Ingeniería en Software

**Trabajo Práctico N°7**

“Medición y Estimación de Costo”

Integrantes:

García, Gabriel

Sastre, Yamila

Turno Noche

**Trabajo Práctico Nro. 7**

**“Medición y Estimación de Costo”**

**Punto 1: Preguntas a responder**

**5.1 - Métricas para la calidad del software y su integración al proceso de ingeniería de software**

**Mediciones**

1. ¿Qué es una medición?

Es una medida cuantitativa que permite tener una visión profunda de la eficacia del proceso del Sw y de los proyectos utilizando un proceso como marco de trabajo. Se reúnen los datos básicos de calidad y productividad.

1. ¿Por qué son importantes?

Son importantes porque permiten valorar la calidad de los productos de ingeniería o de los sistemas que construimos y porque se pueden evaluar productos, procesos y proyectos expresando los resultados en números y colaborando con la mejora continua.

1. ¿Cuáles son los pasos?
   * + 1. Extraer las medidas y métricas del software que son apropiadas para la representación del Sw que está siendo considerado.
       2. Datos para extraer la formulación de métricas agregadas.
       3. Calcular las métricas.
       4. Analizar los resultados obtenidos en base a directrices preestablecidas y datos históricos.
       5. Interpretar el resultado del análisis para hacer las modificaciones en el análisis, diseño, codificación y prueba.
2. ¿Qué producto se obtiene?

Las métricas del Sw. serán calculadas sobre los datos agregados del análisis, de los modelos de diseño, del código fuente y de los casos de prueba.

1. ¿Cuáles son las cuatro razones para medir los procesos de software, los productos y los recursos?

Caracterizar:

* Comprender mejor los procesos, productos y recursos, y entornos
* Establecer líneas bases

Evaluar:

* Determinar el estado con respecto al diseño
* Valorar la consecución de los objeticos de calidad
* Evaluar el impacto de la tecnología
* Evaluar las mejoras del proceso en los productos

Predecir:

* Para planificar
* Establecer objetivos alcanzables para el coste, planificación y calidad
* Extrapolación de tendencias (proyecciones y estimaciones basadas en datos históricos)

Mejorar

* Recolección de información cuantitativa
* Problemas de raíz
* Ineficiencias
* Calidad del producto
* Rendimiento del proceso

1. ¿Cuál es el objetivo de todo proceso de medición?

El objetivo es poder contar con un mecanismo para evaluar productos, procesos y proyectos expresando los resultados en números y colaborando con la mejora continua.

**Factores de calidad**

1. Haga una lista de factores y métricas de calidad (McCall y Furps)

Factores:

1. Facilidad de auditoría
2. Exactitud
3. Estandarización de comunicaciones
4. Complejidad
5. Concisión
6. Consistencia
7. Tolerancia a errores
8. Eficiencia de ejecución
9. Generalidad
10. Independencia de hardware
11. Instrumentación
12. Modularidad
13. Operatividad
14. Seguridad
15. Autodocumentación
16. Simplicidad
17. Independencia del sistema
18. Trazabilidad
19. Facilidad de formación

Métricas de calidad:

1. Corrección
2. Fiabilidad
3. Eficiencia
4. Integridad
5. Mantenimiento
6. Flexibilidad
7. Capacidad de pruebas
8. Portabilidad
9. Reusabilidad
10. Interoperabilidad
11. Usabilidad

**Medidas, métricas e indicadores**

1. ¿Cuál es la definición de: medida, indicador y métricas?

Medida: Indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de algunos atributos de un proceso o producto. La medición es el acto de determinar una medida.

Indicador: Es una métrica o una combinación de métrica que proporciona una visión profunda del proceso proyecto o producto de Sw.

Métricas: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado

1. ¿Qué permiten los indicadores?

* Evaluar el estado del proyecto en curso
* Realizar un seguimiento de los riesgos potenciales
* Detectar las áreas de problemas antes que se conviertan en “criticas”
* Ajustar el flujo y las tareas de trabajo
* Evaluar la habilidad del equipo de proyecto en controlar la calidad de los productos de trabajo de la Ingeniería de Sw.

1. ¿Cómo se puede medir la efectividad de un proceso de software?

Se mide indirectamente. Se extrae un juego de métricas según los resultados que provienen del proceso. Entre los resultados se incluyen medidas de errores detectados e informado a los usuarios finales, producto de trabajos entregados, esfuerzo humano y tiempo consumido, ajuste con la planificación y otras medidas.

