*UNIDAD 1.*

***Técnicas de relevamiento:***

Es una revisión, una investigación o un estudio de algo.convierte los datos en hechos.

* Entrevista: Informe referido a un puesto y un individuo en particular.
* Cuestionarios: Involucra más gente, está destinado a conocer varias opiniones sobre un tema.
* Investigación Externa.
* Revisión, lectura y estudio de la documentación.
* Observación directa – observación indirecta.

***Entrevista:***

Esta técnica es empleada por el ENTREVISTADOR y ENTREVISTADO, donde el primero debe planificar la misma. La entrevista, como otras actividades de Relevamiento son utilizadas en distintas etapas o actividades del ciclo de vida de un sistema.

***Entrevista. Planificación:***

Para la planificación se debe hacer un análisis previo al entrevistado, no debe excederse más de 1 hora y se debe conducir la entrevista hacia los temas deseados para el éxito de la misma.

***Entrevista. Ejecución:***

Se debe establecer una relación con el entrevistado, es importantísimo anticiparse a las respuestas mediante la tarea de planificación, se deben evitar preguntas cerradas y por último registrar lo mínimo necesario.

***Entrevista. Consideraciones:***

Planear la entrevista, no abusar de las herramientas, la información recogida debe ser precisa, participación personal del analista para que el entrevistado responda con más libertad.

* **Dato** = Información no verificada.
* **Hecho** = Datos verificados.
* **0pinión** = Comentario que puede ser verdadero o falso.
* **Deducción** = Afirmación que surge indirectamente de la observación de lo hecho.

***Cuestionario:***

Es una técnica de recopilación de la información que permite que el analista estudie diferentes grupos de personas. Con esta técnica se pueden analizar e investigar mediante preguntas cerradas y abiertas para obtener respuestas tanto cualitativas como cuantitativas.

Las respuestas a cuestionarios con preguntas cerradas pueden ser cuantificadas (números y estadísticas), mientras que las abiertas son cualitativas (desarrollo).

“Mediante el uso del cuestionario el analista podrá estar cuantificando o convalidando lo encontrado en las entrevistas.”

***Cuestionario. Planificación:***

Utilizaremos un cuestionario siempre que queramos conocer acerca de actividades vinculadas al procesamiento de información, esto sirve para producir información significativa mediante la acumulación y manipulación de los datos.

***Cuestionario. Redacción de las preguntas:***

El analista debe ser muy claro en las preguntas y debe diseñarlas de manera lenta y entendible para analizarlas luego.

La selección de palabras es fundamental para reflejar la terminología (glosario de palabras o términos de la empresa) del negocio y la jerga de la empresa.

***Cuestionario. Motivación para responderlo:***

Para que se haga el cuestionario el analista debe motivar mostrando cual es el objetivo para realizar el mismo.

***Otras herramientas:***

***Investigación externa:***

El analista debe tener una visión general del sistema y del negocio, por lo que deberá emplear su profesionalismo.

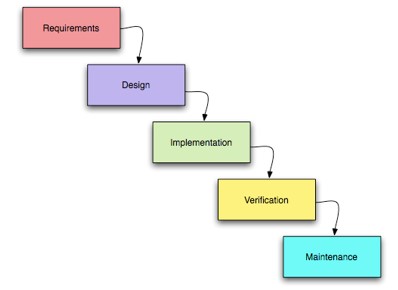
***Revisión, lectura y estudio de la documentación:***

Examinar manuales, procedimientos escritos, reportes, despliegue de plantillas y toda otra documentación de la organización usuaria que resulte de intereses.

***Observación directa:***

El analista interviene en las actividades principales que general el problema.

UNIDAD 2

***Ciclos de vida de desarrollo de software. Actividades dentro del mismo.***

***Ciclo de vida de desarrollo de software. Definición.***

**El ciclo de vida del desarrollo Software, es una secuencia estructurada y bien definida de las etapas en Ingeniería de software para desarrollar el producto software deseado.**

Hay distintos modelos para planificar el desarrollo de software.

