ESTIMACIÓN PARA PROYECTOS DE SOFTWARE

- Fallar en la planificación es uno de los errores más importantes que un proyecto puede tener...
- La planificación efectiva es necesaria para resolver problemas de manera temprana en el Proyecto a bajo costo, en lugar de manera tardía en el proyecto a alto costo.

¿Qué involucra la Planificación de proyectos de software?

Abarca cinco actividades:

1. Estimación,
2. Calendarización (fechas de entrega),
3. Análisis de riesgos,
4. Planificación de gestión de la calidad y
5. Planificación de gestión del cambio.

Antes de que el proyecto pueda comenzar, el equipo de software debe:

- ✓ Estimar el trabajo que se va a realizar,
- ✓ Los recursos que se requerirán y
- ✓ El tiempo que transcurrirá de principio a fin.

Una vez completadas dichas actividades, el equipo de software debe:

- Establecer un calendario del proyecto que defina las tareas e hitos de la ingeniería de software, que identifique quién es responsable de realizar cada tarea y especifique las dependencias entre tareas que puedan imponer una fuerte demora sobre el avance.

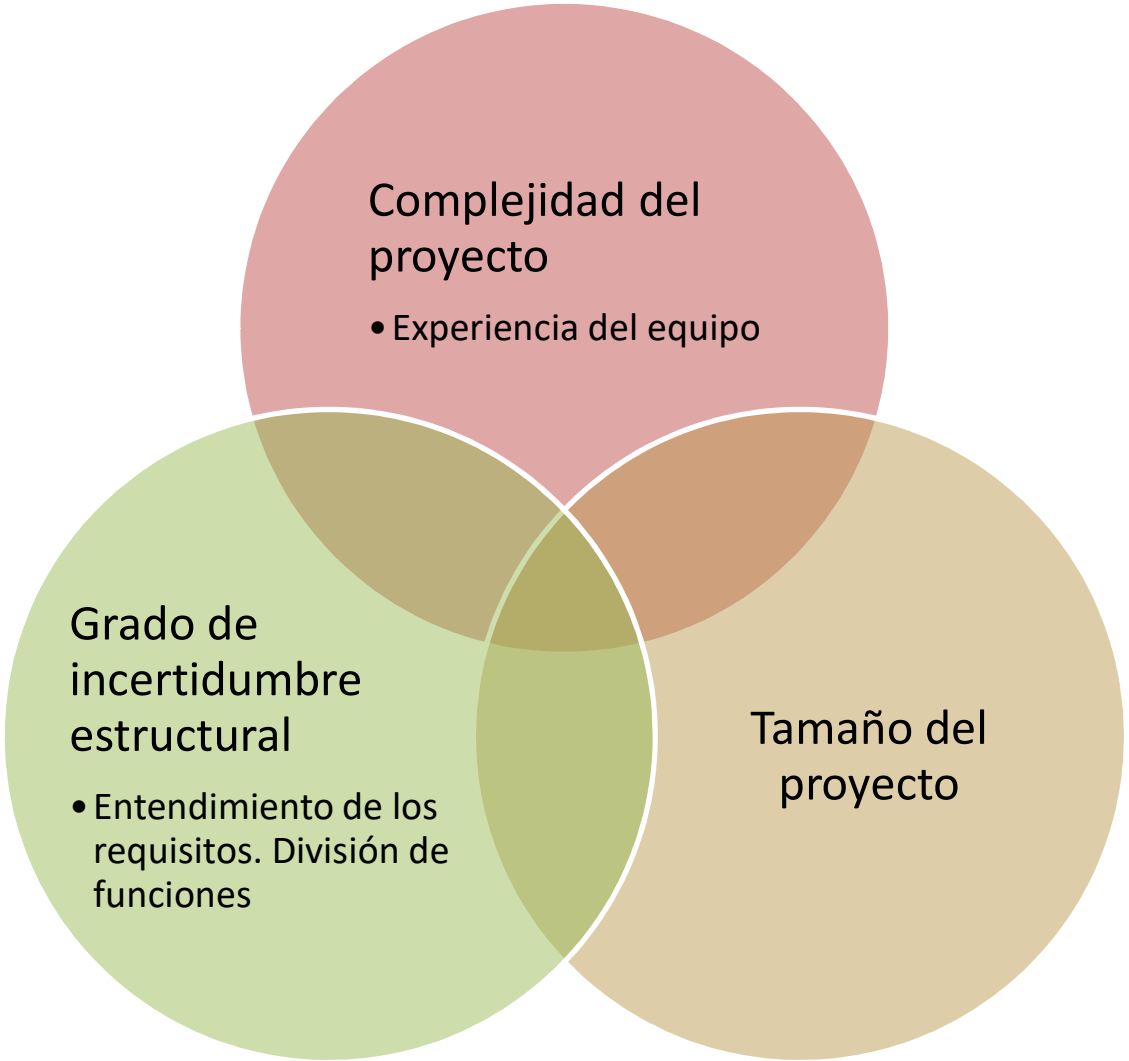
¿Qué es la estimación?

