

Introducción

Los sistemas de software son abstractos e intangibles. No están restringidos por las propiedades de los materiales, regidos por leyes físicas ni por procesos de fabricación. No existen límites naturales a su potencial. Sin embargo, debido a la falta de restricciones físicas, los sistemas de software pueden volverse rápidamente muy complejos, difíciles de entender y costosos de cambiar.

No tiene sentido buscar notaciones, métodos o técnicas universales para la ingeniería de software, ya que diferentes tipos de software requieren distintos enfoques.

Fallas del software

Se dan en consecuencia de dos factores:

- Demandas crecientes: Como las nuevas técnicas de ingeniería de software permiten desarrollar sistemas más grandes y complejos, las demandas cambian. Se necesitan nuevas técnicas para satisfacer nuevas demandas
- Expectativas bajas: Es relativamente sencillo escribir programas sin usar ingeniería de software, pero esto hace a los sistemas más costosos y menos confiables. Es necesaria una mejor capacitación para solucionar esto.

Ingeniería de Software

Se propuso en 1968 en una conferencia para discutir la crisis del software. Los enfoques individuales al desarrollo no escalaban hacia los grandes sistemas, Estos eran menos confiables, se distribuían con demora y eran más costosos.

La ingeniería de software busca apoyar el desarrollo de software profesional, en lugar de la programación individual. Incluye técnicas que apoyan la especificación, el diseño y la evolución del programa. Muchos suponen que el software es tan sólo otra palabra para los programas de cómputo. No obstante, cuando se habla de ingeniería de software, esto no sólo se refiere a los programas en sí, sino también a toda la documentación asociada y los datos de configuración requeridos para hacer que estos programas operen de manera correcta.

Para dar definiciones concretas:

- Software: Programas de cómputo y documentación asociada, pueden ser genéricos o personalizados. El buen software debe entregar al usuario la funcionalidad y el desempeño requeridos, y debe ser sustentable, confiable y utilizable.
- Ingeniería de Software: La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software. Sus actividades fundamentales son: Especificación, desarrollo, validación y evolución del software.

En cuanto a los productos de software, pueden ser:

- Genérico o enlatado: Sistemas independientes que se producen por una organización de desarrollo y se venden en el mercado abierto a cualquier cliente. La organización que lo desarrolla controla la especificación.
- Personalizado o a medida: Destinados para un cliente particular. Un contratista de software desarrolla el programa especialmente para dicho cliente. La organización que compra el software generalmente desarrolla y controla la especificación, por lo que los desarrolladores de software la siguen.

Ingeniería de Software

Es una disciplina que se interesa por todos los aspectos de la producción de software, desde las primera etapas de la especificación del software, hasta el mantenimiento del sistema después de que se pone en operación. Hay dos frases claves para esta definición:

- Disciplina de ingeniería: Los ingenieros aplican teorías, métodos y herramientas donde es adecuado. Los usan de manera selectiva e intentan encontrar soluciones a problemas, trabajando bajo restricciones organizacionales y financieras.
- Todos los aspectos de la producción del software, no es sólo técnico, sino que incluye administración de proyectos, desarrollo de herramientas y los métodos y teorías para apoyar a la producción de software.

Incluye tanto a las disciplinas técnicas, como a las de gestión y soporte.

Proceso de Software

Es el enfoque sistemático que se usa en esta disciplina. Es una secuencia estructurada de actividades, métodos, prácticas, y transformaciones que la gente usa para conducir a la elaboración de un producto de software. Varían dependiendo de la organización y el tipo de sistema, y debe ser explícitamente modelado si va a ser administrado.

En un proceso de software colaboran entre sí personas, procedimientos y métodos, y herramientas y equipos.

Hay 4 actividades fundamentales comunes a todos los procesos de software:

1. Especificación: Clientes e ingenieros definen el software que se producirá.
2. Desarrollo: Donde se diseña y programa el software.
3. Validación: Donde se verifica el software para asegurar que sea lo que el cliente quiere.
4. Evolución: Donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.

