



2022

DOSSIER

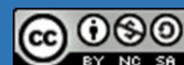


INGENIERIA DE SOFTWARE

Mgtr. Maria Vanesa Doria

Lic. Maria Carolina Haustein

Lic. Daniela de los Ángeles Lobos Anfuso



Dossier de Ingeniería de Software

María Vanesa Doria – María Carolina Haustein – Daniela Lobos Anfuso

Doria, Maria Vanesa

Dossier de Ingeniería de Software / María Vanesa Doria; María Carolina Haustein ; Daniela Lobos Anfuso. - 1a ed. - Catamarca: Editorial Científica Universitaria de la Universidad Nacional de Catamarca, 2023.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-661-457-3

1. Software. I. Haustein, María Carolina. II. Lobos Anfuso, Daniela. III. Título.
CDD 005.12

ISBN: 978-987-661-457-3

Queda hecho el depósito que marca la ley 11.723.

E.C.U. 2023 Avda. Belgrano 300 - Pab. Variante I - Planta Alta - Predio Universitario -
San Fernando del Valle de Catamarca - 4700 - Catamarca - República Argentina

Publicación en *Open Access* - Acceso abierto



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

Adaptar — remezclar, transformar y crear a partir del material

Bajo las condiciones siguientes:

Reconocimiento (BY) — Debe reconocer adecuadamente la autoría, proporcionar un enlace a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciador o lo recibe por el uso que hace.

NoComercial (NC) — No puede utilizar el material para una finalidad comercial.

CompartirIgual (SA) — Si remezcla, transforma o crea a partir del material, deberá difundir sus contribuciones bajo la misma licencia que el original.

INTRODUCCION

Los temas presentados para el abordaje teórico-práctico del espacio curricular Ingeniería de Software III, del 4 año de la carrera Ingeniería en Informática, es desde la postura de la corriente de pensamiento constructivismo sociocultural, considerando a las tecnologías como instrumentos culturales que intervienen en el proceso de enseñanza mediante el diseño y la puesta en práctica de estrategias docentes que facilitan el desarrollo de los contenidos curriculares y menciona los roles de los docentes y los estudiantes de acuerdo a esta corriente:

Los docentes toman el rol de guías u orientadores en la construcción del conocimiento, el cual es generado de manera dinámica e independiente, considerando los conocimientos previos de los estudiantes, desde un contexto sociocultural determinado. El estudiante, a su vez, se concibe como un sujeto activo, participativo, creativo, interesado en su quehacer académico y responsable de su aprendizaje. La asunción de estos dos roles genera interacciones, tanto dentro como fuera del aula, en las cuales las TIC tienen una función importante para mediar las comunicaciones que se efectúan en el proceso educativo.

Los textos que componen el Dossier fueron seleccionados de autores que tratan los contenidos temáticos del espacio curricular, con el objetivo de ser una herramienta de lectura que acompañe a los estudiantes a lo largo de la cursada y le permitirá a los estudiantes desarrollar habilidades para comprender, analizar e interpretar textos académicos, científicos y videos. Y ser además un material didáctico que pueda servir a otros colegas docentes en el ejercicio de la enseñanza de la Ingeniería de Software.

OBJETIVOS DEL ESPACIO CURRICULAR

En función a las competencias que debe tener un ingeniero, los objetivos son:

Objetivos principales

- Situar al alumno como un profesional, en la Gestión de Proyectos desde la perspectiva Ingeniería de Software.
- Presentar en profundidad los temas relativos a la Gestión de Proyectos de Software.

Objetivos específicos:

Que el alumno logre:

- Demostrar aptitud para trabajar en equipo; comunicarse efectiva y eficazmente con sus compañeros de equipo de trabajo.
- Administrar el talento de sus colegas para llegar a una solución óptima e innovadora.
- Resolver problemas de Ingeniería, identificando una situación problemática; los datos pertinentes al problema y evaluando el contexto particular del problema para generar diversas alternativas de solución.
- Administrar, planificar, coordinar, realizar seguimiento y control de todas las actividades y los recursos asignados para la ejecución de un proyecto en el tiempo y con los costos establecidos.
- Reconocer la necesidad de aprender de forma continua y autónoma a lo largo de la vida.
- Reconocer oportunidades, crear escenarios de posibilidades y delinear una visión de futuro.



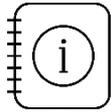
Contenido

INGENIERIA DE SOFTWARE – NOCIONES BÁSICAS.....	0
GESTION DE RECURSOS HUMANOS	3
PLANIFICACION DE PROYECTOS DE SOFTWARE.....	21
PLANIFICACION TEMPORAL Y SEGUIMIENTO DEL PROYECTO	31
GESTION DE PROYECTOS DE SOFTWARE.....	44
METRICAS DEL PROCESO DE SOFTWARE.....	53
GESTION DEL RIESGO.....	60
GESTION DE CONFIGURACIÓN DEL SOFTWARE	69
CALIDAD DEL SOFTWARE.....	75

INGENIERIA DE SOFTWARE – NOCIONES BÁSICAS



Es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Boehm, 1976)



Boehm es un Ingeniero informático estadounidense Profesor emérito en el departamento de Ciencias Tecnológicas en la Universidad del Sur de California

OBJETIVOS DE LA INGENIERIA DE SOFTWARE

- Diseñar aplicaciones informáticas que se ajusten a las necesidades de las organizaciones.
- Mejorar el diseño de aplicaciones o software de tal modo que se adapten de mejor manera a las necesidades de las organizaciones o finalidades para las cuales fueron creadas.
- Brindar mayor exactitud en los costos de proyectos y tiempo de desarrollo de los mismos.

PRINCIPIOS DE LA INGENIERIA DE SOFTWARE



El diccionario define la palabra principio como “una ley importante o suposición que subyace y se requiere en un sistema de pensamiento”.

David Hooker, en 1996, propuso siete principios que se centran en la práctica de la ingeniería de software como un todo:

1. Agregar valor para los usuarios finales
2. Mantenerlo sencillo
3. Fijar la visión (producto-proyecto)
4. Reconocer que los consumidores deben comprender lo que usted ofrece
5. Abrirse al futuro
6. Planear la reutilización
7. ¡pensar!

SISTEMAS BASADOS EN COMPUTADORAS.

Un sistema basado en computadoras es un conjunto de elementos organizados para generar un objetivo predefinido, procesando información.

Metas:

- Soportar alguna función de negocio.
- Desarrollar productos para vender.

Para conseguir el objetivo, un Sistema Basado En Computadoras hace uso de varios elementos:

1. **Software:** Programas de computadora, estructuras de datos y su documentación que sirven para hacer efectivo el método lógico, procedimiento o control requerido.
2. **Hardware:** Dispositivos electrónicos que proporcionan capacidad de cálculo, dispositivos de interconexión (por ejemplo, conmutadores de red, dispositivos de telecomunicación) y dispositivos electromecánicos (por ejemplo, sensores, motores, bombas) que proporcionan una función externa, del mundo real.
3. **Personas:** Usuarios y operadores del hardware y software.
4. **Documentación:** Manuales, formularios y otra información descriptiva que plasma el empleo y/o funcionamiento del sistema.
5. **Procedimientos:** Los pasos que definen el empleo específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

DIFERENCIA ENTRE INGENIERÍA DE INFORMACIÓN E INGENIERÍA DE PRODUCTO.

Ingeniería de la información.	Ingeniería de producto.
“Es la aplicación de una serie de técnicas formales integradas para el planeamiento, análisis, diseño y construcción de sistemas de información para la totalidad de una empresa, o un sector importante de ella” (Murillo Félix)	Proceso de diseño y desarrollo de un equipo, sistema o aparato de forma tal que se obtiene un elemento apto para su comercialización mediante algún proceso de fabricación. (Pressman, 2010)
La meta es definir arquitecturas que permitan a las empresas emplear la información eficazmente y crear un plan global para ampliar dichas arquitecturas.	La meta es traducir el deseo de un cliente, de un conjunto de capacidades definidas, a un producto operativo.
<p>Se deben analizar y diseñar 3 arquitecturas:</p> <ol style="list-style-type: none"> 1. Arquitectura de Datos: Proporciona una estructura para las necesidades de información de un negocio o una función de negocio. Los componentes de la arquitectura son los objetos de datos que emplea la empresa. Estos fluyen entre las funciones de negocio 2. Arquitectura de Aplicación: Comprende aquellos elementos de un sistema que transforman objetos dentro de la arquitectura de datos por algún propósito del negocio. 3. Infraestructura de Tecnología: Proporciona el fundamento de las arquitecturas de datos y de aplicaciones. La infraestructura 	<p>Consta de:</p> <ul style="list-style-type: none"> • Ingeniería De Requisitos: Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado. Es la parte de la ingeniería del software que aborda el problema de la definición de los servicios que el sistema ha de proporcionar y de establecer las restricciones operativas del mismo. • Ingeniería De Componentes: se enfoca en la descomposición del diseño en componentes funcionales o lógicos.

Ingeniería de la información.	Ingeniería de producto.
<p>comprende del hardware y software empleados para dar soporte a las aplicaciones y datos. Esto incluye computadoras y redes de computadoras, enlaces de telecomunicaciones, tecnologías de almacenamiento y la arquitectura diseñada para implementar estas tecnologías. Para modelar las arquitecturas de sistema definidas anteriormente, se define una jerarquía de actividades de ingeniería de la información.</p>	
<p>Las técnicas de la Ingeniería De La Información contienen a las de la Ingeniería Del Software en una forma modificada.</p>	<p>La Ingeniería De Producto es un enfoque de la Ingeniería de Software. Sucede cuando el contexto de trabajo se enfoca en una empresa.</p>

MODELOS DE DESARROLLO DE SOFTWARE Y METODOLOGÍAS DE DESARROLLO DE SOFTWARE



PARA PENSAR

1. Qué significan los siguientes términos:
 - a) Metodología de desarrollo de software.
 - b) Modelo de Ciclo de Vida para el desarrollo de software.
 - c) Etapas dentro de un ciclo de vida de desarrollo de software.
 - d) Rol que puede cumplir una persona en el desarrollo de software.

Bibliografía Consultada

Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

GESTION DE RECURSOS HUMANOS

Trabajar en un proyecto de software, implica el trabajo de seres humanos, cuyo aporte más importante es su creatividad, pero para que dicho trabajo se haga de manera y se lleguen a las metas propuestas debe ser *coordinado*.

La **gestión de los recursos humanos** es la encargada de *coordinar* las funciones que deben cumplir las personas dentro de la organización, como así también buscar las personas calificadas.

EQUIPO DE TRABAJO VERSUS TRABAJO EN EQUIPO EN LA ORGANIZACIÓN

El hombre como ser social se enfrenta continuamente al trabajo en grupo. Los miembros de una empresa, pertenecen simultáneamente a **grupos formales e informales**.

- **Grupos formales:** se establecen de acuerdo a sus prescripciones en el seno de un grupo (por ejemplo equipo de trabajo).
- **Grupos informales:** se constituyen al margen de dichas prescripciones; surgen de interacciones entre personas que dependen mutuamente en ciertos aspectos (por ejemplo, porque tienen preocupaciones o intereses complementarios o similares o porque tienen aspiraciones y afinidades semejantes).

Los autores Stephen Robbins y Timothy Judge hacen una diferencia entre trabajar en equipo y un equipo de trabajo¹:

Trabajo en Equipo	Equipo de Trabajo
Es aquel que interactúa sobre todo para compartir información y tomar decisiones que ayuden a cada uno de sus miembros a realizar el objeto de su responsabilidad.	Es cuando un grupo de trabajo genera una sinergia positiva a través del esfuerzo coordinado y los esfuerzos de sus individuos dan como resultado un nivel de rendimiento superior a la suma de los aportes individuales



[Diferencias entre grupo y equipo | ¿Tienes un equipo de trabajo o trabajas en equipo?](#)

¹ <https://capitalhumano.emol.com/4385/cual-es-la-diferencia-entre-trabajar-en-equipo-y-un-equipo-de-trabajo/>

EL RECURSO HUMANO EN UNA EMPRESA DE SOFTWARE.

En un estudio publicado por el IEEE, se preguntó a los vicepresidentes (VP) de ingeniería de tres grandes compañías tecnológicas cuál era el elemento más importante para el éxito de un proyecto de software. Respondieron de la siguiente manera:

VP 1: Supongo que, si tienes que elegir una cosa que sea la más importante en nuestro ambiente, diría que no son las herramientas que usamos, es el **personal**.

VP 2: El ingrediente más importante que fue exitoso en este proyecto fue tener **gente** inteligente [...] en mi opinión, muy pocas cosas más importan [...] La cosa más importante que haces para un proyecto es **seleccionar al personal** [...] El éxito de la organización de desarrollo de software está muy, muy asociada con la habilidad para reclutar **buen personal**.

VP 3: La única regla que tengo en la administración es asegurarme de que tengo **buen personal**, gente realmente buena, y que hago crecer gente buena y que proporciono un ambiente en el que la gente buena puede producir.

GESTIÓN DEL TALENTO HUMANO

Las nuevas realidades económicas llevaron a la evolución de la Dirección de Personal clásica a la Dirección de Recursos Humanos, ampliando sus funciones, es decir, abarcando no solo el tradicional papel administrativo, sino también un papel estratégico. Puesto que la principal fuente de producción de la economía ha pasado del capital físico al capital intelectual.

OBJETIVOS ESTRATÉGICOS EN LA GESTIÓN DEL TALENTO HUMANO

1. **Lograr la equidad e integración del personal:** Está orientado hacia la empresa, a través de sistemas formales para establecer justicia, normas, valoraciones, comités de consulta, negociación colectiva y mejora en la calidad de vida laboral.
2. **Elevar el rendimiento competitivo y operativo:** Está orientado hacia el mercado, hacia el exterior, la respuesta de la gestión de recursos humanos está encaminada a la dirección por objetivos, remuneración por resultados, planificación y desarrollo de Recursos humanos.
3. **Innovación y flexibilidad (Eficiencia a largo plazo):** Está orientado hacia el interior y el exterior de la empresa, la respuesta de la gestión de recursos humanos está en las coexistencias de estructuras formales e informales, comunicaciones abiertas y gestión participativa, así como la seguridad psicológica, garantía de empleo dirigida

a la formación, movilidad, participación en beneficios, información económica y estrategias, además de las oportunidades del futuro que favorezcan el cambio.

PARA PENSAR



CASO DE ESTUDIO

La teoría de las relaciones humanas surge para corregir la deshumanización en el trabajo. El experimento Hawthorne considera diversos factores y analiza cómo funcionan los grupos de trabajo.

Leer el caso y responder:

- a) ¿Por qué los investigadores se vieron sorprendidos en los experimentos iniciales realizados en Hawthorne?
- b) ¿Por qué el grupo que se investigaba mantuvo en dichas experiencias un alto nivel de producción?
- c) Elabora tus conclusiones de este caso de estudio teniendo en cuenta cómo influye los grupos informales en la experiencia.

LOS EXPERIMENTOS DE HAWTHORNE

Entre 1927 y 1932 los experimentos efectuados en la Western Electrical Company de Chicago en la planta de Hawthorne, al estudiar los grupos informales, trataron de demostrar la relación existente entre el ambiente físico del lugar de trabajo (nivel de temperatura, iluminación, etc.) y la productividad de los obreros.

Sorprendidos al comprobar que una mejor iluminación no producía mejoras considerables en la producción, los investigadores aislaron a un grupo de obreros en una sala de pruebas, para observarles más de cerca. Se realizaron informes diarios de todos los acontecimientos, conversaciones, discusiones, etc. Ninguno de los factores examinados tenía relación directa con los niveles de productividad. No obstante, comprobaron que la producción se incrementó considerablemente y se mantuvo estable, pese a las alteraciones del ambiente físico que dificultaba el trabajo.

Esta situación, no cabe duda, había alterado por completo la situación social de los trabajadores implicados en el estudio, al aislarlos como grupo del resto de la planta. Se comprobó entonces que el trabajo en la planta era rutinario, repetitivo, monótono y no ofrecía la posibilidad del contacto personal entre los obreros. En la sala experimental, en cambio, se había desarrollado un grupo que posibilitó un trabajo coordinado, basado en la cohesión, y fue posible conseguir una elevada autoestima y ser centro de atracción entre los demás trabajadores.

El alto nivel de producción lo mantenían las presiones informales del grupo, pues los obreros habían comprobado que su situación era favorable en todos los aspectos y que solo se mantendría si mantenían alta la productividad.

LIDERAZGO. MOTIVACIÓN. COMUNICACIÓN. RESOLUCIÓN DE CONFLICTOS.

En general, **liderazgo** es el proceso mediante el cual un individuo ejerce más influencia que otros en el desarrollo de las funciones grupales. No se trata de influencias ocasionales, ni de influencias ligadas al ejercicio de una tarea grupal concreta, se trata de una influencia permanente, que se dirige sobre un número relativamente amplio de personas y durante un tiempo considerable. El liderazgo puede ser innato o bien adquirirse con sólo el deseo de ser líder.

La **motivación** es la habilidad que tiene el líder para alentar (mediante “empuje o jalón”) al equipo de trabajo para producir a su máxima capacidad.



En la Tabla 1 se muestran algunas de las definiciones de liderazgo más conocidas.

Tabla 1. Definiciones de liderazgo por autores y estándares

Autor, año	Definición
Verma, V.K. (1996)	Competencia para dirigir al equipo y a los <i>stakeholders</i> con el fin de lograr los objetivos del proyecto dentro de las restricciones estipuladas. Se basa en la obtención de resultados a través de las personas, en lugar de estar por encima de ellas.
Association for Project Management - APM (2006)	Es la capacidad de establecer la visión y la dirección para influir y alinear a otros hacia el logro de un propósito común y para empoderar e inspirar a las personas en pro del éxito del proyecto.
Paredes C., S. & Rojas C., N. (2007)	Capacidad de transformar un sueño o visión en realidad, con y a través de la participación voluntaria de los demás.
International Project Management Association (IPMA) (2015)	Capacidad de proporcionar dirección y orientación a individuos y grupos, de elegir y aplicar estilos de gestión apropiados en diferentes situaciones, incluyendo patrones de comportamiento de acuerdo con tipo de situación, métodos de comunicación, actitudes frente al conflicto, formas de afrontar el comportamiento del equipo del proyecto, toma de decisiones y delegación de funciones.
Australian Institute of Project Management (AIPM) (2016)	Conjunto de habilidades y conocimientos de gestión personal, interpersonal y de equipo, necesarios para que un gerente de proyectos lidere un equipo en un contexto organizacional dinámico. Capacidad de aplicar estrategias y proporcionar orientación a los miembros del equipo para administrar relaciones y resultados diversos, ambiguos y variables dentro de un proyecto.
Project Management Association of Japan (PMAJ) (2016)	Capacidad de generar compromiso en otros y de toma de decisiones con el fin de recuperar la situación.
Cohen, E. (2017)	Habilidad para adaptarse a los cambios, motivar a otros, comunicar su visión y la del proyecto.
Project Management Institute, 2017	Capacidad para guiar, motivar e inspirar a los miembros del equipo y otros <i>stakeholders</i> , con el fin de sobrepasar los problemas que se presenten y alcanzar efectivamente los objetivos del proyecto.
Association for Project Management - APM, 2006	Capacidad de encabezar un equipo e inspirarlo a hacer bien su trabajo.

Fuente: Constructo de liderazgo en Gerencia de Proyectos | Beltrán Sánchez - Salas Ortiz - Vega Vargas | Revista IDGIP | Volumen 1 N.º 3 Enero-diciembre de 2020 | 6-20

COMUNICACIÓN

La comunicación es un el diálogo fluido permanente y consecutivo, entre dos o más personas.



En la Tabla 2 se muestran algunas de las definiciones de comunicación.

Tabla 2. Definiciones de comunicación por autores y estándares.

Autor, año	Definición
Verma, V.K. (1996)	Competencia que le permite al gerente de proyectos dar instrucciones, concretar reuniones y transmitir ideas e información hacia los miembros del equipo o cualquier <i>stakeholder</i> interno o externo. Se dan la comunicación interpersonal y la escucha activa.
Association for Project Management (APM) (2006)	Capacidad de dar, recibir, procesar e interpretar información. La información se puede transmitir de forma verbal, no verbal, activa, pasiva, formal, informalmente, consciente o inconscientemente.
International Project Management Association (IPMA) 2015	Competencia que incluye el intercambio de información adecuada, entregada con precisión y coherencia a todos los miembros del equipo y a los <i>stakeholders</i> . Los aspectos esenciales de una comunicación efectiva son el contenido y los medios de comunicación (tono de voz, canal y cantidad de información), los cuales deben ser claros y apropiados.
Cohen, E. (2017)	Competencia que exige estrategias para mantener una comunicación efectiva con el equipo del proyecto y los <i>stakeholders</i> .
Project Management Institute (2017)	Es un intercambio oportuno, efectivo, apropiado y preciso de información relevante con los <i>stakeholders</i> usando métodos adecuados.
	Intercambio intencionado o involuntario de información, que puede ser en forma de ideas, instrucciones o emociones. Los mecanismos mediante los cuales se intercambia información pueden ser escritos, orales, formales o informales, expresivos o no verbales.

Fuente: Constructo de liderazgo en Gerencia de Proyectos | Beltrán Sánchez - Salas Ortiz - Vega Vargas | Revista IDGIP | Volumen 1 N.º 3 Enero-diciembre de 2020 | 6-20

En las empresas/organizaciones existe la comunicación interna. El esquema presentado a continuación hecho por Lucas Martín, permite visualizar con claridad los distintos tipos de comunicación interna (Ver Tabla 3):

Tabla 3. Tipos de Comunicación Interna

	<i>Descendente</i>	<i>Horizontal</i>	<i>Ascendente</i>
Formal	<i>Con los subordinados</i>	<i>Con los colegas</i>	<i>Con los jefes</i>
Informal	<i>Con los seguidores</i>	<i>Con los amigos</i>	<i>Con los líderes</i>

Comunicación Formal: Es la encargada de transmitir los mensajes oficiales de la organización, respetando y siguiendo las líneas del organigrama, otorgando una visión clara de vías de la información.

Comunicación informal: surge de las relaciones sociales que se desarrollan entre los miembros de la organización y representa una parte importante de toda la

comunicación que se genera en las grandes organizaciones. Esta información que se genera y emite es de forma no oficial. La principal forma de comunicación informal es el *rumor*; este aparece cuando los canales de comunicación formal no proporcionan la suficiente información a los miembros de la organización sobre la vida y funcionamiento de la misma y recurren a otras fuentes para obtenerla.

Rumor: Según la teoría del rumor de Allport y Postman (1967), el rumor se caracteriza por aparecer, en situaciones críticas, a través de canales informales, la falta de posibilidad de verificación y el enganche psicológico de su contenido. Utiliza el canal informal de boca a oreja, de una comunicación oral y personal. Expresa y, al mismo tiempo, satisface emociones y necesidades del individuo.