**Establecimiento de un programa de métricas de software**

1. ¿Qué tendría en cuenta al instituir un programa de métricas de proceso?

Al instituir un programa de métricas de procesos hay que tener en cuenta:

* Usar el sentido común y la sensibilidad organizativa para interpretar datos de las métricas
* Retroalimentar regularmente stakeholders que hayan trabajado en la recopilación de medidas y métricas
* No usar métricas para evaluar a particulares
* Trabajar con profesionales y equipos para establecer objetivos claros y métricas que se vayan a utilizar para alcanzarlos
* No utilizar métricas que amenacen a particulares o equipos
* Los datos de las métricas son un indicador de mejoras de proceso

**5.2- Mediciones del software**

**Métricas técnicas**

1. ¿Cuáles son las actividades de un proceso de medición?

Las actividades son:

• *Formulación*: Obtener medidas y métricas del sw apropiadas

• *Colección*: Acumulación de datos necesarios para obtener las métricas formuladas

• *Análisis*: Calcular métricas mediante el uso de herramientas matemáticas

• *Interpretación*: Evaluación de los resultados de las métricas para conseguir una visión interna de la calidad

• *Realimentación*: Recomendación de la interpretación de métrica

1. ¿Cuáles son los principios que se pueden asociar con las métricas técnicas?

* Los objetivos de la medición deberían establecerse antes de empezar la recogida de datos.
* Todas las técnicas sobre métricas deberían definirse sin ambigüedades.
* Las métricas deberían obtenerse basándose en una teoría válida para el dominio de aplicación.
* Hay que hacer las métricas a medida para acomodar mejor los productos y procesos específicos.
* Siempre que sea posible, la recogida de datos y el análisis debe automatizarse.

1. ¿Cómo podemos valorar la calidad una métrica de Sw?

* La métrica obtenida y las medidas que conducen a ello deberían ser:
* Simples y fácil de calcular
  + Facilidad para obtener una métricas
  + Cálculo con poco esfuerzo o tiempo
* Empírica e intuitivamente persuasivas
* Consistentes y objetivas
  + Resultados no ambiguos
* Consistentes en el empleo de unidades y tamaños
* Independientes del lenguaje de programación
* Un mecanismo eficaz para la realimentación de calidad
  + Conducir a un producto final de mayor calidad

**Métricas manuales y automatizadas**

1. ¿A qué nivel se realizan métodos manuales de estimación de Sw?

Métodos manuales:

* Estimaciones manuales a nivel del proyecto
* Estimaciones manuales a nivel de fase
* Estimaciones manuales a nivel de actividad

1. ¿A qué nivel se realizan métodos automatizado de estimación de Sw?

* Estimaciones automatizadas a nivel del proyecto
* Estimaciones automatizadas a nivel de fase
* Estimaciones automatizadas a nivel de actividad

**Métricas orientadas al tamaño y a la función**

1. ¿Cuál es la diferencia entre medidas directas e indirectas?

* ***Medidas directas***
* Entre las medidas directas del proceso de la ingeniería del software se incluyen el coste y el esfuerzo aplicados.
* Entre las medidas directas del producto se incluyen las líneas de código (LDC) producidas, velocidad de ejecución, tamaño de memoria, y los defectos informados durante un período de tiempo establecido.
* Entre las ***medidas indirectas*** se incluyen la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento y muchas otras capacidades.

1. ¿De dónde provienen las métricas orientadas al tamaño? Ejemplos

Las métricas del software orientadas al tamaño provienen de la normalización de las medidas de calidad y/o productividad considerando el «tamaño» del software que se haya producido.

1. ¿En qué consisten las métricas orientadas a la función?

Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la funcionalidad no se puede medir directamente, se debe derivar indirectamente mediante otras medidas directas.

1. ¿De dónde surgen los puntos función? ¿Y cómo se calculan los puntos de función?

Los puntos de función se derivan con una relación empírica según las medidas contables (directas) del dominio de información del software y las evaluaciones de la complejidad del software.

Los puntos de función se calculan completando la tabla. Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios de información se definen de la forma siguiente:

Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos.