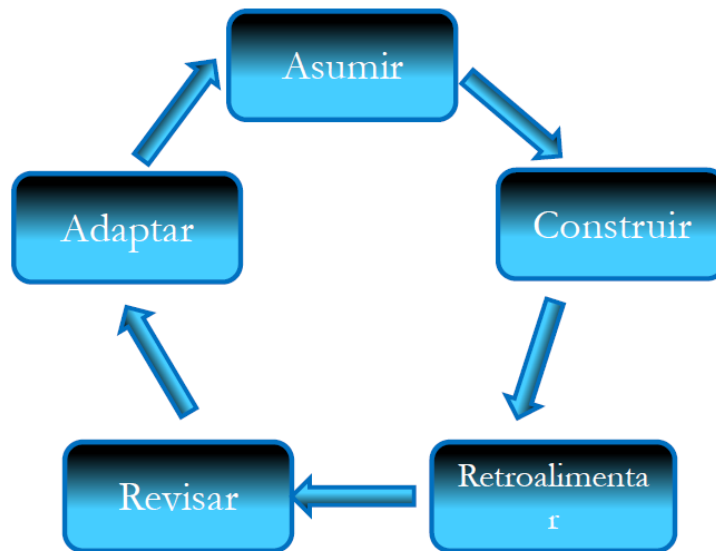
Empírico vs Definido

El primero está basado en la experiencia, tal como las metodologías ágiles, mientras que el segundo asume que el proceso se puede repetir indefinidamente y obtener los mismos resultados, en este, la administración y control provienen de la predictibilidad del proceso definido, como por ejemplo con el PUD.



En un proceso empírico, se asume que los procesos son complicados y con variables cambiantes. Aunque se repita, puede dar distintos resultados. La administración y control se realiza a través de inspecciones frecuentes y adaptaciones. Son procesos que trabajan bien cuando interviene la creatividad y complejidad.

Para implementar un proceso empírico es fundamental compartir el conocimiento:

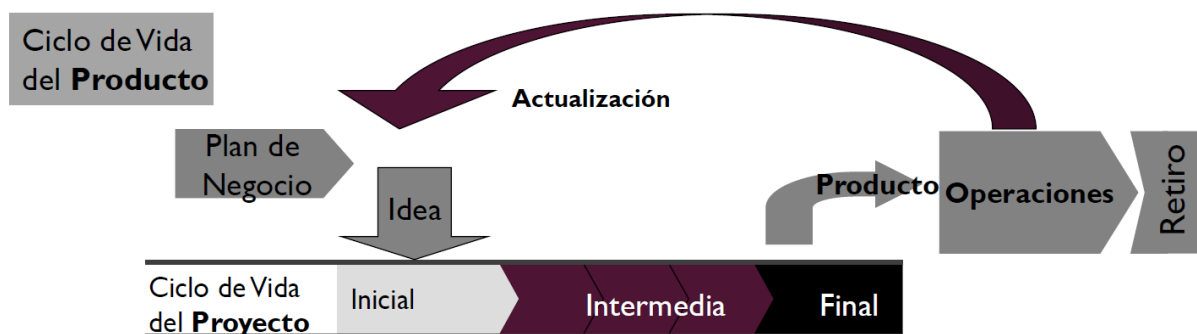


Los tres pilares que soportan toda implementación del control de los procesos empíricos son la transparencia, la inspección y la adaptación.

Ciclos de vida

Un ciclo de vida consiste en todas las actividades que deberían ocurrir entre la primera y la última versión del software, es la serie de pasos a través de los cuales el producto o proyecto progresa. Su propósito es establecer un orden en el que el proyecto se especifica, prototipa, diseña, implementa, revisa, testea y realiza sus otras actividades. Determina un criterio de si proceder o no a la actividad siguiente.

El ciclo de vida del producto suele ser mucho más largo que el del proyecto, el producto se construye a partir de sucesivos proyectos.



Ciclo de vida del Proyecto

Es una representación de un proceso, gráfica una descripción del proceso desde una perspectiva particular. Se especifican las fases del proceso (especificación, diseño, etc) y el orden con el que se llevan a cabo.

Hay 3 tipos básicos:

Secuencial

El proceso progresa a través de una secuencia ordenada de pasos desde el concepto inicial hasta el testing. Al final de cada fase, se hace una revisión para determinar si el proyecto puede avanzar. En un secuencial o cascada puro, no hay solapamiento entre fases, y el principal artefacto es la documentación.

Tiene beneficios cuando tenemos una definición del producto estable, ayudando a encontrar errores en las etapas más tempranas y de menor costo. Puede ser útil cuando por ejemplo tenemos que hacer un porteo de un producto.

Como contraparte, no entrega incrementos, la única versión usable del producto es la versión final, al final del ciclo de vida, además de que tiene el problema de que al desarrollar un nuevo producto los requerimientos no están claros desde el principio. En este ciclo de vida la especificación debe estar completa antes de seguir a otras fases.

Iterativo/incremental

Desagrega al proyecto en “miniproyectos”. Cada miniproyecto se encarga de uno o más de los grandes riesgos, hasta que todos estos sean atendidos.

