TECNOLOGICO NACIONAL CEICOM

CARRERA DE SISTEMAS INFORMATICOS

F

Practica01

MÉTODOS AGILES

Materia: Análisis y Diseño de Sistemas

Elaborado por: Fabrica Plaz Estevan

Docente: Ing. Baltazar Llusco Ever Jaime

Fecha: 13 de septiembre de4 2018

Cochabamba- Bolivia

Proyecto.- un proyecto es un conjunto de actividades relacionadas que utilizan recursos para cumplir con un objetivo deseado dentro de un plazo de tiempo delimitado.

Las actividades son las tareas que se deben realizarse para complimiento del objetivo, mientras que los recursos son los elementos requeridos para cumplir las primeras. Todo proyecto tiene limitantes que pueden ser de orden económico, técnico operativo, etc. A demás presenta las siguientes características:

1. Fecha de inicio y fin
2. Definición de tareas y calendario
3. Sucesión de actividades.
4. Necesidad de recursos.
5. Producción de un resultado único.

Decimos que un proyecto es exitoso cuando:

1. Se termina dentro del tiempo pactado, con el presupuesto comprometido y con la funcionalidad deseada.
2. El cliente se siente satisfecho con el producto.
3. El producto ofrece las ventajas comerciales esperadas.
4. El equipo de proyecto cree que su participación fue valiosa.
5. El proyecto eleva el nivel de conocimiento y permite mejorar a futuro.

Por otro lado lo consideramos un fracaso cuando:

1. Se exceden los costos o los tiempos.
2. El proyecto no cumple con lo prometido.
3. El cliente no se encuentra satisfecho.
4. El proyecto cumple a nivel técnico, pero no ofrece soluciones al negocio.

Los factores que influyen en el fracaso son multiples e infinitos, pudiendo resumirse en los que mencionamos a continuación:

1. Falta de personal capacitado.
2. Error en la selección de recursos.
3. Falta de capacidad administrativa.
4. Falta de comunicación.
5. Incompleto análisis de requisitos.
6. Problemas técnicos de diseño.
7. Fallas en el planeamiento.
8. Fallas en la ejecución de tareas.

Lamentablemente, en el mundo informático los proyectos suelen presentar más fracasos que éxitos. John Avellner utilizo un estudio realizado por accenture para demostrar la situación actual de los proyectos de software y establecer los puntos que considera necesarios para alcanzar el éxito. A partir de eso, podemos decir lo siguiente:

En los proyectos de IT solo el 29% es un éxito.

El costo previsto es sobrepasado en un 56%.

El tiempo es excedido en un 84%

Las técnicas que se pueden utilizar para lograr el éxito es aplicando las siguientes reglas:

1. Dividir en fases cortas para que cada una ofrezca un beneficio inmediato al negocio.
2. Las estimaciones deben ser realistas, considerando días de 6 horas reales de trabajo y meses 15 días, por ejemplo.
3. Los proyectos son realizados por recursos humanos para personas, por lo tanto debemos realizar estas tareas:
   - Seleccionar personal adecuado.
   - Tomar decisiones cuando el proyecto tiene problemas internos.
   - Orientar al equipo para conseguir objetivos concretos.
   - Validar con frecuencia y en intervalos regulares la calidad de los resultados.
4. La comunicación es un factor realmente fundamental, la transparencia y la exactitud deben ser corrientes.
5. Aplicar el esfuerzo en el lugar adecuado, incrementando funcionalidad.
6. La fase beta debe realizarse por duplicado, la primera con personal local y la otra con todos los equipos del proyecto.

Gestión de proyecto.- La gestión de proyectos es la disciplina encargada de organizar y administrar recursos para poder llevar a cabo los proyectos cumpliendo con las restricciones pactadas. La administración de proyectos es una tarea compleja con gran cantidad de variables originadas principalmente por el resultado, único producto del proyecto. El origen de la gestión de proyectos como lo conocemos actualmente puede ser situado en los años 50, donde desarrollaron algunas técnicas matemáticas que comenzaron a aplicar la ejecución de estos los modelos matemáticos denominados **PERT** (Program Evaluation and Review Technique) y **CPM** (Critical Path Method) son aun utilizados.

