Contenido

[Introducción ing sofware 3](#_Toc139380989)

[MODELO DE CICLOS DE VIDA (MVC) 3](#_Toc139380990)

[ESTANDAR IEEE 1074 5](#_Toc139380991)

[GESTION DEL PROYECTO 7](#_Toc139380992)

[4P: (PERSONAL- PRODUCTO- PROCESO- PROYECTO) 7](#_Toc139380993)

[PROCESO DE SOFTWARE 8](#_Toc139380994)

[Proceso de selección de MCV 9](#_Toc139380995)

[Procesos de gestión del Proyecto 9](#_Toc139380996)

[PLANIFICACION DE PROYECTOS DE SOFTWARE 10](#_Toc139380997)

[Secuenciar Actividades: 14](#_Toc139380998)

[Métricas de proyectos 15](#_Toc139380999)

[Métrica orientada al producto: 15](#_Toc139381000)

[Basada en calidad de la especificación 16](#_Toc139381001)

[Métrica orientada a clases: 16](#_Toc139381002)

[Metricas orientadas en puntos de función 17](#_Toc139381003)

[Gestion de riesgo 18](#_Toc139381004)

[Las estrategias de riesgo reactivas 18](#_Toc139381005)

[Proactivo: 18](#_Toc139381006)

[Riesgos del proyecto 19](#_Toc139381007)

[Identificador de riesgos: 19](#_Toc139381008)

[Componentes y controladores del riesgo: 19](#_Toc139381009)

[Proyección de riesgos: 19](#_Toc139381010)

[Calidad 20](#_Toc139381011)

[Modelo de MacCall: 21](#_Toc139381012)

[Control de calidad: 22](#_Toc139381013)

[Terminología: 22](#_Toc139381014)

[Aseguramiento de la calidad: 22](#_Toc139381015)

[Gestion de Configuracion de software 23](#_Toc139381016)

[Linea base 25](#_Toc139381017)

[Identificación de elementos de configuración: 25](#_Toc139381018)

[El control de versiones: 25](#_Toc139381019)

[Control de Cambios en la Configuración: 25](#_Toc139381020)

[Generación de Informes de Estado: 25](#_Toc139381021)

[Auditoría de la Configuración: 25](#_Toc139381022)

[Scrum 25](#_Toc139381023)

[Roles: 25](#_Toc139381024)

[Artefactos: 25](#_Toc139381025)

[Ceremonias: 26](#_Toc139381026)

[Cosas de examen 1: 26](#_Toc139381027)

[**-**Cosas de examen 2: 28](#_Toc139381028)

# Introducción ing sofware

Un software es un conjunto de programas, procedicimientos y datos asociados que forman parte de una operación en el sistema

Primero tiene instrucciones con funciones, luego su estructura para manipular la información bien y por último esa información tiene que describir las operaciones y su uso

Método: Modo ordenado de proceder hasta llegar al resultado o fin

Herramienta: Instrumento de ayuda

Actividad: Conjunto de tareas o acciones para realizar un objetivo

Proceso: Serie de pasos para accionar

Ingeniería del Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales

Diagrama, Texto

Descripción generada automáticamente

Icono

Descripción generada automáticamente con confianza media

## MODELO DE CICLOS DE VIDA (MVC)

**Cascada**:

*Diagrama

Descripción generada automáticamente*

- Se debe completar una epata para pasar a la otra

- No se puede volver

- Es difícil obtener los requisitos al principio

- No hay versión hasta avanzado el proyecto

- Errores muy costosos

**Incremental**

Diagrama

Descripción generada automáticamente

* No se tiene todos los requerimientos desde el inicio
* Se entregan funcionalidades completas en cada incremento
* Primero elementos fundamentales o básicos de evaluar

**Evolutivo**:

Evoluciona con el tiempo

* Se crea produto sin todas sus tareas para poder salir lo antes posible
* Se tiene en claro las funciones básicas, pero no a detalles
* Los requerimientos cambian con el tiempo de desarrollo

Diagrama

Descripción generada automáticamente

# ESTANDAR IEEE 1074

Es un estándar para el desarrollo de procesos de ciclos de vida del software

Diagrama

Descripción generada automáticamente

Como se ve en la imagen se puede dividir en 3 cuadros

Procesos de **Gestión del proyecto**: Crean la estructura del proyecto y aseguran el nivel apropiado de gestión de este durante todo el ciclo de vida del software

Actividad **Orientada al Desarrollo del Software**: Produce, instala, operan y mantienen al software y lo retiran de su uso. Se clasifican en subgrupos

* Pre-Desarrollo: Se deben realizar antes que haya un desarrollo propiamente dicho del software
* Desarrollo: Deben realizar para la construcción del producto de sofware
* Postdesarrollo: Últimas fases del ciclo de vida del software

