

UNIVERSIDAD LAICA ELOY ALFARO DE MANABI
FACULTAD DE CIENCIAS INFORMATICAS
CARRERA TECNOLOGIA DE LA INFORMACION

INGENIERIA DE SOFTWARE

TEMA PROPUESTO:

ANÁLISIS DE METODOLOGÍAS TRADICIONALES Y ÁGILES

INTEGRANTES:

- Arteaga Zambrano Bryan David. ---5 “B”
- Macías Pico Josselyn Stefany. ---5 “B”
- Martínez Anchundia Javier Bernardo ---5 “A”
- Rivera Alvarado Rosa Guadalupe. ---5 “B”
- Villamar Piloso Dayana Liseth. ---5 “B”

CURSO:

Quinto

DOCENTE:

Ing. Alex Santamaria

FECHA ENTREGA:

20/12/2020

MANTA-MANABI-ECUADOR

INDICE

Propósito del documento	5
Fases para un proyecto	5
Metodologías tradicionales	6
Representación gráfica.....	6
Metodologías Tradicionales:.....	6
Rational Unified Process (Rup):	6
Microsoft Solution Framework (Msp):	7
Win-Win Spiral Model:.....	7
Iconix:.....	8
Fases	8
Rational Unified Process (RUP):	8
Microsoft Solution Framework (MSP):	8
Win-Win Spiral Model:.....	8
Iconix:.....	9
Actividades por cada fase	9
Rational Unified Process (RUP):	9
Microsoft Solution Framework (MSP):	9
Win-Win Spiral Model:.....	10
Iconix:.....	10
Ventajas y Desventajas (Escobar, s.f.)	11
Cuadros comparativos	12
Metodologías Agiles.....	13
Representación grafica.....	14
Fases	17
Agile Inception.....	17
Design Sprint	17
Scrum.....	17
Programación Extrema	17

Kanban.....	18
Actividades por cada fase	18
Agile Inception.....	18
Design Sprint	19
Scrum.....	21
Programación extrema.....	22
Kanban.....	23
Ventajas	24
Agile Inception.....	24
Design Sprint	24
Scrum.....	24
Programación extrema.....	25
Kanban.....	25
Desventajas	26
Software que gestiona las fases.....	26
Kanban aplicado al Desarrollo de Software	26
Software para método scrum	28
Software para el método XP (Programación Extrema)	29
Cuadros comparativos	30
Bibliografía	33

INDICE DE IMAGENES

IMAGEN 1: FASES PARA UN PROYECTO -----	5
IMAGEN 2: REPRESENTACIÓN DE METODOLOGÍAS TRADICIONALES -----	6
IMAGEN 3: REPRESENTACIÓN DE RUP -----	6
IMAGEN 4: REPRESENTACIÓN MSP -----	7
IMAGEN 5: REPRESENTACIÓN DE WIN-WIN -----	7
IMAGEN 6: REPRESENTACIÓN DE ICONIX -----	8
IMAGEN 7: CUADRO COMPARATIVO ENTRE METODOLOGÍAS TRADICIONALES Y METODOLOGIAS AGILES-----	12
IMAGEN 8: MEJORES METODOLOGIAS AGILES -----	13
IMAGEN 9: REPRESENTACIÓN DEL MÉTODO ÁGIL -----	14
IMAGEN 10: REPRESENTACION GRAFICA DE SCRUM -----	14
IMAGEN 11: ASPECTOS FUNDAMENTALES DE LA METODOLOGIA SCRUM -----	14
IMAGEN 12: REPESENTACION GRAFICA DE KANBAN -----	14
IMAGEN 13: REPRESENTACION GRAFICA DE KAMBAN -----	15
IMAGEN 14: REPRESENTACIÓN DE LA METODOLOGÍA PROGRAMACIÓN EXTREMA -----	15
IMAGEN 15: BASES QUE CUMPLIR EN EL METODO AGIL -----	15
IMAGEN 16: PRINCIPIOS DE METODOLOGIA AGIL -----	16
IMAGEN 17: ACTIVIDADES DE LAS FASES DE SCRUM-----	21
IMAGEN 18: FASES DE PROGRAMACION EXTREMA -----	22
IMAGEN 19: REGLAS PARA USAR UNA METODOLOGIA KANBAN -----	23

Propósito del documento

El propósito de este documento es mostrar de manera, tanto general como específica algunas de las metodologías Tradicionales y Agiles con cada una de sus fases, ventajas y desventajas. También mostrar mediante cuadros comparativos el tema en sí.

