GESTIÓN DE SISTEMAS DE INFORMACIÓN

Unidad 2:Metodología de análisis, diseño e implementación de SI

La metodología de análisis, diseño e implementación de los sistemas de información suele seguir los siguientes pasos:

1. **Análisis de requisitos**: Recopilar y analizar información sobre los objetivos, los procesos y los datos necesarios para el sistema.
2. **Diseño**: Crear un modelo conceptual y una arquitectura detallada del sistema, incluyendo los procesos, la base de datos y la interfaz de usuario.
3. **Desarrollo**: Codificar el sistema de acuerdo al diseño y realizar pruebas para asegurar su correcto funcionamiento.
4. **Implementación**: Instalar y configurar el sistema en el entorno productivo y capacitar a los usuarios.
5. **Mantenimiento**: Realizar cambios y mejoras en el sistema a lo largo del tiempo para mantener su funcionamiento y adaptarlo a las necesidades cambiantes de la empresa.

Es importante seguir una metodología rigurosa para asegurar la calidad y la eficiencia del sistema, así como para mitigar los riesgos y asegurar su éxito a largo plazo.

Unidad 2: Analisis de Sistemas

El análisis de sistemas es el ***análisis de un problema*** que una empresa trata de resolver mediante un sistema de información. Consiste en ***definir el problema***, identificar sus causas, especificar la solución e identificar los requerimientos de información que debe cumplir una solución de sistemas.

El analista de sistemas crea un mapa de la organización y los sistemas existentes, en el cual se identifica a los propietarios y usuarios principales de los datos, junto con el hardware y software existente. Entonces, el analista de sistemas ***detalla los problemas*** de los sistemas existentes. Al examinar los documentos, papeles de trabajo y procedimientos, observar las operaciones del sistema y entrevistar a los usuarios clave de los sistemas, el analista puede i***dentificar las áreas problemáticas y los objetivos*** que lograría una solución.

El proceso de análisis de sistemas ***identifica varias soluciones*** alternativas para la organización y ***evalúa la viabilidad*** de cada una de ellas. Un informe por escrito de propuesta de sistemas describe los costos y beneficios, además de las ventajas y desventajas de cada alternativa. Es responsabilidad de la gerencia determinar qué mezcla de costos, beneficios, características técnicas e impactos organizacionales representa la alternativa más deseable.

Unidad 2: Diseño de Sistemas

El diseño de sistemas muestra ***cómo cumplirá con este objetivo***. El diseño de un sistema de información es el plan o modelo general para ese sistema. Al igual que el plano de construcción de un edificio o de una casa, consiste en todas las especificaciones que dan al sistema su forma y estructura.

El diseñador de sistemas ***detalla las especificaciones*** del sistema que ofrecerán las funciones que se identificaron durante el análisis de sistemas. Estas especificaciones deben lidiar con todos los componentes administrativos, organizacionales y tecnológicos de la solución del sistema.

Al igual que las casas o los edificios, los sistemas de información pueden tener muchos posibles diseños. Cada diseño representa una mezcla única de todos los componentes técnicos y organizacionales. Lo que hace que un diseño sea superior a los demás es la facilidad y la eficiencia con que cumple los requerimientos del usuario dentro de un conjunto específico de restricciones técnicas, organizacionales, financieras y de tiempo.

Unidad 2: Programación

Durante la etapa de programación, las especificaciones del sistema que se prepararon durante la etapa de diseño se traducen en código de programa de software.

Muchas organizaciones ya no necesitan encargarse de su propia programación para los nuevos sistemas. Compran el software que cumple con los requerimientos de un nuevo sistema a través de fuentes externas, como:

- ***Paquetes de software*** de un distribuidor de software comercial.
- ***Servicios de software*** de un proveedor de servicios de aplicación
- ***Subcontratan empresas*** que desarrollan software de aplicación personalizado para sus clientes

de la Ciudad de Buenos Aires

Unidad 2: Prueba

Se debe realizar una prueba exhaustiva y con detalle para determinar si el sistema produce o no los resultados correctos. La prueba responde a la pregunta: "¿el sistema dará los resultados deseados en condiciones conocidas?".

El proceso de prueba consume tiempo: hay que preparar con cuidado los datos de prueba, revisar los resultados y hacer las correcciones en el sistema. En algunos casos, tal vez sea necesario rediseñar partes del sistema. Si se pasa por alto esta etapa los riesgos resultantes son enormes.

- ***Prueba de unidad***: consiste en probar cada programa por separado en el sistema
- ***Prueba de sistema***: evalúa el funcionamiento del sistema de información como un todo.
- ***Prueba de aceptación***: provee la certificación final para usarse en producción. Los usuarios evalúan las pruebas de sistemas y la gerencia las revisa.

de la Ciudad de Buenos Aire

Unidad 2: Implementación

La conversión es el proceso de cambiar del sistema anterior al sistema nuevo. Se pueden emplear cuatro estrategias principales de conversión:

- ***Estrategia paralela***: tanto el sistema anterior como su reemplazo potencial se operan en conjunto durante un tiempo, hasta que todos estén seguros de que el nuevo funciona correctamente.
- ***Estrategia de reemplazo directo***: se sustituye todo el sistema anterior con el nuevo, en un día programado con anterioridad. Es una metodología muy riesgosa.
- ***Estrategia de estudio piloto***: introduce el nuevo sistema sólo a un área limitada de la organización, como un solo departamento o una sola unidad operacional. Cuando esta versión piloto está completa y trabaja de manera uniforme, se instala en el resto de la organización, ya sea de manera simultánea o en etapas.
- ***Estrategia de metodología en fases***: introduce el nuevo sistema en etapas, ya sea con base en las funciones o las unidades organizacionales.

Unidad 2: Mantenimiento

El mantenimiento comprende los cambios en hardware, software, en la documentación, o los procedimientos de un sistema en producción para corregir errores, cumplir con los nuevos requerimientos o mejorar la eficiencia del procesamiento.

Cerca del 20% del tiempo dedicado al mantenimiento se utiliza para ***depurar o corregir problemas*** de emergencia en producción.

Otro 20% tiene que ver con los ***cambios en los datos, archivos, informes***, hardware o software del sistema.

Y el 60% de todo el trabajo de mantenimiento consiste en ***realizar mejoras para los usuarios***, mejorar la documentación y volver a codificar los componentes del sistema para obtener una mayor eficiencia en el procesamiento. La cantidad de trabajo en la tercera categoría de los problemas de mantenimiento se podría reducir de manera considerable por medio de mejores prácticas de análisis y diseño de sistemas.

