

RESUMEN TEMAS INCLUIDOS EN EL EXAMEN N° 1**¿Qué se entiende por Proyecto Informático?**

Un Proyecto Informático es, básicamente, un producto software que se desarrolla con el objeto de satisfacer una necesidad o brindar un servicio, incluso un servicio recreativo.

Todo Proyecto Informático consta de tres integrantes:

- a) El equipo de desarrollo.
- b) El cliente que solicita o el grupo de potenciales usuarios a los que está dirigido el proyecto.
- c) El producto software en sí mismo, incluyendo la plataforma sobre la que se ejecutará y el hardware necesario para su ejecución.

Siempre que se mencione “Proyecto Informático” hay que tener en cuenta la existencia de esta “trinidad” y al hacer algún análisis sobre el mismo podemos referirnos a todos los integrantes o hacer análisis sobre alguno de ellos en particular. Sea cual fuere el caso, el resultado del análisis tendrá efectos sobre el Proyecto Informático en su totalidad.

¿Qué entiendo por Relevamiento?

El Relevamiento es el conjunto de tareas que se hace imprescindible realizar cuando se toma la decisión de desarrollar un producto software, sea por iniciativa propia del equipo de desarrollo como cuando se recibe el requerimiento de un potencial cliente para satisfacer alguna necesidad.

La palabra clave en todo el proceso de desarrollo de un proyecto informático, desde el relevamiento hasta el mantenimiento pasando por la etapa de desarrollo es INFORMACIÓN (esta será la primera de las palabras mágicas que deberán considerarse en el proceso de desarrollo, implementación y mantenimiento de un proyecto informático) y el proceso de

relevamiento es el primer paso para la adquisición de la misma, aunque no el único medio de adquirirla.

Si bien existen matices en el proceso de Relevamiento dependiendo del origen del proyecto (iniciativa propia o requerimiento de terceros), básicamente la primera etapa consiste en determinar que se quiere hacer, es decir, establecer lo más claramente posible a que apunta el proyecto informático o cual es la necesidad que se pretende satisfacer con el producto software y cual será la plataforma sobre la que correrá el mismo.

La segunda etapa consiste en definir los procesos que deberán o podrán ejecutarse desde el producto software. En esta etapa es donde nace el diseño funcional y, en el caso de ser necesario, el del sistema de archivos tanto en lo que se refiere a la lista de archivos necesarios, la estructura de aquellos que deberán almacenar la información base necesaria para la ejecución de los procesos más complejos (llamados archivos padre o base) así como todos aquellos que servirán de repositorio de la información generada en los procesos, la relación entre archivos, etc.

Para el caso en que el proyecto nazca de un requerimiento específico esta etapa debe incluir la revisión de los procesos que realiza el solicitante ya sea de manera manual o mediante el uso de alguna herramienta informática así como la circulación de la información o de documentación dentro de la empresa o entre los integrantes de la misma.

La tercera etapa consiste en tratar de determinar elementos no requeridos por el solicitante o que no se encuentren en productos softwares similares y que puedan aportarle a nuestro desarrollo lo que se denomina valor agregado, es decir, elementos que le aporten mayor calidad y seguridad a nuestro producto en comparación con productos similares en el mercado o que consideremos serán valorados por los potenciales usuarios a la hora de elegir

el producto que desarrollamos. Esto va desde hacer que la interfaz del producto software sea sencilla, ofrecer cosas no solicitadas pero que consideremos interesantes, facilitar el acceso a la información, flexibilizar los procesos para hacerlos más versátiles, mecanismos de autocontrol o de seguridad.

Por último resta la cuarta etapa que consiste en determinar como se va a implementar el producto software en función de la plataforma en la que va a funcionar el mismo y este punto abarca tanto el lenguaje de programación que se va a emplear en el desarrollo así como herramientas adicionales que puedan ser necesarias y la forma en que se controlará el flujo de datos, etc.

Cabe señalar que todas las etapas se retroalimentan entre sí y las conclusiones a las que se arriban o la información que se obtiene en cada una de ellas puede, y de hecho en la mayoría de los casos lo hace, forzar a una revisión de las conclusiones de las demás.

Otra palabra fundamental durante el relevamiento (y como veremos en todo lo que se refiere al desarrollo de un Proyecto Informático es DOCUMENTACIÓN (segunda palabra mágica).

Esto consiste en llevar un registro escrito o digital de toda la información recabada durante el proceso de relevamiento, las conclusiones a las que se arriban, los procesos, la estructura de archivos, etc.

Y en lo que se refiere a los procesos, un elemento interesante a implementar a la hora de documentarlos es el uso de algún método de graficación o diagramas que expliquen visualmente el funcionamiento lógico de los mismos, el flujo y la transformación de los datos dentro de los procesos y entre distintos procesos.

