

Proyecto Certificador de Desarrollo de Software 2

Tema Nº1:Gestión de Proyectos

Indicador de logro Nº1:

Identifica el enfoque de gestión de proyectos más adecuado logrando desarrollar un producto innovador de software en una empresa

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº1:**

Gestión de Proyectos

OBJETO DE LA EXPERIENCIA

Identifica el enfoque de gestión de proyecto (predictivo o ágil) de acuerdo con las características del producto a desarrollar y conoce los problemas o necesidades del cliente a través del diagrama persona

MARCO TEÓRICO

Los proyectos. - Existen desde siempre, aunque no se tuviera la noción de que lo que se estaba emprendiendo era un proyecto como tal. Cualquier trabajo para desarrollar algo único es un proyecto, pero la gestión de proyectos es una disciplina relativamente reciente que comenzó a forjarse en los años sesenta. La necesidad de su profesionalización surgió en el ámbito militar.

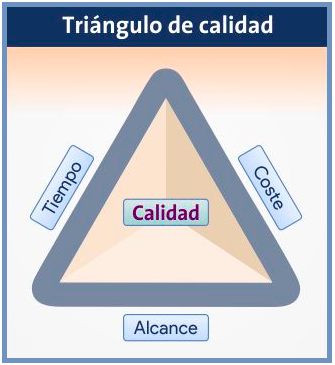
Un proyecto comprende un cúmulo específico de operaciones diseñadas para lograr un objetivo con un alcance, recursos, inicio y final establecidos.

Objetivos:

* Gestionar el inicio y la evolución de un proyecto;
* Controlar y responder ante problemas que surjan durante un proyecto;
* Facilitar la finalización y aprobación del proyecto.

Los proyectos son independientes de la actividad diaria empresarial, por lo que se requiere que se organicen una serie de reuniones para ver cuáles son los objetivos específicos del proyecto. Para que el proyecto tenga éxito es esencial que se realice un trabajo en equipo eficiente. La manera en la que la gestión de proyectos dirigirá el trabajo depende de varios factores, entre ellos: la escalabilidad (la posibilidad de que el proyecto crezca), la importancia y la complejidad de las tareas.

Hay muchas restricciones a la hora de desarrollar un proyecto. Sin embargo, las tres más comunes son el tiempo, el coste y el alcance. Estas restricciones forman parte de todos los proyectos y juntas forman el Triángulo de Gestión de Proyectos. El alcance es importante para especificar todos los pasos del desarrollo del proyecto. Por otra parte, el tiempo es un recurso invaluable. Si bien podemos controlar los procesos, no podemos controlar el tiempo. Por lo que es un verdadero desafío poder utilizar el tiempo de manera eficiente, mantener el proyecto dentro del cronograma y alcanzar los objetivos deseados. Sin embargo, el coste está compuesto por un presupuesto establecido en la etapa inicial del proyecto. Después, éste se compara con la cifra que se propuso inicialmente. Las tres restricciones están interconectadas y dependen mucho la una de la otra. Una vez que se reduce el tiempo asignado para el proyecto, el costo aumenta. Además, el alcance del proyecto dicta el ritmo y una serie de recursos necesarios para realizar y completar con éxito el proyecto.



MATERIALES

Según la experiencia a realizar, de acuerdo al catálogo de equipamiento del taller.

a) EQUIPOS (Hardware)

* Una computadora PC, CoreI5/CoreI7, 8GB/16GB RAM, HD 700GB o superior

b) HERRAMIENTAS (Software)

* Sistema Operativo Ms Windows 10, Microsoft Windows Server
* Microsoft Excel 2017/2019
* Trello, software libre, [www.trello.com](http://www.trello.com)
* Jira, software libre, [www.atlassian.com](http://www.atlassian.com)

c) CONSUMIBLES

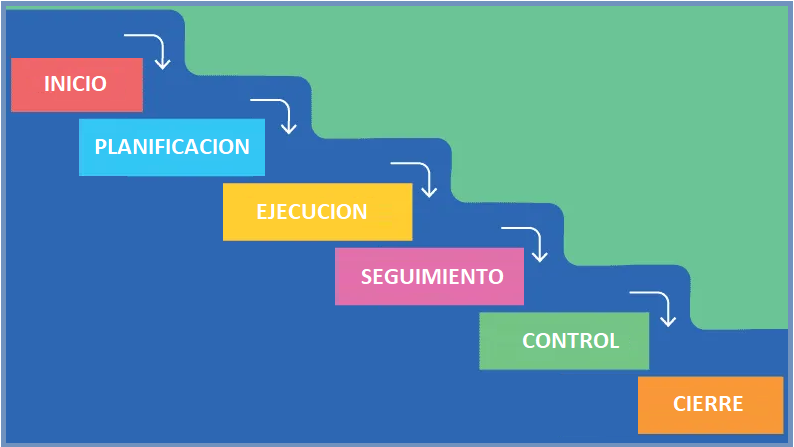
* Pizarra
* Proyector
* Plumones y mota
* Limpiatipo
* Papelógrafo blanco cuadriculado
* Posits de colores