Fases para un proyecto

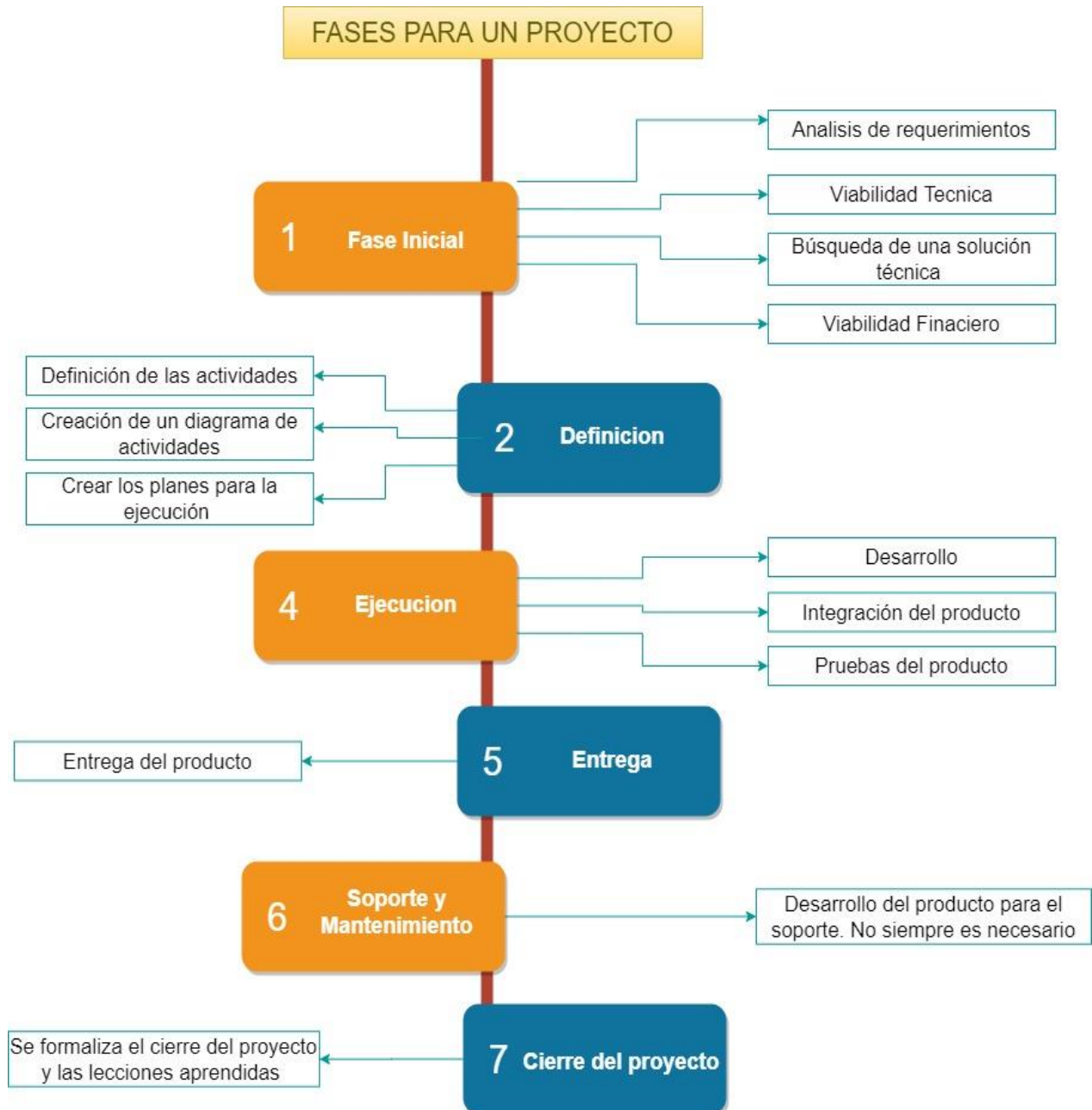


Imagen 1: Fases para un proyecto

Metodologías tradicionales

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas.

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Representación gráfica

Metodologías Tradicionales:



Imagen 2: Representación de Metodologías Tradicionales

Rational Unified Process (Rup):

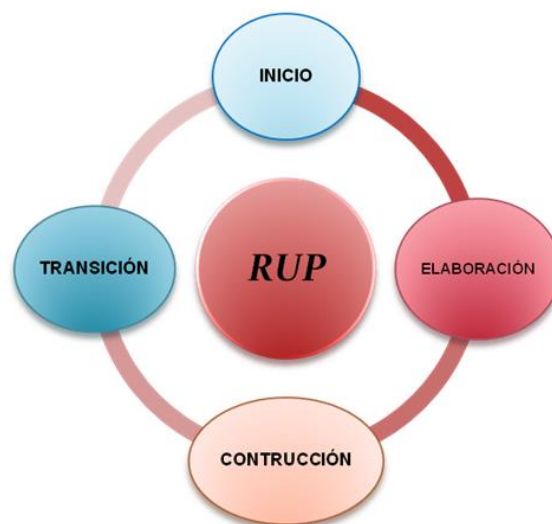


Imagen 3: Representación de RUP

Microsoft Solution Framework (Msf):

Ciclo de Vida del Proyecto

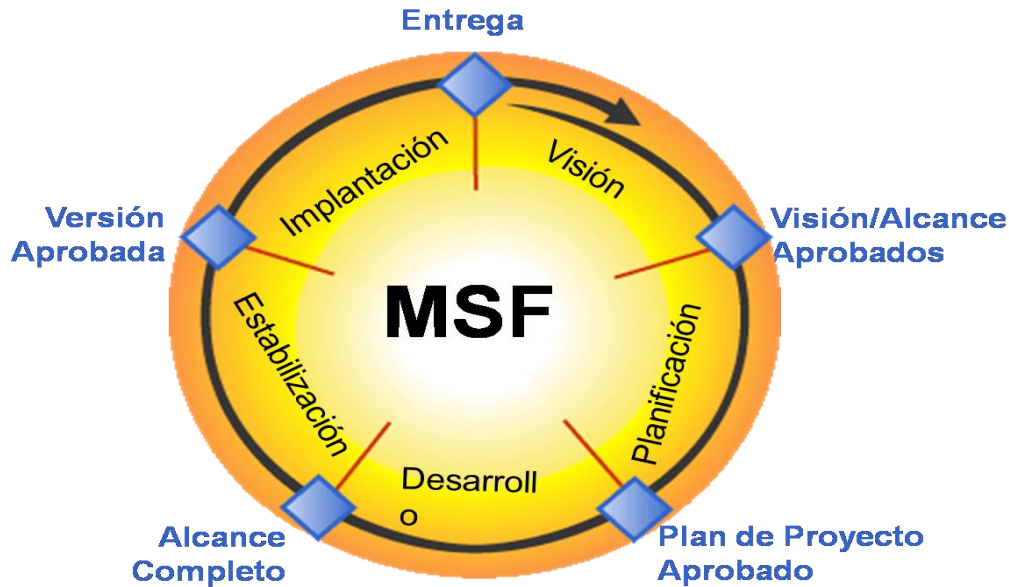


Imagen 4: Representación MSP

Win-Win Spiral Model:



Imagen 5: Representación de WIN-WIN

Iconix:

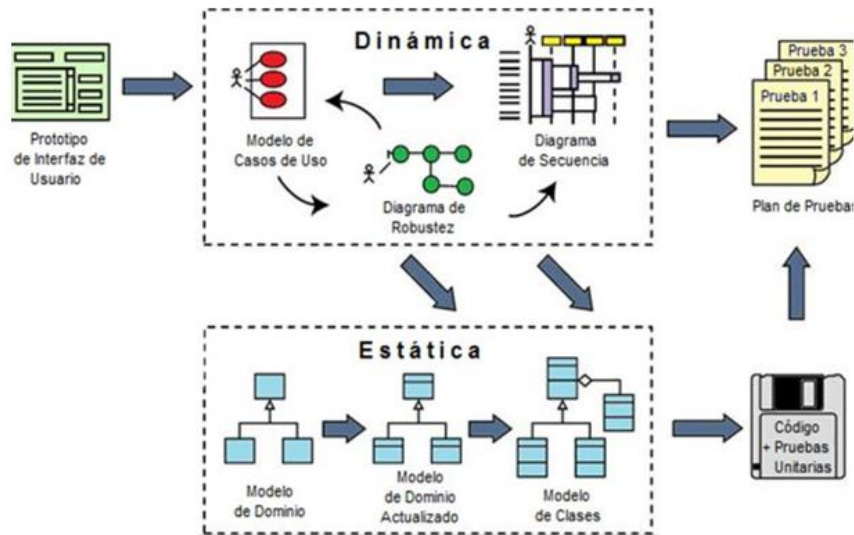


Imagen 6: Representación de Iconix

Fases

Rational Unified Process (RUP):

- Inicio (Concepción)
- Elaboración
- Construcción
- Transición

Microsoft Solution Framework (MSP):

- Visión y Alcances
- Planeación
- Desarrollo
- Estabilización
- Implantación

Win-Win Spiral Model:

- Comunicación con el cliente
- Planificación
- Análisis de riesgo

- Construcción y acción
- Evaluación del cliente

Iconix:

- Análisis de requisitos
- Análisis y diseño preliminar
- Diseño
- Implementación

Actividades por cada fase

Rational Unified Process (RUP):

- **Inicio (Concepción):** El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- **Elaboración:** Etapa en la que se determina la arquitectura optima del proyecto
- **Construcción:** Se obtiene la capacidad operación inicial
- **Transición:** Obtener el producto acabado y definido

Microsoft Solution Framework (MSP):

- **Vision y Alcances:** trata uno de los requisitos más fundamentales para el éxito del proyecto, la unificación del equipo detrás de una visión común. El equipo debe tener una visión clara de lo que quisiera lograr para el cliente y ser capaz de indicarlo en términos que motivaran a todo el equipo y al cliente.
- **Planeación:** Es en esta fase es cuando la mayor parte de la planeación para el proyecto es terminada. El equipo prepara las especificaciones funcionales, realiza el proceso de diseño de la solución, y prepara los planes de trabajo, estimaciones de costos y cronogramas de los diferentes entregables del proyecto.
- **Desarrollo:** Durante esta fase el equipo realiza la mayor parte de la construcción de los componentes , sin embargo, se puede realizar algún trabajo de desarrollo durante la etapa de estabilización en respuesta a los resultados de las pruebas
- **Estabilización:** En esta fase se conducen pruebas sobre la solución, las pruebas de estas etapas enfatizan el uso y operación bajo condiciones realistas. El equipo se enfoca en priorizar y resolver errores y preparar la solución para el lanzamiento.