***Ciclo de vida de desarrollo de software. Metodología.***

La metodología es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas probabilidades de éxito.

La **Norma 12207 de ISO** lo define como: “Un marco de referencia que contiene las actividades.”

La **IEEE 1074** lo define como: “Una aproximación lógica.”

***Ciclo de vida de desarrollo de software. Ventajas de los procesos de desarrollo.***

* **Todos los integrantes del equipo del proyecto trabajan bajo un marco común.**
* Estandarización de conceptos, actividades y nomenclatura.
* Actividades de desarrollo apoyadas por procedimientos y guías.
* Resultados de desarrollo predecibles.
* Uso de herramientas de ingeniería de software.
* Planificación de las actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
* Recopilación de mejores prácticas para proyectos futuros.

*Mejora de los productos:*

* + Se asegura que los productos cumplen con los objetivos de calidad propuestos.
  + Detención temprana de errores.
  + Se garantiza la trazabilidad de los productos a lo largo del proceso de desarrollo.

*Mejoras en la relación cliente usuario:*

* El cliente percibe el orden en los procesos.
* Facilita al cliente el seguimiento de evolución del proyecto.
* Se establecen mecanismos para asegurar que los productos desarrollados cumplan con las expectativas del cliente

***Ciclo de vida de desarrollo de software. Modelos de ciclos de vida.***

*Metodologías:*

* **Prescriptivas o tradicionales:** Mayor énfasis en la planificación y control del proyecto. Impone una disciplina de trabajo eficaz. Se centra en el control del proceso.
* **Cascada o (Waterfall):** **Secuencia ordenada (cada fase comienza cuando termina la anterior, se cumplan los objetivos y se hagan las pruebas de la misma).** Ayuda a revenir que se sobrepase la fecha de entrega y los costos esperados. Admite iteraciones (Se permite volver a una etapa anterior del proyecto). Es complicado detectar los errores y es poco flexible. Se utiliza cuando se disponen de todos los requerimientos.
* **Modelo V:** **Aporta opciones de evaluación del software en cada etapa de manera inversa**, de esta manera se puede monitorear el avance y acompañarlo con validaciones permanentes. El brazo izquierdo de la letra **V** contiene las tareas de diseño y **desarrollo** del sistema, y el derecho las medidas de **control de calidad** de cada fase.
* **Modelo Sashimi:** Es una variación del ciclo en cascada. **Se permite un solapamiento entre fases (Sin tener terminado el diseño se comienza a implementar).** No necesita generar mucha documentación. Se vuelve complicado controlar el progreso del proyecto porque los finales de fase ya no son un punto de referencia.
* **Modelo Evolutivo:** consta del desarrollo de una versión inicial que luego de exponerse se va refinando de acuerdo de los comentarios o nuevos requerimientos por parte del cliente o del usuario final. **Este modelo acepta que los requerimientos del cliente cambien en cualquier momento.** El problema es con la iteración de los ciclos de requerimiento, desarrollo y evaluación. Puede ser útil cuando se desconocen la mayoría de los requerimientos iniciales.
* **Modelo Iterativo:** **Busca reducir los desentendimientos cuando se entrega el producto final, debido a la toma de requerimientos inicial.** Al final de cada iteración se le entrega al cliente una versión mejorada. El cliente es quien propone mejoras luego década iteración. Las mismas se repiten hasta que el cliente quede satisfecho. Se utiliza en las apps medianas o grandes.
* **Modelo por Prototipos:** Usado cuando no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa. Se suele recurrir a las especificaciones iniciales. **El objetivo es realizar un producto intermedio antes de realizar el producto final.** Se utiliza para el desarrollo de productos con innovaciones importantes. **Es el único apto para desarrollos que a priori no se conocen sus especificaciones.** La desventaja es su alto valor.
* **Modelo Incremental:** **Se basa en la filosofía de construir incrementando las funcionalidades del programa.** Se realiza construyendo módulos que cumplen diferentes funciones del sistema. Facilita el desarrollo. Al final de cada ciclo se le entrega al cliente la nueva funcionalidad. Se permite realizar una entrega al cliente antes de terminarlo. Si se detecta un error, se deshecha la última iteración. No se necesitan todos los requerimientos al principio del proyecto.
* **Modelo en Espiral:** **Se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final.** Tiene los mismos beneficios que los ciclos de vida incremental y de prototipo. A medida que el ciclo se cumple (el avance de la espiral), se van obteniendo software que va ganando la satisfacción del cliente. Tiene un bajo riesgo de retraso en la detección de errores. El costo temporal que suma cada vuelta de la espiral.

