



Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería

Cátedra: **Análisis y Diseño de Sistemas**

Unidad 2: Análisis de Requerimientos

- Gestión de riesgo
 - El riesgo afecta a los futuros acontecimientos. El hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado. La pregunta es ¿podemos, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y mejor para el futuro?
 - El riesgo implica **cambio**, que puede venir dado por cambios de opinión, de acción, de lugar. El riesgo implica **elección** y la incertidumbre que entraña la elección. Por lo tanto, el riesgo, como la muerte, es una de las pocas cosas inevitables de la vida.

➡ Lo importante es minimizar los riesgos.

- **Definición**
- Es la probabilidad de ocurrencia de un evento no deseado y que tal ocurrencia signifique consecuencias negativas o positivas al proyecto. En esta definición se destacan tres elementos:
 - ✓ La existencia u ocurrencia de un evento
 - ✓ La probabilidad de ocurrencia de dicho evento
 - ✓ Un impacto, es decir, la generación de consecuencias

- Gestión de riesgo - Certidumbre vs. Incertidumbre
 - El **tener información** es fundamental en el proceso de administración de riesgos, ya que ayuda a obtener **certeza** respecto a los posibles hechos que pueden generar consecuencias negativas o positivas en los proyectos.
 - En el desarrollo de la vida de un proyecto se presenta habitualmente la necesidad de manejar situaciones en las que existen incertidumbre, las que dificultan la toma de decisiones.
 - Esta incertidumbre varía en un rango que va desde la **Certeza Total-Seguridad** a partir del conocimiento adquirido y de la posición de información completa y adecuada para la toma de decisiones; hasta la **Incertidumbre-Total**, en donde el conocimiento es reemplazado por la opinión, y no existe información de respaldo.
 - Generalmente en un proyecto se presentan situaciones intermedias, de **Incertidumbre Relativa** con información parcial y con conocimiento relativo respecto a las nuevas situaciones que se presentan.

- Gestión de riesgo - Los riesgos y el ciclo de vida
 - Durante el ciclo de vida de un proyecto se producen distintos eventos que pueden generar efectos negativos en él. Por tal razón, es conveniente que se tengan en cuenta cuáles son las acciones que deben tomarse para evitar o mitigar dichos efectos.
 - Para el análisis definimos cuatro fases del ciclo de vida:
 - ✓ **Concepto:** debemos identificar todos los posibles riesgos que pueden impactar en el proyecto
 - ✓ **Planeamiento:** determinar la prioridad, rango y estado de cada riesgo en función del tipo, probabilidad e impacto
 - ✓ **Implementación:** establecer la estrategia. Ésta puede abarcar desde una decisión de aceptar el riesgo, poner recursos para controlarlo o buscar la forma de eliminarlos
 - ✓ **Terminación:** construir una base de consulta permanente para su uso durante el proyecto y futuros proyectos
 - El manejo del riesgo surge entonces como la necesidad de manejar formal y sistemáticamente a través de la Identificación, Cuantificación y Control de los elementos o factores con potencial suficiente como para generar cambios no deseados, disminuyendo sus consecuencias negativas, así como maximizando los resultados de los eventos positivos.

- Gestión de riesgo - Estrategias de Riesgo Proactivas y Reactivas
 - En las **estrategias reactivas** nunca nos preocupamos de los problemas hasta que ocurran. Este método se llama “de bomberos”. Cuando falla, la gestión de crisis entra en acción y el proyecto se encuentra en peligro total. Es lo que hacemos.
 - En las **estrategias proactivas** se empieza mucho antes de que comiencen los trabajos técnicos. Se identifican los riesgos potenciales, se valoran su probabilidad y su impacto y se establece una prioridad según su importancia. Se establece un plan para controlar el riesgo. El objetivo es evitar el riesgo, pero si no se puede, entonces se debe desarrollar un plan de contingencia.
 - Es importante entender y hacer entender, incluso a los clientes, que la Administración de los riesgos es justamente un hecho proactivo, anticipatorio, que contribuye al éxito del proyecto en la medida en que sea encarado con el mismo rigor que cualquiera de los otros procesos del Gerenciamiento de proyectos.

- Gestión de riesgo - Riesgos de SW
 - El riesgo implica dos características:
 - ✓ **Incertidumbre:** El acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100% de probabilidad.
 - ✓ **Pérdida:** Si el riesgo se convierte en una realidad, ocurrirían consecuencias no deseadas o pérdidas.
 - Es importante cuantificar el nivel de incertidumbre y el grado de pérdida asociado a cada riesgo.

- Gestión de riesgo - Riesgos de SW (Cont.)
 - Para hacerlo se consideran diferentes categorías:
 1. **Los riesgos del proyecto** amenazan al plan del proyecto. Si los riesgos se hacen realidad es probable que la planificación temporal del proyecto se retrase y los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto en el proyecto.
 2. **Los riesgos técnicos** amenazan la calidad y la planificación temporal del SW que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible. Los riesgos técnicos identifican problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento.
 3. **Los riesgos del negocio** amenazan la viabilidad del SW para construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o producto. Pueden ser: construir un producto excelente que no quiere nadie en realidad (riesgo de mercado), construir un producto que no encaja en la estrategia comercial (riesgo estratégico), construir un producto que no se sabe vender, perder el apoyo de una gestión experta, debido a cambios de enfoques o cambios de personal (riesgo de personal) y perder presupuesto o personal asignado (riesgo de dirección).

- Gestión de riesgo - Identificación de Riesgo
 - Es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc.). Identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario.
 - Existen dos tipos de riesgos:
 - ✓ **Genéricos:** son una amenaza potencial para todos los proyectos
 - ✓ **Específicos del producto:** se pueden identificar los que tienen una clara visión de tecnología, el personal y el entorno específico del proyecto
 - Un método para identificar riesgos es crear una lista de comprobación de elementos de riesgo. Sirve para identificar riesgos y se enfoca en un subconjunto de riesgos conocidos y predecibles en las siguientes categorías:
 - ✓ Tamaño del producto
 - ✓ Impacto en el negocio
 - ✓ Características del cliente
 - ✓ Definición del proceso
 - ✓ Entorno de desarrollo
 - ✓ Tecnología a construir
 - ✓ Tamaño y experiencia de la plantilla

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos del Tamaño del producto
 - Riesgos genéricos asociados con el tamaño general del software.
 - Ejemplo: Tamaño estimado en LDC o PF, grado de seguridad en la estimación del tamaño, tamaño estimado en número de programas, archivos y transacciones, % de desviación del tamaño respecto de la medida de productos anteriores, tamaño de la base de datos creada o empleada por el producto, número de usuarios, número de cambios previstos a los requisitos del producto (antes y después de la entrega), cantidad de software reutilizado.
 - Se debe comparar con experiencias anteriores, si tenemos una gran desviación del porcentaje o las magnitudes son similares pero los resultados anteriores fueron poco satisfactorios el riesgo es grande.

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos del impacto del Negocio
 - Riesgos genéricos asociados con las limitaciones impuestas por la gestión o el mercado. Se debe comparar con experiencias anteriores, si tenemos una gran desviación del porcentaje o las magnitudes son similares pero los resultados anteriores fueron poco satisfactorios el riesgo es grande.
 - Por ejemplo: efecto de ese producto en los ingresos, es razonable la fecha límite de entrega, viabilidad del producto para los gestores expertos, número de clientes que usarán el producto y las consistencias de sus necesidades relativas al producto, número de otros productos con los que va a interaccionar, sofisticación del usuario final, cantidad y calidad de documentación, costos asociados por un retraso en la entrega o con un producto defectuoso, limitaciones gubernamentales.

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos relacionados con el cliente
 - Riesgos asociados con la sofisticación del cliente y la habilidad del desarrollador para comunicarse con el cliente en los momentos oportunos.
 - Los clientes tienen diferentes necesidades: algunos saben lo que quieren, otros no.
 - Los clientes también tienen varios tipos de asociaciones con sus suministradores.
 - Los clientes se contradicen a menudo.
 - Ejemplo: si trabajó con el cliente, si el cliente tiene la idea formal de lo que quiere, aceptará gastar tiempo en reuniones, está dispuesto a mantener una comunicación fluida, está dispuesto a participar en revisiones, es sofisticado técnicamente en áreas del producto, si está dispuesto dejar a su personal hacer el trabajo, si entiende el proceso de software.
 - Si alguna de las respuestas es no, se deberá realizar una investigación más profunda para valorar el potencial del riesgo.

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos del proceso
 - Riesgos asociados con el grado definición del proceso de software y su seguimiento por la organización de desarrollo. Si la mayoría son no, el software es débil y tiene un alto riesgo. Ejemplo de aspectos del proceso: se usa una metodología y cual, se siguen todos los pasos de ingeniería de software, los gestores están de acuerdo, se emplea en todos los proyectos, se adquirieron cursos de formación para los participantes, se desarrollaron diseños de documentos para las entregas, se realizan revisiones técnicas formales, se documenta todo, existe algo para mantener consistencia en la documentación, existe algún mecanismo de control de cambios, seguimiento y revisión de subcontrataciones.
 - Ejemplo de aspectos técnicos: se usan técnicas de especificaciones de aplicaciones para ayudar la comunicación con el cliente, se emplean métodos específicos para análisis, diseño de datos y arquitectónico, se documenta el código con reglas específicas, emplea herramientas de software para la planificación y el seguimiento de actividades, emplea métodos específicos para el diseño de los casos de prueba, se emplean herramientas de gestión de configuración para controlar y seguir cambios, se emplean herramientas para: apoyar el análisis y diseño, crear prototipos, para dar soporte a los procesos de prueba, para soportar la producción y gestión de la documentación, se establecieron métricas de calidad y productividad

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos tecnológicos
 - Riesgos asociados con la complejidad del sistema a construir y la tecnología de punta que contiene el sistema.
 - Por ejemplo: la tecnología a construir es nueva, demandan nuevos requisitos la creación de nuevos algoritmos o tecnología de entrada o salida, el software interactúa con hardware nuevo o no probado, el software a construir interactúa con software suministrado por el vendedor que no se hayan probado, el software a construir interactúa con un sistema de BD cuyo funcionamiento o rendimiento no se ha comprobado en este área de aplicación, demanda una interfaz de usuario especial, demanda creación de componentes de programa, distintos de los que se hayan desarrollado, demandan nuevos métodos de análisis, diseño o pruebas, los requisitos demandan métodos de desarrollo no convencionales, como métodos formales, enfoques basados en IA y redes neuronales, los requisitos del producto imponen restricciones de rendimiento, el cliente no está seguro que la funcionalidad pedida sea factible.
 - Si alguna respuesta es afirmativa, se debería realizar una investigación más profunda para valorar el riesgo potencial

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos del Entorno de Desarrollo
 - Riesgos asociados con la disponibilidad y calidad de las herramientas que se van a emplear para la construcción del producto. Si la mayoría son no, el entorno es débil y tiene un alto riesgo.
 - Por ejemplo: tenemos herramientas de gestión de proyectos, de gestión de procesos, de análisis, de diseño, compiladores, generadores de código apropiados para el producto, tenemos herramientas de pruebas, de gestión de configuración de software, existen herramientas integradas entre sí, existen expertos en el uso de las herramientas, y se capacitó al personal en ellas, es adecuada la ayuda on-line y documentación de las herramientas.