Los beneficios de la documentación son, entre otros:

- La posibilidad de continuar un trabajo comenzado por otros o poder delegar las tareas en personas que no estuvieron en el inicio del proyecto.
- Poner a prueba la lógica y la integridad de los procesos antes de escribir una sola línea de código.
- Realizar lo que se denomina prueba de escritorio para el testeo de los procesos o aplicaciones. Esta prueba de escritorio consiste en seleccionar una colección de datos que nos permita conocer de antemano el resultado que debería arrojar el proceso y cotejarlo con la forma en que los datos se van transformando a lo largo de la ejecución del proceso y/o con el resultado del testeo de la aplicación ya programada.

Siempre hay que considerar a un producto software como si se tratara de un ser vivo ya que tal como ocurre con estos, el producto software nace, se desarrolla a lo largo del tiempo ya sea por la aparición de nuevas necesidades o nuevas ideas y, eventualmente puede llegar a morir ya sea porque cae en desuso o es reemplazado por otro producto.

Podemos entonces establecer similitudes entre los seres vivos y los proyectos informáticos de la siguiente manera.

Seres Vivos

Proyecto Informático

Matrimonio que decide tener un hijo

Etapas de Planificación

Embarazo o gestación

Etapas de Desarrollo

Nacimiento del hijo

Implementación o lanzamiento del software

Crecimiento del hijo

Etapas de mantenimiento

Muerte

Cuando el software deja de ser usado.

El proceso de mantenimiento (actualización o mejoras) del producto software requiere

realizar nuevos relevamientos (con todo lo que ello significa) y continuando todas las prácticas implementadas en el relevamiento inicial, en particular a lo que se refiere a la documentación de todo lo relacionado a las actualizaciones y/o mejoras. Estas actualizaciones o mejoras pueden estar dadas por pedido del cliente, sugerencia de los usuarios, ideas nacidas en el equipo de desarrollo o porque las condiciones del mercado así lo imponen (Por ejemplo un cambio implementado por la A.F.I.P. en cuanto a los requisitos en la impresión de facturas).

La tercera y última palabra mágica en todo proyecto informático es PROACTIVIDAD, y esto significa no limitarnos sólo a lo que recibimos como información al momento del relevamiento sino también tratar de anticiparnos a cosas que puedan llegar a suceder en el futuro (sean problemas o posibles requerimientos) de manera de tratar de anticiparnos y considerar una respuesta en el momento de planificar el desarrollo del proyecto y así tenerla implementada antes que eso suceda. Esto se verá con más detalle cuando se desarrolle el tema Riesgos y Seguridad Informática.

Una vez que tenemos identificados los procesos necesarios para completar nuestro proyecto informático podemos clasificarlos en 4 (cuatro) tipos:

Elementos imprescindibles (*Debe tener*): En esta categoría se apuntan las tareas que no pueden obviarse porque pondrían en riesgo la ejecución del proyecto o impedirían que el proyecto cumpla las funciones para las que fue desarrollado. Por lo general, se expresan como objetivos a gran escala y reflejan directamente las necesidades que nuestro cliente necesita satisfacer con el software.

Elementos importantes (*Debería tener*): Son tareas o aspectos de mediana envergadura, pero que no alcanzan a determinar el curso de la ejecución de un proyecto pero se derivan de

los objetivos incluidos en la primera categoría ya que sin estos elementos no podríamos completar o desarrollar los elementos imprescindibles.

Elementos interesantes (*Podría tener*): Más que tareas o elementos específicos, son oportunidades que se podrían darle mayor valor al proyecto. Muchas de ellas se expresan en condicional y siempre como alternativas a algo. Representan fundamentalmente cosas que no fueron solicitadas por el cliente o no son necesarias para alcanzar el objetivo para el que fue diseñado el software pero aportan mejoras en seguridad, calidad, confiabilidad o brindan herramientas adicionales que el equipo de desarrollo considera que pueden serle útiles al cliente o los usuario.

Elementos opcionales (*No tendrá ahora pero podría en el futuro*): Finalmente, lo opcional es aquello que se podría hacer, pero que por alguna razón el equipo de desarrollo decide dejarlas para su análisis o implementación en un momento futuro. Esta decisión puede deberse a diversos factores tales como la necesidad de investigar sobre como implementar estas ideas y/o el tiempo que se requiere para su desarrollo y testeo y/o la disponibilidad de las herramientas necesarias para su desarrollo. Al ser una especie de descarte de los elementos interesantes, como no son determinantes del éxito del proyecto y dependiendo de las razones por las cuales el equipo de desarrollo decidió postergar su desarrollo e implementación, podemos dejarlas de lado para un futuro que signifique:

- Hacerlo como tareas finales antes de la implementación del proyecto informático si es que quedara algún tiempo entre la finalización de la etapa de desarrollo y la fecha de entrega convenida o establecida en el momento de la planificación.
- Hacerlo después de la implementación del proyecto informático a manera de nueva versión del software o una actualización del mismo.

