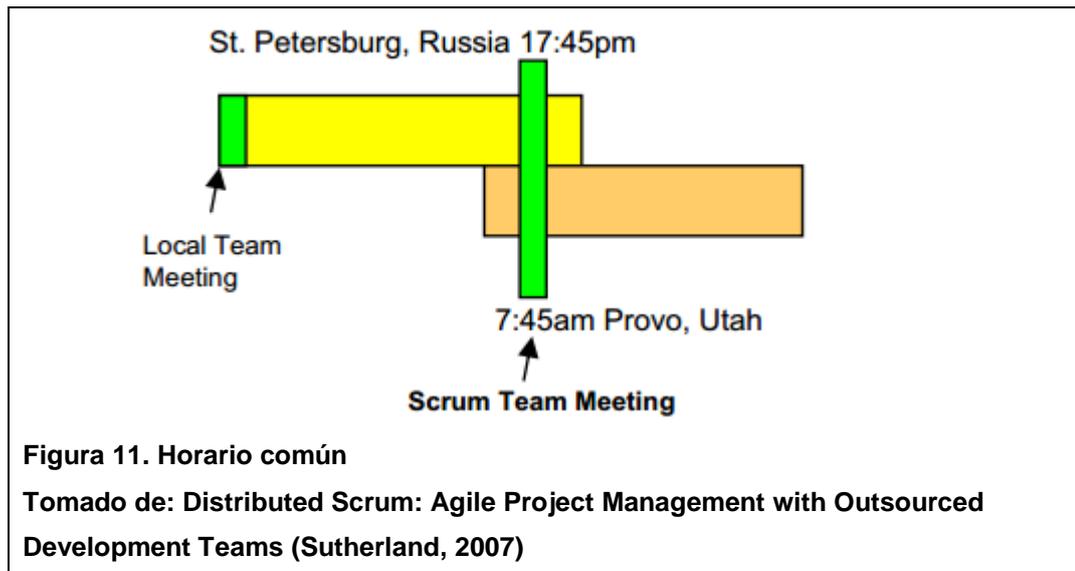
7) Registro de la comunicación

- Etiquetar todas las comunicaciones relacionadas al proyecto en curso, que se realicen entre los integrantes del equipo, a manera de un acta con fecha, tema y participantes.
- Repositorio común para almacenar las comunicaciones, inclusive de manera histórica.
- Los avances tanto individual como total se comunicarán de forma permanente, apoyándose en las herramientas.

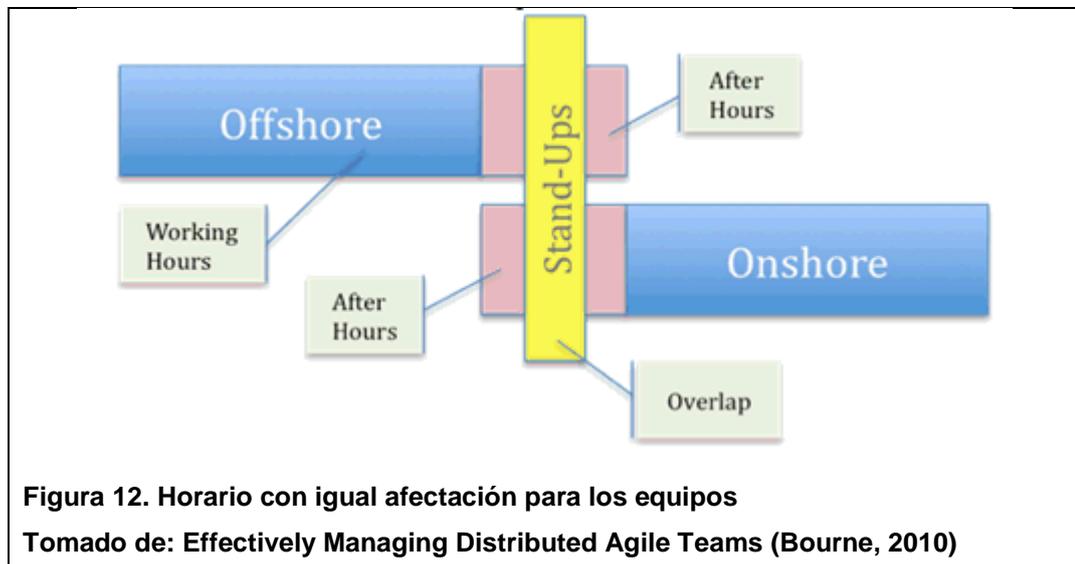
8) Horario común

Con equipos distribuidos es importante establecer un horario común y fijo para realizar las reuniones, de manera que no se perjudique ni al equipo local ni al remoto. La práctica recomendada es que este horario y duración de las reuniones sea fijo

Cuando los equipos están en locaciones geográficas diferentes, separadas por diferentes usos horarios se debe buscar un horario durante las horas laborables en común



De no existir un horario común en horas laborables, se recomienda usar el horario que tenga la misma afectación en ambos equipos, pero en estos casos prima el acuerdo al que se pueda llegar entre ambos equipos



3.3.3 Tecnologías de información

Las herramientas juegan un papel sumamente importante, ya que apoyarán en el proceso, automatizando varios procesos para que el equipo se concentre en lo importante del proyecto.

3.3.3.1 Prácticas para apoyo en la elección de las tecnologías de información

Se sugieren las siguientes prácticas:

- **Realizar un listado de necesidades tecnológicas**
 - Qué tipo de información se transmitirá
 - Cantidad de datos que se transmitirán.
 - Escenario de interacción (1 a N, N a N, 1 a 1)
 - Se requerirá de almacenamiento
 - Tipo de interacción (solo ver o modificar)
 - Calidad mínima necesaria
 - Tipo de conexión a Internet que requiere
 - Movilidad
 - Medios disponibles (ancho de banda de equipos)
 - Infraestructura de salas.
 - Soporte técnico antes y durante las sesiones.
 - Equipos de comunicación confiables

- **Listado de las tareas y servicios**
 - Encuestas
 - Ficheros
 - Foros
 - Blogs
 - Cursos
 - Presentación de proyecto
 - Guardar feedbacks de cada sesión
 - VideoConferencias
 - Mails
 - Pizarras
 - Comunicación Telefónica
 - Encriptación de datos

- Telepresencia, mantener una sesión lo más real posible.
- Despliegue de autotest
- **Revisión de calidad de artefactos a usar**

Hay programas que se pueden usar para realizar la revisión de aseguramiento de la calidad.

 - Audio: verifica eco, calidad de sonido, nivel de ruido.
 - Video: verifica contraste entre fondo y frente, posicionamiento de cámaras, no parpadeo.
 - Red: NTP(Network Time Protocol) Habilitado, unicast, multicast (opcional)
 - Programas compartidos
 - Software: configuraciones
 - Condiciones acústicas.
 - Fluido eléctrico.
- **Capacidades para hardware y software**
 - Contar con equipos de última generación
 - Revisar hardware existente podría soportar lo requerido
 - Internet, buena calidad.
 - Revisar el software requerido, luego hardware
 - Revisar la distribución del equipo, muchas salas con varios integrantes.
 - Red

Las capacidades que se requieran del hardware y software dependerán de la calidad (alta, media, baja) que necesitamos en las tareas y servicios a realizarse.

Tabla 6. Capacidad requerida de hardware y software según calidad esperada

Tareas \ Calidad Esperada	Alta	Media	Baja
Encuestas		X	
Ficheros	X		
Foros		X	
Blogs		X	
Cursos	X		
Presentación de proyecto	X		
Guardar los comentarios y opiniones de cada sesión	X		
VideoConferencias	X		
Mails	X		
Pizarras		X	
Comunicación Telefónica	X		
Encriptación de datos		X	

Tomado de: Autores de la Tesis

- **Dispositivos a usar**

Considera cómo será la interacción y bajo qué circunstancias trabajará el equipo, para tomar en cuenta los dispositivos que el equipo necesitará, como:

- Tabletas
- PDA (Personal Digital Assistant)
- Teléfonos Móviles
- *Helmets*, videoconferencia para ver en 3D a los participantes
- Apuntadores
- Encerados Electrónicos, es una pantalla multimedia, permite la visualización y edición gráfica. Apoya para apuntes informales, *brainstorming*, diseño gráfico.
- Micrófonos con cancelación embebida.

- **Soporte Hardware y Software**
 - Herramientas libres que existen en el mercado.
 - Herramientas licenciadas
 - Consideración de personal técnico que conozca la tecnología para dar soporte (operador + *senior*).

3.3.3.2 Herramientas de colaboración

Las herramientas de colaboración en un pasado eran restringidas por costos, implementación, accesos. En la actualidad con la evolución de la tecnología ha permitido que estos topes desaparezcan, incluso existen paquetes que se los ofrecen de manera gratuita o adicional con algún otro producto que ya use la organización y hoy apoyen en fomentar el trabajo en equipo aumentando la eficiencia y eficacia que provoca el crecimiento de las empresas.

Podríamos mencionar las siguientes:

3.3.3.2.1 Sincrónicas

- **Mensajería instantánea.**

Facilita el envío de cierta información compleja (comandos, partes de código)
- **Audioconferencia.**

Interacción vía audio de dos o más participantes, requiere de dispositivos como teléfonos, equipos de audioconferencia y el software iphone, skype.
- **Videoconferencias H.323**

Sistemas basados en este protocolo, que permiten proveer sesiones de comunicación audiovisual sobre paquetes de red, comunicaciones multimedia, como Office Live Meeting (<http://office.microsoft.com/es-es/live-meeting/>, 2014), usada también para Voz sobre IP (VoIP), Telefonía de Internet, o telefonía IP y para video conferencias basadas en IP. Son sistemas para poder realizar las conferencias muy acercadas a la realidad, sentir que las personas se encuentran en la sala.