PROCEDIMIENTO

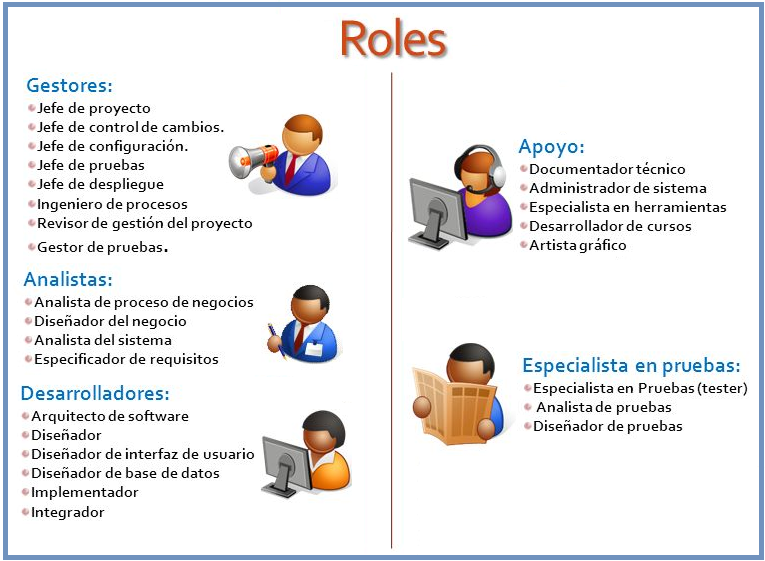
Paradigmas y características en la gestión de proyectos. (Predictivo y Ágil)

Ágil y Predictivo son dos metodologías de trabajo utilizadas para la gestión de proyectos. Cada una de ellas tiene su entorno y su aplicabilidad como consecuencia inherente a sus diferentes planteamientos. Sin embargo, es muy posible que algunos de nuestros proyectos puedan verse beneficiados de la correcta interacción entre ambas metodologías.

En la forma predictiva de proyectos, esta posee una estructura con un orden estricto conformado por las siguientes fases:



Se tienen roles definidos jerárquicamente:



Ventajas:

* Facilidad para llevar el seguimiento y control de lo que se está desarrollando (porcentaje de cumplimiento).
* Cada fase tiene resultados específicos.
* Útil para gestionar dependencias.
* Permite la estimación de calendarios y presupuestos con mayor precisión.
* El cliente sabe que lo que va a recibir una vez finalice el proyecto, desde la fase de inicio, o con los cambios que se hayan podido incluir mediante una gestión de cambios.
* El proceso y los entregables exigen un nivel de documentación.

Desventajas:

* Una vez cerrada la fase de requerimientos, cuesta aceptar un nuevo requerimiento y éste se convierte en una gestión de cambio que afecta al tiempo y coste del proyecto.
* Resulta difícil responder de forma inmediata a los problemas que puedan surgir.
* Si los requisitos no están claros desde un principio, resulta una metodología poco efectiva.
* Las pruebas se inician al finalizar el desarrollo, con lo cual existe una gran probabilidad de encontrar errores y tener que volver a la fase de desarrollo, retrasando así el proyecto.

En la forma agile de proyectos. - Puede considerarse como la nueva forma de gestionar los proyectos. Se basa en una forma de trabajo colaborativo: las fases del proyecto se realizan con todos los actores involucrados desde el inicio del proyecto; y se gestiona en base a entregables. Se trata de una práctica que admite la repetición continua de desarrollo y pruebas durante el proceso de desarrollo de software.



Ventajas:

* Ofrece un modelo increíblemente flexible, que permite adaptarse evolutivamente a las necesidades de un mundo cambiante.
* El proyecto se divide en pequeñas tareas desarrolladas por grupos independientes que trabajan simultáneamente e interaccionan entre ellos.
* Los clientes intervienen y las pruebas del producto ocurren simultáneamente a la realización del proyecto, con lo cual se consigue una mejora continua y una adecuación constante del producto a las necesidades reales.
* La retroalimentación mutua entre el equipo del proyecto y los clientes, conseguirá que progresivamente los objetivos del cliente y el producto facilitado por el equipo vayan convergiendo hacia un mismo punto, consiguiendo un resultado satisfactorio.
* La comunicación integra los procesos de desarrollo, consigue la cohesión entre el personal y los clientes y garantiza la coherencia del resultado del proyecto con los objetivos que se van planteando de manera dinámica.
* Permite la integración con otras metodologías de trabajo actuales y muy interesantes para la organización de una empresa.

Desventajas:

* Precisamente como consecuencia de su flexibilidad, Ágil muestra una estructura muy débil.
* Existe poca exactitud en la planificación del proyecto, desde los plazos de entrega hasta los presupuestos.
* La implicación personal y la colaboración se necesitan para garantizar el éxito de un proceso Ágil. Esto puede ser problemático, especialmente si trabajamos con equipos poco colaboradores.
* La necesidad de una estrecha y permanente comunicación, puede consumir mucho tiempo en reuniones e intercambios de contenidos.
* Por su parte, además, las metodologías ágiles son mucho más dependientes de la presencia de las mismas personas desde el principio hasta el final del proyecto.