Unidad 2: Ciclo de Vida

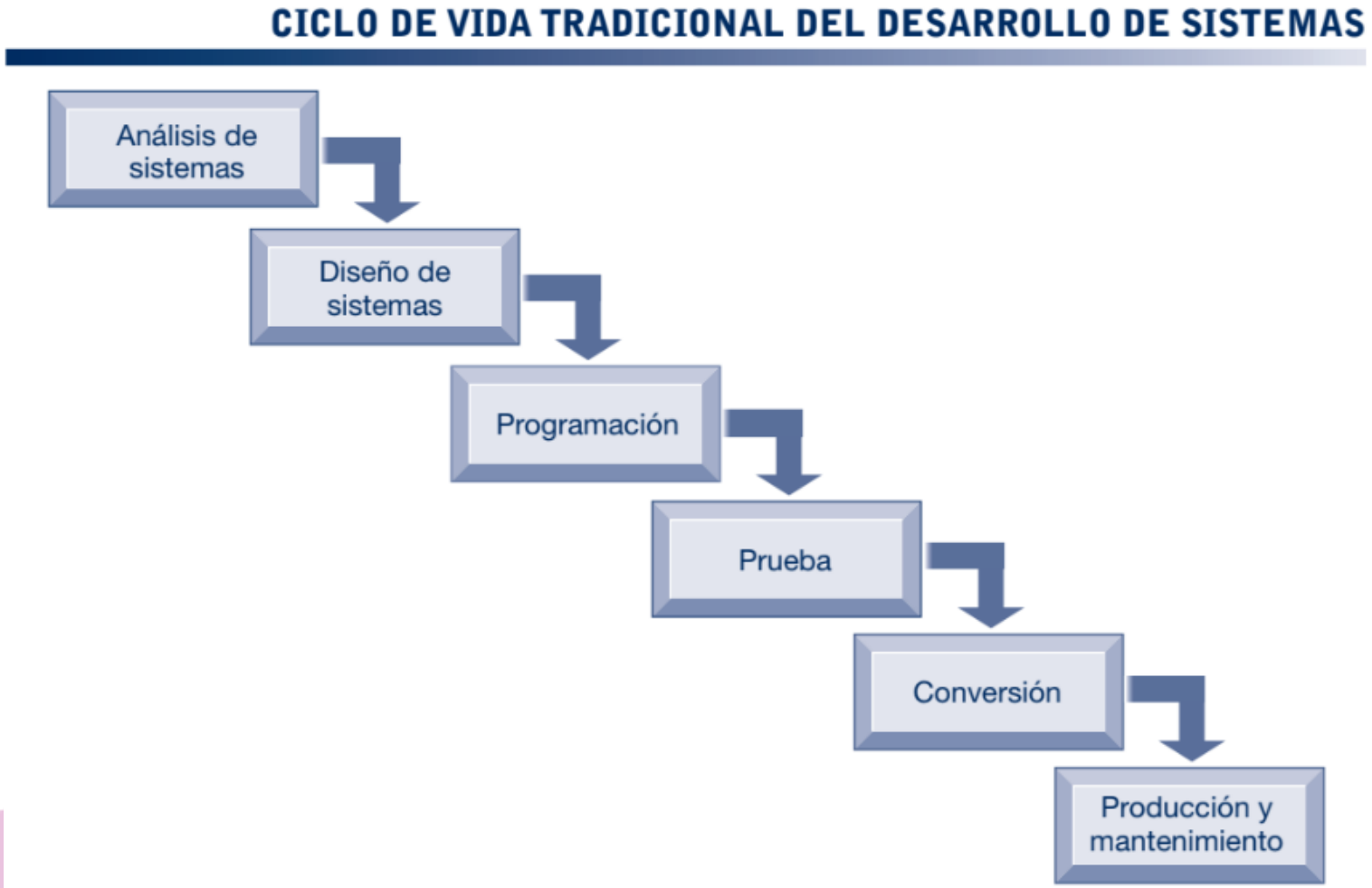
El ciclo de vida de sistemas es el método más antiguo para crear sistemas de información. La metodología del ciclo de vida es un método ***basado en fases*** para la creación de un sistema

Esta metodología mantiene una ***división muy formal*** de la labor entre los ***usuarios finales y los especialistas*** en sistemas de información. Los especialistas técnicos, son responsables de gran parte del trabajo de análisis, diseño e implementación de los sistemas; los usuarios finales se limitan a proveer los requerimientos de información y revisar el trabajo del personal técnico.

El ciclo de vida de sistemas se utiliza para crear ***sistemas complejos extensos*** que necesitan un análisis de requerimientos riguroso y formal, especificaciones predefinidas y controles estrictos sobre el proceso de creación del sistema. Sin embargo, la metodología puede ser ***costosa e inflexible,*** y requerir mucho tiempo.

Esta es una ***metodología de "cascada"*** en la que las tareas en una etapa se completan antes de que empiece el trabajo para la siguiente etapa. Las actividades se pueden repetir, pero hay que generar volúmenes de nuevos documentos y volver a trazar los pasos si es necesario revisar los requerimientos y las especificaciones. Esto contribuye a que se congelen las especificaciones en una etapa muy temprana del proceso de desarrollo.

Unidad 2: Ciclo de Vida



El ciclo de vida de desarrollo de sistemas particiona el desarrollo de sistemas en etapas formales, donde hay que completar cada etapa antes de poder comenzar la siguiente.

Unidad 2: Prototipado

El prototipado consiste en crear un ***sistema experimental con rapidez*** y a un bajo costo para que los usuarios finales lo evalúen. Al interactuar con el prototipo, los usuarios pueden darse una mejor idea de sus requerimientos de información. El prototipo aprobado por los usuarios se puede usar como ***plantilla para crear el sistema final.***

El prototipo es una ***versión funcional*** de un sistema de información o una parte del sistema, pero su único objetivo es ser un ***modelo preliminar***. Una vez operacional, el prototipo se refinará gradualmente hasta que cumpla de manera precisa con los requerimientos de los usuarios. Una vez finalizado el diseño, el prototipo se puede convertir en un reluciente sistema de producción.

El proceso de crear un diseño preliminar, probarlo, refinarlo y probarlo de nuevo se denomina ***proceso iterativo*** del desarrollo de sistemas, debido a que los pasos requeridos para crear un sistema se pueden repetir una y otra vez. Los prototipos son iterativos en un sentido más explícito que el ciclo de vida convencional, además de que promueven activamente los cambios de diseño del sistema. Se dice que los prototipos reemplazan la renovación no planeada con la iteración planeada, donde cada versión refleja de una manera más precisa los requerimientos de los usuarios.

Unidad 2: Prototipado

Pasos en el prototipado

El proceso de creación de prototipos consiste en lo siguiente:

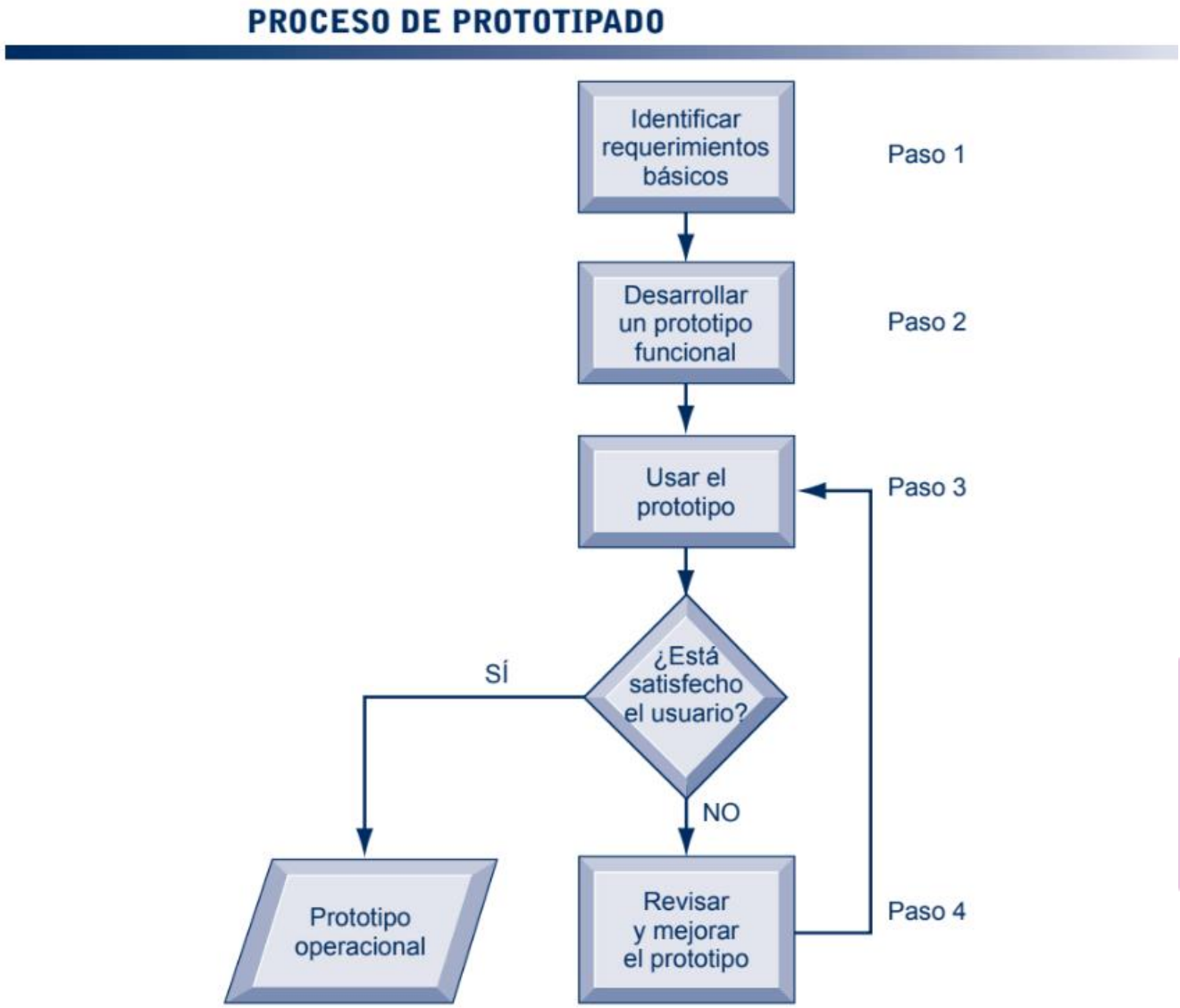
Paso 1: Identificar los requerimientos básicos del usuario. El diseñador del sistema (por lo general, un especialista en sistemas de información) trabaja con el usuario sólo el tiempo suficiente para capturar las necesidades básicas de información del usuario.