Figura 13. Videoconferencia H.323

Fuente: Office Live Meeting (<http://office.microsoft.com/es-es/live-meeting/>, 2014)

- **Videoconferencia**

Posibilita la comunicación entre personas ubicadas en diferentes lugares permitiendo la transmisión de imagen y sonido en tiempo real mediante equipamiento hardware diseñado para esta función integrado en salas comunes. Para establecer la comunicación es necesario que todos los participantes tengan un sistema similar.

- **Entornos colaborativos avanzados**

- **Access Grid**(Universidad Nacional Autónoma de México, 2008)
Es un sistema de videoconferencia avanzado con el que se consigue un entorno para la comunicación dotado de más realismo y se realiza en una sala preparada con la infraestructura, hardware y software configurados con este objetivo. El principal inconveniente de este servicio es que es imprescindible que todos los participantes en la comunicación utilicen el mismo sistema.
- **Conference XP.**
Es una herramienta que facilita desarrollar aplicaciones orientadas a la enseñanza o video conferencias, sin tener que implementar grandes infraestructuras.
- **EVO/RVS**
Plataforma de colaboración basada en web, videoconferencia *multicast*, orientada a comunidades investigadoras.

- **Isabel**

Herramienta de colaboración multipunto sobre PCs, posee un sistema de control de sesiones y salas, es *unicast/multicast*.

- **Webinars**

Es la combinación de las palabras web y seminario, también llamados seminarios web, conferencia web, salones de diálogo sincrónico o reunión a través de Internet. La comunicación se realiza por la combinación de software y hardware. Mediante un navegador web estándar (Mozilla Firefox, Google Chrome, Safari, Internet Explorer, etc.) se transmite la imagen y el sonido usando la cámara y el micrófono conectados al PC. Los usuarios se conectan a una dirección URL (siglas en inglés de Uniform Resource Locator) donde se realizará la interacción.

Una de las maneras más comunes de invitar a la audiencia a una conferencia a través de la web es invitándolos a través de un email. Los invitados reciben un email con un enlace de invitación sobre el que tienes que hacer clic. Ya solo quedará esperar hasta que la conferencia tenga lugar.

Se puede usar para:

- Permite la colaboración sincrónica y distribuida.
- Video\audio unidireccional y secuencial
- Grupos numerosos con ancho de banda limitado
- Presentaciones multimedia
- Escritorio compartido
- Chat y aplicaciones de retroalimentación
- Reuniones de trabajo
- Sesiones de formación
- Demostraciones de programas.

- **Webcast**
Son conferencias donde solo el conferencista habla y los demás escuchan.
- **Herramientas complementarias**
 - Encerado electrónico
 - Compartición de escritorio
 - Retroalimentación inmediata
- **Dispositivos de colaboración**

3.3.3.2.2 Asincrónicas

- Email
- Foros
- *Blogs*
- Intercambio de archivos
- *Streaming*
- Tablón de anuncios
- Calendario

Podemos mencionar algunas herramientas que existen, como: Webex, Breeze Live, Adobe Connect, Webhuddle, Dim Dim, Openmeetings, Elluminate, Big Blue Button, EVO/VRVS.

Hay que notar las características que estas herramientas deberían tener para que cumplan su propósito, entre estas podríamos mencionar las siguientes:

- Fácil de manejar, que sea intuitiva para el usuario, lo que le permitirá adoptarla más rápido.
- Flexible, que permita una organización de acuerdo a las necesidades de la empresa o equipo. Sea esto por proyecto, por evento, etc.
- Más movilidad, permita acceder a los espacios de trabajo comunes o específicos, desde cualquier computadora con acceso a Internet.
- Tareas, que permita la gestión y asignación de tareas a los grupos.

- Visualizar avances, que permitan ir haciendo el seguimiento de las tareas.
- Área de trabajo compartida, permitiendo compartir documentos, notas, incluso de ser requerido videos, mp4. Para verlos en línea o dejarlos para acceso posterior.
- Calendario compartido, podría fusionarse con las herramientas de manejo de mensajería y correo.
- Avisos programados, que permita programar avisos en el correo electrónico, para cuando exista algo nuevo que revisar o se haya realizado modificaciones de los documentos.
- Videoconferencia y Chat, permitiendo la interacción y comunicación eficiente de los equipos en un ambiente exclusivo.
- Muro de anuncios, un lugar en la que las personas puedan describir su perfil, sus habilidades, qué proyectos ha manejado, con qué clientes ha tratado. Esto lograría la sinergia entre integrantes, permitiendo fomentar el sentido incluso de pertenencia con la organización.
- Base de conocimientos, útil como una bitácora de barco, permitiría solucionar problemas que ya pueden haber sido encontrados.
- Herramientas adicionales, podría incluirse encuestas, mapas conceptuales, *dashboards*, *wikis*, *blogs*

3.3.3.3 Herramientas para gestión de proyectos

La gestión de proyectos de software es un proceso continuo que para su adecuada ejecución requiere estar enmarcada dentro de una estrategia empresarial y apoyada por las herramientas que permitan aumentar la productividad. Por lo que, también es importante identificar las herramientas necesarias que nos ayuden a conseguir la mayor productividad posible.

Características sugeridas podrían ser:

- Flexible, permita varios entorno de trabajo (Web, Mobile, Mail)

- Fácil configuración.
- Fácil gestión.
- Apoyen en la estimación de esfuerzos de proyecto y de costes de software.
- Planificación de proyectos.
- Provea métricas específicas del proyecto (Burn down chart o Burn up chart)
- Indicadores de productividad
- Indicadores de calidad
- Seguimientos de tareas
- Tablero de anuncios
- Opciones de colaboración
- Tableros de control y seguimiento (Kanban)

3.3.3.4 Herramientas para diseño de software

Estas herramientas deberían poseer las siguientes características:

- Modelado de sistemas
- Diseño e ingeniería de bases de datos
- Modelado de Datos y Eventos
- Repositorio de definiciones de usuario
- Automatizada, flexible y gráfica, para la construcción rápida de aplicaciones.
- Integradora de herramientas y metodologías.

3.3.3.5 Herramientas de control de versiones

Estas herramientas deberían poseer las siguientes características:

- Almacenamiento seguro de fuentes
- Flexibilidad de almacenamiento (dentro o fuera de la empresa)
- Realice copias de seguridad
- Realice copias completas, incluidos datos

- Sencilla interfaz
- Control de versionamiento.
- Control de accesos
- Perfiles de usuario.

3.3.3.6 Herramientas para pruebas

Los sistemas al ser sometidos a los varios sets de pruebas permiten obtener un producto con más estabilidad, podemos apoyarnos también con herramientas para la gestión de estas pruebas. Las pruebas que podrían mencionarse son de funcionalidad, rendimiento, de carga y de estrés.

Se consideran las pruebas de rendimiento que sirven para predecir cuando los niveles de carga agotarán los recursos del sistema; las de carga, evalúa el rendimiento del sistema con una carga predefinida; pruebas de estrés, evalúan el comportamiento de los sistemas, al ser llevados al límite operacional.

Estas herramientas deberían poseer las siguientes características:

- Flexibilidad.
- Gestión de requisitos.
- Gestión de casos de prueba.
- Gestión de los miembros del equipo.
- Gestión de ambientes
- Permita el ingreso de los criterios de éxito.
- Parametrizable.
- Capacidades para pruebas de estrés
- Trabajar con scripts
- Modularización de pruebas
- Pruebas autoprogramadas.

3.3.3.7 Herramientas para control de calidad

La inspección de código manual es un proceso que lleva demasiado tiempo, sin muchas veces dar los resultados esperados. Ya existen herramientas que apoyarán en esta tarea.

Las características que podríamos considerar son:

- Detección de código duplicado
- Detección de código muerto (variables, parámetros o métodos sin usar)
- Detección de complejidad de métodos (*if* innecesarios)
- Analizador de código en base a reglas predefinidas
- Entorno para evaluación de código.
- Métricas
- Análisis de dependencias
- Generación de test unitarios
- Refactorizaciones

3.3.4 Trazabilidad del proyecto

Una adecuada trazabilidad puede presentar valiosas ventajas durante la gestión de un proyecto de desarrollo de software, especialmente si se quiere determinar si todos los requerimientos han sido satisfechos adecuadamente, que no se realicen implementaciones sin requerimiento y para entender el impacto que podría tener un cambio en un requerimiento del sistema.