- Dejarlo como una buena idea que en algún momento se investigará para pensar en su implementación

Llegado a este punto podemos incorporar los conceptos de **Valor de Negocio** y **Valor Agregado** en lo que se refiere a los elementos (procesos) que van a constituir nuestro proyecto informático (producto software)

Valor de Negocio: A este grupo pertenecen los elementos Imprescindibles y los Importantes ya que estos son los que tienen importancia en el éxito del producto software para satisfacer las necesidades del cliente o brindar efectivamente el servicio que se desea brindar con él. Podríamos decir entonces que el Valor de Negocio está dado por la importancia del proceso dentro del proyecto en su totalidad.

Valor Agregado: A este grupo pertenecen los elementos Interesantes y los Opcionales ya que los primeros en su totalidad y los segundos en caso de incorporarse al proyecto implican mejoras al proyecto en diversos aspectos respecto del proyecto original y constituyen elementos que pueden llegar a diferenciarnos de proyectos similares desarrollados por empresas competidoras. Podríamos decir entonces que el Valor Agregado está dado por el grado de mejora que el proceso aporta al proyecto original.

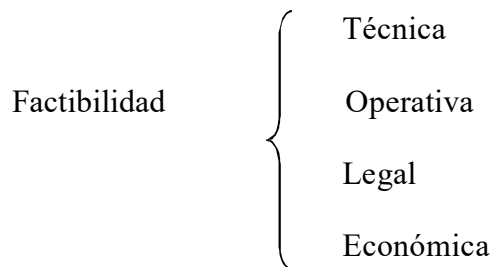
Estudio o Análisis de Factibilidad

Análisis amplio que se realiza a nivel económico, financiero, técnico y social dentro de una empresa de desarrollos informáticos para alcanzar los mejores resultados con el software que se va a desarrollar. El Estudio de Factibilidad es una más de las tantas tareas que se realizan durante el proceso de Relevamiento ya que mucha (o casi toda) la información necesaria para realizar este análisis se obtiene directa o indirectamente junto con los temas que hacen

al propio desarrollo técnico del producto software (procesos necesarios).

Un punto importante al hablar de Factibilidad es que, pese a lo que la palabra puede hacernos suponer, no se trata de ver si el proyecto puede implementarse o no, sino que significa analizar la situación al momento de encarar el desarrollo del proyecto y establecer las cosas que se deberán tener en cuenta para que el proyecto a desarrollar sea exitoso.

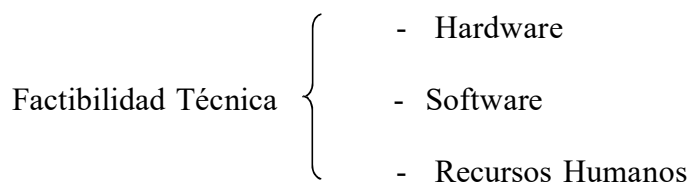
El Análisis o Estudio de factibilidad se divide en 4 (cuatro) categorías:



cada una de las cuales puede tener a su vez subdivisiones.

Factibilidad Técnica: La Factibilidad Técnica tiene que ver fundamentalmente a cuestiones que tienen que ver con el desarrollo y la implementación del producto software y se divide en

3 (tres) elementos:



Factibilidad Técnica – Hardware

Consiste fundamentalmente en establecer, por el lado del cliente/usuario el hardware del que dispone al momento de encararse el desarrollo del proyecto informático y cual sería el hardware con el que debería contar para que el proyecto funcione en condiciones óptimas.

Esto significa analizar si el cliente debería o no incorporar hardware adicional al existente o por ejemplo para una aplicación para celulares si existen equipos en los cuales no se podría instalar la app.

Por el lado del equipo de desarrollo el análisis es similar, es decir, analizar si el hardware del que se dispone es suficiente para encarar el desarrollo del proyecto o si sería necesario incorporar nuevo equipamiento.

Cuando se trate de un cliente o del equipo de desarrollo siempre es conveniente hacer una propuesta de equipamiento óptimo a disponer y en función de eso ver si existe una alternativa opcional (menos hardware del necesario para la condición óptima pero que permita una implementación satisfactoria) para presentarle ambas propuestas quedando a consideración de quien corresponda tomar la decisión que considere accesible o apropiada. Recordemos que cuando se habla de hardware nos estamos refiriendo a cualquier elemento físico, es decir desde un servidor hasta un pedazo de cable y todo debe ser considerado y analizado.

Factibilidad Técnica – Software

Es un análisis similar al del hardware pero referido al o a los programas de los que se dispone al momento de encarar el desarrollo del proyecto y cual o cuales deberían ser los programas a cambiar, incorporar o necesarios para la implementación del proyecto.

Ejemplos:

- En el desarrollo de una app para celulares, cual sería la versión mínima del sistema operativo del aparato que soportaría nuestra aplicación.
- Si se trata de un cliente que tiene una o más computadoras analizar si todas tienen el mismo sistema operativo y en caso contrario si se puede implementar así o es necesario

unificar los sistemas operativos y en este caso, cual sería la versión mínima que podría soportar nuestro software.

- ¿Es necesario incorporar un software de red o antivirus?
- Desde el lado del equipo de desarrollo, ¿se dispone de todo el software necesario para el desarrollo del proyecto informático? (Intérprete de comandos, compiladores, etc.)