Paso 2: Desarrollar un prototipo inicial. El diseñador del sistema crea rápidamente un prototipo funcional mediante el uso de herramientas para generar software con rapidez.

Paso 3: Usar el prototipo. Se anima al usuario a trabajar con el sistema para determinar qué tan bien cumple el prototipo con sus necesidades y para que haga sugerencias sobre cómo mejorar el prototipo.

Paso 4: Revisar y mejorar el prototipo. El creador del sistema anota todos los cambios que el usuario solicita, y refina apropiadamente el prototipo. Después de que se ha revisado el prototipo, el ciclo regresa al paso 3. Los pasos 3 y 4 se repiten hasta
que el usuario queda satisfecho.

Unidad 2: Prototipado



El proceso de desarrollo de un prototipo se puede dividir en cuatro pasos. Puesto que un prototipo se puede desarrollar con rapidez y a un bajo costo, los creadores de sistemas pueden pasar por varias iteraciones en las que se repiten los pasos 3 y 4 para refinar y mejorar el prototipo antes de llegar al prototipo final operacional.

Unidad 2: Prototipado

Ventajas

El prototipado es más útil cuando hay incertidumbre sobre los requerimientos o las soluciones de diseño, y se utiliza con frecuencia para diseñar una interfaz del usuario final del sistema de información (la parte del sistema con la cual interactúan los usuarios finales, como las pantallas de visualización en línea y de captura de datos, los informes o las páginas Web). Ya que los prototipos fomentan la participación intensa del usuario final durante el ciclo de vida de desarrollo de sistemas, es más probable producir sistemas que cumplan con los requerimientos del usuario.

Desventajas

El prototipado rápido puede pasar por alto las etapas esenciales en el desarrollo de sistemas. Si el prototipo terminado funciona de una manera razonable, tal vez la gerencia no vea la necesidad de reprogramar, rediseñar o realizar los procesos completos de documentación y prueba para crear un reluciente sistema de producción. Algunos de estos sistemas que se crean en forma apresurada tal vez no puedan alojar fácilmente grandes cantidades de datos o un gran número de usuarios en un entorno de producción.

Unidad 2: Desarrollo del Usuario Final

El desarrollo del usuario final permite a los ***usuarios finales***, con una mínima cantidad o sin ayuda formal de parte de los especialistas técnicos, reducir el tiempo y los pasos requeridos para ***producir una aplicación*** terminada.

Con el uso de ***lenguajes de consulta*** e informes amigables para los usuarios, desarrollo de sitios Web, gráficos y ***herramientas de software*** de PC, los usuarios finales pueden acceder a los datos, crear informes y desarrollar aplicaciones simples por su cuenta, con poca o ninguna ayuda de los analistas o programadores de sistemas profesionales.

Un lenguaje de consulta es una herramienta de software que da ***respuestas inmediatas***, en línea, a las preguntas que no son predefinidas, como "¿Quiénes son los representantes de ventas con mayor desempeño?". Los lenguajes de consultas están enlazados con frecuencia al software de administración de datos.

Unidad 2: Desarrollo del Usuario Final

Los sistemas desarrollados por el usuario final se pueden completar con más rapidez que los desarrollados a través del ciclo de vida convencional de sistemas.
Permitir que los ***usuarios especifiquen sus propias necesidades*** de negocios mejora la recopilación de los requerimientos, lo cual conduce a un nivel mayor de participación y de satisfacción del usuario con el sistema.

Sin embargo, las herramientas de software del usuario final todavía no pueden reemplazar a las herramientas convencionales para ciertas aplicaciones de negocios, debido a que ***no pueden manejar el procesamiento de grandes cantidades*** de transacciones o de aplicaciones con extensos requerimientos de lógica de procedimientos y de actualizaciones.

La computación del usuario final también impone ***riesgos organizacionales***, puesto que ocurre fuera de los mecanismos tradicionales para la administración y control de los sistemas de información. Cuando los sistemas se crean con rapidez, sin una metodología de desarrollo formal, los procesos de prueba y documentación pueden ser inadecuados. Se puede perder el control sobre los datos en los sistemas que están fuera del departamento

Unidad 2: Paquetes de software de aplicación

Durante las últimas décadas se han creado muchos sistemas basados en un paquete de software de aplicación. Muchas ***aplicaciones son comunes para todas las organizaciones*** de negocios; por ejemplo, nómina, cuentas por cobrar, libro mayor o control de inventario. Para dichas funciones universales con procesos estándar que no cambian mucho en el transcurso del tiempo, un sistema generalizado puede satisfacer los requerimientos de muchas organizaciones.

Si un paquete de software puede satisfacer la mayoría de los requerimientos de una organización, la compañía ***no tiene que escribir su propio software***. La compañía puede ahorrar tiempo y dinero al utilizar los programas de software que el paquete contiene escritos, diseñados y probados con anterioridad. Los distribuidores de los paquetes proveen gran parte del mantenimiento y soporte continuos para el sistema, como las mejoras para mantener el sistema alineado con los continuos desarrollos técnicos y de negocios.

Unidad 2: Paquetes de software de aplicación

Si una organización tiene requerimientos únicos que el paquete no tenga considerados, muchos paquetes cuentan con ***herramientas para adaptación.*** Las características de adaptación permiten modificar un paquete de software para cumplir con los requerimientos únicos de una organización sin destruir la integridad del software empaquetado. Si se requiere un alto grado de adaptación, tal vez los procesos de programación adicional y trabajo de personalización sean tan costosos y requieran tanto tiempo que desaparezcan muchas de las ventajas de los paquetes de software.

Cuando se desarrolla un sistema utilizando un paquete de software de aplicación, el análisis de sistemas integra un ***esfuerzo de evaluación del paquete***, en el cual participan tanto los usuarios finales como los especialistas en sistemas de información.
Los criterios más importantes de evaluación son las funciones que provee el paquete, la flexibilidad, facilidad de uso, recursos de hardware y software, requerimientos de la base de datos, esfuerzos de instalación y mantenimiento, documentación, calidad del distribuidor y costo.

Unidad 2: Outsourcing

Si una empresa no desea usar sus recursos internos para crear y operar sistemas de información, puede ***subcontratar el trabajo a una organización externa*** que se especializa en proporcionar estos servicios.
Los proveedores de cómputo en la nube y de software como un servicio (SaaS) son una forma de outsourcing. Las compañías suscriptoras utilizan el software y el hardware de computadora que proporciona el servicio como la plataforma técnica para sus sistemas.