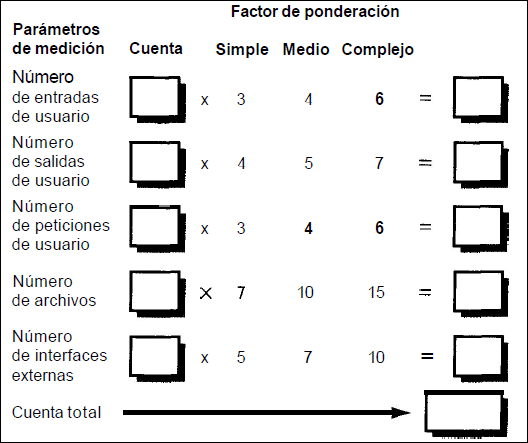
Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación.

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.

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 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 = cuenta-total x [0,65 + 0,01 x 6(Fi )] en donde cuenta-total es la suma de todas las entradas

PF obtenidas de la figura anterior.

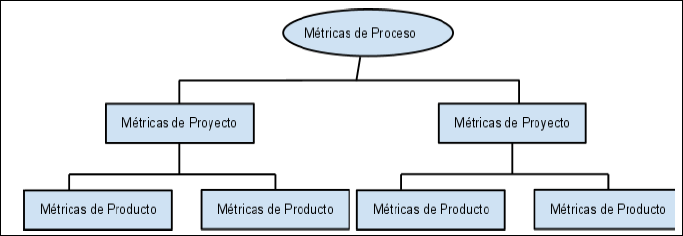
Fi (i = 1 a 14) son valores de ajuste de la complejidad según las respuestas a las siguientes preguntas:

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 ejecuta 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 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?
15. ¿Se puede usar siempre puntos de función?

La medida de punto de función se diseñó para aplicarse a sistemas de información de gestión. Para acomodar estas aplicaciones, se enfatizó la dimensión de datos (los valores de dominios de información tratados anteriormente) para la exclusión de dimensiones (control) funcionales y de comportamiento. Por esta razón, la medida del punto de función era inadecuada para muchos sistemas de ingeniería y sistemas empotrados (que enfatizan función y control). Para remediar esta situación se ha propuesto un número de extensiones a la métrica del punto de función básica.

**Métricas de producto, proceso y proyecto**

1. ¿Cuál es la relación entre las métricas de proceso, proyecto y producto? (ver gráfica)



La relación entre las métricas de proyecto, proceso y producto se muestra en la figura. Como se puede observar, el proceso SW constituye la base a partir de la cual se realiza el trabajo de una organización. Dichos procesos se aplican en la práctica en forma de proyectos. Como resultado de la ejecución de proyectos concretos se utilizan recursos y se obtienen productos. Por lo tanto, para establecer un marco de medición dentro de una organización, es necesario definir, recoger y analizar métricas sobre el proceso, el proyecto y recursos asociados así como el producto SW.

1. ¿Qué implica las mediciones del proceso y de un proyecto?

Las mediciones del proceso implican las mediciones de las actividades relacionadas con el SW, siendo algunos de sus atributos típicos el esfuerzo, el coste y los defectos encontrados.

Cuando se mide el proyecto, el objetivo fundamental que se pretende es el de reducir el coste total y el tiempo de desarrollo del mismo.

1. ¿Para qué le sirven los indicadores a un líder de proyectos?

● Evaluar el estado del proyecto en curso

● Realizar un seguimiento de los riesgos potenciales

● Detectar las áreas de problemas antes que se conviertan en “criticas”

● Ajustar el flujo y las tareas de trabajo

● Evaluar la habilidad del equipo de proyecto en controlar la calidad de los productos de trabajo de la Ing. SW

1. ¿Cuáles son las principales métricas a medir según Putnam y Myers y qué representan? (five core metrics)

Las principales métricas a medir son:

* + - 1. Cantidad de Funcionalidad, obtenida a través de las métricas de tamaño (LOC, Puntos de Función, etc.)
      2. Productividad, relación entre funcionalidad producida en el tiempo y el esfuerzo dedicado.
      3. Tiempo / Calendario, duración del proyecto
      4. Esfuerzo. Cantidad de trabajo en Personas/Mes.
      5. Fiabilidad. Expresada en ratio de defectos.

1. ¿En qué se centra la medición del producto? ¿Cuáles son sus salidas?

La medición del producto está centrada en evaluar la calidad de los entregables. Los productos del Sw son las salidas del proceso de producción del Sw, que incluyen, todos los artefactos entregados o documentos que son productos durante el ciclo de vida del Sw.