**Integrales del proyecto**: Importantes para completar con éxito las actividades del proyecto. Simultáneos a las actividades orientadas al desarrollo de software

Estructura de norma IEEE 1074:

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

**Los grupos de Act de Gestion del Proyecto** --> Ponen condiciones para desarrollar el proyecto, lo inician, asignan recursos, planifica, siguen y controlan el ciclo de vida del software

**Los Grupos de Actividades Orientadas al Desarrollo** -> inician el esfuerzo de desarrollo con la identificación de las necesidades automatización. Una app nueva o cambio de todo o parte de la app puede ser necesario para satisfacer necesidades. Son apoyo para el Grupos de Actividades Integrales. Por último, hacen instalaciones, operación, soporte, mantenimiento y retiro del producto

**Los Grupos de Actividades Integrales** -> son simultáneos y complementarios a los procesos orientados al desarrollo. Incluye las actividades que son esenciales para asegurar que el sistema construido es fiable y se utiliza al máximo de sus posibilidades

# GESTION DEL PROYECTO

Administrar: Ordenar, disponer, organizar bienes. Gobernar, ejercer autoridad o a mando. Suministrar, distribuir algo

## 4P: (PERSONAL- PRODUCTO- PROCESO- PROYECTO)

People-CMM -> [People Capability Maturity Model](https://www.bing.com/ck/a?!&&p=40a74e5783abf1beJmltdHM9MTY4MTc3NjAwMCZpZ3VpZD0yNzIxZTI1Yi01NDgxLTY2YmYtMTQyYi1mMGJiNTUwYjY3ODUmaW5zaWQ9NTE4NQ&ptn=3&hsh=3&fclid=2721e25b-5481-66bf-142b-f0bb550b6785&psq=People-CMM&u=a1aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUGVvcGxlX0NhcGFiaWxpdHlfTWF0dXJpdHlfTW9kZWw&ntb=1)

**Personal**: Mejorar habilidades, motivar, desarrollar, mejorar y conservar la fuerza de trabajo

Define áreas claves como:

■ Ambiente de trabajo / Clima

■ Capacitación

■ Competencias

■ Desarrollo profesional

■ Bonificación por desempeño

■ Beneficios

Los que participan son: Gerentes ejecutivos (influencia del proyecto), gerentes del proyecto(administra), profesionales(aportan conocimientos), clientes(da especificaciones) y usuarios finales (quien interactúa)

Requiere mucho trato con la gente

● Habilidades blandas

● Modelo MOI para liderazgo técnico

* ○ Motivación
* ○ Organización
* ○ Ideas/innovación Los lideres equipo

● Gerentes de proyecto

* ○ Resolución de problemas
* ○ Identidad
* ○ Incentivos por logro
* ○ Influencia y construcción de equipo

Equipo SW (Software)

Cada equipo tiene una estructura y depende de: ○ Organización ○ Dificultad del problema ○ Tamaño del producto ○ Tiempo ○ Grado de modularización ○ Rigidez de la fecha ○ Grado de sociabilidad/comunicación

**Producto**: Primero se piensa en el producto antes del proyecto, objetivos, analizar alternativas, restricciones y su descomposición

Sino es muy difícil hacer estimaciones precisas:

○ Costos

○ Riesgos

○ Tiempos

**Proceso**: Proporciona marco conceptual, buscan el más adecuado para adaptar (PERSONA, PRODUCTO Y ENTORNO), variar las actv correctamente

**Proyecto**: Se plantea y controla, el Gerente del proyecto es responsable de Planificar, Monitorear y Controlar. Establecer indicadores que notifiquen ciertos parámetros como por ej: Alertas

Gráfico, Gráfico circular

Descripción generada automáticamente

# PROCESO DE SOFTWARE

Estos procesos de usan como una guía normativa sobre la documentación a producir para enviar al cliente. Buscan herramientas, técnicas y métodos más apropiadas para las diferentes actividades. También existen marcos para analizar patrones, asignaciones y consumo de recursos a lo largo del todo el ciclo, analizar la producción, costo y calidad del software

Estándar IEEE 1074-1989 marca conjuntos de actv a agregar al MCV como:

* descripción de cada actividad
* Informe de entrada y salida para cada actv
* Fuente y destino de la información
* Productos obtenidos por el desarrollo de cada actividad

descripción global del proceso

El proceso base de construcción de software consiste en analizar las necesidades de la organización en un dominio, desarrollar una solución que las satisfaga y posteriormente reinsertar la solución en el dominio, bajo un marco de gestión, seguimiento, control y gestión de calidad

Diagrama

Descripción generada automáticamente

Proceso de selección de MCV del producto: Identifica y selecciona un MCV para crear

Proceso de gestión del proyecto: crean la estructura del proyecto y aseguran el nivel apropiado de la gestión de este durante todo el ciclo de vida del software.