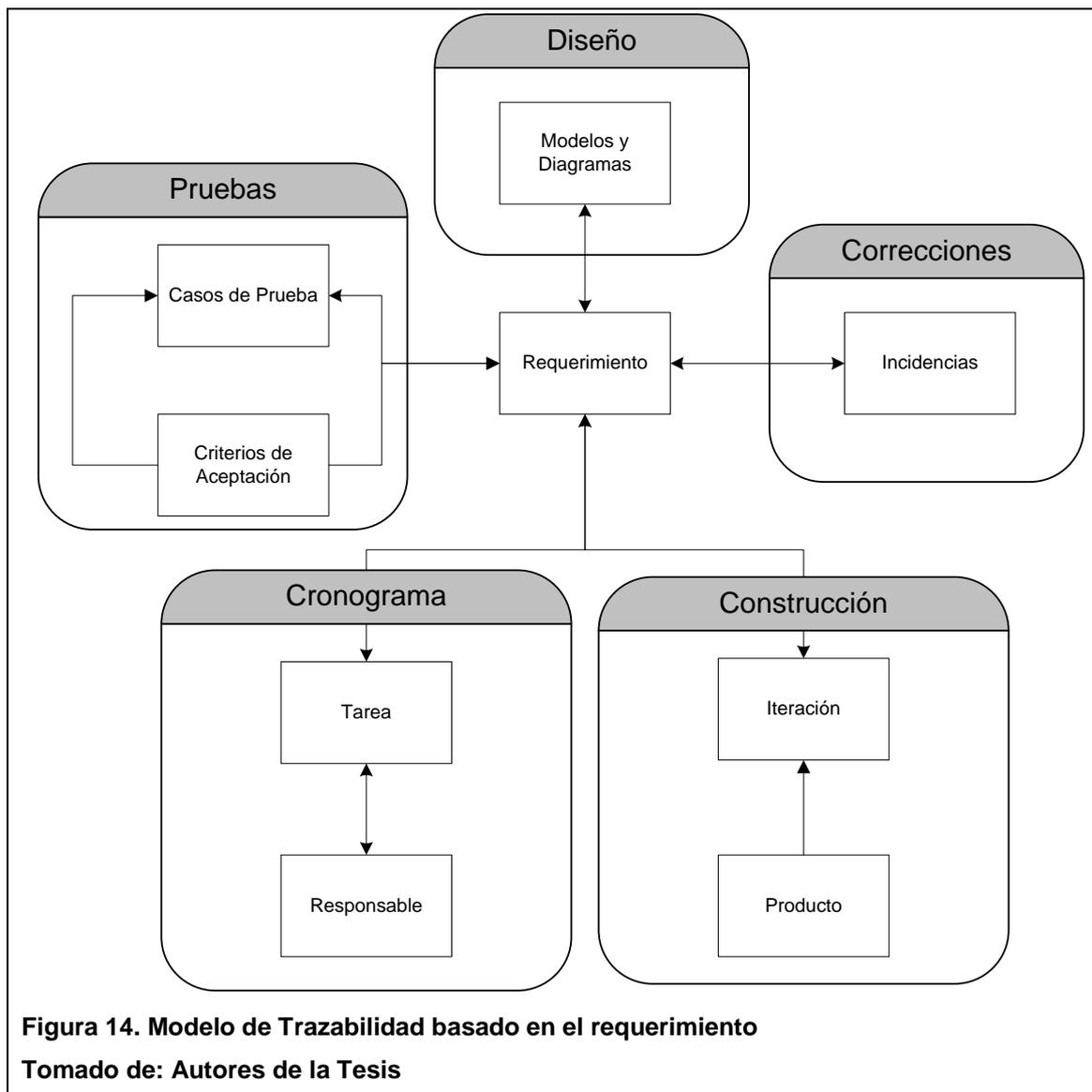
Sin embargo, es importante recordar que mantener la trazabilidad de los requerimientos añade una carga adicional al proyecto, razón por la cual no suele ser muy recomendada por las metodologías ágiles, ya que puede que el esfuerzo sea mayor a los beneficios obtenidos.

Hay que recordar que las metodologías ágiles no necesariamente siguen los caminos convencionales de desarrollo, tal es el caso del desarrollo guiado por

pruebas (TDD), donde del requerimiento primero se desarrollan las pruebas y luego se codifica para satisfacer dichas pruebas.

El tema de la trazabilidad puede ser muy complejo y tener muchas variantes y por ende posibilidades de solución que pueden ser sujetas de un extenso estudio, por lo que, en este marco de gestión se propone realizar una trazabilidad simplificada con el propósito de mantener un adecuado balance entre el costo de realizarla y los beneficios que se obtienen de ella, especialmente en las etapas de mantenimiento o adoptar un software de trazabilidad acorde a la metodología de desarrollo a utilizarse, de modo que la sobrecarga de realizarla sea menor y se aprovechen de mejor manera los beneficios.

Un modelo simplificado basado en el requerimiento sugerido sería el que se muestra en el diagrama a continuación (Ver Figura 14.)



Una Iteración está formada de Requerimientos y estos requerimientos pueden relacionarse entre sí con relaciones de tipo “se reemplaza por”, “derivado de” y “depende de”.

Durante la fase de Planificación se identifican los requerimientos, se los prioriza y se identifica cuáles son resueltos en la Iteración en curso, para lo cual puede que un Requerimiento sea dividido en nuevos Requerimientos más pequeños que se definen como “derivado de”. Luego a este Requerimiento se lo se planifican en Tareas que tienen Responsable y se establecen los Criterios de Aceptación. El Requerimiento para ser construido requiere de Modelos que

definan su forma de implementación y Casos de Prueba, basados en los Criterios de Aceptación, que certifiquen su adecuada implementación. Posteriormente pueden reportarse Incidentes sobre el Requerimiento que deberán ser resueltos

Desde una perspectiva ágil el requerimiento se lo representa desde un inicio en la Historia de Usuario y por tanto ese sería el punto de central de este modelo de trazabilidad.

Lo ideal es contar con una herramienta que automatice el registro de la trazabilidad, pero de no contar con esta se la puede realizar de forma manual mediante una matriz de trazabilidad que puede estar definida en un procesador de texto o en hojas de cálculo.

3.4 Resumen de prácticas por dominio

A continuación se presenta el resumen de las prácticas propuestas por dominio.

Tabla 7. Resumen del dominio: Conformar equipos distribuidos

Dominio 1: Conformar equipos distribuidos	
Prácticas	Fase
P1. Fomentar la unidad	Conceptualización
P2. Construir una identidad de Equipo	Conceptualización
P3. Establecer las reglas para la interacción con el equipo	Conceptualización
P4. Desarrollar los Objetivos como Equipo	Especulación
P5. Esclarecer tareas y procesos, además de objetivos y funciones	Especulación
P6. Entrenamiento al equipo	Especulación
P7. Construir un equipo que marque al mismo ritmo	Especulación
P8. Seguimiento de acuerdos	Revisión
P9. Fomentar el liderazgo compartido	Exploración
P10. Recordar el Uno a Uno	Revisión
P11. Celebrar los logros obtenidos	Cierre

Tomado de: Autores de la Tesis

Tabla 8. Resumen del dominio: Comunicación

Dominio 2: Comunicación	
Prácticas	Fase
P12. Crear un código para facilitar la comunicación	Conceptualización
P13. Acordar el lenguaje a usarse	Conceptualización
P14. Crear un espacio para la interacción informal	Especulación
P15. Comunicación del Líder	Conceptualización Especulación Exploración Revisión Cierre
P16. Comunicación entre miembros del equipo	Conceptualización Especulación Exploración Revisión Cierre
P17. Comunicación virtual	Conceptualización Especulación Exploración Revisión Cierre
P18. Registro de la comunicación	Conceptualización Especulación Exploración Revisión

Tomado de: Autores de la Tesis

Tabla 9. Resumen del dominio: Tecnologías de la información

Dominio 3: Tecnologías de información	
Prácticas	Fase
P19. Realizar un listado de necesidades tecnológicas	Conceptualización
P20. Listado de las tareas	Conceptualización
P21. Calidad de resultados esperados en las tareas a	Exploración

realizarse.	
P22. Revisión de calidad de artefactos a usar.	Especulación
P23. Capacidades para hardware y software	Especulación
P24. Dispositivos a usar	Especulación
P25. Soporte Hardware y Software	Especulación

Tomado de: Autores de la Tesis

Tabla 10. Resumen del dominio: Trazabilidad

Dominio 4: Trazabilidad	
Prácticas	Fase
P26. Requerimiento – Historias de usuario.	Conceptualización
P27. Diseño – Hacer Modelos y Diagramas.	Especulación
P28. Cronograma – Matriz de Tareas y Responsables.	Especulación
P29. Construcción - Interacción y Producto	Exploración
P30. Pruebas – Casos de prueba y criterios de aceptación.	Revisión
P31. Correcciones – Incidencias.	Revisión

Tomado de: Autores de la Tesis

3.4.1 Dominios por fase

En la siguiente matriz se puede observar las prácticas propuestas, enmarcadas en las fases y dominios propuestos.