- **Implantación:** Durante esta fase el equipo implanta la tecnología base y los componentes relacionados, estabiliza la instalación, traspasa el proyecto al personal soporte y operaciones, y obtiene la aprobación final del cliente

Win-Win Spiral Model:

- **Comunicación con el cliente:** Las tareas requeridas para establecer “Comunicación” entre el “Desarrollador” y el cliente
- **Planificación:** Las tareas requeridas para definir “Recursos”, el “Tiempo” y otra “información” relacionadas con el proyecto.
- **Análisis de riesgo:** Las tareas requeridas para “Construir” una o más representaciones de la Aplicación.
- **Construcción y acción:** Las tareas requeridas para “Construir”, “Probar”, “Instalar” y “Proporcionar” soporte al usuario.
- **Evaluación del cliente:** Las tareas requeridas para obtener la Reacción del Cliente según la Evaluación de las representaciones del software creadas durante la etapa

Iconix:

- **Análisis de requisitos:** Elaboración rápida de prototipos, Modelo de dominio, Modelo de casos de usos.
- **Análisis y diseño preliminar:** Descripción de los casos de uso, Diagramación de robustez
- **Diseño:** Diagrama de secuencia
- **Implementación:** Código y pruebas

Ventajas y Desventajas (Escobar, s.f.)

Ventajas		Desventajas
RATIONAL UNIFIED PROCESS (RUP)	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • La evaluación de riesgos es compleja • Excesiva flexibilidad para algunos proyectos • Estamos poniendo a nuestro cliente en una situación que puede ser muy incómoda para el
MICROSOFT SOLUTION FRAMEWORK (MSP)	<ul style="list-style-type: none"> • Crea una disciplina de análisis de riesgos que ayuda a evolucionar con el proyecto • Vinculación con el cliente como también orientado al trabajo en equipo • Tiene facilidad de soporte y mantenimiento • Es adaptable, se puede utilizar para proyectos de cualquier magnitud 	<ul style="list-style-type: none"> • Al estar basado en tecnologías Microsoft, trata de obligar a usar herramientas de ellos mismos • Solicita demasiada documentación en sus fases. • Si el análisis de riesgos se hace muy exhaustivo puede retardar el proyecto • Los precios de licencias, capacitación y soporte de Microsoft son caros
WIN-WIN SPIRAL MODEL	<ul style="list-style-type: none"> • El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. • 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. 	<ul style="list-style-type: none"> • Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable. • Debido a su elevada complejidad no se aconseja utilizarlo en pequeños sistemas. • Genera mucho tiempo en el desarrollo de sistemas.
ICONIX	<ul style="list-style-type: none"> • ICONIX es un modelo pequeño y firme que no desecha el análisis y el diseño. • Usa un análisis de robustez que reduce la ambigüedad al describir los casos. 	<ul style="list-style-type: none"> • No puede ser usado para proyectos grandes. • Necesita información rápida y puntual de los requisitos, el diseño y las estimaciones.



Cuadros comparativos

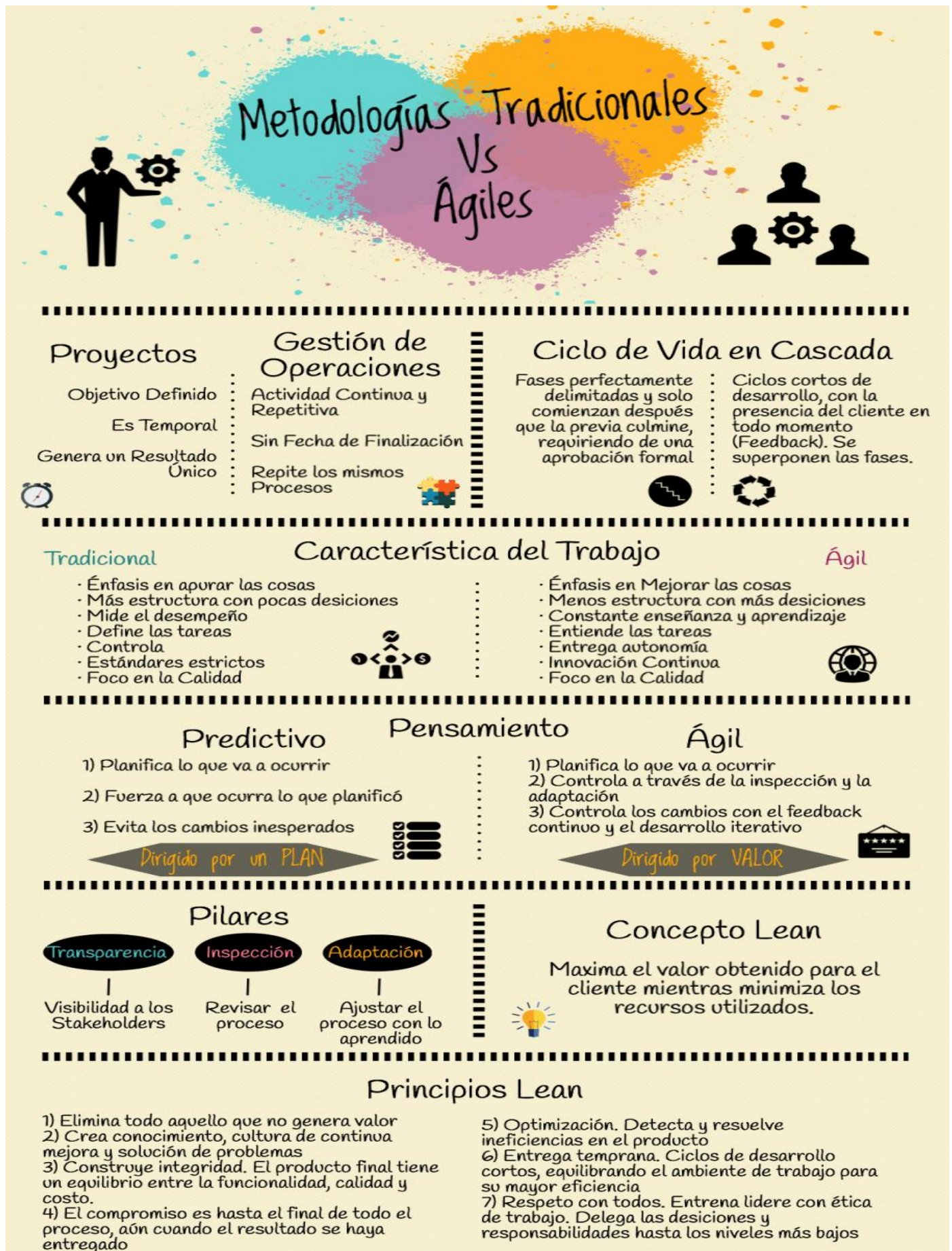


Imagen 7: Cuadro Comparativo entre Metodologías Tradicionales y Metodologías Ágiles

Metodologías Ágiles

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. Comparten ciertas características y buscan la interacción de los miembros del grupo de trabajo, con la meta de satisfacer los requisitos del cliente.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento". Es recomendable que las empresas apliquen estos métodos para eliminar el desperdicio que generan los esfuerzos sin planificación, las reuniones que consumen tiempo y no generan productividad. Se caracterizan por tener las siguientes cualidades como: Desarrollo evolutivo y flexible, Autonomía de los equipos, Planificación y Comunicación.