Procesos orientados al desarrollo del software: producen, instalan, operan y mantienen el software y lo retiran de su uso. Se clasifican en procesos de pre-desarrollo, desarrollo y post-desarrollo.

Procesos integrales del proyecto: son necesarios para completar con éxito las actividades del proyecto software. Aseguran la terminación y calidad de las funciones de este. Son simultáneos a los procesos orientados al desarrollo del software e incluyen actividades de no desarrollo

## Proceso de selección de MCV

Actividades que realizar: Identificar los posibles modelos de ciclo de vida Seleccionar el modelo más adecuado para el proyecto

Documentos de salida Modelo de ciclo de vida seleccionado. (esto incluye la justificación de la elección que debe quedar documenta)

## Procesos de gestión del Proyecto

Involucra actividades de planificación, estimación de recursos, seguimiento y control y evaluación del proyecto

La **planificación** de proyectos se define como la predicción del a duración de las actividades y tareas a nivel individual, los recursos requeridos, la concurrencia y solapamiento de tareas para ser desarrollados en paralelo y el camino crítico a través de la red de actividades.

La **estimación** se define como la predicción de personal, esfuerzo y costos que se requerirá para terminar todas las actividades y productos conocidos, asociados con el proyecto.

[…VER PROCESO DE SOFTWARE…]

# PLANIFICACION DE PROYECTOS DE SOFTWARE

Antes de que el proyecto inicie, el gestor y equipo deben realizar una intimación del trabajo a realizar. **Recursos**, el **tiempo** de inicio a fin para realizar. Siempre se estima mirando hacia el futuro y aceptamos un grado de incertidumbre

Para estimar recursos, **costos** y **planificación** se necesita experiencia, tener información histórica y coraje de confiar en las predicciones. La complejidad causa incertidumbre, pero es una relatividad que se afecta por las experiencias pasadas

El tamaño de proyecto es importante para acertar la precisión y eficiencia de la estimación.

*“La complejidad del proyecto, el tamaño del proyecto y el grado de incertidumbre estructural afectan a la fiabilidad de la estimación”*

El objetivo de planificar es dar un marco de trabajo para gestionar razonablemente los recursos, costos y planificaciones temporales (En un tiempo limitado) que estos deben actualizarse regularmente a medida que avanza el proyecto

La primera tarea para **planificar** es determinar el ámbito del software. Evaluar funciones y rendimiento así no es ambiguo, incomprensible para los directivos y técnicos

Diagrama

Descripción generada automáticamenteLa segunda tarea es la **estimación** de los recursos requeridos para acometer el desarrollo del software

Para simplificar, es necesario tener los recursos, actividades, costos,

riesgos, esfuerzos (Precisión), limitaciones y complejidad

Diagrama

Descripción generada automáticamente

**Recursos**: Descripcion, Enunciados, Cuando y

duracion

**Actividades**: Establecer ámbitos, factibilidad, riesgos, definir recursos (Humanos, SW reutilizable, ambientales), Costos (Descompones, dimensionar), Calendario (Tareas, red, herramientas calendarías, mecanismos de seguimiento)

Descomposición: **ETD** (Estructura de Desglose de Trabajo): Organigrama jerárquico del proyecto donde se subdivide el mismo en menores componentes

El de más bajo nivel se llama “Paquete de trabajo”

● Consiste en dividir en componentes menores para facilitar la planificación y control.

● Es un organigrama jerárquico donde se va subdividiendo en menores componentes.

● El nivel más bajo se denomina paquete de trabajo, generalmente se agrupan bajo “Paquetes de control”.

● No define secuencia

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Tabla

Descripción generada automáticamente

Definir Actiidades: Generar una lista de de Actividades y de Hitos

Tabla

Descripción generada automáticamente

Estimación: No es exacto, hay muchas variables, se asume un riesgo aceptable (costo), empírico. Además de estimar recursos y ascendentes

La estimación del costo y esfuerzo se basa en el problema, proceso y/o caso de uso

Cuando la estimación proviene de actv más difíciles y con mas esfuerzo, es riesgo es mas grande. Podría haber un juicio experto, estimación análoga, análisis de reserva, paramétrica o por 3 valores/matemáticas

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

Tabla

Descripción generada automáticamente con confianza media

Cronograma: Acortar el cronograma sin afectar el alcance, ejecución mas rápida (fasttracking); Realizar tareas en paralelo. Comprensión (Crashing): Agrega recursos para acortar el tiempo

Diagrama

Descripción generada automáticamente

## Secuenciar Actividades:

tipo de dependencias:

● Dependencias obligatorias: también conocidas como "Lógica Dura" Una dependencia obligatoria es inherente a la naturaleza del trabajo (ejemplo: se debe diseñar antes de construir) o es requerida por el contrato.

