

Introduccion

Objetivos de un proyecto software:

- Gestion del proyecto: cumplir con esfuerzo, tiempo y coste.
- Producto: alta calidad. Se deben establecer por anticipado **parametros cuantificables y verificables** que midan la calidad desde un punto de vista

No basta con las pruebas se detectan menos de la mitad de los errores (100 defectos por cada 1000 lineas) y estas a veces se omiten.

Para obtener un producto de calidad hay que trabajar desde el principio y a lo largo de todo el proceso de desarrollo software. Las pruebas con un complemento que mejoran la calidad.

Aspectos implicados en el desarrollo software:

- Producto: se centra en el software (metricas, metodologias...)
- Proceso: un buen proceso implica un buen producto (CMM, ISO 9000-3...)
- Personas: mejorar la capacidad mediante formacion, etc.
- Problema: entender y conceptualizar adecuadamente el problema considerado.

Para mejorar el proceso se debe aumentar el control durante el proceso (desarrollo y mantenimiento). La solución no es trivial:

- el 20% del esfuerzo se establece a la hora de dar soporte tecnologico.
- el 80% restante es necesario para hacer los cambios organizativos.
- ademas solo el 40% de proyectos de mejora tienen éxito.

CALIDAD

definicion: conjunto de propiedades y caracteristicas de un producto o servicio que satisfacen unas necesidades explicitas o implicitas.

Funcion organizativa:

- Staff directamente dependiente exclusivamente de Direccion
- Aporta una vision independiente de la empresa o proyecto.
- Su funcion es velar por el buen hacer y si hay desviaciones comunicarlas a Direccion
- Actividades que realiza:
 - Actividades de aseguramiento de calidad: permiten asegurar que se sigue el proceso definido (por ejemplo, auditoria a fin de fase)
 - Actividades de control de calidad: controlan la calidad tecnica de los productos entregables (**software y documentos**).

Gestion de la calidad abarca:

- Un proceso de garantia de la calidad
- Tareas especificas para el aseguramiento y control de la calidad
- Practicas efectivas de ingenieria del soft (metodos y herramientas)
- Control de todos los productos de trabajo y los cambios que generan (**gestion de la configuracion**)
- Procedimiento para garantizar el ajuste a los estandares de desarrollo soft.
- Mecanismos de medicion y generacion de informes.

Verificación y validación del software

-Verificacion: actividades que garantizan que el resultados de cada fase se corresponde con las entradas en dicha fase. Se realiza para cada fase del ciclo de vida confirmando que lo realizado

hasta el momento es correcto, completo y consistente.

-Validación: actividades que aseguran que el software construido corresponde con los requisitos del cliente (**Pruebas de aceptación**).

Problemas que afectan a la calidad del software:

-Con el cliente:

- no participa en la definición ni desarrollo del proyecto
- pide constantemente cambios

-Con el proyecto:

- se define vaga o incorrectamente
- inexistencia de un ERS
- los requisitos cambian continuamente y no se hace **gestion de requisitos**.

-Con el desarrollo:

- cada miembro trabaja a su manera, sin que se pueda entender o reutilizar nada
- no se dispone de las versiones correctas de los entregables o no hay control sobre ellos.

-Si hay prisa, pruebas y documentación se omiten.

Modelos de calidad del software

-CMM: desarrollado por el SEI con la iniciativa de la administración pública de EEUU. Es marco de referencia para el proceso de desarrollo software.

-Bootstrap: respuesta europea al CMM. Se encuentra alineado con la norma ISO 9000.

-ISO 9000: norma internacional de calidad genérica, que posee guías de aplicación a las empresas desarrolladoras de software.

-SPICE: objetivo de materializar un nuevo estándar para la valoración del proceso software a nivel internacional. Fue apoyado por las iniciativas existentes y promovido por ISO.

Vision general de aplicación en empresa:

-CMM principalmente en EEUU

-ISO en Europa, pero con problemas de adaptación (por ser ISO genérica). Por este motivo se crearon guías de adaptación al ámbito del software. La estrategia habitual es conseguir el certificado ISO 9000 y seguir posteriormente con CMM.

CMM

Guía a las organizaciones indicando cómo mejorar los procesos asociados al desarrollo y mantenimiento del software. Para ello propone diferentes niveles de madurez, en donde cada etapa comprende un conjunto de objetivos a alcanzar. Una vez satisfechos se logra un mayor nivel de capacidad.

Niveles:

- Inicial:
 - El proceso software ad-hoc y el éxito depende de los esfuerzos individuales.
 - No se proporciona un entorno estable para desarrollar y mantener software.
 - Se planifica el trabajo por las necesidades y compromisos según van surgiendo.
 - La efectividad descansa en la experiencia del equipo.
- Repetible:
 - Existen procesos básicos de gestión de proyectos que permiten el seguimiento de costes, planificación y funcionalidad.
 - La **disciplina del proceso** consiste en la experiencia acumulada en éxitos anteriores en proyectos similares.
 - La experiencia es documentada, medida e incluso mejorada y el equipo es entrenado en ella.
 - Proyectos con requisitos muy diferentes a los ya abordados implican riesgo para la empresa.
 - Entorno disciplinado y proceso software bajo el control efectivo de un sistema de gestión guiado por planes realistas basados en el rendimiento en proyectos anteriores (históricos).
- Definido:
 - El proceso está documentado, **estandarizado** e integrado.
 - Uso de prácticas propias de ingeniería del software, los proyectos siguen estándares establecidos y se conoce la situación y progreso de los mismos, las versiones son controladas y su puesta en operación es verificada y aprobada adecuadamente.
 - El proceso es el apoyo básico de gestores y es la guía del resto del personal.
 - Es posible controlar el progreso de los proyectos y actuar en consecuencia.
 - Gestores e ingenieros se rigen por procedimientos repetibles y estables.
- Gestionado:
 - La organización se caracteriza por su capacidad para medir atributos del software, tanto del producto y su calidad como del proceso.
 - Es necesario establecer objetivos cuantitativos y conocer su grado de cumplimiento.
 - La dirección se apoya e involucra en el programa aportando los recursos necesarios.
 - Este conocimiento cuantitativo permite que el proceso sea **predecible** y se puedan tomar acciones correctivas.
 - El proceso de medida es una actividad en la que se sustenta la dirección y la toma de decisiones de los gestores.
- Optimizado:
 - Se incorporan nuevas tecnologías e ideas innovadoras en un proceso de **mejora continua**, que se valoran cuantitativamente en términos de costes y efectos a la organización.
 - La organización conoce los puntos fuertes y débiles, lo que permitirá la prevención de defectos. Además analiza los defectos que se puedan producir y determina las causas.

IDEAL:

Marco de mejoras basado en CMM que establece los pasos necesarios para llevar a cabo un programa de mejora desde un punto de vista ingenieril y disciplinado. Está compuesto por 5 fases:

- **Iniciación:** se aclaran los objetivos que se beneficiarán por el esfuerzo que supone el proceso de mejora.
- **Diagnóstico:** evaluar mediante un método formal las fortalezas y debilidades del proceso seguido
- **Establecimiento:** realizar una planificación detallada específica de las mejoras que se desea alcanzar. Se establece la estrategia y las prioridades.
- **Actuación:** implementar la mejora llevando a cabo el plan de acción.
- **Aprendizaje:** aprender de la experiencia y aumentar la habilidad para mejorar los procesos de forma continua.

ISO:

Describe un sistema de gestión de calidad en términos genéricos. Un sistema de gestión de calidad se define como la estructura organizativa, responsabilidades, procedimientos, procesos y recursos para implementar la gestión de la calidad con el objetivo de garantizar que los productos y servicios ofrecidos por la empresa satisfagan las expectativas de los clientes al cumplir sus especificaciones.

UNE-EN ISO 9000: fundamentos y vocabulario de un SGC.

UNE-EN ISO 9001: especifica los requisitos para la implantación de un SGC. Se centra en la eficacia.

UNE-EN ISO 9004: directrices para la mejora continua centrandose en la eficiencia (minimizar recursos) y ventaja competitiva (cuota de mercado).