- Gestión de riesgo - Identificación de Riesgo (Cont.)
- Riesgos asociados al tamaño del grupo y experiencia
 - Riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo. Si alguna respuesta es negativa, se debería realizar una investigación más profunda para valorar el riesgo potencial.
 - Por ejemplo: disponemos de la mejor gente, tiene los conocimientos adecuados, existe suficiente personal, se ha asignado al personal para toda la duración del proyecto, habrá parte del personal del proyecto que trabaje sólo durante parte de él, dispone el personal con las expectativas correctas, recibió formación adecuada, será mínimo el movimiento de personal.

- Gestión de riesgo - Identificación de Riesgo (Cont.)
 - Componentes y controladores de riesgo
 - Según la FA de EEUU tiene otro enfoque para identificar riesgos y evitarlos y requiere que el administrador del proyecto identifique los **controladores de riesgo** que afectan a los **componentes de riesgo**.
 - Los componentes de riesgo se definen de la siguiente manera:
 - ✓ **Riesgo de rendimiento:** el grado de incertidumbre con que el producto encontrará sus requisitos y se adecue para su empleo pretendido.
 - ✓ **Riesgo de costo:** el grado de incertidumbre que mantendrá el presupuesto del proyecto.
 - ✓ **Riesgo de soporte:** el grado de incertidumbre de la facilidad del SW para corregirse, adaptarse y ser mejorado.
 - ✓ **Riesgo de la planeación temporal:** el grado de incertidumbre de la planeación y que el proyecto se entregue a tiempo.
- El impacto de cada controlador de riesgo se divide en cuatro categorías: despreciable, marginal, crítico y catastrófico.

- Gestión de riesgo - Proyección de Riesgo
 - También denominada **estimación de riesgo**, intenta medir cada riesgo de dos maneras - la probabilidad de que el riesgo sea real y las consecuencias de los problemas asociados con el riesgo, si ocurriera.
 - Actividades:
 - ✓ Establecer una escala que refleje la probabilidad percibida del riesgo
 - ✓ Definir las consecuencias del riesgo
 - ✓ Estimar el impacto del riesgo en el proyecto y en el producto
 - ✓ Apuntar con exactitud general de la proyección del riesgo de manera que no haya confusiones

- Gestión de riesgo - Proyección de Riesgo (Cont.)
- Desarrollo de una tabla de riesgo
 - La tabla de riesgo le proporciona al jefe del proyecto una sencilla técnica para la proyección de riesgo.
 - **Primera columna:** todos los riesgos posibles
 - **Segunda columna:** categorización del riesgo
 - **Tercera columna:** probabilidad de aparición del riesgo
 - **Cuarta columna:** Cada componente de riesgo se valora usando la caracterización (catastrófico, crítica, marginal o despreciable).
 - Las categorías por cada uno de los cuatro componentes de riesgo - rendimiento, soporte, coste y planificación temporal- son promediadas para determinar un valor general de impacto.
 - Luego la tabla se ordena por **probabilidad y por impacto**. Los riesgos de alta probabilidad y de alto impacto pasan a lo alto de la tabla. (Priorización de riesgos)
 - Se define una **línea de corte**, donde se identifican los riesgos a considerar en la siguiente columna Plan de reducción, supervisión y gestión del riesgo (RSGR)
 - Los controladores de riesgo pueden valorarse en una escala de probabilidad cualitativa con los siguientes valores: imposible, improbable, probable y frecuente. (dar rango numérico a cada valor)

- Gestión de riesgo - Proyección de Riesgo (Cont.)
- Evaluación del impacto del riesgo
 - Tres factores afectan a las consecuencias probables de un riesgo, si ocurre:
 - ✓ La naturaleza del riesgo indica los problemas probables que aparecerán si ocurre. (interfaz externa mal definida)
 - ✓ El alcance de un riesgo combina la severidad (como de serio es el problema) con su distribución general (qué proporción del proyecto será afectado y cuántos clientes se verán perjudicados).
 - ✓ La temporización del riesgo considera cuando y por cuánto tiempo se dejará sentir el impacto. Un jefe de proyecto prefiere las malas noticias cuanto antes, pero en algunos casos, cuanto más tarden mejor.
 - Para que sea útil una evaluación se debe definir un nivel de referencia de riesgo. Para los componentes de riesgo, rendimiento, costo, soporte y planificación temporal también representan niveles de referencia de riesgos.
 - Este nivel de referencia o punta de ruptura sirve para analizar la decisión de seguir con el proyecto o dejarlo.

- Gestión de riesgo - Proyección de Riesgo (Cont.)
- Reducción, Supervisión y Gestión de riesgo
 - Se debe desarrollar una estrategia para tratar los riesgos. Una estrategia eficaz debe considerar:
 - ✓ **Evitar el riesgo:** Si un equipo adopta un enfoque proactivo frente al riesgo, evitarlo es siempre la mejor estrategia. Esto se consigue desarrollando un plan de reducción de riesgo. (Reducir la movilidad)
 - ✓ **Supervisar el riesgo:** El jefe de proyecto supervisa factores que pueden proporcionar una indicación de si el riesgo se está haciendo más o menos probable.
 - ✓ **Gestión del riesgo y planes de contingencia:** Asumen que los esfuerzos de reducción han fracasado y que el riesgo se ha convertido en realidad
 - Es importante advertir que los pasos RSGR provocan aumentos del costo del proyecto.
 - El Seguimiento y Control de Riesgos es el proceso de identificar, analizar y planificar nuevos riesgos, realizar el seguimiento de los riesgos identificados, volver a analizar los riesgos existentes, realizar el seguimiento de los riesgos residuales y revisar la ejecución de las respuestas a los riesgos mientras se evalúa su efectividad. Este es un proceso continuo que se realiza durante la vida del proyecto.

- Qué es un Requerimiento ?
 - Un **requerimiento** es una condición o capacidad a la que el sistema (siendo construido) debe conformar. [Rational]
 - Un **requerimiento de software** puede ser definido como:
 - ✓ Una capacidad del software necesaria por el usuario para resolver un problema o alcanzar un objetivo.
 - ✓ Una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar u otra documentación formal.
- Tipos de Requerimientos
 - Los **requerimientos de usuario** representan el conjunto completo de resultados a ser obtenidos utilizando el sistema.
 - Los **requerimientos de sistema** deben mostrar todo lo que el sistema debe hacer más todas las restricciones sobre la funcionalidad.
- ↳ Los requerimientos conforman el modelo completo, representando el sistema total a algún nivel de abstracción.

- Ingeniería de Requisitos
 - El **proceso de desarrollo de la especificación de requisitos** es lo que se denomina ingeniería de requisitos.
 - Es la primera fase del ciclo de vida del SW donde **se produce una especificación** a partir de ideas informales.
 - Se obtienen y documentan los **requisitos funcionales y no funcionales**.
- Proceso de Ingeniería de requisitos
 - **Elicitación de requisitos:** es probablemente el más crítico y difícil de realizar. Tiene como **objetivos** buscar, investigar y ayudar a los clientes y usuarios a documentar sus necesidades.
 - **Análisis de requisitos:** consiste en la separación de las partes de un todo hasta llegar a conocer sus elementos. Tiene como **objetivo principal** detectar conflictos en los requisitos obtenidos, normalmente mediante técnicas de modelado y prototipado.

- ¿Qué describe un requisito?
 - Los requisitos deben mostrar a los desarrolladores y usuarios qué se necesita de un sistema. Proveerlos forma parte de un lenguaje que todos comprenden, ya que todos están involucrados, incluyendo los clientes.
 - Ejemplos de requisitos:
 - ✓ Una utilidad para el usuario: Los textos tienen que incluir comprobación y corrección gramatical.
 - ✓ Una propiedad general del sistema: El sistema ha de garantizar que la información personal solamente será accesible mediante autorización explícita.
 - ✓ Una restricción general del sistema: El sensor se debe muestrear 10 veces por segundo.
 - ✓ Cómo llevar a cabo cierto cálculo: Calificación final = nota examen + nota trabajo.
 - ✓ Una restricción sobre el desarrollo del sistema: El sistema debe implementarse en C#.

- Captura de requisitos
 - Es el acto de descubrimiento, el **proceso de averiguar lo que se debe construir**.
 - Lo primero que consideramos es la **interacción con los potenciales usuarios** de nuestra aplicación. Muchas veces se cree que del diálogo con cada uno de ellos aparecerán todas las funcionalidades que nuestro desarrollo debe tener, ya que son ellos quienes realmente conocen el negocio. Pero la realidad nos dice otra cosa.
 - Cada persona puede darnos solo “su” punto de vista, su forma de ver una determinada función, pero lo que nosotros realmente estamos buscando es una **visión global e integradora** que apunte a dar soporte a la misión para la cual se construye el sistema y no sólo el soporte a los usuarios. Además debemos tener presente que inconscientemente **se pueden agregar vicios personales** a la función que él está desempeñando y de esa forma podemos caer en el riesgo de incorporar estos vicios en el nuevo sistema.
 - Para eliminar todos estos potenciales conflictos de requerimientos es indispensable que además de captar requerimientos de usuarios finales lo hagamos con **usuarios de niveles superiores en la jerarquía organizacional** y que contemos con analistas que realmente puedan **integrar y depurar toda la información** recabada.

- Captura de requisitos (Cont.)
 - Para lograr una captura de requisitos adecuada y ordenada es recomendable utilizar un flujo de trabajo que incluya los siguientes pasos:
 - ✓ Enumerar los requisitos candidatos
 - ✓ Comprender el contexto del sistema
 - ✓ Capturar los requisitos funcionales
 - ✓ Capturar los requisitos no funcionales
- Técnicas de Relevamiento
 - La recopilación de requerimientos consiste en emplear **diferentes técnicas y herramientas** para hallar los datos de interés, que luego de verificar su veracidad se convertirán en hechos.
 - Algunas técnicas que pueden utilizarse son:
 - ✓ Recopilación de datos impresos
 - ✓ Entrevistas
 - ✓ Cuestionarios
 - ✓ Observación, etc.

- Técnicas de Relevamiento - Muestreo
 - El **muestreo** es el proceso de seleccionar sistemáticamente elementos representativos de una población.
 - Cuando estos elementos seleccionados son examinados de cerca, se supone que el análisis revelará información útil acerca de una población como un todo.
 - El analista de sistemas tiene que tomar una decisión sobre dos puntos importantes:
 - ✓ Hay muchos reportes, formularios, documentos de salida y memorándums que han sido generados por miembros de la organización. ¿A cuáles de éstos se debe prestar atención y cuáles ignorar?
 - ✓ Gran cantidad de empleados pueden ser afectados por el sistema de información propuesto. ¿A qué personas debe entrevistar el analista de sistemas, buscar información por medio de cuestionarios y observar en el proceso de llevar a cabo sus papeles de toma de decisión?