Factibilidad Técnica – Recursos Humanos

La Factibilidad Técnica – Recursos Humanos apunta directamente a establecer las características del equipo de desarrollo, sus habilidades, conocimientos y la forma de organizarlos para que rindan al máximo y se alcancen los objetivos establecidos en los tiempos acordados. Por este motivo (y porque se verá nuevamente que aparecen los Recursos Humanos en otra de las categorías del Estudio de Factibilidad) identificaremos éstos como *Recursos Humanos Internos*.

Organización del Equipo de Trabajo

En cuanto a la organización de los equipos de trabajo, en la actualidad se aplica lo que se conoce como Scrum que es una metodología de trabajo que se inscribe en lo que se conoce como prácticas de Desarrollos Ágiles, ampliamente difundido en el ambiente de las empresas dedicadas a los desarrollos informáticos.

SCRUM

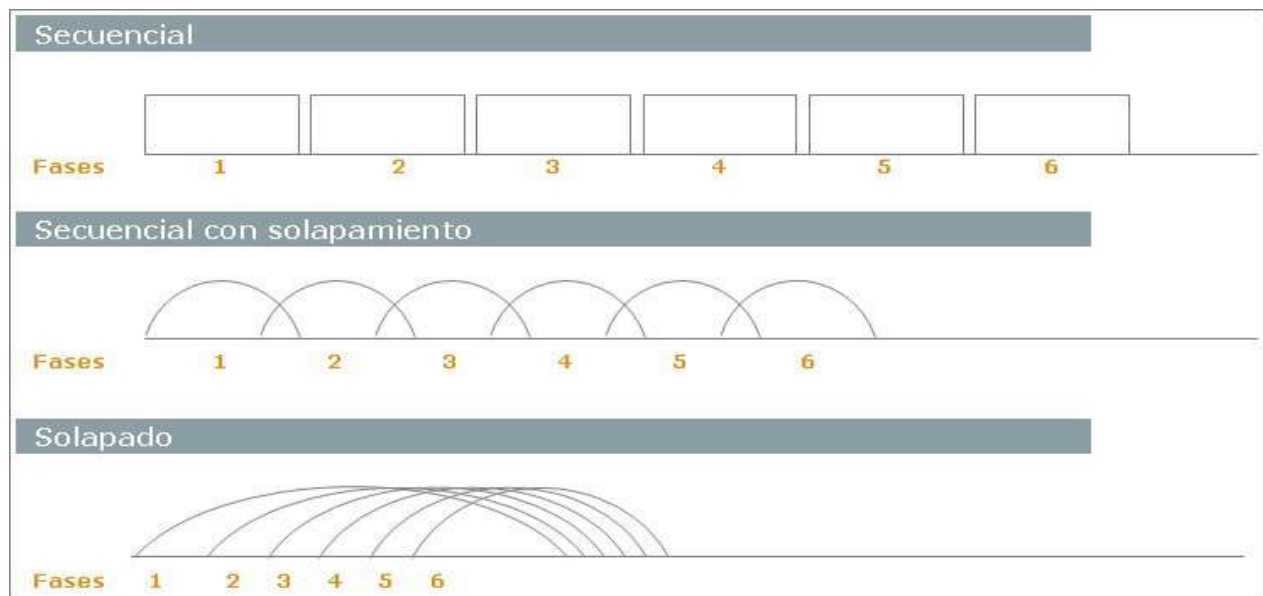
La aplicación de Scrum tiene algunas características fundamentales:

- No es un método de programación sino una forma de organizar el trabajo en equipo.
- Estrategia basada en el desarrollo incremental y no de la totalidad del proyecto. Esto significa que el proyecto va avanzando de manera tal que con cada cierre de etapa se va completando lo que sería la versión final del producto software.

- Calidad del resultado basado en el funcionamiento de equipos auto-organizados más que en los procesos empleados. Esto significa que se apunta a que los equipos de trabajo funcionen sin necesidad de estar permanentemente diciéndoles que deben hacer y que el éxito del trabajo se mide por los resultados y no por la forma en que se desarrollaron los procesos.
- Se basa en la designación de roles y la aplicación de determinadas prácticas. (esto se verá más adelante)
- Apunta a buscar el solapamiento del trabajo por sobre el trabajo secuencial.

Sobre el tema de solapamiento de trabajo por sobre el trabajo secuencial, debemos distinguir la existencia de 3 (tres) formas de desarrollar un proyecto informático, conocidos como Ciclos de Desarrollo.

Ciclos de Desarrollo



<https://commons.wikimedia.org/w/index.php?curid=717094>

Secuencial: Se comienza con la etapa de desarrollo o proceso una vez que se terminó con el proceso que se inició anteriormente.

Secuencial con solapamiento: Se comienza la etapa de desarrollo o proceso poco antes de terminar la etapa o el proceso anterior, cuando las tareas pendientes no tendrán influencia sobre la siguiente etapa.