**Métricas de calidad**

1. Arme un cuadro con las principales medidas de calidad (corrección, facilidad de mantenimiento, integridad y facilidad de uso) y sus medidas.

|  |  |
| --- | --- |
| **Medidas de calidad** | **Medidas** |
| Corrección | Es el grado en el que el software lleva a cabo su función requerida. La medida más común de corrección es defectos por KLDC, en donde un defecto se define como una falta verificada de conformidad con los requisitos. |
| Facilidad de mantenimiento | La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba. |
| Integridad | Mide la capacidad de un sistema para resistir ataques. Se definen dos atributos: amenaza y seguridad.  integridad = C [( 1 - amenaza) x (1 - seguridad)] |
| Facilidad de uso | Grado en que el software es fácil de usar.  Subatributos: facilidad de comprensión,  facilidad de aprendizaje y operatividad. |

**Integración de métricas y líneas base**

1. Explique cómo aplicar las líneas base para poder integrar las métricas

La línea base sirve como base para la estimación desde donde se pueden evaluar mejoras.

**Aplicación de las métricas en Administración de Proyectos**

1. ¿Cuáles son los métodos prácticos genéricos para un líder de proyecto?

Los métodos prácticos genéricos para gerentes de proyectos de software, entre clientes de SPR, asemejan los siguientes:

* Un gerente de proyecto por cada ocho miembros del personal técnico
* Un gerente de proyecto de tiempo completo, por cada 1500 puntos función.
* Un gerente de proyecto por aproximadamente cada 150000 instrucciones de código fuente
* La administración de un proyecto comienza antes de los requisitos y se extiende hasta después de terminar el proyecto
* Trabajo de administración del proyecto = 35% de tiempo de administración entregable
* Trabajo del personal = 30% de tiempo de administración entregable
* Reuniones con otros gerentes o clientes = 22% de tiempo de administración entregable
* Trabajo departamental = 8% de tiempo de administración entregable
* Trabajo misceláneo: 5% de tiempo de administración entregable

1. ¿Cuáles son las tres ventajas de adquirir y usar herramientas de estimación de costos?

Las tres ventajas de adquirir y usar herramientas de estimación de costos son:

* + - 1. La curva de aprendizaje para manejar las herramientas de manera efectiva no es trivial
      2. Las herramientas necesitan calibrarse para ajustarlas a condiciones locales
      3. Algunas de las mejores herramientas de estimación de costos son bastantes costosas

**Métricas del modelo de análisis, diseño, desarrollo, pruebas y mantenimiento. Métricas orientadas a objetos**

1. ¿Cómo funciona la métrica Bang?

La métrica Bang puede emplearse para desarrollar una indicación del tamaño del software a implementar como consecuencia del modelo de análisis. Es “una indicación, independiente de la implementación, del tamaño del sistema”. Para calcular la métrica Bang, el desarrollador de software debe evaluar primero un conjunto de primitivas. Éstas se determinan evaluando el modelo de análisis y desarrollando cuentas para los siguientes elementos:

- Primitivas funcionales: Transformaciones que aparecen en el nivel inferior de un diagrama de flujo de datos.

- Elementos de datos: Los atributos de un objeto de datos, los elementos de datos no compuestos y aparecen en el diccionario de datos.

- Objetos: Objetos de datos.

- Relaciones: Las conexiones entre objetos de datos.

- Transiciones: El número de transiciones de estado en el diagrama de transición de estado.

Además de las seis primitivas nombradas arriba, se determinan medidas adicionales para:

- Primitivas modificadas de función manual: Funciones que caen fuera del límite del sistema y que deben modificarse para acomodarse al nuevo sistema.

- Elementos de datos de entrada: Aquellos elementos de datos que se introducen en el sistema.

- Elementos de datos de salid: Aquellos elementos de datos que se sacan en el sistema.

- Elementos de datos retenidos: Aquellos elementos de datos que son retenidos (almacenados) por el sistema.

- Muestras de datos: Las muestras de datos que existen en el límite de la i-ésima primitiva funcional.

- Conexiones de relación: Las relaciones que conectan el i-ésimo objeto en el modelo de datos con otros objetos.

1. ¿Qué métrica aplicarías en la calidad de las especificaciones?

* Especificidad (ausencia de ambigüedad)
* Compleción
* Corrección
* Comprensión
* Capacidad de verificación
* Consistencia interna y externa
* Capacidad de logro
* Concisión
* Trazabilidad
* Capacidad de modificación
* Exactitud
* Capacidad de reutilización

Además, las especificaciones de alta calidad deben estar almacenadas electrónicamente, ser ejecutables o, al menos, interpretables, anotadas por importancia y estabilidad relativas, con su versión correspondiente, organizadas, con referencias cruzadas y especificadas al nivel correcto de detalle.