Al mismo tiempo se aplican nuevas técnicas, herramientas para gestión de **c**ostos y el control de la calidad. En 1969 se creó el Proyect Management Institute (PMI) para difundir y mejorar las técnicas de gestión de proyectos, paralelamente en Europa aparecía la **Internacional Proyect Managenment Association (IPMA).** Actualmente, el PMI sigue siendo organismo que dicta los conocidos estándares.

Nuevo escenario.- En este escenario competitivo, las variables más importantes, siguiendo a Juan Palacio en su libro flexibilidad con Scrum, son las siguientes:

- Retroalimentación del producto y el entorno.
- Mayor innovación del producto.
- Reducción de tiempos
- Salida inmediata al mercado.
- Los productos deben evolucionar, no están terminados.

Requisitos impredecibles.- Los requisitos son predecibles, podemos afirmar que salvo excepciones, como por ejemplo los proyectos aeroespaciales (que consumen muchos recursos, tiempo y dinero), todos los demás tienen requisitos cambiantes.

Gestión ágil de proyectos.- La gestión ágil intenta convivir con la idea, y a la vez fomentarla, de que no existen productos finales. Todos los productos son versiones beta en constante mejora. Por lo tanto, la gestión ágil intenta responder ante los cuatro nuevos valores de la industria: **valor, tiempo, fiabilidad** y **agilidad.** Dar un mayor valor al producto es fundamental este factor se encuentra compuesta por la **innovación** y la **flexibilidad.** Los valores de la industria son cumplidos mediante la aplicación de distintas prácticas como observamos a continuación:

- Fases de desarrollo superpuestas.
- Incrementos de funcionalidad.
- Entrega temprana del producto.
- Mayor contacto con el cliente.

El desarrollo ágil es un proceso iterativo que presenta, en menor o mayor detalle, las fases que mencionamos a continuación:

- Concepto
- Especulación
- Exploración
- Revisión
- Cierre

Concepto.- A partir de los deseos o requerimientos del cliente y mediante la interacción y participación con el equipo, se desarrolla la visión del producto.

Especulación.- Permite que el equipo realice distintos avances sobre las posibles construcciones y sus limitaciones. Esta fase se repite continuamente a lo largo de desarrollo. Y además en esta etapa se llevan las siguientes tareas:

- Detalle del producto.
- Armado de requisitos.
- Determinación de tareas.
- Determinación de plan de entrega.
- Determinación de los responsables de las tareas.

Exploración.- En esta fase todos los integrantes del equipo desarrollan sus tareas. El objetivo de la fase es el incremento de funcionalidad propuesto en la etapa anterior.

Revisión.- Con la finalidad de la ejecución de las tareas se alcanza el producto real, que debe ser utilizado y comparado con la propuesta inicial.

Cierre.- se lleva al fin de la iteración y se entrega el producto pactado, el cual puede seguir siendo desarrollado con las mismas técnicas.

De acuerdo a estas características y fases de desarrollo, podemos pensar que el desarrollo y la gestión ágil pueden adaptarse mejor a ambientes complejos. Además de todos los elementos que han impactado en el desarrollo de software, tradicionalmente se han utilizado pautas que no son siempre reales, denominadas **mitos del desarrollo ortodoxo.**

- **Mito de la especificación:** Una vez que tengamos el análisis de requisitos completos comenzaremos a diseñar. Esta calase es uno delos componentes más importante, que guía el proyecto hacia el fracaso.
- **Mito del mantenimiento.-** el mantenimiento es necesario para corregir el software. Debemos remarcar que el software no tiene desgaste ni degradación en sí mismo y de la misma del contexto técnico, tenemos que tener en cuenta los cambios de negocio que se producen alrededor de un software que se encuentra implementando.
- **Mito de caja negra.-** Si el software ofrece una salida correcta a mi entrada, el problema está solucionada. Es fundamental tener en cuenta en los desarrollos actuales que no solo debemos

considerar al software como un emisor de respuesta, sino que sus procesos deben ser refinados a fin de incrementar su calidad.

Estas mitos en el marco de trabajo ideal hace que la decantación por técnicas agiles no sea prioritaria. Pero en los contextos de cambio surgieron las metodologías agiles, que hasta el día de hoy se siguen expandiendo y asentando.