- Es el intento por determinar cuánto dinero, esfuerzo, recursos y tiempo tomará construir un sistema o producto específico basado en software.

¿Cuáles son los pasos generales?

1. Descripción del ámbito del problema. (Objetivo general del sistema).
2. Luego éste se descompone en un conjunto de problemas más pequeños (requerimientos) y
3. cada uno de éstos se estima, usando como guías datos históricos y experiencia (métricas).
4. La complejidad y el riesgo del problema se consideran antes de realizar una estimación final.

¿Qué afectan a la estimación?



El riesgo de estimación se mide por el grado de incertidumbre en las estimaciones cuantitativas establecidas para recursos, costo y calendario.

Si el ámbito del proyecto se describe de manera pobre o si los requisitos del proyecto están sujetos a cambios, la incertidumbre y el riesgo en la estimación se vuelven peligrosamente altos.

Sugerencia:

- ✓ Con el cliente se deben reconocer que la variabilidad en los requisitos del software significa inestabilidad en costo y tiempo de desarrollo.
- ✓ Tomar una visión iterativa del desarrollo. Es decir, actualizar la estimación (conforme se conoce más información) – No muy aconsejable - y revisarla cuando el cliente hace cambios a los requisitos.

¿En qué se basa el éxito de la estimación?

- ✓ El grado en el que se estimó adecuadamente el ***tamaño del producto*** que se va a construir,
- ✓ La habilidad para ***traducir*** la estimación de ***tamaño en esfuerzo humano, tiempo calendario y dinero.***
- ✓ El grado en el que el plan del proyecto refleja las ***habilidades del equipo de software*** y
- ✓ La ***estabilidad de los requisitos*** del producto y el entorno que soporta el esfuerzo de ingeniería de software.

¿Qué podemos utilizar para estimar proyectos de software?

Técnicas de descomposición

Modelos empíricos

TÉCNICAS DE DESCOMPOSICIÓN

1. **Dimensionar el software (Problema de dimensionamiento del software):**
    1. **Dimensionamiento de punto de función**
    2. **Dimensionamiento de componente estándar**: se estima el tamaño a través del número de ocurrencias de c/componente y luego usa datos de proyecto históricos para estimar el tamaño entregado por componente estándar (módulos, pantallas, reportes, etc.).
    3. **Dimensionamiento del cambio**: Este enfoque se usa cuando un proyecto abarca el uso de software existente que debe modificarse en alguna forma como parte de un proyecto. Se estima el número y tipo de las modificaciones que deben lograrse.

2. **Estimación basada en problema**: A partir del enunciado del alcance del software descomponer el enunciado en problemas que puedan estimarse cada uno de manera individual. Puede elegir como componente para dimensionamiento, clases u objetos, módulos, cambios o procesos empresariales afectados. Luego calcular para cada problema el "valor esperado" para el tamaño del software:

Tamaño esperado = ($T_{optimista}$ + 4 T_{medio} + $T_{pesimista}$)/ 6

3. **Estimación con casos de uso**: se utiliza si para el diseño y desarrollo utilizo UML y RUP. Se basa en el método de PF.

Utiliza actores y casos de uso relevados para calcular el esfuerzo que significará desarrollarlos.

A los casos de uso se les asigna una complejidad basada en transacciones o clases de análisis, mientras que a los actores se les asigna una complejidad basada en su tipo, es decir, si son interfaces con usuarios u otros sistemas.

También se utilizan factores de entorno y de complejidad técnica para ajustar el resultado.

El objetivo de la técnica de estimación con casos de uso

- Estimar las horas necesarias para ejecutar un conjunto de casos de uso. Es decir, necesitamos predecir cuánto tiempo llevará el desarrollo de software y cuántas personas se requieren para realizarlo. Para ello, es necesario cuantificar la complejidad del sistema y el tiempo necesario para producir una unidad de complejidad.

MODELOS DE ESTIMACIÓN EMPÍRICOS

- Un modelo de estimación refleja la población de proyectos de los cuales se derivó. Por tanto, el modelo es sensible al dominio.
- **Un modelo de estimación debe calibrarse para que refleje las condiciones locales**.
- El modelo debe probarse aplicando los datos recopilados de los proyectos completados, alimentando los datos en el modelo y luego comparando los resultados reales con los predichos. Si la concordancia es pobre, el modelo debe afinarse y volverse a probar antes de poder usarse.