Comunicación ascendente: suministra información a los niveles superiores respecto a lo que sucede en los niveles más bajos.

Comunicación descendente: va desde la alta dirección a los subordinados. Su objetivo es informar, dirigir, instruir y evaluar a los subordinados, propiciar a los integrantes información sobre metas y políticas organizacionales.

Comunicación horizontal: tiene lugar entre los miembros de los grupos de trabajo, entre grupos de trabajo. Permite a los miembros de la organización establecer relaciones con sus colegas (parte importante de la satisfacción del empleado).

RESOLUCIÓN DE CONFLICTOS.

Costa y Sessa (2016) mencionan que resolver un conflicto en las organizaciones se requiere de ambas partes implicadas para llegar a un acuerdo o consenso. Un conflicto no es ni bueno ni malo, esto dependerá de la manera cómo será gestionado dentro de la organización (Toro, 2017, p. 18). Además, se presenta por diversos factores: costumbres, creencias, diversidad de gustos y pensamientos dependiendo de ello es que se brinda una solución.

Las características de la resolución de un conflicto están relacionadas a las etapas del conflicto, y a los factores que dan lugar al conflicto en el ámbito laboral y finalmente pero no menos importante los estilos de resolución o manejo de conflictos.



En la Tabla 4 se muestran algunas de las definiciones de manejo y resolución de conflictos y problemas.

Tabla 4. Definiciones de manejo y resolución de conflictos y problemas por autores y estándares.

Autor	Definición
Verma, V.K. (1996)	Capacidad de comprender el conflicto y diagnosticarlo correctamente. Posteriormente, se deben aplicar estrategias tales como la comunicación y la negociación; enseguida, generar confianza y respeto entre las partes en conflicto y, finalmente, desarrollar una estrategia de gana-gana, con el fin de resolver el conflicto.
Association for Project Management (APM) (2006)	Capacidad de identificar y abordar las diferencias que, si no se administran, afectarían los objetivos del proyecto. Busca evitar que las diferencias se conviertan en factores destructivos en un proyecto.
International Project Management Association (IPMA) (2015)	Capacidad de moderar o dar solución a los conflictos y crisis que puedan presentarse en el entorno del proyecto. Se deben propiciar estos escenarios de manera apropiada y estimular un proceso de aprendizaje para futuros conflictos y crisis. Su propósito es permitir que el gerente de proyectos tome medidas efectivas cuando se produce una crisis o choque de intereses opuestos/personalidades incompatibles, entre otras.
Project Management Institute (2017)	Capacidad de resolver los conflictos de manera oportuna y constructiva con el fin de lograr un equipo de alto desempeño. Existen cinco técnicas generales de resolución de conflictos: retirarse/eludir, suavizar/adaptarse, consensuar/conciliar, forzar/dirigir y colaborar/resolver.

Fuente: Constructo de liderazgo en Gerencia de Proyectos | Beltrán Sánchez - Salas Ortiz - Vega Vargas | Revista IDGIP | Volumen 1 N.º 3 Enero-diciembre de 2020 | 6-20

Etapas del Conflicto

El proceso del conflicto comprende 5 etapas:

1. **Incompatibilidad:** indica la personalidad del conflicto, las causas (son la comunicación inadecuada que provoca malos entendidos), desacuerdos y conductas dadas en la situación, la estructura del grupo de trabajo y esto comprende a los estilos de liderazgo, los sistemas de recompensa que benefician a sólo ciertos miembros y el grado de dependencia que generará que algunos miembros busquen la interdependencia a costas de los esfuerzos de otro; por otro lado, tenemos las variables personales que comprenden idiosincrasia, personalidad, valores, defectos, etc.
2. **Conocimiento y personalización:** busca reconocer la clase de conflicto que se está presentando, ya sea percibido objetivamente por ambas partes o sea un conflicto sentido en el cual se ven integrados sentimientos que causan la frustración.
3. **Intenciones:** las percepciones y emociones, así como también su conducta honesta. En esta etapa se desarrolla el conocido modelo teórico de estilos de manejo de conflicto de Thomas y Kilmann, el cual es un modelo bidimensional que se basa en detectar la capacidad del ser humano en tratar de satisfacer sus propios intereses (*asertividad*) o tratar de satisfacer los intereses de los demás (*cooperación*), la combinación de estas dos dimensiones y sus variantes surgen las 5 intenciones de manejar el conflicto:
 - 1) **Estilo competidor:** busca satisfacer sus propias preocupaciones sin importarle el impacto que estas tengan en la otra parte; el estilo colaborador refiere que ambas partes del conflicto desean satisfacer de forma plena y completa los intereses de todas las partes.

- 2) **Estilo comprometido:** es en el que ambas partes ceden en algún punto con el objetivo de que no haya ni ganador ni perdedor.
 - 3) **Estilo acomodador:** es aquel en el que una de las partes está dispuesto a satisfacer los intereses de la otra parte por encima de los suyos.
 - 4) **Estilo evitativo:** refiere que una de las partes se retira del conflicto evadiendo sus responsabilidades y haciendo como si no se hubiera dado ningún conflicto.
4. **Comportamiento:** en este se desarrollan actos, afirmaciones o reacciones por parte de los involucrados, aquí es donde el conflicto se intensifica o se atenúa.
 5. **Resultados:** se divide en resultados funcionales y disfuncionales, el primero ha construido un aprendizaje y mejora la calidad de relaciones, funciones, procesos, rendimiento, etc. dentro de la organización, y el resultado disfuncional solo alienta el descontento, destroza vínculos y la cohesión del grupo de trabajo.

Modelos de Manejo de Conflictos

Negociación: este es un modelo o estrategia en la cual las partes involucradas acuerdan un intercambio, en esta estrategia los negociadores abordan múltiples soluciones a la problemática para luego establecer pautas que los hagan llegar a un acuerdo.

Mediación: es un proceso determinado como una negociación asistida, debido a que será guiada por un tercero en cuestión que busca la solución y satisfacción de las partes a partir del intercambio de información, ideas y control; este tercero, al no tener nada que ver con la problemática es neutral y actúa de forma asertiva objetiva, sugerente y disuasivo.

Conciliación: busca compatibilizar y equiparar las oportunidades de ambas partes en pocas palabras eliminar la desigualdad, manteniendo el equilibrio por medio de concertaciones o convenios.



[Liderazgo, Motivación, Comunicación, Trabajo En Equipo](#)



[Liderazgo en tiempos del coronavirus](#)

MODELO DE MADUREZ DE CAPACIDAD DE GESTIÓN PERSONAL (P-CMM). NIVELES DE P-CMM.

Toda organización debe tender a una gestión de Recursos Humanos de carácter:

- Especializado y actualizado.
- Con procesos de mejoramiento continuo.
- Buscando formular, establecer y aplicar procesos gerenciales.



Software Engineering Institute

Los modelos de madurez de la capacidad (CMM por sus siglas en inglés) del SEI (Software Engineering Institute) son un grupo de estrategias que ayudan a mejorar a las organizaciones, descubriendo el verdadero valor que pueden ofrecer mediante la construcción de capacidad en sus personas y procesos.



Según el Diccionario, a continuación se conceptualiza cada uno de los términos del siguiente tema Modelo P-CMM:

MODELO: punto de referencia para imitarlo o reproducirlo. *Un modelo dice QUE hacer.*

MADUREZ: condición o estado maduro. Maduro: que ha alcanzado un estado de desarrollo adecuado para su utilización, funcionamiento o empleo.

CAPACIDAD: Oportunidad, lugar o medio para ejecutar algo.

El SEI ha desarrollado los siguientes modelos:

1. Modelo de Madurez de capacidad para el software (SW-CMM),
2. Modelo de Madurez de Capacidad para Recursos Humanos (P-CMM),
3. Modelo de Madurez de Capacidad para la ingeniería de sistemas (SE-CMM),
4. Modelo de Madurez de Capacidad para el desarrollo de productos integrados (IPD-CMM),
5. Modelo de Madurez de Capacidad para la adquisición de software (SA-CMM).

P-CMM es un modelo que se puede representar como una **hoja de ruta** para la **mejora continua de una organización**. Puede considerarse como una brújula que orienta y pone orden en las actividades de gestión de las personas, incluyendo prácticas vanguardistas de gestión del cambio organizativo para ayudar a las organizaciones a mejorar la capacidad de su equipo humano y la efectividad de la organización. Ofrece también un **marco de mejora de la madurez que las organizaciones pueden utilizar gestionando y mejorando sus acciones para atraer, motivar y retener al personal mejor calificado**. Las prácticas incluidas en el modelo ayudan a una organización a convertirse en un empleador deseado y a asegurar que el personal posee las competencias necesarias para conseguir los objetivos de negocio actuales y futuros de la organización.

MÉNDEZ, Instructor Autorizado y Asesor-Líder para el People CMM por Carnegie Mellon University–SEI, define el P-CMM como una guía estratégico-táctica para desarrollar una cultura de gestión de las personas.

La finalidad del modelo P-CMM es:

- Mejorar continuamente las capacidades individuales.
- Conformar y desarrollar a equipos de trabajo eficaces.
- Motivar la mejora en el rendimiento.
- Formar al recurso humano que se necesita dentro de la organización para lograr alcanzar las metas y formular planes para encarar proyectos en el futuro.

ESTRUCTURA

El P-CMM tiene una estructura escalonada que se muestra en la figura 1:



Figura 1 Estructura del Modelo P-CMM – Traducción de People Capability Maturity Model (P-CMM) Version 2.0, Second Edition

ORGANIZACIÓN DEL MODELO

- **Niveles de madurez:** Un nivel de madurez representa un nuevo nivel de la capacidad organizativa creada por la transformación de uno o más dominios de procesos de una organización. Provee el establecimiento de mejoras continuas, con herramientas poderosas para el desarrollo de la capacidad de su personal.
- **Áreas de procesos:** Grupo de prácticas relacionadas, que cuando están realizadas colectivamente, satisfacen un sistema de metas que contribuyen a alcanzar un nivel de madurez. Las áreas de procesos constituyen un importante proceso organizacional y en cada nivel de madurez crean un sistema de interrelación de procesos que transforma la capacidad de la organización para una gestión de la mano de obra. En ellas se identifican las prácticas que deben ser institucionalizadas para alcanzar un nivel de madurez y describen las prácticas que una organización debe implementar para mejorar la capacidad de su mano de obra. Cada área

contiene de 3 a 5 metas que declaran los objetivos que fueron diseñados para ser cumplidos.

- **Metas:** Estado organizacional para ser alcanzado por la implementación de prácticas de un área de procesos. Cada Área de proceso contiene de tres a cinco metas, en conjunto indican el alcance, los límites e intención de un área de proceso, cada una de las metas solo se aplica en una sola de estas áreas y son características únicas que describen qué debe ser implementado para satisfacer el propósito de éstas. Si las metas de todas las áreas de procesos incluidas en un nivel de madurez están satisfechas, la organización habrá alcanzado y establecido uno nuevo nivel de la capacidad de la gestión de la mano de obra. Son usadas como línea base para determinar la satisfacción de las áreas de procesos y de la madurez de los procesos organizacionales.
- **Prácticas:** para referirse a procesos estándares que debería realizar el recurso humano. Se clasifican en:
 - **prácticas de implementación:** describen a aquellas prácticas que deben estar implementadas para alcanzar las metas correspondientes. Se agrupan en *Prácticas de Desempeño*, las mismas son una amplia categoría de prácticas que describe la implementación de la actual área de proceso.
 - **prácticas de institucionalización:** son prácticas que ayudan a institucionalizar las prácticas de implementación en la cultura de la organización para que éstas sean efectivas, repetibles y duraderas. Estas se organizan en 4 categorías de prácticas:
 - 1.- Compromiso a realizar: es el compromiso para llevar a cabo las actividades, normalmente implica el establecimiento de políticas de organización (para establecer expectativas de desempeño) y se les asignará la responsabilidad de asesorar y coordinar la aplicación de las prácticas de trabajo.
 - 2.- Habilidad de desempeño: describe las condiciones previas que deben existir en la unidad u organización para ejecutar prácticas competentes. Implica recursos, estructuras organizacionales y preparación para desempeñar las prácticas del área de proceso.
 - 3.- Medición y Análisis: implica medir el estado de las prácticas realizadas, agregar algunas medidas de la unidad al nivel de organización, y evaluar la eficacia de las prácticas realizadas.
 4. Verificación de la Implementación: implica asegurar que las prácticas se están realizando en el cumplimiento de las políticas, valores declarados, los planes, leyes y reglamentos, y que la dirección ejecutiva mantiene conciencia del nivel de cumplimiento.

Relación de los componentes

Por cada **área de proceso** existe **una sola meta** para las **Prácticas de Institucionalización**, por otro lado, por cada **área de proceso** existe **más de una meta** para las **Prácticas de Implementación**.

En las siguientes figuras 2 y 3 se muestra gráficamente la organización y relación de componentes del modelo.

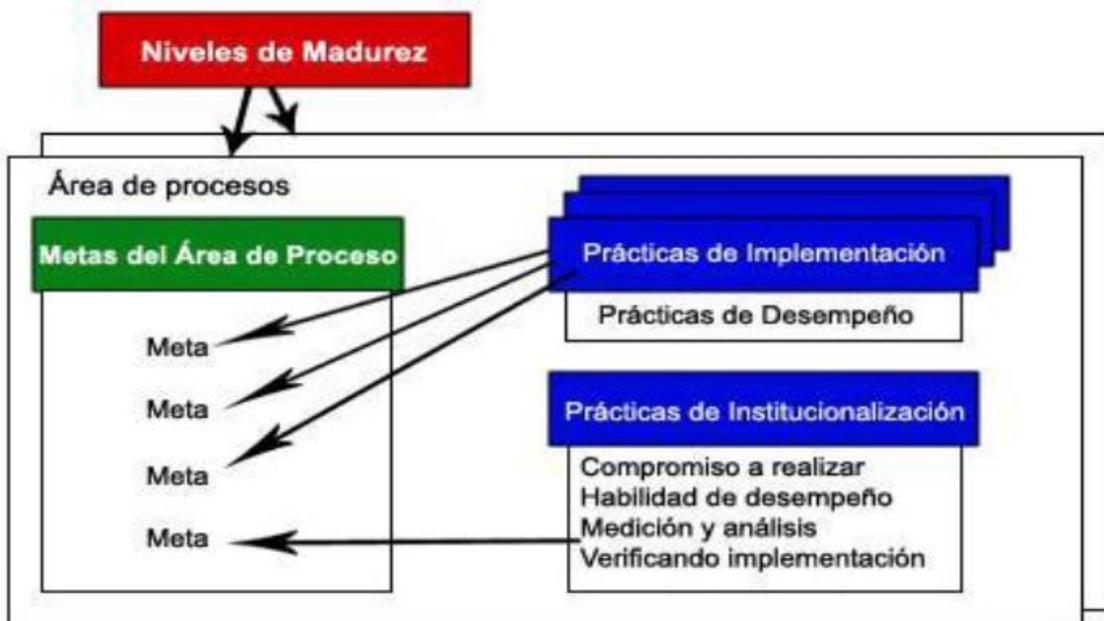


Figura 2. Organización del Modelo



Figura 3. Organización del Modelo. Fuente: Software Process Improvement

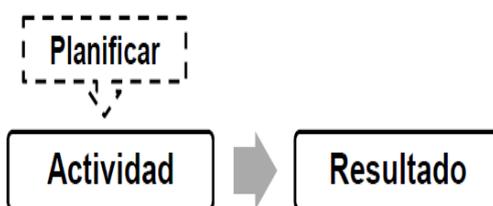
DESCRIPCION DE LOS NIVELES DE MADUREZ.

Nivel 1 – Nivel Inicial: Las organizaciones en el nivel inicial exhiben típicamente cuatro características:

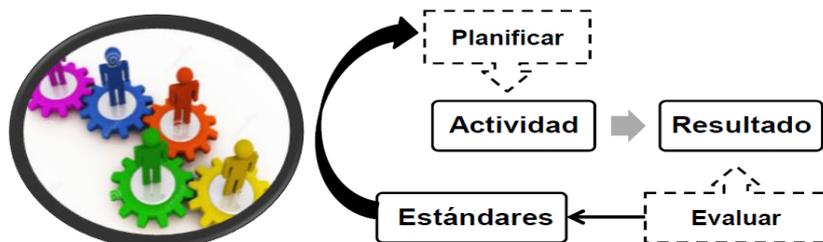
1. Inconsistencia en la ejecución de prácticas.
2. Dislocación de la responsabilidad.
3. Prácticas ritualistas.
4. Una mano de obra emocionalmente separada.



Nivel 2 – Gestionado: La organización establece una cultura concentrada a nivel de unidad para garantizar que las personas sean capaces de conocer sus compromisos de trabajo. Alcanzando el nivel 2, la organización **desarrolla las capacidades para gestionar las habilidades y el desempeño a nivel de unidad.**



Nivel 3 – Definido: La organización desarrolla los **planes estratégicos** según las necesidades del personal para lograr los objetivos de negocios actuales y futuros. Las prácticas implementadas en el nivel 2 son ajustadas para motivar y apoyar el desarrollo de habilidades del personal. Las habilidades de proceso definidas se utilizan para adaptar los procesos definidos y establecer los roles que proporcionan el siguiente paso en el desarrollo del grupo de trabajo. Se establece una **cultura de participación** que permite el uso más eficaz del talento de la organización para tomar decisiones y ejecutar el trabajo.



Nivel 4 – Predecible: se enfoca en el aprovechamiento del conocimiento y de la experiencia del marco de la mano de obra desarrollado en el nivel 3. Los **procesos basados en competencia** usados por diversas habilidades del personal se entretrejen para crear procesos integrados y multidisciplinarios. Los individuos y los grupos de trabajo **manejan cuantitativamente** los procesos basados en competencia que son importantes para alcanzar sus objetivos de desempeño. La organización maneja la capacitación del personal. El efecto de las capacitaciones en el trabajo del personal se evalúa y se toman medidas correctivas en caso de necesidad. Los mentores usan la infraestructura proporcionada por las competencias del personal de la organización para ayudar a las personas y los grupos de trabajo en el desarrollo de su capacidad.



Nivel 5 – Optimizado: La organización **evalúa y mejora la alineación del funcionamiento entre sus individuos, grupos de trabajo, y unidades con los objetivos de negocio** de la organización. La organización **evalúa continuamente** las oportunidades para mejorar sus prácticas del personal con ajustes incrementales o adaptándole a la mano de obra prácticas innovadoras y de tecnología.



El P-CMM se centra sus cuatro niveles de madurez en cuatro ejes temáticos sobre los que desarrollan las correspondientes categorías de procesos:

1. Desarrollo de las capacidades
2. Creación de equipos y cultura
3. Gestión y motivación del desarrollo
4. Estructuración de la capacidad de trabajo.

Dentro de cada eje temático se encuentran las 22 áreas procesos, ver Figura 4.

	Desarrollo de las capacidades individuales	Construcción de grupos de trabajo y cultura	Motivación y gestión del desempeño	Forma de la fuerza de trabajo
5 Optimizado	Mejora continua de la capacidad		Alineación del desempeño organizacional	Innovación continua de la fuerza de trabajo
4 Predecible	Activos basados en competencias Asesoría (<i>mentoring</i>)	Integración de competencias Delegación de responsabilidad	Gestión cuantitativa del desempeño	Gestión de la capacidad organizacional
3 Definido	Desarrollo de competencias Análisis de competencias	Desarrollo de grupos de trabajo Cultura participativa	Prácticas basadas en competencias Desarrollo de carrera	Planeación de la fuerza de trabajo
2 Gestionado	Capacitación y desarrollo	Comunicación y coordinación	Compensación Gestión del desempeño Ambiente de trabajo	Recursos humanos
1 Inicial	Inconsistencia total			

Figura 4. Categorías y Niveles del P-CMM

EJEMPLO PARA REGISTRAR UNA REUNION DE EQUIPO DE TRABAJO

MINUTA DE REUNION

Fecha:	Lugar:
Duración:	
Proyecto:	
Equipo de Trabajo:	

Nombre	Rol	Firma

Objetivo / s	Estatus
	Alcanzado/ No alcanzado

Actividades Realizadas	Estatus
	Iniciado/En proceso/ Terminado

Actividades pendientes	Responsable	Fecha Compromiso

Acuerdos
<ul style="list-style-type: none"> • • • • •



RECURSOS HUMANOS 4.0

El concepto de RR.HH 4.0 está estrechamente relacionado con la llamada Industria 4.0, o la cuarta gran revolución industrial. Es una revolución en el área de recursos humanos que pretende automatizar sus procesos y enfocarse en la estrategia, dejando de lado las actividades manuales, burocráticas y repetitivas.

RR.HH 4.0 está esencialmente impregnado de tecnología y está en línea con los nuevos escenarios del mercado laboral, con especial influencia de la Generación Millennial, que ahora alcanza posiciones de liderazgo en las organizaciones, y otras que le han seguido.

En RRHH 4.0 se entiende que los profesionales de hoy tienen otras necesidades, y muchos de ellos reciben ofertas de trabajo de otras compañías con mucha más frecuencia que hace 10 años. Por eso, es necesario trabajar por la mejor experiencia, haciendo el ambiente de trabajo más atractivo, menos aburrido – para las generaciones que nacieron inmersas en la tecnología, es sumamente importante que esté presente en su día a día.

Con la tecnología democratizada de la computación en la nube ya no es necesario invertir mucho dinero en infraestructura de TI. Prácticamente todo lo que en RR.HH necesita en términos de tecnologías (software, servidores, etc.) se puede adquirir y utilizar virtualmente.

La tecnología digital abarata las inversiones, da más movilidad a los profesionales del área (que pueden trabajar desde cualquier lugar, en cualquier momento y utilizando cualquier dispositivo conectado), es fácil de implementar y mantener, entre otras ventajas.

Fuente: RR.HH 4.0: cómo la tecnología está transformando la gestión de personas
<https://www.mjvinnovation.com/es/blog/rr-hh-4-0/>

La siguiente charla de Nasser Mohamed demuestra que la industria 4.0 no representa una amenaza para los profesionales, sino un reto en la transformación de sus habilidades, lo cual también implica nuevos esquemas de trabajo que las empresas deben empezar a asimilar y acoger como parte de su ADN para una correcta gestión del talento humano.



[Industria 4.0 "Cambiando esquemas y creando nuevas oportunidades" | Nasser Mohamed | TEDxUANL](#)

Bibliografía Consultada

Álvarez, F. J., Alegría, J. A. H., Arellano, M. M., Arteaga, J. M., Amador, C. E. V., & Bieliukas, Y. C. H. (2019). Gestión de proyectos de software. Repositorio INSTITUTO SUPERIOR TECNOLÓGICO JAPÓN (Biblioteca) Colección: INSTITUTO SUPERIOR TECNOLÓGICO JAPÓN Online: <http://190.57.147.202:90/xmlui/handle/123456789/421>

Allport, G. W. y Postman, L. (1967). Psicología del rumor. Editorial Psique, Buenos Aires.