*El mismo se divide en 4 etapas:*

**Planificación 🡪** Relevamiento de requerimientos iniciales o luego de una iteración.

**Análisis de riesgos 🡪** De acuerdo con el relevamiento de requerimientos, se decide si se continúa con el desarrollo.

**Implementación 🡪** Desarrollo de un prototipo basado en los requerimientos.

**Evaluación 🡪** El cliente evalúa el prototipo, si da su conformidad, termina el proyecto. En caso contrario, incluimos los nuevos requerimientos solicitados por el cliente en la siguiente iteración.

***Metodologías Agiles.***

Una metodología ágil es una innovadora forma de trabajar y organizar flujos, que divide los proyectos en partes, permite adaptarse sobre la marcha, complementa y resuelve etapas en poco tiempo.

¿Cuál es la diferencia entre Scrum y Metodologías Agiles?

Las metodologías son principios que se utilizan en proyectos, mientras que, en Scrum son reglas definidas que se deben seguir para la realización del proyecto.

El origen de las metodologías agiles.

Se comenzaron a aplicar en los 90´. En 2001 se reunieron 17 profesionales y crearon una organización sin fines de lucro llamada **“Agile Alliance”** con el objetivo de promover los valores y principios de la filosofía ágil.

El manifiesto Ágil – (4 valores y 12 principios).

**Individuos e interacciones sobre** **procesos y herramientas**

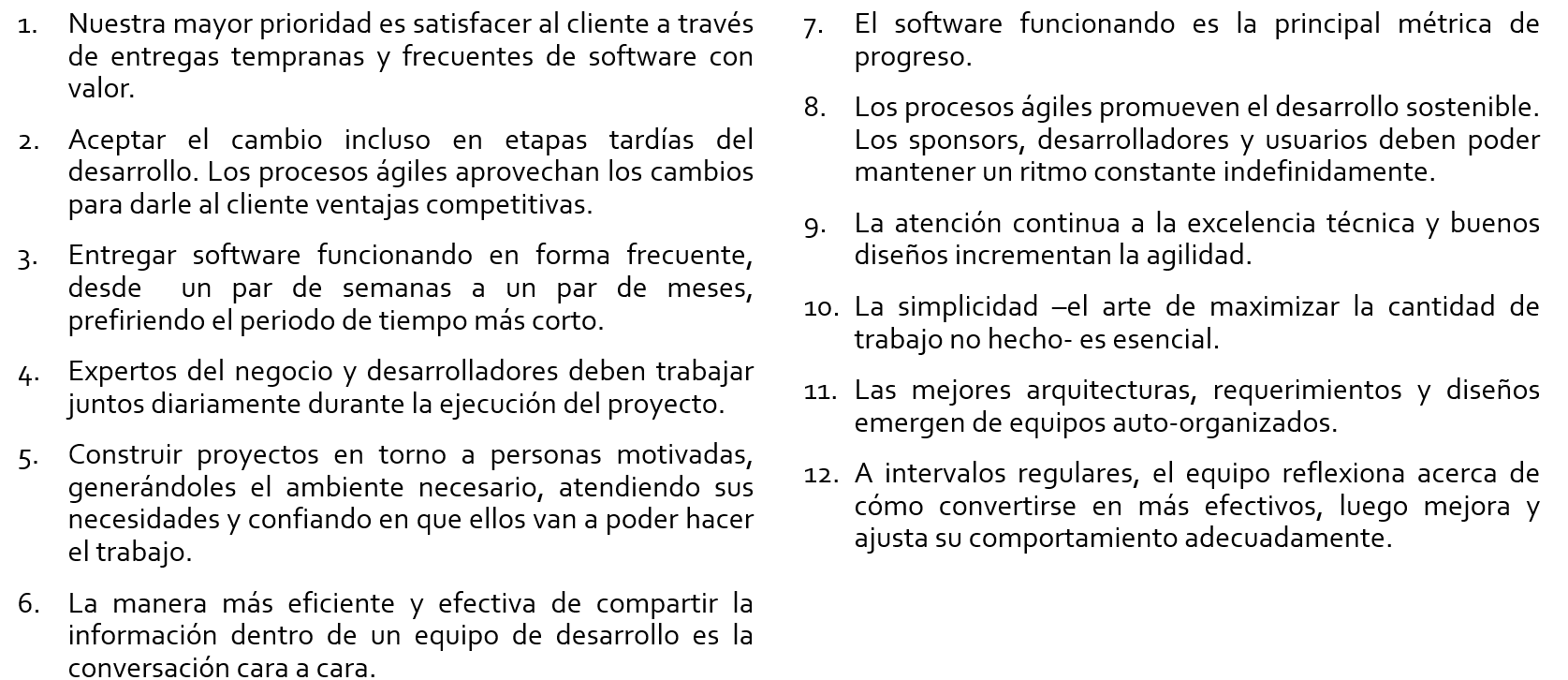
**Software funcionando sobre** **documentación extensiva**