Cualquier compañía que use outsourcing para sus aplicaciones debe ***entender totalmente el proyecto***; sus requerimientos, el método de implementación, los beneficios anticipados, los componentes del costo y la métrica para medir el desempeño.

Muchas empresas subestiman los ***costos para identificar y evaluar a los proveedores*** de servicios de TI, de cambiar a un nuevo proveedor, de mejorar los métodos de desarrollo de software internos para estar a la par con los métodos de los proveedores de outsourcing, y de monitorearlos para asegurarse de que estén cumpliendo con sus obligaciones contractuales.

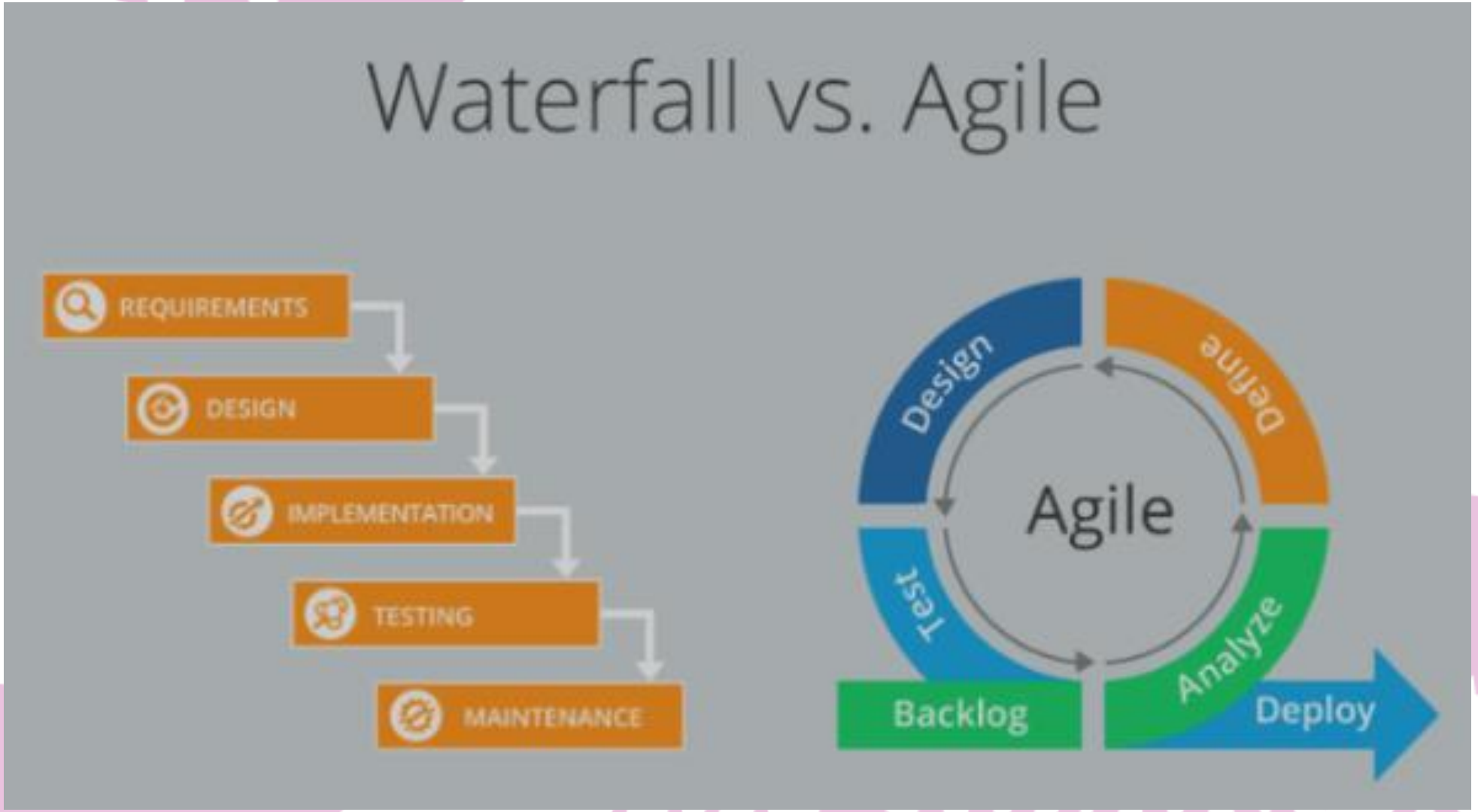
Las empresas deben tener mucho ***cuidado*** al usar un subcontratista para desarrollar u ***operar aplicaciones*** que le den algún tipo de ***ventaja competitiva***.

Unidad 2: Métodos de desarrollos Ágiles

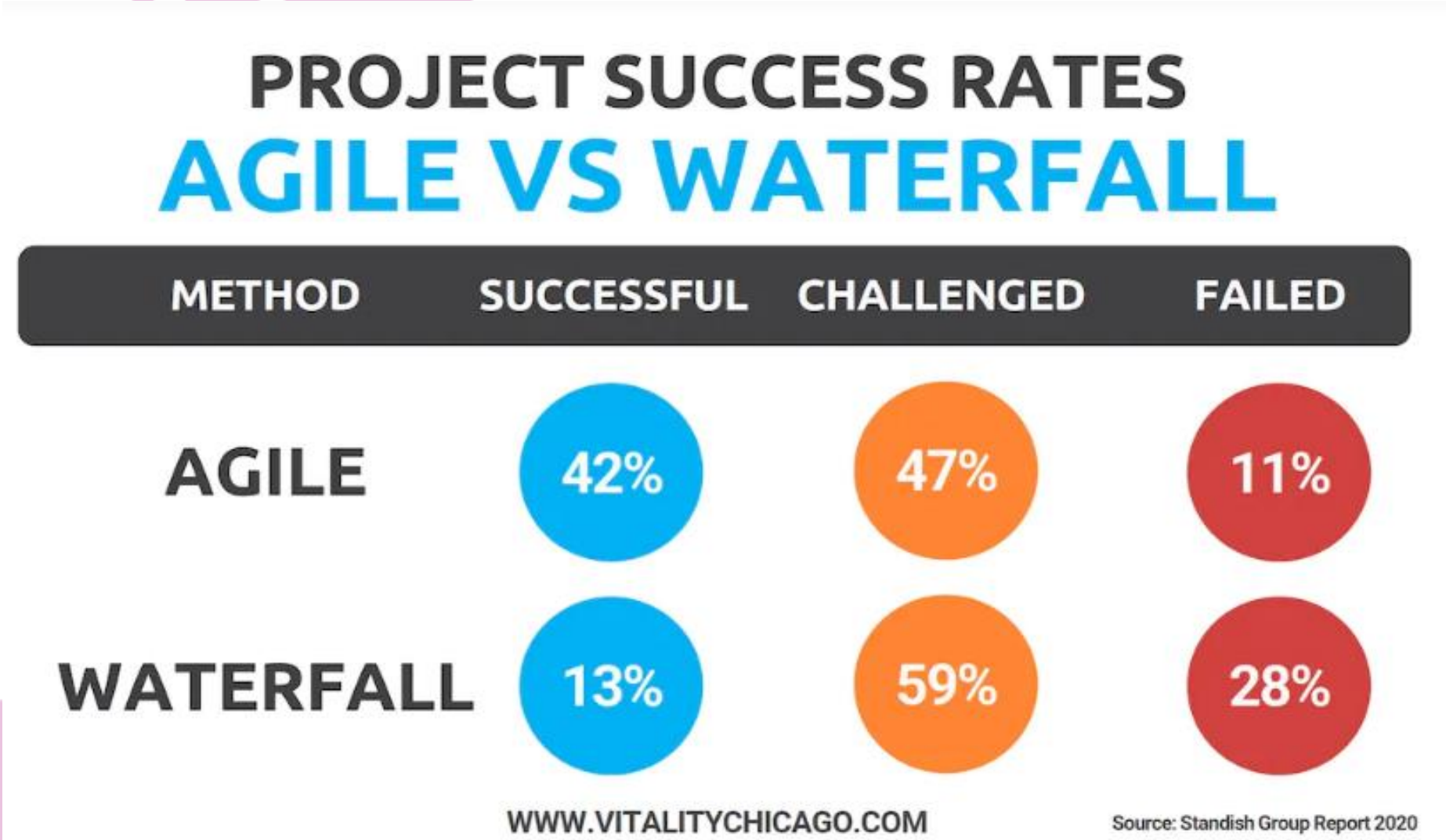
Desarrollo Ágil de Software: tiene como objetivo ofrecer pequeños fragmentos de software en funcionamiento en un corto período de tiempo para mejorar la satisfacción del cliente.