Framework Cynefin

Cynefin se trata de una palabra de origen galés que significa “hábitat” como sustantivo, y “habitual” como adjetivo, para intentar expresar la naturaleza evolutiva de los sistemas complejos, como lo son esas organizaciones cada vez más complejas. Según Dave Snowden, el Cynefin nació para mejorar tanto la percepción y como la comprensión de las situaciones que vive una organización y capacitar para tomar las decisiones correctas según el contexto en el que se encuentre.



**Obvio (Simple)**

Todo es conocido. Una de las características principales de los dominios simples son las relaciones de causa y efectos fáciles de percibir para todos. De este modo, la forma de toma de decisiones es asumida con naturalidad por el conjunto de la organización y se es capaz de dar una respuesta correcta en todo momento. Sentir (Percibir), Categorizar y Responder son las acciones que priman en este contexto para los líderes de las organizaciones

**Complicado**

Este es el escenario en el que se sabe lo que no se sabe. La relación causa y efecto siguen existiendo, aunque ya no son tan claras como en el dominio simple. Además, ya no hay una sola respuesta. Se necesita experiencia y capacidad de análisis, y para ello como norma general se va a requerir expertos, consultores para que ayuden a los líderes a tomar las decisiones correctas. En este caso, Sentir (Percibir), Analizar y Responder son las acciones que requieren un escenario complicado.

**Complejo**

El dominio de lo desconocido, de donde no se sabe lo que no se sabe. Las organizaciones no pueden predecir los resultados porque la relación entre causa y efecto, aunque existe, no se puede localizar y, como en el ejemplo de la selva, es imposible determinar si una respuesta concreta va a suponer el mismo resultado en cualquier otro momento. Y como con la selva, la ayuda de expertos no tiene que llevar a los líderes hacia las respuestas correctas. Sondear, percibir y responder son las acciones propias del contexto complejo.

El dominio complejo se corresponde con la mayoría de las organizaciones actuales, alejadas de las formas tradicionales de gestión, mucho más elementales.

**Caos (Caótico)**

Como su nombre indica, se trata de un contexto en el que es imposible tratar de encontrar respuestas correctas, muy difícil encontrar relaciones claras de causa y efecto, patrones o la ayuda de expertos. La organización necesita estabilizar la situación y contener el desorden. Actuar, Sentir (Percibir) y Responder son las claves de este cuarto cuadrante.

**Desordenado**

Cuando, una vez que se ha realizado el análisis, los líderes de una organización no son capaces de determinar en qué contexto se encuentra, es posible que la respuesta sea el dominio del Desorden. En tal situación, la gestión correcta sería el análisis por partes de la organización, para intentar situar cada una de ellas en el dominio que les corresponden y pasar entonces a actuar sobre las distintas partes según su dominio.

El Manifiesto Ágil o Agile Manifesto. - Es un documento firmado en febrero de 2001 por 17 pensadores independientes del mundo del desarrollo de software con el ánimo de descubrir “mejores formas de desarrollar software” en base a su experiencia propia o ayudando a terceros.

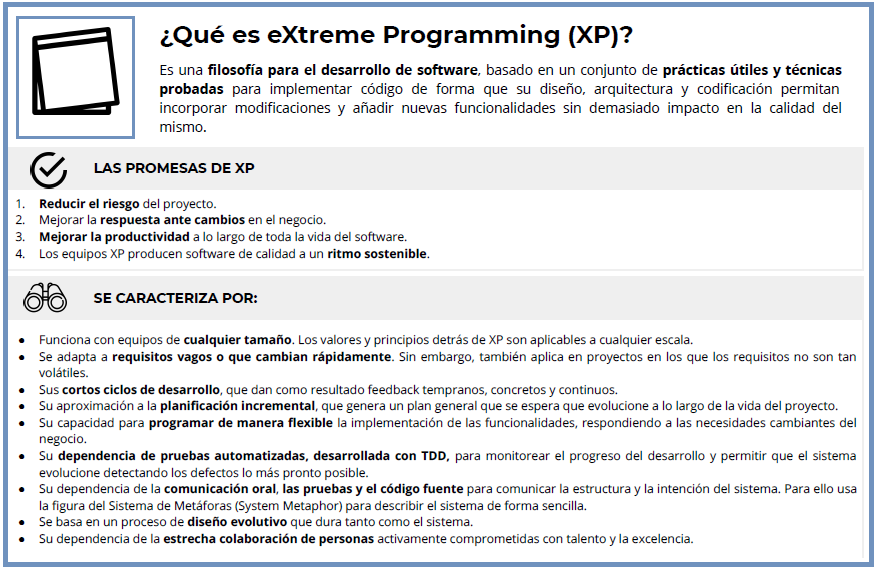
Está compuesto por 4 valores y 12 principios que conforman los pilares del manifiesto Agile.