Modelo COCOMO II

CO*nstructive* ***CO****st* ***MO****del*: modelo constructivo de costos http://softwarecost.org/tools/COCOMO

Es un modelo empírico que se obtuvo recopilando datos de varios proyectos grandes.

Estos datos fueron analizados para descubrir las fórmulas que mejor se ajustaban a las observaciones. Estas fórmulas vinculan el tamaño del sistema y del producto, factores del proyecto y del equipo con el esfuerzo necesario para desarrollar el sistema.

Algunos modelos de estimación orientados a PF

- **Modelo Albrecht y Gaffney:**
  - Esfuerzo = -91.4 + 0.355 * PF
- **Modelo Kemerer:**
  - *Esfuerzo*= -37 + 0.96 * PF
- **Pequeño modelo de regresión de proyecto:**
  - *Esfuerzo*= -12.88 + 0.405 PF

- Es claro que cada uno producirá un resultado diferente para los mismos valores de PF. Por tal motivo

¡Los modelos de estimación deben calibrarse para las necesidades locales!

Estimación para desarrollo ágil

- La estimación ágil es un trabajo en equipo
- Cada miembro del equipo aporta una perspectiva diferente sobre el producto y el trabajo necesario para entregar una historia de usuario.

Usa un enfoque de descomposición, con los siguientes pasos:

1. Cada historia de usuario (el equivalente de un minicaso de uso creado al comienzo de un proyecto por los usuarios finales u otros participantes) se considera por separado con propósitos de estimación.
2. La historia de usuario se descompone en el conjunto de tareas de ingeniería de software que será necesario desarrollar.
3. El esfuerzo requerido por cada tarea se estima por separado.
4. Las estimaciones para cada tarea se suman a fin de crear una estimación para la historia de usuario.
5. Las estimaciones de esfuerzo para todos los escenarios (historias de usuario) que se implementan para un incremento de software determinado se suman a fin de desarrollar la estimación del esfuerzo para el incremento.

¿En qué momento se realiza la estimación en Scrum?

Durante el **Sprint Planning**. Es la etapa de reunión inicial, donde todo el equipo se prepara para el Sprint que está por comenzar.

El equipo de desarrollo se sienta junto al Product Owner y al Srum Master, en la sesión de planificación. Se presentarán los PBI (Product Backlogs Items) y se eligen los elementos según la prioridad y asocia una estimación de tiempo y esfuerzo.

Matriz de Eisenhower

Urgente e Importante (Hacer) **Ejemplo:** Un error crítico en la aplicación que causa un fallo total y afecta a los usuarios. **Acción:** Resolver el error de inmediato, ya que el fallo es urgente y causa un impacto negativo en los usuarios.	**Importante pero no Urgente (Programar)** **Ejemplo:** Implementar una nueva funcionalidad en la aplicación que mejorará la experiencia del usuario, pero no es esencial para el lanzamiento actual. **Acción:** Programar la implementación para un futuro cercano, asegurando que se realice a tiempo para no afectar el lanzamiento.
Urgente pero no Importante (Delegar) **Ejemplo:** Una solicitud de cambio menor o una corrección de errores que, aunque urgentes, no son críticas para el éxito del producto. **Acción:** Delegar la tarea a un miembro del equipo o un tercero que pueda realizarla rápidamente.	**No Urgente y No Importante (Eliminar)** **Ejemplo:** Requisitos de software que son considerados triviales o que no aportan valor significativo al producto. **Acción:** Eliminar o posponer estos requerimientos hasta que sean necesarios o que se comprenda su valor real.

Algunos métodos de estimación ágil

Estimación en Scrum Delphi

- El grupo estima el esfuerzo anónimamente de un ítem del Backlog. Se repite hasta que las estimaciones del equipo estén cerca unas de otras, y se pueda llegar a un consenso para la estimación final.

Story Points

- Se da puntos a cada historia de usuario, o elemento del backlog, según su nivel de dificultad.

Planning Poker

- Similar a Delphi, solo que se utilizan cartas con números que suelen estar en la secuencia de Fibonacci: 0, 1, 2, 3, 5, 8, 13 y 21.

Técnica de las camisetas

- Si consideran que una actividad es compleja, la colocarán con el tamaño XL. En cambio, una actividad que consideren sencilla, puede colocarse en la categoría de XS.

La estimación de proyecto de software nunca puede ser una ciencia exacta, pero una combinación de buenos datos históricos y técnicas sistemáticas pueden mejorar la precisión de la estimación.