Bill Curtis, Bill Hefley, Sally Miller (2009) People Capability Maturity Model (P-CMM) Version 2.0, Second Edition (2010)

Cornejo Silva, A. T., & Meléndez Arangurí, R. D. R. (2021). Factores relacionados a la resolución de conflictos en las organizaciones: una revisión sistemática.

Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

Pérez, O. (2009). Modelo de Madurez de la Capacidad de las Personas. Tesis de Ingeniería, Universidad de las Ciencias Informáticas, La Habana, Cuba.

Rubio, R. A. L., & Culquichicón, J. I. L. (2019). RUIDO Y RUMOR EN LA COMUNICACIÓN. HELIOS, 3(1).

Lectura recomendada

Bertoldi, Carolina Alejandra (2020), Trabajo de Investigación: “Reuniones efectivas”, Licenciatura en Administración (UNCUYO) Online: https://bdigital.uncu.edu.ar/objetos_digitales/15720/ti-reuniones-efectivas-trabajo-investigacion.pdf

PLANIFICACION DE PROYECTOS DE SOFTWARE



Según el diccionario:

Planificación: Plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado, tal como el desarrollo de software, el desarrollo económico, la investigación científica, el funcionamiento de una industria, etc.

Proyecto: Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería.

Todo proyecto de ingeniería debe partir de un buen plan, es muy importante tener presente que, la planificación es necesaria por la existencia de incertidumbre en el entorno del desarrollo de software y debe enfocar su atención en las metas del proyecto.

PLANIFICACIÓN ÁGIL VS PLANIFICACIÓN TRADICIONAL

A continuación se muestran las especificidades de cada uno de los planteamientos de planificación mencionados (Figura 5, 6 y 7).

Planificación tradicional

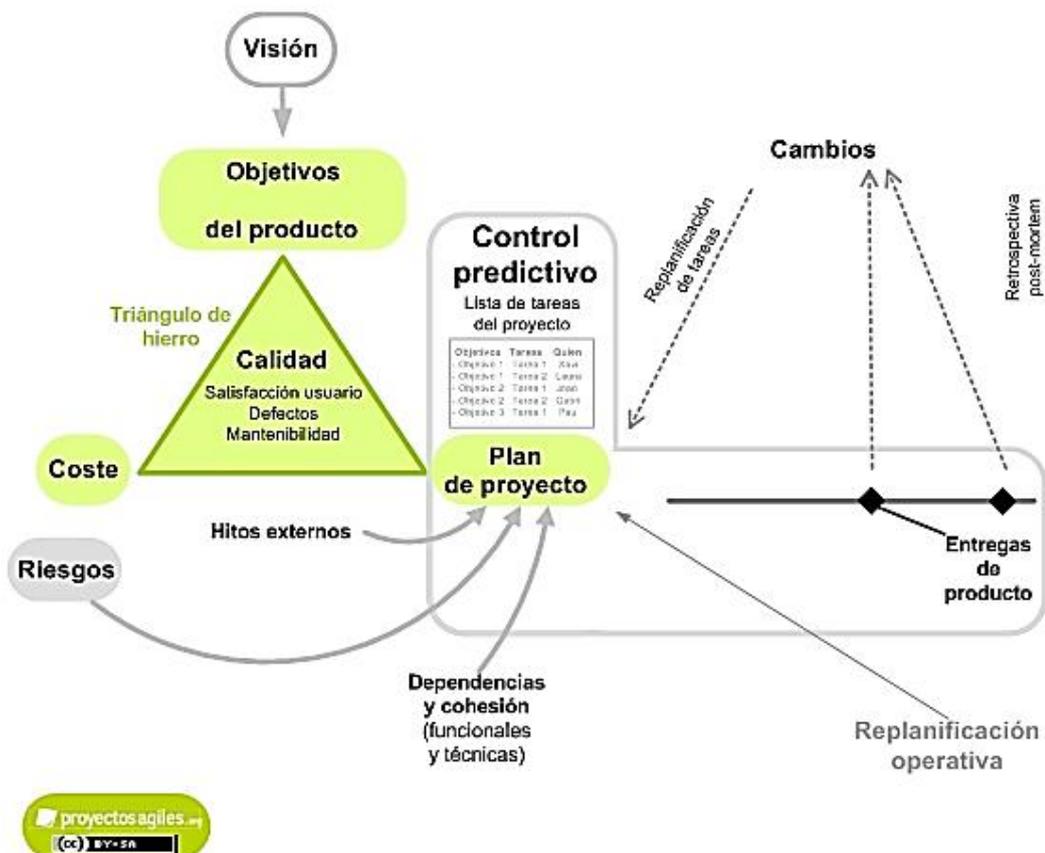
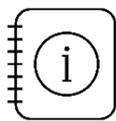


Figura 5. Planificación Tradicional. Fuente:

<https://proyectosagiles.org/2010/12/15/planificacion-agil-vs-planificacion-tradicional/>

La planificación tradicional toma como base el control predictivo de un PROYECTO, con lo que:

- Está basada en la identificación inicial de las TAREAS necesarias para elaborar el producto (EDT(estructura de desglose del trabajo) en inglés WBS(Work Breakdown Structure)), planteamiento que se va modificando (replanificando) según el devenir de acontecimientos durante el proyecto.
- Realiza pocas entregas de producto durante el proyecto (normalmente realiza una única entrega en su finalización), con lo que el feedback que se genera es tardío y, dado que se ha construido mucho producto sin haber verificado su adecuación, los cambios que sean necesarios (si grandes y caros) pueden comprometer los plazos y el presupuesto del proyecto.
- A lo sumo realiza una única retrospectiva (post-mortem) al finalizar el proyecto, con lo que las lecciones aprendidas ya no son aplicables en el propio proyecto.



Una **estructura de desglose del trabajo (EDT)**, también conocida por su nombre en inglés **Work Breakdown Structure o WBS**, es una herramienta fundamental que consiste en la descomposición jerárquica, orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de este y crear los entregables requeridos, donde cada nivel descendente de la EDT representa una definición con un detalle incrementado del trabajo del proyecto.

(https://es.wikipedia.org/wiki/Estructura_de_descomposici%C3%B3n_del_trabajo)

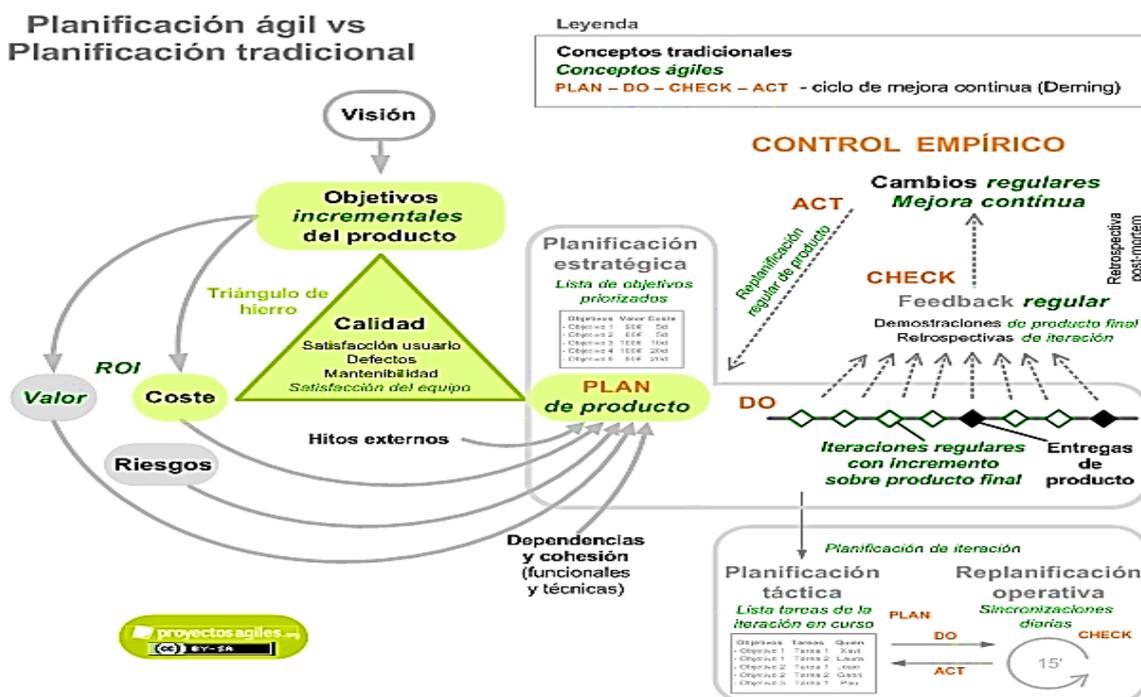


Figura 6. Planificación Ágil. Fuente:

<https://proyectosagiles.org/2010/12/15/planificacion-agil-vs-planificacion-tradicional/>

Planificación ágil vs Planificación tradicional

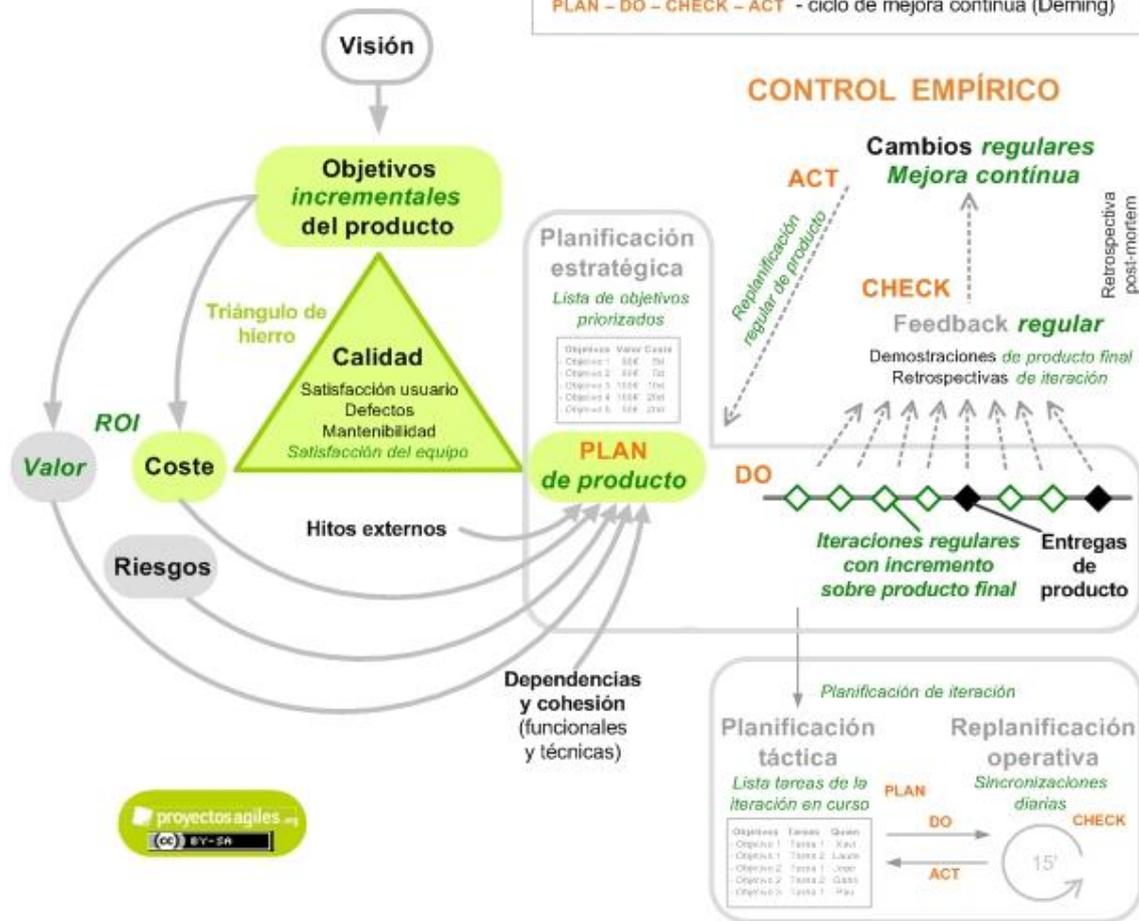


Figura 7. Planificación Ágil vs Planificación Tradicional. Fuente:

<https://proyectosagiles.org/2010/12/15/planificacion-agil-vs-planificacion-tradicional/>

La planificación ágil toma como base el control empírico de la construcción de producto (inspección y adaptación), por lo cual:

- Plantea fases basadas en OBJETIVOS DE PRODUCTO (desarrollo iterativo e incremental) priorizados balanceando beneficios de negocio respecto a sus costes de desarrollo.
- Realiza fases de trabajo muy intenso (de 2 a 4 semanas) con demostraciones al cliente de ese incremento de producto, de manera que se facilita el realizar cambios que consigan el máximo el alineamiento con sus expectativas y se cree un espacio natural para la replanificación estratégica de los objetivos todavía no abordados.
- Dispone de varios niveles de planificación, dado que asume un horizonte de incertidumbre a partir del cual no tiene sentido planificar tareas detalladas:
- Nivel de estratégico: planificación de objetivos de producto (Product Backlog).
- Nivel táctico: planificación de tareas para la iteración en curso, en la reunión de planificación de iteración.

- Nivel operativo: replanificaciones diarias de las tareas de la iteración, como consecuencia de las reuniones diarias de sincronización.
- Realiza retrospectivas durante todo el proyecto, de manera que se mejore la productividad y calidad dentro del propio proyecto.
- Hace partícipe al equipo en el proceso, tanto en la planificación (de proyecto e iteraciones) como en la mejora del procedimiento de trabajo (retrospectivas), añadiendo como parámetro de calidad la satisfacción del equipo, que se consigue mediante su participación activa, de manera que su implicación y motivación revierta en el resultado del proyecto.

EL PROBLEMA EN LA GESTIÓN DEL PROYECTO: VISIÓN GLOBAL PARA EL DESARROLLO DE UN PROYECTO DE SOFTWARE

Los problemas más comunes que enfrentan hoy en día los proyectos de desarrollo de software, tales como: retrasos en la entrega del producto final, aumento de los costos de desarrollo y mantenimiento y escasa calidad del software, se deben principalmente a una mala o escasa planificación.

Las principales causas de las fallas en la planificación de proyectos de Desarrollo de Software son las siguientes:

- Inadecuada definición del proyecto.
- Comprensión errónea del problema.
- Desconocimiento o inexperiencia de cómo planificar.
- Incumplimiento del ciclo de planificación.
- Escasa negociación de compromisos con el usuario al inicio del proyecto
- Definición incompleta de los requerimientos.
- Estimaciones optimistas.
- Supuestos y restricciones del proyecto inválidos o no verificados.
- Aplicación errónea o no-utilización de la información histórica de la organización.
- Mala administración del proyecto.
- Fallas en el uso de los planes.
- Carencia de control de cambios.
- Escasa motivación.
- Estilo erróneo de liderazgo.
- Carencia de control y gestión.
- Organización errónea del grupo de trabajo.

INTRODUCCIÓN A: ÁMBITO DEL SOFTWARE. ESTIMACIÓN DEL PROYECTO. PLANIFICACIÓN DE UN PROYECTO DE SOFTWARE. EQUIPO DE UN PROYECTO DE SOFTWARE, ROLES.

Ámbito del software

La primera actividad de gestión de un proyecto de software es determinar el ámbito del software. El ámbito se define respondiendo a las siguientes cuestiones:

1. **Contexto.** ¿Cómo encaja el software a construir en un sistema, producto o contexto de negocios mayor y qué limitaciones se imponen como resultado del contexto.
2. **Objetivos de información.** ¿Qué objetos de datos visibles al cliente se obtienen del software? ¿Qué objetos de datos son requeridos de entrada?
3. **Función y rendimiento.** ¿Qué función realiza el software para transformar la información de entrada en una salida? ¿Hay características de rendimiento especiales que abordar?



SCCT (Sistema de clasificación de cinta transportadora)

SCCT clasifica las cajas que se mueven por una cinta transportadora (la cinta se mueve a 5 pies por minuto). Cada caja está identificada por un código de barras (nº de pieza) y se clasifica en uno de los 6 compartimentos que hay al final de la cinta. Las cajas pasan en orden aleatorio y están espaciadas entre sí uniformemente.

Existe una estación de clasificación (un PC y un lector de código de barras), por donde pasan las cajas, que conectado a un mecanismo de maniobra, clasifica las cajas en los compartimentos.

El software del SCCT deberá recibir información de un lector de código de barras a intervalos de tiempo en función de la velocidad de la cinta. El código será decodificado al formato de identificación de caja. Se realizará una inspección en la base de datos para determinar el compartimento al que irá a parar la caja que este en ese momento en el lector. La posición correcta del compartimento se pasará al mecanismo de maniobra que sitúa la caja en el lugar adecuado. Se recibirá la entrada de un tacómetro de pulsos para sincronizar la señal de control del mecanismo de maniobra.

El *planificador del proyecto* examina la especificación del ámbito y se extraen las funciones importantes:

- Lectura del código de barras.
- Lectura del tacómetro.
- Decodificación de los datos.
- Búsqueda en la BD.
- Determinar la posición del compartimento.
- Generar la señal de control para el mecanismo de maniobra

Estimación Del Proyecto

La segunda tarea en la planificación, es la estimación de los recursos requeridos para de desarrollo software.

- Personas, recursos humanos
- Componentes de software reutilizables (componentes ya desarrollados o experimentados), que reducen el coste de desarrollo y aceleran la entrega.
- Componentes nuevos
- Recursos de ingeniería de software y de hardware.

La estimación del costo y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas las variables humanas, técnicas, de entorno, políticas que pueden afectar al costo final del software y al esfuerzo aplicado para desarrollarlo. Sin embargo, la estimación del proyecto de software puede dejar de ser un oscuro arte para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable.

Para realizar estimaciones seguras de costos y esfuerzos tenemos varias opciones posibles:

- Utilizar “técnicas de descomposición” relativamente sencillas para generar las estimaciones de costo y de esfuerzo del proyecto.
- Desarrollar un modelo empírico para el cálculo de costos y esfuerzos.
- Adquirir una o varias herramientas automáticas de estimación.



Dejar las estimaciones para más adelante o retrasarlas no es una opción ya que estas se necesitan de antemano.

Las tres opciones restantes son métodos viables para la estimación del proyecto.

1. Las *técnicas de descomposición* utilizan un enfoque divide y vencerás.
2. Los modelos empíricos son utilizables como complemento de las técnicas de descomposición donde cada modelo se basa en la experiencia (*datos históricos*),
3. Las *herramientas automáticas de estimación* ponen en ejecución una o varias técnicas de descomposición o modelos empíricos.

Se profundizan las estimaciones, más adelante en el tema [GESTION DE PROYECTOS DE SOFTWARE](#) pág. 44.

Planificación De Un Proyecto De Software

El principal objetivo de la planificación en proyectos de desarrollo de software es ordenar el qué hacer durante el proyecto y asignar adecuadamente los recursos y tareas para cumplir los objetivos propuestos.

En general se planifica para:

- Organizar el qué hacer del proceso de desarrollo de software.
- Minimizar tiempo y costos involucrados.
- Maximizar el uso de recursos disponibles.
- Establecer hitos del proyecto.

- Medir el avance.
- Mejorar la comunicación.
- Obtener soporte técnico, de gerencia y político.

La planificación es una tarea que se desarrolla al inicio del proyecto pero rige el resto de las fases. Una buena planificación inicial ayudará a que las metas propuestas se cumplan y que los eventuales inconvenientes sean abordados de mejor forma.

Todas las organizaciones planifican, pero por lo general no se realiza de la manera adecuada, muchas veces la planificación se realiza de manera informal cuando debiera ser formal. La planificación formal es aquella que es:

- Documentada.
- Uniforme y regularmente aplicada.
- Con resultados concretos, distribuidos, entendidos y comprometidos por la organización.

En una planificación formal deben quedar claramente identificados los planes, procedimientos, la organización, la asignación de recursos y las responsabilidades:

- **Planificación Temporal:** en esta etapa se debe indicar cuando comienza y cuando termina cada etapa, es fundamental para controlar si el proyecto se está ejecutando a tiempo ó si está atrasado, si tenemos algo de holgura en el tiempo o en el presupuesto.
- **Planificación Personal:** en esta etapa se debe especificar la cantidad de personas que se necesitan para cada etapa del proyecto, la especialidad, el tiempo que le dedicarán al proyecto (hs/día, hs/semana, etc.) cada etapa puede requerir mayor o menor cantidad de personas y no todos las personas trabajan en todas las etapas del proyecto. Los gestores son quienes seleccionan el personal que va a trabajar en el proyecto.
- **Estructura del equipo de trabajo:** este es un aspecto muy importante donde se debe considerar el tipo de persona que se necesitan para que trabaje en el equipo.
- **Verificación y control de calidad:** para que se genere un producto de calidad es necesario verificar si los componentes del proyecto están cumpliendo con los requisitos establecidos. En el plan de trabajo se deberán establecer de forma específica los mecanismos de verificación y control de calidad que se utilizará en cada etapa.
- **Gestión de la configuración:** se debe especificar los mecanismos que se utilizarán para atender las necesidades y solicitudes de cambio de en cada uno de las etapas, solicitadas por el cliente. Es muy importante saber que partes del proyecto afectará y comunicarlos, ya que realizar cambios sin comunicar a las partes afectadas indirectamente por el cambio, pueden echar abajo todo el trabajo realizado.
- **Gestión de riesgo:** Todo proyecto conlleva un riesgo, en el plan se debe establecer que acciones se harán en caso de retraso o que se pierda parte del personal. En determinadas circunstancias se deberá decidir si se continua con el proyecto, ya que siempre existe la posibilidad de que el proyecto se salga de control y resulte más caro continuarlo que detenerlo.

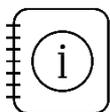
Equipo de un proyecto de software. Roles.

Para el **gerente de proyecto**, la meta es garantizar que el producto se desarrolla dentro de cierto marco temporal. Por tanto, monitorea el progreso del desarrollo y reconoce y reacciona ante los problemas. Esto se hace mediante la generación y el análisis de reportes acerca del estado del sistema de software y al realizar revisiones al sistema.