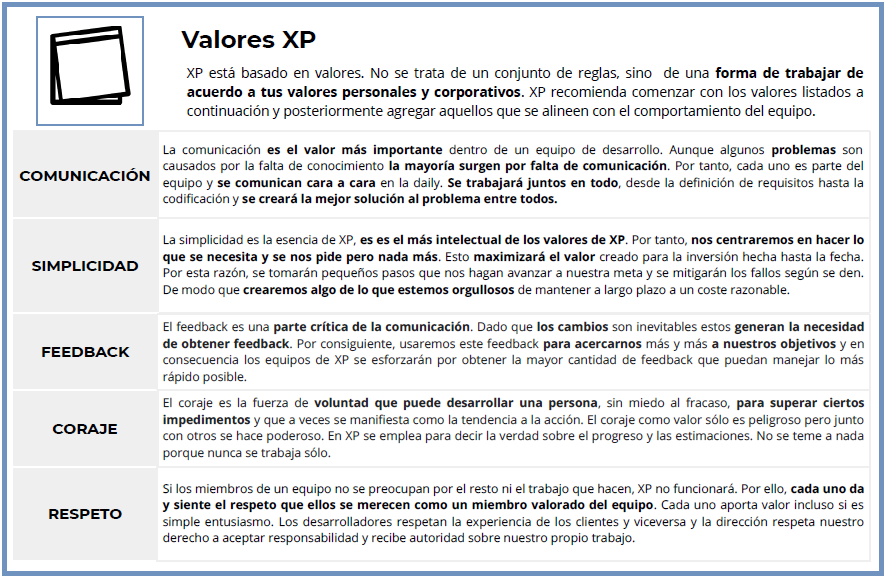
Valores del manifiesto Ágil

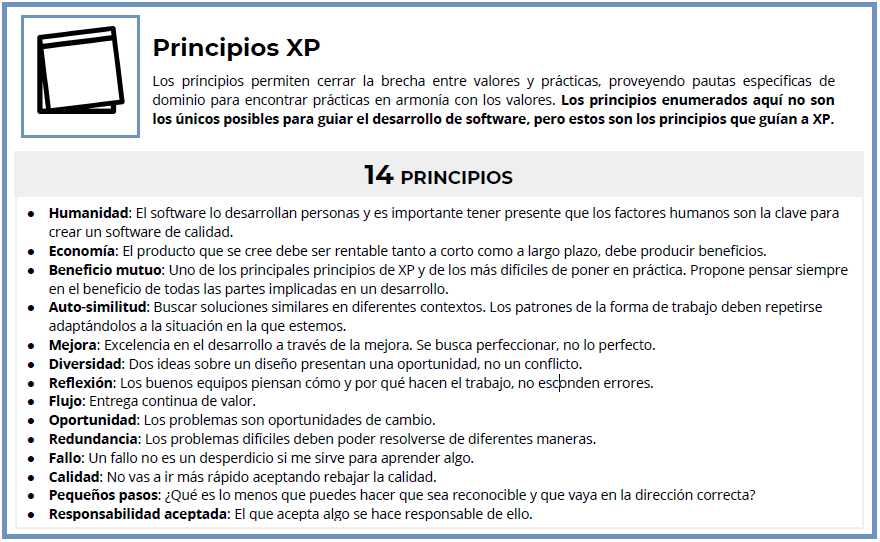
****

****

Metodologías XP

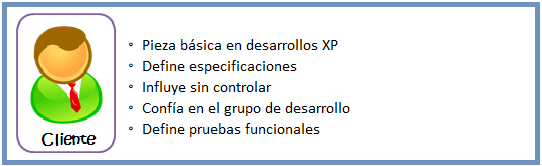


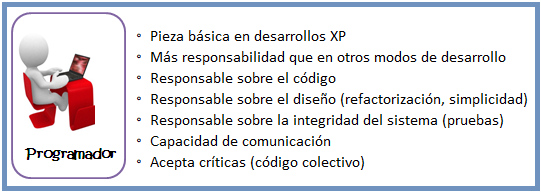


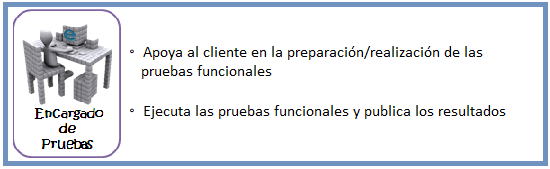


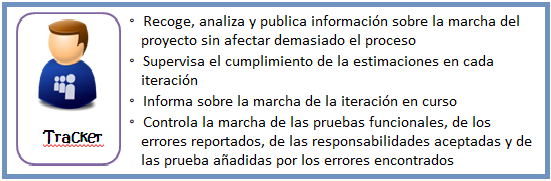
Características

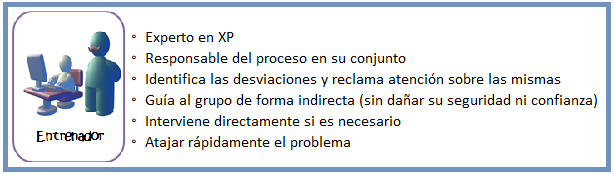
* Se considera al equipo de proyecto como el principal factor de éxito del proyecto
* Software que funciona por encima de una buena documentación.
* Interacción constante entre el cliente y el equipo de desarrollo.
* Planificación flexible y abierta.
* Rápida respuesta a cambios.