- Técnicas de Relevamiento - Necesidad del muestreo
 - Hay razones por las que un analista de sistemas quiera seleccionar una muestra representativa de los datos a examinar, o personas representativas a entrevistar, aplicar cuestionarios u observar:
 - ✓ Reducir costos
 - ✓ La agilización de la recolección de datos
 - ✓ La mejora de la efectividad
 - ✓ La reducción de la parcialidad (ascendencia)
 - Analizar cada pedazo de papel, hablar con todos y leer cada página Web de la organización sería demasiado costoso.
 - El muestreo ayuda a acelerar el proceso, recolectando datos seleccionados en vez de todos los datos de la población completa. Además, el analista de sistemas evita el problema de analizar todos esos datos de la población.
 - La efectividad en la recolección de datos también es una consideración importante. El muestreo puede ayudar a mejorar la efectividad si se obtiene información más precisa.
 - Finalmente, es posible reducir la parcialidad en la recopilación de datos mediante el muestreo. Cuando el analista de sistemas entrevista a un ejecutivo de la corporación, éste debe estar involucrado con el proyecto.

- Técnicas de Relevamiento - Diseño del muestreo
 - Los pasos para diseñar una buena muestra son:
 - ✓ **Determinar los datos a ser recolectados o descritos:** En este punto los deberes y responsabilidades del analista de sistemas consisten en identificar las variables, atributos y los elementos relacionados con los datos que necesitan recopilarse en la muestra. Se deben considerar los objetivos del estudio así como el método de recopilación de datos que se utilizará
 - ✓ **Determinar la población a ser muestreada:** Se tiene que decidir si los últimos dos meses son suficientes para el análisis, o si éste requiere un año completo de informes. Del mismo modo, al decidir a quién entrevistar, el analista de sistemas tiene que determinar si la población debe incluir un solo nivel de la organización o todos los niveles, gerentes, vendedores, etc.
 - ✓ **Seleccionar el tipo de muestra:** Se puede utilizar uno de cuatro tipos principales de muestras: de conveniencia, intencional, simple y compleja.
 - ✓ **Decidir el tamaño de muestra:** Es importante recordar que en el muestreo es de mayor importancia el número absoluto que el porcentaje de la población y es necesario establecer un tamaño de muestra mayor que uno pero menor que el tamaño mismo de la población.

- Técnicas de Relevamiento - Investigación de datos impresos
 - Conforme el analista de sistemas trabaja para comprender la organización y a sus requerimientos de información, llega a ser importante el **examen de diferentes tipos de datos impresos** que proporcionan información que no se encuentra disponible por ningún otro método de recopilación de datos. Los datos impresos revelan dónde ha estado la organización y hacia dónde creen sus miembros que están yendo. El analista necesita examinar los datos impresos, tanto cuantitativamente como cualitativamente, para ensamblar una imagen precisa.
 - 1. Documentos cuantitativos (son aquellos que miden algo)
 - ✓ Reportes usados para la toma de decisiones
 - ✓ Reportes de desempeño real vs. programado
 - ✓ Registros de movimientos (p.e. ingreso de insumos)
 - ✓ Formularios de captura de datos
 - 2. Documentos cualitativos (sin un formato predefinido o su contenido puede ser muy variable, difícil de acotar a un dominio de valores posibles)
 - ✓ Memorándums, notas
 - ✓ Consignas en el tablero de noticias o en áreas de trabajo
 - ✓ Manuales de funciones y procedimientos
 - ✓ Manuales de política

- Técnicas de Relevamiento - Entrevistas
 - Una entrevista para recolección de información es una conversación dirigida con un propósito específico que usa un formato de preguntas y respuestas.
 - En la entrevista se quiere obtener la opinión del entrevistado y sus sentimientos acerca del estado actual del sistema, los objetivos de la organización, los personales y los procedimientos informales.
 - Observe las opiniones de la persona a la que está entrevistando.
 - Las opiniones pueden ser más importantes y más reveladoras que los hechos.
 - Además de las opiniones se debe tratar de capturar los sentimientos del entrevistado. Recuerde que el entrevistado conoce la organización mejor que usted.
 - Los sentimientos expresados ayudan a capturar la emoción y las actitudes.

- Técnicas de Relevamiento - Entrevistas (Cont.)
 - Los objetivos son información importante que puede ser recogida de las entrevistas.
 - Los hechos que se obtienen de los datos relevantes pueden explicar el desempeño pasado, pero los objetivos proyectan el futuro de la organización.
 - Trate de encontrar tantos objetivos como sea posible a partir de las entrevistas. Tal vez no sea capaz de determinar los objetivos por medio de ningún otro método de recopilación de datos.
 - Se necesita dar confianza y comprensión rápidamente, pero al mismo tiempo, se debe mantener el control de la entrevista. La manera para comenzar a hacer esto es planear la entrevista antes de ir a ella, para que le sea natural conducirla.
 - El diseño conjunto de aplicaciones (JAD) puede servir como una alternativa a las entrevistas persona a persona en determinadas circunstancias.

- Técnicas de Relevamiento - Entrevistas - Planificación
 - Los pasos principales para la preparación de entrevista son:
 - ✓ **Lectura del material a fondo:** Lea y comprenda tanta **información** acerca del entrevistado y su organización como sea posible. El material se puede obtener haciendo una llamada rápida a la persona de contacto para pedirle un reporte anual, una carta corporativa y cualquier publicación que ha sido enviada para explicar la organización ante el público.

Sensibilícese particularmente con el **lenguaje** que usan, para construir un vocabulario común.

Otro beneficio de la investigación sobre la organización es **maximizar el tiempo** que se gasta en la entrevista, en vez de perder tiempo preguntando asuntos generales de fondo.
 - ✓ **Establecimiento de los objetivos de la entrevista:** Use la información que recopiló, así como su propia experiencia para establecer los objetivos de la entrevista. Existen áreas que se relacionan con el procesamiento de la información y con el comportamiento para la toma de decisiones acerca de las cuales querrá hacer preguntas, estas son: fuentes de información, formatos de la información, frecuencia de la toma de decisiones, cualidades de la información y estilo de la toma de decisiones.

- Técnicas de Relevamiento - Entrevistas – Planificación (Cont.)
 - ✓ **Decidir a quién entrevistar:** Cuando esté decidiendo a quién entrevistar incluya a gente clave de todos los niveles que serán afectados por el sistema en alguna forma. Trate de obtener balance para que sean tratadas tantas necesidades de usuarios como sean posibles. El contacto de la organización también tendrá algunas ideas sobre quién debe ser entrevistado.
 - ✓ **Prepare el entrevistado:** Prepare a la persona a ser entrevistada, llamándole con anticipación y permitiendo que el entrevistado tenga tiempo para pensar acerca de la entrevista. Acomode tiempo para llamadas telefónicas y reuniones. Las entrevistas deben durar de 45 minutos a una hora, a lo mucho. Sin importar que tan deseoso esté el entrevistado para extender la entrevista, recuerde que cuando ellos emplean tiempo con usted, no están haciendo su trabajo. Si las entrevistas van más allá de una hora, es probable que los entrevistados resientan la intrusión, sin importar si manifiestan o no su resentimiento.
 - ✓ **Decida sobre tipos de preguntas y estructuras:** Escriba preguntas para tratar las áreas principales de la toma de decisiones, descubiertas cuando se averiguaron los objetivos de la entrevista. Las técnicas adecuadas de cuestionamiento son el corazón de la entrevista. Las preguntas tienen algunas formas básicas que es necesario saber. Los dos tipos de preguntas son **abiertas y cerradas**.

- Técnicas de Relevamiento - Entrevistas - Tipos de Preguntas
 - **Abiertas**
 - Las preguntas abiertas describen las opciones del entrevistado para responder. La respuesta puede ser de dos palabras o de dos párrafos. Por ejemplo: ¿qué opina acerca de? ¿cómo cree que afectará...? ¿dónde se presenta la mayoría de los problemas y por qué?
 - Beneficios:
 - ✓ Pone comfortable al entrevistado
 - ✓ Permite que el entrevistador recoja vocabulario del entrevistado, el cual refleja su educación, valores, actitudes y creencias.
 - ✓ Proporciona riqueza de detalles.
 - ✓ Revela caminos para preguntas posteriores que podrían haber quedado sin atacar.
 - ✓ Hace que sea más interesante para el entrevistado.
 - ✓ Permite más espontaneidad.
 - ✓ Hace que la construcción de frases sea más fácil para el entrevistador.
 - ✓ Se les puede usar en un aprieto si es que el entrevistador es tomado por sorpresa.

- Técnicas de Relevamiento - Entrevistas - Tipos de Preguntas (Cont.)
 - Desventajas:
 - ✓ El hacer preguntas que puedan dar como resultado mucho detalle irrelevante.
 - ✓ La posibilidad de perder el control de la entrevista.
 - ✓ El permitir respuestas que pueden llevarse demasiado tiempo para la cantidad de información útil obtenida.
 - ✓ Puede mostrar potencialmente que el entrevistador no está preparado.
 - ✓ Pueden dar la impresión de que el entrevistador está sin un objetivo real de la entrevista.

- Técnicas de Relevamiento - Entrevistas - Tipos de Preguntas (Cont.)
 - **Cerradas**
 - Las preguntas cerradas limitan las respuestas disponibles al entrevistado. Se le hace una pregunta y se le da un número de respuestas, pero no se le permite que escriba la suya propia y que cuente como una respuesta correctamente contestada. Por ejemplo: ¿Cuántas personas trabajan en este sector? ¿quién realiza esta tarea?
 - Beneficios:
 - ✓ Se ahorra tiempo
 - ✓ Se facilita la comparación de entrevistas
 - ✓ Se llega al punto
 - ✓ Se mantiene el control sobre la entrevista
 - ✓ Se tratan muchos temas rápidamente
 - ✓ Se obtienen datos relevantes
 - Desventajas:
 - ✓ Suelen ser aburridas para el entrevistado.
 - ✓ No llegan a obtener grandes detalles (debido a que el entrevistador brinda el marco de referencia).
 - ✓ Se pierden ideas principales por la razón anterior.
 - ✓ No se llega a establecer una relación entre el entrevistado y el entrevistador.

- Técnicas de Relevamiento - Entrevistas - Tipos de Preguntas (Cont.)
- Atributos de las preguntas abiertas y cerradas

Atributo	Abiertas	Cerradas
Confiabilidad de los datos	Bajo	Alto
Uso eficiente del tiempo	Bajo	Alto
Precisión de los datos	Bajo	Alto
Anchura y profundidad	Mucho	Poco
Aptitud del entrevistador requerida	Mucho	Poco
Facilidad de análisis	Difícil	Fácil

- Técnicas de Relevamiento - Entrevistas - Estructuradas contra No Estructuradas
 - **Estructuradas**
 - En una entrevista estructurada todo está planeado y el plan es seguido estrictamente. Las preguntas cerradas son la parte medular de una entrevista completamente estructurada.
 - **No Estructuradas**
 - Una entrevista no estructurada se refiere simplemente al orden en que son hechas las preguntas y no implica la falta de otra preparación.
- ⇒ La conciencia de los compromisos entre las entrevistas estructuradas y no estructuradas le permitirá tomar una mejor decisión acerca de cuál tipo de entrevista es más adecuado para una situación particular.