La familia ISO 9000 se basa en 8 **principios**:

- **Enfoque al cliente:** la empresa debe esforzarse en satisfacer los requisitos y esforzarse en exceder sus expectativas.
- **Liderazgo:** los líderes deben crear y mantener un ambiente interno en el que los empleados puedan involucrarse totalmente en el logro de los objetivos de la empresa.
- **Participación del personal:** buscar el compromiso total de estos con la empresa
- **Enfoque basado en procesos:** proporciona el control continuo sobre los vínculos entre los procesos individuales, así como sobre su combinación e interacción para el seguimiento de la satisfacción del cliente.
- **Enfoque de sistema para la gestión:** Identificar, entender y gestionar los procesos interrelacionados como un todo contribuye a la eficacia y la eficiencia (se pueden elaborar mapas de procesos). Para cada proceso hay que definir:
 - Entradas y salidas.
 - Responsable
 - Requisitos del cliente (externo o interno)
 - Evaluación de conformidad sobre los requisitos
 - Medidas de control del proceso
 - Oportunidad de mejorar el proceso y priorizar dicha oportunidad.
- **Mejora continua:** se sigue el ciclo de Deming (planificar->hacer->chequear->actuar)
- **Enfoque basado en hechos para la toma de decisiones:** Las decisiones eficaces se basan en el análisis de los datos y la información real.
- **Relaciones mutuamente beneficiosas con el proveedor:** establecer condiciones de aceptación, de esta manera nos aseguramos de que los productos que nos proveen son de calidad.

ISO 9001 especifica los requisitos para un SGC de forma prescriptiva, no descriptiva (indica que tienes que hacer pero no el como). Los requisitos se estructuran en los apartados del 4 al 8, siendo los primeros de introducción.

- **Apartado 4. Sistema de gestión de calidad:**
 - La organización debe realizar un mapa de procesos (identificar, secuencia e interacción).
 - Determinar criterios y metodos de evaluacion de ejecucion y control procesos para asegurar de que son eficaces.
 - Asegurarse de la disponibilidad de recursos e información para realizar el seguimiento, medición y análisis de los procesos.
 - Implementar acciones correctivas y preventivas para alcanzar los resultados planificados.
 - La documentación del SGC debe incluir declaraciones escritas de la política y de los objetivos de la calidad. Debe haber un **manual de calidad** donde se documenten los procedimientos y **registros de calidad** que requiere esta norma.
- **Aparado 5. Responsabilidad de la dirección:**
 - Establecer y mejorar continuamente el SGC.
 - Asegurar que se determinan los requisitos y se cumplen para mejorar su satisfacción (pruebas de aceptación).
 - Establecer y comunicar una política de calidad.
 - Establecer los objetivos de la calidad y hacer un planificación del SGC para abordar todos los requisitos del apartado anterior.
 - Responsabilidad, autoridad y comunicación: definición y comunicación de las responsabilidades y autoridades. Es necesario establecer mecanismos de comunicación y de que la eficacia del SGC se comunica a todos los empleados.
 - Revisión: investigar las deficiencias o posibles cambios en el SGC para iniciar acciones correctivas y preventivas. (si un empleado ve un defecto en el SGC tiene la obligación de notificarlo).
- **Apartado 6. Gestión de los recursos:**
 - La organización debe determinar y proporcionar los recursos necesarios para implantar, mantener y mejorar el SGC y de esta forma incrementar la satisfacción del cliente.
 - Los empleados cuyo trabajo afecte directa o indirectamente a la calidad del producto deben poseer el nivel de competencia adecuado. Para ello deben identificarse las competencias necesarias y abordarlas por medio de la formación adecuada evaluando el efecto (si se han aprendido los contenidos y en que medida). Se deben mantener registros de la competencia de los empleados (curriculum).
 - La organización debe determinar, proporcionar y mantener la infraestructura necesaria para satisfacer los requisitos del producto (en nuestro caso hardware, software...).
 - Debe determinar y gestionar el ambiente de trabajo (físico, ambiental, social...).
- **Apartado 7. Realización del producto:**
 - Permite algunas exclusiones, estas no pueden disminuir la capacidad de satisfacer los requisitos aplicables o impedir cumplir las responsabilidades de forma adecuada.
 - Planificación de la realización del producto: se debe planificar y desarrollar los procesos necesarios para la realización del producto. Para ello hay que determinar:
 - los objetivos de la calidad y los requisitos para el producto.
 - Procesos, documentos y recursos necesarios
 - actividades de verificación, validación, revisión y comprobación.
 - Registros que proporcionen evidencia de que el producto cumple los requisitos.
 - Procesos relacionados con el cliente: se debe determinar:
 - Requisitos especificados por el cliente, incluyendo los requisitos para las actividades de entrega y posteriores a la misma.
 - Requisitos no especificados por el cliente pero esenciales.

- Requisitos legales y reglamentarios.
- Requisitos adicionales determinados por la organización.
Además debe revisar los requisitos para asegurarse de que puede hacerlo antes de hacerlo. También se debe implantar mecanismos eficaces para comunicarse con el cliente.
- Diseño y desarrollo: aborda la planificación y el control del producto concreto con los procesos establecidos.
 - Deben planificarse y controlarse el diseño y desarrollo del producto mediante revisiones, verificaciones y validaciones manteniendo los registros correspondientes.
 - Se deben identificar los cambios mediante el control de la configuración dentro de la gestión de la configuración software GCS.
- Compras: Tienen que afectar al productos que realizamos.
 - Se deben identificar los requisitos que queremos para darselos al proveedor antes de la compra.
 - Se debe definir mecanismo de verificación para garantizar que se satisfacen dichos requisitos.
 - Se debe evaluar y seleccionar a sus proveedores para satisfacer los requisitos de la organización.
 - El tipo y el alcance de control ejercido sobre el proyecto y el producto deben ser acordes con el impacto en el producto.
- Producción y prestación del servicio: se deben planificar y abordar en condiciones controladas. En software engloba:
 - Actividades de liberación: desarrollar el software, confeccionar el paquete que se vaya a liberar y duplicar la copia original del software para su entrega al cliente.
 - Actividades de entrega e instalación del producto.
 - Actividades de mantenimiento y soporte técnico.
Debe mantenerse la trazabilidad del producto (mediante la GCS). Se debe también preservar el producto (identificación, manipulación, embalaje, almacenamiento y protección).
- Control de los dispositivos de seguimiento y medición: se debe identificar las actividades de seguimiento y medición para verificar la conformidad del producto con los requisitos. En nuestro caso los dispositivos comprenden a los equipos informáticos, herramientas software, datos derivados de comprobaciones y equipos auxiliares necesarios para interactuar con el hardware.
- **Medición, análisis y mejora**: la organización debe planificar e implantar los procesos de seguimiento, medición, análisis y mejora. La organización debe:
 - Medir la satisfacción del cliente en lo que respecta al cumplimiento de los requisitos.
 - Realizar auditorías internas.
 - Realizar seguimiento de los procesos del SGC para verificar que cumplen con los objetivos de calidad.
 - Realizar seguimiento y medición de las características del producto para verificar que se han cumplido los requisitos.
 - Identificar, recopilar y analizar los datos de seguimiento y medición apropiados para demostrar la idoneidad y eficacia del SGC e identificar oportunidades de mejora.
 - Mejorar de forma continua de la eficacia de su SGC por medio de su política de calidad, objetivos de calidad, resultados de las auditorías, análisis de los datos de las mediciones, acciones correctivas y preventivas.

ISO 90003

Norma UNE-ISO 90003:2005 (“Ingeniería del software. Guía de aplicación de la ISO 9001:2000 al software”). No añade ni cambia requisitos, solo los interpreta a nuestro caso debido a que la norma ISO 9001 es muy general.

Certificación ISO 9001

Certificar una empresa indica que está organizada y sigue un proceso que, con alta probabilidad, dará lugar a un producto de calidad. Para obtener el certificado requiere que el sistema y las operaciones sean auditados por auditores externos que verifican la concordancia con el estándar.

El proceso de certificación se puede dividir en 3 fases:

- Auditoría de evaluación previa: es opcional y es anterior a la auditoría de certificación para dar tiempo a implantar acciones correctivas y preventivas. Permite determinar el nivel de preparación, mitigar riesgos en áreas que la empresa teme no cumplir los requisitos y hacer un simulacro de auditoría de certificación.
- Auditoría de certificación: es completa. Los auditores externos examinan si las actividades de calidad son conformes con los requisitos de la ISO 9001 y los requisitos internos de la empresa (recogidos en el SGC). Los pasos básicos son:
 - Plan de auditoría de certificación.
 - Auditoría de la documentación.
 - Reunión inaugural
 - Realización de la auditoría.
 - Reunión de clausura.
 - Informe de auditoría.
 - Promocionar la certificación.
- Auditorías de seguimiento: se establecen para continuar demostrando que se siguen cumpliendo los requisitos iniciales. Cada auditoría de seguimiento conlleva la evaluación de una parte del SGC, que en conjunto auditarán el sistema completo. Hay determinados apartados clave que siempre se revisan en cada auditoría de seguimiento. Además se elaborará un informe de auditoría para cada auditoría de seguimiento.