● Dependencias discrecionales: también conocidas como "Preferida, Preferencial o Lógica Blanda". Existen otras formas en que se podría realizar el trabajo, pero este es el enfoque preferido. Puedes cambiar una dependencia discrecional si es necesario, mientras que no puedes cambiar fácilmente los otros tipos de dependencias.

● Dependencia externa Esta dependencia se basa en las necesidades o deseos de una parte externa del proyecto (ejemplo: gobierno o proveedores).

● Dependencia interna Esta dependencia se basa en las necesidades del proyecto y puede ser algo que el equipo del proyecto puede controlar

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# Métricas de proyectos

La medición permite tener una visión profunda para una evaluación objetiva. Se puede utilizar para ayudar en la estimación, control de calidad, evaluación productiva y control de proyectos

Hay cuatro razones para medir los procesos del software, los productos y los recursos:

* caracterizar
* evaluar
* predecir
* mejorar

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Evaluamos para determinar el estado con respecto al diseño. Las medidas utilizadas son los sensores que nos permiten conocer cuándo nuestros proyectos y nuestros procesos están perdiendo la pista. También evaluamos para valorar la consecución de los objetivos de calidad y para evaluar el impacto de la tecnología y las mejoras del proceso en los productos y procesos. Predecimos para poder planificar, esto porque queremos establecer objetivos alcanzables para el coste, planificación, y calidad así se puedan aplicar recursos apropiados

Los proyectos y **estimaciones** que están basadas en datos históricos ayudan a analizar y realizar mejor. Medimos para mejorar, identificar obstáculos, problemas de raíz, ineficiencias y otras

Veremos 3 tipos de métricas: **De producto, de proceso, de proyecto**

## Métrica orientada al producto:

Ayudan a los ingenieros de software a obtener comprensión acerca del **diseño y la construcción del software** que elaboran, al enfocarse en **atributos** mensurables específicos de los productos de trabajo

de la ingeniería del software.

Basada en funciones: La *métrica de punto de función (PF) puede usarse de manera efectiva como medio para medir la* funcionalidad que entra a un sistema. Al usar datos históricos, la métrica PF puede entonces usarse para:

1. estimar el costo o esfuerzo requerido para diseñar, codificar y probar el software
2. predecir el numero de errores que se encontraran durante las pruebas, y
3. prever el numero de componentes y/o de líneas fuente proyectadas en el sistema implementado.

Texto

Descripción generada automáticamente con confianza mediaBasada en calidad de la especificación:

**nr: Requerimientos en una especificación**

**nf: Requerimientos funcionales**

**nnf: Requerimientos no funcionales (por ejemplo rendimiento)**

**Q: Especificidad**

**nui: Números de requerimientos para los cuales los revisores tienen interpretación idénticas.**

## Métrica orientada a clases:

Métodos ponderados por clases MPC

Profundidad de árbol de herencia PAH: La máxima longitud desde el nodo hasta raíz del árbol

Diagrama

Descripción generada automáticamenteMétricas de proyecto:

* Auxiliar la estimación
* Control de calidad
* Valoración de productividad
* Control del proyecto
* Métricas de proceso:
* Productividad
* Aseguramiento de la Calidad

Las métricas de software orientadas a tamaño se derivan al normalizar las medidas de calidad y/o productivadad para considerar el tamño del software que se construyó.

Ventajas:

Muy simple de conseguir

Desventajas:

Depende del lenguaje de programación

Los programas bien estructurados y que reutilizan mucho código se ven perjudicados

## Metricas orientadas en puntos de función

Usan una medida de la funcionalidad como valor de normalización

Ventaja: Independiente del lenguaje

Desventaja: Utiliza valores subjetivos

Aunque muchas medidas de calidad pueden recopilarse, el empuje primario en el nivel del proyecto es medir los errores y los defectos

Las métricas derivadas de estas medidas proporcionan un indicio de la efectividad de las actividades de aseguramiento y control de la calidad del software.

Métricas posibles derivadas de las medidas:

1. Errores de producto operativo por punto de función
2. Errores descubiertos por hora de revisión
3. Errores descubiertos por prueba por hora

\* Amenza: Probabilidad de ocurrir un ataque

Seguridad: Probabilidad de repeler el ataque

Eficiencia de la remoción de defecto es una métrica que nos indica la capacidad del equipo para detectar y corregir errores.

Imagen que contiene Diagrama

Descripción generada automáticamenteSiendo E la cantidad de errores detactado en las distintas etapas de desarrollo, y D los defectos que fueron detectados por el cliente una vez entregado el producto, idealmente el valor ideal de ERD es 1.