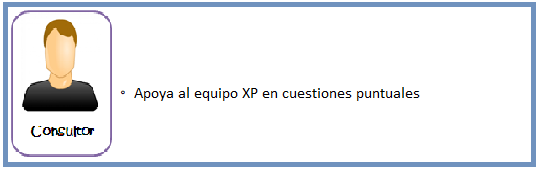
Roles

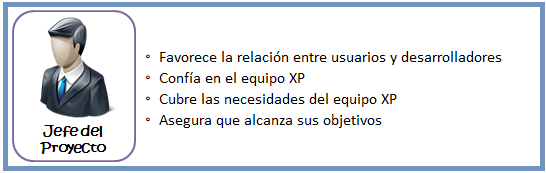






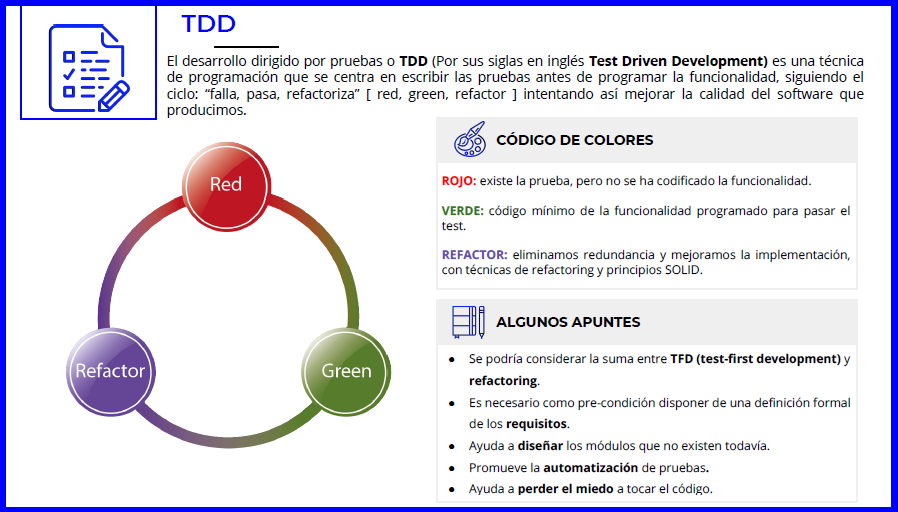






El Extreme Programming tiene como gran ventaja el de la programación organizada y planificada para que no haya errores durante todo el proceso.

TDD (Test Driven Development)- Desarrollo dirigido por pruebas

****

Es una técnica de ingeniería de software dirige el desarrollo de un producto a través de ir escribiendo pruebas, generalmente unitarias.

Las tres leyes del TDD

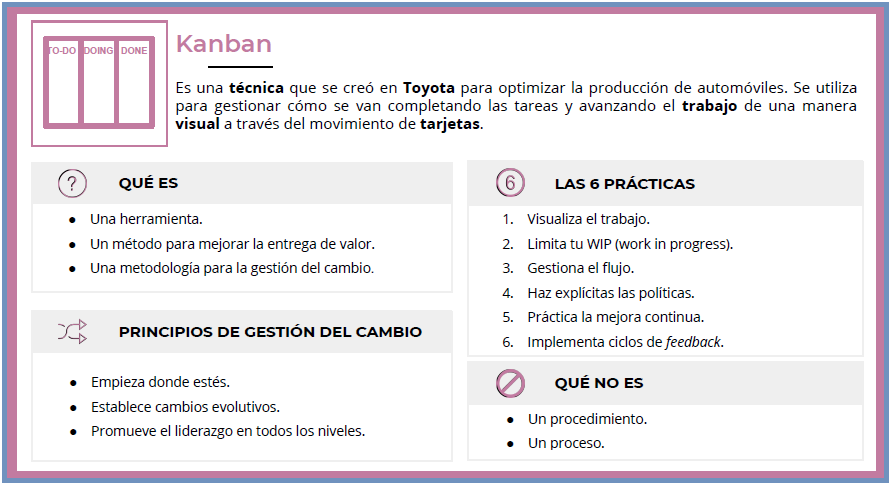
* No escribirás código de producción sin antes escribir un test que falle.
* No escribirás más de un test unitario suficiente para fallar (y no compilar es fallar).
* No escribirás más código del necesario para hacer pasar el test.