Entidades involucradas:

- ENAC: acredita a las entidades que tengan competencia técnica para que puedan certificar a las empresas.
- AENOR: entidad que se dedica al desarrollo de la normalización y certificación.
- Entidades certificadoras: son aquellas que están acreditadas por ENAC para que puedan certificar a otras empresas. Ejemplos:
 - Lloyd's Register Quality Assurance
 - Bureau Veritas Quality International

Pasos básicos para la obtención del certificado ISO9001:

- Establecer un plan de trabajo.
- Iniciar el proyecto de mejora con recursos, conocimientos, formación y voluntad de toda la empresa.
- Establecer el manual de calidad.
- Establecer el manual de procesos.
- Implantar el sistema de calidad aplicando lo recogido en el SGC establecido progresivamente a través de proyectos piloto.
- Recoger datos e información para mejorar el sistema.
- Mejora el sistema en base al punto anterior.
- Obtener la certificación.
- Hacer partícipe a toda la empresa del reconocimiento obtenido
- Seguir trabajando para mejorar continuamente.

Ciclos de vida

Un proyecto software es una actividad que transforma unos requisitos de un cliente/usuario en un producto software entregable. Esta actividad es una secuencia bien definida de eventos, con un principio y un fin temporales, dirigidos a alcanzar unos objetivos que no son rutinarios. Presentan características especiales respecto a otros proyectos:

- Invisibilidad: el software es un producto lógico, no tangible. Es difícil de medir y estimar.
- Complejidad: el software posee mayor complejidad que cualquier otro producto de ingeniería. Esto queda patente ya que el objetivo de las pruebas no busca que este libre de errores, sino que el objetivo es encontrar errores.
- Flexibilidad: se cree que el software es fácil de cambiar y que por ellos los requisitos pueden cambiar sin problemas, en otras ingenierías no es así
- Juventud: Se cree que carece de procesos y estrategias de desarrollo variadas, probadas y definidas.

Actualmente en cualquier ciclo de vida se identifican tres macro-fases genéricas:

- Definición: pensar, se centra en el qué vamos a hacer.
- Desarrollo: hacer, se centra en el cómo lo vamos a hacer.
- Mantenimiento: se centra en el cambio. Este puede ser
 - correctivo
 - preventivo
 - perfectivo
 - adaptativo.

Modelos tradicionales

Cascada (años 70)

Es el más conocido y pone especial énfasis en la definición de requisitos y de documentación de análisis y diseño como paso previo a la codificación. Es lineal y en secuencia, es decir, cuando se acaba completamente una fase se pasa a la siguiente. Se pueden considerar bucles de retroalimentación.

Sigue un paradigma ETVX (entry, task, validation, exit).

Debido a su profuso uso se han desarrollado versiones distintas, por ejemplo dividir la tarea de diseño y programación entre varios equipos y luego pasar por una fase de integración. Para ello el software después del análisis de arquitectura debería ser modular, donde cada equipo se ocuparía de un módulo.

Ventajas:

- Marco de referencia para asignar actividades de desarrollo de software.
- Muy estructurado con pautas de trabajo muy claras.
- Facilita la coordinación de recursos.
- Facilita la disposición de hitos de seguimiento/control.
- Facilita la estimación y seguimiento del progreso.
- Facilita la detección de desviaciones y la realización de acciones correctivas.
- Proporciona productos entregables intermedios que conforman el producto final.

Inconvenientes:

- Comienza estableciendo **todos** los requisitos y muchas veces no es viable.
- No soporta prácticas modernas de desarrollo (prototipado)
- Los únicos productos aprovechables están en forma de documentos, un cambio debido a un error supone un gran coste.

Modelo en V

Similar al de cascada, pero tiene 2 tipos de tareas:

- Una rama descendente: actividades de desarrollo. Cada fase parte de las salidas de la fase previa y las verifica. Al finalizar la fase se **preparan** las pruebas que van a permitir validar lo realizado y producen productos que sirven de entrada a la siguiente fase.
- Una rama ascendente: actividades de prueba. Cada fase ejecuta las pruebas que han sido preparadas para validarlas, además verifica las actividades que se han realizado durante la fase.

Ventajas:

- Hereda todas las ventajas de Cascada.
- Establece un marco de referencia en cuanto a actividades de validación y verificación, considerando las pruebas lo antes posible.

Desventajas:

- Se comienza estableciendo todos los requisitos del sistema.
- No soporta prototipado.
- Posee gran rigidez y es utópicamente lineal.
- Los únicos productos parciales están en forma de documentos.

Prototipado

Proceso que facilita al desarrollador la creación de un modelo del software a desarrollar. Es muy adecuado cuando no se sabe exactamente lo que se quiere construir. Establece un ciclo repetitivo de 3 fases :

- Escuchar al usuario/cliente: recolección de requisitos.
- Construir y revisar el prototipo: se realiza un desarrollo rápido centrado exclusivamente en los aspectos del software visibles para el usuario empleando lenguajes de 4ª generación.
- El usuario prueba el prototipo: permite refinar los requisitos del software.

Idealmente se debe desechar y empezar de nuevo, solo debe servir para identificar los requisitos software cuando no estén claros. Con los lenguajes de 4ª generación se podrían aprovechar las pantallas y dinámica de las mismas adecuándose a los estándares y normas seguidos en un producto no prototipo.

Ventajas:

- Mecanismo ideal para extraer requisitos cuando no están claros.

Inconvenientes:

- El desarrollador toma necesariamente decisiones de implementación para desarrollar el prototipo de forma rápida, el problema es que a menudo se mantienen en el sistema real y pueden ser no correctas.

DRA

Es una adaptación a alta velocidad del modelo en cascada. Lo consigue gracias a un ciclo de desarrollo extremadamente corto. La filosofía empleada se basa en la reutilización de componentes. Se aconseja si los requisitos están claros y se limita el ámbito del proyecto que se va a acometer. Permite crear un sistema en poco tiempo (60-90 días). Para ello se divide en equipos de desarrollo que trabajan de forma paralela donde cada uno sigue las siguientes fases:

- Modelado de gestión: flujo de información entre las funciones de gestión
- Modelado de datos: se detalla un conjunto de objetos de datos (atributos y relaciones).
- Modelado de procesos: Los objetos sufren una transformación para conseguir el flujo de información necesario para implementar una función de gestión.

- Generación de aplicaciones: asume lenguajes de 4ª generación y uso de herramientas CASE. Prioriza la reutilización:
 - Se usan componentes existentes (y ya probados).
 - Se crean nuevos componentes reusables si es necesario.
- Pruebas y entrega: la mayoría de los componentes ya están probados, solo hay que probar los nuevos y la integración con los ya existentes.

El propio modelo impone restricciones de tipo temporal: un sistema es candidato si se puede articular de forma que cada función de gestión se pueda completar en menos de 3 meses. Si es posible, cada equipo desarrollara una función de gestión y posteriormente se integraran.

Ventajas:

- Introduce la reutilización: mayor productividad y menor probabilidad de errores.
- Introduce el paralelismo.
- Ideal para considerarlo en un paradigma de OO.

Inconvenientes:

- En proyectos grandes demanda una gran cantidad de recursos.
- Compromiso entre cliente y desarrolladores para con la necesaria celeridad en las actividades del modelo.
- Si no se puede segmentar en módulos de forma sencilla es complicado hacer los componentes debido a los solapes entre ellos.
- Si los riesgos técnicos son altos es mejor considerar otro modelo.

Modelos evolutivos

Es necesario un modelo que se haya diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo. Se caracterizan por permitir desarrollar versiones cada vez más completas del software.

Incremental

Proceso de construir una implementación parcial del sistema global y posteriormente ir aumentando la funcionalidad del sistema. Se basa en la aplicación reiterada de un modelo en cascada en la que cada ciclo constituye un incremento del software que da lugar a un producto operativo. Estos incrementos pueden ser refinamientos de los anteriores o desarrollo de nuevas funcionalidades. En el primer incremento se debe abordar el núcleo esencial, requisitos fundamentales que requieren de una priorización. Incrementos posteriores abordaran nuevas funcionalidades. El usuario puede evaluar el resultado de un incremento y establecer nuevos requisitos para el siguiente incremento. El software debe construirse de tal forma que facilite la incorporación de nuevos requisitos (muy técnico en el análisis).

Ventajas:

- La dotación de personal no tiene por que estar disponible para una implementación completa.
- Reduce la posibilidad de que los requisitos cambien durante el desarrollo.
- Ofrece mayor flexibilidad.

