

Práctico 14 - Comparación de Enfoques para Gestión del Desarrollo de Productos de Software

Consigna:	Presentar los diferentes enfoques para la gestión del desarrollo de productos de software, considerando los enfoques de gestión basada en procesos definidos (Tradicional), Scrum y Kanban y realizar una tabla comparativa que explique cada enfoque.
Objetivo:	<p>Que el estudiante sea capaz de abordar la temática, no solo a partir de lo brindado por el docente durante las clases, sino también a través de investigación bibliográfica.</p> <p>Que consiga identificar aspectos que puedan ser comparados para obtener una decisión acertada en la elección de un determinado enfoque de gestión, en el momento en que lo necesite.</p>
Propósito:	Aplicar los conceptos de Gestión del Desarrollo de Productos de Software basados en procesos definidos y basados en procesos empíricos
Entradas:	<p>Conceptos teóricos sobre los temas desarrollados en clase.</p> <p>Bibliografía sobre el tema.</p> <p>Artículos científicos de interés.</p>
Salida:	<p>Se evaluará que:</p> <ul style="list-style-type: none">● Introduzca brevemente el trabajo a presentar y los enfoques de gestión con la información de interés.● Realice un cuadro comparativo de los diferentes aspectos a contrastar.● Presente una conclusión de elaboración propia del grupo en relación al tema abordado.

Instrucciones:	<ul style="list-style-type: none">● En grupos trabajarán la consigna.● Preparan un informe que contenga el trabajo● Subirán en el aula virtual en la sección indicada por el docente
Observaciones:	Deben referenciar el material bibliográfico utilizado.

Tabla comparativa:

Item a comparar	Tradicional	Ágil	Lean
Proceso	<p>Basado en procesos definidos, estables y estables, se asume que es posible detallar las principales variables de un proyecto (requisitos, tareas, recursos, costos y tiempos) y predecir su comportamiento a largo plazo.</p> <p>Se crea una cadena de actividades donde cada una recibe una entrada, ejecuta una serie de tareas preestablecidas y produce una salida esperada. Cada una de estas tareas está asociada a un determinado rol que posee ciertas habilidades para transformar la entrada en una salida.</p> <p>No se adapta adecuadamente a los cambios</p>	<p>Basado en procesos empíricos, acepta la complejidad y en vez de pretender controlar la realidad, su propósito es medir constantemente los resultados y adaptar el comportamiento en consecuencia.</p> <p>Es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento).</p> <p>Manteniendo entonces un intervalo de tiempo pequeño y constante, el control puede ser bastante granular al revisar sistemáticamente el resultado del trabajo realizado durante ese intervalo de tiempo y ajustando las variables del contexto y permitiendo así una constante estabilización y optimización del sistema.</p> <p>El rol del equipo de trabajo es fundamental en un proceso empírico ya que los individuos pasan a ser parte de un conjunto interdependiente de personas que forman un equipo, donde el éxito o el fracaso es colectivo y no individual.</p> <p>Especialmente preparado para cambios durante el desarrollo del proyecto.</p>	

Gestión de requerimientos	<p>El involucramiento del cliente también es un factor clave para la Gestión de Requerimientos Tradicional El cliente tiene el conocimiento acerca del sistema que será desarrollado, conocimiento que se espera sea transferido al equipo de desarrollo.</p> <p>Aunque se debe tener en cuenta que cliente está involucrado principalmente durante algunos etapas de los proceso tradicionales, como por ejemplo en el PUD: elicitación, luego durante el proceso de validación de requerimientos y al final para</p> <p>La aceptación del producto, prácticamente no tiene participación durante la etapa de desarrollo.</p>	<p>La participación e involucramiento del cliente es a lo largo del proceso de desarrollo completo.</p> <p>Los aspectos fundamentales del enfoque ágil para el tratamiento de los requerimientos es el Involucramiento del Cliente, esto significa tener al cliente “accesible” o “en el sitio”.</p> <p>Este es el cimiento para la retroalimentación rápida y la comunicación entre los involucrados. Ambos conducen a una mejor comprensión de las necesidades y expectativas en ambos lados (equipo de desarrollo y cliente).</p> <p>El involucramiento del cliente es el primer objetivo del enfoque ágil, que asume un usuario “ideal” Por ejemplo en SCRUM el representante del cliente, el Product Owner, que puede responder correctamente todas las preguntas que el equipo de desarrollo pudiera hacerle y tiene el poder de tomar decisiones vinculantes. Es importante recordar que el Product Owner debe tener un entrenamiento previo para poder ejercer este rol adecuadamente.</p>
----------------------------------	--	--