El producto se desarrolla como una serie de incrementos donde cada uno añade funcionalidades a la versión anterior, entrelazando la especificación, desarrollo y validación. Esto lo vuelve muy útil para sistemas de requerimientos volátiles. Es más fácil y menos costoso realizar cambios, y además al producir incrementos funcionales para el cliente, reducen la ansiedad del mismo y nos ayudan a entender mejor los requerimientos. Como desventajas, los incrementos progresivos tienden a degradar la estructura del sistema.

Recursoivo

Como el espiral. A diferencia del anterior, se inicia con algo en forma completa, y se presenta un prototipo que se va mejorando con cada vuelta refinando las funcionalidades sucesivamente. El producto funcional 100% lo tendremos solo al final del proyecto. Se generan productos independientes de la implementación que pueden ser usados en sistemas de características similares, pero puede ser costoso readaptarlos.

Elegir un Ciclo de Vida

La elección de ciclo de vida depende de varios factores, incluyendo riesgos técnicos y administrativos, la volatilidad de los requerimientos, el tamaño del equipo o los aspectos del cliente. Sin embargo, se suele recomendar el ciclo de vida iterativo e incremental salvo en casos específicos. Esto es porque es difícil tener una especificación 100% completa de los requerimientos antes de comenzar a desarrollar o entregar el producto o versiones de este, además de que los requerimientos suelen ser volátiles. Como nota, cabe aclarar que cualquier ciclo de vida puede funcionar mal si se implementa incorrectamente.

El Proyecto en Sistemas de Información



Este gráfico representa las relaciones entre los componentes más importantes. Las personas se incorporan a un proyecto en roles que asumen responsabilidades. El proyecto se adapta a un proceso (que puede automatizarse con herramientas), y que produce como resultado un producto.

Proyecto

Un proyecto es una unidad de organización del trabajo, que nos guía en cómo asignamos y distribuimos recursos y cómo realizamos el paso a paso para llegar al objetivo. El objetivo del proyecto es generar un producto que le aporte valor al cliente. Es una parte esencial de la ingeniería de software ya que se necesita administrar los proyectos porque la disciplina en sí está sujeta siempre a las restricciones organizacionales de presupuesto y fecha. Una buena gestión de proyecto no asegura el éxito del mismo, pero una mala gestión da generalmente un proyecto fallido.

Las características del proyecto son:

- Orientado a objetivos: Los proyectos se dirigen a obtener resultados y esto se refleja a través de los objetivos. Los objetivos guían al proyecto, por lo que no pueden ser ambiguos, deben ser claros y viables.
- Duración limitada: Un proyecto tiene un inicio y un fin, cuando se alcanza el objetivo, el proyecto termina.
- Son únicos e irrepetibles, por más que puedan tener características similares con otros, son únicos.
- Elaboración gradual: El proceso nos da pautas a seguir para alcanzar el objetivo, el proyecto es un medio para alcanzar el producto.
- Tareas interrelacionadas basadas en esfuerzos y recursos: Aquí viene la complejidad sistémica, a las tareas se le deben asignar recursos y esfuerzos, los cuales son limitados
- Focalizados en un plan: La planeación es todo, el plan es nada. Planeando aprendemos, el documento final no es lo más valioso.

Administración de proyectos

Es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para satisfacer los requerimientos del proyecto. Básicamente, administrar un proyecto incluye:

- Identificar los requerimientos
- Establecer objetivos claros y alcanzables
- Adaptar las especificaciones, planes y el enfoque a los diferentes intereses de los involucrados (stakeholders)

Implica estimar el uso de recursos y personas para entregar un producto o servicio según plazos y presupuestos acordados.

La restricción triple

El balance de los siguientes tres factores afecta directamente la calidad del proyecto, los proyectos de alta calidad entregan el producto requerido, en el tiempo y presupuesto acordado. Estos tres factores están bajo el control del líder de proyecto:

- Objetivos del proyecto
- Tiempo para completarlo
- Cuánto debería costar

El único de estos factores que no podemos controlar es el objetivo del proyecto, este está totalmente bajo el control del cliente, los otros dos factores son los que podemos negociar. Esto sucede porque el cliente fija el alcance y luego de este se derivan los costos y tiempo necesarios para alcanzarlos.

Si modificamos uno de los tres factores, hay que modificar los otros para mantener el equilibrio. Es importante que el líder los balancee, ya que si no es el caso no se tendrá mucho margen de acción y lo primero que se sacrificara será la calidad del producto. Este es el punto central, la calidad no es una variable, por lo que debemos sacrificar alguno de los 3 factores pero nunca la calidad. Recordemos, calidad no es solo cumplir requerimientos, sino también comportamiento, estructura y organización, documentación, etc.