Sabiendo que nunca va a ser 1 porque D nunca va a ser 0, lo ideal es tratar que ERD al menos tienda a 1 en la medida de lo posible. Notar que cuanto mas errores detecte el equipo de desarrollo, por mas defectos que se encuentren en producción, la eficiencia de remoción de defectos se mantendrá adecuada.

Establecimiento de una línea de referencia

La línea de referencia consiste en los datos recopilados a partir de los proyectos de desarrollo de software.

Para ser un auxiliar efectivo en el mejoramiento del proceso y/o estimación de costo y esfuerzo los datos de la línea de referencia deben tener los siguientes atributos:

1. Precisos
2. Considerar tantos proyectos como sea posible
3. Consistentes (por ej. Al interpretar una línea de código)
4. Las aplicaciones deben ser similares al trabajo que debe estimarse

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

# Gestion de riesgo

## Las estrategias de riesgo reactivas

se le hizo humor en películas de Indiana Jones, cuando se enfrentaban a algo insuperable. Entonces la mayoría de los equipos de software confían solamente en las estrategias de riesgo reactivas

## Proactivo:

Una estrategia mas inteligente para el control de riesgos es ser **proactivo.**

Esta estrategia comienza antes que los trabajos tecnicos. Se identifican riesgos potenciales, probabilidades, impacto y establece prioridad a su importancia

En los riesgos de software destacan la incertidumbre y la perdida. Es importante cuantificar el nivel de incertidumbre y el grado de perdida por cada riesgo

## Riesgos del proyecto

Los **riesgos del proyecto** amenazan el plan del proyecto, es decir, haciéndose realidad es probable que se retrase y los costos aumenten, problema de presupuestos, planificación temporal, personal (asignación y organizacion), recursos, cliente y requisitos

Los **riesgos técnicos**

amenazan la calidad y planificación temporal, si uno técnico se hace realidad puede llegar a ser difícil o insuperable, como problemas al diseño, implementación, interfaz, verificación y mantenimiento

Los **riesgos de negocions**

amenazan con viabilidad del software a construir, esto pone en peligro el proyecto o producto

Los **riesgos conocidos**

son todos aquellos que se descubren después de una evaluación del plan del proyecto, entorno técnico y comercial

Los **riesgos predecibles**

son de la experiencia de proyectos anteriores

También hay **riesgos imposibles de predecir,** el comodin de la baraja, difíciles de identificar por adelantado

## Identificador de riesgos:

Es un intento sistemático para identificar amenazas al plan del proyecto, un método seria la lista de comprobación de elementos de riesgo, se utiliza para identificar riesgos y enfoca en un subconjunto de riesgos conocidos y predecibles en las categorías genéricas (tamaño del producto, impacto de negocio, características del cliente, definición del proceso, entorno de desarrollo, tecnología a construir, tamaño y experiencia de la plantilla)

## Componentes y controladores del riesgo:

Controladores del riesgo que afectan a los componentes de riesgo software (rendimiento, coste, soporte y planificación temporal)

## Proyección de riesgos:

Intenta medir cada riesgo en dos maneras, probabilidad y consecuencia, si ocurren:

Establece una escala que refleje la probabilidad, definen las consecuencias, estiman el impacto en el proyecyo y producto, apuntan la exactitud general de la proyección del riesgo para que no haya confusiones

Diagrama

Descripción generada automáticamenteDesarrollo de una tabla de riesgos: lista todos los riesgos, categoriza segunda columna, la probabilidad de aparición, el valor de la probabilidad de cada riesgo se estima por cada miembro del equipo individualmente y luego se valora el impacto de cada riesgo. La tabla es ordenada por probabilidad y por impacto, el jefe del proyecto estudia la tabla ordenada resultante y define una línea de corte. Todos los riesgos que se encuentran por encima de la línea de corte deben ser considerados

Evaluación de impacto de riesgo: Su naturaleza, alcance y cuando ocurre. Indica los problemas probables que aparecen si ocurriera. El **alcance** de un riesgo combina la severidad (¿cómo de serio es el problema?), con su distribución general (¿qué proporción del proyecto se verá afectado y cuántos clientes se verán perjudicados?). Finalmente, la temporización de un riesgo considera cuándo y por cuánto tiempo se dejará sentir el impacto.