**Colaboración con el cliente sobre** **negociación contractual**

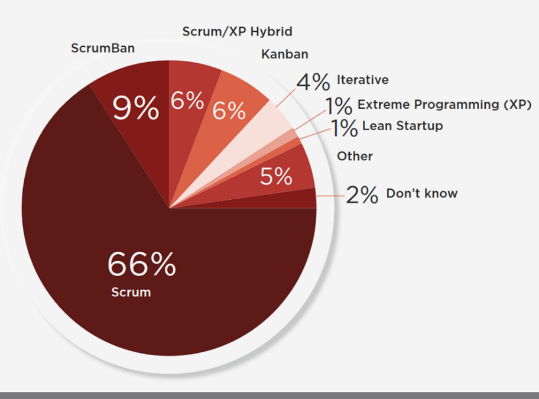
**Respuesta ante el cambio sobre** **seguir un plan**

Aunque valoramos los elementos de la derecha, valoramos mas los de la izquierda.

**Los 12 principios del manifiesto ágil:**



Técnicas agiles y madurez.



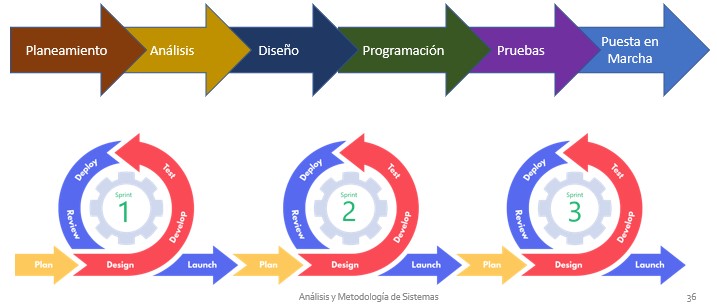
Scaling Agile.

Marco Ágil Escalado (SAFe)

SAFe® **combina las prácticas Lean, Agile y DevOps para la agilidad del negocio**. Proporciona orientación sobre tres niveles para la entrega de productos en un entorno ágil a escala y añade orientación sobre la ampliación de la agilidad en toda la empresa con su cuarto nivel, el de la cartera.