Tabla 11. Matriz de dominios por fase

DOMINIO FASE	Conformación de Equipos	Comunicación	Tecnologías de Información	Trazabilidad
Conceptualización	P1. Fomentar la unidad P2. Construir una identidad de Equipo P3. Establecer las reglas para la interacción con el equipo	P12. Crear un código para facilitar la comunicación P13. Acordar el lenguaje a usarse P15. Comunicación del Líder P16. Comunicación entre miembros del equipo P17. Comunicación virtual P18. Registro de la comunicación	P19. Realizar un listado de necesidades tecnológicas. P20. Listado de las tareas.	P26. Requerimiento –Historias de usuario.
Especulación	P4. Desarrollar los Objetivos como Equipo P5. Esclarecer tareas y procesos, además de objetivos y funciones. P6. Entrenamiento al equipo P7. Construir un equipo que marque al mismo ritmo	P14. Crear un espacio para la interacción informal P15. Comunicación del Líder P16. Comunicación entre miembros del equipo P17. Comunicación virtual P18. Registro de la comunicación	P22. Revisión de calidad de artefactos a usar. P23. Capacidades para hardware y software. P24. Dispositivos a usar. P25. Soporte Hardware y Software.	P27. Diseño – Hacer Modelos y Diagramas. P28. Cronograma – Matriz de Tareas y Responsables.
Exploración	P9. Fomentar el liderazgo compartido	P15. Comunicación del Líder P16. Comunicación entre miembros del equipo P17. Comunicación virtual P18. Registro de la comunicación	P21. Calidad de resultados esperados en las tareas a realizarse.	P29. Construcción - Interacción y Producto
Revisión	P8. Seguimiento de acuerdos P10. Recordar el Uno a Uno	P15. Comunicación del Líder P16. Comunicación entre miembros del equipo P17. Comunicación virtual P18. Registro de la comunicación		P30. Pruebas – Casos de prueba y criterios de aceptación. P31. Correcciones – Incidencias.
Cierre	P11. Celebrar los logros obtenidos	P15. Comunicación del Líder P16. Comunicación entre miembros del equipo P17. Comunicación virtual P18. Registro de la comunicación		

Tomado de: Autores de la Tesis

4 Capítulo IV: ¿Como una empresa tradicional puede aplicar el marco propuesto?

4.1 Visión General

La aplicación de este marco de trabajo en una empresa tradicional puede inicialmente no ser una tarea fácil, pero como se había mencionado este marco de trabajo se fundamenta principalmente en principios ágiles, por lo que el éxito en su aplicación empieza con la transformación de la empresa hacia la adopción de un marco de gestión ágil. Para empresas que ya tengan adoptado un marco de gestión ágil como por ejemplo Scrum, la adopción de este marco de trabajo es más sencillo y deberán guiarse en la Tabla 11. Matriz de dominios por fase.

Para una mejor comprensión el presente capítulo supondrá que se tiene un marco de gestión tradicional y por lo tanto la transición debe empezar desde una reestructuración empresarial en diferentes aspectos y con el remplazo de la metodología de gestión, para luego aplicar el marco de trabajo propuesto en sus cuatro dominios antes descritos para la gestión de los proyectos de desarrollo de software que vayan a realizarse con equipos geográficamente dispersos.

4.2 Transición a una organización ágil

Cada vez es más común que las empresas adopten prácticas que les permiten ser más ágiles, pero las estructuras funcionales tradicionales se convierten en un impedimento para aprovechar adecuadamente estos cambios.

Las metodologías ágiles requieren de mayor participación por parte de los equipos, de la gente; es por ello que las funciones gerenciales también deben adaptarse y algunas de sus responsabilidades dejan de ser necesarias.

Para una adecuada transición de la metodología de gestión se debe administrar el cambio desde cuatro puntos:

- 1) Alta dirección y liderazgo
- 2) Estructura organizacional
- 3) La gente
- 4) Tecnología

Los cuatro puntos antes mencionados deberán ser trabajados de forma simultánea y el proceso de transición es aconsejable que se lo realice de forma iterativa e incremental hasta lograr con la experiencia acumulada la mejor aproximación a la realidad deseada para la empresa.



Figura 15. Transición de una gestión tradicional a una gestión ágil

Fuente: Autores de la Tesis

4.2.1 Alta dirección y liderazgo

Es fundamental para el cambio tener el apoyo de la alta dirección de la empresa ya que sin ella todo proceso de cambio es difícil que llegue a feliz término y que se mantenga en el tiempo. Sin embargo, contar con apoyo en los diferentes niveles también puede resultar valioso para que estos líderes influyeran a las demás personas en todos y cada uno de los niveles de la empresa.

No contar con ningún tipo del apoyo antes descrito podría indicar que los esfuerzos de transición serán grandes y muy probablemente no se logre ningún cambio.

Para que los equipos puedan aprovechar de mejor manera las metodologías ágiles deben poder asumir más habilidades transversales, es decir poder trascender de su área de especialización, por lo que los directivos deben tener mayor tolerancia a los errores que puedan suceder al intentar nuevas actividades. Este proceso de aprendizaje ayuda mucho para que se logre flexibilidad, agilidad y se refuerce el trabajo en equipo

4.2.2 Estructura organizacional

Una estructura organizacional con muchos niveles de autorización o con muchas capas para llegar a un nivel de decisión requerirá de un gran esfuerzo para lograr la transición.

Una cultura organizacional en que el ambiente es rígido sin espacios abiertos, donde las personas están intencionalmente separadas entre oficinas o con divisiones, también requerirá una gestión del cambio mayor.

Ajustar la estructura organizacional para evitar las situaciones antes descritas permitirá y facilitará la gestión ágil de proyectos de software.

4.2.3 La Gente

Las personas son las que van a realizar el cambio y de ellos depende que este sea un éxito o no. Personas acostumbradas a trabajar aisladamente que no quieren formar parte de un equipo, que no aceptan nuevas ideas y no quieren cambiar su forma de trabajar representan un gran obstáculo. Todo cambio es difícil y tiene cierta resistencia, por lo que es importante formar un grupo de personas que guíen la transición y actúen como motivadores para este cambio, las personas de este grupo deben ser muy entusiastas, sociables, con gran energía y conocimiento del objetivo a alcanzar para que motiven y guíen adecuadamente al resto de equipos en su proceso de cambio

Prácticas que deben realizarse son:

- Entrenamiento para que las personas entiendan en qué consiste la transición.
- Mantener a una o varias personas que tengan el conocimiento para que actúen como entrenadores y puedan solventar dudas de otras personas durante toda la transición y durante al menos dos iteraciones del primer piloto de aplicación del marco.
- Mantener un ritmo sostenible que no agote a las personas y que evite realizar sobreesfuerzos.

4.2.4 Tecnología

Este punto se refiere no solo a la necesidad de establecer las herramientas tecnológicas básicas a utilizar (como ScrumWorks, Jira, Kanban Tool, etc) sino también es necesario tener un adecuado conocimiento del marco de trabajo a adoptar (ej. Scrum más el marco aquí propuesto). Es decir que es necesario tener claro los procedimientos y herramientas que van a apoyar durante la gestión ágil de un proyecto. Como estas dependen de las diferentes necesidades, en este punto, solo se pide tener una visión general ya que es toda una dimensión a tratar dentro del marco propuesto

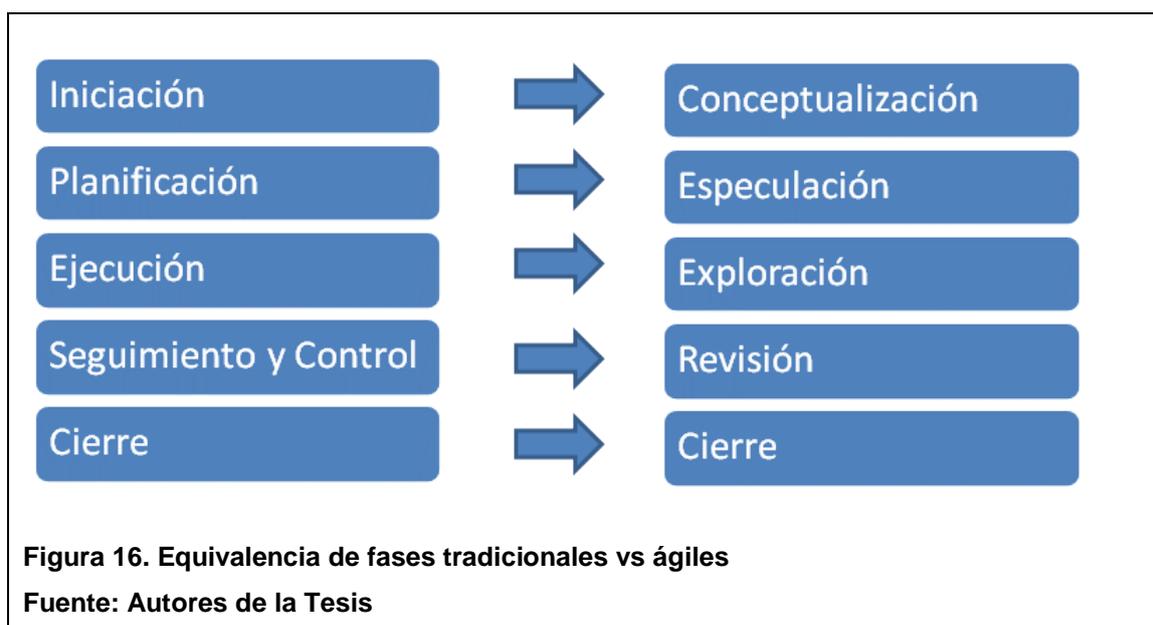
4.3 Transición de una gestión tradicional a una gestión ágil

Uno de los marcos tradicionales de gestión más utilizado es el PMBOOK, por lo que se lo tomará como referencia para describir los cambios necesarios en cada una de sus áreas de conocimiento.

4.3.1 Equivalencia de las fases del proyecto

Las fases tradicionales pueden parecer muy similares a las fases de un marco de gestión ágil, ya que en ambos casos son cinco, pero la diferencia radica en que un marco ágil se desarrolla de forma iterativa e incremental, utilizando iteraciones cortas, alcances acotados y por ende las actividades de cada fase son reducidas.

La equivalencia de las fases se muestra en la Figura 16. Equivalencia de fases tradicionales vs ágiles



4.3.2 Gestión de la integración del Proyecto

En esta área de conocimiento se establece la visión del producto y se definen las expectativas. Las decisiones se las toma de forma participativa y la planificación se la realiza en forma iterativa, no todos los elementos se definen desde el inicio, la intención es elaborar planes de liberación que indiquen las

fechas esperadas con las funcionalidades requeridas para cada iteración, considerando el esfuerzo necesario para su implementación dentro del tiempo establecido.