Inconvenientes:

- Puede ser muy complejo definir el núcleo operativo.
- Puede ser complicado establecer y priorizar funcionalidades nuevas.
- Las soluciones de incrementos pueden no ser válidas en incrementos posteriores.

Espiral (1988)

Propuesto por Boehm, permite combinar la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistematicos del modelo en cascada, poniendo especial énfasis en el análisis de riesgos para reducir su impacto.

El software se desarrolla en una serie de versiones incrementales, donde las primeras producen prototipos y en las ultimas se obtienen versiones mas completas.

Cada vuelta de espiral considera:

- Determinacion de objetivos, alternativas y restricciones.
- Evaluacion de alternativas (análisis de riesgos)
- Desarrollo del siguiente nivel de producto.
- Planificacion de la siguiente fase.

El ciclo mas interno representa la viabilidad del sistema, el siguiente el análisis de requisitos y así sucesivamente de forma que el modelo se adapta a las necesidades que surjan en cada proyecto. Los bloques de diseño, codificación y prueba se ejecutan de forma secuencial. Después se vuelve a revisar el producto y como mejorar las características obteniendo un nuevo prototipo que se pueda validar. El sistema pasa por un proceso actualizado de desarrollo en cascada que finalmente obtiene una versión del producto.

Ventajas:

- Permite adaptar el proceso de desarrollo a las necesidades cambiantes del proyecto y al conocimiento que se va adquiriendo.
- Permite el manejo de prototipos
- Gestión de riesgos explicita.

Inconvenientes:

- Requiere de mucha habilidad para considerar riesgos.
- Es relativamente nuevo y no se ha manejado tanto como los anteriores.

Gestión y planificación de proyectos

Dirigir un proyecto software requiere:

- Procesos de ingeniería y de gestión bien definidos.
- Metodologías, técnicas y herramientas adecuadas.
- Buena especificación de requisitos.
- Planificación adecuada con calendarios realistas.
- Presupuestos correctos, ajustados y proporcionados
- Personal formado, motivado y adecuado.
- Control y coordinación
- Buena comunicación
- Gestión de riesgos.

La base de una mejora continua es tener procesos repetibles que emplee toda la organización.

Proyecto: acción iniciada por la empresa en la que recursos se organizan para acometer un trabajo único en el que dadas unas especificaciones y dentro de unas limitaciones presupuestarias se intenta conseguir un cambio beneficioso definido por unos objetivos cualitativos o cuantitativos.

Gestión de proyectos: sistema de procedimientos, prácticas, tecnologías y conocimientos que facilitan la organización, gestión, dirección, planificación y control para que el proyecto termine con éxito.

A la hora de aperturar de un proyecto se debería considerar:

- definición
- justificación
- objetivos y metas
- factores críticos de éxito
- riesgo
- alcance
- estructura
- fecha prevista de finalización
- fases, actividades, tareas e hitos.
- Organización (quién hace qué)
- Sistema de control (cómo se va revisar el trabajo)
- Supuestos, dependencias y restricciones.
- Presupuesto y gestión del presupuesto.
- Proyectos relacionados.

A la hora de finalizar el proyecto:

- Finalización del trabajo.
- Transferencia del producto al usuario (formación del usuario, garantizar mantenimiento...)
- Obtención de beneficios (análisis con lo previsto y toma de decisiones correctoras si es necesario)
- Disolución del equipo
- Revisiones post-finalización

Dimensiones de un proyecto software:

- Personas: todos los aspectos relacionados tienen un gran impacto en la productividad del software y su calidad. Se deben cuidar los temas de:
 - motivación
 - equipo de trabajo
 - selección de personal

- formación
 - gestión de personal
- Boehm propone:
- poco pero buen personal
 - explotar las habilidades y motivar a la gente
 - progresión profesional
 - equilibrio en el equipo
 - eliminar la inadaptación.
- Proceso: aplicación de metodologías de desarrollo y gestión y planificación de proyectos.
 - Producto: ocuparse del tamaño y características intentando aplicar las siguientes reglas:
 - Desarrollar el 80% del producto en el 20% del tiempo (pantallas).
 - Desarrollar el producto en 3 meses trabajando 3 personas.
- Reducir el tamaño mejorara la velocidad del desarrollo desproporcionadamente, esto conlleva a:
- Desarrollar solo las prestaciones esenciales.
 - Desarrollar el producto en etapas.
 - Emplear lenguajes de alto nivel.
 - Emplear herramientas CASE, técnicas y metodologías adecuadas.
- Tecnología: se deben usar herramientas efectivas en cada fase (herramientas CASE ...)

Pilares fundamentales

El camino a seguir en la gestión y planificación de proyectos es:

- Control de tiempos y costes del proyecto.
- Realizar proyectos de alta calidad
- Incrementar la productividad
- Reducir la sensación de incompetencia
- Mejorar la imagen externa y seriedad en el trabajo.

Pero este proceso requiere tiempo (de 3 a 4 años), esfuerzo por parte de la organización, formación y herramientas de soporte. Se base en 4 pilares fundamentales:

- Evitar errores clásicos:
 - Relacionados con las personas: motivación débil, personal mediocre, expectativas poco realistas, entorno de trabajo inadecuado, falta de participación...
 - Relacionados con el proceso: inicio difuso, escatimar en actividades iniciales, control insuficiente, omitir estimaciones, planificación incorrecta, gestión de riesgos insuficiente...
 - Problemas de comunicación :
 - entre usuarios y desarrolladores: distorsión
 - entre miembros del equipo de desarrollo.
 - Relacionados con el producto: desarrolladores fascinados por la tecnología, desarrollo orientado a la investigación, cambios en las prestaciones (un 20% de los cambios se producen a lo largo del ciclo de vida, lo que supone un 25% de aumento en el plan)...
 - Relacionados con la tecnología: sobreestimación de ventajas del empleo de nuevas herramientas, cambio de una herramienta o filosofía a mitad del proyecto.
- Bases de desarrollo: se aborda primeramente la gestión de proyectos y luego el desarrollo de software propiamente dicho. La gestión controla la planificación, el coste y el producto. Se fundamenta en determinar el tamaño del producto, asignar los recursos apropiados, crear un plan y controlarlo. La estimación (de tamaño y esfuerzo) y planificación son las bases del desarrollo. Además es necesario un seguimiento mediante reuniones periódicas, informes periódicos sobre el estado del proyecto, revisiones de hitos e informes de presupuestos. Hay que realizar medidas e históricos recogiendo datos métricos para analizar la calidad y productividad del software. Se deben recoger datos del tamaño de los programas,

planificación, tiempo y costes. Con respecto al desarrollo software propiamente dicho todo funcionara mejor si no se cometen los errores en el primer paso. Si un software tiene demasiados errores, se emplea más tiempo corrigiendolo que escribiendolo (en 1995 el 95% de los costes fueron de mantenimiento).

- Gestión de riesgos: los proyectos software incluyen un amplio rango de riesgos (cambios en los requisitos, mala estimación, problemas con la tecnología...). Por esta razón es necesario identificar, estudiar y eliminar las fuentes de riesgo antes de que empiecen a amenazar la finalización satisfactoria de un proyecto. Según Pressman se pueden controlar a 4 niveles:
 - Control de crisis: controlar cuando se han convertido ya en problemas.
 - Arreglar cada error: detectar y reaccionar rápidamente ante cualquier riesgo, después de haberse producido.
 - Mitigación de riesgos: planificar con antelación el tiempo que se necesitaría para cubrir riesgos en el caso de que ocurran.
 - Prevención: crear y llevar a cabo un plan para identificar riesgos y evitar que se conviertan en problemas.
 - Eliminación de las causas principales: identificar y eliminar los factores que pueden hacer posible la presencia de algún tipo de riesgo.

Para la estimación de riesgos es necesario hacer

- una identificación: se parecen a los errores clásicos, pero son menos frecuentes.
- un análisis: Se determina la exposición al riesgo como el producto de la probabilidad y el impacto, que hay que estimar.
- priorizar riesgos :Los proyectos suelen gastar el 80% de su presupuesto en arreglar el 20% de sus problemas. Hay que centrarse en los riesgos que están en ese 20%, para ello se ordena por exposición de riesgo. Hay que poner especial cuidado con aquellos que tengan baja probabilidad pero un fuerte impacto.