Solapado: Cada etapa o proceso se inicia poco después de iniciada la etapa o el proceso anterior y se va avanzando hasta el punto donde se tiene que aportar información esencial del proceso para que el siguiente equipo pueda avanzar con sus tareas.

La condición fundamental para la aplicación de los últimos métodos o ciclos de desarrollo es que las etapas o los procesos que se inician después deben depender de alguna manera de las etapas o procesos que se iniciaron anteriormente ya que si no existe esta dependencia no se justifica que una etapa o proceso se desarrolle.

La posibilidad de aplicar cada ciclo de desarrollo está dado por las características del equipo de desarrollo y las características del propio proyecto ya que cuando existe un único desarrollador, este no tiene más alternativa que trabajar bajo un método secuencial. Pero si se cuenta con más de un desarrollador y no se puede establecer la dependencia entre etapas o procesos, tendremos a todos los programadores trabajando de manera secuencial al mismo tiempo.

Roles

La asignación de roles es fundamental en la aplicación de un esquema Scrum. Pero si bien es posible que alguno de esos roles sean ejercido siempre por las mismas personas, la asignación de un rol no es como la elección de un presidente que una vez elegido cumple con el cargo por un tiempo pre establecido sino que se trata de ejercer determinadas actividades en un determinado proyecto, pudiendo cumplir otro rol en un proyecto distinto aún cuando estos proyectos se desarrollen al mismo tiempo.

Los Roles se dividen en dos categorías: Los roles Principales y los roles Auxiliares o Secundarios.

La división entre ambas categorías está dado por el hecho de que los roles principales se cubren exclusivamente dentro del equipo de desarrollo e intervienen en el proceso de desarrollo del proyecto informático mientras los secundarios o auxiliares no participan en el desarrollo pero podrían participar en la implementación, por lo que son externos a la empresa encargada de desarrollar el proyecto.

Una observación importante es que en lo que se refiere a Roles como en lo que se refiere a Documentos las palabras Proyecto y Producto pueden ser consideradas como sinónimos.

Roles Principales

Propietario del Producto o del Proyecto (En inglés: Product o Project Owner):

Es quien interactúa con el cliente discutiendo los aspectos comerciales del proyecto y recibe del cliente los requerimientos específicos del proyecto a desarrollar.

Por esto es el encargado de definir las tareas a desarrollar y de designar algunos de los demás roles principales.

Administrador del Producto o del Proyecto (En inglés: Product o Project Manager):

Es el encargado de asignar las tareas a los desarrolladores en función de las instrucciones recibidas del Propietario y de crear nuevas tareas en función de los requerimientos del cliente ya que una vez finalizadas las negociaciones comerciales con el cliente, el Propietario lo pone en contacto con el Administrador para continuar con diversos aspectos exclusivamente técnicos referidos al proyecto.

También tiene la función de coordinar el trabajo de los desarrolladores o equipos de desarrolladores bajo su conducción y ayudarlos en caso de presentarse algún problema, inconveniente o duda en ellos.

Facilitador (En inglés Scrum Master):

Tiene por tarea fundamental la coordinación general de todos los desarrolladores que se encuentren trabajando en la empresa pudiendo ellos pertenecer a proyectos diferentes. Colabora con los Administradores en la eliminación de obstáculos o interferencias en el trabajo y aportando soluciones o resolviendo dudas operativas. Tiene además una activa participación en lo que

veremos más adelante como Ceremonias o Eventos ya que hace las veces de un coordinador de dichas ceremonias.

Desarrolladores o Equipo Scrum (En inglés Scrum Team):

Son los encargados de dar forma al proyecto realizando las distintas tareas necesarias, sea en el desarrollo como en la corrección o el testeo de los distintos procesos.

Roles Auxiliares o Secundarios**Destinatarios:**

Este grupo está integrado por los Clientes y los Usuarios. Distinguimos aquí Clientes de Usuarios considerando a los primeros como aquellos que nos solicitan el desarrollo del proyecto informático y a los segundos como aquellas personas que usarán nuestro software.

Administradores:

Son las personas o empresas responsables del administrar el o los entornos de implementación del proyecto. Por ejemplo los administradores de red, administradores de base de datos o los administradores de hosting en la nube. Este rol puede o no ser necesario (dependiendo de las características del proyecto informático) o también podría ser ejercido por la propia empresa de desarrollo pese a lo cual se seguiría considerando como un rol Secundario o Auxiliar.

Proveedores:

Son las personas o empresas que proveen bienes o servicios necesarios para la implementación del proyecto informático. Pueden ser proveedores de hardware, de software, de servicios de internet o servicios de hosting en la nube. Este rol puede o no ser necesario (dependiendo de las características del proyecto informático) o también podría ser ejercido por la propia empresa de desarrollo pese a lo cual se seguiría considerando como un rol Secundario o Auxiliar.

Antes de continuar con los siguientes temas es necesario incorporar el concepto de Sprint o Iteración.