Las metas del **gerente de configuración** son garantizar que se sigan los procedimientos y políticas para crear, cambiar y probar el código, así como hacer accesible la información acerca del proyecto. Para implantar técnicas a fin de mantener el control sobre los cambios de código, este gerente introduce mecanismos para: realizar peticiones oficiales de cambios, evaluarlos (mediante un **Consejo de Control de Cambios** que sea responsable de aprobar los cambios al sistema de software) y autorizarlos. El gerente elabora y difunde la lista de tareas para los ingenieros y básicamente crea el contexto del proyecto. Además, recopila estadísticas acerca de los componentes que hay en el sistema de software, tales como la información que determina cuáles componentes del sistema son problemáticos.

Para los **ingenieros de software**, la meta es trabajar eficazmente. Esto significa que los ingenieros no deben interferir innecesariamente unos con otros en la creación y prueba del código y en la producción de productos operativos de apoyo. Pero, al mismo tiempo, deben intentar comunicarse y coordinarse de manera eficiente. Específicamente, los ingenieros usan herramientas que ayudan a construir un producto de software consistente. Se comunican y coordinan al notificarse unos con otros las tareas requeridas y las tareas completadas. Los cambios se propagan a través del trabajo mutuo mediante fusión de archivos. Existen mecanismos para garantizar que, para componentes que experimentan cambios simultáneos, hay alguna forma de resolver los conflictos y la fusión de cambios. Se conserva una historia de la evolución de todos los componentes del sistema, una bitácora con las razones de los cambios y un registro de lo que realmente cambió. Los ingenieros tienen su propio espacio de trabajo para crear, cambiar, poner a prueba e integrar código. En cierto punto, el código se convierte en una línea de referencia desde la cual continúan mayores desarrollos y se realizan variantes para otras máquinas objetivo.

El cliente usa el producto, sigue procedimientos formales para solicitar cambios y para indicar errores en el producto.



Como acotación histórica, cabe decir que una de las primeras organizaciones de equipo de software fue una estructura de paradigma cerrado originalmente llamado equipo de programador jefe. El núcleo del equipo estaba compuesto de: un ingeniero ejecutivo (el programador jefe), quien planeaba, coordinaba y revisaba todas las actividades técnicas del equipo; personal técnico (por lo general de dos a cinco personas), quienes realizaban análisis y desarrollaban actividades; y un ingeniero de respaldo, quien apoyaba al ingeniero ejecutivo en sus actividades y podía sustituirlo con mínima pérdida en la continuidad del proyecto. El programador jefe puede auxiliarse con uno o más especialistas (por ejemplo, experto en telecomunicaciones, diseñador de bases de datos), personal de apoyo (por ejemplo, escritores técnicos, oficinistas) y un bibliotecario de software.

En la siguiente tabla se muestra la responsabilidad de cada rol en un proyecto de software:

Tabla 5. Roles y Responsabilidades

Puesto	Responsabilidades
Jefe de Proyecto	Asignar los recursos, gestionar las prioridades, coordinar las interacciones con los clientes y usuarios, mantener al equipo del proyecto enfocado en los objetivos. También establecer un conjunto de prácticas que aseguran la integridad y calidad del proyecto. Supervisará el establecimiento de la arquitectura del sistema, la gestión de riesgos y la planificación y control del proyecto.
Analista de Sistemas	Capturar, especificar y validar requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaborar el Modelo de Análisis y Diseño. Colaborar en la elaboración de las pruebas funcionales y el modelo de datos.
Analista - Programador	Construir prototipos. Colaborar en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaborar el modelo de datos, preparar las pruebas funcionales, elaborar la documentación. Elaborar modelos de implementación y despliegue.

Fuente: Los participantes en un Proyecto de Software

http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro22/1411_los_participantes_en_un_proyecto_de_software.html



No todos los roles y responsabilidades residen dentro de los profesionales del software. Clientes, usuarios y otros participantes también tienen responsabilidades.

Bibliografía Consultada

Bedini González, Alejandro (2010), gestión de Proyectos –Ebook

Pressman, R. (2002). Ingeniería del Software. Un enfoque práctico. Ed. McGraw-Hill

Lecturas Recomendadas

Asociación Española de Derecho del Entretenimiento con la participación de Organización Mundial de la Propiedad Intelectual (2018) Guía para la redacción y negociación de contratos de software. Online: <https://www.wipo.int/export/sites/www/amc/en/docs/denaeguia2018.pdf>

Augspach et al (2021) Desarrollar roles y habilidades para una organización 4.0. Revista Énfasis Sudamericana. Online: <https://www.austral.edu.ar/industria/simulacion-en-tiempos-de-4-0-uso-actual-y-perspectiva-a-futuro/>

PLANIFICACION TEMPORAL Y SEGUIMIENTO DEL PROYECTO

Un PROYECTO es un esfuerzo **único y temporal** que se lleva a cabo para crear un resultado específico (Objetivos) para contribuir a solucionar una necesidad de mejora o de nuevos servicios (Propósito del proyecto).

- El proyecto es **único**, porque se emprende en cierto momento, con determinadas personas, bajo ciertas condiciones y atraviesa un conjunto de actividades para lograr el resultado. Si se desea repetir el proyecto, es probable que las condiciones, las personas y cualquier otro factor ya no sean igual y por ello van a existir pequeños cambios.
- El proyecto es **temporal**, porque debe realizarse en cierta fecha, con un plazo determinado y una fecha de finalización.

Para que un proyecto finalice con éxito es necesario determinar y gestionar de forma eficiente las diferentes etapas que lo componen, de tal manera que el cliente quede satisfecho con los resultados. Todo proyecto se encuentra limitado por los costes, el tiempo de realización y la calidad deseada. Estos tres objetivos están relacionados entre sí y nos obliga a optimizar los recursos disponibles:

- Recursos humanos y materiales
- Costes
- Tiempos

Por ello, es imprescindible la utilización de métodos de gestión de proyectos que nos ayuden a planificar las etapas del mismo de tal manera que, teniendo en cuenta las limitaciones impuestas, el resultado sea de total satisfacción para el cliente.

La gestión de proyectos y la planificación de la producción se puede controlar mediante diferentes métodos, entre ellos el PERT:

Métodos basados en la construcción de redes:

- PERT: "Program and Evaluation Review Technic", Técnicas de Evaluación y Control de Proyectos, fue utilizado por primera vez en Estados Unidos en 1958 en el proyecto Polaris.
- Potenciales: creado por Bernard Roy en 1960.

Métodos de ordenación y jerarquía:

- Taylor: aparece en 1880.
- Kanban: se utiliza por primera vez en Japón en 1960 y en Europa a partir de 1980.

Métodos mixtos:

- GANTT: desarrollado por Henry Gantt, es utilizado a partir de 1918. Representa cada tarea de un proyecto mediante una barra horizontal de longitud proporcional a su duración, mostrando las fechas de comienzo y finalización de las tareas.

Todos estos métodos tienen sus ventajas e inconvenientes dependiendo del proyecto que se desee gestionar. No son excluyentes. De hecho, en muchas ocasiones, se utilizan dos de ellos para facilitar la representación de los diferentes procesos.

RELACIÓN PERSONA-ESFUERZO.

El esfuerzo se refiere a la suma de los tiempos que le dedicarán los diferentes recursos a cierta actividad o al proyecto. Se mide en horas/hombre, días/hombre, semanas/hombre, etc. No importa que el trabajo se haga de forma secuencial por un solo recurso o en paralelo por diferentes personas. Se suman los tiempos de cada uno de ellos para obtener el esfuerzo total.

Cada participante del equipo de trabajo debería de estimar el esfuerzo de sus actividades, desglosando dichas actividades a un nivel de granularidad tal que las actividades tengan un esfuerzo menor a 2 o 3 días, y que incluso puede ser de sólo unas pocas horas.

La estimación del esfuerzo queda plasmada en el plan del proyecto, por ejemplo en el Diagrama de Gantt que se verá más adelante en el apartado Gráficos de tiempo para la planificación temporal.

DEFINICIÓN DE LAS TAREAS PARA EL PROYECTO SOFTWARE: SELECCIÓN DE LAS TAREAS DE INGENIERÍA DEL SOFTWARE.

Las actividades son las tareas que debe realizar el equipo de proyecto para cumplir sus obligaciones. A alto nivel, estas actividades son concebidas como las fases del proceso (especificación, análisis,...), mientras que a más bajo nivel son tareas más concretas (crear ciertos diagramas, escribir código,...).

Cada actividad debe tener siempre como principal objetivo generar ciertos productos bien definidos y especificados.

PRINCIPALES TAREAS DE LOS PROCESOS SOFTWARE.

1. Entender la naturaleza de la aplicación: entender lo que desean los clientes.
2. Establecer el plan de trabajo.
3. Generar y gestionar la documentación: desde el principio debe disponerse de una estrategia para mantener los documentos que se generen.
4. Captura de los requerimientos: Hay que reunir los requisitos que ha de cumplir la aplicación. Gran parte de esta actividad es conversar con los “interesados”
5. Diseñar y construir el producto: Hay que analizar el problema, diseñar la solución y codificar los programas.
6. Probar y validar el producto: El producto inicial y final debe probarse en forma exhaustiva en todos sus aspectos.
7. Entregar y mantener el producto: Una vez entregado el producto, entra el modo “mantenimiento, que incluye reparaciones y mejoras.

DEFINICIÓN DE RED DE TAREAS.

Una red de tareas, también llamada red de actividad, es una representación gráfica del flujo de tareas para un proyecto. En ocasiones se usa como el mecanismo mediante el cual la secuencia y las dependencias de tareas se integran en una herramienta

automatizada de calendarización de proyecto. En su forma más simple (usada cuando se crea un calendario macroscópico), la red de tareas muestra las principales acciones de la ingeniería del software.

Puesto que las tareas paralelas ocurren de manera asíncrona, debe determinar las dependencias intertarea para asegurar el progreso continuo hacia la conclusión. Además, debe estar al tanto de aquellas tareas que se encuentren en la ruta crítica, es decir, aquellas que deben concluirse conforme al calendario si el proyecto como un todo debe completarse de acuerdo con ese calendario.

Tanto la PERT como el CPM proporcionan herramientas cuantitativas que permiten:

- 1) determinar la ruta crítica (la cadena de tareas que determina la duración del proyecto),
- 2) establecer estimaciones de tiempo “más probables” para tareas individuales aplicando modelos estadísticos
- 3) calcular “tiempos frontera” que definen una “ventana” de tiempo para una tarea particular.

PERT significa Program Evaluation and Review Technique, que traducido es Técnica de revisión y evaluación de programas.

La técnica PERT, desarrollada originalmente en 1958 para usarse en la Oficina de Proyectos Especiales de la Marina de los Estados Unidos, es un sistema de gestión de proyectos diseñado para asistir en la planificación de proyectos grandes y complejos.

El objetivo general de la PERT es que los proyectos se completen a tiempo y dentro del presupuesto.

Las actividades en PERT son las tareas que requieren recursos, del siguiente modo:

- Una actividad PERT se refiere a llevar a cabo una tarea que agota tus recursos, ya sea tiempo, materiales, dinero o incluso maquinaria. No puedes realizar una actividad PERT hasta que el evento que marca su comienzo haya finalizado; por ej., si algo en la fase del evento anterior ha quedado sin finalizar, el nuevo no puede comenzar.
- Una subactividad PERT se refiere a las tareas más pequeñas en las que se puede desglosar una actividad PERT.

Los tiempos de PERT son clave para la técnica y están diseñados para permitir un rendimiento optimizado al igual que contratiempos, del siguiente modo:

- **Tiempo optimista** se refiere al tiempo mínimo que se necesita para completar una actividad, suponiendo que todo fluye sin inconvenientes y sale mejor de lo esperado
- **Tiempo pesimista** es lo opuesto y establece la cantidad máxima del tiempo que se necesita para completar una tarea, suponiendo que todo lo que puede salir mal, sale mal, sin tener en cuenta los desastres totales
- **Tiempo más probable** es el tiempo intermedio en relación con el tiempo necesario predicho, suponiendo que todo fluye normalmente

- **Tiempo esperado** es tu mejor cálculo de entrega, teniendo en cuenta que algunas cosas pueden salir mal, y debe basarse en el tiempo promedio modelo de la finalización de una tarea de rutina. La ecuación básica de cálculo PERT que se emplea para determinar el tiempo esperado es $E = (O + 4M + P)/6$.

Hay varios otros términos para definir la importancia, el progreso y la expectativa de tiempo de cada evento o actividad:

- **Ruta crítica** ofrece una descripción general del proyecto global desde el día uno hasta la fecha de finalización y te permite definir el tiempo total en días, meses o años que tu equipo necesitará para completar todo. El **Método de la ruta crítica** en realidad fue un enfoque diferente que se desarrolló alrededor de la misma época de la PERT, a pesar de que los dos se usan principalmente en conjunto.
- **Actividad crítica** es el término que se otorga a las tareas que no pueden sobrepasar u ocupar más recursos de los que se planificaron. Una actividad crítica es, como el nombre lo sugiere, crítica para la finalización exitosa del proyecto.
- **Tiempo de holgura** se refiere al tiempo y a los recursos disponibles durante el proyecto. Tiempo de holgura libre se refiere a la ventaja de recursos que tienes si tu equipo tiene un retraso, mientras que el tiempo de holgura negativo se refiere a las tareas que tienen un déficit de recursos.
- **Plazo** es el tiempo en el cual debe completarse un evento predecesor para alcanzar un evento.
- **Demora** es el tiempo más corto en que un evento puede seguir a otro.
- **Fast tracking** es cuando tu equipo decide ejecutar tareas críticas en conjunto para acortar la ruta crítica general.
- **Ruta crítica con crashing** significa que introduces recursos adicionales a una actividad de ruta crítica y, como consecuencia, se acorta el tiempo necesario para que se complete, por lo que la ruta crítica se acorta.



PASO 1: ACTIVIDADES DEL PROYECTO

La primera fase corresponde a identificar todas las actividades que intervienen en el proyecto, sus interrelaciones, sucesiones, reglas de precedencia. Además, deberán relacionarse los tiempos estimados para el desarrollo de cada actividad.

El método PERT asume tres estimaciones de tiempo por cada actividad, estas estimaciones son:

Tiempo optimista (a): Duración que ocurre cuando el desarrollo de la actividad transcurre de forma perfecta, es decir el mínimo tiempo en que una actividad de esas características haya sido ejecutada.

Tiempo más probable (m): Duración que ocurre cuando el desarrollo de la actividad transcurre de forma normal. En la práctica suele tomarse como el tiempo más frecuente de ejecución de una actividad de iguales características.

Tiempo pesimista (b): Duración que ocurre cuando el desarrollo de la actividad transcurre de forma deficiente, o cuando se materializan los riesgos de ejecución de la actividad.

Matriz de dependencias y sus tiempos:

Actividad	Tiempo Optimista (a)	Tiempo más probable (m)	Tiempo pesimista (b)	Actividad Precedente
A	3	5,5	11	-
B	1	1,5	5	-
C	1,5	3	4,5	A
D	1,2	3,2	4	B
E	2	3,5	8	C
F	1,8	2,8	5	D
G	3	6,5	7	E
H	2	4,2	5,2	F
I	0,5	0,8	2,3	G - H
J	0,8	2,1	2,8	I

PASO 2: ESTIMAR EL TIEMPO ESTIMADO (DURACIÓN PROMEDIO) Y LA VARIANZA

Para efectos de determinar la ruta crítica del proyecto se acude al tiempo de duración promedio, también conocido como tiempo estimado. Este tiempo es determinado a partir de las estimaciones como:

$$T_e = \frac{a + 4m + b}{6}$$

El cálculo del tiempo estimado deberá hacerse entonces para cada actividad. Por ejemplo para la actividad A:

$$T_e = \frac{3 + 4(5,5) + 11}{6} = 6$$

Además de calcular el tiempo estimado, deberá calcularse la varianza de cada actividad. El cálculo de esta medida de dispersión se utiliza para determinar la incertidumbre de que se termine el proyecto de acuerdo al programa. Para efectos del algoritmo PERT, el cálculo de la varianza se hará a partir de sus estimaciones tal cómo se muestra a continuación:

$$\text{Varianza } (\sigma^2) = \left(\frac{b - a}{6}\right)^2$$

El cálculo de la varianza deberá hacerse entonces para cada actividad. Por ejemplo para la actividad A:

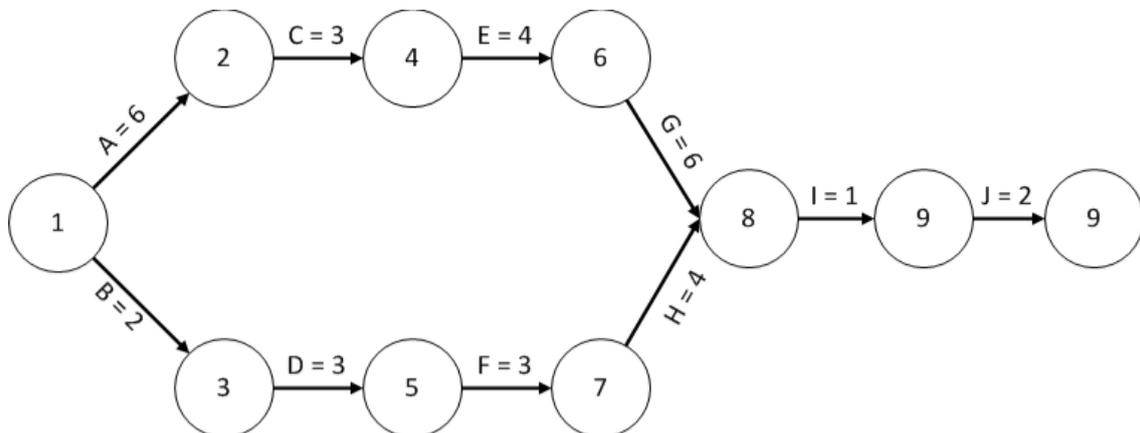
$$\sigma^2 = \left(\frac{11 - 3}{6}\right)^2 = 1,78$$

Para las actividades del tabulado mencionado en el Paso 1, los tiempos estimados y varianzas serían las siguientes:

Actividad	Tiempo Optimista (a)	Tiempo más probable (m)	Tiempo pesimista (b)	Tiempo estimado	Varianza
A	3	5,5	11	6	1,78
B	1	1,5	5	2	0,44
C	1,5	3	4,5	3	0,25
D	1,2	3,2	4	3	0,22
E	2	3,5	8	4	1,00
F	1,8	2,8	5	3	0,28
G	3	6,5	7	6	0,44
H	2	4,2	5,2	4	0,28
I	0,5	0,8	2,3	1	0,09
J	0,8	2,1	2,8	2	0,11

PASO 3: DIAGRAMA DE RED

Con base en la información obtenida en la fase anterior y haciendo uso de los conceptos básicos para diagramar una red, obtendremos el gráfico del proyecto (los tiempos relacionados con cada actividad en el gráfico corresponden a los tiempos estimados):



PASO 4: CALCULAR LA RED

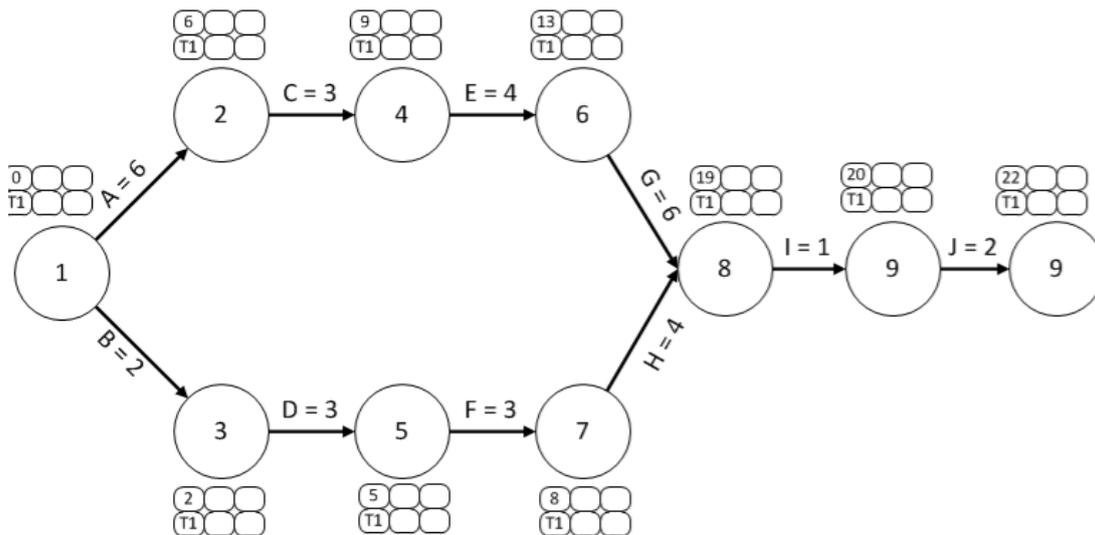
Para el cálculo de la red se consideran 3 indicadores

- T1: tiempo más temprano
- T2: tiempo más tardío
- H: tiempo de holgura

Estos indicadores se calculan en cada evento o nodo (entiéndase nodo entonces como un punto en el cual se completan actividades y se inician las subsiguientes).

T1: Tiempo más temprano de realización de un evento. Para calcular este indicador deberá recorrerse la red de izquierda a derecha y considerando lo siguiente:

- T1 del primer nodo es igual a 0.
- T1 del nodo n = T1 del nodo n-1 (nodo anterior) + duración de la actividad (tiempo estimado) que finaliza en el nodo n.
- Si en un nodo finaliza más de una actividad, se toma el tiempo de la actividad con mayor valor.



En este caso para el cálculo del T1 en el nodo 8, en el que concurre la finalización de 2 actividades, deberá considerarse el mayor de los T1 resultantes:

$$T1 (\text{nodo } 6) + G = 13 + 6 = 19$$

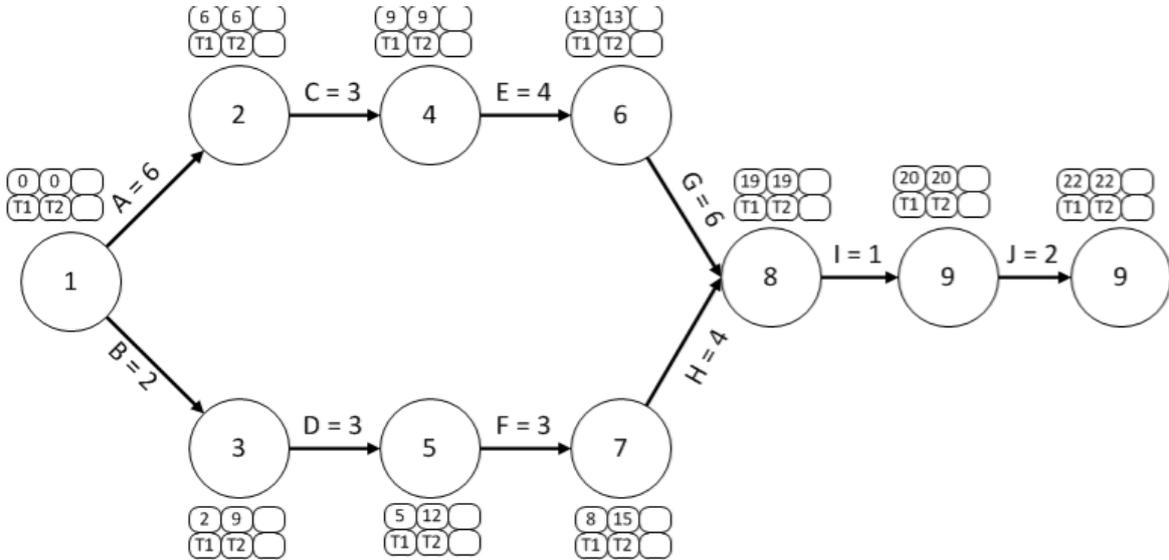
$$T1 (\text{nodo } 7) + H = 8 + 4 = 12$$

Así entonces, el T1 del nodo 8 será igual a 19 (el mayor valor).