El equipo participa en la creación del plan y se compromete a cumplirlo.

Para equipos geográficamente dispersos se requerirá de herramientas para comunicarse y compartir esta información como lo son portales, wikis, o herramientas de planificación ágil.

El control de cambios es dinámico y se lo gestiona diariamente a través de la lista de requerimientos, buscando siempre priorizar las que dan mayor valor al negocio, de manera que los requerimientos de más alta prioridad son los primeros en construirse en las iteraciones.

Al final de cada iteración, se hace una presentación del producto mediante la cual se obtiene retroalimentación que puede modificar la lista de requerimientos, ayudar a mejorar el proceso y corregir el producto. Las decisiones se toman en equipo de forma rápida con poco o ningún protocolo.

Una herramienta que nos permite visualizar y controlar fácilmente el avance y cuellos de botella del proyecto es Kanban, el cual es un tablero que utiliza tarjetas para representar las tareas y permite visualizar en qué estado se encuentran durante el proyecto.

4.3.3 Gestión del alcance y tiempo del proyecto

De forma tradicional la gestión de alcance y la gestión de tiempo representan un verdadero reto ya que los requisitos suelen cambiar en el tiempo, por lo que la planificación, definición, verificación y control son muy importantes.

La gestión ágil tiene otro enfoque, aunque también se considera importante la gestión de alcance y la gestión de tiempo, no se intenta ni prever ni evitar los cambios, se intenta que el costo y el tiempo sean fijos buscando implementar los requerimientos de mayor valor para el negocio.

En el tiempo la lista de requerimientos puede ir variando, pero durante cada iteración los requerimientos definidos para su implementación son fijos e invariables.

Técnicas para estimación del tiempo comúnmente usadas son:

- Por analogía
Requerimientos del mismo tamaño se estimarán en tiempos iguales.
- Descomposición
Se descomponen los requerimientos para estimarlos más fácilmente.
- Planning Poker
Técnica basada en consenso de los miembros del equipo.

Una vez priorizados los requerimientos y estimado su duración, se sabrá el número de programadores necesarios para cumplir con el tiempo planificado. Las herramientas que ayudan a gestionar el costo y tiempo durante la ejecución del proyecto son el *Burn down chart* ó *Burn up chart* (ver anexo 3)

4.3.4 Gestión de costos del proyecto

Una práctica común es medirlo como el número de recursos necesarios (programadores y número de horas por programador), por lo que está muy ligado a lo antes definido en la gestión de alcance y tiempo.

Es importante recordar que para la gestión ágil, el alcance es variable por lo que el tiempo y costo deben permanecer fijos.

4.3.5 Gestión de calidad del proyecto

La gestión de calidad se la realiza desde el análisis y diseño del producto, ya que los requerimientos deben establecer condiciones de aceptación y al ser el desarrollo iterativo, su control es mucho más sencillo que al esperar la finalización de un proyecto muy grande. El control de calidad se busca automatizarlo en la mayor medida posible, para no basarse únicamente en pruebas funcionales. Existen técnicas ágiles de desarrollo, que empiezan por las pruebas como es el caso de TDD (Test Driven Development).

Todo el equipo participa en la definición de las herramientas que se utilizarán para definir y ejecutar las pruebas, lo cual es valioso ya que los programadores son los que escriben las pruebas unitarias y buscan automatizar las pruebas de regresión.

Una gestión ágil de la calidad busca prevenir los defectos por lo que como parte del equipo se incluyen testers, que al ser parte desde un inicio del equipo tienen mayor capacidad para definir las pruebas necesarias a realizarse.

4.3.6 Gestión de RRHH del proyecto

Las metodologías ágiles promueven la conformación de equipos multidisciplinarios para que con cada iteración se obtenga un producto operativo. Lo que significa que el equipo realizará no sólo la programación del producto, sino su diseño funcional, arquitectura y control de calidad.

En un inicio cada integrante del equipo posee conocimiento y habilidades individuales pero en la medida que el proyecto avanza los integrantes aprenden de los otros, permitiéndoles ayudar y participar en tareas en las que inicialmente no tenían experiencia. Un equipo multidisciplinario requiere de personas comprometidas, capaces de auto-organizarse para concluir exitosamente cada iteración.

Esta autonomía del equipo promueve un mejor desempeño y el deseo de mejorar continuamente, pero también puede ser difícil para los gerentes de proyecto ya que no realizan las tareas habituales de administración y al contrario se convierte en un facilitador que busca mantener al equipo alejado de cualquier interrupción y que debe ayudar al equipo a colaborar y tomar decisiones en grupo.

Para entender los requerimientos y la forma en que debe conformarse un equipo para trabajar de forma ágil y geográficamente dispersos se debe referir al dominio Conformación de Equipos del marco propuesto.

4.3.7 Gestión de las comunicaciones del proyecto

La gestión de las comunicaciones esta descrita en el dominio Comunicación del marco propuesto.

4.3.8 Gestión de riesgos del proyecto

Debido a que la metodología de gestión ágil es iterativa con periodos de corta duración, el riesgo es menor y se gestiona con reuniones diarias cortas y de forma más extensa durante la planificación de cada iteración y en las reuniones de revisión.

Todo el equipo participa en la identificación de riesgos durante cada iteración y realizan un análisis cualitativo basado en su experiencia, como los ciclos de desarrollo son cortos este tipo de análisis es eficaz. También todo el equipo es el encargado de definir las acciones que reduzcan las amenazas y de discutir las estrategias de control y monitoreo.

Un riesgo importante de considerar es el trabajar con equipos geográficamente dispersos razón por la cual se debe hacer observancia del marco aquí propuesto.

4.3.9 Gestión de stakeholders del proyecto

Los stakeholders según la metodología ágil están representados por un dueño del producto, el cual es parte esencial en el equipo, ya que su inclusión permite una estructura horizontal auto-organizada y capaz de auto-gestionarse.

La forma iterativa y de ciclos cortos en que se realiza el proyecto da un mayor control y transparencia a los stakeholders sobre el proyecto. El seguimiento es más cercano y se minimiza el riesgo de inconformidad del cliente con el producto.

4.4 Resumen del marco de trabajo

Para la aplicación del marco de trabajo se recomienda seguir los siguientes pasos:

- 1) Transicionar a una organización ágil

- 2) Transicionar a un marco de gestión ágil
- 3) Aplicar el marco de trabajo propuesto en la gestión de proyectos de desarrollo de software con equipos geográficamente dispersos

A continuación se resumen cada uno de los pasos anteriores

4.4.1 Transicionar a una organización ágil

- Actividades de la alta dirección y liderazgo
 - Brindar apoyo a los directivos de la empresa que participan en este proceso.
 - Respaldar la ejecución del proceso en todos los niveles de la organización.
 - Conformar equipos con habilidades transversales
 - Tener mayor tolerancias de los errores por parte de los directivos.
- Cambio en la estructura organizacional
 - Reducir los niveles de autorización
 - Fomentar un ambiente más amigable, con espacios abiertos.
- Actividades relacionadas con el personal
 - Identificar a las personas con habilidades para trabajar en equipo, que acepten retos y nuevas ideas, entusiastas, sociables y con gran energía.
 - Con las personas identificadas, conformar un grupo que guíe la transición.
 - Comunicar claramente los objetivos planteados.
 - Difundir los objetivos a toda la organización.
 - El equipo seleccionado actuará como entrenador para todos los equipos conformados, solventando toda duda durante la transición, al menos durante dos iteraciones del primero piloto de aplicación del marco.
 - Mantener un ritmo sostenible que no agote a las personas.
- Tecnología
 - Establecer las herramientas tecnológicas a usar

- Tener claro los procedimientos y herramientas que apoyarán en el proceso.
- Tener conocimiento del marco de trabajo a usar y la tecnología recomendada.

4.4.2 Transicionar a un marco de gestión ágil

- Identificar las similitudes y diferencias de las fases del proyecto
- Transicionar cada una de las áreas de conocimiento del marco tradicional.
- Aplicar el marco de trabajo propuesto en cada una de las fases del proyecto.
 - Conceptualización
 - Conformar los equipos distribuidos
 - Fomentar la unidad
 - Construir identidad de equipo
 - Establecer las reglas para la interacción
 - Comunicación
 - Crear un código de comunicación
 - Establecer el lenguaje común
 - Seguir las recomendaciones de la comunicación de líder.
 - Seguir las recomendaciones para la comunicación entre los miembros del equipo.
 - Establecer las reglas de la comunicación virtual.
 - Registrar las comunicaciones.
 - Tecnologías de información
 - Realizar un listado de necesidades tecnológicas.
 - Realizar un listado de las tareas a ejecutarse.
 - Trazabilidad
 - Registrar las historias de usuario.
 - Especificación
 - Conformar los equipos distribuidos