Sprint - Iteración: Es una unidad o período de tiempo de desarrollo que marca con su finalización

la fecha en que los procesos deben estar terminados y listos para incorporarse a la versión del proyecto que se completa para esa fecha (Desarrollo incremental) pudiendo haberse empezado con las tareas de programación en un sprint anterior. Recordemos que es responsabilidad del equipo de desarrollo no sólo la programación de los procesos sino también el testeado o prueba de los mismos para confirmar que no surjan errores de programación en tiempo de ejecución y que el proceso efectivamente haga aquello para lo que fue creado, quedando por lo general a consideración del cliente opiniones de tipo formal (apariencia) y la aprobación de la cuestión funcional.

Documentos y Ceremonias de Scrum

Los Documentos y las Ceremonias (o Eventos) son parte de las prácticas que implica la aplicación de Scrum en la organización de equipos de trabajo en Proyectos Informáticos.

Documentos de Scrum

Existen 3 (tres) documentos fundamentales:

Pila de Producto o Pila de Proyecto (Product Backlog o Project Backlog en inglés)

Documento de alto nivel que se elabora en la etapa de planificación del Proyecto y contiene todos los requerimientos necesarios para el desarrollo y mantenimiento del Proyecto Informático.

Es dinámico y abierto, esto quiere decir que sufre modificaciones a lo largo del tiempo y que estas modificaciones pueden ser realizadas por distintos integrantes del equipo de desarrollo (cualquiera de los roles principales) pese a que la elaboración en primera instancia es responsabilidad del Propietario del Proyecto ya que él es quien está en contacto directo con el cliente, recibe todos sus requerimientos y establece los procesos o tareas que pueden ser necesarias para cumplir con lo solicitado por el cliente.

Como a lo largo de la etapa de desarrollo y en la etapa de mantenimiento del software, el Administrador del Proyecto mantiene contacto con el cliente, este puede agregar o eliminar información de este documento y por sus características puede ser modificado aún por miembros del

Equipo Scrum.

Podría pensarse este documento como si fuera un archivo de Excel con 9 columnas con los títulos o encabezados que se indican a continuación y que en cada fila contendría un proceso, una tarea o una funcionalidad a desarrollar para cumplimentar o corregir el Proyecto Informático.

Id – Es un número o clave que identifica la tarea en particular. Se emplea para hacer referencia a cada tarea y la condición es que sea una clave única, irrepetible y no reutilizable.

Prioridad – Es una clave establecida por la empresa de desarrollo que indica que tan urgente es necesario realizar la tarea correspondiente. Esta puede ser una clave numérica, alfabética, de colores o simple y más comúnmente palabras tales como Muy Alta – Alta – Media – Baja – Muy Baja.

Título – Breve texto que da una idea de que se trata el proceso, la tarea o la funcionalidad.

Descripción – Explicación detallada de lo que se debe realizar o corregir.

Estimación de Tiempo – Tiempo que se estima que puede llevar realizar este proceso, tarea o funcionalidad.

Número de Sprint – Sprint en el que debe estar terminado el proceso, tarea o funcionalidad contemplando los tiempos de desarrollo (programación) y prueba del producto terminado.

Estado – Es una clave establecida por la empresa de desarrollo y que indica la situación del proceso, tarea o funcionalidad. Al igual que la Prioridad puede ser una clave numérica, alfabética, de colores o simplemente palabras tales como Pendiente, En Pausa, En Desarrollo, Lista Para Probar, Descartada, Terminada, etc. Esta clave depende no solo de cada empresa sino que también puede variar entre distintos proyectos desarrollados por la misma empresa.

Responsable – Es el nombre de la persona que tiene a su cargo la tarea. Ya sea la persona que tiene que hacerla, la persona que la está haciendo o la persona encargada de hacer las pruebas.

Comentarios – Sirve para que las personas involucradas en la vida de esa tarea se comuniquen advertencias o avisos.

NOTA: Este documento puede tener más datos (columnas) ya que cada equipo de desarrollo(empresa o

equipo dentro de una misma empresa) pueden determinarlo de acuerdo con lo que establezca cada Propietario de Proyecto pero sirven como ejemplo de la mínima información necesaria para una correcta documentación de los procesos/tareas/funcionalidades.

Este documento se va completando a medida que se van determinando los procesos, tareas o funcionalidades, es decir que no se hacen de una manera ordenada (así como no aparecían los procesos en orden en el momento del relevamiento).

Luego de esto se hace la determinación de los tiempos estimados para completar cada tarea y en función de esa estimación y de la duración de cada sprint, se establece en cuál de ellos debe estar terminado el desarrollo y prueba del proceso, tarea o funcionalidad.

Una vez completados esos datos, y continuando con la idea de un documento de Excel, podemos ordenar este documento por el número de sprint + prioridad de manera que en la parte superior aparecerían las tareas correspondientes al Sprint 1 y dentro de esto las tareas ordenadas de las más prioritarias a las menos prioritarias. Luego seguirían las tareas correspondientes al Sprint 2 y así con todos los Sprints.