El desarrollo de Software

Cuando desarrollamos un producto de software deberíamos desarrollar cada nueva versión incrementalmente respecto a la anterior a través de una serie de pasos.

El líder de proyecto es la cara visible del equipo, y es el que va a manejar la relación con los interesados, además de cargar con la responsabilidad del trabajo en equipo ante niveles altos de administración de la organización y sus generaciones.

Por otro lado distinguimos al equipo de proyecto. Este es un grupo de personas comprometidas en alcanzar un conjunto de objetivos de los cuales se sienten mutuamente responsables. Debe ser un equipo con una diversidad de conocimientos y habilidades, con la posibilidad de trabajar juntos efectivamente o desarrollar sinergia. Usualmente es un grupo pequeño y tiene sentido de responsabilidad como una unidad.

Artefactos del proyecto de software

Plan de Proyecto

También conocido como plan de desarrollo de software. Es lo que es una hoja de ruta para un viaje. Se crea al comienzo del proyecto y se usa para comunicar al equipo y a los clientes cómo se realizará el trabajo, para poder valorar el avance del proyecto.

En él se documenta:

- Qué hacer
- Cuando hacer
- Cómo hacerlo
- Quién lo va a hacer

Como diferencia, tenemos por un lado al plan de proyecto y a la ERS. El plan de proyecto es un artefacto del proyecto, mientras que la ERS es un artefacto del producto, ya que define objetivos y alcances del producto, junto con ciertos criterios de aceptación. Hay una íntima relación entre ellos, pero un cambio en uno no implica un cambio en el otro.

La planeación se presenta durante 3 etapas de un ciclo de vida del proyecto

- En la etapa de propuestas, cuando se presenta una licitación con vistas a obtener un contrato para desarrollar o proporcionar un sistema de software. En esta etapa se necesita un plan para ayudar a decidir si se cuenta con los recursos para realizar el trabajo, y poder cotizar el precio para el cliente. Es especulativa.
- En la fase de inicio, donde se determina quién va a trabajar en el proyecto, como dividirlo en incrementos, cómo asignar recursos, etc. Aquí se cuenta con más información que en la anterior etapa, por lo que se pueden afinar las estimaciones.
- Periódicamente a lo largo del proyecto, el plan se modifica a la luz de la experiencia obtenida y la información del monitoreo del avance del trabajo. Se aprende más acerca del sistema y de las capacidades del equipo. Se aumenta aún más la precisión de las estimaciones. También es posible que los requerimientos cambien, lo que hace necesario que el proyecto cambie extendiendo plazos, modificando la división de trabajo, etc.

A medida que avanzamos en el ciclo de vida del proyecto, mejoramos sucesivamente el plan del proyecto.

Planear en la metodología tradicional implica varias tareas:

Definir alcance del proyecto

Es todo el trabajo y solo el trabajo que debe hacerse para entregar el producto o servicio con todas las características y funciones especificadas. El cumplimiento se mida contra el plan de proyecto

Definición de proceso y ciclo de vida

Adaptar el proceso al proyecto y elegir el ciclo de vida. Determina el orden de las tareas que se hacen en el proceso, tiempo de cada tarea. Generalmente se recomienda un ciclo de vida iterativo e incremental.

Estimación

No debe implicar compromiso, sino una aproximación anticipada de un valor asociado al proyecto o producto. Existe un orden para determinarlos:

1. **Tamaño:** Antes la métrica era líneas de código, aunque hoy en día es una métrica inútil. Con el POO surgió como métrica la cantidad de clase, métodos, puntos de función o puntos casos de uso. Hoy en día se estima con requerimientos, ya que los tenemos antes que el código en sí mismo, como los story points.
2. **Esfuerzo:** Se estima a partir del tamaño, se estima el esfuerzo ya que obtenemos una "bolsa" de horas necesarias para construir el producto con el tamaño estimado. Se mide en horas persona lineales, es decir, una sola persona haciendo una cosa a la vez.
3. **Calendario:** Descomponemos el proyecto en tareas detalladas a realizar e indicamos cuándo hacer cada una, indicando aproximadamente cuando terminan y cuando inician, quien las hará, etc. También se hace una descomposición del esfuerzo o estructura de descomposición del trabajo WBS.
4. **Costo:** Se realiza luego de estimar el calendario ya que depende de este, por ejemplo, si el tiempo es ajustado y debemos pagar horas extra. El 80% de los costos surgen del esfuerzo y hora de trabajo, aunque también hay costos de software, hardware, capacitación, viáticos, etc.