Artefactos			
Roles	<p>Depende del proceso que se seleccione, como por ejemplo en el caso del proceso unificado de desarrollo (PUD): dependiendo del flujo de trabajo podremos encontrar:</p> <ol style="list-style-type: none"> 1. Analista de sistemas 2. Arquitecto 3. Especificador de casos de uso 4. Diseñador de interfaz de usuario 5. Ingeniero de casos de uso 6. Ingeniero de componentes 7. Integrador de sistemas 8. Ingeniero de pruebas 9. Ingeniero de pruebas de integración 10. Ingeniero de pruebas de sistema 	<p>Scrum se basa en el compromiso conjunto y la colaboración entre los miembros del equipo. La transparencia entre todos es fundamental para poder inspeccionar la situación real del proyecto y así poder hacer las mejores adaptaciones que permitan conseguir el objetivo común.</p> <p>1. Cliente (Product Owner): Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo.</p> <p>El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog).</p> <p>2. Facilitador (Scrum Master): es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.</p> <p>3. Equipo (Team): Consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto "Terminado", que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de</p>	<p>Kanban no establece ningún rol en absoluto.</p> <p>Sin embargo, dos roles han emergido de la práctica común y ahora se definen en el método en sí mismo.</p> <p>1. El Gestor de Peticiones de Servicio: es responsable de entender las necesidades y expectativas de los clientes, y de facilitar, seleccionar y ordenar los elementos de trabajo en la Reunión de revisión de la cartera de trabajo.</p> <p>2. El Gestor de Prestación de Servicio (Service Delivery Manager) es el responsable del flujo de trabajo entregando los elementos seleccionados a los clientes y facilitando la Reunión de Kanban y la planificación de la entrega.</p> <p>En Kanban, los equipos pueden ser multifuncionales o especializados.</p> <p>El tablero no necesita estar en manos de un equipo específico sino que puede estar compartido por varios equipos.</p>

		Desarrollo participan en la creación del Incremento.	
Participación del cliente	<p>El cliente solo interactúa con el equipo de desarrollo mediante reuniones</p> <p>Como el proyecto no debería tener muchos cambios, no suele haber una correcta retroalimentación con el cliente, lo cual suele acarrear un desfase importante en el proyecto</p> <p>Se prioriza en base a la visión propia de la dirección del proyecto y no siempre en base a la visión del negocio, mientras que en las metodologías ágiles priorizamos más en base a las necesidades del cliente.</p> <p>Cuando se presenta un prototipo, el cliente evalúa en forma temprana el producto, e interactúa con los diseñadores y desarrolladores para saber si se está cumpliendo con las expectativas y las funcionalidades acordadas.</p>	<p>La relación entre el cliente y el proveedor del proyecto debe estar basada en el principio de obtener el mayor beneficio posible en todos los puntos del proyecto. En lugar de estar ligados por un contrato férreo de alcance, tiempo y coste, las dos partes asumen que habrá cambios para que cliente pueda obtener lo que realmente necesita, no lo que está escrito en un documento inicial o seguir un plan inicial que vaya perdiendo su sentido. Debe existir transparencia en la ejecución del proyecto para facilitar esta relación. El cliente colabora con el equipo: Para cada nuevo objetivo/requisito, conjuntamente se hace una identificación inicial de sus condiciones de satisfacción (que se detallarán en la reunión de planificación de la iteración). El cliente obtiene del equipo la re estimación de costes de desarrollo para completar los objetivos/requisitos siguientes. El equipo ajusta el factor de complejidad, el coste para completar los requisitos y su velocidad de desarrollo en función de la experiencia adquirida hasta ese momento en el proyecto. El cliente re prioriza la lista de objetivos/requisitos en</p>	<p>Si bien se mantiene una relación con el cliente, lean otorga una mayor importancia a la calidad y la eficiencia, por lo que su objetivo es eliminar desperdicios quedándose únicamente con lo que le aporte valor al proceso, y por ende al cliente.</p> <p>En lugar de aprender del lanzamiento de un producto con retroalimentación en base a los comentarios de los clientes, se tiene en cuenta lo que el mercado demanda en ese momento y se elimina del proceso todo aquello que no sirve en pro de la calidad, es decir, que el producto no tenga ningún defecto.</p> <p>Gira en torno al enfoque en los clientes y a darles el producto que desean de la manera más eficiente posible</p> <p>El resultado final debe crear valor para el cliente, y es el único objetivo del proceso de desarrollo.</p> <p>El cliente debe obtener siempre exactamente lo que espera.</p>