Existen diferentes opciones ágiles donde se destacan:



Imagen 8: Mejores metodologías Ágiles

Estos se encuentran compuestos por doce principios agrupados en cuatro que son fundamentales:

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Respuesta ante el cambio sobre seguir un plan.

Representación grafica

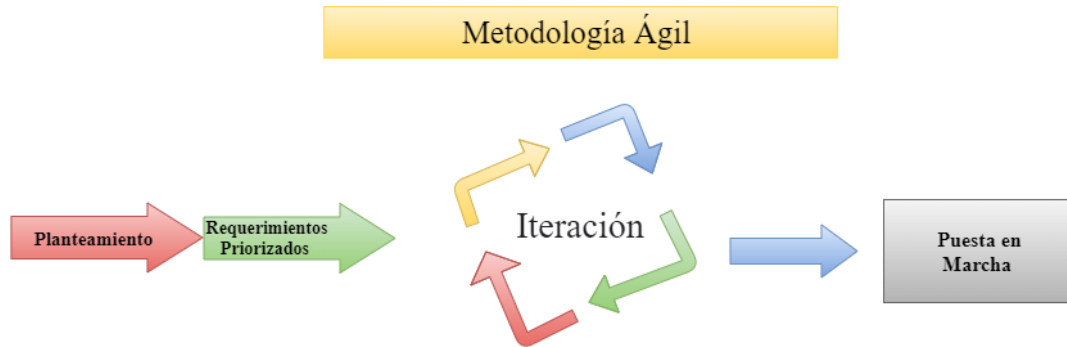


Imagen 10: Representación del Método Ágil

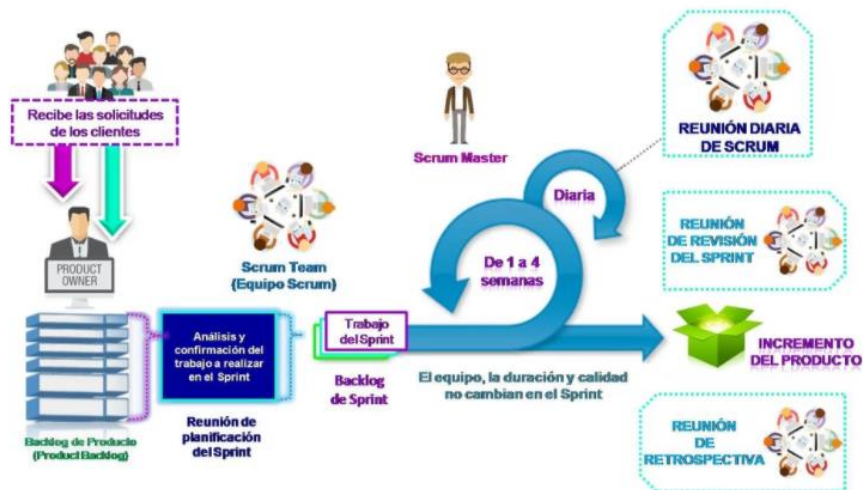


Imagen 9: Representacion Grafica de Scrum



Imagen 11: Aspectos fundamentales de la metodología Scrum



Imagen 12: Representacion Grafica de Kanban

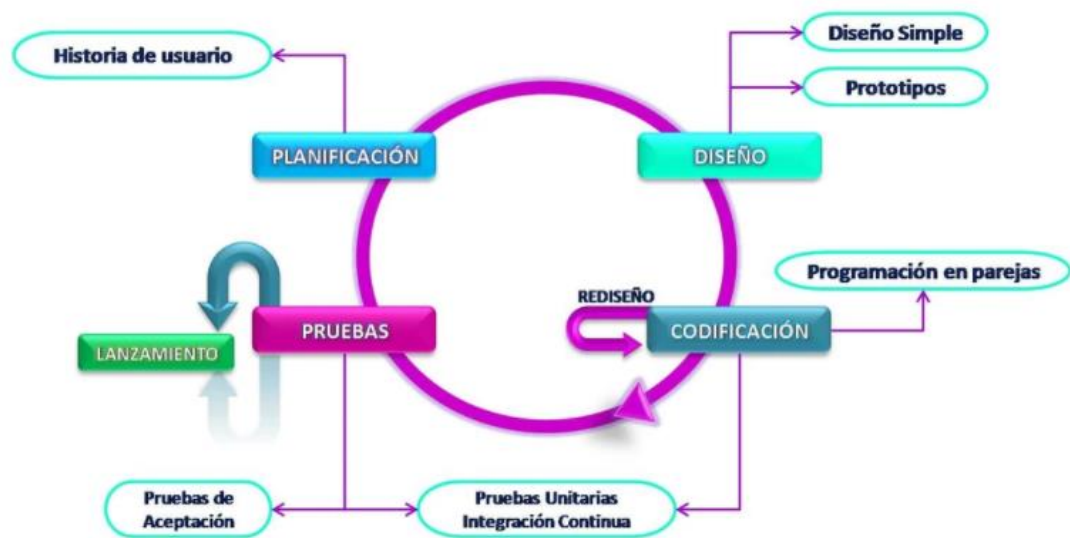


Imagen 13: Representación Gráfica de Kamban



Imagen 15: Bases que cumplir en el Metodo Agil

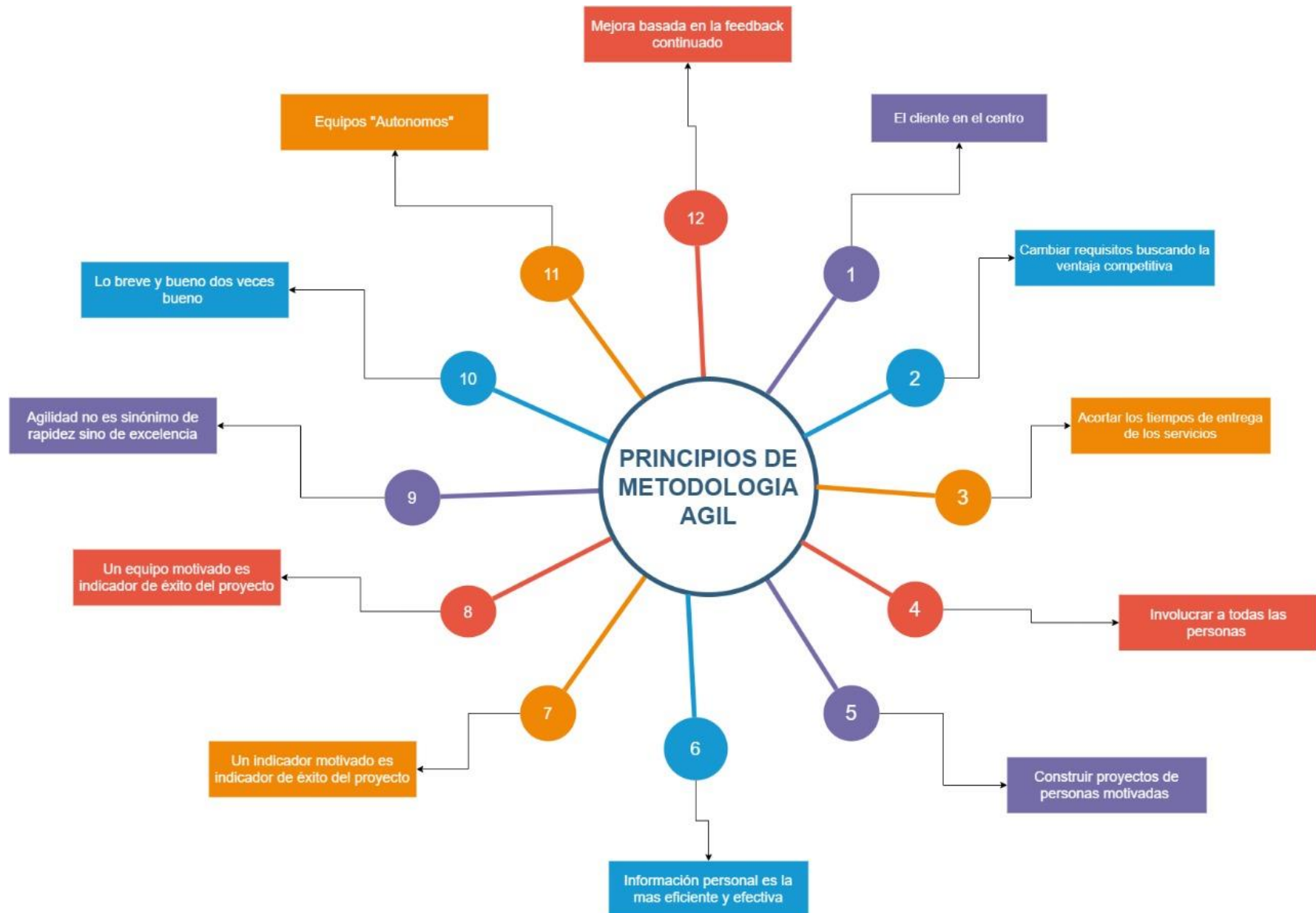


Imagen 16: Principios de Metodología Ágil

Fases

Agile Inception

- Why are we here?
- Elevator Pitch
- Product Box
- NOT List
- Meet your neighbors
- Show the solution
- Keep us at night
- Size it up
- Be clear on What's going to give
- Show what it's going to take (htt)

Design Sprint

- Comprender
- Idear
- Decidir
- Prototipar
- Testar (theherocamp)

Scrum

- Iniciacion
- Planificacion y Estimacion
- Implementacion
- Revision y Retrospectiva
- Lanzamiento

Programación Extrema

- Planeacion de proyecto
- Diseño
- Coodificacion

- Pruebas (tripod, s.f.)

Kanban

- Pendiente
- Analisis
- Desarrollo
- Pruebas
- Preproduccion
- Produccion
- Hecho

Actividades por cada fase

Agile Inception

1. Why are we here?

La primera de las preguntas que debemos hacernos en Agile Inception es encontrar los motivos por los que este proyecto tiene sentido para el cliente preguntándonos ¿por qué estamos aquí? Descubriendo los motivos principal de nuestro proyecto

2. Elevator Pitch

En este apartado el objetivo es construir un briefing de 45 segundos que resuma el proyecto y se centre en el valor real del producto.

3. Product Box

En este apartado el objetivo es construir una caja de producto como si la ofreciéramos en un centro comercial, junto a sus slogans principales.

4. NOT List

Una lista de Noes lo que hace es establecer las fronteras del producto a la hora de empezar a hablar, estableciendo un punto de referencia a las expectativas sobre lo que no se va a considerar como parte del proyecto.

5. Meet your neighbors

Es importante que todos los stakeholders del proyecto tengan en su radar al resto. Los entornos empresariales suelen ser grandes, y habitualmente no somos capaces de identificar cuántas personas están involucradas para lograr que el proyecto sea un éxito.