T2: Tiempo más tardío de realización del evento. Para calcular este indicador deberá recorrerse la red de derecha a izquierda y considerando lo siguiente:

- T2 del primer nodo (de derecha a izquierda) es igual al T1 de este.
- T2 del nodo n = T2 del nodo n-1 (nodo anterior, de derecha a izquierda) - duración de la actividad que se inicia (tiempo estimado).

- Si en un nodo finaliza más de una actividad, se toma el tiempo de la actividad con menor valor.

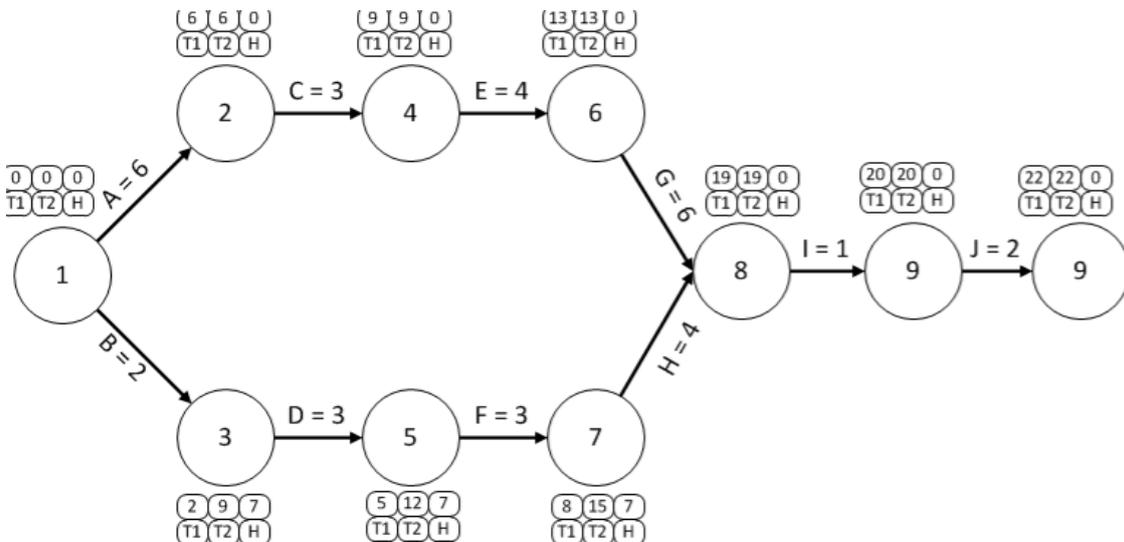


En este caso para el cálculo del T2 del nodo 1, en el que concurren el inicio de 2 actividades deberá entonces considerarse lo siguiente:

$$\begin{aligned} \text{T2 nodo 2} - \text{B} &= 6 - 6 = 0 \\ \text{T2 nodo 3} - \text{C} &= 9 - 2 = 7 \end{aligned}$$

Así entonces, el T2 del nodo 1 será 0, es decir el menor valor.

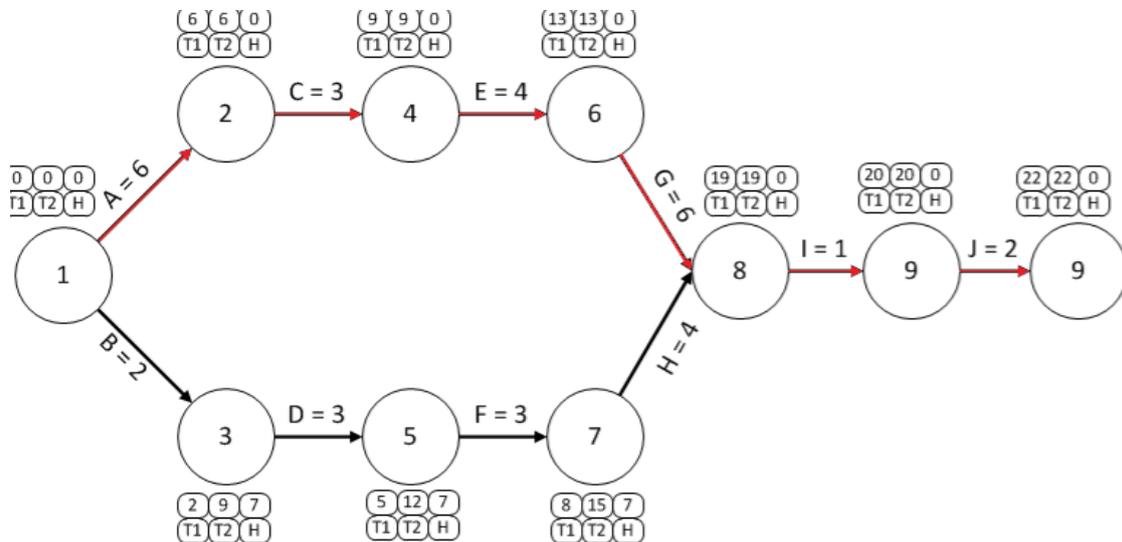
H: Tiempo de holgura, es decir la diferencia entre T2 y T1. Esta holgura, dada en unidades de tiempo corresponde al valor en el que la ocurrencia de un evento puede tardarse. Los eventos en los cuales la holgura sea igual a 0 corresponden a la ruta crítica, es decir que la ocurrencia de estos eventos no puede tardarse una sola unidad de tiempo respecto al cronograma establecido, dado que en el caso en que se tardara retrasaría la finalización del proyecto.



Las actividades críticas por definición constituyen la ruta más larga que abarca el proyecto, es decir que la sumatoria de las actividades de una ruta crítica determinará

la duración estimada del proyecto. Puede darse el caso en el que se encuentren más de una ruta crítica.

Ruta crítica:



Esta ruta se encuentra compuesta por las actividades A, C, E, G, I, J. La duración del proyecto sería de 22 semanas.

Se pueden utilizar herramientas para realizar el diagrama de PERT (sinnaps, Lucidchart)

Reglas básicas

- Regla 1: Se usa una, y sólo una flecha para representar una actividad a ejecutarse.
- Regla 2: El diagrama se construye conectando, bajo una relación causal, flechas que representa actividades, considerando para cada una, las tres preguntas siguientes:
 - 1 qué precede inmediatamente a esta actividad,
 - 2 qué sigue inmediatamente a esta actividad, y
 - 3 qué actividades son concurrentes.
- Regla 3: Iniciar el diagrama con una flecha preliminar.
- Regla 4: Enumerar los acontecimientos.
- Regla 5: Utilice las actividades ficticias, solo cuando precise mantener la lógica del diagrama.

Ver ejemplo en el siguiente enlace:



PERT
 Técnica de Evaluación y Revisión de Proyectos
<https://padlet.com/dlobos4152/PERT>

En general, los diagramas de Gantt son más populares para planificar en detalle las tareas del proyecto a lo largo del ciclo de vida del mismo. A partir de las tareas se genera un cronograma, (ver figura 8)

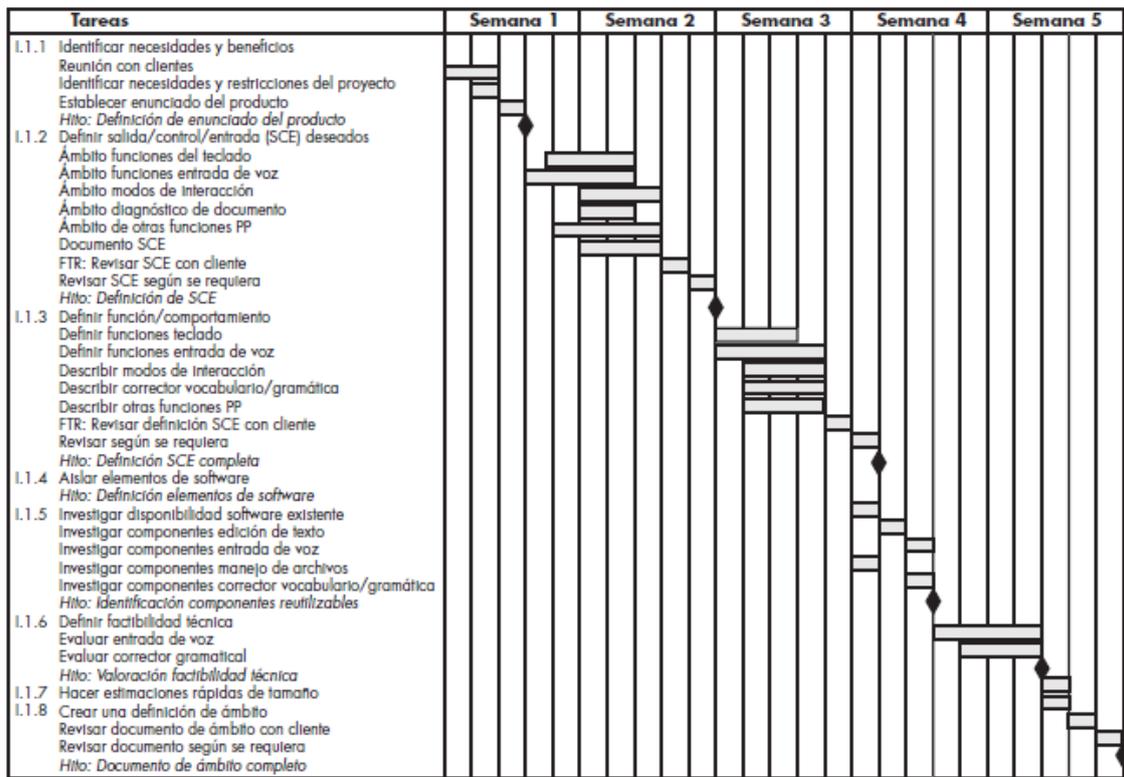


Figura 8. Diagrama de Gantt

SEGUIMIENTO DE LA PLANIFICACIÓN TEMPORAL.

A partir del Gráfico de Tiempo se puede realizar el seguimiento del proyecto en varias formas diferentes:

- Realizar reuniones periódicas del estado del proyecto, en las que cada miembro del equipo reporte avances y problemas
- Evaluar los resultados de todas las revisiones realizadas a través del proceso de ingeniería del software
- Determinar si los hitos formales del proyecto (los diamantes que se muestran en la figura 8) se lograron en la fecha prevista
- Comparar la fecha de inicio real con la fecha de inicio planeada para cada tarea de proyecto mencionada en la tabla de recursos.
- Reunirse informalmente con los profesionales para obtener su valoración subjetiva del avance a la fecha y los problemas en el horizonte.

DIAGRAMA DE PERT VS. DIAGRAMA DE GANTT

Estas son algunas características clave que debes tener en cuenta al decidir cuál es el adecuado para ti:

- Los diagramas de PERT son diagramas de flujo, mientras que los diagramas de Gantt son gráficos de barras: Una de las mayores diferencias está en su diseño

visual. Los diagramas de Gantt adoptan un enfoque de diagrama de barras tradicional, mientras que los diagramas de PERT tienen una apariencia menos estructurada y adoptan diferentes diseños según el proyecto.

- Los diagramas de Gantt ofrecen organización, mientras que los diagramas de PERT ofrecen personalización: Los diagramas de Gantt están organizados de manera más estructural. Por otro lado, los diagramas de PERT permiten una personalización de diseño simple que es mejor para las necesidades de proyectos de nivel más general.
- Los diagramas de PERT se pueden utilizar antes de que comience el proyecto: Dado que los diagramas de PERT ofrecen un plan de proyecto y un diseño de cronogramas simples, a menudo se usan como guías visuales durante el inicio del proyecto. Los gerentes de proyectos luego usan otro método, como una estructura de desglose del trabajo o un diagrama de Gantt, específicamente para planificar en detalle las tareas y dependencias del proyecto.

PLAN DEL PROYECTO.

Un plan de proyecto es una herramienta para cumplir el cronograma de un proyecto dentro del presupuesto. Se trata de establecer los elementos que requieres para que se cumpla un objetivo específico. Lograrás identificar el panorama general y cada una de las acciones que te llevarán a concluir el proyecto de manera eficaz.

Un plan de proyecto es la mejor herramienta que te ayudará en la gestión de tareas y acciones, tanto con tu equipo como con el cliente: desde la planificación, ejecución, verificación hasta la concreción del proyecto.

¿Cómo elaborar un plan de proyecto?

1. Realiza una visión general

Debes tener una visión general del proyecto y sus componentes clave; aquí identificas las necesidades del cliente y las soluciones que ofreces. Determinas también quiénes son las partes interesadas.

2. Establece el alcance

Aquí describes cuáles son los alcances de este proyecto, es decir, qué acciones implica y hasta dónde llega. Recuerda que deben quedar muy claros los límites. Quedará asentado asimismo el consenso o acuerdo entre las partes involucradas.

3. Desarrolla los objetivos

Es necesario establecer objetivos SMART, para que quede claro qué se buscará. Cada proyecto puede tener uno o más objetivos, los cuales debes enumerar y describir de forma concreta.

4. Identifica roles y responsabilidades

Asegúrate de establecer el mando y flujo del proyecto, así como el equipo de trabajo que se encargará del mismo. Desde la planificación, cada miembro sabrá quién toma las decisiones, quién establecerá el contacto con el cliente, qué hará cada uno, cuál es su rol y sus responsabilidades.

5. Determina los entregables

Este apartado es esencial porque enumerarás los productos o servicios específicos que ofreces al cliente (recuerda que deben describirse detalladamente). Además, determinarás los entregables y responsables de cada uno de esos productos o servicios.

6. Genera un calendario

Este elemento es fundamental para la administración de los tiempos, de tal forma que puedas dividir el proyecto en fases, qué tareas forman cada fase, cuáles serán los entregables y las fechas de inicio y finalización de cada etapa. Esto te ayuda a contar con un cronograma y un recurso con el que visualizarás los avances. Te puede servir realizar un diagrama de Gantt.

7. Asigna el presupuesto

Hasta este punto ya tienes noción de cuáles serán las tareas y entregables. A continuación, calcula el precio de cada una de las actividades o artículos con base en el tiempo que llevará su realización. Con esto presente podrás asignar presupuestos para tener un panorama de cada artículo, su precio y cuántos se harán.

Puedes incluir en el presupuesto los artículos de precio fijo y por hora, así como los costes recurrentes y únicos.

8. Registra la aprobación

Todos los involucrados en el proyecto, incluyendo evidentemente al cliente, deben revisar este plan de proyecto para aprobarlo. Esto indicará que están de acuerdo y queda un registro de todos los aspectos para que no haya problemas en cuanto a lo establecido.

Bibliografía Consultada

Drake, José M. (2008) Proceso de desarrollo de aplicaciones software. Disponible en: https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf

Manual de Administración de Proyectos (Consulta agosto 2022). Disponible en: http://www.liderdeproyecto.com/manual/estimacion_de_esfuerzo_del_proyecto.html

Perissé, M. (2021) PERT: una metodología simplificada. Online: http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/manual_pert/manual_pert.htm#Toc62595259

Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

Sordo, Ana. Plan de proyecto: qué es y cómo crearlo (publicado originalmente el 30 de agosto de 2021, actualizado el 23 de marzo de 2022). Disponible en: <https://blog.hubspot.es/marketing/crear-plan-proyecto>

GESTION DE PROYECTOS DE SOFTWARE

Antes de poder planear un proyecto, deben establecerse **los objetivos** y el **ámbito del software**, considerarse soluciones alternativas e identificar las restricciones técnicas y administrativas. Sin esta información, es imposible definir estimaciones razonables (y precisas) del costo, una valoración efectiva del riesgo, una descomposición realista de las tareas del proyecto y un calendario de proyecto manejable que proporcione en cada momento un indicio significativo del progreso (Pressman, 2010).

DEFINICIÓN DEL ÁMBITO DEL SOFTWARE.

El ámbito del software se define al responder las siguientes preguntas:

Contexto. ¿Cómo encaja en un sistema, producto o contexto empresarial más grande, el software que se va a construir y qué restricciones se imponen como resultado del contexto?

Objetivos de información. ¿Qué objetos de datos visibles para el cliente se producen como salida del software? ¿Qué objetos de datos se requieren como entrada?

Función y desempeño. ¿Qué función realiza el software para transformar los datos de entrada en salida? ¿Existe alguna característica de desempeño especial que deba abordarse?

DEFINICIÓN DE RECURSOS DEL PROYECTO.

La estimación de recursos consiste en la planificación y garantía de la disponibilidad de los mismos para asegurar el buen desarrollo y éxito de un proyecto. La gestión de los recursos disponibles es esencial para cualquier proyecto y debe tenerse en cuenta incluso antes de que el mismo comience.

En proyectos de software existen tres principales categorías de recursos desde la perspectiva de la ingeniería de software:

1. Personal (recursos humanos): El número de personas requeridas para un proyecto de software puede determinarse sólo después de hacer una estimación del esfuerzo de desarrollo (por ejemplo, persona-meses).
2. Componentes de software reutilizables: con frecuencia llamados componentes. Pueden ser:
 - a. Componentes comerciales: se compran de una tercera parte, están listos para su uso en el proyecto actual y están completamente validados.
 - b. Componentes de experiencia completa: son especificaciones, diseños, código o datos de prueba existentes, desarrollados por el equipo de trabajo para proyectos anteriores que son similares al software que se va a construir para el proyecto en cuestión.
3. Entorno de desarrollo (herramientas de hardware y software): Cada elemento de hardware debe especificarse como parte de la planificación, por ejemplo, si se necesita un hardware especializado para el producto que se va a desarrollar. Otro hardware a considerar es la computadora y los servidores sobre la cual se

ejecutará el software cuando éste se haya liberado al usuario final. El software está referido al necesario para desarrollar el producto, por ejemplo, un IDE, o también llamado Entorno de Desarrollo Integrado, es un programa que nos ayuda en la tarea de programar de modo que facilita enormemente el proceso de desarrollo y depuración de un software (Eclipse, Netbeans, etc.).

Cada recurso se especifica con cuatro características:

1. descripción del recurso.
2. Enunciado de disponibilidad.
3. Momento en el que se requerirá el recurso.
4. Duración del tiempo que se aplicará el recurso.

ESTIMACIÓN DE UN PROYECTO (COSTO Y TIEMPO).

La estimación de costos está asociada con la estimación fiable del tamaño y los recursos necesarios para producir el producto de software. El objetivo de la estimación de costo de software consiste en estimar el tamaño, el esfuerzo, la complejidad y el costo del proyecto de software para poder encontrar la mejor decisión de desarrollo y asegurar que el gasto se encuentre de acuerdo con lo presupuestado.

FORMAS DE ABORDAR LA ESTIMACIÓN DE COSTOS.

- Opinión de expertos: Un desarrollador o gestor describe los parámetros del proyecto y los expertos hacen estimaciones basadas en su experiencia.
- Técnica Delphi: permite sistematizar y mejorar la opinión de los expertos consultados.
- Analogía: Enfoque más formal que el de la opinión de expertos. Los expertos comparan el proyecto propuesto con uno o más proyectos anteriores intentando encontrar similitudes y diferencias particulares.
- Base Histórica de proyectos.



[Estimación de tiempos y costos del proyecto](#)

TÉCNICAS DE DESCOMPOSICIÓN: DEL PROYECTO (O POR FASES).DEL PRODUCTO (O POR MÓDULOS).

La estimación del proyecto completo se calcula mediante la suma de las cantidades parciales, mediante el criterio abajo-arriba (bottom-up) o arriba-abajo (top-down). Lo más adecuado es utilizar las técnicas de descomposición estructurada (EDT/WBS).

Entre las ventajas se encuentran:

- La posibilidad de que el responsable del componente a estimar participe en dicha estimación.
- Ayuda a analizar con detalle cada componente.

Entre los inconvenientes se encuentran:

- La dificultad para contemplar los costes de:
- Actividades relacionadas con el proyecto como lectura de código, revisión, reuniones,... (según Barry Boehm supone el 40% del proyecto).
- Actividades no relacionadas con el proyecto relacionados con los hábitos de trabajo (según Barry Boehm supone el 30% del proyecto).

Técnicas de descomposición: Del proyecto (o por fases).

Ver la figura 9.