Para el control de riesgos es necesario:

- planificar la gestión: determinar quien, que, cuando y como se actúa ante un riesgo.
 - resolución de riesgos: se puede evitar el riesgo, trasladarlo a una actividad que no sea crítica en planificación, informarse, asumirlo, comunicar el posible impacto, controlarlo en base a acciones planificadas o recordar el riesgo para proyectos futuros.
 - Monitorizarlos: permite comprobar como progresa el control de un riesgo e identificar nuevos riesgos.
- Métodos orientados a la planificación: el primer paso para poder efectuar un plan de proyecto es la estimación. Cualquier proyecto debe justificarse, para ello debe existir una valoración de la viabilidad técnica y la inversión detallada, con el coste y los plazos manejados. Una buena regla sería que los beneficios esperados superasen en al menos 5 veces los costes previstos.

Estimación

La estimación tiene diversas dificultades ya que no hay un modelo de como hacerla en todas las organizaciones y se necesitan diferentes estimaciones para las diferentes personas. La utilidad depende de la etapa de desarrollo y se suele hacer muy superficialmente y bajo presión. Además suele haber muchos cambios y la información de partida es muy vaga a veces. Los rápidos cambios en las tecnologías de información son problemas para la estabilidad de un proceso de estimación. Además existe una escasez de históricos y experiencia en estimar proyectos y se tiende a subestimar.

El proceso para planificar requiere de tres estimaciones:

- tamaño: se puede utilizar un enfoque algorítmico para estimar el tamaño del programa a partir de sus prestaciones, un software de estimación o emplear experiencias en proyectos similares.
- Esfuerzo: ayuda a saber cuantas personas hay que incorporar al proyecto y a estimar la

planificación. Para pasar de la estimación de tamaño a la de esfuerzo se puede emplear software de estimación, tablas de conversión tamaño-esfuerzo, históricos o métodos algorítmicos de aproximación. (COCOMO)

- planificación: $\text{planificación} = 3 * \text{personas-mes estimadas}^{1/3}$

Debido al incremento de esfuerzo de intercomunicación entre las personas que componen el equipo de desarrollo a medida que este crece, no existe una relación lineal entre el tamaño y el esfuerzo.

La estimación dentro del desarrollo se escalona a lo largo de varios puntos:

- referencia 0: cuando se realiza la oferta, se da un presupuesto inicial
- referencia 1: al principio de la fase de especificación se da un presupuesto de desarrollo inicial considerando la información del cliente y las técnicas elegidas en la oferta para desarrollar el producto.
- Referencia 2: al concluir la fase de especificación se afina el presupuesto previo con las precisiones aportadas por el contenido funcional y operacional del producto y la naturaleza organizativa del proyecto, presupuesto de desarrollo final.
- Referencia 3: al concluir la fase de diseño de arquitectura que constituye el presupuesto final.

Técnicas de estimación:

- Juicio de expertos: esta basada en la experiencia personal, se suelen utilizar a varios expertos para mayor fiabilidad. El cálculo de la estimación se puede realizar calculando la media de las estimaciones dadas por los expertos (se resiente por valores extremos) o mediante una reunión ininterrumpida hasta llegar a un consenso (personas influyentes determinan el resultado final).
- Delphi: técnica de juicio de expertos donde un coordinador da una especificación del proyecto, los expertos lo rellenan de forma anónima pudiendo formular preguntas y sin poder intercambiar opiniones entre ellos. Una vez realizado este proceso el coordinador ofrece el valor medio de todas las estimaciones para que cada experto las compare con las suyas y realizan una nueva estimación indicando las posibles razones, repitiendo todo el proceso hasta que se llegue a un consenso.
- Delphi de banda ancha: se diferencia de la anterior en que el coordinador reúne a los expertos para intercambien puntos de vista sobre el proyecto antes de realizar la estimación anónima y, una vez comparado la media de estimaciones, los vuelve a reunir para discutir las razones de las diferencias hasta llegar a un consenso.
- Estimación por analogía: variante formal de juicio de expertos en la que se compara el proyecto con uno o más proyectos ya terminados de los que existen datos reales y se deduce una estimación en función de las similitudes y diferencias. Las personas trabajan con su experiencia y con datos reales, si existen o son suficientes para comparar.
- Estimación por descomposición (analythic hierarchy process): consiste en descomponer el producto en componentes y estimar cada uno de ellos por separado. La estimación total será la suma de todas ellas. Para ello es necesario disponer de un diagrama de descomposición (WBS). Es muy intuitiva y permite comprender mejor la tarea, pero puede incurrir a olvidar actividades o incluir actividades que no son propias del proyecto.
- Modelo COCOMO: Constructive Cost Model publicado por Boehm en 1981. Se basa en ecuaciones no lineales obtenidas mediante técnicas de regresión sobre un histórico de proyectos. Tiene en cuenta 3 variables:
 - Tamaño, en KLOC. Es necesario definir que es una línea de código
 - Esfuerzo, expresado en personas-mes
 - Duración, en meses.

Además distingue entre 3 tipos de proyectos:

- Orgánicos: dominio totalmente conocido, con pocas restricciones, un entorno de desarrollo estable, con pocas innovaciones en la arquitectura del software y una organización simple del proyecto.

- Medianos: Equipos formados por personas con y sin experiencia en el dominio, existen restricciones operacionales, el entorno es inestable y la organización del proyecto es de complejidad mediana. Engloba a los proyectos que no se pueden encuadrar en las otras categorías.
- Complejos: existen grandes restricciones, suelen ser sistemas hardware/software complejos con influencia clara de la seguridad o tiempo real. La especificación o arquitectura son nuevos o complejos y la organización es de complejidad alta.

Para cada una de estas categorías se asocian 2 ecuaciones y unas constantes de la forma:

- $Esfuerzo = A * (Tamaño^B)$
- $Duración = C * (Esfuerzo^D)$

Además según el estado en el que se encuentre el proyecto, existen 3 versiones de estimación:

- Básico: permite una estimación durante la fase de análisis previo cuando la mayoría de los factores se desconocen.
- Intermedio: se aplica cuando el proyecto se ha dividido en subsistemas. Permite considerar características más significativas mediante la aplicación de factores correctores relativos a atributos del producto. Boehm identifica 15 factores que se pueden valorar entre 6 valores.
- Detallado: Distribuye el esfuerzo corregido entre las fases del proyecto y sus actividades principales con base en unas tablas de distribución proporcionadas por este modelo.

Este modelo asume que la base de estimación es el número de líneas de código y no es lo mismo ensamblador que Java... Además presupone el uso de un modelo en cascada, por lo que si usamos otros modelos hay que interpretar los porcentajes de distribución a nivel detallado. Las actividades de especificación se deberán estimar por separado, ya que solo contempla desde la fase de diseño hasta las pruebas internas. Presupone un buen hacer del proyecto (planificación, gestión, metodología) y que los factores correctores son independientes entre sí (discutible)

- COCOMO II: se basa en el concepto de punto objeto. Estos pretenden representar la complejidad de las pantallas, informes y componentes con una serie de valores (baja, media, alta que a su vez toma un valor numérico). Se realiza una suma ponderada y luego se corrige la suma con un factor de reutilización. Una vez obtenidos los puntos objeto existe una serie de modelos en función de la fase de desarrollo que se desea estimar:
 - Modelo de composición de aplicación: se emplea durante las primeras etapas, cuando es fundamental la elaboración de prototipos.
 - Modelo de diseño temprano: se emplea una vez que se han estabilizado los requisitos y se ha establecido la arquitectura básica. Se basa en puntos función no ajustados.
 - Modelo post-arquitectura: se emplea durante la construcción del software y se basa en convertir los puntos función no ajustados a KLOC para aplicar un modelo parecido al COCOMO original, pero con otros factores.
- Puntos de función: Function Point Analysis (1979). Pretenden medir la funcionalidad que hay que entregar al usuario. Se requiere un mínimo conocimiento de las funcionalidades y entidades que intervienen. El número de puntos función se basa en el número y complejidad de:
 - Entradas externas: añaden o cambian información en un grupo lógico interno. Una entrada es única si difiere en su formato o arranca procesos diferentes.
 - Salidas externas: es única si difiere en formato o viene de procesos diferentes.
 - Consultas externas: son entradas que generan una salida simple e inmediata.
 - Archivos lógicos internos o grupos lógicos de datos internos: se generan, son usados y mantiene la aplicación. (Base de datos, array...)
 - Archivos de interfaz externos o grupos lógicos de datos de interfaz: grupos lógicos de datos compartidos con otra aplicación, recibidos o enviados a ella.