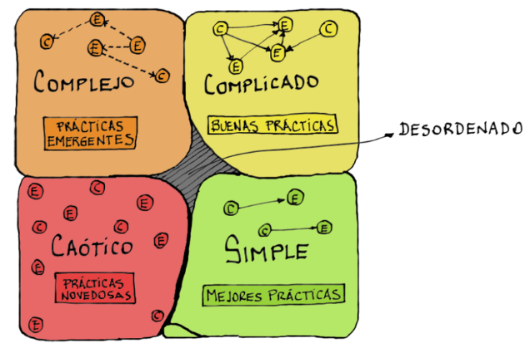
Durante las últimas cinco encuestas, el uso de SAFe® ha crecido significativamente para convertirse en el enfoque dominante, en uso por más de un tercio de los encuestados.

*Metodología* ***Tradicional*** *VS.* ***Agile****.*





Cynefin Framework.

Es un marco de trabajo utilizado para facilitar la toma de decisiones, otorgando a quienes toman decisiones ***cinco dominios***, en donde pueden clasificar cualquier problema o necesidad que tengan que resolver y cuál es, según este enfoque, la manera más eficiente de responder a cada una de ellas.

Dominio simple.

* Se opera con problemáticas simples.
* Es muy fácil identificar las causas y sus efectos.
* La respuesta correcta es clara, conocida por todos e indiscutible.
* Existen las mejores prácticas, soluciones conocidas para problemas conocidos.
* Detectar, Clasificar y Responder

Dominio complicado.

* Encontramos problemas complejos, buenas prácticas y perfiles expertos.
* Hay múltiples soluciones correctas para una misma problemática, pero se requiere del involucramiento de expertos para poder identificarlas.
* Una práctica habitual es el mantenimiento de sistemas y soporte técnico.
* Detectar, Analizar y Responder

Dominio complejo.

* Cuando nos enfrentamos a problemas complejos, los resultados se vuelven más impredecibles.
* No existen ni mejores ni buenas prácticas catalogadas.
* No sabemos con anticipación si una determinada solución va a funcionar. Solo podemos examinar los resultados y adaptarnos.
* Este es el dominio de las prácticas emergentes.
* Experimentar, Detectar y Responder

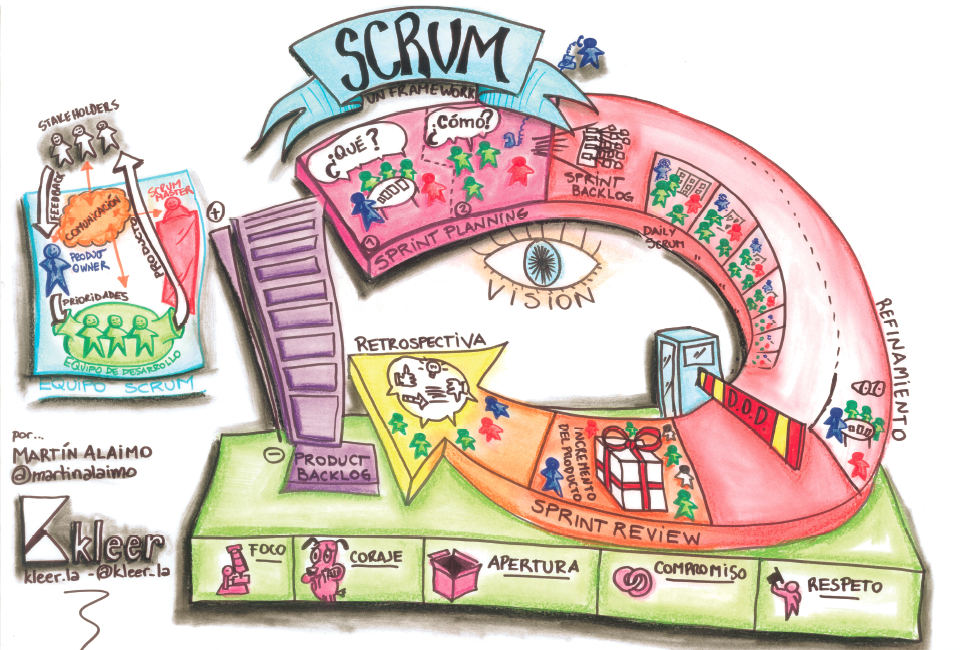
Domino caótico.