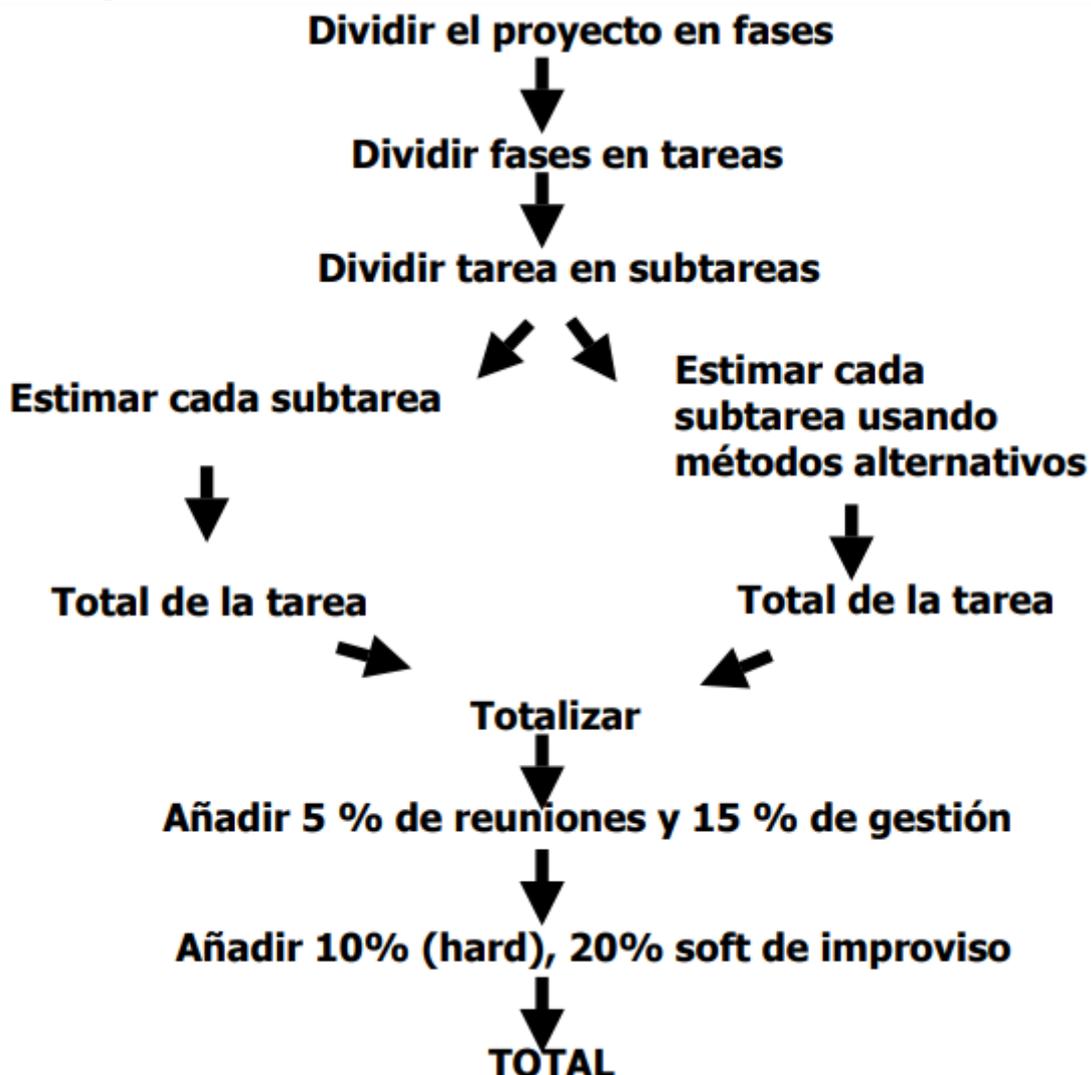


Figura 9. Descomposición por fases. Fuente: <http://www.lsi.us.es/docencia/get.php?id=326>

Del producto (o por módulos)
Ver Figura 10.

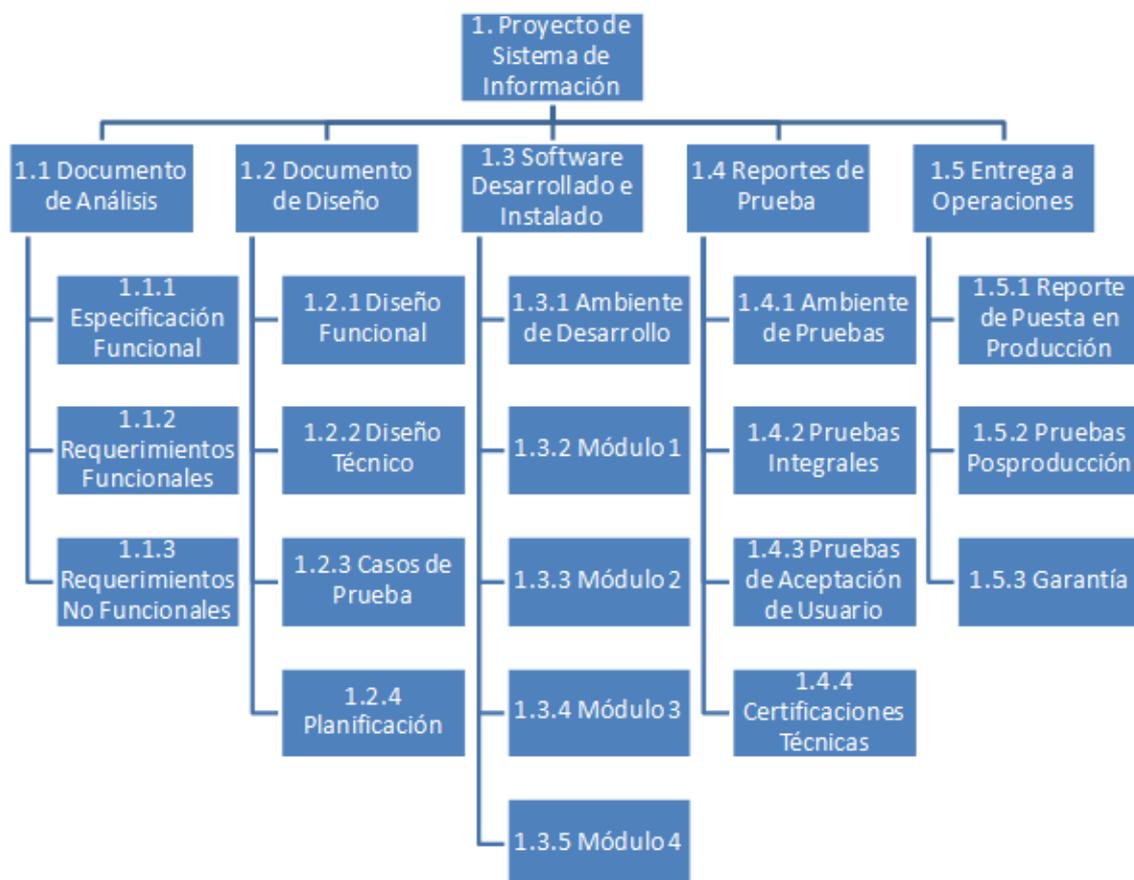


Imagen elaborada por www.pmoinformatica.com

Figura 10. Descomposición por módulos. Fuente:
<http://www.pmoinformatica.com/2018/08/tecnicas-estimacion-software.html>

PLANES DE CALIDAD, DE VALIDACIÓN, DE CONFIGURACIÓN, DE DESARROLLO DE PERSONAL.

Plan de Calidad: En este plan se deben identificar los diferentes criterios de calidad que se van a considerar en cada fase del desarrollo, tanto los relativos al producto como los relativos al proceso, así como los métodos y recursos que se van a utilizar para comprobar dichos criterios, y los informes que será necesario producir.

Plan de validación y verificación: Según Boehm, la **validación** es un proceso más general. Se debe asegurar que el software cumple las expectativas del cliente. Va más allá de comprobar si el sistema está acorde con su especificación, para probar que el software hace lo que el usuario espera a diferencia de lo que se ha especificado. La

verificación es comprobar que el software está de acuerdo con su especificación. Se comprueba que el sistema cumple los requerimientos funcionales y no funcionales que se le han especificado.

Está encuadrado en un modelo de referencia de la función de Verificación & Validación, basado en el estándar IEEE Std. 1012 (ver figura 11).

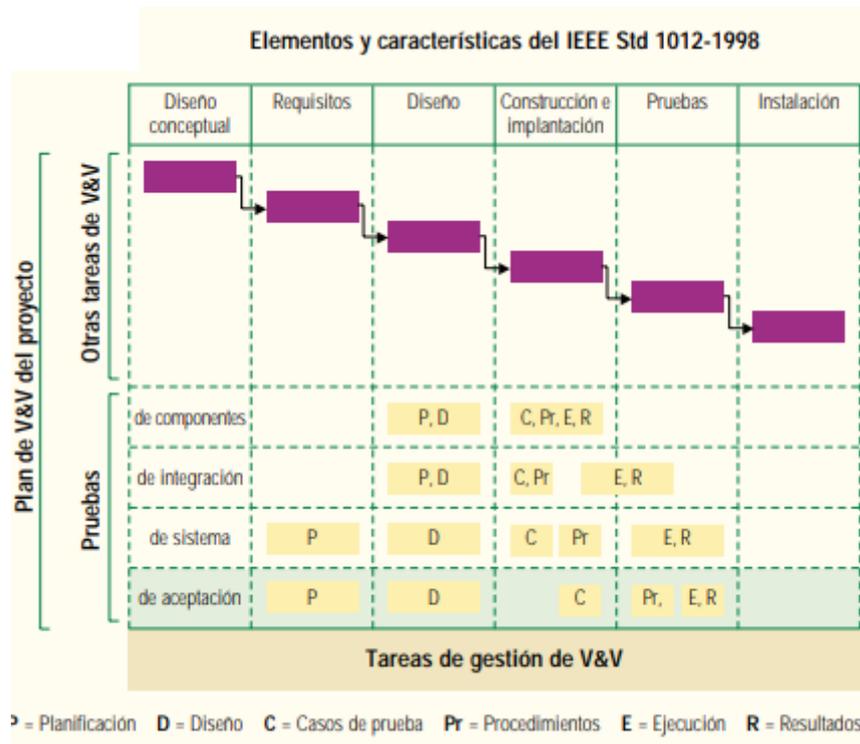


Figura 11. IEEE Std. 1012- 1998. Fuente:

https://www.aec.es/c/document_library/get_file?uuid=58b72afd-2050-4d40-9bc4-c875a777cc3d&groupId=10128

Plan de configuración de software: es un conjunto de actividades aplicadas durante el proceso de desarrollo dirigido a:

- **Identificación de la Configuración:** Todos los requisitos de configuración de un proyecto, incluidos los requisitos de funcionalidad, los requisitos de diseño y cualquier otra especificación, deben identificarse y registrarse. La realización de este proceso tiene como resultado la referencia de configuración para el proyecto.
- **Control de la Configuración:** A medida que se modifica el alcance del proyecto, se debe evaluar, aprobar y documentar su impacto en la configuración. Esto suele ser parte del proceso de control de cambios del proyecto.
- **Administración del Estado de la Configuración:** Realiza un seguimiento de la configuración del proyecto en todo momento, para saber en qué versión se encuentra la configuración del software y tener un historial de las versiones

anteriores. Es fundamental llevar un registro de todas las versiones para poder controlar los cambios a lo largo del proyecto.

- **Auditorías y Revisiones de la Configuración:** Se refiere a cualquier prueba para demostrar que el producto cumple con los requisitos de configuración. Ejemplo: un informe debe ejecutarse en 10 segundos. La auditoría realiza una prueba para ver si el informe terminado realmente se ejecuta tan rápido. A menudo, las auditorías y revisiones se llevan a cabo al finalizar las fases principales del proyecto y tienen como objetivo la detección temprana de problemas.

Plan de desarrollo de personal: describe cómo se desarrollarán las habilidades y experiencia de los miembros del equipo del proyecto. Dichas personas deberán tener un nivel de responsabilidad y se les deberán asignar premios de acuerdo con sus capacidades. Los administradores de proyectos tienen que resolver problemas técnicos y no técnicos, motivar a la gente, planear y organizar su trabajo, asegurando que este se realice de manera adecuada. La administración pobre del personal es uno de los factores más importantes para el fracaso de los proyectos.

SOFTWARE PARA GESTIONAR PROYECTOS

Las principales ventajas que podemos obtener de usar alguna herramienta para gestionar los proyectos son:

- **Recogida automática de información:** muestran forma automática y objetiva los tiempos dedicados a cada proyecto sin tener que reportarlo manualmente.
- **Control total sobre el proyecto.** Se aporta una visión de conjunto y ayuda a organizar la planificación y seguimiento desde el origen hasta el final del proyecto.
- **Análisis de desviaciones:** en cuanto a tiempos, presupuesto, gastos, avances del proyecto según previsiones etc.
- **Mejora la comunicación interna.** Cualquier proyecto, independientemente de su tamaño, implica la colaboración de varios miembros de la plantilla. Contar con una herramienta para gestionar proyectos ayuda a que la comunicación entre las personas sea más fluida, directa y sencilla.



Los software de gestión de proyectos como:

- Jira (<https://www.atlassian.com/es/software/jira>)
- Trello (<https://trello.com/es>)
- Entre otros

¿Son el futuro de los proyectos de desarrollo de software?... NO, son el futuro de la gestión de proyectos, independientemente del tipo de proyecto en el que esté trabajando.



[Industria 4.0 y Gestión de Proyectos: 5 ideas clave por Juan Luis Vila Grau](#) [| Oct 3, 2017](#)

La importancia que la gestión de proyectos en este entorno de cambio planteado por la Industria 4.0 es de vital importancia. Independientemente de su ámbito de trabajo, su tamaño o el tiempo que lleve operando en el mercado cualquier organización se tienen que plantear un proceso de renovación y mejora. Para hacerlo con ciertas garantías de éxito, debería hacerse desde el enfoque de la gestión de proyectos. Al respecto se destacan 5 ideas claves:

#1. Proyectos como el medio para introducir el cambio

El concepto de proyecto tradicionalmente ligado al ámbito de la ingeniería y la construcción ha evolucionado. Todas las organizaciones en algún momento se encontrarán en situación de realizar o llevar a cabo una iniciativa diferente de las operaciones diarias con el objetivo de lograr un cambio. Para afrontar con ciertas garantías de éxito dicho cambio, lo mejor es hacerlo desde la perspectiva de la gestión de proyectos.

#2. Diferenciar entre proyectos y operaciones

Es necesario diferencias entre las operaciones habituales, que se refiere a las actividades del negocio habitual de las empresas, y la gestión de proyectos. La necesidad se basa en el hecho de que son diferentes; presentan oportunidades y amenazas diferentes, y por lo tanto se deberían gestionar de manera diferente.

#3. Necesidad de contar con un método de gestión de proyectos

Los proyectos cuentan con una serie de características particulares y es necesario gestionarlos de manera diferente a las operaciones. Es recomendable contar con un método de gestión de Proyectos para los proyectos. Por ejemplo:



Figura 12. Métodos de Gestión de Proyecto.

#4. Beneficios de contar con una cultura de gestión de proyectos.

Los beneficios de contar con una metodología de gestión de proyectos en la organización beneficios se extienden más allá del ámbito de gestión de proyectos. La cultura de la organización puede beneficiarse positivamente de la adopción de una metodología de gestión de proyectos.

#5. Un método de gestión de proyectos no garantiza el éxito de los proyectos.

Y por último, aunque como suele decirse no por ello menos importantes, es necesario ser consciente de que emplear un método de gestión de proyectos no garantiza el éxito de nuestro proyecto, pero sí que contribuye a su éxito, y a la correcta gestión del mismo.



Para Pensar

Como Jefe de Proyecto de la situación planteada en el comic, cuál sería la respuesta adecuada para el cliente.

Panel 1 (Top Left): Client: "Así que, ... ¿cuánto tardas en montar el entorno?" Manager: "1 semana." Client: "¿y lo demás?" Manager: "análisis 1 semana, diseño 1 semana y desarrollo, 2 semanas."

Panel 2 (Top Middle): Manager: "vale, te pongo 4 semanas." Client: "serán 5..." Manager: "eeeuhh... sí, claro." Client: "4 se-ma-nas" (thought bubble). Manager: "una cosita más sin importancia..." Client: "¿sí?"

Panel 3 (Top Right): Client: "que al final, en vez de hacerlo en JSP, tienes que hacerlo en VB.NET, en vez de ser una página web es una aplicación de **guindous** y en vez de ser una aplicación de **venta de entradas de cine** va a ser una plataforma de edición de **video en 3d** con tecnología **experimental** que tienes que **investigar** para ver si es **viable**..."

Panel 4 (Second Row Left): Client: "¿puedes hacer los ajustes de las estimaciones?" Manager: "¿ahora?" Client: "sí, claro" Manager: "un momentito..."

Panel 5 (Second Row Middle): Manager: "date prisa, que tengo una reunión con el cliente en media hora.." Client: "solo será un segundo..."

Panel 6 (Second Row Right): Manager wearing a wizard hat.

Panel 7 (Third Row Left): Manager with wizard hat: "¡Ejem! Ooooooh, poderoso Dormammu, señor del inframundo... ¡¡¡YO TE INVOCO!!!! concédeme tu visión de futuro y tu percepción extrasensoriallll..."

Panel 8 (Third Row Middle): Wizard hat: "¿QUÉEEE QUIERES, POBRE MORTAL???"

Panel 9 (Third Row Right): Wizard hat: "¿Puedes estimar los cambios de recursos necesarios para realizar una tarea sobre la que no tengo ni puta idea, en un lenguaje que desconozco, que requiere una labor de investigación previa y que ni siquiera sabemos si es viable?"

Panel 10 (Bottom Row Left): Wizard hat: "POR SUPUEEEEEESTO... EXACTAMENTE TARDARÁS EN HACERLO... ESPERA QUE CONSULTE A MEFISTO Y A LILITH... 3 SEMANAS, 2 DÍAS, 3 HORAS, 5 MINUTOS Y 24 SEGUNDOSSSS." Client: "¿qué fiabilidad la das a esta estimaciónnnn?"

Panel 11 (Bottom Row Middle): Wizard hat: "UN 78.356437654%, EXACTAMENTE. AHORA, SACRIFICA UN CORDERO LECHAL EN LA MESA DE REUNIONES PARA SATISFACER MIS HONORARIOOOOS..." Client: "¿No admites Mastercard?" Wizard hat: "NOO..." Client: "Peste de consultorías extradimensionales..."

Panel 12 (Bottom Row Right): Wizard hat: "¡Freeeeed! ¿dónde vas? oyeeeee... ¿tendrás una demo en 1 semana? ¡pregúntaselo al dormilón ese! ¿vas a por el cordero? ¿o te deja pagar aplazado?" Client: "quizá decirle que estaba siendo irónico no hubiera estado de más..."

Bibliografía Consultada

Galindo González Carlos. (2009). Administración de riesgos y manejo del personal en proyectos de software. Recuperado de <https://www.gestiopolis.com/administracion-riesgos-manejo-personal-proyectos-software/>

Morales, L. (2005) Validación del producto de Software. Revista Calidad. Aec.es. Recuperado el 4 de agosto de 2022, de https://www.aec.es/c/document_library/get_file?uuid=58b72afd-2050-4d40-9bc4-c875a777cc3d&groupId=10128

Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

Wrike (2022) Guía Gestión De Proyecto. Online: <https://www.wrike.com/es/project-management-guide/fag/que-es-la-gestion-de-la-configuracion-en-gestion-de-proyectos/>

Bibliografía Consultada

IEEE Std. 1012- 2016
[https://katie.cs.mtech.edu/classes/esof411/Schedule/IEEE Standard System Software HardwareV&V 160pages.pdf](https://katie.cs.mtech.edu/classes/esof411/Schedule/IEEE%20Standard%20System%20Software%20HardwareV&V%20160pages.pdf)

MÉTRICAS DEL PROCESO DE SOFTWARE

Aunque los términos medida, medición y métrica con frecuencia se usan como sinónimos, es importante observar las sutiles diferencias entre ellos.

MEDIDAS, MÉTRICAS E INDICADORES.

Medida proporciona un indicio cuantitativo de la extensión, cantidad, dimensión, capacidad o tamaño de algún atributo de un producto o proceso. Se clasifican en:

- Medidas Directas: una entidad o atributo no involucra a ninguna otra entidad atributo ejemplo: cantidad de líneas de código fuente (LDC)
- Medidas indirectas: se obtiene a partir de medidas directas. Ejemplo: funcionalidad.

Medición es el acto de determinar una medida.

Métrica una medida cuantitativa del grado en el que un sistema, componente o proceso posee un atributo determinado.

Indicador es una métrica o combinación de métricas que proporcionan comprensión acerca del proceso de software, el proyecto de software o el producto en sí.



Para Pensar

Señale con una cruz cual es la opción correcta

	Métricas	Medidas	Medición	Indicadores
El equipo A escribió 35.000 líneas de código				
Ana será la encargada de medir las líneas de código de cada módulo del sistema				
La productividad del proyecto fue 500 líneas de código (LDC/personas-mes)				
El equipo A encontró 342 errores durante el proceso del software antes de su realización.				
El jefe de proyecto será encargado de medir las desviaciones entre el desarrollo planificado y el desarrollo realizado				
Densidad de defectos entregada				
Índice de trabajo acumulado				
La funcionalidad de una aplicación desarrollada en Java es de 63 LDC por punto de función				

Respuestas Correctas

	Métricas	Medidas	Medición	Indicadores
El equipo A escribió 35.000 líneas de código		X		
Ana será la encargada de medir las líneas de código de cada módulo del sistema			X	
La productividad del proyecto fue 500 líneas de código (LDC/personas-mes)	X			
El equipo A encontró 342 errores durante el proceso del software antes de su realización.		X		
El jefe de proyecto será encargado de medir las desviaciones entre el desarrollo planificado y el desarrollo realizado			X	
Densidad de defectos entregada				X
Índice de trabajo acumulado				X
La funcionalidad de una aplicación desarrollada en Java es de 63 LDC por punto de función	X			

MÉTRICAS DEL PROCESO Y DOMINIOS DEL PROYECTO.

Las **métricas del proceso** son **estratégicas**, determinan el curso del proceso de producción del software, mejoran la calidad de una operación o un proceso mediante la medición de sus atributos y descubrir errores antes de liberar el software desarrollado. En el proceso de mejoramiento de procesos se detectan y reportan defectos emitidos por los usuarios finales.

Al desarrollar un conjunto de métricas para mejorar los procesos se desarrollan un conjunto de métricas clasificadas como privadas y públicas.

- **Métricas privadas:** denominadas como defectos por individuos por componente durante el desarrollo del proyecto.
- **Métricas públicas:** denominadas como índices a nivel de proyecto, esfuerzo, planificación, etc.

Las métricas hacia la mejora de los procesos ofrecen indicadores que conducen a estrategias aún más específicas.

Las **métricas del dominio del proyecto** son **tácticas**, determinan el curso del proyecto actual, la primera aplicación ocurre cuando se realiza la estimación.

OTRAS MÉTRICAS EN LA GESTIÓN DE PROYECTOS DE SOFTWARE

Métricas de calidad: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente.

Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad.



Constanzo, M. A., Casas, S. I., & Marcos, C. A. (2014). Comparación de modelos de calidad, factores y métricas. *Informes Científicos Técnicos-UNPA*, 6(1), 1-36. Recuperado de: <https://publicaciones.unpa.edu.ar/index.php/ICTUNPA/article/view/395>



León Perdomo, Yeniset, Enrique Góngora Rodríguez, Asnier, & Febles Estrada, Ailyn. (2013). **Aplicando métricas de calidad a proyectos y procesos durante las pruebas exploratorias.** *Revista Cubana de Ciencias Informáticas*, 7(2), 193-205. Recuperado en 09 de agosto de 2022, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992013000200008&lng=es&tlng=es.

Métricas orientadas al tamaño se utilizan para obtener medidas directas del resultado y de la ingeniería del software.

Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño, como la que muestra la figura 13.

Proyecto	LDC	Esfuerzo	Coste £(000)	Pag. Doc.	Errores	Defectos	Personas
Alfa	12.100	24	168	365	134	29	3
Beta	27.200	62	440	1224	321	86	5
Gamma	20.200	43	314	1050	256	64	6
.