6. Show the solution

En este apartado se debe de trabajar en conjunto para mostrar una primera aproximación de la solución, planteando arquitectura, tecnología y herramientas que serían necesarias para la construcción del producto.

7. Keep us at night

En este apartado debemos detectar aquellas cosas que nos pueden mantener despiertos por la noche. Es el lugar adecuado para que salgan a la luz aquellos riesgos que puedan influir en el desarrollo del producto, desde todas las perspectivas posibles. De tal manera que detectando estos riesgos podremos manejarlos y tenerlos en mente durante la evolución del proyecto, siendo consciente de las posibles desviaciones que puedan conllevar.

8. Size it up

En esta dinámica se trata de lograr una estimación de alto nivel de cuándo esfuerzo en recursos y tiempo puede conllevar el proyecto. No se trata de dar una estimación exacta, pero si de algo aproximado teniendo en cuenta todas las implicaciones que conlleva.

9. Be clear on What's going to give

En esta dinámica se trata de establecer ciertas prioridades en los factores claves de ejecución del proyecto, y si entran en conflicto cuales son más prioritarios.

10. Show what it's going to take

En esta dinámica se trata de calcular aproximadamente el número de personas involucradas para cumplir con las necesidades en el tiempo establecido y estimar el coste que pueda conllevar. (htt)

Design Sprint

1. Comprender

Primera fase, y quizás la más importante. En este momento el equipo de trabajo debe:

- Saber quién es el usuario, cuáles son sus necesidades, motivaciones y sueños.
- Conocer el contexto del producto y de la organización.

2. Idear

Después de comprender el contexto del producto y a los potenciales usuarios a los que nos vamos a dirigir, es momento de idear posibles soluciones al problema planteado. Para ello el equipo debe:

- Plantear ideas sin importar la calidad, solo la cantidad (tengamos en cuenta la divergencia)
- Ser multidisciplinar con el objetivo de tener diferentes visiones.

3. Decidir

Fase de toma de decisiones sobre lo trabajado en las dos fases anteriores. Es aquí cuando el equipo debe:

- Tomar decisiones apoyándose en las conclusiones y la información recabada en las fases anteriores.
- Trabajar en equipo y discutir sobre las decisiones a tomar.

4. Prototipar

En esta fase hay que hacer, hacer y hacer. El equipo debe:

- Decidir si en este sprint va a querer validar toda la solución o solamente una parte.
- Bocetar prototipos, quitando importancia al diseño final. Recordemos que lo importante es la validación de la idea.