1. ¿Cuáles son las medidas de complejidad del diseño? ¿Qué miden?
2. ¿Cuáles son las métricas de diseño a nivel de componentes? ¿En qué consisten?
3. ¿Qué se mediría en una IGU?
4. ¿Que sugiere Halstead que se mida?
5. ¿Cómo aplicarías Bang a los casos de prueba?
6. ¿Qué es el IMS?
7. ¿Qué sugieren medir en OO Lorenz y Kidd?

**Factores de ajuste**

1. ¿Qué factores influyen en la productividad del desarrollo?
2. ¿Cuáles son artefactos reutilizables?
3. ¿Cuáles son los factores principales de ajuste en el mantenimiento?

**5.3- Introducción a la estimación de costos de software**

**Funcionamiento de las herramientas de estimación de costos de software**

1. ¿Cómo funcionan las herramientas de estimación de costos de sw?

Las herramientas de estimación de costos de software funcionan tomando como base:

* + - 1. Contienen bases de conocimiento de cientos o miles de proyectos de software
      2. Realizan predicciones de tamaño, que las herramientas genéricas son incapaces de realizar
      3. Ajustan automáticamente estimaciones basadas en herramientas, lenguajes y procesos
      4. Predicen calidad y confiabilidad
      5. Predicen costos de mantenimiento y soporte antes de que estos ocurran en realidad

Lo que se realiza primordialmente es determinar el tamaño del proyecto basándose en las bases de conocimiento (principalmente en aplicaciones comerciales) o en entradas que se le dan a la herramienta por parte del gerente del proyecto o encargado de realizar estos cálculos. Luego se ingresan otros atributos que pueden afectar a estas estimaciones.

1. ¿Cuáles son los principios básicos de las herramientas comerciales de estimaciones de costos de sw? (fórmula)

Las herramientas comerciales se basan en enormes bases de conocimientos de proyectos anteriores, por lo que permiten heredar los atributos que afectan a las estimaciones de proyectos anteriores u otros proyectos similares. La fórmula siguiente ilustra los principios básicos de las herramientas comerciales modernas de estimación de costos de software:

Tamaño del proyecto X atributos del proyecto = Tiempo límite, esfuerzo, costos y entregables (estimados)

1. ¿Cuáles son los atributos que pueden afectar el resultado de una estimación?

Luego de determinado el tamaño se ingresan otros atributos que pueden afectar los resultados de la estimación, estos suelen ser:

* + - 1. Velocidad a que pueden cambiar los requisitos del proyecto
      2. Experiencia del equipo de desarrollo con este tipo de proyecto
      3. Proceso o métodos empleados para desarrollar el proyecto
      4. Actividades específicas a realizarse durante el desarrollo
      5. Número de incrementos, iteraciones o “carreras” que se utilizarán
      6. El o los lenguajes de programación a ser manejados
      7. Presencia o ausencia de artefactos reutilizables
      8. Suites de herramientas de desarrollo que se usarán para desarrollar el proyecto
      9. Entorno o ergonomía del espacio de trabajo en la oficina
      10. Separación geográfica del equipo en múltiples lugares
      11. Presión de los tiempos límites ejercida en el equipo por clientes o ejecutivos
      12. Obligaciones contractuales en términos de costos, fechas, defectos o características

1. ¿Cuáles son los tipos clave de atributos heredados que están asociados a la estimación? (Gráfica)
2. Nombre los pasos para las estimaciones manuales

**Omisiones accidentales**

1. ¿Qué actividades se incluye al estimar?

Las actividades que se incluyen al estimar son:

* Las estimaciones de costos precisas deben iniciar con predicciones de calidad, pues los costos de eliminación de defectos suelen ser más elevados que cualquier otro.
* En segundo lugar, como elementos de costo importantes, se encuentran gastos y esfuerzos dedicados a la producción de documentos en papel: planes, especificaciones, manuales de usuario, etc.
* Muchos proyectos grandes sitúan costos y fechas límite en el manejo de “requisitos progresivos” o nuevas características agregadas al proyecto, tras la fase de requisitos. Todos los desarrollos de software crecerán por requisitos progresivos y, por tanto, este factor debe ser parte integral de las estimaciones de todos los proyectos importantes.
* Una omisión frecuente en las estimaciones de costos es la exclusión accidental de viáticos (líneas aéreas, hoteles, etc.), para reuniones entre equipos de desarrollo en diferentes ciudades y países. Los viáticos pueden exceder el costo de la redacción del código de forma significativa y el tema no puede omitirse de manera accidental.
* Muchas estimaciones de costos de software, cubren sólo las actividades centrales del desarrollo de software y omiten temas tales como administración y soporte a proyectos (secretarias, administración, etc.). Estas actividades auxiliares son parte del proyecto y pueden ascender, en algunos casos, a 20% de los costos totales.
* Es muy común omitir las contribuciones de especialistas si sus habilidades sólo se necesitan sólo en una etapa del ciclo de desarrollo del software. Algunos grupos de especialistas que tienden a ser omitidos accidentalmente de las estimaciones de los costos de software, integrarían las áreas de control de calidad, redacción técnica, puntos de función, administración de bases de datos, optimización de rendimiento, redes y administración de sistemas. Las aportaciones combinadas de estos y otros especialistas pueden totalizar más de 20% del costo total de para el desarrollo de software y no deben omitirse de forma accidental.
* La omisión más común de las estimaciones internas de los costos de software para sistemas de información, son costos en que incurren los usuarios durante la definición de requisitos, creación de requisitos, revisiones de estado, revisiones de fases, documentación, inspecciones, pruebas de aceptación y otras actividades en que los desarrolladores tienen un rol clave. Como los representantes de usuarios no suelen considerarse parte del equipo del proyecto, sus contribuciones al proyecto rara vez se incluyen en las estimaciones del costo del software y estudios de medición. El esfuerzo real que aportan los usuarios a proyectos de desarrollo de software importantes puede acercarse al 20% del trabajo total en algunos casos.
* En el caso de muchos proyectos, el mantenimiento después de la entrega cuesta más que el desarrollo de la aplicación misma. No es recomendable detener la estimación en el punto de entrega del software, sin incluir cuando menos 5 años de estimaciones de mantenimiento y mejoras.

**Estimación de costos**

1. ¿Qué tipo de actividad es la estimación de costos? ¿De dónde derivan?

La estimación de costos de software no es una actividad “autónoma”. Las estimaciones se derivan en gran medida de requisitos del proyecto y otros atributos asociados con el proyecto. Una estimación de costos es precursora de presupuestos departamentales y sirve también como documento base para comparar costos acumulados contra proyectados.

1. ¿Qué estimaciones se realizan durante el proceso de desarrollo?

Para cualquier proyecto más grande que uno trivial, se elaborarán múltiples estimaciones de costos durante el proceso de desarrollo, incluyendo, pero limitado, a las siguientes:

* Una estimación aproximada de requisitos del proyecto.
* Una estimación formal inicial derivada de requisitos de proyecto.
* Una o más estimaciones a la mitad del ciclo de vida, reflejando cambios en los requisitos.
* Una acumulación final de costos, empleando datos históricos del proyecto.

1. ¿Cuáles son los consejos para realizar estimaciones?

Sea preciso.

* Sea conservador.
* Base la estimación en datos históricos sólidos.
* Procure calidad, pues las características del software afectarán fechas límite y costos.
* Procure no incluir documentos en papel, ya que pueden valer más que el código fuente.
* Incluya administración de proyectos.
* Incluya los efectos de requisitos progresivos.
* No exagere el efecto de las herramientas, lenguajes y métodos.
* Lleve las fases anteriores a estimaciones de costos, a nivel de actividades.
* Esté preparado para defender las suposiciones de su estimación.

**Goal Question Metrics**

1. ¿Cuál es el objetivo de GQM?

El objetivo de estos marcos de trabajo es proporcionar las referencias necesarias para poder llevar a cabo el proceso de medición de una forma efectiva y sistemática, en base a una serie de objetivos.

1. ¿Cuáles son sus fases?
2. ¿Qué se planifica en el proceso de GQM?

**Seis formas de estimar costos de software**

1. Arme un cuadro sinóptico explicando las seis formas de estimar costos (cap. 3)

**5.4- Estimaciones de costos de software**

**Herramientas de medición de la calidad de un producto**

1. ¿Cuáles son las principales tareas que debe realizar una herramienta de métricas?