- Desarrollar objetivos como equipo.
- Establecer tareas, procesos, objetivos y funciones.
- Entrenar al equipo
- Construir un equipo que marque al mismo ritmo.
- Comunicación
 - Crear un espacio para la interacción informal.
 - Seguir las recomendaciones de la comunicación de líder.
 - Seguir las recomendaciones para la comunicación entre los miembros del equipo.
 - Establecer las reglas de la comunicación virtual.
 - Registrar las comunicaciones.
- Tecnologías de información
 - Revisar la calidad de los artefactos a usar.
 - Planificar las capacidades de hardware y software.
 - Establecer dispositivos a usar
 - Planificar soporte de hardware y software.
- Trazabilidad
 - Registrar modelos y diagramas
 - Crear matriz de tareas y responsabilidades.
- Exploración
 - Conformar los equipos distribuidos
 - Fomentar el liderazgo compartido
 - Comunicación
 - Seguir las recomendaciones de la comunicación de líder.
 - Seguir las recomendaciones para la comunicación entre los miembros del equipo.
 - Establecer las reglas de la comunicación virtual.
 - Registrar las comunicaciones.
 - Tecnologías de información

- Realizar revisión de aseguramiento de la calidad de los artefactos.
- Trazabilidad
 - Registrar elementos generados en la interacción.
- Revisión
 - Conformar los equipos distribuidos
 - Realizar seguimiento de acuerdos.
 - Recordar el uno a uno
 - Comunicación
 - Seguir las recomendaciones de la comunicación de líder.
 - Seguir las recomendaciones para la comunicación entre los miembros del equipo.
 - Establecer las reglas de la comunicación virtual.
 - Registrar las comunicaciones.
 - Trazabilidad
 - Registro de casos de prueba y criterios de aceptación.
 - Registro de correcciones e incidencias.
- Cierre
 - Conformar los equipos distribuidos
 - Celebrar los logros obtenidos
 - Comunicación
 - Seguir las recomendaciones de la comunicación de líder.
 - Seguir las recomendaciones para la comunicación entre los miembros del equipo.
 - Establecer las reglas de la comunicación virtual.

4.5 Futuras líneas de investigación

A partir de esta tesis se pueden generar nuevos temas que al no ser parte del alcance o debido a su alta complejidad requieren de investigaciones más profundas, los que se han identificado son:

- Desarrollar una metodología para desarrollo de software con equipos geográficamente dispersos basada en el marco aquí propuesto
- Trazabilidad en proyectos de desarrollo de software con metodología ágil
- Incorporar en este marco de trabajo Gestión de Calidad
- Metodologías de selección de personal basadas en aspectos psicológicos de actitud y aptitud para proyectos de desarrollo de software con metodologías ágiles.

5 Capítulo V: Conclusiones y Recomendaciones

5.1 Conclusiones

A partir del estudio realizado se observa la importancia de contar con un marco de gestión para los proyectos de desarrollo de software aplicable a entornos de trabajo con equipos distribuidos geográficamente, de lo que se obtienen las siguientes conclusiones:

- Los equipos que se conforman en esta modalidad de trabajo distribuido, deben ser inicialmente profesionales de alto nivel, comprometidos, con experiencia en trabajo por objetivos, para que su nivel de desempeño sea aceptable y pueda apoyar al equipo y a la formación de nuevos integrantes.
- El trabajo de líder es pieza clave en la gestión, debido a la dinámica que se maneja, sin permitirse dilatar demasiado los conflictos que podrían afectar gravemente al proyecto.
- Las herramientas de comunicación son indispensables para apoyar las gestiones de estos equipos, sacar el máximo provecho hará que se obtengan excelentes resultados.
- Al eliminarse las distancias y conseguir expertos en cualquier lugar del mundo permite poder realizar proyectos complejos apoyándose de una buena gestión.
- El desarrollo de las tecnologías de información ha permitido que esta propuesta pueda ser válida, ya que los límites establecidos se han minimizado o han desaparecido.
- El marco de gestión propuesto es lo bastante flexible como para usarlo como se lo describe o para integrarse a otros marcos de trabajo complementándolo
- El análisis y comparación de las diferentes metodologías sirvió principalmente para entender sus similitudes y diferencias, resaltando la

necesidad de la realización de este marco de trabajo para gestión de proyectos de desarrollo de software con equipos distribuidos.

5.2 Recomendaciones

- Para empezar un proceso de transformación de un proceso tradicional a uno ágil se recomienda formar un equipo de personas que además de tener las habilidades y conocimientos se sientan comprometidas con esta transformación
- No iniciar un proyecto de desarrollo con equipos distribuidos únicamente motivado por un ahorro de costos ya que existen costos ocultos como se describe en el capítulo 3.3.1 Conformación de equipos
- Es altamente recomendable contar de inicio con herramientas tecnológicas adecuadas para emprender un proyecto de desarrollo de software distribuido
- Para que una metodología de gestión ágil funcione correctamente es necesario mantener una excelente comunicación entre el equipo
- Registrar la trazabilidad de un proyecto de desarrollo es un trabajo laborioso, por lo que lo más recomendado es usar una herramienta tecnológica que permita simplificarlo
- Es recomendable que todos los integrantes de un equipo de desarrollo geográficamente disperso se conozcan de forma personal al inicio de un proyecto ya que la confianza es importante para que trabajen coordinadamente y difícilmente se logra entre personas que solo se conocen virtualmente
- Cuando se trabaja de forma geográficamente dispersa se debe hacer un mayor esfuerzo por explicar claramente los detalles, ya que con las reuniones virtuales es fácil perderselos y dar lugar a malos entendidos
- Realizar actas de cada reunión y entregárselas a los participantes es muy útil con equipos dispersos para garantizar un correcto entendimiento de los temas tratados
- El marco propuesto debe ser utilizado cuando se vayan a ejecutar proyectos de desarrollo de software con equipos geográficamente

dispersos de manera que se consiga mitigar los riesgos inherentes a este tipo de proyectos mediante el uso de todas las recomendaciones y actividades mencionadas

- Para la conformación de equipos debe tomarse muy en cuenta los motivantes del proyecto ya que es el tipo de interacción que tendrán los equipos local y remoto lo que determina el tipo de equipo que debe formarse, siendo siempre la recomendación buscar la mayor integración entre los miembros ya sea que estén trabajando local o remotamente y procurando lograr un equilibrio entre las habilidades de ambos equipos y un trato igualitario en cuanto a derechos y beneficios
- El presente trabajo es un marco de trabajo para un grupo muy específico de gestión de proyectos de software, como lo es en el caso de equipos geográficamente dispersos, pero como tal es importante implementarlo y llevarlo a nivel de una metodología que se ajuste a la realidad de la empresa
- La selección de personal es un aspecto que debe ser consultado en lo posible con los equipos de trabajo que van a acoger al nuevo integrante ya que una buena actitud, aptitud y afinidad pueden ayudar a una rápida integración
- Muchas de las actividades planteadas por el presente trabajo pueden ser tomadas en cuenta como recomendaciones para fomentar la unidad y mejorar la comunicación de los equipos de desarrollo de software en general
- La gestión de calidad es un tema extenso que no ha sido tomado en cuenta en este trabajo por lo que deberá ser analizada y tomada en cuenta en cada proyecto según su necesidad

REFERENCIAS

- AESOFT. (2011). *Estudio de Mercado del Sector de Software y Hardware del Ecuador*.
- Ambler, S. W. (19 de 8 de 2002). *Agile Modeling*. Recuperado el 23 de 11 de 2013, de <http://www.agilemodeling.com/shared/AMPamphletSpanish.pdf>
- Ambler, S. W. (2011). *Agile Modeling*. Recuperado el 1013 de 11 de 23, de <http://agilemodeling.com/>
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). <http://www.agilemanifesto.org/iso/es/>.
- Belzunagui, Á. (2001). *Diversificación de las condiciones de trabajo y cambios organizativos en las empresas: un estudio sobre el teletrabajo. Tesis de doctorado*. Barcelona.
- Bourne, G. (26 de 3 de 2010). *Effectively Managing Distributed Agile Teams*. Obtenido de Dr Dobbs: <http://www.drdoobs.com/architecture-and-design/effectively-managing-distributed-agile-t/224200459>
- Christensen, J. (2010). Team Bonding with a Virtual Workforce. *Hypergrid Business*.
- Conway, M. E. (1968). How Do Committees Invent? *Datamation magazine*.
- Deemer, P. (2012). The Distributed Scrum Primer. En *The Distributed Scrum Primer*. Bangalore: GoodAgile.
- DSDM CONSORTIUM. (s.f.). [dsdm.org](http://www.dsdm.org). Obtenido de <http://www.dsdm.org/dig-deeper/book/dsdm-atern-handbook>
- European Commission. (4 de 11 de 2013). *Digital agenda for Europe*. Obtenido de <https://ec.europa.eu/digital-agenda/en/content/%C2%BFqu%C3%A9-deber%C3%ADa-hacer-espaa%C3%B1a-gobierno-empresas-ciudadanos-para-crear-empleo-en-sector-tic-o-en>

