

**3. ¿Qué relación guarda un modelo de proceso con la agenda de un proyecto y con la estimación de costes? Pon un ejemplo que justifique la respuesta. (2,5p)**

El modelo de proceso condiciona qué actividades y en qué orden se incluirán en la agenda del proyecto, necesaria para poder conocer el tiempo y el coste total de desarrollo, y que son responsabilidad del gestor de un proyecto.

El modelo de proceso facilita la subdivisión de actividades en otras más "pequeñas", ya que proporciona el conjunto de actividades que hay que realizar. Estas actividades pueden subdividirse y estructurarse convenientemente, por ejemplo, utilizando representaciones WBS que son muy útiles para estimar los costes, en donde, además, se favorece el uso de una estimación ascendente o descendente.

Las estadísticas de proyectos anteriores nos van a ayudar en las estimaciones, más aún cuando el proyecto se haya desarrollado con el mismo modelo de proceso. En el caso de UP incluso nos podría decir los % de las estimaciones que se usan en cada fase.

Si se sigue un modelo de proceso UP nos dará lugar a las 4 fases (inicio, elaboración, construcción y transición), así como a un conjunto de n iteraciones y a una serie de tareas de cualquier disciplina dentro de las iteraciones. Por otro lado, las relaciones de precedencia de dichas actividades se corresponden con la secuencia temporal de las actividades que se llevan a cabo en el modelo UP.

**2. ¿Qué unidades de medida se emplean para dar los datos de las estimaciones de costes? ¿Qué ventajas presentan estas unidades respecto a otras? ¿Qué unidades se emplean en los presupuestos? (2 p)**

Todas las técnicas de estimación estiman el esfuerzo necesario para completar un proyecto. El esfuerzo se mide en Personas/mes. Tiene la ventaja de permitirnos comparaciones entre proyectos de forma atemporal y entre distintos lugares (empresas/proyectos). Por ejemplo, si utilizásemos euros para guardar el esfuerzo cuando quisiéramos utilizar proyectos de años anteriores deberíamos tener en cuenta las subidas salariales (ipc).

En los presupuestos utilizamos unidades monetarias (precio venta) y de tiempo (fecha entrega).

**Explica la diferencia entre WBS, plan general y plan detallado de un proyecto. ¿Qué diferencia existe entre el plan general y detallado en los modelos UP y cascada? (2,5p)**

El WBS es un esquema de tareas estructuradas. El plan general fija las fases e iteraciones (siempre que el modelo tenga), así como sus objetivos. El plan detallado concreta las tareas, establece el orden y precedencias de las tareas y asigna duraciones y recursos a las tareas y establece las fechas.

La primera diferencia es que el modelo en cascada no tiene iteraciones. La segunda diferencia es que en el modelo en cascada el plan detallado se establece al principio del proyecto, mientras que en el modelo up se va detallando poco a poco. Otra diferencia son las fases, en cascada son análisis, diseño, implementación, integración y prueba, mientras que en el modelo UP son inicio, elaboración, construcción y transición.

**Diferencias entre un presupuesto y un documento de estimación de costes:**

El documento de estimación de costes es un documento interno de la empresa que indica el esfuerzo necesario para realizar un proyecto (personas/mes), el coste temporal y económico del proyecto.

El presupuesto es un documento que se va a entregar al cliente para informarle sobre los precios de venta, tiempos de entrega y funcionalidades de la aplicación.

La diferencia fundamental es que la estimación habla de costes, mientras que el presupuesto habla en términos de precios de venta y fechas de entrega. El documento de estimación de costes debe tener estimaciones mediante distintas técnicas (Pricing to Win, Parkinson...)

El presupuesto debe tener los datos de proyecto, empresa y cliente, una descripción del producto a desarrollar, requisitos mínimos de SW y HW que necesita el programa, disposiciones legales, plazos de pago...

**Visibilidad de un proyecto:**

Consiste en conocer exactamente qué es lo que se está haciendo en cada momento del desarrollo. La única forma de hacer que el proceso de desarrollo muestre el estado del producto es mediante la documentación generada en el proceso. La posibilidad de ver en qué estado se encuentra un producto software durante su desarrollo es de vital importancia ya que, gracias a esto, podremos hacer una planificación efectiva que nos permitirá monitorizar y controlar el desarrollo de este.

Ejemplos que dan visibilidad a un proyecto: Gantt, hacer selección de un plan adaptativo en vez de predictivo (hitos), informe de iteración (documentación), uso de diferentes niveles en el plan.

**Diferencia entre gestión de configuraciones y mantenimiento:**

La gestión de configuración es una actividad de autoprotección que se lleva a cabo a lo largo de todo el ciclo de vida del software. El mantenimiento es un subconjunto de la gestión de configuraciones que tiene lugar desde el momento que el software se pone en producción.

El modelo de proceso UP sólo abarca el desarrollo del software hasta su puesta en producción, no modela el mantenimiento.