- Técnicas de Relevamiento - Entrevistas - Estructuradas contra no estructuradas (Cont.)
- Atributos de las entrevistas estructuradas y no estructuradas

Atributo	Estructuradas	No estructuradas
Evaluación	Fácil	Difícil
Cantidad de tiempo requerido	Bajo	Alto
Entrenamiento requerido	Limitado	Muy necesario
Permite espontaneidad	Poco	Mucho
Proporciona perspicacia del entrevistado	Muy pequeño	Mucha oportunidad
Flexibilidad	Pequeño	Grande
Control del entrevistador	Alto	Bajo
Precisión	Alto	Bajo
Confiabilidad	Alto	Bajo
Amplitud y profundidad	Bajo	Alto

- Técnicas de Relevamiento - Entrevistas - Conducción
 - Cuando llegue salude de mano y con firmeza al entrevistado, esta situación ayuda a establecer la credibilidad y confianza.
 - Tome la grabadora y/o su cuaderno de notas.
 - Dependiendo de la estructura que se va a seguir en la entrevista se puede comenzar con algunas preguntas abiertas generales y no amenazadoras. El abrir la entrevista de esta forma ayuda a relajarlo a usted y al entrevistado. Escuchando cuidadosamente las primeras respuestas se puede atrapar el vocabulario y la jerga.
 - Las primeras respuestas abiertas también pueden revelar actitudes, moral y creencias del entrevistado que le ayudarán a comprender la manera en que usa la información y cómo se siente hacia los demás en la organización. Se debe escuchar y responder adecuadamente a lo que el entrevistado esté diciendo.

- Técnicas de Relevamiento - Entrevistas – Conducción (Cont.)
 - Usted controle la entrevista.
 - Todo el material de la entrevista debe ser cubierto en 45 minutos y usted está consciente de la planificación y administración necesarias para lograr esto. El cerrar adecuadamente la entrevista es igualmente importante que la apertura.
 - Durante la entrevista regrese a algunas respuestas del entrevistado por medio de la sumarización para volver a confirmar que se comprendió lo que quería decir. Si en cualquier momento no está seguro, se deben pedir definiciones u otro tipo de aclaraciones.
 - Conforme concluya la entrevista hay otros procedimientos a seguir. Resuma y haga saber sus impresiones generales. Infórmele al entrevistado acerca de los pasos subsecuentes a tomar y qué es lo que harán a continuación usted y otros miembros del equipo. Fije citas futuras para entrevistas de seguimiento, déle las gracias al entrevistado por haberle dado su tiempo y despídase con un apretón de manos.

- Técnicas de Relevamiento - Cuestionarios
 - Los cuestionarios son una técnica de recopilación de información que permite a los analistas de sistemas estudiar actitudes, creencias, comportamientos y características de varias personas principales en la organización y que pueden ser afectadas por los sistemas actual y propuesto.
 - ✓ Las **actitudes** son lo que la gente de la organización dice que quiere.
 - ✓ Las **creencias** son lo que la gente piensa que es, de hecho cierto.
 - ✓ El **comportamiento** es lo que hacen los miembros de la organización.
 - ✓ Las **características** son propiedades de las personas o cosas.
 - Las respuestas obtenidas mediante los cuestionarios usando **preguntas cerradas** pueden ser cuantificadas.
 - Las respuestas a cuestionarios que utilizan **preguntas abiertas** se analizan e interpretan de otras maneras.
 - Las preguntas sobre actitudes y creencias son notablemente sensibles a la redacción escogida por el analista de sistemas.
- Mediante el uso de cuestionarios el analista puede estar buscando cuantificar lo que ha encontrado en las entrevistas.

- Técnicas de Relevamiento - Cuestionarios (Cont.)
 - Hay muchas similitudes entre la entrevista y el cuestionario, y tal vez el ideal sería usarlas en conjunción, ya sea averiguando respuestas no claras de los cuestionarios con una entrevista, o diseñando el cuestionario con base en lo que se descubrió en la entrevista. Sin embargo, cada técnica tiene sus propias funciones específicas y no es siempre necesario o deseable usar ambos.
 - A primera vista el cuestionario puede parecer una forma rápida para recolectar enormes cantidades de datos acerca de la manera en que los usuarios valoran el sistema actual, qué problemas tiene y qué espera la gente del nuevo sistema.
 - El desarrollo de un cuestionario útil lleva un **gran tiempo de planificación**.
 - Los cuestionarios se deben utilizar cuando:
 - ✓ La cantidad de personas a entrevistar es grande e interesa conocer qué proporción de la muestra aprueba o no algo.
 - ✓ Las personas están muy dispersas.
 - ✓ Se desea realizar un sondeo de opinión.

- Técnicas de Relevamiento - Cuestionarios (Cont.)
- Compromiso entre el uso de preguntas abiertas y cerradas en cuestionarios

Atributo	Abiertas	Cerradas
Velocidad de llenado	Lento	Rápido
Naturaleza exploratoria	Alto	Bajo
Amplitud y profundidad	Alto	Bajo
Facilidad de preparación	Fácil	Difícil
Facilidad de análisis	Difícil	Fácil

- Técnicas de Relevamiento - Observación
 - La observación tanto del tomador de decisiones como del ambiente físico de éste son técnicas importantes de recopilación de información para el analista de sistemas.
 - Mediante la observación de las actividades de los **tomadores de decisiones**, el analista busca obtener una percepción de lo que realmente se hace y no sólo de lo que está documentado o explicado. Además, al observar al tomador de decisiones, el analista intenta ver de primera mano las relaciones que existen entre éstos y los demás miembros de la organización.
 - El analista examina los elementos físicos del **espacio de trabajo** del tomador de decisiones para ver su influencia sobre el comportamiento del mismo. Además, mediante la observación de los elementos físicos sobre los que el tomador de decisiones tiene control, el analista trabaja para comprender qué mensaje está enviando este tomador de decisiones. El analista trabaja para comprender la influencia del tomador de decisiones sobre los demás en la organización.
 - Los analistas usan la observación por muchas razones. Una razón es para obtener información acerca de los tomadores de decisiones y su ambiente, que está inaccesible para cualquier otro método. La observación también ayuda a confirmar o negar e invertir lo que ha sido encontrado por medio de entrevistas, cuestionarios y otros métodos.

- JAD (Diseño Conjunto de Aplicaciones)
 - Sin importar que tan experto se llegue a ser como entrevistador, inevitablemente se experimentarán situaciones donde las entrevistas persona a persona no son tan útiles como uno quisiera.
 - Las entrevistas personales son consumidoras de tiempo, propensas a error y sus datos son propensos a malas interpretaciones. Un enfoque alternativo a la entrevista de usuarios uno en uno, llamado **Diseño conjunto de Aplicaciones (JAD)** fue desarrollado por **IBM**.
 - La motivación para el uso de JAD es:
 - ✓ **Reducir el tiempo** (y por lo tanto el costo) requerido para las entrevistas personales.
 - ✓ **Mejorar la calidad** de los resultados de la valoración de requerimientos de información.
 - ✓ **Generar una mayor identificación del usuario** con el nuevo sistema de información a consecuencia del proceso participativo.
 - Aunque JAD puede ser sustituido con entrevistas personales en cualquier momento apropiado del ciclo de vida de desarrollo de sistema, normalmente ha sido usado como una técnica que le permite, como analista de sistemas, lograr los requerimientos de análisis y diseño de interfaz de usuario conjuntamente con los usuarios en un grupo.

- JAD (Cont.)
 - JAD requiere algunas aptitudes especializadas por parte del analista y muchas aptitudes y un compromiso firme por parte de la organización y los usuarios que se comprometerán a usar este enfoque. Sin embargo, en determinadas situaciones el JAD puede ser efectivo y debe ser considerado como una alternativa a métodos más tradicionales de análisis de sistemas.
- Condiciones que dan soporte al uso del JAD
 - La siguiente lista de condiciones le ayudará a decidir cuándo puede ser beneficioso el uso del JAD:
 1. Los grupos de usuarios están impacientes y quieren algo nuevo, y no una solución estándar a un problema típico
 2. La cultura organizacional da soporte a los comportamientos de la solución de problemas en conjunto entre varios niveles de empleados.
 3. Los analistas predicen que la cantidad de ideas generadas por medio de entrevistas persona a persona no será tan abundante como la cantidad de ideas posibles del ejercicio de un grupo amplio.
 4. El flujo de trabajo de la organización permite la ausencia de personal importante durante un período de tiempo de dos a cuatro días.

- JAD - Participantes
 - Las sesiones de JAD incluyen una variedad de participantes, analistas, ejecutivos, etc., que contribuirán con sus diferentes experiencias y aptitudes a las sesiones. El interés principal de usted aquí es que todos los miembros del equipo del proyecto estén abiertos al enfoque JAD y lleguen a estar involucrados.
 - Escoja un **patrocinador ejecutivo**, una persona de alto nivel que abra y cierre las sesiones de JAD. De preferencia, seleccione a un ejecutivo del grupo de usuarios que tenga algún tipo de autoridad sobre la gente del sistema de información que está trabajando en el proyecto. Esta persona será un símbolo visible e importante de la disposición organizacional al proyecto del sistema.
 - Debe estar presente al menos un **analista de sistemas** de información, pero el analista, por lo general, toma un papel pasivo, a diferencia de las entrevistas tradicionales, donde el analista controla la interacción. Como analista de proyecto debe estar presente durante el JAD para escuchar lo que dicen los usuarios y lo que requieren. Adicionalmente, se querrá dar una opinión de experto acerca de cualquier costo desproporcionado o soluciones propuestas durante la sesión de JAD misma. Sin este tipo de retroalimentación inmediata, puede ser que se deslicen soluciones no realistas con costos excesivos en la propuesta y posteriormente sea costoso desvalorarlas.

- JAD – Participantes (Cont.)
 - Se puede escoger de **8 a 12 usuarios** de cualquier rango para que participen en las sesiones de JAD. Trate de seleccionar usuarios que estén arriba del nivel de empleado de oficina que pueda decir claramente qué información necesita para ejecutar su trabajo, así como qué es lo que desea en un sistema de computadora nuevo o mejorado. Aquí se pueden emplear adecuadamente algunas de las **ideas sobre el muestreo**, debido a que el objetivo es obtener una muestra representativa de usuario sin formar un grupo muy grande que llegue a ser inmanejable durante las interacciones del grupo.
 - El **líder de sesión** no debe ser un experto en análisis y diseño de sistemas, sino alguien que tenga habilidades excelentes de comunicación para facilitar las interacciones adecuadas. Tome en cuenta que no se quiere usar un líder de sesión que reporte a otra persona del grupo. Para evitar esta posibilidad, una organización podría contratar a un consultor externo que trabaje como líder de sesión. El punto es contar con una persona que atraiga la atención del grupo para tratar las cuestiones importantes de los sistemas, negociar satisfactoriamente, resolver los conflictos y ayudar a los miembros del grupo a alcanzar un acuerdo general.