Gestión de riesgos

Es especialmente importante en los proyectos de software, debido a la incertidumbre inherente a la que se enfrentan la mayoría de proyectos, principalmente relacionados a la falta de especificación de requerimientos, su volatilidad, dificultades de estimación de tamaño, costo, capacidades del equipo, etc.

Es importante identificar riesgos, anticiparse, comprender su impacto y crear un plan de contingencia.

Un riesgo es la probabilidad de ocurrencia de una pérdida o daño que afecte al proyecto ya sea a su calendario o a su calidad, y también debe estimarse. Si la probabilidad es 1, no es un riesgo, es un problema que debe gestionarse. El riesgo se mide en exposición al riesgo, que está dado por la probabilidad de ocurrencia multiplicado por el impacto que ocasiona si sucede. De todas formas, no suelen usarse métricas tan exactas, sino más bien generales como impacto "bajo", "medio" o "alto", debido a que no es sencillo

realizar un juicio sobre la probabilidad e impacto del riesgo, basado principalmente en la experiencia previa.

Podemos tomar distintas actitudes frente al riesgo:

- Negación
- Actitud reactiva: pensamos como solucionar el problema cuando el riesgo se manifieste
- Actitud proactiva: disminuimos la probabilidad de ocurrencia y cómo tratarlo si se manifiesta. Es lo mas optimo pero también apareja costos.

En un proyecto de software, los riesgos se pueden clasificar en el siguiente orden:

1. Volatilidad de requerimientos
2. Personas: pueden abandonar el proyecto, no contar no conocimientos necesarios, etc
3. Tecnologías, especialmente si son nuevas y no tenemos el conocimiento suficiente
4. Estimación
5. Organización
6. Herramientas

Una técnica es seleccionar el top 10 de los riesgos con más exposición (10 es arbitrario, podemos elegir el número máximo de riesgos que podamos manejar y monitorizar) e intentar mitigarlos, evitarlos, establecer planes de contingencia. Esto se hace debido a que nunca podremos evitar ni identificar todos los riesgos. Esta gestión de riesgos se hace a través de todo el ciclo de vida del proyecto. También existen riesgos de producto, empresariales, y hay riesgos que pueden pertenecer a más de una categoría.

El proceso de gestión de riesgos implica identificación, análisis, planeación y monitorización, documentando todo en un plan de gestión de riesgo.

Asignación de recursos

Se establece una división del proyecto en actividades, identificando plazos y entregas asociados con cada actividad, y se asignan los recursos necesarios para satisfacerlo.

Programación de Proyectos

Se indican las dependencias entre actividades, tiempo estimado requerido para alcanzar cada plazo, asignación de personal y recursos para cada actividad y cómo deben realizarse. Se pueden usar herramientas como Microsoft Project o diagramas de Gantt. Vamos a tener un calendario inicial que sirve para planear cómo asignar las personas al proyecto y comprobar sus avances, pero esto se modifica a lo largo del avance del proyecto. Cada tarea debe durar al menos una semana, pero si dura más de dos meses se subdividen. Una técnica es estimar que nada va a salir mal, y luego ampliar la estimación para enfrentar problemas anticipados, o añadir la estimación de un factor de contingencia para problemas no anticipados.

Definición de controles

Se define cada cuanto como y con quienes se hacen reuniones va comparar el avance del proyecto contra su planificación. Aquí toman importancia las métricas. Se definen qué variables se van a monitorear y los mecanismos para llevar un control del proyecto.

Definición de métricas

Una métrica es un número para dar visibilidad sobre el proyecto, producto o proceso.

Las métricas del proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software. Las métricas básicas para un proyecto pueden ser referidas al tamaño del producto, el esfuerzo, el tiempo o defectos.

Al planificar se deben elegir métricas para realizar el monitoreo y control. Es importante que sean simples, y solo elegir métricas más complejas si nos dan más información de la que tenemos ahora, que sea de beneficio práctico, significativa y que nos diga lo que queremos saber. No tiene sentido medir la distancia a Mendoza en milímetros si podemos usar kilómetros.

Planes de Soporte

Planes de mantenimiento, calidad, capacitación, iteración, subcontratación, etc.

Complementan al plan del proyecto.

De esta forma, los tres principales factores para el éxito de un proyecto son:

- Tener un objetivo o misión claro
- Lograr una buena comunicación (tanto interna con el propio equipo como externa con los stakeholders)
- Monitoreo y feedback

Las principales causas del fracaso en proyectos son:

- Fallas al definir el problema
- Planificar basado en datos insuficientes
- No hay seguimiento del plan de proyecto
- Plan de proyecto pobre en detalles
- Planificación de recursos inadecuada
- Las estimaciones se basaron en supuestos sin consultar datos históricos
- Nadie estaba a cargo