Figura 13. Métricas orientadas al Tamaño

Ejemplo: En el proyecto alfa se desarrollaron 12.100 líneas de código (LDC) con 24 personas-mes y con un coste de E 168.000. Debe tenerse en cuenta que el esfuerzo y el coste registrado en la tabla incluyen todas las actividades de ingeniería del software (análisis, diseño, codificación y prueba) y no sólo la codificación. Otra información sobre

el proyecto alfa indica que se desarrollaron 365 páginas de documentación, se registraron 134 errores antes de que el software se entregara y se encontraron 29 errores después de entregárselo al cliente dentro del primer año de utilización. También sabemos que trabajaron tres personas en el desarrollo del proyecto alfa.

Métricas orientadas a la función: Son medidas indirectas del software y del proceso por el cual se desarrolla, se centran en la funcionalidad o utilidad del programa, fueron propuestas por Albercht quien sugirió un acercamiento a la medida de la productividad denominado método del punto de función. Los puntos de función se calculan completando la siguiente tabla:

Tabla 6. Tabla de Factores de Ponderación.

Parámetros de medición	Factor de ponderación				=	[]
	Cuenta	Simple	Medio	Complejo		
Número de entradas de usuario	[] x	3	4	6	=	[]
Número de salidas de usuario	[] x	4	5	7	=	[]
Número de peticiones de usuario	[] x	3	4	6	=	[]
Número de archivos	[] x	7	10	15	=	[]
Número de interfaces externas	[] x	5	7	10	=	[]
Cuenta total	→					[]

- Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.
- Número de peticiones de usuario. Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- Número de archivos. Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
- Número de interfaces externas. Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Para calcular puntos de función (PF), se utiliza la relación siguiente:

$$PF = \text{cuenta-total} \times [0,65 + 0,01 \times 6(F_i)]$$

en donde cuenta-total es la suma de todas las entradas PF obtenidas de la tabla.

F_i (i=1 a 14) son «valores de ajuste de la complejidad» según las respuestas a las siguientes preguntas:

Nota: Cada una de las preguntas es respondida usando una escala con rangos desde 0 hasta 5:

0	1	2	3	4	5
No influencia	Incidental	Moderado	Medio	Significativo	Esencial

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualizan los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Con el dato de los puntos de función se pueden obtener los siguientes datos del proyecto:

$$\begin{aligned} \text{Productividad} &= PF / \text{personas-mes} & \text{Calidad} &= \text{Errores} / PF \\ \text{Coste} &= \text{Dinero} / PF & \text{Documentación} &= \text{Paginas documentación} / PF \end{aligned}$$

MEDICIONES DEL SOFTWARE. MEDICIONES DE CALIDAD.

Muchos diseñadores del software no coleccionan las medidas. Sin las medidas es imposible determinar si un proceso está mejorando o no. La medición es un elemento clave en cualquier proceso de ingeniería. Las medidas se emplean para comprender mejor los atributos de los modelos que se crean y evaluar la calidad de los productos de la ingeniería. Esto implica la gestión cuantitativa de los proyectos de software, mediante el seguimiento continuo de procesos y productos, con el fin de predecir su comportamiento y detectar desviaciones durante su ejecución. Las mediciones, cuando son analizadas, constituyen una base importante para una gestión efectiva por parte del equipo de desarrollo.

MEDIDAS DE CALIDAD

- Corrección: Grado en que el software realiza la función encomendada.
La medida más común es Defectos por KDLC (miles de líneas de código)
- Facilidad de mantenimiento: Facilidad con la que se puede corregir un programa si se encuentra un error, adaptarlo a un cambio de entorno, o mejorarlo ante una petición del cliente.
Medida indirecta: Tiempo medio de Cambio (TMC)
Se mide cuanto se tarda en cumplir una petición, diseñar una modificación, implementar el cambio, probar y realizar el cambio.
- Integridad: Habilidad de un sistema para resistir ataques (accidentales o intencionados) contra su seguridad tanto a los programas como a los datos y documentos. Para medir la integridad, se tienen que definir dos atributos adicionales: amenaza y seguridad. Amenaza es la probabilidad de que un ataque de un tipo determinado ocurra en un tiempo determinado. La seguridad es la probabilidad de que se pueda repeler el ataque de un tipo determinado. La integridad del sistema se puede definir como:
$$\text{Integridad} = \sum [(1 - \text{amenaza}) \times (1 - \text{seguridad})]$$

Donde se suman la amenaza y la seguridad para cada tipo de ataque.
- Facilidad de uso: La facilidad de uso es un intento de cuantificar lo amigable que puede ser el programa con el usuario. Se puede medir en función de cuatro características: Habilidad intelectual y/o física requerida para aprender el sistema. El tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema. Aumento neto en productividad, medida cuando alguien utiliza el sistema moderadamente y eficientemente. Valoración subjetiva de la disposición de los usuarios hacia el sistema, a veces obtenida mediante un cuestionario.

Bibliografía Consultada

Greiner, C. L., Dapozo, G. N., Acosta, J. C., & Chiapello, J. (2014, October). Medición de software como aporte a la gestión cuantitativa de proyectos. In XVI Workshop de Investigadores en Ciencias de la Computación. Recuperado de: <https://core.ac.uk/download/pdf/296374769.pdf>

Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

GESTION DEL RIESGO

RIESGO DEL SOFTWARE.

Robert Charette, define al riesgo como algo que:

- Afecta acontecimientos futuros, pueden hacer fracasar nuestro proyecto.
- Implica un cambio, que pueden ser de requisitos, de tecnología, de desarrollo.
- Implica una elección, de métodos y herramientas, de RH, qué importancia le daremos a la calidad.



Riesgo: Se conoce la probabilidad de ocurrencia del evento. Se busca establecer el impacto que ocasionará sobre el proyecto.

CARACTERÍSTICAS

- Incertidumbre: puede ocurrir o no. Es decir, No se conoce la probabilidad de ocurrencia del evento.
- Pérdida: es decir sí se convierte en realidad ocurrirán consecuencias no deseadas.

ESTRATEGIAS PARA GESTIONAR EL RIESGO

- Reactiva: Modo bombero. Gestión de Crisis.
- Proactiva: Plan de Contingencia. Evaluación previa de los riesgos

¿Para qué gestionar el riesgo?

- Aumentar la probabilidad y el impacto de los eventos positivos.
- Implica una elección, de métodos y herramientas, de RH, qué importancia le daremos a la calidad



El riesgo no se puede eliminar, pero sí se puede tratar de minimizarlo, para ello es muy importante identificar los “riesgos adecuados” para nuestro proyecto, para poder hacer esto es importante que los profesionales de software identifiquen todos los riesgos obvios.

Los riesgos que ocurren, no son los riesgos. Son los problemas!

Tipos de riesgo.

- Riesgos Del Proyecto: Afectan la planificación temporal y el costo. Identifican eventos potenciales en el presupuesto, calendario, personal y recursos. Ejemplos:

Miembros clave del proyecto renuncian, originando un retraso significativo
Cambio en la administración originando desconcierto en el equipo
Hardware indispensable no esta a tiempo originando retrasos
Cambio excesivo de requerimientos originando retraso y mayor costo
Se subestimó el tamaño, originando mayores costos
Se subestima el número de defectos originando retraso

- Riesgos Técnicos: Amenazan la calidad del software que hay que producir. Identifican posibles riesgos en el diseño, implementación, interfaz y mantenimiento. Ejemplos:

Riesgo
Cambio excesivo de requerimientos origina mala funcionalidad
Los componentes de software elegidos no trabajan adecuadamente
El manejador de bases de datos no soporta el volumen de transacciones
Requerimientos no verificables causan rechazo en usuarios
Algoritmo inadecuado no cumple restricciones de tiempo de respuesta

- Riesgos Del Negocio: Amenazan la viabilidad del proyecto, el cumplimiento de metas y resultados. Ejemplos:

La tecnología fundamental se sustituye por una nueva, originando dudas en la viabilidad del proyecto
Una compañía rival ofrece un producto similar antes, originando pérdida de mercado para el producto
Cambia la alta gerencia del cliente y reduce su interés en el proyecto, originando problemas financieros



MÉTODO DE GESTIÓN DE RIESGO

Creado por Barry Boehm bajo la premisa "La Gestión de Riesgos ayuda a la gente a evitar desastres, re-trabajo, excesos, y estimula situaciones gana-gana". Con esa idea desarrolló un conjunto de principios y prácticas para la gestión de riesgos. Su método hace

foco en el concepto de "exposición al riesgo" definido por el producto entre la probabilidad de obtener un resultado no satisfactorio y la pérdida debido a ese resultado. El objetivo de la gestión de riesgos es reducir la exposición al riesgo.

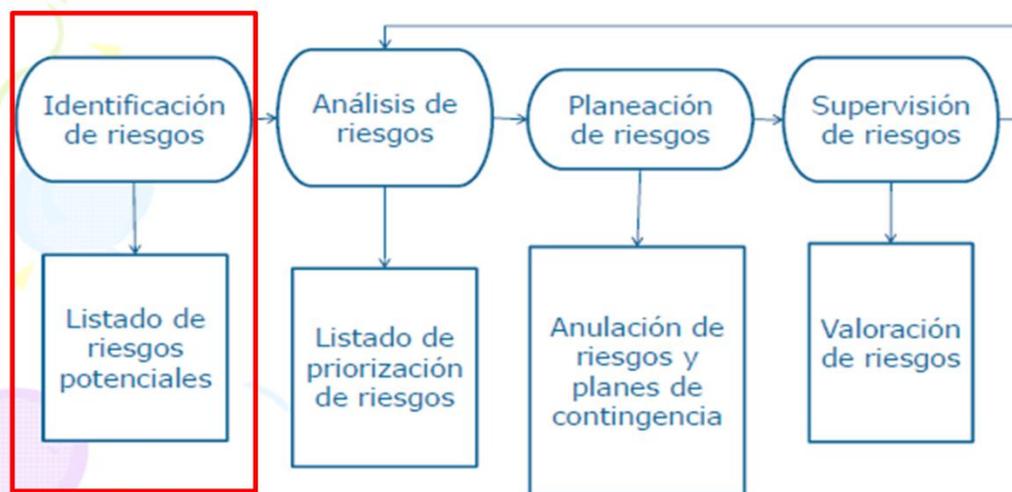
Boehm consolidó éstas técnicas de gestión de riesgo en un único método. El principal aporte de Boehm con su método fue la introducción de la disciplina de la gestión de riesgos a lo que era la gestión de proyectos a fines de la década de 1980. Este método es el primer acercamiento a la formalización de un método de gestión de riesgos en el cual se ha basado mucho del trabajo posterior.

En principio Boehm dividió la gestión de riesgos en dos aspectos fundamentales:

1. Evaluación del Riesgo: compuesto por Identificación de riesgos y Análisis de Riesgos.
2. Control del Riesgo. Compuesto por Planeación de Riesgos y Supervisión de Riesgos.

A continuación se describe el método en 4 pasos con sus respectivas técnicas:

Identificación del riesgo.



Las siguientes técnicas de recopilación de información sirven para identificar riesgos de un proyecto:

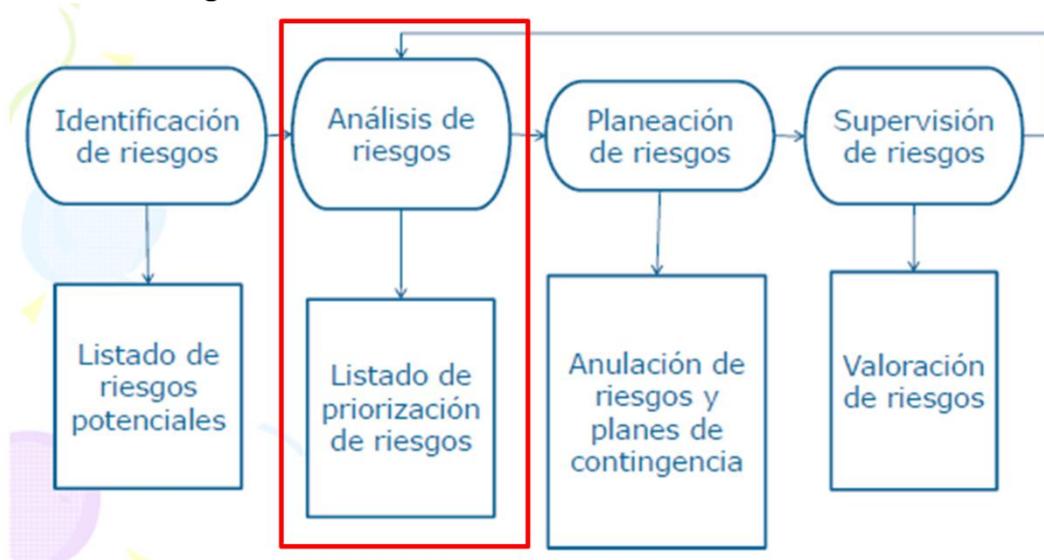
- **Tormenta de Ideas:** Técnica mediante la cual un grupo trata de generar ideas o de encontrar una solución para un problema específico mediante la acumulación de ideas de forma espontánea y sin juicio. La sesión de lluvia de ideas debe ser dirigida por un facilitador experimentado.

- **Técnica Delphi:** Se utiliza para obtener un consenso de un grupo de expertos, quienes hacen predicciones sobre futuros comportamientos. Provee entradas independientes y anónimas sobre acontecimientos futuros. Usa repetidas rondas de preguntas y respuestas por escrito y evita los posibles efectos de polarización que se presentan en los métodos orales, como es el caso de la lluvia de ideas.
- **Entrevistas:** Técnica para recopilar información por medio de encuentros cara a cara, por teléfono, vía e-mail o mensajería instantánea. Entrevistar a personas con experiencia en proyectos similares es una herramienta importante para identificar riesgos potenciales.

Clasificarlos según correspondan a:

- **Riesgos específicos del producto:** Son riesgos asociados a un producto en particular en forma única y específica. Solo son identificables por aquellos que tienen una visión clara de la tecnología, personal y entorno específico del proyecto en cuestión.
- **Riesgos genéricos:** Son todos los riesgos que afectan a cualquier proyecto. Cualquier proyecto puede presentar este tipo de problemas:
 - Tamaño del producto. (TP)
 - Impacto en el negocio. (IN)
 - Características del cliente. (CC)
 - Definición del proceso. (DP)
 - Entorno de desarrollo. (ED)
 - Tecnología a construir. (TC)
 - Tamaño y experiencia de la plantilla (TEP)

Análisis de Riesgos



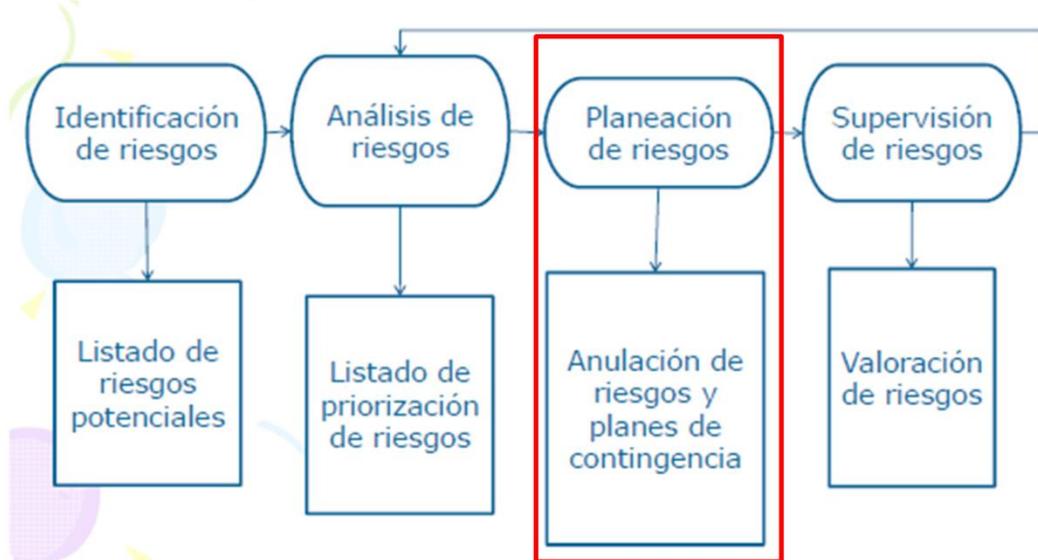
Es necesario analizar que probabilidad de ocurrencia tiene el riesgo y el impacto. La probabilidad es un valor que surge de la experiencia del equipo de proyecto.

El impacto puede ser:

- Despreciable: menos del 20% de ocurrencia
- Marginal entre 20 y 40 % de ocurrencia
- Crítico entre 41 y 69 % de ocurrencia
- Catastrófico. Mayor o igual a 70% de ocurrencia

El listado de priorización de riesgos produce un ordenamiento priorizado de los riesgos identificados y analizados.

Planeación de Riesgos

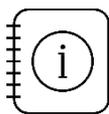


Tiene como cometido crear planes para atacar cada riesgo:

Planes para el Manejo o tratamiento del Riesgo

- EVITAR: Se modifica el plan de proyecto para evitar el riesgo. Por ejemplo, si traer una tecnología importada traerá graves problemas en el equipo que desarrolla el proyecto, “evitar” sería desestimar la utilización de esa tecnología y reemplazarla por alguna otra.
- REDUCIR/CONTROLAR/MITIGAR: la probabilidad de ocurrencia o bien su impacto si acaba materializándose. Por ejemplo: Aplicar pruebas (tests). Desarrollar prototipos
- ASUMIR: Aceptar las consecuencias. dejar establecida una política de cómo actuar en caso que ocurra el evento negativo. Por ejemplo, instrucciones de cómo seguir el trabajo en caso que exista un corte de energía.
- TRANSFERIR: trasladar el impacto negativo del riesgo hacia un tercero. Por ejemplo: si se decide traer la tecnología importada contratar un personal especializado en esa tecnología

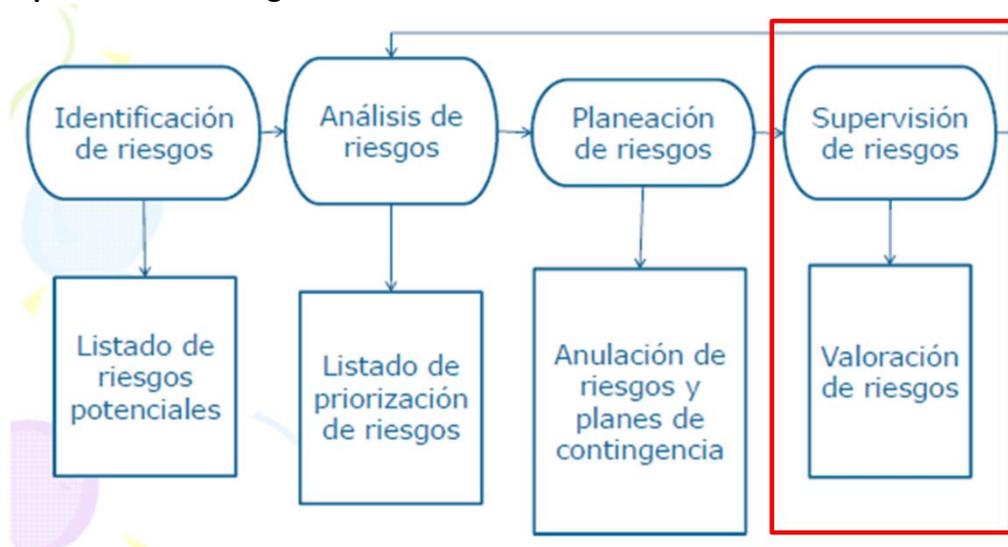
Plan de contingencia: entra en una situación en la que no es posible eliminar el riesgo. Se utiliza en situaciones en las que, por más que reduzca la probabilidad de que ocurra el riesgo, siempre habrá una posibilidad (aunque pequeña y bien monitoreada) de incidencia. De esta forma, incluso actuando preventivamente, hay que hacer más.



¿Todo riesgo necesita un Plan de Contingencia?

¡No! Esta metodología no se aplica a todos los riesgos. Lo que pasa es que algunos riesgos pueden traer impactos muy severos

Supervisión del Riesgo



Implica realizar el seguimiento del progreso del proyecto para resolver los riesgos y tomar acciones correctivas donde sea necesario.

VALORACION una forma de priorizar los riesgos para supervisar los riesgos con más valoración y mayor impacto en el proyecto.

- muy bajo(<10%),
- bajo(11-25%),
- Moderado (medio) (26-50%)
- Alto (51-75%)
- muy alto (>75%).

SUPERVISIÓN

Mediante la siguiente representación gráfica de la valoración de cada uno de los Riesgos identificados, que están en función de la probabilidad de ocurrencia y el factor de impacto que determine el equipo de proyecto, se puede determinar los riesgos que necesiten mayor supervisión.

	DESPRECIABLE	MARGINAL	CRITICO	CATASTROFICO
MUY ALTO				
ALTO				
MODERADO				
BAJO				
MUY BAJO				

PLAN DE RIESGO.

Plan RSGR constituye un mecanismo formal. A continuación se expone un esquema del Plan RSGR.

I. Introducción
1. Alcance y propósito del documento
2. Visión general de los riesgos principales
3. Responsabilidades
a. Gestión
b. Personal técnico
II. Tabla de riesgo del proyecto.
1. Descripción de todos los riesgos por encima de la línea de corte
2. Factores que influyen en la probabilidad e impacto
III. Reducción, supervisión y gestión del riesgo
n. Riesgo # n
a. Reducción
i. Estrategia general.
ii. Pasos específicos.
b. Supervisión
i. Factores a supervisar
ii. Enfoque de supervisión
c. Gestión
i. Plan de contingencia
ii. Consideraciones especiales.
IV. Planificación temporal de revisión del Plan RSGR
V. Resumen



Realizar el proceso de administración de riesgos en el proyecto «Implementación, desarrollo y mantenimiento de componentes de Web 4.0 una Biblioteca Especializada.»

1.- Identificación de Riesgos.

Riesgo	Categoría
Errores en la estimación del presupuesto	Tamaño del Producto (TP)
Cambio de políticas de gestión	Impacto de negocio (IN)
Soporte y mantenimiento	Definición de Procesos (DP)
Seguridad del Sitio	Tamaño y Experiencia de la plantilla de personal (TEP)
Desconocimiento o poco conocimiento por parte del equipo de desarrollo en la utilización de la herramientas web	TEP
Inexperiencia del equipo técnico	TEP
Dificultad de comunicación entre los miembros del grupo de desarrollo del proyecto	TEP

2.- Análisis de Riesgos.