**Explica que es un presupuesto y explica cuáles son sus 3 objetivos:**

Es un documento contractual entre el cliente y proveedor que sirve para establecer los productos y o servicios, la fecha de entrega del servicio, así como el precio de los mismos. Los objetivos del presupuesto son: Vender, fortalecer las relaciones entre cliente y proveedor y crear oportunidad de negocio. Para vender haríamos un presupuesto bien diseñado, con una presentación clara, información detallada y unos precios ajustados al mercado. Para fortalecer las relaciones haremos que los productos y o servicios estén bien definidos con el fin de evitar discrepancias futuras. Y para crear oportunidades de negocio ofreceremos servicios adicionales tales como hardware, migración datos, mantenimiento, formación.

**¿Qué diferencia existe en la pirámide de Maslow entre las necesidades de autorealización, estima y necesidad social? (0,5 p)**

La necesidad social hace referencia a la necesidad del individuo de sentirse reconocido como miembro de un grupo. La necesidad de estima es la necesidad del individuo de que se reconozca su trabajo y sus logros. La autorealización es la necesidad psicológica más elevada del ser humano, se halla en la cima de la pirámide, y viene marcada por el carácter vocacional y el desarrollo personal.

**Explica la vinculación de la estimación realizada con Puntos Objeto con Pricing to Win. (0,5p)**

La técnica de Puntos Objeto estima el coste del proyecto, mientras que Pricing to Win estima el precio de venta. Al comparar ambos resultados nos dará el número de licencias de software que tenemos que vender para cubrir los costes. Esto implicará hacer un estudio de mercado para ver las ventas esperadas y el tiempo necesario para recuperar la inversión.

**¿Qué dos formas hemos empleado en prácticas para realizar el seguimiento de un proyecto? ¿Qué ventajas presenta cada una de ellas? (0,75p)**

Diagrama Gantt de Seguimiento y Modelo Eva. El Gantt de seguimiento permite realizar el seguimiento de una manera visual y rápida. El modelo Eva nos permite hacer un seguimiento objetivo y cuantitativo (con datos numéricos).

**Tema 2:**

**Estimación:** Consiste en predecir los recursos (monetarios, temporales, humanos, materiales, ...) necesarios para llevar a cabo el proceso de desarrollo del software.

**Factores del coste (5) (+soluciones (Siempre cae, veremos este año))**

- **Oportunidad de mercado:** Calidad/precio, diferenciar mi producto de la competencia.
- **Incertidumbre en la estimación de costes:** Guardar datos de anteriores proyectos para poder estudiar errores cometidos para así corregirlos.
- **Términos contractuales:** Venta de licencias = minimización de costes, manteniendo nosotros la propiedad del software y sus derechos comerciales.
- **Volatilidad de los requerimientos:** Mantener mejor comunicación con el cliente a lo largo del proyecto.
- **Salud financiera:** Cobrar un anticipo en la firma del contrato.

**Productividad:** La productividad de un programador es la velocidad a la que los ingenieros implicados en el desarrollo del software producen dicho software y su documentación.

Podemos medir la productividad en relación con el tamaño (líneas de código) o en relación con la funcionalidad (puntos de función, puntos objeto).

#### Medir en relación con el tamaño:

- Cuanto **mayor sea la expresividad** del lenguaje, **más baja será su productividad** aparente. (asm vs C++)
- Cuantas **más líneas de código** emplee el programador, **mayor será su productividad**.

Por lo tanto, podremos decir que comparar la productividad utilizando lenguajes diferentes puede llevar a conclusiones erróneas respecto a la productividad de los programadores.

#### Medir en relación con la funcionalidad:

- **Puntos de función:** La técnica de estimación de puntos de función se basa en la contabilización de unos contadores (características del programa): Entradas y salidas externas, interacciones de usuario, interfaces externas y ficheros usados por el sistema.

Se asocia un peso con cada uno de estos contadores y los puntos de función se calculan multiplicando cada factor por su peso y sumando todos ellos.

- **Ventajas frente a líneas de código:**
  - Independientes del lenguaje de programación.
  - Se pueden calcular a partir de la especificación.
  - Usa información del dominio del problema.
  - Resulta más fácil a la hora de reusar componentes.
  - Se encamina a aproximaciones orientadas a objetos.

**Uso:** Los Puntos de función pueden usarse para estimar el número de líneas de código\* para un lenguaje dado ( $LOC = AVC * \text{número de puntos de función}$ ). AVC es un factor dependiente del lenguaje.

**Problema:** Un problema que nos encontramos es que los puntos de función son muy subjetivos y son totalmente dependientes del estimador.

- **Puntos objeto:** Medida alternativa relacionada con la funcionalidad cuando se utilizan lenguajes 4GL o similares para el desarrollo. El número de puntos de objeto en un programa es una estimación ponderada de: Número de pantallas visualizadas por separado, número de informes que el sistema produce y módulos 3GL que deben desarrollarse para completar el código 4GL.
- **Ventajas frente a puntos de función:**
  - Son más fáciles de estimar a partir de una especificación que los puntos de función, ya que solo se consideran pantallas, informes y módulos 3GL.
  - Pueden estimarse en fases tempranas del desarrollo (en estas etapas resulta muy difícil estimar el LOC de un sistema)

Algunos factores que afectan a la productividad son: Experiencia en el dominio de la app, calidad del proceso, tamaño del proyecto, entorno de trabajo...