Ahora bien... Si de este documento de Excel copiáramos todas las tareas asignadas a un determinado Sprint (Por ejemplo el 1) y lo copiáramos en otro documento de Excel, tendríamos el documento conocido como **Pila de Sprint (Sprint Backlog en inglés)**. Entonces podríamos decir que la Pila de Sprint es un subconjunto de la Pila de Producto y que contiene todas las tareas a realizar en un mismo Sprint.

El tercer documento es el denominado **Cuadro de Avance (Burn Down Chart en inglés)**. Pesemos en una gran plancha de telgopor o de corcho con distintas columnas que llevaran por título cada uno de los Estados establecidos para las tareas por el equipo de desarrollo empezando por la izquierda por el estado original de cada tarea (Por ejemplo Pendiente) y terminando a la derecha con el estado final de cada tarea (Por ejemplo Terminada) y colocando entre medio los distintos estados ordenados de manera lógica (Por ejemplo: En Pausa, En Desarrollo, Lista Para Probar).

Pensemos también en representar cada tarea (fila del Excel de la Pila de Producto) mediante una tarjeta en la que se pudiera ver la Identificación, Título y Prioridad de cada tarea.

Ahora colocamos cada tarjeta (tarea) en la columna correspondiente... Pero al estar en la etapa inicial, es decir, todavía no hemos empezado a trabajar, todas las tarjetas estarían en la columna que lleva por título Pendiente quedando todas las demás columnas sin ninguna tarjeta debajo del título.

A medida que los proceso empiezan a cambiar de estado, vamos sacando la tarjeta correspondiente de la columna de Pendiente por lo que vamos observando que esa pila de tarjetas/tareas se va reduciendo... Como si se fuera consumiendo... De ahí el término en inglés Burn Down Chart y el concepto de Cuadro de Avance ya que nos permite determinar a simple vista como va avanzando o no el proyecto y llevar un control de lo que puede faltar para llegar al final del trabajo.

Como a lo largo del proceso de desarrollo o de mantenimiento pueden ir apareciendo nuevas tareas a desarrollar (nuevas tarjetas) y estas inician el circuito en la primera columna de la izquierda daría la impresión de que esta columna empieza a crecer en lugar de decrecer... Por esta razón, el nombre en inglés que yo le doy a este Cuadro de Avance es el de **Burn Down/Up Chart**.

Así como la Pila de Producto es el registro de la totalidad de las tareas necesarias, el Cuadro de Avance permite llevar un control de la productividad del equipo de trabajo.

En el caso de que una empresa de desarrollos informáticos pudiera estar trabajando en varios proyectos a la vez, cada proyecto debería tener sus propia Pilas de Proyecto y de Sprint y su propio Cuadro de Avance ya que puede que no coincidan las duraciones y fechas de finalización de los sprints de cada proyecto así como cada proyecto puede tener su lista de posibles estados.

Ceremonias o Eventos de Scrum

Existen, básicamente, 6 (seis) ceremonias en la aplicación de Scrum. Estas se realizan según las necesidades de cada equipo de trabajo por lo que si bien hay tres de ellas que se hacen siempre (las dos primeras y la cuarta), puede que cada empresa en particular realice la tercera y/o las últimas dos.

Es decir... Cada empresa de desarrollos informáticos y aún dentro de cada empresa cada equipo de trabajo (Administrador de Proyecto + Equipo Scrum) pueden determinar la necesidad de realizar determinadas ceremonias y no realizar las demás.

Para la explicación de cada ceremonia voy a explicar la forma en que se realizan en la empresa en la que trabajo ya que las características pueden variar de acuerdo a las necesidades y metodología de trabajo de cada empresa de desarrollo.

Scrum Diario: Es una ceremonia que se realiza todos los días, tiene una duración limitada (15 a 30 minutos) y de la que participan el o los Administradores de Proyecto, los desarrolladores (equipo scrum) y el facilitador. Los Propietarios de Proyecto pueden estar presentes pero por lo general sólo para informarse, aunque por su experiencia pueden participar para aclarar algún tema o aportar soluciones a los posibles problemas.

En esta ceremonia se responden las siguientes preguntas:

- ¿Qué se hizo desde la última ceremonia?
- ¿Qué se va a hacer hasta la próxima ceremonia?
- ¿Se encontró algún problema o inconveniente que frenara una tarea?
- ¿Algún equipo de desarrollo/programador le entregará algo a otro equipo/programador?

En caso de que la empresa esté trabajando con muchos proyectos a la vez, estas ceremonias pueden realizarse en distintos horarios para cada proyecto pero siempre con la presencia del Facilitador en todas.

Scrum del Scrum: Es una ceremonia que se realiza con una frecuencia pre establecida (7 días, 15 días, etc.) en la que participan el o los Propietarios de Proyectos con el o los Administradores de Proyectos y el facilitador y sirve, fundamentalmente, para informar a los Propietarios acerca del avance de los proyectos o de cualquier novedad al respecto.

En esta ceremonia se responden las siguientes preguntas:

- ¿Qué se hizo desde la última ceremonia?