Evaluación del riesgo: En este punto del proceso de gestión de riesgo hemos establecido un conjunto de ternas de forma CHA89 [r1, l1, y1]

donde r1 es el riesgo. l1 es la probabilidad del riesgo y1 es el impacto del riesgo

Reducción, Supervisión y Gestion de riesgo:

Evitar riesgos, supervisar riesgos, gestionar y platinificar contingencias

# Calidad

La calidad es relativa, depende de los requisitos o necesidades que se necesitan satisfacer. Entonces va a implicar comparar requisitos preestablecidos y el producto a desarrollar, aunque el problema de esto es que una parte de los requisitos estará explicita y la otra implícita, siempre hay que intentar que se reduzca lo implícito para una mejor calidad y entendimiento

**La calidad en desarrollo del software** incluye que el grado de diseño cumpla funciones y características especificadas cumpliendo sus metas y desempeño.

**Calidad de producto:** Encontrar un conjuno de propiedades que nos den un indicador de calidad

**Calidad del proceso**: Construir calidad durante el proceso de su desarrollo

Diagrama

Descripción generada automáticamente

Factores de calidad (Revision, transición, operacion)

## Modelo de MacCall:

Diagrama

Descripción generada automáticamente

Operación: Que funcione bien

Transición: Como se adapta a los cambios y evoluciones

Revisión: El fácil entendimiento de quien prueba el software

Criterios de calidad

* Usabilidad: fácil de operar y aprender
* Integridad: Control de acceso y auditoria
* Corrección: Consistencia
* Confiabilidad: Tolerancia a fallos
* Eficiencia: En su ejecución y rapidez de almacenamiento
* Facilidad de mantener: Modularidad, consistencia
* Fácil a probar: Simplicidad y modularidad
* Flexible: Capacidad de expandirse y generalidad
* Reusable: Independiente del harware y sistema
* Interoperabilidad: Compatible con comunicaciones y compartibilidad de datos
* Portable: Independiente del harware y sistema

¿Como lograr la calidad?

Métodos de ing del software, administrar bien el proyecto, realizar control de calidad y contar con infraestructura de aseguramiento de calidad

## Control de calidad:

Comprobar si el producto posee o no una determinada característica de calidad en el grado requerido. Con controles estáticos (revisiones) o controles dinámicos (Prueba de software)

Diagrama

Descripción generada automáticamente

## Terminología:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Aseguramiento de la calidad:

Actv. Especificas del proceso que permiten a una organización garantizar que hace “ cosas correctas en el momento correcto de forma correcta”

La garantía de calidad del software es una actividad de protección que se aplica a lo largo de todo el proceso del software.

Las decisiones administrativas pueden afectar significativamente en la calidad del producto. Por ejemplo pensemos en los efectos de una mala estimación.

Aseguramiento de calidad:

**Estándares**.

El IEEE, ISO y otras organizaciones que establecen estándares han producido una amplia variedad de ellos para ingeniería de software y documentos relacionados.

Los estándares los adopta de manera voluntaria una organización de software o los impone el cliente u otros participantes. El trabajo del ACS es asegurar que los estándares que se hayan adoptado se sigan, y que todos los productos del trabajo se apeguen a ellos.

**Revisiones y auditorías**.

Las revisiones técnicas son una actividad del control de calidad que realizan ingenieros de software para otros ingenieros de software. Su objetivo es detectar errores. Las auditorías son un tipo de revisión efectuada por personal de ACS con objeto de garantizar que se sigan los lineamientos de calidad en el trabajo de la ingeniería de software. Por ejemplo, una auditoría del proceso de revisión se efectúa para asegurar que las revisiones se lleven a cabo de manera que tengan la máxima probabilidad de descubrir errores.

**Pruebas**.

Las pruebas del software son una función del control de calidad que tiene un objetivo principal: detectar errores. El trabajo del ACS es garantizar que las pruebas se planeen en forma apropiada y que se realicen con eficiencia, de modo que la probabilidad de que logren su objetivo principal sea máxima.

**Colección y análisis de los errores**.

La única manera de mejorar es medir cómo se está haciendo algo. El ACS reúne y analiza errores y datos acerca de los defectos para entender mejor cómo se cometen los errores y qué actividades de la ingeniería de software son más apropiadas para eliminarlos.

**Administración del cambio**.

El cambio es uno de los aspectos que más irrumpe en cualquier proyecto de software. Si no se administra en forma adecuada, lleva a la confusión y ésta casi siempre genera mala calidad. El ACS asegura que se hayan instituido prácticas adecuadas de administración del cambio.

# Gestion de Configuracion de software

Diagrama

Descripción generada automáticamente

Es una actividad autoproteccion que se aplica durante el proceso del software. Como el cambio puede ser en cualquier momento, las actividades **GCS sirven para** identificar cambios, controlar cambios, garantizar que el cambio sea adecuado e informar del cambio a todos los que estén interesados

**Su éxito depende de:**

* + Gestion del proyexto
  + Desarrollo técnico
  + Sistema de calidad
  + Sist de gestión de la confuguracion

El objetivo es la **integridad**:

* + Satisfacer al usuario
  + requisitos de rendimiento
  + un trazo evolutivo

**Mantenimiento de software:**

es un conjunto de actividades de ingeniería del software que se producen después de que el software se haya entregado al cliente y esté en funcionamiento.