Crystal clear.- Es un conjunto de metodologías con un código genérico común para el desarrollo de software, difundida por Alistair Cockburn. Plantea que un software es “un juego económico de cooperación, basado en objetos, buscando un equilibrio entre la ambición (terminar el proyecto pronto) y la inversión (actividades pensando en el futuro)”. En el crystal se miden los proyectos de acuerdo a tamaño y complejidad, y el objetivo es tener un marco de trabajo para cada tipo de proyecto. Y así de acuerdo a estos parámetros Cockburn menciona:

- **Criticidad:** este concepto hace referencia a la dimensión de las pérdidas que ocasionaría un mal funcionamiento de sistema.
- **Dimensión:** determinación del tamaño del sistema por el número de personas empleadas en su desarrollo.

Colaboración.- En la fase de colaboración de ASD se realizan las tareas propias de desarrollo: se incorpora la funcionalidad pactada en el software y se desempeña las actividades técnicas paralelas.

Aprendizaje.- La fase de aprendizaje supone la revisión de lo desarrollado así como la crítica del producto y del proceso realizado. Esta etapa puede suponer conflicto de intereses, por loa tanto las técnicas de desarrollo personal y la facilitación de la comunicación se hacen más vitales que nunca. Y el producto está libre de errores y es funcional, es evaluado bajo la visión del cliente.

Existen ciertas tecnologías que permiten que ASD pueda operar con este ciclo de vida y cumplir las expectativas.

- **Lenguaje de programación dinámica.-** Estos lenguajes son especialmente aptos para ambientes cambiantes; por eso se utilizan con frecuencia en simulaciones y modelado. Utilizando este tipo de lenguaje el desarrollador tiene la posibilidad de introducir cambios de forma rápida y efectuar una evaluación sobre las modificaciones.
- **Tecnología de agentes.**- Un agente es un elemento que realiza una tarea y lo hace de acuerdo a la elección de otro. En el desarrollo de software deseamos que los agentes realicen las tareas de forma adecuada.
- **Teoría de la decisión.**- Es un lenguaje mediante el cual podemos describir, construir y comunicar el software.
- **Aprendizaje.-** El aprendizaje mediante técnicas nos permite acercarnos al problema y luego de una exploración adecuada, brindar soluciones cada vez más inmediatas a la deseada.
- **Redes de probabilidad.-** Ofrecen la posibilidad de realizar la presentación grafica de distribuciones de probabilidad. Se pueden observar las variables y su dependencia.

DSDM (DYNAMIC SYSTEMS DEVELOOMENT METHOD)

El DSDM (método de desarrollo de sistemas dinámico) es una metodología creada por un conjunto de grandes empresas británicas a principios de los años 90 para el desarrollo rápido de aplicaciones. Actualmente posee un gran soporte y certificaciones, gracias a la difusión lograda por la autora Jennifer Stapledon. Es uno de los métodos con más tradición.

Fases de desarrollo

Es importante destacar que el método de desarrollo de sistema dinámico sostiene la necesidad de mantener el tiempo y el costo como constante y que la variable debe ser la funcionalidad. Veamos a continuación un detalle de fases:

- **Estudio de viabilidad.-** Este estudio tiene de duración semanas quizás hasta meses, las tareas que se realizan son similares a las de una metodología clásica: se definen los riesgos del proyecto y se analizan los requisitos y la visión básica del sistema por último se genera información pertinente sobre las opciones disponibles y la selección de Dynamic Systems Development Method.
- **Estudio de negocio.-** Este estudio pretende poner al equipo en contacto con los requisitos reales del cliente y el análisis de una solución. Se realiza la definición de arquitectura del sistema y el plan de prototipado.
- **Iteraciones funcional y de diseño.**- Estas dos etapas son iterativas aunque pequeñas diferencias. En primero momentos los desarrolladores trabajan sobre el modelo y lo ponen a prueba con los clientes. En la iteración de diseño es donde se desarrolla realmente.
- **Implementación.**- El sistema desarrollado en su estado actual es llevado al ambiente real, en donde se confronta con los requerimientos y las planificaciones realizadas.
- **Practicas.**- En DSDM se tiene nueve prácticas, las cuales intentan mantener en concreto su filosofía de desarrollo. Veamos cada uno de ellas.