Por definición, 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.

Unidad 2:Métodos de desarrollos Ágiles



Unidad 2:Métodos de desarrollos Ágiles



Unidad 2:Métodos de desarrollos Ágiles

Unidad 2: 12 Principios del manifiestos ágil

Unidad 2: 12 Principios del manifiestos ágil

El software funcionando es la medida principal. **07**

Los procesos ágiles promueven el desarrollo sostenible. **08**

La atención continua a la excelencia técnica. **09**

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial. **10**

Las mejores arquitecturas emergen de equipos autoorganizados. **11**

A intervalos regulares, el equipo reflexiona sobre cómo ser más eficaz. **12**

Unidad 2: Ventajas de la Gestión Ágil de Proyectos

- ***Mejora de la calidad del producto***: Estas metodologías fomentan el enfoque proactivo de los miembros del equipo en la búsqueda de la excelencia del producto.
- ***Mayor satisfacción del cliente***: El cliente está más satisfecho al verse involucrado y comprometido a lo largo de todo el proceso de desarrollo.
- ***Mayor motivación de los trabajadores***: Los equipos de trabajo autogestionados, facilitan el desarrollo de la capacidad creativa y de innovación entre sus miembros.
- ***Trabajo colaborativo***: La división del trabajo por distintos equipos y roles junto al desarrollo de reuniones frecuentes, permite una mejor organización del trabajo.
- ***Uso de métricas más relevantes***: Las métricas utilizadas para estimar parámetros como tiempo, coste, rendimiento, etc. son normalmente más reales en proyectos ágiles que en los tradicionales. Gracias a la división en pequeños equipos y fases podemos ser más conscientes de lo que está sucediendo.
- ***Mayor control y capacidad de predicción***: La oportunidad de revisar y adaptar el producto a lo largo del proceso ágil, permite a todos los miembros del proyecto ejercer un mayor control sobre su trabajo, cosa que permite mejorar la capacidad de predicción en tiempo y costes.
- ***Reducción de costes***: La gestión ágil del proyecto elimina prácticamente la posibilidad de fracaso absoluto en el proyecto, porque los errores se van identificando a lo largo del desarrollo en lugar de esperar a que el producto esté acabado y toda la inversión realizada.

Unidad 2:Métodos de desarrollos Ágiles

Algunos de los métodos más comunes son:

- ***Scrum***: Es un marco que se enfoca en la entrega iterativa e incremental de productos y soluciones.
- ***Kanban***: Es un marco visual de gestión de proyectos que se enfoca en la mejora continua y la entrega de valor.
- ***Programación extrema (XP):*** Es un marco de desarrollo ágil con el objetivo de desarrollar y gestionar proyectos con eficiencia, flexibilidad y control

El método de desarrollo de software elegido depende de factores como la naturaleza del proyecto, los recursos disponibles y las preferencias de la empresa y el equipo de desarrollo.

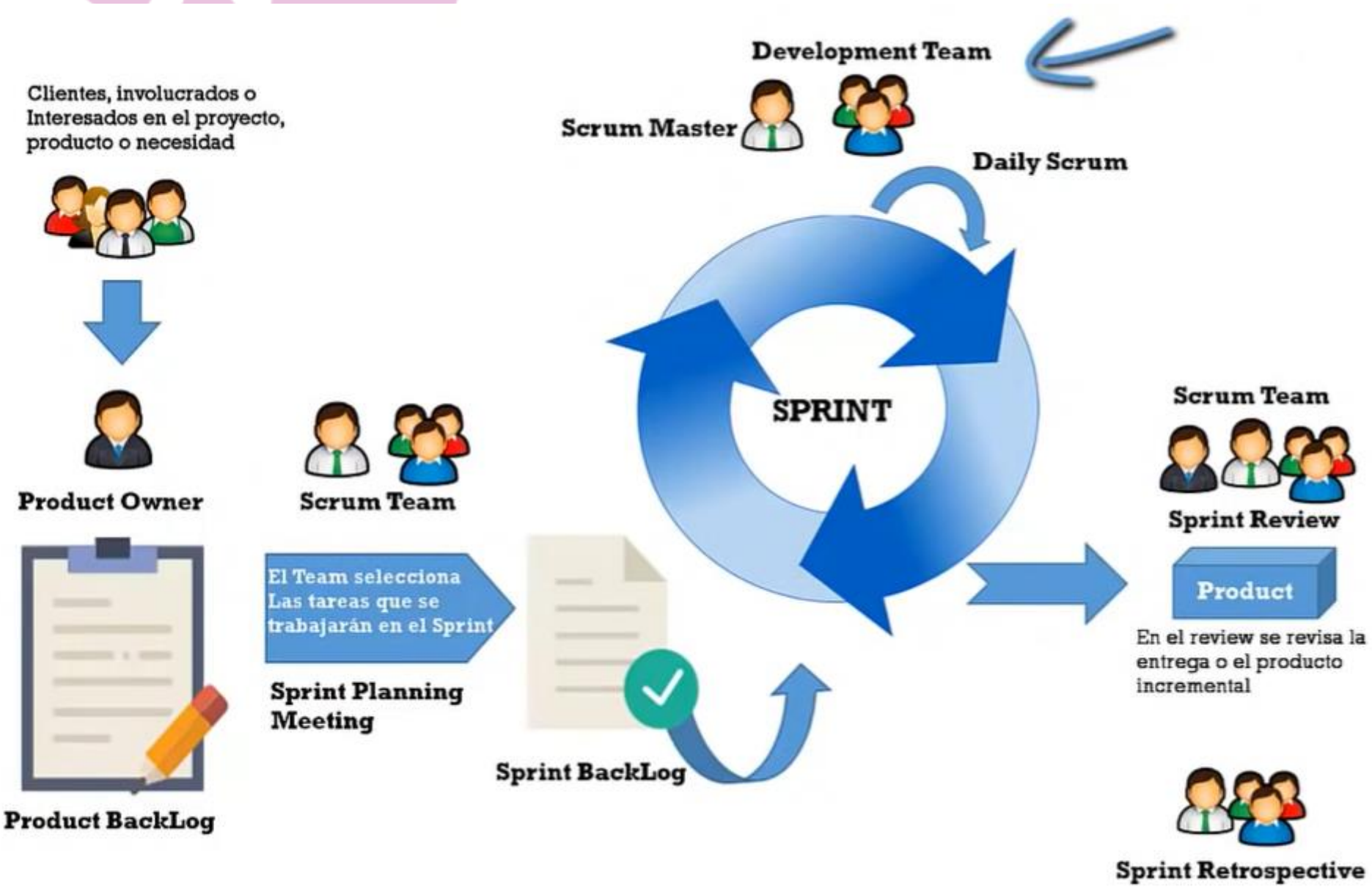
Unidad 2: Scrum

Se caracteriza por ser la ***metodología del caos*** que se basa en una estructura de ***desarrollo incremental***, esto es, cualquier ciclo de desarrollo del producto y/o servicio se desgrana en ***pequeños proyectos*** divididos en distintas etapas: ***análisis, desarrollo y testing***. En la etapa de desarrollo encontramos lo que se conoce como interacciones del proceso o ***Sprint***, es decir, entregas regulares y parciales del producto final.