**GCS**: es un conjunto de actividades de seguimiento y control que comienzan cuando se inicia el proyecto de ingeniería del software y termina sólo cuando el software queda fuera de la circulación.

Para controlar y gestionar los elementos de configuración, se debe identificar cada uno de forma Única y luego organizarlos mediante un enfoque orientado a objetos:

Se pueden identificar dos tipos de objetos [CH089]: objetos **básicos** y objetos **compuestos**. Un objeto básico es una «unidad de texto» creado por un ingeniero de software durante el análisis, diseño, codificación o pruebas. Por ejemplo, un objeto básico podría ser una sección de una especificación de requisitos, un listado fuente de un módulo o un conjunto de casos prueba que se usan para ejercitar el código. Un objeto compuesto es una colección de objetos básicos y de otros objetos compuesto

## Linea base

Para controlar los cambios sin impedir los cambios justificados se utiliza el concepto de **LÍNEA BASE**. Se puede definir una línea base como un punto de referencia en el proceso de desarrollo del software que queda marcado por la aprobación de uno o varios Elementos de Configuración del Software, mediante una revisión técnica formal (**Proceso formal**).

## Identificación de elementos de configuración:

(CI, por sus siglas en inglés) Esta actividad implica identificar y seleccionar los elementos del proyecto que serán gestionados como parte de la configuración. Estos elementos pueden incluir archivos de código fuente, documentos, bases de datos, artefactos de diseño, bibliotecas de software, entre otros.

## El control de versiones:

Combina procedimientos y herramientas para gestionar las versiones de los objetos de configuración creados durante el proceso del software

## Control de Cambios en la Configuración:

Consiste en controlar las versiones y entregas de un producto y los cambios que se producen en él a lo largo del ciclo de vida

## Generación de Informes de Estado:

Consiste en informar acerca del estado de los componentes de un producto y de las solicitudes de cambio, recogiendo estadísticas acerca de la evolución del producto.

## Auditoría de la Configuración:

Consiste en validar la completitud de un producto y la consistencia entre sus componentes, asegurando que el producto es lo que el usuario quiere.

# Scrum

Es un modelo de desarrollo ágil identificado en los años 80, surgió para empresas tecnológicas, pero se usa para proyectos inestables que requieren rapidez y flexibilidad. Aplicado en 1993 por primera vez, en 1996 presentado y finalmente en 2001 promulgado del manifiesto Ágil

Entonces, **Scrum es un framework o marco de trabajo que será el andamiaje que ayuda a encontrar, iteración a iteración, el mejor proceso posible dada la realidad y potencia que tengan**

## Roles:

* Product Owner: Táctica, maximizar el ROL, Vision y Proprizacion
* Equipo: Estrategia, Poseer todos los skills, multidisciplinario, Auto-Organizado, no define roles
* Scrum Master: Preguntas, responsales del proceso, facilitador, mentor

## Artefactos:

* Product Backlog: PO, Lista de req (PBI), CU-HU, Cada PBI aumenta el valor del producto, ordenados para maximizar valor, estimación
* Sprint Backlog: Equipo, realizar durante Sprint, Definida por el PO
* Burndown chart: Scrum Master, muestra trabajo pendiente del Sprint, visible y publico
* Release Backlog: PO, Basada en PB, Hitos

## Ceremonias:

* Sprint Plannig: PO+EQUIPO/4-8hs, PB proparado, selecciona PBI a realizar en el sprint, PO disponible, equipo decide como crear tareas y crea SB
* Daily: Equipo/15’, termómetro diario, revisar compromisos, 3 preguntas, dar visibilidad
* Sprint Review: Equipo-PO/2-4hs, Solamente el incremento, No es una demo, Feedback, evalua el “Que”
* Retro: Equipo-SM-PO/3hs, reflexión de sprint, fortaleza y Op de mejoras, bueno, lo malo, acciones, evaluar el “como”

Diagrama

Descripción generada automáticamente

# Cosas de examen 1:

Son tipos de estructuras de equipo: **Descentralizado democrático, Descentralizado controlado, Centralizado controlado**

En un proyecto Soft la relación gente trabajo es una **relación no lineal**

Los proceso y actividades para la gestion de proyectos minimizan la probabilidad de fracaso garantizando el exito del proyecto. **Verdadero**

Caracteristica/s de los distintos tipos de recursos que se analizan en la planificación: **Todas las opciones(Componentes en la nube,oficinas,entornos,habilidades personales, compotenentes reutilizables)**