- JAD – Participantes (Cont.)
 - La sesión de JAD debe incluir también **uno o más observadores** que sean analistas o expertos técnicos de otras áreas funcionales para proporcionar explicaciones técnicas y consejos al grupo durante las sesiones.
 - Además, **un escribano** del departamento de sistemas de información debe asistir a las sesiones de JAD para escribir formalmente todo lo que se hace. Asegúrese de que el escribano publique el registro de resultados de JAD rápidamente, una vez que el grupo se haya reunido. La publicación lenta de los resultados lleva al riesgo de perder ahorros de tiempo y el *momentum*, que son las principales motivaciones para el uso de JAD en primer lugar. Considere la selección de un segundo escribano que sea de un departamento de usuarios. Luego, la responsabilidad de las cuestiones del registro de sistemas pueden ser dejadas al escribano del área de sistemas de información, y el escribano del grupo de usuarios pueden tomar nota del contenido relacionado con el negocio.

- JAD - Planificación de la sesión de JAD
 - Una de las claves para un trabajo de grupo JAD exitoso es el poner las bases por medio de estudio y planificación avanzada.
 - El líder escogido para la sesión debe trabajar con un patrocinador ejecutivo para determinar el alcance del proyecto que debe cubrir el JAD.
 - A veces el proyecto requiere más de un taller de JAD, pero algunos de los beneficios del JAD se erosionan si planean muchas sesiones independientes. A continuación de la definición del alcance del proyecto se seleccionarán los participantes, y el líder de la sesión aprenderá la aplicación realizando algunas entrevistas con los usuarios principales.
 - El objeto de las entrevistas es recopilar información para facilitar la comprensión de lo que está sucediendo en el negocio.

- JAD – Niveles y Actividades (Cont.)
 - JAD puede ser utilizado a varios niveles de detalle, utilizando la visión del negocio, análisis de conceptos (a través de análisis de requerimientos) y especificación de diseños.
 - Distintas actividades JAD serán necesarias para cada nivel de detalle.
 -
 - Las actividades JAD típicas son las consistentes en:
 - ✓ Un Plan JAD
 - ✓ Seguido por uno o más Diseños JAD
 - Cada actividad está compuesta por tres fases:
 - ✓ Acostumbramiento
 - ✓ Sesión
 - ✓ Conclusión

- JAD – Plan JAD (Cont.)
 - La sesión de **Plan JAD** usualmente dura entre **uno y cinco días**, dependiendo en el tamaño y la complejidad del sistema.
 - El líder de la sesión guía a los participantes a lo largo de ocho tareas predefinidas:
 - ✓ Orientación: Explicación de objetivos y tareas a desarrollar
 - ✓ Definición de requerimientos de alto nivel: Incluyendo objetivos, beneficios anticipados del sistema, consideraciones estratégicas y futuras, suposiciones y beneficios, seguridad, auditoría y control de requerimientos
 - ✓ Límites y alcances del sistema: Incluyendo diagrama de flujo de negocio, usuarios y ubicaciones, áreas funcionales fuera del alcance del sistema
 - ✓ Identificar y estimar tiempos de los Diseños JAD
 - ✓ Identificar los participantes de los Diseños JAD
 - ✓ Programar días y horarios para los Diseños JAD
 - ✓ Acordar los puntos y consideraciones de la documentación a generar del Plan JAD
 - ✓ Concluir la sesión

- JAD – Diseño JAD (Cont.)
 - La sesión de **Diseño JAD** dura aproximadamente entre **tres y diez días**.
 - El líder de la sesión guía a los participantes a lo largo de las siguientes tareas:
 - ✓ Orientación: Explicación de objetivos y tareas a desarrollar
 - ✓ Revisión y refinación de los requerimientos y alcance del Plan JAD
 - ✓ Desarrollar diagrama de flujo del trabajo
 - ✓ Desarrollar descripción del flujo de trabajo
 - ✓ Identificar funciones y grupos de datos del sistema
 - ✓ Especificar los requerimientos de procesamiento
 - ✓ Acordar los puntos y consideraciones de la documentación a generar del Diseño JAD
 - ✓ Concluir la fase de sesión
 - Las sesiones de JAD están acompañadas por **libros de trabajo** que contienen una colección de formularios predefinidos para los grupos, para ser completados durante la sesión. P.e. formularios de participantes, de resultados, de estimaciones, de salidas por pantalla, de reportes, de descripción de interfaces y de descripción de funciones.

- JAD (Cont.)
 - Además de proveer soporte para analizar requerimientos en sesiones grupales, JAD también contribuye en las áreas de comunicación humana y desarrollo del conocimiento.
 - Para el proceso de requerimientos, JAD soporta:
 - ✓ La articulación del concepto de producto, requerimientos, medición de resultados
 - ✓ Análisis de problemas
 - ✓ Estudios de factibilidad y análisis de opciones de costo-beneficio
 - ✓ Análisis y modelado
 - ✓ La documentación de requerimientos

- JAD (Cont.)
 - En términos de comunicación humana, JAD soporta:
 - ✓ La identificación de varios puntos de vista
 - ✓ La conciliación de los puntos de vista
 - ✓ La revisión por parte del usuario de los modelos desarrollados
 - ✓ El análisis de los propios problemas del usuario y la identificación de la necesidad de cambio
 - JAD colabora con el desarrollo de conocimiento de:
 - ✓ Estructuras relevantes en el trabajo actual del usuario
 - ✓ Visiones y propósitos de diseño
 - ✓ Revisión de las distintas opciones tecnológicas

- JAD - Donde efectuar las reuniones de JAD
 - Si todo esto es posible, recomendamos que se realicen las sesiones de dos a cuatro días en un lugar aparte, **fuera de la organización, en ambientes agradables.**
 - Algunos grupos usan centros ejecutivos o hasta instalaciones con apoyo de decisiones de grupo que encuentran disponibles en las universidades principales.
 - La idea es **minimizar las distracciones y responsabilidades diarias** del trabajo regular de los participantes.
 - El cuarto en sí mismo debe poder alojar confortablemente a las cerca de 20 personas invitadas.
 - Se debe contar con equipo de apoyo a presentaciones, como mínimo debe incluir dos proyectores, un pizarrón blanco, un rotafolio y fácil acceso a una fotocopidora.
 - Las salas destinadas al apoyo a la toma de decisiones en grupo también proporcionarán PC's conectadas en red, un sistema de proyección y software escrito para facilitar la interacción del grupo y que al mismo tiempo se minimicen los comportamientos de grupo improductivos.

- JAD - Donde efectuar las reuniones de JAD (Cont.)
 - Dé importancia adecuada a la creación de confort para los participantes, debido a que ésta será una experiencia intensa, bastante diferente en naturaleza de un día de trabajo típico para la mayoría de las personas. Disponga de suficiente comida, así como bebida, durante los cortes planeados antes del almuerzo y nuevamente avanzada la tarde.
 - **Asistencia Total:** Programe las sesiones de JAD cuando todos los participantes puedan estar dispuestos a asistir. No realice las sesiones a menos que puedan de hecho asistir todos aquellos que hayan sido invitados. Esto es crítico para el éxito de sesiones.
 - Asegúrese de que todos los participantes reciban una **agenda** antes de la reunión, y considere la realización de una reunión de orientación, con duración de medio día, una semana antes aproximadamente del taller del trabajo, para que los que estén involucrados sepan lo que se espera de ellos. Una reunión previa de este tipo le permite moverse rápidamente y desenvolverse con confianza una vez que se haya iniciado la reunión definitiva.

- JAD - Realización de un análisis estructurado de las actividades del proyecto
 - IBM recomienda que las sesiones de JAD examinen estos puntos en los proyectos de sistema propuestos: planificación, recepción, procesamiento y seguimiento de lo recibido, monitoreo y asignación, procesamiento, registro, envío y evaluación.
 - Para cada tema deben ser preguntadas y respondidas las preguntas sobre quién, qué, cómo, dónde y por qué.
 - Claramente los sistemas *ad hoc* o interactivos, tales como el apoyo de decisiones y otros tipos de sistemas que dependen del estilo del tomador de decisiones (incluyendo prototipos de sistemas), no se analizan con tanta facilidad como con el enfoque estructurado de JAD.
 - Siendo el **analista** en las sesiones JAD, usted debe recibir las notas de los escribanos y preparar un documento de las especificaciones técnicas basado en lo que sucedió en la reunión. Presente los objetivos de administración sistemáticamente, así como el alcance y los límites del proyecto. También deben ser incluidos detalles específicos del sistema, incluyendo aquellos sobre la presentación de pantallas y reportes.

- JAD - Beneficios potenciales del uso de JAD en vez de las entrevistas tradicionales
 - Hay cuatro beneficios potenciales principales que usted, los usuarios y el equipo de analistas de sistemas deben considerar cuando valoren las posibilidades del uso del JAD:
 - El primero es el **ahorro de tiempo** sobre las entrevistas tradicionales. Algunas organizaciones han estimado que las sesiones JAD han proporcionado ahorros del 15% del tiempo sobre el enfoque tradicional.
 - Junto a los ahorros de tiempo está el **desarrollo rápido** posible por medio del JAD. Dado que las entrevistas de usuarios no se realizan consecutivamente a lo largo de semanas o meses, el desarrollo puede proceder con mayor rapidez.
 - Un tercer beneficio es la posibilidad de **mejorar el concepto de propiedad del sistema de información**. Como analistas, siempre nos esforzamos por involucrar a los usuarios en formas significativas y los animamos a que sientan como suyos los sistemas que estamos diseñando. Debido a su naturaleza interactiva y a su alta visibilidad, JAD ayuda a los usuarios a involucrarse en las etapas tempranas de los proyectos de sistemas y le da seriedad a la retroalimentación que proporcionan → El trabajo continuo en una sesión de JAD ayuda a reflejar las ideas del usuario en el diseño final.

- JAD - Beneficios potenciales del uso de JAD en vez de las entrevistas tradicionales (Cont.)
 - Un beneficio final de la participación en sesiones JAD es el **desarrollo creativo de diseños**. El carácter interactivo del JAD tiene mucho en común con técnicas de *brainstorming* (lluvia de ideas), que generen nuevas ideas y nueva combinación de éstas debido al ambiente dinámico y estimulante. Los diseños pueden evolucionar por medio de interacciones facilitadas en vez de un aislamiento relativo.
 - **Recordemos:**
 - JAD es una herramienta que permite a los usuarios finales y a los profesionales, diseñar sistemas en forma conjunta, en sesiones grupales.
 - Gibson y Jackson afirman que los resultados de los estudios realizados aplicando esta técnica aumentan de un 20% a un 60% la productividad sobre los métodos de diseño tradicionales.
 - Además, afirman que promueve la cooperación, el entendimiento y el trabajo grupal entre distintos grupos de usuarios y el staff de profesionales.