5. Testar

Llegamos a uno de los momentos clave: la fase de pruebas con usuarios reales. En este momento el equipo debe:

- Probar el prototipo con usuarios reales.
- Dejar total libertad de uso. No olvides que el usuario es el rey.
- Aprender del uso dado por el usuario. (theherocamp)

Scrum

- **Planeación del Sprint (o Sprint Planning):** En esta etapa, los involucrados se reúnen para planificar el Sprint y designar las tareas que desarrollará cada persona del equipo y cada uno de ellos deberá asignar un tiempo determinado para realizar su tarea. De esta forma se definirá el tiempo de duración del Sprint.
- **Scrum team meeting (reunión de equipo Scrum):** Estas reuniones deben tener una duración de 15 min diarios y en estas sirven para darse apoyo mutuo en caso de encontrar problemas en el desarrollo de alguna actividad. Se debe responder 3 preguntas:
 - ¿Que se hizo ayer?
 - ¿Qué tienes planeado hacer hoy?
 - ¿Qué obstáculos encontraste en el camino?
- **Backlog Refinement (refinamiento del backlog):** Es una nueva revisión por parte del Product Owner de los requerimientos para aclarar dudas que pueda tener el equipo de desarrolladores. Si es necesario, se volverán a definir los plazos.
- **Sprint Review (revisión del Sprint):** Es una revisión de lo que se ha realizado dentro del Sprint y se muestra el trabajo finalizado. Esto está a cargo de Scrum Master y el Product Owner.
- **Retrospective (retrospectiva del Sprint):** En este punto el Product Owner se reúne con su equipo para hablar sobre lo ocurrido durante el Sprint y se tratan estos puntos:
 - Qué se hizo mal durante el Sprint para buscar posibles mejoras
 - Qué se hizo bien para seguir esa misma línea (Navarrete, 2019)

(Continental10, 2020)



Imagen 17: Actividades de las fases de Scrum

Programación extrema

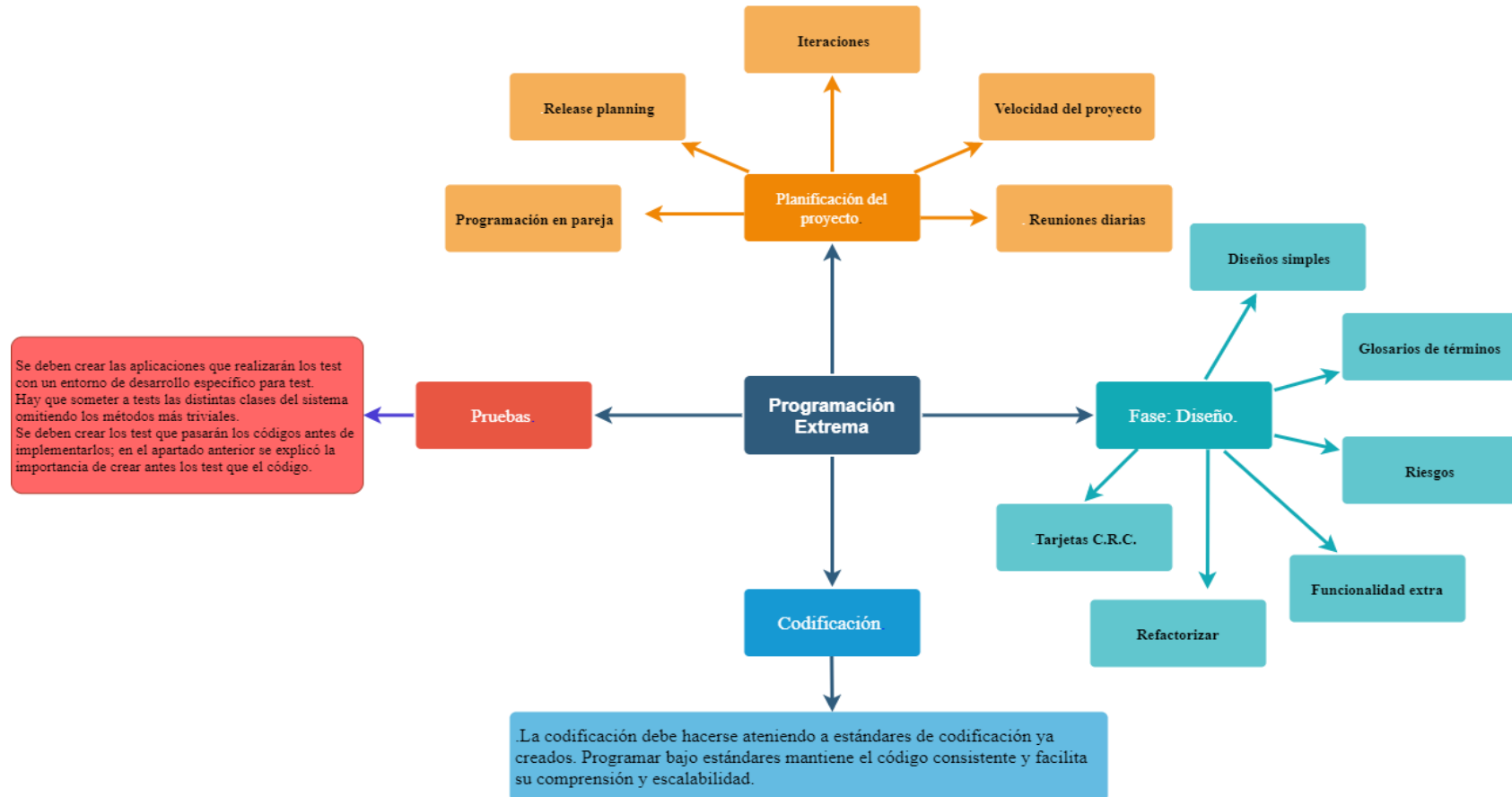


Imagen 18: Fases de Programación Extrema

Kanban

1. Instrucción de todo el personal acerca de la metodología Kanban. Esta fase ayuda a cada miembro del equipo a tener los conocimientos y conciencia en cuanto al manejo de la metodología y los beneficios de su aplicación.
2. Implementación del sistema Kanban en los componentes con más problemas. Esta implementación se lleva a cabo en los componentes que presenten mayores problemas o dificultades, todo con la intención de facilitar su ejecución o manufactura. La segunda fase permite, a su vez, enfatizar o resaltar problemas que no se habían detectado.
3. Implementación de Kanban en los componentes restantes. Solo se implementa el sistema en el resto de los componentes cuando han sido halladas soluciones para aquellos que presenten problemas de mayor envergadura. En esta fase, los miembros del equipo tienen una mayor conciencia y manejo de Kanban, por lo tanto, ya conocen sus ventajas. Además, se supone que los operarios de la empresa, categorizados por áreas, ya manejan pormenorizadamente el sistema Kanban, de modo que es fundamental informarles cuándo se estará trabajando en su área y escuchar sus dudas / opiniones.
4. Revisión del sistema o metodología Kanban. Consiste en una revisión exhaustiva del sistema para determinar qué puntos deben reordenarse. En esta fase, es muy importante constatar que ningún trabajo se realice fuera de secuencia y que cualquier problema se notifique lo más pronto posible al supervisor.