Estas tres leyes derivan en la repetición de lo que se conoce como el ciclo Red-Green-Refactor



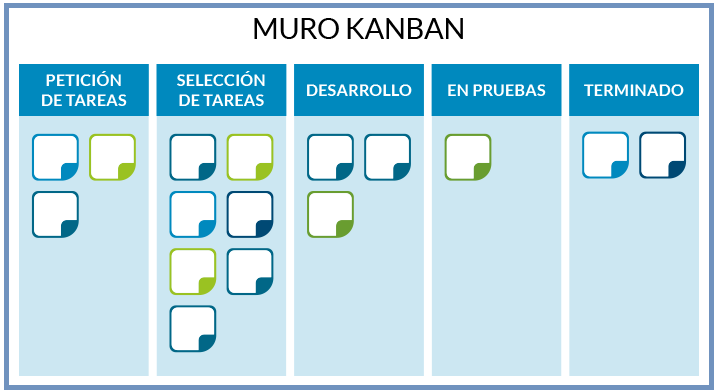
TDD como herramienta de diseño. - Para ello, una vez que tenemos una lista con los primeros requisitos que debe satisfacer el producto, debemos seguir los siguientes pasos:





El trabajo se divide en partes, normalmente cada una de esas partes se escribe en un posit y se pega en una pizarra. Los post-it suelen tener información variada, si bien, aparte de la descripción, debieran tener la estimación de la duración de la tarea.

La pizarra tiene tantas columnas como estados por los que puede pasar la tarea.



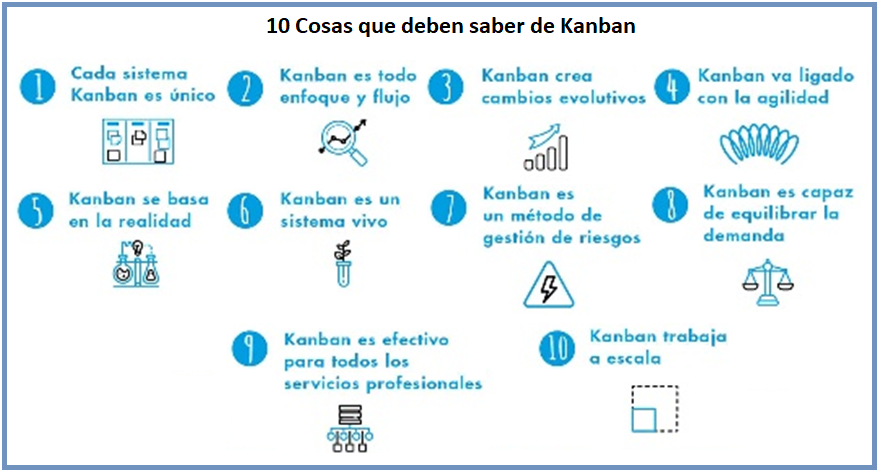
**Framework Scrum**

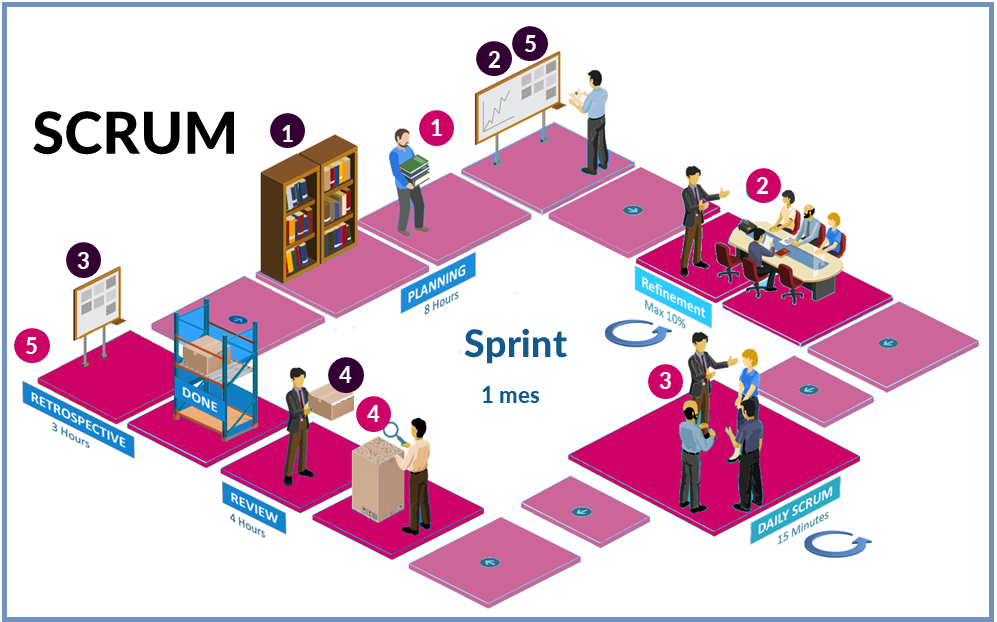
Es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo.

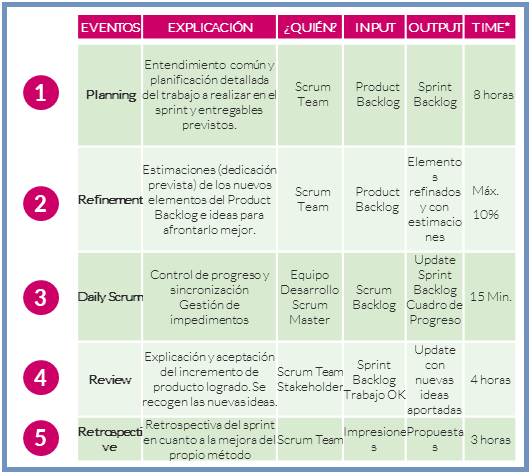
El marco de Scrum, tal como se define en la Guía SBOKTM, está estructurado de tal manera que es compatible con los productos y el desarrollo de servicio en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad. Una fortaleza clave de Scrum radica en el uso de equipos multifuncionales, auto organizados y con poder, que dividen su trabajo en ciclos de trabajo cortos y concentrados llamados Sprints.