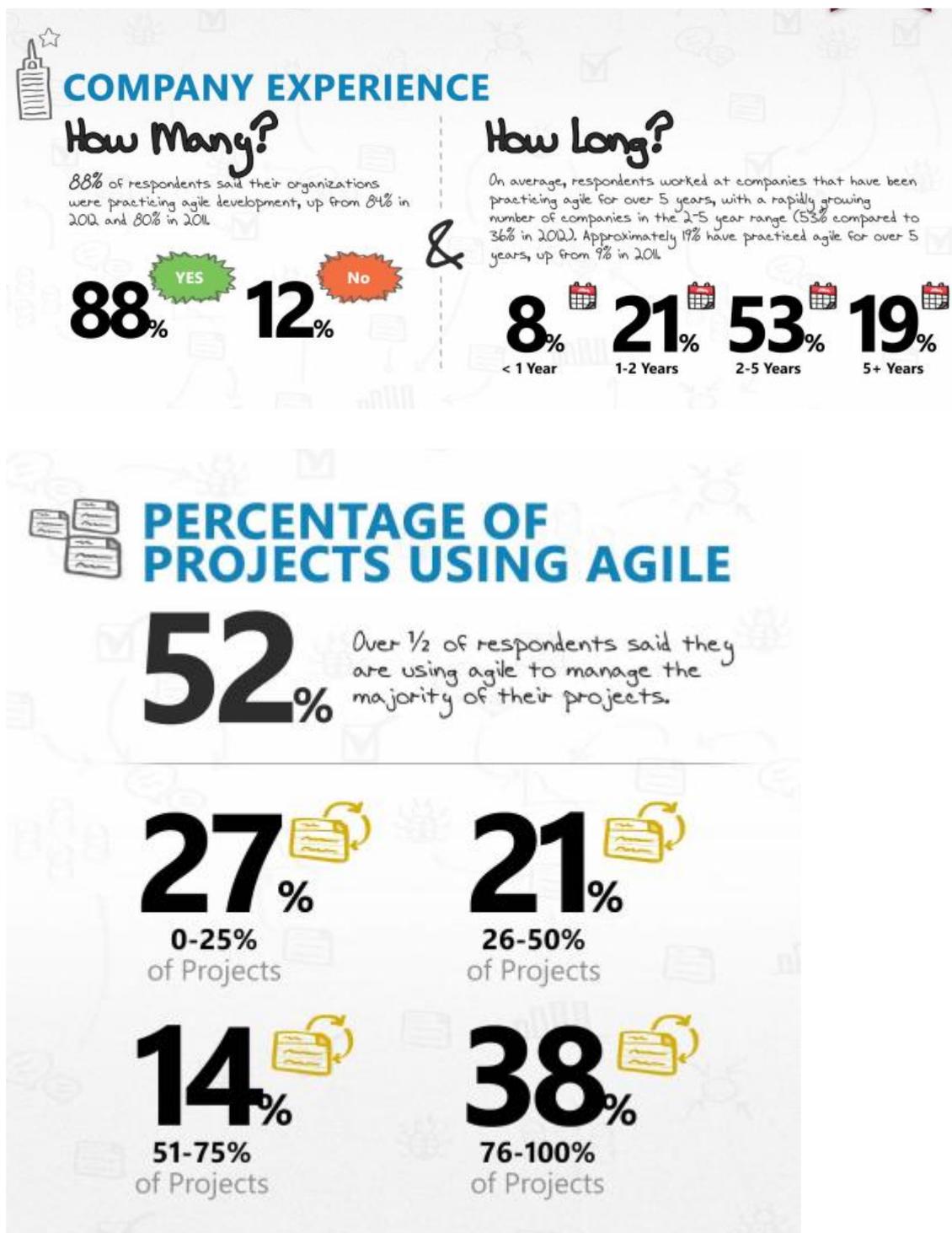
- Foro de empresas efr. (2012). *El libro Blanco del teletrabajo en España*.
- Gotel, O., & Finkelstein, A. (1994). Analysis of the Requirements Traceability Problem. *First International Conference on Requirements Engineering*. Colorado Springs.
- <http://office.microsoft.com/es-es/live-meeting/>. (2014). Obtenido de <http://office.microsoft.com/es-es/live-meeting/>.
- IDC. (19 de junio de 2013). *Worldwide Mobile Worker Population 2011-2015 Forecast*. Obtenido de BusinessWire: <http://www.businesswire.com/news/home/20120105005455/en/Mobile-Worker-Population-Reach-1.3-Billion-2015#.Uwp8u42YaM8>
- JALA International. (15 de 10 de 2013). *JALA International*. Obtenido de <http://jala.com/definitions.php>
- Kniberg, H., & Skarin, M. (2010). *Kanban y Scrum – obteniendo lo mejor de ambos*. C4Media.
- Levit, A. (2012). The 7 Habits of Highly Effective Virtual Teams. *OPENforum*.
- Microsoft. (s.f.). Obtenido de <http://msdn.microsoft.com/es-es/library/jj161047.aspx>
- Padilla, A. (20 de 10 de 2013). *educaweb.com*. Obtenido de <http://www.educaweb.com/esp/servicios/monografico/teletrabajo/1201207-a.html>
- Poppendieck, M., & Poppendieck, T. (2006). *Implementing Lean Software Development: From Concept to Cash (The Addison-Wesley Signature Series)*. Addison-Wesley Professional.
- Rubin, K. S. (2013). *Essential Scrum*. The Addison-Wesley Signature Series.
- Sánchez, M. (2012). *El teletrabajo como tendencia laboral*. Quito: Abya-Yala/UPS.

- Schwaber, K., & Sutherland, J. (2013). *La Guía de Scrum*. Scrum.org.
- SEI. (1 de 02 de 2014). *Software Engineering Institute*. Obtenido de <http://www.sei.cmu.edu>
- Shea, V. (2012). <http://www.eduteka.org/Netiqueta.php3>.
- Sutherland, J. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams. *40th Hawaii International Conference on System Sciences*.
- The Standish Group International. (2013). *Chaos Manifesto 2013*.
- Universidad Nacional Autónoma de México. (2008). Obtenido de <http://www.enterate.unam.mx/Articulos/2005/septiembre/access.htm>.
- VersionOne. (2014). *8th Annual State of Agile Survey*.
- Watkins, M. (27 de 6 de 2013). *Making Virtual Teams Work: Ten Basic Principles*. Obtenido de Harvard Business Review: <http://blogs.hbr.org/2013/06/making-virtual-teams-work-ten/>

ANEXOS

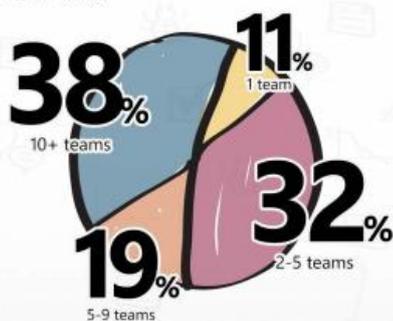
ANEXO 1

Estadísticas destacadas del documento State of Agile (VersionOne, 2014) donde se puede observar la preferencia a utilizar metodologías ágiles



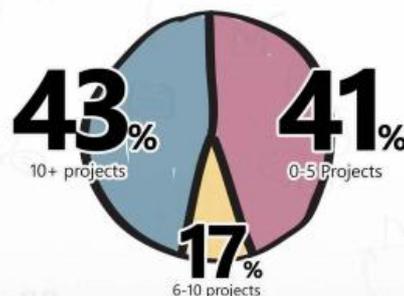
HOW MANY TEAMS ADOPTED AGILE?

The number of teams practicing agile at each organization continues to grow, and this year there is evidence that it's growing fastest at the enterprise level. 57% of respondents said their companies had adopted agile practices across 5 or more teams. This number has nearly doubled in the last 2 years (48% in 2012 and 33% in 2011). The fastest-growing group is those who practice agile in 10 or more teams (38% of respondents), which is an 8% increase over 2012.



NUMBER OF COMPANY PROJECTS USING AGILE

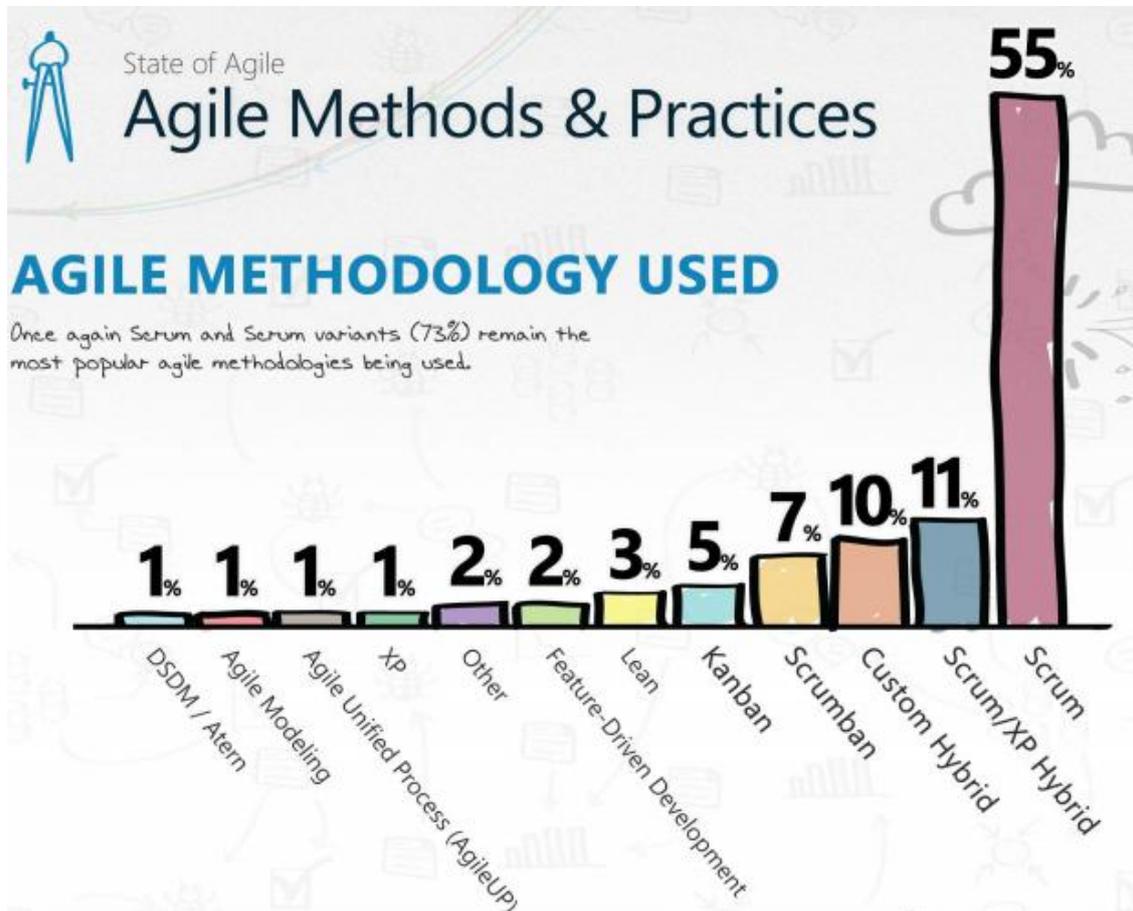
In addition to scaling agile across more teams, organizations are also scaling agile to a greater number of projects. The percentage of respondents with 10 or more agile projects has jumped from 30% in 2012 to 43% in 2013. The trend has shifted from most having 0-5 agile projects last year to heaviest representation in the 10-or-more projects group.