* Los problemas caóticos requieren una respuesta inmediata.
* Se está en crisis y es necesario actuar de inmediato para restablecer cierto orden.
* Este es el dominio de la improvisación.
* Actuar, Detectar, Responder

Domino desordenado.

* No sabemos en qué dominio estamos.
* Zona Peligrosa: no podemos medir las situaciones ni determinar la forma de actuar.
* El gran peligro es actuar de manera diferente a la que se necesita para resolver ciertos problemas.
* Todo el esfuerzo debe estar enfocado netamente a salir de este espacio hacia uno mejor identificado, para luego actuar de la manera en que dicho dominio lo requiera.

***SCRUM.***



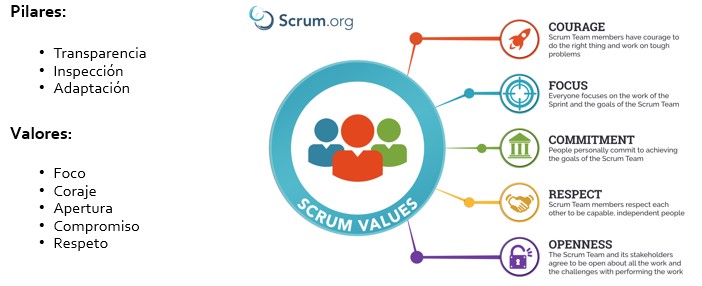
Un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos.

***Scrum es:***

es un conjunto específico de reglas **que** deben seguirse mientras se practica el desarrollo de software.

**NO es un proceso, una técnica o método definitivo.** Se basa en la teoría de control de procesos empírica o empirismo. Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Consiste en los equipos Scrum y sus roles, eventos, artefactos y reglas asociadas.

***Los pilares y valores SCRUM.***



**Transparencia:** La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se están viendo.

**Inspección:** Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo para detectar variaciones indeseadas.

**Adaptación:** Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables y que el producto resultante será inaceptable, el proceso o el material que está siendo procesado deben ajustarse.

**Foco**: Los Equipos Scrum se enfocan en un conjunto acotado de características por vez.

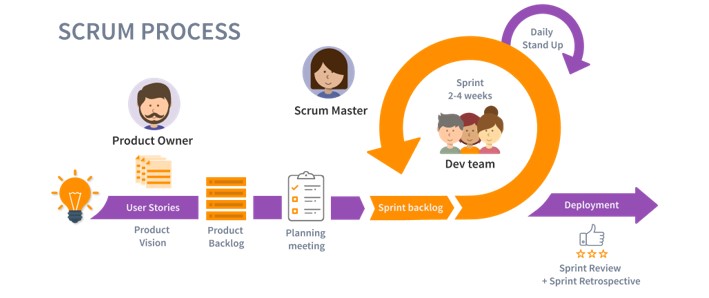
**Coraje:** Debido a que los Equipos Scrum trabajan como verdaderos equipos, pueden apoyarse entre compañeros, y así tener el coraje de asumir compromisos desafiantes que les permitan crecer como profesionales y como equipo.

**Apertura:** Los Equipos Scrum privilegian la transparencia y la discusión abierta de los problemas. No hay agendas ocultas ni triangulación de conflictos. La sinceridad se agradece y la información está disponible para todos, todo el tiempo.

**Compromiso:** Los Equipos Scrum tienen mayor control sobre sus actividades, por eso se espera de su parte el compromiso profesional para el logro del éxito.

**Respeto:** Debido a que los miembros de un Equipo Scrum trabajan de forma conjunta, compartiendo éxitos y fracasos, se fomenta el respeto mutuo, y la ayuda entre pares es una cuestión a respetar.