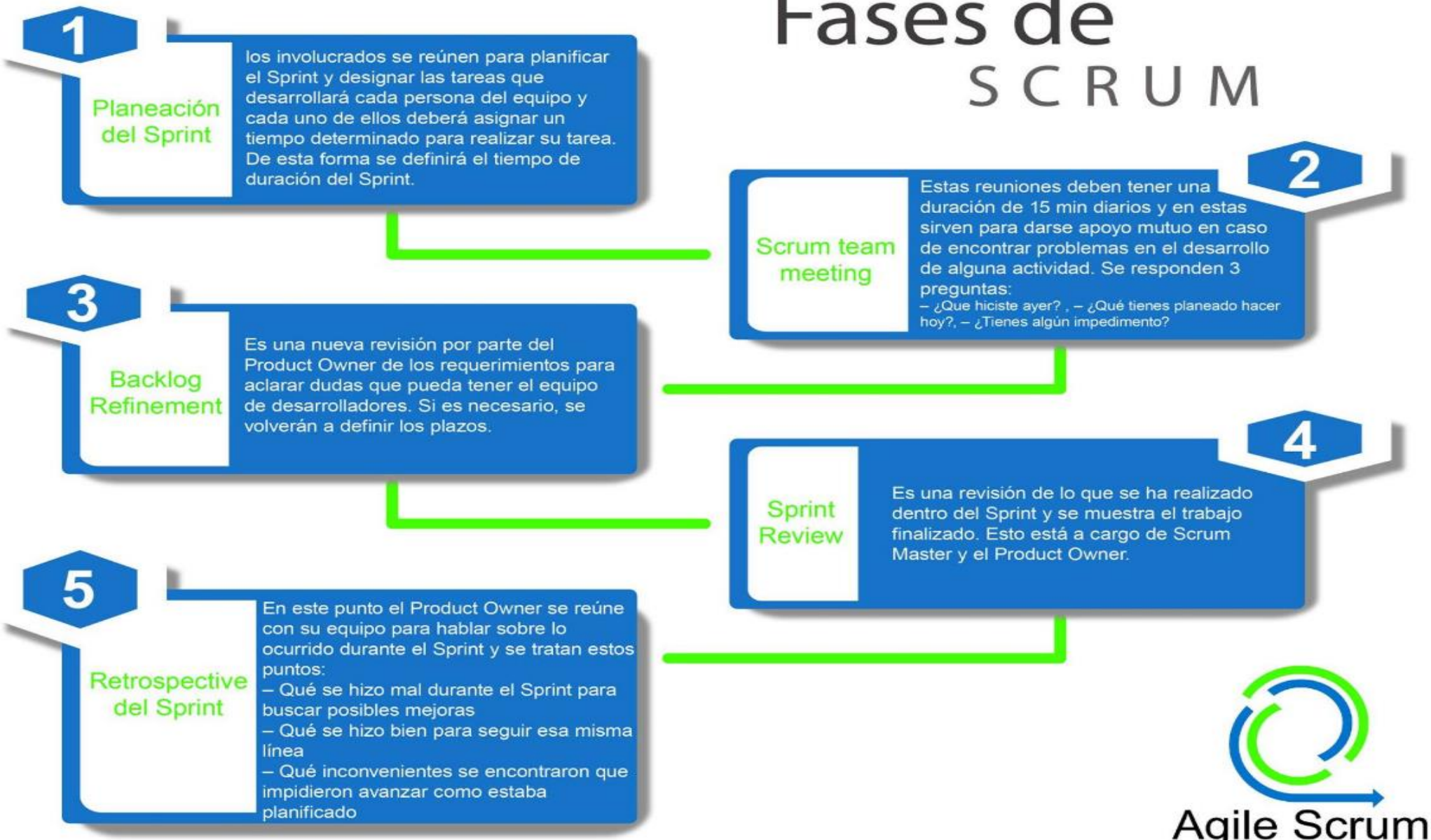
Las ***reuniones*** son el pilar ***fundamental*** de la metodología, donde diferenciamos entre: reuniones de ***planificación***, ***diaria***, de ***revisión*** y de ***retrospectiva***, la más importante de todas ellas, ya que, se realiza después de terminar un sprint para ***reflexionar*** y ***proponer mejoras*** en los avances del proyecto.

Los aspectos clave por los que se mueve el Scrum son: innovación, flexibilidad, competitividad y productividad.

Unidad 2: Scrum



Fases de SCRUM



Agile Scrum

Unidad 2: Habilidades de un equipo Scrum

Product Owner

- Saber traducir los objetivos de la empresa en user stories.
- Gestión de los grupos de interés.
- Solución creativa de problemas.
- Capacidad para comprensión técnica del producto.
- Grandes habilidades de liderazgo, comunicativas y destrezas de presentación.
- Atención al detalle.

Scrum Master

- Coaching.
- Conocimiento básico acerca del desarrollo de software o del proyecto en el que se trabajará.
- Técnicas como: user stories, continuous testing, agile games, continuous integration, etc.
- Gestión del tiempo, asegurar que el producto se entregue a tiempo.

Development Team

- Auto-organizados.
- Cross-funcionales.
- Dentro del equipo de desarrollo no existen los títulos.
- No existen sub-equipos dentro del equipo de desarrollo.
- Los distintos miembros del equipo de desarrollo tendrán distintas especialidades, pero responden como un equipo.

Unidad 2: Kanban

La estrategia ***Kanban*** conocida como ***"Tarjeta Visual"*** es muy útil para los responsables de proyectos. Esta consiste en la elaboración de un ***cuadro*** o diagrama en el que se reflejan ***tres columnas*** de tareas; ***pendientes***, ***en proceso*** o ***terminadas***. Este cuadro debe estar al alcance de todos los miembros del equipo, evitando así la repetición de tareas o la posibilidad de que se olvide alguna de ellas. Por tanto, ayuda a mejorar la productividad y eficiencia del equipo de trabajo.

"Kanban" es una ***palabra japonesa***, conformada por "***Kan***", que significa '***visual***', y "***Ban***", '***tarjeta***'. Es decir que utilizarás 'tarjetas visuales' para organizarte y producir mejor.

La sencillez radica en que, de ***un solo vistazo***, se puede apreciar el ***estado de cada proyecto***, y de esta forma, según el grado de avance, saber cuándo poder sumar nuevas responsabilidades.

Unidad 2: Kanban

Unidad 2: Pasos para implementar Kanban

PASO 1: divide tu proyecto en sus fases de ejecución

Observar de forma general todo el proyecto. Crear las diferentes fases que lo componen. Incluir todo lo que necesite para que el proyecto llegue a su fin, desde una lluvia de ideas hasta la intervención de profesionales externos que ayudarán en su avance.

PASO 2: clasifica y ordena las fases del proceso

Ordenar las diferentes fases en orden de ejecución. Se recomienda que utilice diferentes colores para cada fase.

PASO 3: escribe las diferentes tareas de cada fase

Crear tarjetas que corresponden a las diferentes tareas que forman cada fase. Lo bueno de estas tarjetas es que son movibles, algo facilita mucho el trabajo en caso de tener que incorporar una nueva tarea o resolver cualquier imprevisto, cambiando el orden de ejecución de las mismas. Coloca las tareas en orden descendente de prioridad debajo de la fase correspondiente.

PASO 4: lleva el control del proyecto

Una vez que empieza a ponerse en marcha el proyecto, y gracias a este sistema, puedes llevar un control muy visual de cómo va evolucionando. Tener en cuenta que, en muchas ocasiones, puede que haya más de una fase que puede llevarse a cabo de forma simultánea.