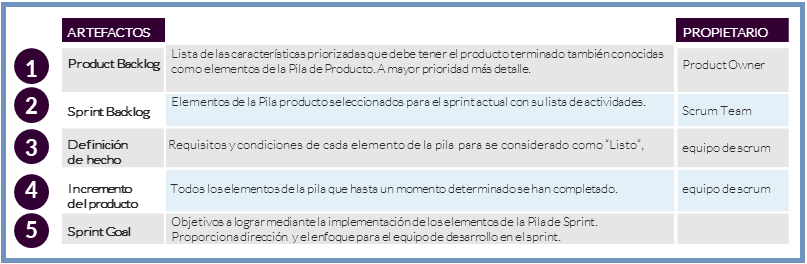
Por tanto, en los proyectos Scrum:

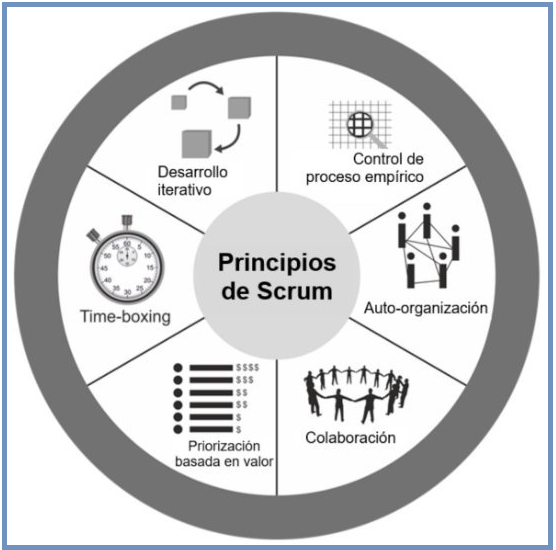
* Se define la necesidad del negocio.
* Se priorizan los requerimientos.
* Se genera el Backlog a trabajar en el sprint.
* Se tienen reuniones de seguimiento y avance del desarrollo.
* Se generan los entregables del sprint.













Características del equipo de desarrollo Scrum

* Multifuncional: Es capaz de realizar cada una de las historias de usuario del Product Backlog de principio a fin, siguiendo el definition of done.
* Auto-organizado: Nadie, ni siquiera el Scrum Master, tiene que decirle al equipo de desarrollo cómo tiene que convertir cada elemento del Product Backlog en incremento de producto.
* Se recomienda que no existan “títulos” dentro del equipo, para evitar responsabilidades únicas en elementos del Product Backlog.
* No existen sub-equipos dentro del equipo de desarrollo.
* Cada miembro del equipo puede ser especialista en un área en la cual se enfoque, pero la responsabilidad de cualquier fallo recae en todo el equipo.

Responsabilidades del equipo Scrum

* Son responsables íntegros de convertir los elementos del Product Backlog en incrementos de producto. Nadie debe decirles cómo hacer este trabajo.
* Son responsables de realizar las estimaciones de cada elemento del Product Backlog.
* Son responsables del Sprint Backlog, no así del Product Backlog, responsabilidad del Product Owner.
* Son responsables de solucionar cualquier problema interno dentro del equipo.
* Son responsables de asistir a cada Daily Scrum Meeting, es el único rol dentro de Scrum que tiene que asistir sí o sí a este evento.

Tamaño recomendado de un Equipo Scrum

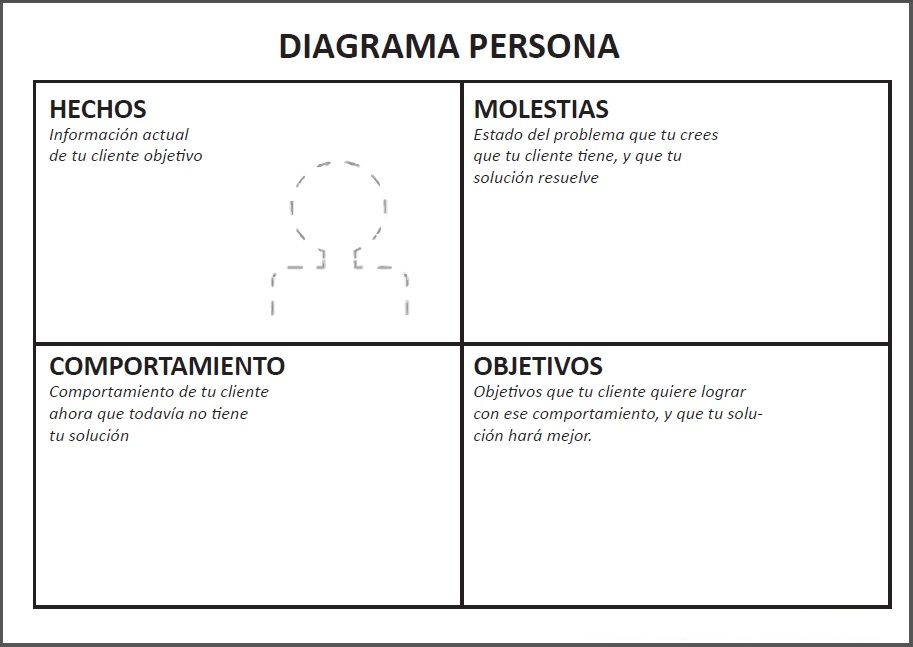
El tamaño de un equipo de desarrollo Scrum debe ser lo suficientemente bajo como para mantener la agilidad, y lo suficientemente alto como para poder completar un incremento de producto aceptable en el contexto del proyecto.