- JAD - Desventajas potenciales de su uso
 - Hay tres desventajas o fallas que también se deben ponderar cuando se tome una decisión sobre si emplear las entrevistas tradicionales o JAD:
 - La primer desventaja es que JAD requiere la **dedicación** de un gran bloque de tiempo por parte de 18 o 20 participantes. Debido a que JAD requiere la disponibilidad de dos o cuatro días, no es posible hacer ninguna otra actividad en forma concurrente o desplazar en el tiempo ninguna actividad, como se hace típicamente con las entrevistas persona a persona.
 - Una segunda falla sucede cuando la **preparación** de las sesiones JAD es inadecuada en cualquier aspecto o cuando el reporte de seguimiento y la documentación de especificaciones es incompleto. En estos casos el diseño resultante puede ser menos que satisfactorio. Se necesita conjugar muchas variables correctamente para que JAD sea satisfactorio. Caso contrario, muchas cosas pueden salir mal. El éxito del diseño resultante de sesiones JAD es menos predecible que el logrado por medio de entrevistas estándar.
 - Por último, las habilidades organizacionales necesarias y la **cultura organizacional** pueden no estar desarrolladas lo suficiente para el esfuerzo concertado que se requiere para ser productivo en un ambiente JAD. Usted tendrá que tomar la decisión sobre si la organización está realmente dispuesta y preparada para ese enfoque.

- Planificación de proyectos
 - El proceso de gestión de proyectos de software comienza con un conjunto de actividades que, globalmente, se denominan planificación de proyecto.
 - La primera de las actividades es la **estimación**. Aunque la estimación es más un arte que una ciencia, es una actividad importante que no debe llevarse a cabo en forma descuidada. Existen técnicas útiles para la estimación de costos y tiempo.
 - Un **gestor de proyectos** es una persona con la habilidad de saber qué va a ir mal antes de que ocurra y con el coraje para hacer estimaciones cuando el futuro no está claro.
 - La estimación de recursos, costos y planificación temporal de un esfuerzo en desarrollo de software requiere experiencia, acceder a una buena información histórica y el coraje de confiar en medidas cuantitativas cuando todo lo que existe son datos cualitativos.
 - La estimación conlleva un riesgo inherente y es este riesgo el que lleva a la incertidumbre.
 - La **complejidad** del proyecto tiene un gran efecto de incertidumbre, que es inherente en la planificación. Sin embargo, la complejidad es una medida relativa que se ve afectada por la familiaridad con esfuerzos anteriores.

- Planificación de proyectos (Cont.)
 - El **tamaño del proyecto** es otro factor importante que puede afectar a la precisión de las estimaciones. A medida que el tamaño aumenta, crece rápidamente la interdependencia entre varios elementos del software. El problema de la descomposición, un enfoque importante hacia la estimación, se hace más difícil porque los elementos descompuestos pueden ser todavía excesivamente grandes.
 - El **grado de incertidumbre estructural** tiene también efecto en el riesgo de la estimación. En este contexto, la estructura se refiere al grado en el que los requisitos se han definido, la facilidad con la que pueden subdividirse funciones, y la naturaleza jerárquica de la información que debe procesarse.
 - La **disponibilidad de información histórica** también determina el riesgo de la estimación. Podemos emular lo que se ha trabajado y evitar las áreas en donde surgieron problemas. Cuando se dispone de las métricas completas de software de proyectos anteriores, se pueden hacer estimaciones con mayor seguridad; establecer planificaciones para evitar dificultades anteriores, y así reducir el riesgo global.

- Planificación de proyectos (Cont.)
 - El riesgo se mide por el grado de incertidumbre en las estimaciones cuantitativas establecidas por recursos, costos y planificación temporal.
 - Si no se entiende bien el ámbito del proyecto o los requisitos del proyecto están sujetos a cambios, la incertidumbre y el riesgo son peligrosamente altos.
 - El planificador de software debería solicitar definiciones completas de rendimiento y de interfaz.
 - El planificador y, lo que es más importante, el cliente, deben tener presente que cualquier cambio en los requisitos del software significa inestabilidad en el costo y en la planificación temporal.

- Planificación de proyectos - Objetivos
 - El objetivo de la planificación del proyecto de software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costos y planificación temporal.
 - Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto.
 - Además las estimaciones deberían definir los escenarios del mejor caso y peor caso de forma que los resultados del proyecto puedan limitarse.
 - El objetivo de la planificación se logra mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables.

- Planificación de proyectos - **Ámbito del Software**
 - La primera actividad de la planificación del proyecto de software es determinar el **ámbito del software**. Se deben evaluar la función y el rendimiento que se asignaron al software durante la ingeniería del sistema de computadora, para establecer un ámbito del proyecto que no sea ambiguo, ni incomprensible para directivos y técnicos.
 - Se debe delimitar la declaración del ámbito del software.
 - El ámbito del software describe la función, el rendimiento, las restricciones, las interfaces y la fiabilidad.
 - Se evalúan las funciones descritas en el enunciado del ámbito, y en algunos casos se refinan para dar más detalles antes del comienzo de la estimación. Dado que las estimaciones del costo y de la planificación temporal están orientadas a la función, muchas veces es útil llegar a un cierto grado de descomposición.
 - Las consideraciones de rendimiento abarcan los requisitos de tiempo de respuesta y de procesamiento. Las restricciones identifican los límites del software originados por el hardware externo, por la memoria disponible y otros sistemas existentes.

- Planificación de proyectos - Ámbito del Software (Cont.)
- Obtención de la información necesaria para el ámbito
 - Al principio de un proyecto de software las cosas siempre están un poco borrosas. Se ha definido una necesidad y se han enunciado las metas y objetivos básicos, pero todavía no se ha definido la información necesaria para definir el ámbito (prerrequisito para la estimación).
 - La técnica utilizada con más frecuencia para acercar al cliente y al desarrollador, y para hacer que comience el proceso de comunicación es establecer una reunión o una entrevista preliminar. La primera reunión entre un ingeniero de software y el cliente puede compararse a la primera cita entre adolescentes. Ninguna persona sabe lo que decir o preguntar; ambos están preocupados por si lo que dicen es malinterpretado; ambos están pensando hasta dónde podría llegar (probablemente los dos tienen aquí diferentes expectativas); ambos quieren quitárselo pronto de encima; pero al mismo tiempo quieren que salga bien.
 - Sin embargo, se debe iniciar la comunicación. Gause y Weinberg sugieren que el analista comience haciendo **preguntas de contexto libre**. Es decir, una serie de preguntas que lleven a un entendimiento básico del problema, por ejemplo, las personas que están interesadas en la solución, la naturaleza de la solución que se desea, y la efectividad prevista del primer encuentro.

- Planificación de proyectos - Ámbito del Software (Cont.)
- Obtención de la información necesaria para el ámbito
 - El primer conjunto de cuestiones de contexto libre se centran en el cliente, en los objetivos globales y en los beneficios. Por ejemplo, el analista podría preguntar:
 - ✓ ¿Quién está detrás de la solicitud de este trabajo?
 - ✓ ¿Quién utilizará la solución?
 - ✓ ¿Cuál será el beneficio económico de una buena solución?
 - ✓ ¿Hay otro camino para la solución?
 - Las preguntas siguientes permiten que el analista comprenda mejor el problema y que el cliente exprese sus percepciones sobre una solución:
 - ✓ ¿Cómo caracterizaría [el cliente] un resultado «correcto» que se generaría con una solución satisfactoria?
 - ✓ ¿Con qué problema(s) se enfrentará esta solución?
 - ✓ ¿Puede mostrarme (o describirme) el entorno en el que se utilizará la solución?
 - ✓ ¿Hay aspectos o limitaciones especiales de rendimiento que afecten a la forma en que se aborde la solución?

- Planificación de proyectos - Ámbito del Software (Cont.)
- Obtención de la información necesaria para el ámbito
 - La última serie de preguntas se centra en la efectividad de la reunión. Gause y Weinberg las llaman «meta-cuestiones» y proponen la lista siguiente (abreviada):
 - ✓ ¿Es usted la persona apropiada para responder a estas preguntas?
 - ✓ ¿Son «oficiales» sus respuestas?
 - ✓ ¿Son relevantes mis preguntas para su problema?
 - ✓ ¿Estoy realizando muchas preguntas?
 - ✓ ¿Hay alguien más que pueda proporcionar información adicional?
 - ✓ ¿Hay algo más que debiera preguntarle?
 - Estas preguntas (y otras) ayudarán a «romper el hielo» y a iniciar la comunicación esencial para establecer el ámbito del proyecto. Sin embargo, una reunión basada en preguntas y respuestas no es un enfoque que haya tenido un éxito abrumador. En realidad, la sesión P&R sólo se debería utilizar para el primer encuentro, reemplazándose posteriormente por un tipo de reunión que combine elementos de resolución de problemas, negociación y especificación.

- Planificación de proyectos - Ámbito del Software (Cont.)
- Técnicas útiles para la especificación de aplicaciones (TUEA)
 - Los clientes y los ingenieros de software con frecuencia tienen establecido inconscientemente el pensamiento de «nosotros y ellos». En lugar de trabajar como un equipo para identificar y refinar los requisitos, cada uno define su propio «territorio» y se comunica por medio de memorandos, documentos formales de situación, sesiones de preguntas y respuestas e informes. La historia ha demostrado que este enfoque es muy pobre. Abundan los malentendidos, se omite información importante y nunca se establece una relación de trabajo con éxito.
 - Un grupo de investigadores independientes desarrolló un enfoque orientado al equipo para la recopilación de requisitos que pueden aplicarse para ayudar a establecer el ámbito de un proyecto. (facilitated application techniques - FASTI), pertenecen a un enfoque que alienta a la creación de un equipo compuesto de clientes y de desarrolladores que trabajen juntos para identificar el problema, proponer elementos de solución, negociar diferentes enfoques y especificar un conjunto preliminar de requisitos.

- Planificación de proyectos - Recursos

- La segunda tarea de la planificación del desarrollo de SW es la estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de SW.



- En la base de pirámide de recursos se encuentra el entorno de desarrollo -hardware y software- que proporciona la infraestructura de soporte al esfuerzo de desarrollo. En nivel más alto se encuentran los componentes de SW reutilizables - los bloques de SW que pueden reducir drásticamente los costes de desarrollo y acelerar la entrega. En la parte más alta de la pirámide está el recurso primario -las personas. Cada recurso queda especificado mediante cuatro características: descripción del recurso, informe de disponibilidad, fecha cronológica en la que se requiere el recurso, tiempo durante el que será aplicado el recurso. Las dos últimas características pueden verse como una ventana temporal. La disponibilidad del recurso para una ventana específica tiene que establecerse lo más pronto posible.