Las herramientas como **Trello, Asana o Wrike están basadas en Kanban para la ejecución de los proyectos y son perfectas para aplicar esta metodología.*

Unidad 2: Kanban

Unidad 2: Programación Extrema

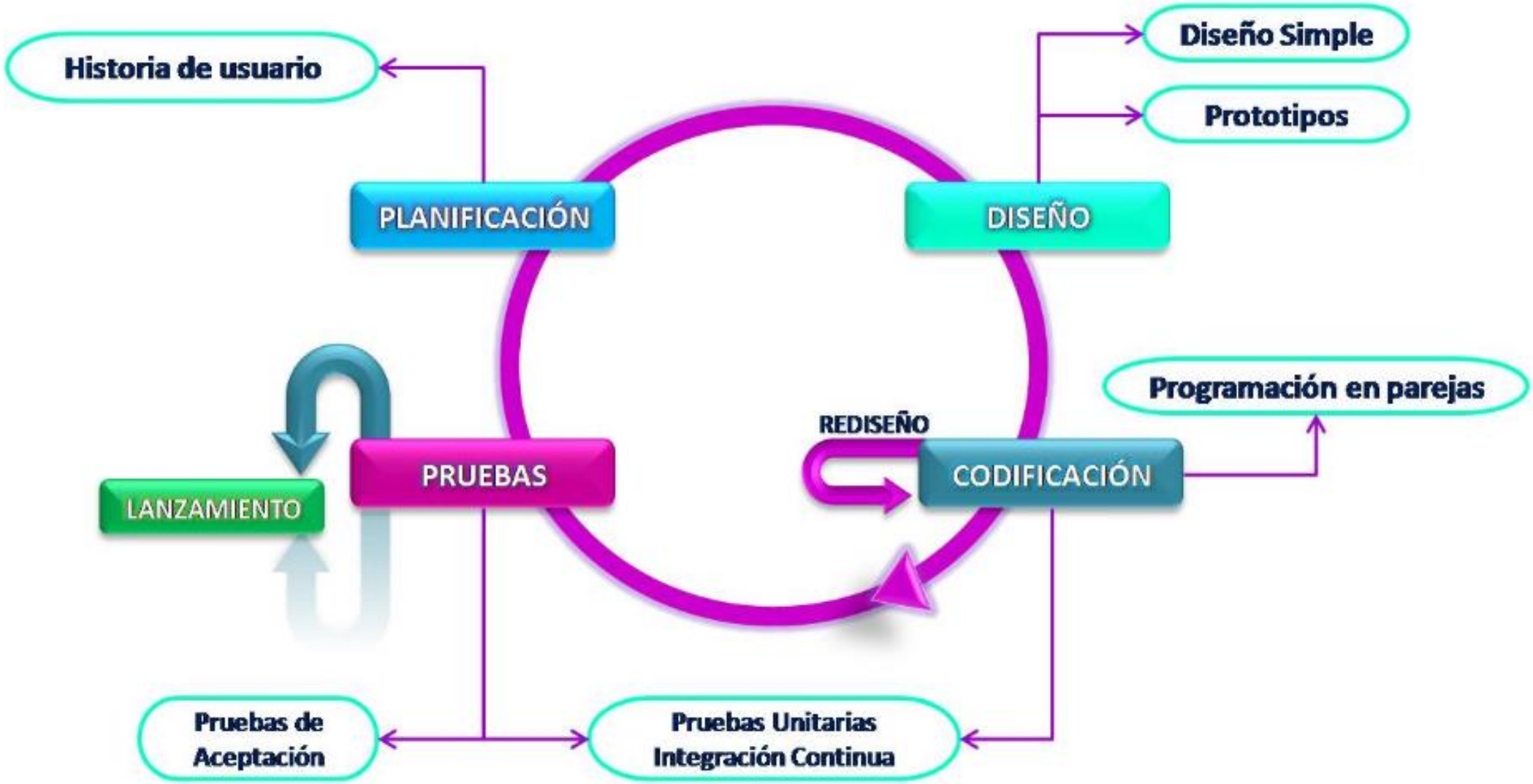
Conocida por sus siglas XP (***eXtreme Programming***), es una metodología basada en un conjunto de ***reglas*** y ***buenas prácticas*** para el desarrollo de software en ***ambientes*** muy ***cambiantes*** con ***requisitos imprecisos***, por ende está enfocada en la ***retroalimentación*** continua entre el equipo de desarrollo y el cliente.

Es por ello que iniciando el proyecto se deben definir todos los requisitos, para luego invertir el ***esfuerzo*** en ***manejar*** los ***cambios*** que se presenten y así minimizar las posibilidades de error. XP tiene como base la ***simplicidad*** y como objetivo la ***satisfacción*** del ***cliente***.

Este enfoque se puede utilizar cuando se requieren los siguientes factores:

- La ***comunicación*** entre el cliente y el equipo de desarrollo es siempre abierta.
- El ***cambio constante*** requiere una reacción rápida.
- Con un ***calendario flexible*** de actividades, la planificación está abierta.
- El ***software funcional*** tiene prioridad sobre todas las demás formas de documentación.
- Los principales criterios de ***éxito*** del proyecto son las ***necesidades*** del ***cliente*** y los esfuerzos del equipo del proyecto.

Unidad 2: Programación Extrema



Unidad 2: Programación Extrema

FASE 1: PLANIFICACIÓN

Según la identificación de las historias de usuario,se priorizan y se descomponen en mini-versiones. La planificación se va a ir revisando. Cada dos semanas aproximadamente de iteración, se debe obtener un software útil, funcional, listo para probar y lanzar.

FASE 2: DISEÑO

En este paso se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Se obtendrá el prototipo. .

FASE 3: CODIFICACIÓN «DE TODOS»

La programación aquí se hace "a dos manos", en parejas en frente del mismo ordenador. Incluso, a veces se intercambian las parejas. De esta forma, se asegura que se realice un código más universal, con el que cualquier otro programador podría trabajar y entender. Debe parecer que ha sido realizado por una única persona. Así se conseguirá una programación organizada y planificada.

FASE 4: PRUEBAS

Se deben realizar pruebas automáticas continuamente. Al tratarse normalmente de proyectos a corto plazo, este testeo automatizado y constante es clave. Además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones.

FASE 5: LANZAMIENTO

Si se llego a este punto, significa que se ha probado todas las historias de usuario o mini-versiones con éxito, cumpliendo los requerimientos del clientes. Se tiene un software útil y se puede incorporar al producto.

Unidad 2: RFI, RFP y RFQ

La RFI, la RFP y la RFQ tienen funciones diferentes, pero todas tienen el objetivo principal de ayudar a los clientes a conseguir los mejores proveedores para sus proyectos.

- ***RFI*** educa al cliente, ayudándole a entender las posibles soluciones que ofrecen los proveedores.
- ***RFP*** compara los diferentes valores que ofrecen los proveedores.
- ***RFQ*** detallan los costes para cumplir los requisitos del proyecto.

Combinar la solicitud de información, la solicitud de propuestas y la petición de oferta simplifica el proceso de selección de proveedores y ahorra a los clientes mucho tiempo y dinero. También ayuda a los clientes a establecer relaciones a largo plazo con los proveedores, ya que encuentran rápidamente proveedores que satisfacen sus necesidades individuales. Un cliente debe utilizar todas estas solicitudes de compra para encontrar proveedores que satisfagan todas sus necesidades.

Unidad 2: RFI

SOLICITUD DE INFORMACIÓN

La solicitud de información (RFI) es anterior a otras solicitudes de contratación. Algunas organizaciones necesitan una RFI ***antes de iniciar el proceso de adquisición***. Proporciona información general sobre los artículos y servicios que puede necesitar comprar y sobre los proveedores.

La estructura de la solicitud de información contiene una ***introducción en la que se expone el problema*** que tiene la organización contratante. Suelen abarcar la ubicación, la capacidad, el tamaño, la capacidad de producción, las tendencias del mercado y otra información de alto nivel relacionada con el proyecto.