Para calcular el número de puntos función se toma cada elemento y se multiplica por un peso según una tabla (parámetro x complejidad). La suma da el total de puntos función no ajustados. Se calcula el factor de ajuste basado en la influencia de 14 factores sobre el proyecto. Las características más destacables son la independencia del entorno tecnológico, metodología, experiencia y estilo de programación; además es fácilmente entendible por el usuario. Después de obtener los puntos función hay que hacer una traducción al esfuerzo que supone, y es aquí donde si se contempla la experiencia y estilo de programación basando en históricos o en aproximaciones que establecen una relación entre LOCs y puntos función en algunos lenguajes. El número de puntos función puede servir para comparar el tamaño y planificación con proyectos anteriores. Existen variantes que incluyen más factores de complejidad o hacen referencia a tipos de proyectos más específicos (tiempo real, sistemas...)

- Estimación de primer orden de Jones: permite un cálculo aproximado de la duración del proyecto en meses. Consiste en tomar el total de los puntos de función y elevarlo a la potencia apropiada según una tabla, de forma que nos dice según el tipo de proyecto el resultado en el mejor, en el peor caso y la media. Cruza así el tipo de software con el tipo de empresa. No es la mejor forma de estimar la planificación porque no tiene en cuenta el esfuerzo.

Criticas y recomendaciones:

Los que dependen de LOC hay que establecer un criterio para identificar que es una LOC. Los puntos función pueden dar problemas de duplicidad en la influencia de los factores de corrección al pasarlos al modelo COCOMO. Todos los modelos surgen de estudios estadísticos, de forma que los modelos pierden bastante precisión al utilizarse en entornos distintos a los que se originaron (hay que adaptarlos a cada empresa). Los factores de corrección son difíciles de cuantificar y se consideran independientes cuando no tiene por que ser así. Además manejan un cierto margen de error, que puede ser poco aceptable en algunos casos.

Por todo esto se recomienda que la estimación se ajuste a la empresa y se usen varios métodos de estimación. Un enfoque recomendado sería:

- Primeras estimaciones en base a juicio de expertos basadas en analogías.
- Una vez haya especificaciones medianamente detalladas y se pueda medir su tamaño se pueden aplicar modelos o ecuaciones de estimación.
- Conviene aplicar modelos o ecuaciones locales, de la empresa.

Planificación de proyectos:

Los objetivos de un sistema de gestión de proyectos son satisfacer las necesidades de información de gestión, fomentar la cultura de gestión para que contribuya al aumento de la productividad y a la estructuración y homogeneización de los métodos de trabajo, además de mejorar la comunicación entre todo el personal involucrado.

Para ello hace uso de los siguientes elementos:

- Metodología (base)
- Arquitectura
- gestión de proyectos
- técnicas
- entorno organizativo

Metodología

Sus aspectos fundamentales son la definición de los proyectos, un modelo genérico del ciclo de vida de los proyectos y metodologías específicas por cada tipo de proyecto.

- La definición debe adaptarse a las necesidades de cada organización. Genéricamente un proyecto se compone de un conjunto de recursos, organización, planificación, productos, objetivos y un entorno de riesgo.
- Modelo de ciclo de vida de un proyecto: como fases propias se incluyen las de evaluación y aprobación, planificación y puesta en marcha, ejecución y finalización del proyecto. Además tiene fase adyacentes con respecto a la gestión de la calidad, cambios y riesgos.

Recomendaciones:

Una efectiva gestión del proyecto sucede si se produce una planificación disciplinada, realizando el trabajo de acuerdo a estándares preestablecidos, con mediciones adecuadas y evaluación de resultados, acciones correctivas adecuadas y liderazgo en los equipos.

Conceptos de planificación y seguimiento del proyecto

- **Actividad:** unidad más elemental del nivel de planificación. Se realiza de forma independiente del resto y conduce a la obtención de un resultado tangible a utilizar para el desarrollo de otras actividades del proyecto.
 - **Evento o hito:** actividad especial que no tiene duración y sirve para indicar un acontecimiento. No consume recursos y normalmente se utiliza para describir puntos de control.
 - **Hamaca:** actividad especial que mide el tiempo transcurrido entre dos puntos de la red, las restricciones se deben hacer con las tareas elementales de la red, no con las hamacas.
 - **Recurso:** elemento que va a estar sujeto a compartición. Se pueden hacer diferentes clasificaciones:
 - Humanos, materiales, maquinaria.
 - Consumibles o recurrentes.
- Se distingue entre:
- **Planificación:** establecer el plan.
 - **Programación:** planificación y asignación de recursos.
- **Otras definiciones:**
 - Calendarios
 - Duraciones
 - Esfuerzo
 - Pool de recursos humanos y datos asociados.
 - Elementos de costes adicionales y datos asociados.

Técnicas más usuales en un sistema de gestión de proyectos:

- **Técnicas de representación:**
 - **Diagrama de Gantt:** representa en una escala de tiempos cada una de las actividades mediante barras, que representan su duración en fechas de calendario.
 - **Red de precedencia:** son técnicas basadas en grafos. La red obtenida por estas técnicas es un modelo gráfico que señala las relaciones de secuencia. El análisis de esta red permite identificar el camino crítico. Existen dos técnicas:
 - **Precedence Diagram Method PDM:** utiliza nodos para representar actividades y vectores que representan las dependencias entre ellas. Hay 4 tipos de relaciones:
 - **CC:** La actividad B no puede comenzar hasta que la A haya comenzado, no tiene por qué ser inmediato.
 - **CF:** La actividad B no puede terminar hasta que la actividad A haya comenzado.
 - **FC:** La actividad B no puede comenzar hasta que la actividad A haya terminado.
 - **FF:** La actividad B no puede terminar hasta que la actividad A no haya terminado.
 - **Arrow Diagram Method ADM:** utiliza vectores para representar las actividades y los nodos para indicar las dependencias a través del manejo de hitos. Existen 3 tipos de

relaciones: lineales, de convergencias y de divergencia.

- Histograma: diagrama de barras que muestra de forma visual la distribución de datos cuantitativos de una misma variable. Es típico usarlo para mostrar la asignación de recursos a lo largo del tiempo para analizar su carga de trabajo.
- Técnicas de estructuración:
 - Work BreakDown Structure WBS: técnica que consiste en obtener una estructura de descomposición de un proyecto en actividades con diferentes niveles de detalle.
 - Organisational Breakdown Structure OBS: técnica que consiste en estructurar la organización de un proyecto por unidades organizativas que poseen responsabilidad sobre la realización del proyecto.
- Técnicas de programación:
 - Program Evaluation and Review Technique PERT: se orienta a los eventos (técnica ADM) que permite estimar la duración de un proyecto partiendo de la secuencia de actividades y de una estimación ponderada de la duración de cada actividad.
 - Critical Path Method CPM: se orienta a las actividades (técnica PDM) y permite calcular la lista de actividades con menor flexibilidad en su ejecución, obtener las fechas mínimas esperadas y fechas máximas permitidas y determinar el camino crítico, siendo este el conjunto de actividades cuya holgura total es 0.
- Técnicas de seguimiento (métricas):

Pueden ser de 2 tipos:

 - De producto: cuantifican atributos del producto (tamaño, fiabilidad, complejidad)
 - De proceso: cuantifican atributos del proceso y su entorno (productividad de herramientas, capacidad de las personas...)
 - Tiempo: miden el grado de realización de las actividades, donde la diferencia entre lo real y lo planificado es una indicación de la adherencia del proyecto al plan. Desviaciones significativas indican problemas. Se comparan:
 - fechas de comienzo y fin planificadas y reales de las actividades
 - diferencias entre 2 hitos
 - se hace un análisis del número de requisitos a lo largo del tiempo
 - Esfuerzo: Miden las dedicaciones de los recursos humanos para la realización de las actividades. Se pueden analizar 2 situaciones:
 - una situación prolongada de muy pocos recursos trabajando puede haber un progreso inadecuado por que no se comprenden los requisitos, o por que el equipo es muy productivo...
 - una situación prolongada de exceso de recursos trabajando puede deberse a un problema más complejo de lo esperado, numerosos conflictos...

Además se pueden considerar 4 casos:

 - hacer más horas y terminar antes:
 - hacer más horas y terminar más tarde:
 - hacer menos horas y terminar antes:
 - hacer menos horas y terminar más tarde:

típicamente los recursos más experimentados se necesitan en los requisitos y en el diseño, mientras que en la etapa de construcción se necesita una mayor cantidad de recursos.

 - Coste: miden los costes reales del proyecto con respecto al plan. Hay varios indicadores:
 - Variación de coste: coste presupuestado del trabajo realizado – coste real del trabajo realizado
 - Variación de programa: coste presupuestado del trabajo realizado – coste presupuestado del trabajo planificado.
 - Coste de estimación a la finalización: coste real del trabajo realizado + el coste estimado pendiente hasta finalizar.