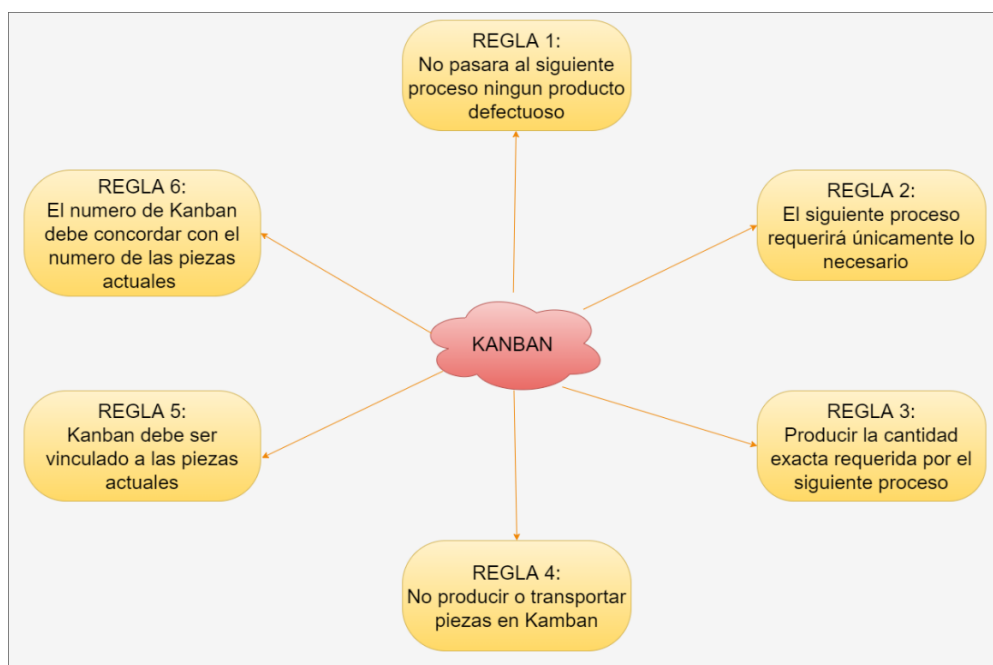


Imagen 19: Reglas para usar una Metodologia Kanban

Ventajas

Agile Inception

La mayor ventaja que nos trae esta metodología, es que nos evita el rotundo fracaso del proyecto cuando esta a punto de ser terminado ya que esta metodología nos permite al equipo estar preparados para las circunstancias y problemas que pueden venir adelante. Nos ayuda a conocer mejor el producto o proyecto a construir, determina que es necesario, Confirma el tiempo estimado de la inception y define las prioridades para el desglose a mas bajo nivel. (2017)

Design Sprint

- Los Design Sprint te ayudan a ahorrar tiempo y dinero
- Los Design Sprint reducen rápidamente los ciclos de desarrollo de producto
- Feedback real con los Design Sprint
- Valida tus ideas de negocio gracias al Design Sprint
- Genera negocio e innovación.
- Alineas expectativas con tu equipo
- Te ayudan a medir
- Metodología ágil y rápida que puedes aplicar a tu negocio. (letshackity)

Scrum

- **Gestión de las expectativas del usuario.** Los usuarios pueden participar en cada una de las etapas del proceso y proponer soluciones. De hecho, el proceso en su conjunto está pensado para un tipo de evaluación conjunta.
- **Resultados anticipados.** Cada etapa del proceso arroja una serie de resultados. No es necesario, por tanto, que el cliente espere hasta el final para ver el resultado.
- **Flexibilidad y adaptación a los contextos.** Se adapta a cualquier contexto, área o sector de la gestión. Es decir, no es una técnica exclusiva de ninguna disciplina.
- **Gestión sistemática de riesgos.** Del mismo modo, los problemas que aparecen durante los procesos de gestión que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición. Esto es posible debido a que la intervención de los equipos de trabajo puede ser inmediata. (Español, 2020)

Programación extrema

- Da lugar a una programación sumamente organizada.
- Ocasiona eficiencias en el proceso de planificación y pruebas.
- Cuenta con una tasa de errores muy pequeña.
- Propicia la satisfacción del programador.
- Fomenta la comunicación entre los clientes y los desarrolladores.
- Facilita los cambios.
- Permite ahorrar mucho tiempo y dinero.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.
- La XP es mejor utilizada en la implementación de nuevas tecnologías. (Alberto, 2015)

Kanban

- Planificación de tareas.
- Tiempos de ciclos reducidos.
- Rendimiento del equipo de trabajo.
- Métricas visuales.
- Menos cuellos de botella.
- Entrega continua.
- Mejora la calidad: Minimiza los errores en los entregables y mejora la experiencia y las funcionalidades para el cliente.
- Mayor compromiso: Mejora la satisfacción del empleado y genera conciencia de equipo.
- Rapidez: Acorta los ciclos de producción y minimiza los tiempos de reacción y toma de decisiones.
- Aumento de la productividad: Al asignar mejor los recursos, y de forma más dinámica, mejora la producción según las prioridades que tenga la empresa.

Desventajas

Fuerte dependencia de los líderes. Los equipos de trabajo dependen en buena medida del liderazgo de la persona responsable. Las reuniones continuas y las evaluaciones periódicas hacen que la persona que encabeza el proyecto centralice casi todas las decisiones y responsabilidades



Falta de documentación. Las metodologías Ágiles no plantean alternativas a para la recolección de la información de los proyectos. Simplemente plantea la manera cómo se llevarán a cabo las acciones.



Soluciones erróneas en etapas largas. Cuando las iteraciones tienden a ser muy largas, se corre el riesgo de que las soluciones esbozadas al inicio de las etapas no sean las correctas. Una fase larga puede evolucionar mientras se está ejecutando y, por tanto, las medidas tomadas tienen a perder vigencia.



Se requiere un **alto nivel de interacción** entre el cliente y los desarrolladores.

Software que gestiona las fases

Kanban aplicado al Desarrollo de Software

Se presenta un cuadro comparativo de herramientas software externas que aporten tableros Kanban para apoyar a los equipos de desarrollo qué aportan a los equipos y las funcionalidades que tienen sus respectivos tableros.

Característica	Jira Software	Targetprocess	PivotalTracker
Drag and drop de las fichas	SI	SI	SI
Cambiar de orden las columnas	SI	NO	SI
Cambiar de orden las columnas	SI	SI	NO
Cambiar de orden las columnas	SI	SI	SI
Añadir directamente nuevas fichas	NO	SI	SI
Contador de fichas	SI	SI	SI
Ordenación de las fichas	NO	NO	NO
Ocultar columnas si están vacías	NO	SI	SI
Ocultar columnas si están vacías	NO	SI	NO
Editar el ítem al acceder a él	SI	SI	SI
Priorizar un ítem al moverlo	NO	SI	NO

Software para método scrum

<p>Jira</p>	<ul style="list-style-type: none"> • Tableros de Scrum personalizables. • Informes de progreso en el desarrollo del proyecto. • Rastreador de errores y problemas. • Filtros personalizados • Gestión de usuarios. • Registro del tiempo. • Cuadro de mandos personalizable. • Reportes en tiempo real
<p>QuickScrum</p>	<ul style="list-style-type: none"> • Tableros Scrum para la visualización de flujos de trabajo. • Interfaz drag and drop sencilla de utilizar. • Soporte para archivos adjuntos. • Visualización del progreso. • Integración de comentarios en cada ítem de trabajo. • Status de progreso individualizado para cada tarea. • Filtros personalizables.
<p>ScrumDo</p>	<ul style="list-style-type: none"> • Gestión de tareas y subtareas • Tableros de proyectos • Integración de calendario del proyecto con actualizaciones en tiempo real. • Creación de casos de usuario

Software para el método XP (Programación Extrema)

Software para el método XP (Programación Extrema)	Características
SQL SERVER MANAGER STUDIO 2012	<ul style="list-style-type: none"> • Un sistema de administración de datos gratuito, eficaz y confiable que • Ofrece un almacén de datos completo y confiable para sitios web ligeros y aplicaciones de escritorio.
VISUAL STUDIO 2013	<ul style="list-style-type: none"> • Visual Studio 2013 es el IDE de programación por excelencia de la plataforma .NET y, por ende, de las aplicaciones que corren en los dispositivos del ecosistema de Microsoft.
ASP.NET	<ul style="list-style-type: none"> • ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. • ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework.
Entorno de Diseño Dirigido por el Modelo de Alta Productividad	<ul style="list-style-type: none"> • Verdadera separación de modelos físicos y lógicos con integración completa, transformación automatizada, y mapeo de tipos personalizados de datos.