❖ **Compromiso del usuario:** el desarrollo con las fases iteradas sobre el final requiere de un alto nivel de compromiso del usuario para poder conocer las debilidades del proyecto.
❖ **Equipo con toma de decisiones:** a pesar de que los roles sean establecidos en la toma de decisión, se espera que el equipo se organice, gestione su proceso de desarrollo, y que su resultado sea producto de sus deserciones.
❖ **Entrega frecuente.-** las iteraciones cortas permiten actuar sobre el producto de forma temprana. La ventaja de esto puede mejorar el producto antes de que el problema sea demasiado grande y su solución muy completa.
❖ **Desarrollo incremental:** los cambios y las modificaciones son agregadas al software.
❖ **Conocer el negocio:** la aceptación de un entregable debe ser siempre de acuerdo al beneficio que este le puede brindar al negocio del cliente.
❖ **Cambios reversibles:** así como incrementamos la funcionalidad, también debemos poder deshacer los cambios.
❖ **Requerimientos de alto nivel:** la definición de requisitos debe ser representada de forma tal de que el conocimiento pueda ser difundido y luego se profundice lo necesario.
❖ **Pruebas integradas:** se hacen pruebas de integración y regresión en todo el proceso.
❖ **Colaboración: se** relacionan los desarrolladores y los usuarios para poder observar los mayores beneficios.

Roles.- los roles más destacados por la metodología son:

- **Coordinador técnico:** es el responsable de la calidad del proyecto. Debe estar familiarizado y en contacto no solo con las especificaciones técnicas y funcionales sino con el desarrollo diario y sus métodos.
- **Usuario embajador:** es un usuario experto con conocimientos suficientes como para poder asumir diferentes puntos de vista y trasladar ese conocimiento al desarrollo.
- **Visionario:** conoce los objetivos y se encarga de comunicar su complimiento. El usuario embajador y el visionario pueden ser la misma persona.

- **Patrocinador ejecutivo:** es el responsable financiero del proyecto y, por lo tanto, es quien en última instancia toma todas las decisiones concernientes a él.
- **Facilitador:** es el encargado de ofrecer lo necesario a fin de mejorar e incentivar la comunicación del equipo.
- **Desarrolladores:** todos los analistas, diseñadores, DBA (Date Base Administrators), programadores, etc. son agrupados bajo esta denominación, que puede ser de dos niveles: desarrollador o desarrollador sénior.

Lean development (LD) y lean software development (LSD)

Lean es una metodología para la mejora de la calidad y la profundidad. Sus variantes más conocidas son Lean Product Development y Lean Software Development. La segunda se aplica específicamente al desarrollo de software. Su origen se debe a Mary y Tom Poppendiek, quienes presentaron las primeras ideas para aplicar Lean al desarrollo informático y crearon una serie de herramientas útiles.

Lean ha sido utilizado principalmente en empresas de manufacturas y logística, mejorando la finalización de los proyectos. El conocimiento propuesto por los autores de LSD está directamente relacionado con sus observaciones sobre distintas empresas asiáticas. El método está inspirado en la producción automovilística con JIT (Just in time), que produce solo lo necesario en el tiempo adecuado, y los controles de TQM (Total Quality Management), control de calidad total donde se ataca el error en el sitio.

Lean no pretende guiar el desarrollo desde el punto de vista técnico, sino conocer el negocio en profundidad. Tenemos doce valores de gestión de acuerdo a Jim Highsmith:

- Satisfacer al cliente es la máxima prioridad.
- Proporcionar siempre el mejor valor por la inversión.
- El éxito depende de la activa participación del cliente.
- Cada proyecto LD es un esfuerzo de equipo.
- Todo se puede cambiar.
- Soluciones de dominio, no construir.
- Una solución al 80% hoy, en vez de una al 100% mañana.
- El minimalismo es esencial.
- La necesidad termina la tecnología.
- El crecimiento del producto es el incremento de sus prestaciones, no de su tamaño.
- Nunca empujar LD más allá de sus límites.