Riesgo	Categoría	Probabilidad	Impacto
RI-06 Dificultad de comunicación entre los miembros del grupo de desarrollo del proyecto	TEP	70%	CATASTRÓFICO
RI-02 Cambio de políticas de gestión	IN	40%	CATASTRÓFICO
RI-05 Inexperiencia del equipo técnico	TEP	25%	CRITICO
LINEA DE CORTE			
RI – 03 Seguridad del Sitio	TEP	60%	DESPRECIABLE
RI- 04 Soporte y mantenimiento	DP	30%	MARGINAL
RI- 07 Desconocimiento o poco conocimiento por parte del equipo de desarrollo en la utilización de la herramientas web	TEP	20%	MARGINAL
RI – 01 Errores en la estimación del presupuesto	TP	5%	MARGINAL

3.- Planeación de riesgos

Riesgo	Probabilidad	Estrategia
RI-06 Dificultad de comunicación entre los miembros del grupo de desarrollo del proyecto	70%	Reducir o Controlar los problemas de comunicación
RI-02 Cambio de políticas de gestión	40%	Aceptar
RI-05 Inexperiencia del equipo técnico	25%	Transferir

Plan de Contingencia

RI-06 Dificultad de la comunicación entre los miembros del grupo de desarrollo del proyecto.

Plan de Acción:

Realizar talleres y actividades integradoras.

Reuniones semanales entre informáticos y bibliotecarios en donde se expresen diferencias de criterios.

Plan de Contingencia

Disparador: problemas laborales entre el equipo de desarrollo, deficiencia en la calidad del producto en las fases en donde las actividades sean multidisciplinarias. Se deberá: Contratar un personal capacitado para validar las funciones desde el punto de vista bibliotecario/informático.

Remover a los miembros del equipo que no posean una predisposición al trabajo en grupo y multidisciplinario.

4.- Supervisión y Valoración de Riesgos.

	DESPRECIABLE	MARGINAL	CRITICO	CATASTROFICO
MUY ALTO				
ALTO	RI-03 (60%)			RI-06 (70%)
MODERADO		RI-04 (30%)		RI-02 (40%)
BAJO		RI-07 (20%)	RI-05(25%)	
MUY BAJO	RI-01 (5%)			

VALORACION ALTA IMPACTO CATASTRÓFICO

RI-06 Dificultad de la comunicación entre los miembros del grupo de desarrollo del proyecto.

Supervisión: presentando informes periódicos grupales de estado de avance, en donde los miembros del proyecto, trabajan en conjunto en la elaboración del informe.

Responsable de realizar las actividades de supervisión: Jefe del Proyecto.

Recursos para la supervisión: se necesitará una metodología que abarqué informes periódicos de estados de situación del proyecto grupales.



[Entrevista con Liliana Buchtik, especialista en gestión de proyectos y riesgos.](#)



[Liliana Buchtik, Gestión de Riesgos en Proyectos en 360 grados](#)

Bibliografía Consultada

Jaureche Ballefín, S. (2012). Métodos de gestión de riesgos en proyectos de software. Recuperado de: <https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/2967/1/tesis-jaureche.pdf>

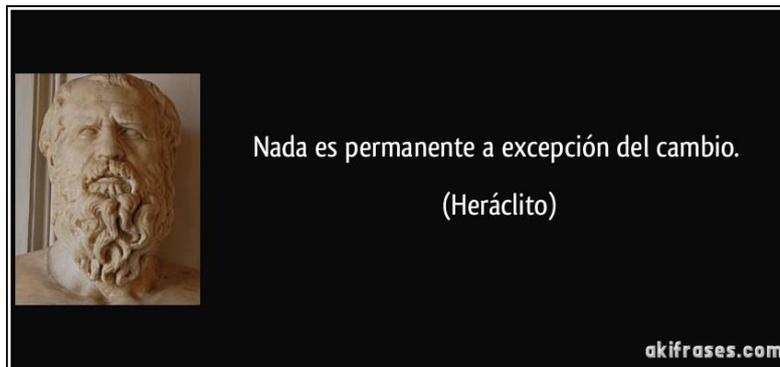
Pressman, R. (2010). Ingeniería del Software. Un enfoque práctico (7ª Edición) Ed. McGraw-Hill

GESTION DE CONFIGURACIÓN DEL SOFTWARE

El cambio es un factor que se encuentra presente durante todo el ciclo de vida del software:

- Los clientes desean modificar los requerimientos.
- El equipo de desarrollo desea modificar el enfoque técnico.
- Los gestores desean modificar el enfoque del proyecto.

Por lo que es muy importante llevar un control y registro riguroso del mismo. Este control debe ir orientado a reducir errores, aumentar la calidad y productividad; así como, evitar los inconvenientes que puedan causar una mala sincronización de los elementos de configuración del software.



La Gestión de Configuración del Software (GCS) es un proceso de ingeniería, que indica procedimientos técnicos para el control y mejora de la calidad del software.

La primera Ley de la Ingeniería de Sistemas establece:

“Sin importar en que momento del ciclo de vida del sistema nos encontremos, el sistema cambiará y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida”.

E. H. Bersoff, V. D. Henderson, S. G. Siegel

Software Configuration Management, Prentice-Hall, 1980

CONFIGURACIÓN DEL SOFTWARE: CONCEPTOS.

La gestión de la configuración del software ha sido definida por Bersoff et al. (1980) como la disciplina de identificar la configuración de un sistema en momentos determinados con el fin de controlar sistemáticamente los cambios en esta configuración y mantener la integridad y trazabilidad de esta configuración a lo largo del ciclo de vida del sistema. Por lo tanto, se basa en la creación sucesiva de líneas base, cada una de las cuales define el producto tal como existe en ese momento en el tiempo. Se debe controlar cualquier cambio en un elemento que aparece en una línea de base.

ELEMENTOS DE CONFIGURACIÓN DEL SOFTWARE. LÍNEA DE BASE.

Los elementos que componen toda la información producida como parte del proceso de Ingeniería del Software se denominan colectivamente "configuración del software". Dado que la configuración software es la única representación tangible de un programa

o sistema software, debe ser controlada para conservar su exactitud, mantener la información actualizada, y asegurar una información clara y concisa conforme avanzamos paso tras paso en el proceso de Ingeniería del Software.

Los siguientes elementos de configuración del software ECS son el objeto de las técnicas de gestión de configuraciones y forman un conjunto de líneas base:

- 1) Especificación del sistema
- 2) Plan del proyecto software
- 3) a) Especificación de requerimientos del software
b) Prototipo ejecutable o en papel
- 4) Manual de usuario preliminar
- 5) Especificación de diseño:
 - a) Diseño preliminar
 - b) Diseño detallado
- 6) Listados del código fuente
- 7) a) Planificación y procedimiento de prueba:
b) Casos de prueba y resultados registrados
- 8) Manuales de operación y de instalación
- 9) Programas ejecutables
- 10) Manual de usuario
- 11) Documentos de mantenimiento:
 - a) Informes de problemas del software
 - b) Peticiones de mantenimiento
 - c) Órdenes de cambios de ingeniería
- 12) Estándares y procedimientos de ingeniería del software

El IEEE Std. No. 610.12-1990, es el Glosario de Terminología de Ingeniería de Software y define una **línea base** como una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambios.

El concepto de línea base es similar al de una boda...

Durante la preparación del evento se pueden cambiar fechas, horas y lugares..... pero una vez mandadas las invitaciones los cambios no son tan sencillos.

En el contexto de la ingeniería de software, una línea base es un hito en el desarrollo del software; se marca al entregar uno o más ECS que se aprobaron como consecuencia de una revisión técnica.



Cuando se aprueban cambios en el proyecto que NO afectan la línea base, el director de proyecto puede aprobar la implantación del cambio.

Los cambios aprobados que requieren ajustes a las líneas base, y al plan para la dirección del proyecto, deberán pasar por un comité de control de cambios, que decidirá su aprobación.

PROCESO DE GESTIÓN DE CONFIGURACIÓN

Se resuelven en las cuatro tareas de las que consta la GCS:

- 1. Identificación.** Se trata de establecer estándares de documentación y un esquema de identificación de documentos.
- 2. Control de cambios.** Consiste en la evaluación y registro de todos los cambios que se hagan de la configuración software.
- 3. Auditorías de configuraciones.** Sirven, junto con las revisiones técnicas formales para garantizar que el cambio se ha implementado correctamente.
- 4. Generación de informes.**

La tarea de **identificación de la Gestión de Configuraciones Software** tiene tres objetivos:

1. Definir una estructura de documentación organizada de un modo inteligible y predecible. Es decir, dar un formato.
2. Proporcionar métodos para revisiones y añadir los cambios conforme se producen (Identificar cada documento para la revisión y los cambios).
3. Relacionar los cambios con “quién, qué, cuándo, porqué, cómo” para facilitar el control.

Propuesta para Identificar cada ECS es la siguiente manera:

Identificador (número, letra, ambos. No ambiguo)

Nombre (descriptivo)

Tipo (documento, código, producto de terceros, etc.)

Localización

Fecha

Versión

Estado (p.e. Para un documento En elaboración, finalizado, revisado, aceptado)

Proyecto/producto

EJEMPLO:

SPC-001-P-0-3/80	Este es el plan del proyecto 1 de la empresa "Special Purpose Computer Center". Es el documento original. Puesto bajo control de cambios en Marzo de 1980.
SPC-001-P-1-5/80	Esta es la revisión 1 al plan. Puesta bajo control de cambios en Mayo de 1980.
SPC-005-R-3-9/81	Esta es la revisión 3 de la Especificación de Requerimientos para el proyecto número 5 de SPCC. Puesto bajo control de cambios en Septiembre de 1981.

SPC: identificador de la empresa de software

001: identificador del proyecto

P ó R: es un identificador del ECS

P Plan

R Especificación de Requisitos

D Documento de diseño

S Listado fuente

T Documentación de prueba

U Manual del usuario

I Guía de instalación

M Manual de mantenimiento

0 ó 1 ó 3: es el nivel de revisión

3/80: Fecha

Control de cambios: Es un mecanismo para la evaluación y aprobación de los cambios hechos a elementos de la configuración software durante el ciclo de vida.

Pueden establecerse tres distintos tipos de control:

1) Control individual: el técnico responsable cambia la documentación como se requiere. Aunque se mantiene un registro informal de revisiones, tales registros no se ponen generalmente en el documento. El control individual se aplica durante las etapas más importantes del desarrollo del documento y se caracteriza por los cambios frecuentes antes de aprobarse un nuevo elemento.

2) Control de Gestión (u organizado): implica un procedimiento de revisión y aprobación para cada cambio propuesto en la configuración. Ocurre durante el proceso de desarrollo pero es usado después de que haya sido aprobado un elemento de la configuración software. Este nivel de control de cambios se caracteriza por tener menos cambios que el control individual. Cada cambio es registrado formalmente y es visible para la gestión.

3) Control formal: ocurre durante la fase de mantenimiento del ciclo de vida software (el producto ya está implantado). El impacto de cada tarea de mantenimiento se evalúa por un Comité de Control de Cambios (CCC), el cual aprueba las modificaciones de la configuración software. A menudo se ordena que se establezcan mecanismos de arreglo rápido (quick-fix). El procedimiento de cambios quick-fix no debe usarse para involucrar otros niveles de control de cambios, pero sí para proporcionar significados temporales

para modificación rápida de la configuración software en situaciones de emergencia. Esto es especialmente importante cuando ocurre un error considerable en el elemento y el problema deniega el acceso al cliente.

El CCC es el "órgano de gobierno" para todos los problemas relacionados con la GCS. En general, la CCC está compuesta por los miembros de las organizaciones de usuarios/pedidores de cambios y de desarrolladores.

Para pequeños proyectos, el CCC puede estar formado por uno de los representantes de los usuarios, requeridores de cambios y desarrolladores. Para grandes proyectos, el CCC puede estar organizado en una jerarquía que trate los problemas del sistema, del hardware y del software por separado

Auditorias de la configuración: Se centran en responder a las siguientes cuestiones:

1. ¿Se ha hecho el cambio especificado en la orden de cambio de ingeniería (OCI)? ¿Se han incorporado modificaciones adicionales?
2. ¿Se ha realizado una revisión técnica formal para comprobar la corrección técnica?
3. ¿Se han seguido adecuadamente los estándares de ingeniería del software?
4. ¿Se han marcado los cambios en el ECS? ¿Se han especificado la fecha y el autor del cambio? ¿Refleja la identificación del ECS los cambios?
5. ¿Se han seguido los procedimientos del GCS para señalar el cambio, registrarlo y divulgarlo?
6. ¿Se han actualizado adecuadamente todos los ECS relacionados?

Informes de estado: responde a las preguntas:

1. ¿Qué pasó?
2. ¿Quién lo hizo?
3. ¿Cuándo pasó?
4. ¿Qué más se vio afectado?



En el siguiente libro encontrará ejemplos interesantes para profundizar el tema



[Teruel, Alejandro \(2014\). Gestión de configuraciones y versiones de Software](#)

ESTÁNDARES PARA LA CONFIGURACIÓN DEL SOFTWARE.

ISO 10007:2017 Gestión de la calidad — Directrices para la gestión de la configuración: Este documento presenta orientación sobre el uso de la gestión de la configuración dentro de una organización. Es aplicable al soporte de productos y servicios desde su conceptualización hasta su disposición final.

Traducción Oficial. Recuperado de: <https://cdn.standards.iteh.ai/samples/70400/50e52548da654d9996d8a2a3d7a723fa/ISO-10007-2017.pdf>

Versión original: <https://www.iso.org/obp/ui/#iso:std:iso:10007:ed-3:v1:en>

Estándar ISO/IEC 12207:2017 : se encarga de todo lo relacionado al ciclo de vida del software, desde la conceptualización de ideas hasta la retirada y consta de procesos para la adquisición y suministro de proyectos y servicios del software, estableciendo pautas para su control y mantenimiento. Con respecto a la GESTIÓN DE LA CONFIGURACIÓN, contiene qué información se debe registrar en la documentación producida por un proceso o actividad del ciclo de vida. Consiste en aplicar procedimientos técnicos y administrativos a lo largo del ciclo de vida del software.

Información general: «ISO/IEC/IEEE 12207:2017 - Systems and software engineering -- Software life cycle processes». www.iso.org (en inglés). En <https://selenerouvier.github.io/EstandarISO12207/>

Lectura Recomendada

Gallego Velasquez, A. & Ortiz Rodriguez, P. (2011) Elaboración del Estandar de Aplicación de la Norma ISO/IEC12207 [Trabajo Final de Grado]. Recuperado de <https://repositorio.espe.edu.ec/bitstream/21000/4275/1/T-ESPE-032633.pdf>

Bibliografía Consultada

IEEE Standard 610.12-1990 Glossary of Software Engineering Terminology.

Manns, T., Coleman, M. (1996). Gestión de la configuración del software. En: Aseguramiento de la Calidad del Software. Serie de Ciencias de la Computación de Macmillan. Palgrave, Londres. https://doi.org/10.1007/978-1-349-13285-0_8

Pressman, R. (2002). Ingeniería del Software. Un enfoque práctico. Ed. McGraw-Hill

CALIDAD DEL SOFTWARE

La presencia del software en dispositivos que usamos en casa (e.g. horno de microondas, televisión plana), para comunicarnos con otras personas (e.g. teléfono inteligente, skype, blue jeans), o el software que se usa en empresas que han implementado tecnologías de la información se ha convertido en un aliado fundamental para simplificar las tareas humanas. Las tendencias actuales de la construcción de software se dirigen a abordar las necesidades particulares de los distintos tipos de usuarios y cómo el software podría modificarse para desplegarse fácilmente en nuevos entornos de operación. Desarrollar software que satisfaga a clientes y usuarios no sólo trata de lograr que el software ejecute un conjunto de funciones determinado por los objetivos del cliente, sino que también satisfaga los aspectos de calidad relevantes para clientes y usuarios. Por ejemplo, el cambio que se realiza en el software para adaptarse a nuevas plataformas de hardware, sistemas operativos, o mejoras en los procesos de negocios y operativos requiere que durante el desarrollo del software se consideren aspectos de su mantenibilidad. Además, lograr que el software facilite las tareas de interacción a un gran número de perfiles de usuario necesita mejorar su usabilidad (García Mireles, 2016).

INTRODUCCIÓN A LA CALIDAD DEL SOFTWARE.

Según el Diccionario de la lengua española (RAE, 2014), el término calidad tiene varias definiciones, entre ellas:

“Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”.

Los gurús de la calidad definen calidad de la siguiente manera:

- Edwards Deming, libro "Out of the Crisis"(1982):
“El control de calidad no significa alcanzar la perfección. Significa conseguir una eficiente producción con la calidad que espera obtener en el mercado.”
- Joseph Juran, Trilogía Planificación de la Calidad, Control de Calidad y Mejora de la Calidad (1986); definió Planificación de la Calidad como "rendimiento del producto que da como resultado la satisfacción del cliente; libertad de deficiencias en el producto, que evita la falta de satisfacción del cliente". El Control de la Calidad lo define “como un proceso que debe seguir toda empresa para asegurarse que sus productos o servicios mantengan un nivel mínimo de Calidad, el cual es definido por la propia empresa, de acuerdo a las características de lo que genera, de las características de sus clientes y de los objetivos de eficiencia que se hayan planteado y que deban alcanzar con regularidad”. Y La Mejora de la Calidad la define como el proceso de elevarse a niveles de rendimiento sin precedente.

La garantía de calidad se inicia con las acciones que se realizan durante la planificación como el conjunto de procedimientos, técnicas y herramientas durante el ciclo de vida, las actividades de auditoría tales como revisiones técnicas o inspecciones, optimizando los criterios de correctitud y completitud y las funciones de información de la gestión,

más orientadas a la documentación y al desarrollo de pruebas (Valencia Ayala et al, 2008).

El concepto de calidad de software, según Pressman (2010) se asocia a la "concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo plenamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente"

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, 1990) define calidad de software como "el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario", denotando que el énfasis radica en los requisitos específicos del sistema y en la búsqueda de la satisfacción del cliente.



[7 desastres del desarrollo de software | La importancia de la calidad](#)

NOCIÓN DE ESTÁNDARES DE CALIDAD

Según el diccionario de la RAE un estándar sirve como tipo, modelo, norma, patrón o referencia.

Los estándares de calidad de software hacen parte de la ingeniería de software, utilización de estándares y metodologías para el diseño, programación, prueba y análisis del software desarrollado, con el objetivo de ofrecer una mayor confiabilidad, mantenibilidad en concordancia con los requisitos exigidos, con esto se eleva la productividad y el control en la calidad de software, parte de la gestión de la calidad se establecen a mejorar su eficacia y eficiencia.

La Organización Internacional de Estándares (ISO, por sus siglas en inglés) ha publicado hasta hoy varios estándares relacionados con la calidad en general y también de manera particular con la calidad de software.

Organización Internacional de Estándares (ISO) y la Comisión Internacional Electrotécnica (IEC) editaron de manera conjunta la norma internacional ISO/IEC 9126.

Software Engineering Institute (SEI) es un instituto federal estadounidense de investigación y desarrollo, fundado por Congreso de los Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba al ejército estadounidense la programación e integración de los sub-sistemas de software en la construcción de complejos sistemas militares. Financiado por el Departamento de Defensa de los Estados Unidos y administrado por la Universidad Carnegie Mellon.

Es un referente en Ingeniería de Software por realizar el desarrollo del modelo SW-CMM (1991) que ha sido el punto de arranque de todos los que han ido formando parte del modelo que ha desarrollado sobre el concepto de capacidad y madurez, hasta el actual CMMI.

A continuación se mencionan algunos estándares de calidad:

Norma ISO 90003: Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software, cuyo propósito es ofrecer una guía de aplicación de la norma 9001 que pretende ser utilizada para demostrar o soportar que la entidad está en capacidad de desarrollar software con criterios de calidad.

Norma ISO/IEC 25000 (SQuaRE -Software Quality Requirements and Evaluation) es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software).

- La norma ISO/IEC 9126 de 1991, esta norma nos indica las características de la calidad y los lineamientos para su uso. Se basa en el modelo de McCall.
- El estándar ISO/IEC 14598 es actualmente usado como base metodológica para la evaluación del producto software. En sus diferentes etapas, establece un marco de trabajo para evaluar la calidad de los productos de software proporcionando, además, métricas y requisitos para los procesos de evaluación de los mismos. La norma define las principales características del proceso de evaluación: Repetitividad. Reproducibilidad. Imparcialidad. Objetividad.

Consta de seis partes:77

ISO/IEC 14598-1 Visión General: provee una visión general de las otras cinco partes y explica la relación entre la evaluación del producto software y el modelo de calidad definido en la ISO/IEC 9126.

ISO/IEC 14598-2 Planeamiento y Gestión: contiene requisitos y guías para las funciones de soporte tales como la planificación y gestión de la evaluación del producto del software.

ISO/IEC 14598-3 Proceso para desarrolladores: provee los requisitos y guías para la evaluación del producto software cuando la evaluación es llevada a cabo en paralelo con el desarrollo por parte del desarrollador.

ISO/IEC 14598-4 Proceso para adquirentes: provee los requisitos y guías para que la evaluación del producto software sea llevada a cabo en función a los compradores que planean adquirir o reutilizar un producto de software existente o pre desarrollado.

ISO/IEC 14598-5 Proceso para avaladores: provee los requisitos y guías para la evaluación del producto software cuando la evaluación es llevada a cabo por evaluadores independientes.

ISO/IEC 14598-6 Documentación de Módulos: provee las guías para la documentación del módulo de evaluación

CMMI es un modelo de buenas prácticas. Este modelo se representa de dos maneras: escalonada y continua, donde el modelo escalonado está dirigido al software y permite

clasificar las organizaciones en cinco tipos de nivel establecidos: Inicial, gestionado, definido, gestionado cuantitativamente y en optimización; y por su parte el modelo continuo se enfoca al análisis de la capacidad de cada proceso inmerso en las áreas de la ingeniería de sistemas y lo clasifica en uno de los siguientes seis niveles: Incompleto (0), ejecutado (1), gestionado (2), definido (3), cuantitativamente gestionado (4) y en optimización (5)

Bibliografía Consultada

Callejas-Cuervo, M., Alarcón-Aldana, A. C., & Álvarez-Carreño, A. M. (2017). Modelos de qualidade de software, um estado da arte. *Entramado*, 13(1), 236-250. Disponible en: <https://www.redalyc.org/journal/2654/265452747018/html/>

García Mireles, G. (2016). Conceptos básicos de calidad del software. Recuperado de <http://mat.uson.mx/mireles/conceptosCalidad/introduccion.html>

VALENCIA AYALA, LUZ ESTELA, & CABRERA, CESAR, & LÓPEZ ECHEVERRY, ANA MARIA (2008). INTRODUCCIÓN A LA CALIDAD DE SOFTWARE. *Scientia Et Technica*, XIV(39),326-331.[fecha de Consulta 24 de Agosto de 2022]. ISSN: 0122-1701. Disponible en: <https://www.redalyc.org/articulo.oa?id=84920503058>