***El proceso SCRUM.***



***Se divide en:***

* *Product Backlog: Son los requerimientos.*
* *Sprint Plaining: Se inspecciona el backlog.*
* *Sprint Backlog: Son las tareas definidas por cada sprint, deben ser completadas.*
* *Sprint: Son periodos de breve tiempo donde el equipo trabaja para completar la cantidad de trabajo establecido. (1 – 4 semanas)*
* *Incrementos de producto terminado: Son las implementaciones de los objetivos terminados, esto cubre las necesidades del cliente.*
* *Sprint review: Demostración del prototipo del producto, determinar que objetivos se cumplieron y cuáles no.*

***El equipo SCRUM.***

**El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master.** Los Equipos Scrum son autoorganizados y multifuncionales.

***El dueño del producto. (PRODUCT OWNER)***

El Dueño de Producto es el responsable de maximizar el valor del producto resultante del trabajo del Equipo de Desarrollo.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). *La gestión de la Lista del Producto incluye:*

• Expresar claramente los elementos de la Lista del Producto

• Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible

• Optimizar el valor del trabajo que el Equipo de Desarrollo realiza

• Asegurar que la Lista del Producto es visible, transparente y clara para todos y que muestra aquello en lo que el equipo trabajará a continuación

• Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

***El equipo de desarrollo. (DEVELOPMENT TEAM)***

**El Equipo de Desarrollo consiste en los profesionales que realizan el trabajo de entregar un Incremento de producto “Terminado” que potencialmente se pueda poner en producción al final de cada Sprint.** Un Incremento “Terminado” es obligatorio en la Revisión del Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

*Los Equipos de Desarrollo tienen las siguientes características:*

• Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegables

• Los Equipos de Desarrollo son multifuncionales, esto es, como equipo cuentan con todas las habilidades necesarias para crear un Incremento de producto

• Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo independientemente del trabajo que realice cada persona

• Scrum no reconoce subequipos en los equipos de desarrollo, no importan los dominios que requieran tenerse en cuenta, como pruebas, arquitectura, operaciones o análisis de negocio

• Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

*Tamaño del equipo de desarrollo.*

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. ***Entre 3 (si es menor se reducen la interacción y ganancias) y 9 (si es mayor requiere mucha coordinación).***

***El Scrum master.***

El Scrum Master es un líder que está al servicio del Equipo Scrum. **El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser útiles y cuáles no.** El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

*El Scrum Master da servicio al Dueño de Producto de varias formas, incluyendo:*

• Asegurar que los objetivos, el alcance y el dominio del producto sean entendidos por todos en el equipo Scrum de la mejor manera posible.

• Encontrar técnicas para gestionar la Lista de Producto de manera efectiva.

• Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos.

• Entender la planificación del producto en un entorno empírico.

• Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor.

• Entender y practicar la agilidad.

• Facilitar los eventos de Scrum según se requiera o necesite.

El Servicio del Scrum Master al Equipo de Desarrollo El Scrum Master da servicio al Equipo de *Desarrollo de varias formas, incluyendo:*

• Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional.

• Ayudar al Equipo de Desarrollo a crear productos de alto valor.

• Eliminar impedimentos para el progreso del Equipo de Desarrollo.

• Facilitar los eventos de Scrum según se requiera o necesite.

• Guiar al Equipo de Desarrollo en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

*El Servicio del Scrum Master a la Organización El Scrum Master da servicio a la organización de varias formas, incluyendo:*

• Liderar y guiar a la organización en la adopción de Scrum.

• Planificar las implementaciones de Scrum en la organización.

• Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto.

• Motivar cambios que incrementen la productividad del Equipo Scrum.

• Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

UNIDAD 3

Documento de toma de requerimientos del cliente:

Se trata, en resumen, de un documento que **describe qué diseñarás, por qué y cómo**. Su objetivo es articular de manera clara y sin ambigüedades el propósito del producto, sus características, funcionalidad y comportamiento.

Este documento describe como vas a diseñar el producto y por qué. El objetivo de este documento es que no haya diferencias de lo descrito en el mismo con el producto final (características, funcionalidad…).

**1-Objetivo:** El documento tiene por objetivo, ***identificar la funcionalidad del sistema***afectada por el proyecto.

**2 – Alcance:** Define el alcance del sistema, detalla las actividades que serán incluidas y las actividades excluidas del proyecto.