Pasos basicos para planificar:

- Definir las actividades
- Determinar las restricciones logicas entre actividades
- Asignar a cada actividad su duracion y recursos.
- Obtener el camino critico
- Afinar la planificacion
 - nivelacion de recursos: para ello hay que definir los perfiles necesarios de los recursos en cada actividad y asignarlos. Se obtienen las cargas de los recursos y mediante histogramas se puede detectar sobrecargas o infrautilizaciones. La eliminacion de dichas sobrecargas puede ser:
 - A tiempo: donde se mantiene la fecha de fin de proyecto y se pueden modificar las duraciones de las actividades que no pertenecen al camino critico.
 - A recurso: se puede variar la fecha de fin de proyecto y las duraciones de cualquier actividad para ajustarte a la disponibilidad de los recursos.
 - romper el camino critico
- establecer la linea base del proyecto: la linea base es una foto fija de la planificacion a efectos de comparacion en los controles o durante el seguimiento.

Seguimiento de proyectos

Persigue comparar los resultados actuales con los planificados para tomar acciones correctivas si existen desviaciones significativas(20%) y acordar los compromisos necesarios con el personal afectado por las acciones correctivas. Para ello es necesario definir una normativa estandar (trimestral, mensual...) y automatizar la recogida de datos aumentando la fiabilidad de los mismos a traves de hojas de trabajo con una periodicidad concreta. Al realizar el seguimiento se establecera el momento en el cual se efectua. Se necesita hacer seguimiento de :

- Tiempo:
 - fecha real de comienzo
 - fecha real de fin
 - duracion pendiente
- Esfuerzo:
 - esfuerzo real realizado
 - esfuerzo pendiente
- Coste

Ademas hay que hacer un analisis y nivelar hasta obtener una replanificacion idonea.

Entorno organizativo

Debe ser part de un procedimiento o manual dentro de cada organización. Debiera contemplar:

- Una estructura de organos y responsabilidades
- normas y politicas de actuacion
- conjunto de procedimientos administrativos
- tecnicas, herramientas y estandares de planificacion a emplear y la forma de operar con ellos.

Gestion de Riesgos

Es un factor importante cuyo objetivo es identificar, estudiar y eliminar las fuentes de riesgo antes de que empiecen a amenazar el proyecto. Un riesgo es la probabilidad de que en un punto del ciclo de vida no se alcancen los objetivos propuestos con los recursos disponibles, es decir que tenga un efecto negativo.

Se pueden clasificar en :

- Estrategicos: relacionados con la organización (perdidas y beneficios, inversiones, imagen..)
- Comerciales: relacionados con la venta del proyecto, seguimiento del cliente, precio...
- Contractuales y financieros: se refiere a penalizaciones, niveles de calidad, calendario de pagos...
- De gestion: relacionados con la organización de los proyectos (recursos y equipos, calendarios...)
- De proyecto: causados por la especificacion, diseño, desarrollo...
- Riesgos de explotacion: que pueden ocurrir durante la explotacion (fallo del sistema que puede llevar a un accidente).

Una buena planificacion, organización y gestion del proyecto puede verse muy afectada por los riesgos que se van produciendo. Para abordarlos se puede optar por solventar los problemas a medida que aparecen o mitigar los posibles impactos con una planificacion.

Identificacion de riesgos

Hay varias formas de identificarlos:

- Identificacion de resultados insatisfactorios permite obtener las posibles causas y por tanto identificar los riesgos.
- Se puede considerar cada area o clase mediante clasificaciones o marcos. Lo importante es disponer de unas tipologias establecidas y que esta aproximacion sea lo mas metodica posible.
- Particionar el espacio del problema, esto es, examinar individualmente cada tarea principal del plan de proyecto (WBS). Se trata de identificar entre 4 y 6 riesgos principales en cada area.
- Estudio de evento de riesgo y efecto, el mismo evento de riesgo y la misma causa pueden tener a veces efectos muy distintos.
- Listas de comprobacion: se construyen a partir de informacion historica, conteniendo los riesgos mas habituales en los proyectos de una organización. Permiten una identificacion muy rapida y relativamente sencilla, aunque es practicamente imposible que incluya todos los riesgos posibles. Se aconseja utilizar una lista de comprobacion como punto de partida, pero permitiendo despues incluir nuevos riesgos especificos del proyecto.

Una vez identificados se elabora una lista de riesgos que pueden afectar al proyecto y se agrupan para saber sobre cuales poner especial atencion.

Valoracion o cuantificacion de riesgos

Esta labor es fundamental para la priorizacion, y asi atender aquellos que puedan afectar mas al proyecto. Una vez cuantificados, ya se pueden comparar entre si. Para cada riesgo se calcula su exposicion como el producto entre la probabilidad por el impacto, y luego se determina distintos niveles de actuacion y se establecen planes de prevencion o de contingencia. Tambien se pueden valorar de forma subjetiva (estimacion de la persona con mas experiencia, metodo delphi, aproximaciones subjetivas y asignacion posterior de valores numericos, trabajar directamente con subjetividad...). Lo normal es que se necesiten datos historicos para asignar un valor numerico a la probabilidad de ocurrencia.

Una vez se tenga la lista hay que analizar cuales son los que mayor exposicion al riesgo tienen, y

tener especial cuidado con aquellos que tiene un alto impacto y baja probabilidad.

Analisis de riesgos

Esta tarea involucra:

- Estudio de las posibles alternativas
- definicion de estrategias de mitigacion (contencion o prevencion)
- planificacion de las actividades de control de riesgo

Algunas de las estrategias consisten en:

- Transferir: sacar el riesgo a traves de subcontratas
- Prevenir: considerar caminos alternativos que eviten el riesgo
- Controlar: aceptar la posibilidad de que se de, establecer medidas, planificas las acciones de contencion y hacer un seguimiento del riesgo frecuente.

Control y seguimiento de riesgos

Se centra en asegurar que los planes se lleven a cabo como estaba previsto, revalorizar y reanalizar los riesgos y efectuar el seguimiento del riesgo del proyecto. Los riesgos deben ser revisados según determine la organización y por regla general se realiza cada vez que se tenga que hacer un informe de seguimiento de proyecto.

Gestion de la configuracion del software

Los objetivos son facilitar la visibilidad sobre el estado del producto y su evolución, y mantener su integridad (satisface las necesidades de los usuarios, requisitos de rendimiento y se puede trazar su evolucion).

Configuracion Software: conjunto de toda la informacion y productos utilizados o generados en un proyecto.

Elemento de configuracion software (ECS): cada uno de los componentes de la configuracion software. Debe ser un elemento que se pueda definir y controlar de forma separada. Además dependiendo de sus características puede requerir de su descomposicion en varios ECS.

Linea base: facilita el control de cambios ya que permite cambios rapidos antes de que el ECS pase a formar parte de una linea base y en el momento que se establece se debe aplicar un procedimiento formal. Hay dos definiciones formales:

- Visto desde el proceso: punto de referencia que queda marcado por la aprobacion de uno o mas ECS mediante una revision tecnica formal.
- Visto desde el producto: ECS revisados y aceptados que sirven como base para el desarrollo posterior y que solo se pueden cambiar a traves de un proceso formal de control de cambios.

Existen herramientas CASE que automatizan la GCS:

- Control de versiones: registro historico de las distintas versiones que permita la recuperacion de cualquiera de ellas. Se trabaja con:
 - Version: instancia de un ECS en un momento dado que es almacenada en un repositorio y puede ser recuperada.
 - Revision: distintas versiones que aparecen a lo largo del tiempo. Cada revision en la cadena de revision es una actualizacionde, y viene a sustituir, la revision anterior.
 - Variantes: versiones de un ECS que coexisten en un determinado momento y se diferencian en distintas características. Representa que un objeto satisfaga varios requisitos. Una variante no reemplaza a otra si no que abre un nuevo camino de desarrollo y se puede trabajar simultaneamente sobre ellas. Pueden ser:

- Temporales: a veces es necesario que trabajen simultaneamente sobre la misma version para que no ocurran conflictos. Una vez acabadas las modificaciones es necesario mezclar todas las variantes.
 - De usar y tirar: se pretende explorar diferentes soluciones alternativas en paralelo y quedarse con la mejor. Tambien se pueden hacer variantes para las pruebas.
 - Permanentes: que pueden ser por requisitos del usuario (el idioma) o de plataforma (sistema operativo).
- Release: se suele llamar a una configuracion del sistema que se va a comercializar o entregar al cliente.
- Construccion: gestiona la compilacion y enlazado de componentes de forma eficiente. Para ello es necesario saber que componentes enlazar, en que version y donde estan tomando esta informacion de la identificacion de la configuracion y del control de versiones.
- Gestion de problemas: seguimiento de los problemas detectados sobre el software tanto en pruebas como de los informes de fallos de usuario. Se necesita un registro de la incidencia, un responsable, monitorizacion, registro de actividades de correccion...
- Control de trabajo en equipo: controlar las interacciones que se producen entre los desarrolladores de un producto para que evitar el peligro de sobreescritura de cambios.