- ¿Qué se va a hacer hasta la próxima ceremonia?
- ¿Se encontró algún problema o inconveniente que frenara una tarea? ¿Cómo se resolvió? O bien, en esta se resuelve como solucionarlo.

Poker and Grooming: Es una ceremonia que se realiza con cierta periodicidad (no necesariamente una frecuencia fija) en función de la cantidad de trabajo nuevo que pueda haber ingresado a los proyectos o las tareas que se encuentren sin una estimación de tiempos. En esta ceremonia que participan todos los integrantes de la empresa, desde los propietarios hasta los desarrolladores ya que en ella se establecen colectivamente las estimaciones del tiempo que puede requerir terminar cada tarea. Sirve además para que los programadores más experimentados y que más conocen el proyecto aclaren dudas u orienten con precisión a los programadores novatos sobre las características de cada tarea a desarrollar, especialmente cuando se está en la etapa de mantenimiento. Con esta estimación de tiempos se puede asignar cada tarea a un sprint determinado.

Planificación del Sprint: Es una ceremonia que se realiza al inicio de cada sprint en la que participa el Administrador del Proyecto, el Facilitador y los Desarrolladores y en la cual se establecen las estrategias con las que se encararán las tareas correspondientes al Sprint, el orden en las que se van a desarrollar, se aclaran dudas sobre las tareas y se establece quien o quienes de los integrantes del equipo de desarrollo podrían encargarse de cada una de ellas.

Revisión del Sprint: Es una ceremonia que se realiza al finalizar cada Sprint en la que se presentan aquellas tareas o procesos que se terminaron, se indica los que se descartaron y se informa de los que no se pudieron terminar durante el Sprint señalando los motivos del descarte o de la demora. Participan el Administrador, el Facilitador y los desarrolladores.

¿Qué sucede con las tareas no terminadas? Simple: se las asigna a otro Sprint, ya sea el siguiente u otro futuro. Esto dependerá de la urgencia o prioridad de la tarea.

Retrospectiva del Sprint: Es una ceremonia que se realiza al finalizar cada Sprint, luego de la Revisión en la que se analiza la forma en que se trabajó durante el Sprint a los efectos de implementar, de ser posible, mejoras en la dinámica del trabajo para aumentar la productividad.

Participan el Administrador, el Facilitador y los desarrolladores.

Hay que recordar que la fecha de finalización del Sprint, señala la fecha en la que debe terminarse el desarrollo y prueba de las tareas o procesos, aunque el inicio de las tareas puede darse en un Sprint anterior. Todo dependerá de la disponibilidad de recursos ya que si se hizo una estimación pesimista de tiempos en alguna o algunas de las tareas, puede que todo lo asignado a un determinado Sprint termine de desarrollarse y probarse antes de la fecha límite (Final del Sprint) por lo que, en lugar de adelantar la fecha de finalización del Sprint, lo que se hace es empezar a adelantar trabajo que deberá ser terminado en algún sprint posterior. Esto lo va estableciendo el Administrador de Proyecto con la colaboración del Facilitador.

NOTAS FINALES SOBRE LAS CEREMONIAS

Como la aplicación de Scrum se encuadra dentro de lo que se conoce como Metodologías Ágiles, este concepto es el que prevalece (la AGILIDAD) por lo que, así como existe cierta libertad en el formato de los documentos y los roles pueden ser cubiertos por personas que no son las designadas oficialmente para ejercerlo (Por ejemplo, en mi caso soy Administrador de uno de los proyectos que desarrolla la empresa para la que trabajo, pero por mis horarios en la escuela a veces no puedo cumplir con mi función, pese a lo cual alguien lo hace), existe también cierta flexibilidad en la celebración de las distintas Ceremonias.

Por ejemplo:

Si bien se establecen horarios para los Scrum Diarios y fechas para los Scrum del Scrum, estas fechas u horarios pueden cambiar a lo largo del tiempo. Algún Scrum Diario puede no realizarse, pero no pueden (no deberían) cancelarse por más de un día. La fecha o frecuencia pre establecidas para un Scrum del Scrum pueden cambiarse circunstancialmente o de manera permanente.

Alguna o algunas de las ceremonias pueden directamente no realizarse (Por ejemplo las Revisiones y Retrospectivas del Sprint) ya que la realización de los Scrum Diarios hacen que no sean necesarias las tareas de esas ceremonias por tenerse la información permanentemente actualizada con reuniones diarias.

La clave para una metodología ágil de trabajo es justamente estar abiertos a la posibilidad de realizar cambios sobre la marcha y no encasillarse en formas y tiempos fijos e inamovibles, siempre con el objetivo de mejorar la productividad de la empresa.

Beneficios de aplicar Scrum

- Flexibilidad a cambios. Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- Reducción del Time to Market. El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- Mayor calidad del software. El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- Mayor productividad. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- Maximiza el retorno de la inversión. Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- Predicciones de tiempos. A través de este marco de trabajo se conoce la velocidad

media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.

- Reducción de riesgos El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.