Marque la opción incorrecta sobre ETD “Estructura de desglose de trabajo” (WBS): **El nivel más alto de cada división se llama paquete de trabajo (Es erroneo ya que el nivel mas bajo se llama asi)**

Cuando se selecciona un modelo de Proceso, se debe asegurar de implementar todos los subprocesos y realizar todas las actividades que el modelo propone. **Verdadero?** (ver IAP las 4 P, pdf)

Los tipos de dependencias entre tareas pueden ser: **Mandatorias, discrecionales, externas e internas**

Un equipo de alto rendimiento se caracteriza por: **Confianza entre sus miembros**

La selección de un ciclo de vida implica el establecimiento de un orden de ejecución de las distintas actividades, marcadas por el proceso e involucradas en el proyecto. **Falso**

Los beneficios de descomponer el problema son:**Facilita la estimación , Entender que se debe hacer y como se va a hacer , Aumenta la probabilidad de exito**

Un marco de proceso me define las actividades esenciales con un propósito específico que puedo utilizar dentro del ciclo de vida del proyecto, pero no me define ni obliga a utilizar herramientas. **Verdadero**

Para una exitosa gestion de Proyectos se sugiere: **Hacer un buen seguimiento del progreso en todo momento**

Camino crítico es: **Las actividades que de demorarse, se posterga la fecha de fin de proyecto (Si se atraza aumenta la cantidad maxima)**

En un MCV Incremental, marque la opción correcta: **Se entregan funcionalidades completas en cada incremento**

Existe un modelo de madurez para las personas llamado People-CMM. **Verdadero**

Para la gestion de proyectos un proceso debe tener una salida explicita bien definida, pero puede carecer de entradas. **Falso**

Es una forma de estimación: **Paramétrica, 3 Valores (Pert), Juicio experto**

La planificacion puede llevarse a cabo con: **Fecha resultante (Que se da calculando y estimando) o fecha estipulada (fecha limite en donde se termina)**

En relacion con la metodología para seleccionar el MCV visto en clase, marque la opción correcta: **El valor mas alto es MCV recomendado para utilizar (Es el calculo con la matriz)**

Seleccione la opción mas se acerque a la definición “Reunión e planificación” (Planning Meeting) **Se realiza al inicio de cada Sprint donde se define que se va a construir, como, y se define el incremento**

Diferencia entre Fast Tracking con Crashing es: **Fast tracking es la paralelización de tareas del camino critico y Crashing es adjuntar o incrementar recursos mientras se comprimen los tiempos planificados siempre manteniendo el alcance original (Fast hace tarea a la vez y Crashing mismo objetivo pero se agregan mas cosas pero se apretan tiempos)**

Características para un producto con calidad de Soft: **Satisfaion de usuario, cumplimiento de requisitos**

El control de calidad afecta directamente al: **Producto**

¿Qué factor de calidad determina le “Grao en el que es posible controlar accesos personales no autorizados”? **Integridad**

# **-**Cosas de examen 2:

¿Que características no pueden faltar para considerar un producto de software de calidad? **Satisfacción del usuario y cumplimiento de requisitos**

Que factor de calidad determina el “Grado en el que es posible controlar el acceso a personas no autorizadas al software o datos” **Integridad**

El control de calidad afecta directamente: **Al producto**  
Una Métrica es:  **Medida cuantitativa del grado en el que un sistema, componente o proceso posee un atributo determinado**

Las Métricas orientadas al tamaño: **Ninguna de las anteriores**

Como se pueden modificar un elemento de configuracion de una línea base establecida: **Mediante un proceso formal**

El objetivo de una línea base es la de: **Controlar cambios sin impedir aquellos que están debidamente justificados**

Las actividades de la gestión de configuracion son: **Identificación, control de cambios y versiones, auditoria de la configuracion, generación de informes de estado**

Si digo “Combina procedimientos y herramientas para gestionar las versiones de el proceso del software” Estoy hablando de: **Control de versiones**

Seleccione la respuesta mas apropiada: **Los workitems se diferencian por tipo, y cada tipo de proyecto tiene sus workitems**

La razón por la que medimos y posteriormente utilizamos métricas son: **Caracterizar, Evaluar, Predecir y Mejorar**

Las diferencias entre medición, medida, métrica e indicador son: **Medicion es acto de determinar una metrica, la cual mediante la aplicación de algún calculo se transformara en una medida que proporciona una indicación cuantitativa de algún atributo y conjunto de métrica y medidas nos da un indicador**

Seleccione la opción que mas se acerque a la definición de “Reunión de Planificación” (Planning Meeting): **Se realiza al inicion de cada Sprint donde se define que se va a construir y como, se define el incremento**

Marque cuales son Artefactos de Scrum: **Product Backlog y Strint Backlog**