* Adquisición de datos: conjunto de métodos y técnicas que permiten la obtención de datos necesarios para realizar la medición. Esta tarea presenta las siguientes posibilidades:
  + Manual
  + Semiautomática
  + Automática
  + Programable
* Análisis de las mediciones: incluye la habilidad para almacenar, recuperar, manipular y llevar a cabo el análisis de los datos. Esta tarea supone realizar las siguientes actividades:
  + Almacenamiento de los datos de la medición
  + Recuperación de los datos de la medición
  + Análisis Aritmético de resultados
  + Análisis estadístico de resultados
* Presentación de los datos: formatos que facilita la herramienta de medición para generar la documentación obtenida. Destacan las siguientes posibilidades de representación:
  + Tablas
  + Gráficos
  + Exportación de archivos a otras aplicaciones

1. ¿Cómo se clasifican las herramientas?

* Herramientas universales de métricas: aquellas que están diseñadas para soportar el trabajo con métricas y enfocadas a su obtención, análisis y presentación. Además estas herramientas deben proveer al usuario de:
  + Una interfaz de usuario flexible que permita realizar una obtención de datos exacta y efectiva.
  + Una amplia variedad de algoritmos o estándares para el análisis de un gran conjunto de métricas
  + La capacidad de generar un conjunto de informes flexibles y orientado a la representación gráfica que además permita personalizar su aspecto en función de los requisitos del usuario.
* Herramientas especializadas de métricas: aquellas que están diseñadas para soportar:
  + Otro tipo de actividad o funciones aparte de la medición, pero que aportan ciertas métricas especificas acerca del ciclo de vida del software, por ejemplo las herramientas de gestión del proyecto.
  + Una métrica especifica o conjunto concreto de métricas, por ejemplo las herramientas de medición de la complejidad del código

1. ¿Cuáles son los tipos de herramientas especializadas?

* Herramientas de estimación software: aquellas herramientas que permiten estimar esfuerzo y duración del proyecto en función de parámetros de tamaño y del entorno de desarrollo software
* Herramientas de gestión de proyecto: aquellas herramientas que soportan la planificación, seguimiento y gestión de un proyecto basándose en los recursos (tiempo, personal, etc.)
* Herramientas de contabilidad temporal: aquellas herramientas que aportan información temporal sobre la ejecución carga del código
* Herramientas de análisis de código fuente: aquellas herramientas que recorren el código fuente escrito en un determinado lenguaje de programación y cuentan líneas de código, puntos función, etc. y calculan ciertas métricas de complejidad.
* Herramientas de seguimiento de cambios y defectos: las herramientas de seguimientos de cambios permiten almacenar y analizar los cambios realizados en código como resultado de la corrección de defectos o actividades de mejora. Las herramientas de seguimiento de defectos permiten detectar los defectos que presenta el código aportando información sobre la severidad del defecto, el tipo de defecto o la localización dentro del código.

**Punto 2: Cuadro comparativo de estimaciones costos de software**

Arme un cuadro para presentar las estimaciones de costo.

* Presentar su definición, aplicación, comparar, encontrar diferencias, quienes las realizan, herramientas, tipos (de requisitos, prototipos, pruebas, etc.), impacto, mediciones, cambios, cosas comunes, etc. entre los siguientes tipos de estimaciones:

1. **Estimación de requisitos.**
2. **Estimación de prototipos.**
3. **Estimación de especificaciones y diseño de software.**
4. **Estimaciones de inspecciones de diseño.**
5. **Estimaciones de codificación.**
6. **Estimación del control de configuración de software.**
7. **Estimación de pruebas de software.**
8. **Estimación de documentación del usuario y del proyecto.**
9. **Estimación de la administración de proyectos.**
10. **Estimación de costos de mantenimiento y mejoras.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Definición** | **Aplicación** | **Comparar** |
| **Estimación de requisitos** | Los requisitos de software son el punto de partida de todo proyecto nuevo y también un coadyuvante clave en proyectos de mejoras. Son también muy ambiguos, a menudo repletos de conjeturas erradas y errores severos, en general resultan difíciles de esclarecer en forma concisa y comprensible. | Captura de requisitos y análisis. |  |
| **Estimación de prototipos** | Costo asociado a la creación de una réplica inicial y parcial de una aplicación, desarrollada para probar principios de diseño y características operacionales. | Captura de requisitos | Desechables, Cuadros de tiempo, evolutivos. |
| **Estimación de especificaciones y diseño de software** | Las especificaciones y diseño del software son una respuesta técnica a requisitos del usuario y sirven para describir en qué forma la aplicación de software que se construye, manejará los requisitos del usuario de manera automatizada. | Diseño |  |
| **Estimaciones de inspecciones de diseño** | Procedimiento formal en el cual se inspeccionan artefactos de Sw. | Diseño |  |
| **Estimaciones de codificación** | La programación o redacción de código representa un reto interesante a estimadores de software, debido al constante aumento de lenguajes de programación y del nivel de cada uno. | Etapa de desarrollo. |  |
| **Estimación del control de configuración de software** | Uno de los retos principales de la industria del software ha sido manejar el cambio de sus proyectos lo más eficientemente posible. También es importante estimar el trabajo de administración del cambio. | En todas las etapas. |  |
| **Estimación de pruebas de software** | La estimación del esfuerzo, fechas límites y costos de pruebas son temas complejos por las muchas formas de prueba en que podrían realizarse. Además, se complica también porque los defectos presentes al iniciar las pruebas pueden variar ampliamente. | Etapa de pruebas. |  |
| **Estimación de documentación del usuario y del proyecto** | Administración y producción de todo tipo de documentos de un proyecto. | Análisis |  |
| **Estimación de la administración de proyectos** | La administración de proyectos implica administrar personal, departamentos y proyectos con el fin de | En todas las etapas. |  |
| **Estimación de costos de mantenimiento y mejoras** | Estimar costo y presupuesto debido a mejoras realizadas en la estructura del código o a que se han añadido nuevas características que cumplen con nuevos requisitos de usuario. | Mejoras importantes, mejoras menores, mantenimiento, soporte al cliente, migración, conversión, soporte por garantía, reingeniería, optimización, instalación de nuevas versiones, etc. |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Quiénes las realizan** | **Herramientas** | **Tipos** |
| **Estimación de requisitos** | Clientes, personal de mercadotecnia, personal de ventas, personal de ingeniería, analistas de sistemas, programadores, personal de control de calidad y gerentes de proyectos de software. | Requisite de Rational, Bachman Analyst Workbench, Texas Instruments Information Engineering Facility. | De requisitos. |
| **Estimación de prototipos** | Desarrolladores | Materiales reutilizables, lenguaje de programación elegido, Casos de prueba. | De prototipos. |
| **Estimación de especificaciones y diseño de software** | Analistas de sistemas, personal de desarrollo y arquitectos. | Plantillas para métodos de diseño comunes. | De especificaciones y diseño. |
| **Estimaciones de inspecciones de diseño** | Creador, moderador y registrador. | Actividad manual de inspección página por página. SPQR/20 | De inspecciones de diseño. |
| **Estimaciones de codificación** | Personal de desarrollo. |  | De codificación. |
| **Estimación del control de configuración de software** | Especialistas en control de cambio, programadores, auditores, clientes. |  | De control de configuración. |
| **Estimación de pruebas de software** | Programadores, especialistas en pruebas, control de calidad, clientes. |  |  |
| **Estimación de documentación del usuario y del proyecto** | Redactor técnico, | SPQR/20, CHECKPOINT, COCOMO, COCOMO II, SLIM | De documentación del usuario y del proyecto. |
| **Estimación de la administración de proyectos** | Gerentes de proyectos. | Project, Timeline, Primavera, Artemis. | De administración de proyectos. |
| **Estimación de costos de mantenimiento y mejoras** | Especialistas en mantenimiento y mejoras. Programadores. | COCOMO II, SLIM, PRICE-S KnowledgePlan. | De mantenimiento y mejoras. |

|  |  |  |
| --- | --- | --- |
|  | **Impacto** | **Mediciones** |
| **Estimación de requisitos** |  | Puntos función.  Puntos de característica relacionada. |
| **Estimación de prototipos** | Puede llegar a convertirse en un producto final. | Puntos de función |
| **Estimación de especificaciones y diseño de software** |  |  |
| **Estimaciones de inspecciones de diseño** | Inspecciones formales para eliminar defectos y errores en el código y en el diseño. | Puntos de función |
| **Estimaciones de codificación** | Código reutilizable.  Experiencia del equipo.  Errores de código o defectos.  Incremento de requisitos. | Puntos función. |
| **Estimación del control de configuración de software** |  |  |
| **Estimación de documentación del usuario y del proyecto** |  |  |
| **Estimación de la administración de proyectos** |  | Estimación de fechas límites, de costos, de calidad, |
| **Estimación de costos de mantenimiento y mejoras** |  | Puntos función. |