NUMBER OF DISTRIBUTED TEAMS USING AGILE

The number of respondents who have distributed software teams practicing agile has more than doubled in one year. 76% had distributed software teams in 2013 compared to only 35% in 2012.

Estadísticas que muestran las metodologías ágiles más utilizadas

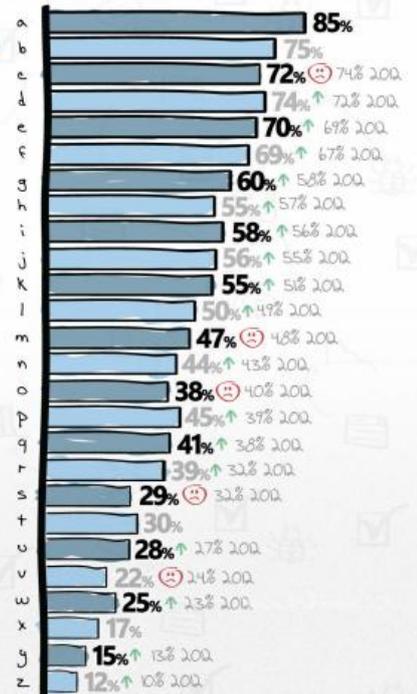


AGILE TECHNIQUES EMPLOYED

Respondents are making use of a wide variety of different agile management techniques. More than 85% practice Daily Standups, 3/4 are using Iteration Planning and Retrospectives, and nearly the same proportion said they maintain Burndown charts. Over the last 2 years we've seen a 10% increase in the use of Retrospectives (from 64% in 2011 to 74% in 2013).

*Respondents were able to select multiple options.

- a Daily Standup
- b Iteration Planning
- c Unit Testing
- d Retrospectives
- e Release Planning
- f Burndown/ Team-Based Estimation
- g Velocity
- h Coding Standards
- i Continuous Integration
- j Automated Builds
- k Dedicated Product Owner
- l Integrated Dev/QA
- m Refactoring
- n Open Workarea
- o TDD
- p Digital Taskboard
- q Story Mapping
- r Kanban
- s Collective Code Ownership
- t Pair Programming
- u Automated Acceptance Testing
- v Analog Taskboard
- w Continuous Deployment
- x Agile Games
- y Cycle Time
- z BDD



ANEXO 2

Estadísticas destacadas del documento The Chaos Manifesto (The Standish Group International, 2013) que muestran un ligero aumento en los proyectos de software que terminan con éxito, aunque el porcentaje sigue siendo menos de la mitad de los proyectos realizados

RESOLUTION					
	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

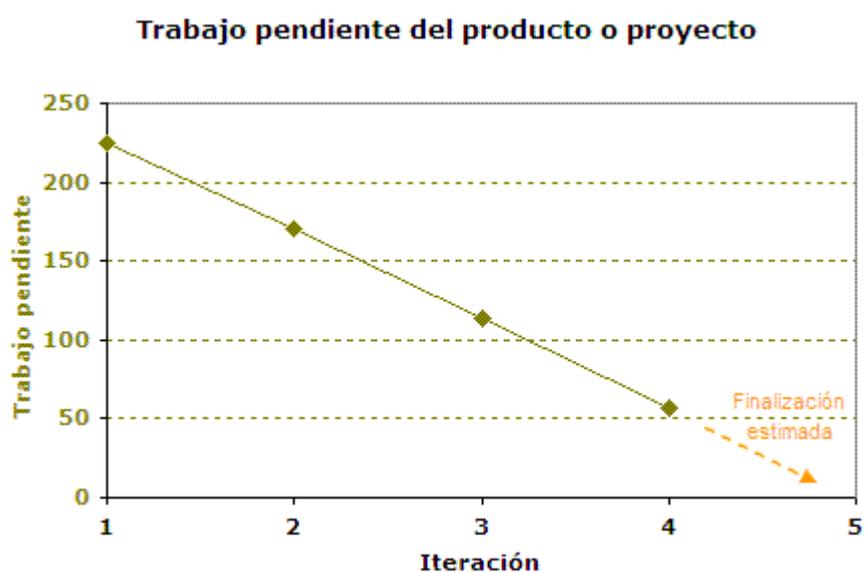
Factors of Success	Points
Executive management support	20
User involvement	15
Optimization	15
Skilled resources	13
Project management expertise	12
Agile process	10
Clear business objectives	6
Emotional maturity	5
Execution	3
Tools and infrastructure	1

ANEXO 3

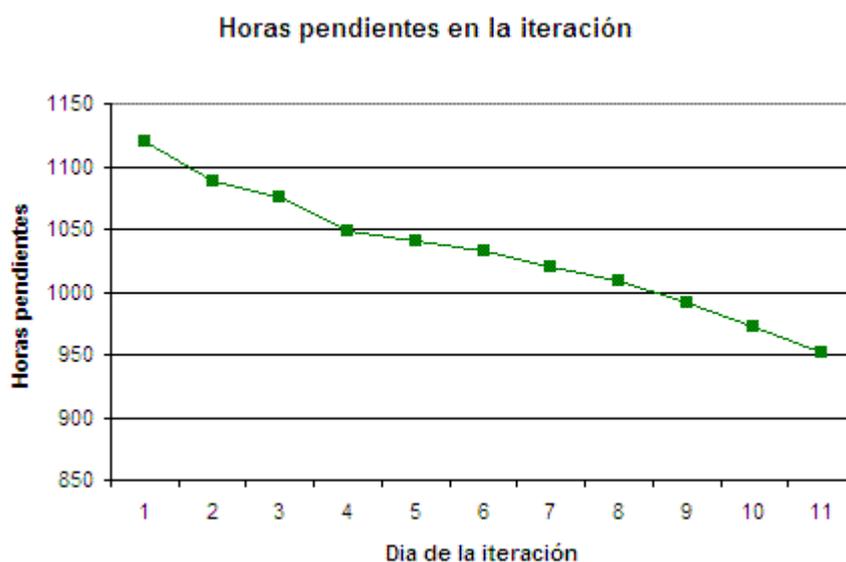
Descripción y ejemplos de algunas herramientas utilizadas en metodologías ágiles.

Burn down chart

Es una representación gráfica del trabajo que se realiza en un proyecto durante el tiempo. Permite estimar si se podrá realizar el trabajo en el tiempo deseado



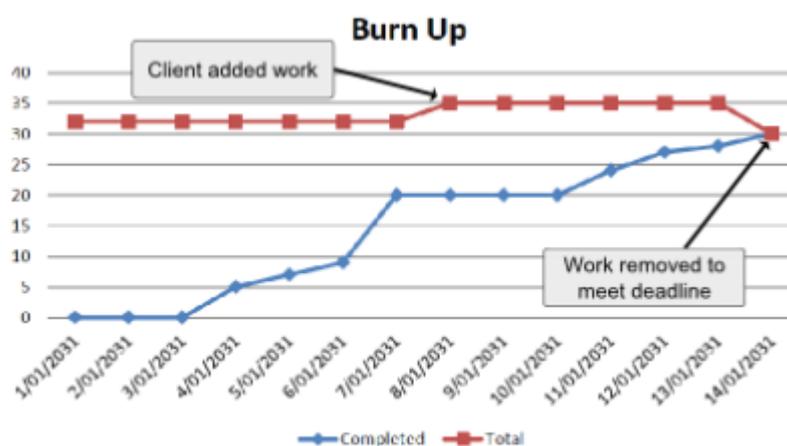
Fuente: <http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>



Fuente: <http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>

Burn up chart

Al igual que el *burn down chart* este gráfico permite visualizar el trabajo pendiente pero en vez de ver como este disminuye se visualiza su avance en contraste con la línea superior que representa el total de trabajo por realizar



A burnup chart clearly shows both completed work and project scope. The project will be completed when the lines meet.

Fuente: <http://www.proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts>

Brown bag

Se refiere a una sesión generalmente de entrenamiento que se la realiza durante la hora de almuerzo. Su nombre simboliza la bolsa de papel que es utilizada por los asistentes para llevar el almuerzo.

Una sesión suele durar una o dos horas y el objetivo es proporcionar información a los asistentes en un ambiente informal

ANEXO 4

Datos útiles del libro Kanban vs Scrum (Kniberg & Skarin, 2010) que nos permiten comparar diferentes metodologías