Cuadros comparativos

- **Tipo de iteraciones:** puede ser de alcance o plazo fijos” termina el tiempo alcance.
- **Roles (Facilitador, Administrador Requerimientos, Equipo Proyecto):** indican cuáles son los roles en común para cada metodología.
- **Tipo de equipo:** Indica las características que debe tener el equipo que adopte cierta metodología.
- **Limitación del trabajo en progreso:** Se indican las diferencias de cómo limitado el número de elementos de trabajo que son realizados al mismo tiempo el flujo de trabajo.
- **Incorporar tareas:** Debido a que existe la posibilidad de que los requerimientos cambien o se agreguen nuevos, se evalúa cuando es posible incorporarlos.
- **Seguimiento de procesos:** Se indica la forma como es realizado el seguimiento performance de las actividades.
- **Estimaciones:** Se compara la necesidad de realizar la estimación de las tareas inicio.

Metodologías Ágiles			
Criterios	SCRUM	XP	Kanban
Tipo de Iteración	Iteraciones de plazo fijo	Iteraciones de plazo variable	Iteraciones plazo fijo variable
Roles-Facilitador	Scrum Máster	Coach, Big Boss	N/A
Roles Administrador Requerimiento	Product Owner	Cliente	N/A
Roles – Equipo Project	Equipo de Desarrollo	Programador, Tester	N/A
Prácticas / Reglas	9	12	3
Limitación Work InProgress	Limitación por iteración	Limitación por iteración	Limitación por estado
Incorporación de Tareas	No es posible hasta finalizar el sprint	No es posible hasta terminar la iteración	Es posible, en tanto exista capacidad
Seguimiento de tareas	Gráfico Burn-Down	Velocity	Tablero Kanban
Estimación	Obligatoria	Obligatoria	Opcional

Metodología Agiles	Comunicación	Planeación	Modelado	Construcción	Despliegue
Extreme Programing xp	Se pone en comunicación continua con los clientes y desarrolladores	clasificación y priorización de requerimiento. Generar plan de Iteraciones, es decir se planifica al inicio del proyecto, de cada iteración y aún más detalladamente, todos los días.	Diseño, Plan de Prueba, codificación, en resumen se mejora el diseño y la arquitectura poco a poco y de forma constante.	Aprobación del usuario, publicación de la versión.	Al terminar cada semana, el software obtenido pueda ser puesto en producción, pues la funcionalidad comprometida está asegurada, éstas se realizarán siguiendo un plan de entregas establecido.
Scrum	Identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.	Desarrollo de un backlog completo. Determinación de la fecha de entrega y la funcionalidad de una o más versiones.	El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días.	Todos los involucrados en el equipo se reúnen para planificar el Sprint. Durante este evento se decide qué requerimientos o tareas se le asignará a cada uno de los elementos del equipo	Cuando el equipo de gestión siente que las variables de tiempo, parte completada, requisitos, coste y calidad están alineadas para producir una nueva versión
Kanban	Desarrollo incremental, dividiendo el trabajo en distintas partes. Esto significa que no hablamos de la tarea en sí, sino que lo dividimos en distintos pasos para agilizar el proceso de producción.	Permite obtener beneficios del enfoque visual, mediante el uso de columnas y tarjetas. Es posible trabajar a la vez con el tablero y colaborar con los compañeros	Inicia desde su petición hasta su entrega, mientras que cycle-time inicia desde que una actividad, comienza hasta su finalización, es decir mide el rendimiento del proceso	El número de columnas y la denominación de estas las establece el equipo de trabajo.	Consta de 3 columnas que indican el estado de cada actividad, en este caso se nombra a los estados como: ToDo, Doing y Done.

Metodología Tradicional	Metodología Ágil
<ul style="list-style-type: none"> Orientadas a proyectos de cualquier tamaño 	<ul style="list-style-type: none"> Orientadas a proyectos pequeños
<ul style="list-style-type: none"> Equipos grandes y Dispersos 	<ul style="list-style-type: none"> Equipos pequeños, sobre 10 personas
<ul style="list-style-type: none"> Proyectos de media/larga duración 	<ul style="list-style-type: none"> Proyectos de corta duración
<ul style="list-style-type: none"> Proyecto cerrado 	<ul style="list-style-type: none"> Proyecto abierto a cambios
<ul style="list-style-type: none"> El cliente mantiene reuniones con la dirección 	<ul style="list-style-type: none"> El cliente está integrado en el equipo
<ul style="list-style-type: none"> Arquitectura prefijada 	<ul style="list-style-type: none"> Arquitectura se va mejorando
<ul style="list-style-type: none"> Documentación rigurosa 	<ul style="list-style-type: none"> Poca documentación
<ul style="list-style-type: none"> Centrada en los procesos 	<ul style="list-style-type: none"> Centrada en las personas
<ul style="list-style-type: none"> Roles no intercambiables 	<ul style="list-style-type: none"> Roles flexibles
<ul style="list-style-type: none"> Roles específicos 	<ul style="list-style-type: none"> Roles genéricos
<ul style="list-style-type: none"> Gestión dirigida 	<ul style="list-style-type: none"> Gestión colaborativa
<ul style="list-style-type: none"> Alto coste en prototipado 	<ul style="list-style-type: none"> Bajo coste de prototipado
<ul style="list-style-type: none"> Proceso lineal 	<ul style="list-style-type: none"> Proceso iterativo

Bibliografía

Alberto, S. (14 de Septiembre de 2015). Obtenido de <https://iswugaps2extremeprogramming.wordpress.com/2015/09/14/ventajas-y-desventajas/>

Continental10. (07 de junio de 2020). *issu*. Obtenido de https://issuu.com/continental10/docs/metodologias_tradicionales

Escobar, J. O. (s.f.). *slideshare*. Obtenido de https://www.slideshare.net/MarianaGonzalezGomez/enfoque-tradicional-y-actual-del-maestro?next_slideshow=1

Españon, A. P. (12 de 02 de 2020). Obtenido de <https://blog.wearedrew.co/ventajas-y-desventajas-de-la-metodologia-scrum>

Mobiliza Academia. (2019 de Julio de 09). Obtenido de <http://academia.mobiliza.cl/fases-de-la-metodologia-scrum/>

Muradas, Y. (08 de 04 de 2018). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Navarrete, C. (12 de julio de 2019). *Mobiliza Academia*. Obtenido de <http://academia.mobiliza.cl/fases-de-la-metodologia-scrum/>

ThecnologyConsultin. (16 de 10 de 2018).

tripod. (s.f.). Obtenido de <https://programacionextrema.tripod.com/fases.htm>