- Planificación de proyectos – Recursos (Cont.)
- Recursos Humanos
 - El encargado de la planificación comienza elevando el ámbito y seleccionando las habilidades técnicas que se requieren para llevar a cabo el desarrollo. Hay que especificar la posición dentro de la organización (por ejemplo: ingeniero de software experimentado) y la especialidad (por ejemplo: telecomunicaciones, bases de datos, microprocesadores).
 - Para proyectos relativamente pequeños (una persona-año o menos) una sola persona puede llevar a cabo todos los pasos de ingeniería del software, consultando con especialistas siempre que requiera.
 - El número de personas requerido para un proyecto de software sólo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (por ejemplo, personas-mes o personas-año).

- Planificación de proyectos – Recursos (Cont.)
- Recursos de SW reutilizable
 - Cualquier estudio sobre recursos de SW estaría incompleto sin estudiar la reutilización, esto es, la creación y la reutilización de bloques de construcción de SW. Tales bloques deben establecerse en catálogos para una consulta más fácil, estandarizarse para una fácil aplicación y validarse para una fácil integración.
 - Se sugiere cuatro categorías de recursos de SW que se deberían tener en cuenta a medida que se avanza con la planificación:
 - ✓ **Componentes ya desarrollados.** El SW existente se puede adquirir de una tercera parte o provenir de uno desarrollado internamente para un proyecto anterior. Estos componentes están listos para utilizarse en el proyecto actual y se han validado totalmente.
 - ✓ **Componentes ya experimentados.** Las especificaciones, diseños, código o datos de prueba existentes y desarrollados para proyectos anteriores que son similares al SW que se va a construir para el proyecto actual. Los miembros del equipo de SW actual ya han tenido la experiencia completa en el área de la aplicación representada para estos componentes. Las modificaciones, por tanto, requeridas para componentes de total experiencia, tendrán un riesgo relativamente bajo.

- Planificación de proyectos – Recursos (Cont.)
- Recursos de SW reutilizable
 - ✓ **Componentes con experiencia parcial.** Las especificaciones, los diseños, código o los datos de prueba existentes desarrollados para proyectos anteriores que se relacionan con el software que se va a construir para el proyecto actual, pero que requerirán una modificación sustancial. Los miembros del equipo de SW actual han limitado su experiencia sólo al área de aplicación representada por estos componentes. Las modificaciones, por tanto, requeridas para componentes de experiencia parcial tendrán bastante grado de riesgo.
 - ✓ **Componentes nuevos.** Los componentes de software que el equipo de software debe construir específicamente para las necesidades del proyecto actual.
- De forma irónica, a menudo se descuida la utilización de componentes de SW reutilizables durante la planificación, llegando a convertirse en la preocupación primordial durante la fase de desarrollo del proceso de software. Es mucho mejor especificar al principio las necesidades de recursos del SW. De esta forma se puede dirigir la evaluación técnica de alternativas y puede tener lugar la adquisición oportuna.

- Planificación de proyectos – Recursos (Cont.)
- Recursos de SW reutilizable
 - Deberían considerarse las directrices siguientes por el planificador de SW cuando los componentes reutilizables se especifiquen como recurso:
 1. Si los componentes ya desarrollados cumplen los requisitos del proyecto, adquiéralos. El coste de la adquisición y de la integración de los componentes ya desarrollados serán casi siempre menor que el coste para desarrollar el SW equivalente. Además, el riesgo es relativamente bajo.
 2. Si se dispone de componentes ya experimentados, los riesgos asociados a la modificación y a la integración generalmente se aceptan. El plan del proyecto debería reflejar la utilización de estos componentes.
 3. Si se dispone de componentes de experiencia parcial para el proyecto actual, su uso se debe analizar con detalle. Si antes de que se integren adecuadamente los componentes con otros elementos del SW se requiere una gran modificación, proceda cuidadosamente. El coste de modificar los componentes de experiencia parcial algunas veces pueden ser mayores que el coste de desarrollar componentes nuevos.

- Planificación de proyectos – Recursos (Cont.)
- Recursos de Entorno
 - El entorno es donde se apoya el proyecto de SW, llamado a menudo entorno de ingeniería del software (EIS), incorpora HW y SW.
 - El hardware proporciona una plataforma con las herramientas (software) requeridas para producir los productos que son el resultado de una buena práctica de la ingeniería del software. Como la mayoría de las organizaciones de software tienen muchos aspectos que requieren acceso a EIS, un planificador de proyecto debe determinar la ventana temporal requerida para el hardware y el software, y verificar que estos recursos estarán disponibles.
 - Cuando se va a desarrollar un sistema basado en computadora (que incorpora hardware y software especializado), el equipo de software puede requerir acceso a los elementos en desarrollo por otros equipos de ingeniería. Por ejemplo, el software para un control numérico (CN) utilizado en una clase de máquina herramienta puede requerir una máquina herramienta específica (por ejemplo, el CN de un torno) como parte del paso de prueba de validación.
 - Un proyecto de software para el diseño de páginas avanzado puede necesitar un sistema de composición fotográfica o escritura digital en alguna fase durante el desarrollo. Cada elemento de hardware debe ser especificado por el planificador del proyecto de software.

- Planificación de proyectos – Estimación del proyecto de SW
 - Al principio, el costo del software constituía un pequeño porcentaje del costo total de los sistemas basados en computadora. Un error considerable en las estimaciones del costo del software tenía relativamente poco impacto. Hoy en día, el software es el elemento más caro de la mayoría de los sistemas informáticos. Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y pérdidas. Sobrepasarse en el costo puede ser desastroso para el equipo de desarrollo.
 - La estimación del costo y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas las variables -humanas, técnicas, de entorno, políticas- que pueden afectar al costo final del software y al esfuerzo aplicado para desarrollarlo. Sin embargo, la estimación del proyecto de software puede dejar de ser un arte oscuro para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.)
 - Para realizar estimaciones seguras de costes y esfuerzos tenemos varias opciones posibles:
 1. Dejar la estimación para más adelante (obviamente, ¿podemos realizar una estimación al 100% fiable tras haber terminado el proyecto!).
 2. Basar las estimaciones en proyectos similares ya terminados.
 3. Utilizar «técnicas de descomposición» relativamente sencillas para generar las estimaciones de costo y de esfuerzo del proyecto.
 4. Desarrollar un modelo empírico para el cálculo de costes y esfuerzos del software.
 - Desafortunadamente, la primera opción, aunque atractiva, no es práctica. Las estimaciones de costes han de ser proporcionadas a priori. Sin embargo, hay que reconocer que cuanto más tiempo esperemos, más cosas sabremos, y cuanto más sepamos, menor será la probabilidad de cometer serios errores en nuestras estimaciones.
 - La segunda opción puede funcionar razonablemente bien si el proyecto actual es bastante similar a los esfuerzos pasados y si otras influencias del proyecto (p. e.: el cliente, las condiciones de gestión, el EIS, las fechas límites) son similares. Desafortunadamente, la experiencia anterior no ha sido siempre un buen indicador de futuros resultados.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.)
 - Las opciones restantes son métodos viables para la estimación del proyecto de software. Desde un punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas, usando cada una de ellas como comprobación de las otras. Las técnicas de descomposición utilizan un enfoque de «divide y vencerás» para la estimación del proyecto de software. Mediante la descomposición del proyecto en sus funciones principales y en las tareas de ingeniería del software correspondiente, la estimación del costo y del esfuerzo puede realizarse de una forma escalonada idónea. Se pueden utilizar los modelos empíricos de estimación como complemento de las técnicas de descomposición, ofreciendo un enfoque de estimación potencialmente valioso por derecho propio. Cada modelo se basa en la experiencia (datos históricos) y toma como base:
$$d = f(v_i)$$
 - Donde d es uno de los valores estimados (por ejemplo, esfuerzo, coste, duración del proyecto) y los v_i son determinados parámetros independientes (por ejemplo, LDC o PF estimados).

- Planificación de proyectos – Estimación del proyecto de SW (Cont.)
 - Las herramientas automáticas de estimación implementan una o varias técnicas de descomposición o modelos empíricos. Cuando se combinan con una interfaz interactiva hombre-máquina, las herramientas automáticas son una opción atractiva para la estimación. En sistemas de este tipo, se describen las características de la organización de desarrollo (por ejemplo, la experiencia, el entorno) y el software a desarrollar. De estos datos se obtienen las estimaciones de costo y de esfuerzo.
 - Cada una de las opciones viables para la estimación de costos del software sólo será buena si los datos históricos que se utilizan como base de la estimación son buenos. Si no existen datos históricos, la estimación del coste descansará sobre una base muy inestable.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.)
- Técnicas de Descomposición
 - La estimación de proyectos de software es una forma de resolución de problemas y, en la mayoría de los casos, el problema a resolver (es decir, desarrollar estimaciones de costo y de esfuerzo para un proyecto de software) es demasiado complejo para considerarlo como un todo. Por esta razón, descomponemos el problema, volviéndolo a definir como un conjunto de pequeños problemas esperando que sean más manejables.
 - El enfoque de descomposición tiene dos puntos de vista diferentes:
 - ✓ Descomposición del problema.
 - ✓ Descomposición del proceso.
 - La estimación hace uso de una o ambas formas de particionamiento. Pero antes de hacer una estimación, el planificador del proyecto debe comprender el ámbito del software a construir y generar una estimación de su «tamaño».