La RFItambién explica el alcance del problema y las ***expectativas de las posibles soluciones***. Hace referencia a todas las abreviaturas y la terminología y ofrece una plantilla clara para ayudar a los proveedores a responder con una RFI satisfactoria.

Dado que la solicitud de información es un documento de investigación, las preguntas deben ser abiertas para permitir que el proveedor ***describa por completo lo que ofrece*** y sus soluciones empresariales para responder a sus retos.

La RFI ***no es vinculante*** para el proveedor o la empresa solicitante. Ayuda al cliente a ***entender el mercado*** y permite a los proveedores identificar nuevos clientes potenciales y mostrar sus habilidades.

Unidad 2: RFP

SOLICITUD DE PROPUESTA

La solicitud de propuesta (RFP) es un ***documento formal*** para buscar propuestas de proveedores de servicios sobre bienes, servicios o productos valiosos. En la RFP ***se evalúa y compara*** lo que los proveedores de servicios pueden ofrecer y cuál es la mejor propuesta en términos de ***presupuesto*** y entrega del proyecto.

La RFP sigue un ***calendario estricto***, la respuesta del proveedor y las normas de contenido. Las personas y organizaciones suelen utilizar las RFP cuando buscan ***conocimientos técnicos*** o proyectos de construcción complejos y sofisticados. La RFP solicita información a los proveedores sobre cómo van a llevar a cabo el proyecto. También muestra al proveedor cualquier ***solución*** interna que tenga el cliente, y el proveedor puede ofrecer una ***alternativa***. La RFP también es específica y concisa y se utiliza para elegir un proveedor en la fase de decisión.

La solicitud de propuestas suele convertirse en un ***proceso de licitación***, ya que el objetivo principal es dar a los proveedores la información suficiente para que propongan una solución razonable y, al mismo tiempo, dejarles suficiente margen para la creatividad y la aplicación de las mejores prácticas empresariales.

En algunos proyectos, el proceso de aprovisionamiento termina en la RFP, pero en otros, se procede a la RFQ (solicitud de oferta) para garantizar que haya una evaluación de precios justa entre los proveedores.

Unidad 2: RFQ

SOLICITUD DE PRESUPUESTO

La petición de oferta (RFQ) también se conoce como invitación a licitar. En el sector gubernamental y público, se denomina ***invitación a licitar***.

La petición de oferta suele ser la ***última fase del proceso de contratación***, pero a veces puede preceder a la solicitud de propuestas si el precio es un factor de diferenciación fundamental para el proyecto.

Cuando un cliente envía una petición de oferta, ***conoce los problemas que debe resolver*** y las soluciones necesarias.

La RFQ ***no permite*** a los proveedores ofrecer ***soluciones creativas***, sino que busca que los proveedores completen el proyecto utilizando el método de adquisición que el cliente ha elegido.

La petición de oferta se ***refiere principalmente a los costes***, y una petición de oferta detallada tiene todos los componentes que influyen en los precios. La petición de oferta también indica la cantidad de productos necesarios, las fechas de entrega previstas y las condiciones de pago estándar.

Sólo emita una petición de oferta cuando ***tenga los detalles de lo que necesita adquirir.***

Unidad 2: KPI

key performance indicator - indicador clave de rendimiento

Son métricas cuantificables que se utilizan para evaluar el rendimiento de una organización, un proceso o un proyecto en función de sus objetivos y metas. Estos indicadores proporcionan información valiosa para medir el progreso, identificar áreas de mejora y tomar decisiones informadas

Aspectos clave

- Relevancia para Objetivos Estratégicos: Deben estar alineados con los objetivos estratégicos de la organización. Cada indicador debe tener un propósito claro y contribuir al logro de metas específicas.
- Medición Cuantitativa: Son medidas cuantificables y específicas. Deben expresarse en términos numéricos para proporcionar una evaluación clara del rendimiento.
- Periodo de Medición: Se miden en intervalos de tiempo definidos. Pueden ser evaluados diaria, semanal, mensual, trimestral o anualmente, dependiendo de la naturaleza de la métrica y los objetivos.
- Comparación y Análisis: Son efectivos cuando se pueden comparar con objetivos anteriores, benchmarks de la industria o metas establecidas.
- Variedad de Áreas de Negocio: Pueden aplicarse a diversas áreas de negocio, como ventas, marketing, finanzas, recursos humanos, operaciones, etc.

Unidad 2: SLA

El significado del acrónimo ***SLA*** es "Service Level Agreement", en español (***Acuerdo de nivel de servicio*** o Garantía de nivel de servicio). Se trata de un contrato firmado entre las partes involucradas en una negociación que determina cuáles son las ***responsabilidades de cada uno*** en relación a los servicios contratados.

El SLA sirve para ***definir*** el ***alcance*** del trabajo y ***establece*** las ***reglas***, ***acuerdos***, ***metas*** y demás cuestiones relacionadas con el servicio que se prestará. Con el paso del tiempo, las demandas pueden cambiar, por lo que es importante que el documento sea revisado periódicamente para que las pautas se redefinan según las necesidades del momento.

Durante la elaboración del SLA, también es importante ***analizar*** los ***posibles escenarios*** de ***riesgo*** y anticiparlos. Para esto, vale la pena insertar cláusulas en el contrato que protejan a la empresa de la malversación en las negociaciones y aseguren que todos estén de acuerdo con ellas.

El SLA es ***esencial*** para ***proteger*** a todas las partes durante ***la negociación***, ya que estipula los deberes y responsabilidades del contratista y el contratado. Así, el documento ***garantiza*** la ***transparencia*** del proceso y facilita la resolución de posibles conflictos que puedan surgir con el tiempo.

Además, es posible ***insertar cláusulas*** sobre ***sanciones*** y ***multas*** en el SLA si alguno de los involucrados en la negociación no cumple con sus obligaciones establecidas en el contrato. Esto también da más seguridad a todos.

Unidad 2: SLA

Los contratos de SLA tienen varias aplicaciones, por lo que se dividió en dos categorías:

SLA con ***enfoque*** en el ***cliente***: el documento se refiere a las ***demandas del cliente*** en cuanto a la contratación del servicio. Es más complejo porque cada uno tendrá demandas específicas

SLA con ***enfoque*** en ***servicio***: se dirige específicamente al ***propio servicio***, siendo un modelo más sencillo. Se puede aplicar a diferentes negociaciones sin realizar cambios en el contrato.

Para calcular el SLA del servicio, debes seguir algunos pasos:

1. ***Define*** cuáles son las ***métricas*** que son realmente significativas para tu negocio.
2. ***Establezca*** cuáles serán las ***metas*** y los objetivos.
3. ***Supervise*** constantemente los ***resultados*** recopilados.

Entre los KPI de servicio al cliente que se pueden utilizar se encuentran el ***tiempo medio*** de ***espera***, el ***tiempo total*** de servicio, el ***tiempo*** de ***respuesta*** y el ***nivel*** de ***satisfacción***, por nombrar solo algunos.

Unidad 2:Tendencias

Algunas tendencias futuras en metodologías de desarrollo de sistemas de información son:

- ***Modelado de microservicios***: El uso de microservicios seguirá aumentando como una forma de desarrollar aplicaciones escalables y flexibles.

- MicroFronting:
- ***Automatización de pruebas***: La automatización de pruebas será cada vez más importante para aumentar la eficiencia y la calidad del desarrollo.
- ***Aprendizaje automático***: La incorporación del aprendizaje automático en el desarrollo de sistemas de información seguirá creciendo para mejorar la eficiencia y la personalización de la experiencia del usuario.
- ***Desarrollo basado en componentes***: El desarrollo basado en componentes permitirá a los desarrolladores crear aplicaciones más rápido y fácilmente utilizando componentes previamente desarrollados.