		función de estas nuevas estimaciones	
Dominio	Funcionan en situaciones u organizaciones no cambiantes, que realizan siempre los mismos procedimientos y tareas.	Funcionan en dominios complejos. Escenarios cambiantes y dinámicos.	
Ciclo de vida	Secuencial, plantean que las fases del proceso deben ejecutarse en forma completa y en orden, avanzando con el cien por ciento de los artefactos que deben construirse en cada fase antes de poder pasar a la siguiente.	Iterativo/incremental: plantean que puede avanzarse con una porción del producto que debe construirse, evolucionando a través de cada una de las etapas hasta obtener una versión completa, para una parte del producto y luego repetir el proceso hasta terminar el desarrollo completo.	
Tamaño del equipo	No hay un tamaño definido para los equipos, está formado por: Los roles a considerar son: Cliente, Líder de proyecto, Analista, Arquitecto, Diseñador, Programador y Tester.	Equipo multidisciplinario. Lo suficientemente chico para ser ágil y lo suficientemente grande para poder cumplir lo que tiene que hacer, recomendado no menos de 3 y no más de 9.	Equipo multidisciplinario o especializado. No hay parámetro definido en cuanto a cantidad, solo debe cumplirse que en ninguna de las colas de trabajo haya tiempo ocioso ni sobrecargo de tareas.
Objetivo	Cumplir con un plan definido de antemano. Se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y	Kanban es una herramienta de proceso que se enfoca en la entrega de servicios para un cliente, el cual pide el trabajo o aquellas necesidades que son identificadas, y es quien acepta y da el visto bueno de las entregas del trabajo completado. Tiene como objetivo: <ul style="list-style-type: none">● Incrementar la fuerza de trabajo	En particular scrum es un proceso que está en el contexto de los procesos empíricos para que en el desarrollo de software se puedan crear nuevos productos. Se realizan entregas parciales y regulares del producto final, priorizadas por un objetivo en particular: entregar valor al cliente. Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados

	<p>en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto.</p>	<ul style="list-style-type: none"> ● Minimizar el stock de inventario ● Recortar tiempos muertos ● Incrementar el nivel de servicio al cliente ● Incrementar productividad ● Reducción de desperdicios de materia prima ● Reducción de desperdicio de tiempo ● Reducción de Inventario en Proceso 	<p>pronto, donde los requisitos son</p> <p>cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.</p>
Métricas	<p>Tenemos 3 tipos de métricas:</p> <p>Métricas de proceso. Métricas de proyecto. Métricas de producto.</p> <p>Y para un proyecto de software las métricas se basan en el:</p> <p>Tamaño del producto Esfuerzo Tiempo (Calendario) Defectos</p>	<p>Lead Time = Vista del cliente, representa el tiempo que transcurre desde que ingresa la pieza de trabajo hasta que esta terminada. Para mi cliente la entrega de valor tiene que ver con esta métrica.</p> <p>Cycle Time = Vista interna, saliendo de esa cola de acumulacion inicial tiempo que sucede entre el inicio y el final del proceso, para un ítem de trabajo dado</p> <p>Touch Time = tiempo específico en el que cada uno de los recursos que está en mi proceso está trabajando en una determinada pieza de trabajo, cuanto tiempo paso el ítem en columnas "trabajado en curso".</p> <p>Diagrama de flujo acumulativo</p>	<p>Capacidad: Es una estimación de cuánto trabajo puede completarse en un período de tiempo basado en la cantidad de tiempo ideal disponible del equipo. Es teórica, y resulta de gran utilidad al momento de estimar, derivando de esta la velocidad.</p> <p>Velocidad: observación empírica de la capacidad del equipo para completar el trabajo por iteración, comprobable entre iteraciones de un equipo dado. No se estima, se calcula. Está determinada por la cantidad de Stories Points que el Product Owner aceptó, y se toma en la Sprint Review.</p> <p>Burndown</p>

CONCLUSIÓN

Como conclusión podemos decir que las metodologías tradicionales de desarrollo de software han quedado obsoletas en determinados sectores, en los que la propia demanda de los usuarios es más rápida y cambiante que la capacidad de producción de las empresas ancladas a las viejas metodologías de gestión de proyectos de sistemas informáticos.

Este gran impacto en las tecnologías, ha generado la necesidad de encontrar y crear nuevas metodologías de trabajo y gestión, que aseguren la entrega en tiempo y forma del producto. Esta necesidad de calidad, eficiencia, flexibilidad y rapidez en la entrega de un producto informático se volvió prioridad y en conjunto con su necesidad se crearon las nombradas Metodologías Ágiles.

Los métodos ágiles y los tradicionales no son competidores directos. Y ambas metodologías, pueden fracasar si son mal implementadas, gestionadas y administradas.

En cuanto a Scrum y Kanban, como se detalló, no hay tantas diferencias. Son maneras distintas de llegar al mismo resultado.

Scrum es más compleja, estricta y difícil de adoptar. En su núcleo se encuentran los sprints de Scrum que establecen objetivos y plazos claros, dan control total al proyecto y funcionan bien para proyectos grandes y de rápido movimiento.

Kanban es el método más simple y fácil de adoptar. Visualiza el trabajo y el proceso de trabajo en un tablero Kanban que garantiza que todos los miembros del equipo estén en la misma página y revela cuellos de botella en el flujo de trabajo mientras se mantiene flexible en la producción.