- Planificación de proyectos – Estimación del proyecto de SW (Cont.)
- Tamaño del SW
 - La precisión de una estimación del proyecto de software se predice basándose en una serie de cosas:
 1. El grado en el que el planificador ha estimado adecuadamente el tamaño del producto a construir.
 2. La habilidad para traducir la estimación del tamaño en esfuerzo humano, tiempo y dinero (una función de la disponibilidad de métricas fiables de software de proyectos anteriores)
 3. El grado que el plan del proyecto refleja las habilidades del equipo de software.
 4. La estabilidad de los requisitos del software y el entorno que soporta el esfuerzo de la ingeniería del software.
 - Aquí consideramos el problema del tamaño del software. Como una estimación del proyecto es tan buena como la estimación del tamaño del trabajo que va a llevarse a cabo, el tamaño representa el primer reto importante del planificador de proyectos. En el contexto de la planificación de proyectos, el tamaño se refiere a una producción cuantificable del proyecto de software. Si se toma un enfoque directo, el tamaño se puede medir en **LDC**. Si se selecciona un enfoque indirecto, el tamaño se representa como **PF**.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Tamaño del SW
 - Se sugieren cuatro enfoques diferentes del problema del tamaño:
 - ✓ **Tamaño en lógica difusa:** Este enfoque utiliza las técnicas aproximadas de razonamiento que son la piedra angular de la lógica difusa. Para aplicar este enfoque, el planificador debe identificar el tipo de aplicación, establecer su magnitud en una escala cuantitativa y refinar la magnitud dentro del rango original. Aunque se puede utilizar la experiencia personal, el planificador también debería tener acceso a una base de datos histórica de proyectos para que las estimaciones se puedan comparar con la experiencia real.
 - ✓ **Tamaño en punto de función:** El planificador desarrolla estimaciones de características del dominio de información.
 - ✓ **Tamaño de componentes estándar:** El SW se compone de un número de componentes estándar que son genéricos para un área en particular de la aplicación. P. e. Los componentes estándar de un sistema de información son módulos de subsistemas, pantallas, informes, programas interactivos, programas por lotes, archivos, LDC, e instrucciones para objetos. El planificador de proyectos estima el número de incidencias de cada uno de los componentes estándar, y utiliza datos de proyectos históricos para determinar el tamaño de entrega por componente estándar.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Tamaño del SW
 - ✓ **Tamaño de componentes estándar (cont.):** Para ilustrarlo, considere una aplicación de sistemas de información. El planificador estima que se generarán 18 informes. Los datos históricos indican que por informe se requieren 967 líneas de Cobol. Esto permite que el planificador estime que se requieren 17.000 LDC para el componente de informes. Para otros componentes estándar se hacen estimaciones y cálculos similares, y se producen resultados combinados de valores de tamaños (ajustados estadísticamente).
 - ✓ **Tamaño del cambio:** Este enfoque se utiliza cuando un proyecto acompaña la utilización de SW existente que se debe modificar de alguna manera como parte del proyecto. El planificador estima el número y tipo de modificaciones que se deben llevar a cabo mediante una proporción de esfuerzo (p. e. reutilización, añadir, cambiar o suprimir código). Para cada tipo de cambio, se puede estimar el tamaño del cambio.
- Los resultados de cada uno de los métodos de tamaño señalados anteriormente se combinan estadísticamente para crear una **estimación de tres puntos** o del valor esperado. Esto se lleva a cabo desarrollando valores de tamaño optimistas (bajos), y más probables, y pesimistas (altos), y se combinan.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el problema
 - Las líneas de código (LDC) y los puntos de función (PF) se describen como medidas básicas desde donde se calculan métricas de productividad. Los datos de LDC y PF se utilizan de dos formas durante la estimación del proyecto de software: (1) como una variable de estimación que se utiliza para «dimensionar» cada elemento del software, y (2) como métrica de línea de base recopiladas de proyectos anteriores y utilizadas junto con variables de estimación para desarrollar proyecciones de costo y de esfuerzo.
 - Las estimaciones de LDC y PF son técnicas de estimación distintas, a pesar de que ambas tienen varias características en común. El planificador del proyecto comienza con un enfoque limitado para el ámbito del software y desde esta sentencia intenta descomponer el SW en funciones que se pueden estimar individualmente. Para cada función entonces se estiman las LDC y el PF (la variable de estimación). De forma alternativa, el planificador puede seleccionar otro componente (estimación con casos de uso) para dimensionar clases u objetos, cambios, o procesos de gestión en los que puede tener impacto.
 - Las métricas de productividad de línea base, p. e. LDC/pm o PF/pm (persona mes) se aplican entonces para la variable de estimación adecuada y se extrae el costo o el esfuerzo de la función. Las estimaciones de función se combinan para producir una estimación global del proyecto entero.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el problema
 - Es importante señalar, sin embargo, que hay una sustancial diversidad en métricas de productividad, haciendo sospechar que se utilice únicamente una métrica de productividad de línea base. En general, el dominio del proyecto debería calcular las medias de LDC/pm o PF/pm.
 - Es decir, los proyectos se deberían agrupar por tamaño de equipo, área de aplicación, complejidad y otros parámetros relevantes. Entonces se deberían calcular las medias del dominio local. Cuando se estima un proyecto nuevo, primero se debería asignar a un dominio, y a continuación utilizar la media del dominio adecuado para la productividad al generar la estimación.
 - Las técnicas de estimación de LDC y PF difieren en el nivel de detalle que se requiere para la descomposición y el objetivo de la partición. Cuando se utiliza LDC como variable de estimación, la descomposición es absolutamente esencial y a menudo se toman para considerables niveles de detalle. Cuando más grande sea el grado de particionamiento, más probable será que pueda desarrollar estimaciones más exactas.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el problema
 - Para estimaciones de PF, la descomposición funciona de diferente manera. En lugar de centrarse en la función, se estiman cada una de las características del dominio de información -entradas, salidas, archivos de datos, peticiones, e interfaces externas- y los catorce valores de ajuste de la complejidad. Las estimaciones resultantes se utilizan para derivar un valor de PF que se pueda unir a datos pasados y utilizar para generar una estimación.
 - Con independencia de la variable de estimación que se utilice, el planificador del proyecto comienza por estimar un rango de valores para cada función o valor del dominio de información. Mediante los datos históricos o (cuando todo lo demás falla) la intuición, el planificador estima un valor de tamaño optimista, más probable y pesimista para cada función, o cuenta con cada valor de dominio de función. Al especificar un rango de valores, se proporciona una indicación implícita del grado de incertidumbre.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el problema
 - Entonces se calcula un valor de tres puntos o esperado. El valor esperado de la variable de estimación (tamaño), VE, se puede calcular como una media ponderada de las estimaciones optimistas (s_{opt}), las más probables (s_m), y las pesimistas (s_{pess}). Por ejemplo:
$$VE = (s_{opt} + 4 s_m + s_{pess})/6$$
 - Da crédito a la estimación «más probable» y sigue una distribución de probabilidad beta.
 - Se asume que existe una probabilidad muy pequeña en donde el resultado del tamaño real quedará fuera de los valores pesimistas y optimistas. Mediante técnicas estadísticas estándar, se calcula la desviación de las estimaciones. Sin embargo, se debería señalar que una desviación basada en datos inciertos (estimados) se tiene que utilizar con precaución.
 - Una vez que se ha determinado el valor esperado de la variable de estimación, se aplican datos históricos de productividad de LDC y PF. ¿Son correctas las estimaciones? La única respuesta razonable a esto es: «No podemos estar seguros». Cualquier técnica de estimación, no importa lo sofisticado que sea, se debe volver a comprobar con otro enfoque. Incluso entonces, va a prevalecer el sentido común y la experiencia.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el proceso
 - La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar. Es decir, el proceso se descompone en un conjunto relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para llevar a cabo la estimación de cada tarea.
 - Al igual que las técnicas basadas en problemas, la estimación basada en el proceso comienza con una delineación de las funciones del software obtenidas a partir del ámbito del proyecto. Para cada función se debe llevar a cabo una serie de actividades del proceso de software. Las funciones y las actividades del proceso de software relacionadas se pueden representar como parte de una tabla.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Estimación basada en el proceso
 - Una vez que se mezclan las funciones del problema y las actividades del proceso, el planificador estima el esfuerzo (por ejemplo: persona-mes) que se requerirá para llevar a cabo cada una de las actividades del proceso de software en cada función. Los índices de trabajo medios (es decir, esfuerzo coste/unidad) se aplican entonces al esfuerzo estimado a cada actividad del proceso. Es muy probable que en cada tarea varíe el índice de trabajo. El personal veterano se involucra de lleno con las primeras actividades y generalmente es mucho más caro con el personal junior, quienes están relacionados con las tareas de diseño posteriores, con la codificación y las pruebas iniciales.
 - Como último paso se calculan los costes y el esfuerzo de cada función, y la actividad del proceso de SW. Si la estimación basada en el proceso se realiza independientemente de la estimación de LDC y PF, ahora tendremos dos o tres estimaciones del coste y del esfuerzo que se pueden comparar y evaluar.
 - Si ambos tipos de estimaciones muestran una concordancia razonable, hay una buena razón para creer que las estimaciones son fiables. Si, por otro lado, los resultados de estas técnicas de descomposición muestran poca concordancia, se debe realizar más investigación y análisis.

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - Modelos empíricos de estimación
 - Un modelo de estimación para el software de computadora utiliza fórmulas derivadas empíricamente para predecir el esfuerzo como una función de LDC y PF. Los datos empíricos que soportan la mayoría de los modelos de estimación se obtienen de una muestra limitada de proyectos. Por esta razón, el modelo de estimación no es adecuado para todas las clases de SW y en todos los entornos de desarrollo. Por consiguiente, los resultados obtenidos de dichos modelos se deben utilizar con prudencia.
- Estructura de los modelos estimación
 - Un modelo común de estimación se extrae utilizando el análisis de regresión sobre los datos recopilados de proyectos de SW anteriores. La estructura global de tales modelos adquiere la forma:
$$E = A + B \times (ev)^c$$
 - Donde A, B y C son constantes obtenidas empíricamente, E es el esfuerzo persona/mes, y ev es la variable de estimación (de LDC o PF). La mayoría de los modelos de estimación tienen algún componente de ajuste del proyecto que permite ajustar E por otra característica del proyecto (p. e. complejidad del problema, experiencia de personal, entorno de desarrollo).

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - El Modelo COCOMO
 - Barry Boehm introduce una jerarquía de modelos de estimación de software con el nombre de **COCOMO**, por **CO**nstructive **CO**st **MO**del (MOdelo COnstructivo de COste).
 - La jerarquía de modelos de Boehm está constituida por los siguientes:
 - ✓ **Modelo 1:** El modelo COCOMO **básico** calcula el esfuerzo (y el coste) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).
 - ✓ **Modelo 2:** El modelo COCOMO **intermedio** calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de «conductores de coste» que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
 - ✓ **Modelo 3:** El modelo COCOMO **avanzado** incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de coste en cada fase del proceso de ingeniería del software (análisis, diseño, etc.).

- Planificación de proyectos – Estimación del proyecto de SW (Cont.) - El Modelo COCOMO
 - Además, COCOMO hace una clasificación en tres tipos de proyecto, ya que la productividad no es la misma para todos los proyectos.
 - Los tipos de proyecto son:
 - ✓ **Orgánico:** Proyectos pequeños con pocos requerimientos, sin innovación tecnológica, usualmente un solo equipo de trabajo.
 - ✓ **Semiacoplado:** Proyectos de complejidad media, con algo de innovación tecnológica, donde interviene más de un equipo de trabajo.
 - ✓ **Encajado:** Proyectos de complejidad alta, con requerimientos rigurosos y con varios equipos trabajando coordinadamente.
 - Para que COCOMO funcione se supone que:
 - ✓ No se tienen en cuenta los comentarios en el recuento de las LDC.
 - ✓ Se admite la equivalencia siguiente: 1 persona/mes son 152 horas de trabajo.
 - ✓ Se considera que los requerimientos son estables.
 - ✓ Se acepta que el proyecto está bien gestionado.

- Bibliografía:
 - Ingeniería de software – Un enfoque práctico – Cuarta Edición de Roger S. Pressman - Editorial Mc Graw Hill
 - Joint Application Design – The Group Session Approach to System Design de Judy H. August
 - Análisis y Diseño de Sistemas – Tercera Edición de Kendall & Kendall – Editorial: Printice Hall