**2.1 Actividades incluidas:** definir actividades que incluyen en la propuesta de solución del proyecto.

**2.2 Actividades identificadas fuera del alcance:** Enumerar todas aquellas actividades que quedan fuera del alcance del proyecto.

**3- Situación Actual:** Se debe definir cuál es la situación actual del cliente y que lo lleva a necesitar una nueva solución.

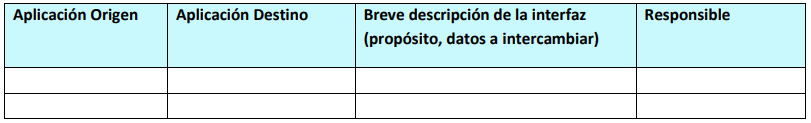
**4- Requerimientos del cliente (Requisitos):** Enumerar los requisitos del cliente, estos no son una solución, sino que detallan los problemas a resolver.

**5- Requerimientos no funcionales:** Enumerar aquellos requerimientos que hay que atender, pero no afectan a un caso de uso en cuestión. Son restricciones técnicas a las funcionalidades definidas.

**6- Requisitos Funcionales del Sistema:**

1. Modelo de Proceso de Negocio (BMP). Actividades del proceso de negocio.
2. Casos de Uso. Funcionalidad de subprocesos o tareas.

**7- Interfaces con otros Sistemas:** Utilizar otras interfaces como ejemplo y ayuda para la realización de la misma.  
(ej: una base de datos de un banco necesita una tabla de personas con su respectiva info).



**8 – Glosario de Términos:** Enumerar aquellos términos que se consideren importantes, para el entendimiento por parte del cliente.

**9 – Anexos:** Se adjunta toda la documentación de soporte que sea necesaria.

**Casos de uso:** Los casos de uso son un tipo de requerimientos utilizados para especificar funcionalidad.

**¿Para qué sirven los Casos de Uso?**

Con ellos, se obtienen los requerimientos, se define la arquitectura, sus pautas de diseño y sus pruebas.

Los modelos CU se dividen en:

# Diagrama

# Descripción (texto narra pasos)

**¿Qué son los actores y cómo identificarlos?**

**Actor:** algo o alguien fuera del Sistema que interactúa con el Sistema – Representa un rol de usuario.

**¿Qué son Casos de Uso y cómo identificarlos?**

Los Casos de Uso representan lo que el actor desea que haga el sistema. Se identifican mediante las funcionalidades que queremos que tenga el sistema, los casos de uso están escritos desde el punto de vista del actor.

**¿Como especificar Casos de Uso?**

Para especificar los Casos de Uso debemos describir sus partes para lograr una descripción completa.

**Precondiciones:**

Son un conjunto de condiciones que se deben cumplir para iniciar un caso de uso.

**Postcondiciones:**

Refleja el estado del sistema una vez ejecutado el caso de uso.

**Observaciones / Notas**

Reflejan el análisis y el funcionamiento del sistema. Se utilizan para aplicar explicaciones o detalles del mismo.

**La restricción ACID sobre los Casos de Uso**

Pedimos que en su definición los casos de uso cumplan la condición de ACID.

Atómicos: Se cumplen si o si, sino no.

Cohesivos: Poseen una única responsabilidad.

Aislados (Isolation): Deben ser aislados entre sí.

Durables: Si no se modifican las reglas de negocio se preserva en el tiempo.

**Casos de uso de negocio (CUN)**

Los casos de uso de negocio sirven para clarificar el ambiente en el que será utilizado el sistema. Son “de negocio” porque estos casos de uso son externos al negocio, pueden ser clientes, accionistas, etc.

**Requerimientos trazados**

formas de enlazar los CU o especificaciones funcionales con el resto de las especificaciones no funcionales.

**Nivel de detalle esperado**

Se espera que el nivel de detalle del caso de uso no contemple detalles sobre la interfaz, ni la arquitectura, ni el procesamiento interno de los datos cuando estos no sean concernientes a los involucrados. Las redacciones del documento deben ser cortas, concisas y completas.