El tamaño recomendado es entre 3 y 9 personas. Menos de 3 difícilmente pueda ser llamado un equipo de desarrollo Scrum. En caso de ser más de 9 personas, termina siendo muy difícil poder gestionar todos los eventos de Scrum: Sprint Planning, Review y Retrospectiva. Equipos grandes encuentran difícil tomar decisiones dentro de los tiempos que duran los eventos.

El Scrum Master y el Product Owner no están incluidos en este número de personas, aunque ellos también participen en el desarrollo de producto.

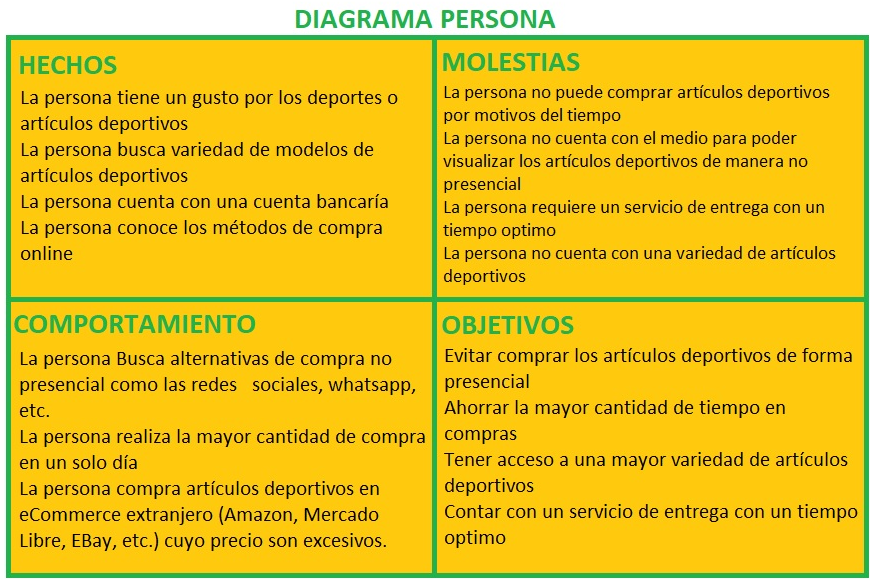
Diagrama Persona

Herramienta que permite estudiar a fondo a un cliente “prototipo”. Contemplando los siguientes elementos: Hechos, Molestias, Comportamiento y Objetivo



**Ejemplo: Caso\_01**

El presente proyecto eCommerce y mCommerce está dirigida a empresas dedicadas a la venta de artículos deportivos que permitirá un mejor posicionamiento en el mercado nacional.



ACTIVIDAD VIRTUAL:

Ingresa a la plataforma virtual, luego responde las siguientes preguntas propuestas:

a) CUESTIONARIO TÉCNICO

1. En la gestión de proyectos. ¿Qué metodología debe usar? ¿Predictiva o Ágil? Analizar en grupo su proyecto Solución Web y justificar en forma detallada su proyecto y cual paradigma correspondería ¿Cuándo aplicar la metodología en los dos escenarios Predictivo y Ágil?

|  |  |
| --- | --- |
| **Proyectos Predictivos** | **Proyectos Ágiles** |
|
|  |  |
|
|
|
|
|
|
|
|
|
|
|
|
|

2.Tomando como ejemplo el Caso\_01 de la presente guía. Se pide elaborar el Diagrama Persona del Proyecto Solución Web, que están desarrollando de acuerdo a los grupos de trabajo.

|  |  |
| --- | --- |
| **HECHOS** | **MOLESTIAS** |
|
|  |  |
|
|
|
|
|
|
|
|
|
|
|
|
|
| **COMPORTAMIENTO** | **OBJETIVO** |
|
|  |  |
|
|
|
|
|
|
|
|
|
|
|
|
|

3. De acuerdo a los 5 entornos o dominios de complejidad del Framework Cynefin (simple, complicado, complejo,

caótico, desordenado), explicar y justificar en qué escenario del Framework Cynefin pertenece tu proyecto de Solución Web a desarrollar.

|  |
| --- |
| **Dominio:** ……………………………… |
| <<Características del dominio y su proyecto>> |

b) CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_