Identificacion de la configuracion

Consiste en identificar y asignar nombre significativos a cada uno de los ECS. Al principio del proyecto se deben proporcionar: :

- servicios de identificacion:
 - respecto a los ECS:
 - Identificacion de la estructura y componentes: identificar la jerarquia software (sistema, subsistema, componente...). Lo importante es la visibilidad que se va a controlar.
 - Selección de los ECS: sobre los productos generados indicados por la metodologia y sobre los productos utilizados durante el desarrollo.
 - Esquema de identificacion: definicion del metodo que se va a utilizar para identificar de forma univoca cada ECS (codigo, nombre, linea base a la que pertenece...) y su version (numero y fecha). Toda la informacion puede ir codificada en un unico codigo y este puede ser significativo (facil localizacion, interesante si es facil de recordar) o no (facil de asignar, dificil de localizar).
 - Relaciones en GCS: se puede considerar que los ECS son objetos y que estan conectados entre si mediante relaciones. Esta informacion ayuda a comprender el impacto de un cambio y donde se situa un ECS respecto a otro. Estas pueden ser, entre otras, de composicion, dereivacion, dependencia, sucesion, equivalencia...
 - respecto a las lineas base:
 - Definicion de las lineas base. Se establecen en hitos previamente especificados, lo normal al final de determinadas fases. Se puede definir de manera fisica o logica.
- Servicios de accesibilidad con la definicion de bibliotecas: una biblioteca es una colección controlada de software o documentacion cuyo objetivo es ayudar en el desarrollo, uso o mantenimiento del software. Hay que definir cuales se van a usar, quien las gestiona y establecer procedimientos para introducir elementos a la biblioteca y de acceso. Pueden trabajar a distintos niveles:
 - En cada puesto o grupo:
 - Area de trabajo
 - Area de integracion
 - Nivel de proyecto:
 - Biblioteca de proyecto
 - Biblioteca maestra

- Nivel de empresa:
 - Repositorio SW
 - Repositorio de componentes reutilizables

Durante el proyecto se lleva a cabo la identificacion/etiquetacion de los ECS, establecer las lineas base, mantener las relaciones de configuracion y las bibliotecas software. Servicios de accesibilidad con la definicion de bibliotecas. Durante el proyecto se lleva a cabo la identificacion/etiquetacion de los ECS, establecer las lineas base, mantener las relaciones de configuracion y las bibliotecas software.

Control de cambios en la configuracion

Es la actividad mas importante que proporciona un mecanismo riguroso para controlar los cambios. Combina procedimientos humanos y uso de herramientas automaticas. Considera 2 tipos de cambios:

- Correccion de errores
- Mejora del sistema

Para ello es fundamental la trazabilidad de los requisitos.

Se pueden establecer 3 niveles de control:

- Informal: antes de que el ECS forme parte de una linea base.
- Semiformal: donde un ECS sufre un revision tecnica formal y se convierte en una linea de base. Debe recibir aprobacion del responsable según procede.
- Formal: se suele adoptar una vez que se empieza a comercializar el producto. Debera ser aprobado por el comité de control de cambios.

El comité de control de cambios tiene la autoridad sobre la aprobacion, denegacion y priorizacion de un cambio. Este debe tener una vision global para poder evaluar el impacto. La responsabilidad sobre del control recae sobre el comité, el jefe de proyecto, bibliotecario y miembros de un proyecto. Es necesario establecer de forma precisa cual sera el proceso de gestion de cambios que se va a utilizar.

El proceso formal de control es el siguiente:

- Se presenta una solicitud de cambio: puede ser mediante un formulario de solicitud o por un informe de incidencia.
- Se clasifica y registra la solicitud
- Se aprueba o rechaza
- se evalua y como resultado se obtiene el informe de cambio.
- Se presenta el informe al comité. Si se considera beneficioso se genera una orden de cambio y el ECS se da de baja en la biblioteca.
- Se realiza, controla y se sigue el cambio.
- Se certifica mediante una revision, cargando en la biblioteca el ECS corregido.
- Notificacion al originador del cambio.

La generacion de informes de estados de la configuracion permiten mantener el estado y evolucion de la configuracion. Su importancia es que da continuidad al proyecto, evita duplicidad, evitar repetir errores, repetir lo que se hizo bien... y para ello es necesario registrar toda la informacion mediante actividades de:

- Captura de la informacion: la cantidad y tipo depende de las características del proyecto. El plan de GCS debera indicar la informacion a capturar mediante registros:
 - Identificacion de la configuracion.
 - Control de cambios
 - Auditoria de la configuracion

- Actas de las reuniones del comité de control de cambios
- Almacenamiento de la información: suele ser en una base de datos y utilizar herramientas automatizadas para gestionarla.
- Generación de informes: hay que definir que tipo y para quien.
Según prevision:
 - Planificados
 - Bajo demanda
 Según complejidad:
 - Directos: contenidos de los registros.
 - Indirectos: suponen una mayor elaboración.

Auditoria de la configuración

Tipos de actividades:

- revisiones de fase: se realizan al final de cada fase y el objetivo es descubrir problemas.
- Revisiones de cambios: verificar que se han realizado correctamente los cambios aprobados sobre una línea base.
- Auditorias: se realizan al final y su objetivo es examinar el producto en su conjunto.

La tarea de revisión implica:

- Verificar la configuración actual del software con respecto a la línea base anterior (correspondencia y trazabilidad)
- Validar que la configuración actual del software satisface la función que se esperaba.
- Valorar si una línea base es aceptable o no.

Se pueden distinguir 3 tipos de auditorias:

- Funcional: cuyo objetivo es asegurar que se han completado todas las pruebas necesarias y que satisface los requisitos impuestos.
- Física: verificar la adecuación, completitud y precisión de la documentación.
- Revisión formal de certificación: que certifica que el ECS se comporta correctamente una vez que se encuentre en su entorno operativo.

Plan de gestión de la configuración

La estructura según IEEE:

- Introducción: definición de alto nivel de la GCS.
- Especificaciones de gestión para actividades de GCS:
 - organización
 - responsabilidades
 - implantación del plan
 - políticas, directivas y procedimientos aplicables a la GCS: niveles de software, nombrado de programas y módulos...
 - Actividades de GCS:
 - Identificación de la configuración:
 - Descripción del esquema de identificación
 - Enumeración de las líneas base.
 - Bibliotecas y repositorios a utilizar
 - Control de la configuración:
 - Mecanismos de :
 - iniciación
 - evaluación de cambios
 - aprobación o rechazo

- verificación de cambios aprobados
 - gestión de problemas
 - control de versiones
- Informes de estado de la configuración
- Auditoría de la configuración: Descripción de cada una de las auditorías y revisiones que se van a realizar.
- Control de proveedores y subcontratas.
- Recogida y retención de registros

Historia

Finales de los años 70 -> crisis del software mas del 75% de proyectos terminados sin explotacion
-> 19% de soft usados con trabajo extra o abandonados despues

Finales de los años 80-> entre el 5% y el 10% son usables directamente, entre el 30%-40% nunca fueron usados

1994 proyectos exitosos 16%

1996 proyectos exitosos 27%

1998 proyectos exitosos 26%

2000 proyectos exitoso 28%

Actualmente(2006):

- proyectos exitosos 35%

- proyectos deficientes (chagenlled) 46%

- proyectos fallidos 19%

Siglas

ERS: especificacion de recursos software

ROI: retorno de la inversion

CMM: capability maturity model

SEI: Software Engeneering Institute

ISO: International Organization for Standardization

SGC: sistema de gestion de calidad

ANSI: American National Standards Institute (organismo nacional de normalizacion)

AENOR: Asociacion Española de Normalización y Certificación, publica normas UNE(Una Norma Española)

CEN/CENELEC:Comite Europeo de Normalización, publica normas EN (European Norm)

GCS: Gestión de Configuración Software

ENAC: Entidad Nacional de Acreditación.