Metodologías de Análisis de Sistemas de Información

1. Ciclo de vida del sistema

Los sistemas, desde su concepción hasta su desaparición, pasan por una serie de etapas. Desde que surge la necesidad, pasando por la propia construcción, puesta en marcha y continuas revisiones hasta su abandono o reemplazo, el sistema pasa por una serie de fases que constituirán su ciclo de vida.

- INVESTIGACION PRELIMINAR
- DETERMINACION DE REQUERIMIENTOS
- DISEÑO DEL SISTEMA
- DESARROLLO DEL SOFTWARE
- PRUEBA DEL SISTEMA
- PUESTA EN MARCHA
- MANTENIMIENTO Y EVOLUCION
- FIN DEL SISTEMA

INVESTIGACION PRELIMINAR

Comienza cuando se plantea la necesidad de contar con un sistema de información.

Trata tres aspectos:

Aclaración de la solicitud
Estudio de pre–factibilidad Técnica, Económica y Operativa
Aprobación de la solicitud

DETERMINACION DE REQUERIMIENTOS

1) Permite conocer las funciones a automatizar
2) Identifica los problemas a resolver
3) Define la información a obtener del sistema

¿QUÉ ES UN REQUERIMIENTO?

Característica o función que debe incluirse en un sistema
Es necesario estudiar un sistema para conocer su funcionamiento y en qué puede ser mejorado.

¿CÓMO DETERMINARLOS?

1) ANTICIPACION DE REQUERIMIENTOS: Básicamente apoyada en la experiencia que permite anticipar problemas o características del sistema. Puede sesgar la investigación.

2) INVESTIGACION DE REQUERIMIENTOS: Se trata de una recopilación de hechos, se realiza un estudio del sistema actual (manual o informatizado) y se busca definir qué es lo que pretende la organización

3) ESPECIFICACION DE REQUERIMIENTOS: A partir del análisis de los datos obtenidos en el paso anterior y basado en hechos reales, se identifican los requerimientos esenciales y se genera una estrategia para satisfacerlos. Esta es la base para el diseño del futuro sistema.

ES MUY IMPORTANTE EN ESTA ETAPA, LA **PARTICIPACION ACTIVA DEL USUARIO**.

COMO SE CONCRETA LA INVESTIGACION DE REQUERIMIENTOS

1) POR MEDIO DE ENTREVISTAS
   - Permiten obtener información cualitativa
   - Generalmente se realizan con personas que conocen el sistema
   - Su éxito depende de a) habilidad del entrevistador b) preparación para la entrevista y c) participación de usuarios calificados (interlocutor válido)

2) POR MEDIO DE LA OBSERVACION DIRECTA
   - Esta técnica permite comprobar diferentes elementos obtenidos en las entrevistas

ENTREVISTA:

Antes de hacerla hay que definir:
- Qué preguntas se harán
- A quién entrevistar

Debe ser satisfactoria para ambos, entrevistado y entrevistador

Sirve para conocer:
- Opiniones y sentimientos del entrevistado acerca del estado actual de los sistemas y su expectativa sobre el futuro sistema
- Metas personales del entrevistado
- Metas de la organización

PLANEAMIENTO DE LA ENTREVISTA

- Obtener antecedentes, tanto de la organización como del entrevistado
- Fijar objetivos de la entrevista. Determinar aspectos fundamentales respecto al procesamiento de información y al proceso de toma de decisiones.
- Seleccionar a los entrevistados. Incluir personal clave de todos los niveles del sistema.
- Preparar el momento más adecuado para la entrevista, plantear temario con tiempo
- Seleccionar el tipo de pregunta, abierta o específica, puntual.

ENTREVISTA

- Es muy importante saber escuchar. Si no se obtienen requerimientos claros no será posible satisfacerlos en el diseño.
- Conocer la jerga de la actividad para evitar confusiones (En Empresas de Radio Taxis: Clave 50 = 0 ficha, X4 = Fuera de servicio) y no utilizar la jerga informática (Giga, Caché, Router, etcétera)
- Es normal que los requerimientos no se mencionen en forma directa, es posible que al entrevistado le resulte difícil trasmitir exactamente lo que quiere, tomar notas es fundamental.
- Es muy importante determinar las excepciones, muchas veces las excepciones son la regla y son la prueba de fuego del sistema.

DISEÑO DEL SISTEMA

Establece la forma en que se cumplirá con los requerimientos del sistema.

Debemos distinguir entre:

- Diseño Lógico: describe las entradas, salidas y procedimientos del sistema
- Diseño Físico: define archivos y bases de datos, pantallas de ingreso de datos o consulta, salidas impresas y especificaciones de los programas.

Esta etapa se va a ver en profundidad más adelante.

DESARROLLO DEL SOFTWARE

Esta es la etapa de la programación de los sistemas relevados.

- Desarrollos internos o externos, depende de las dimensiones de la organización
- Tendencia actual: tercerización, se da en empresas grandes o software house que contratan a otras más pequeñas para un desarrollo específico
- Es de vital importancia en esta etapa generar una buena documentación. Es esencial para el mantenimiento del sistema

- Una variante en esta etapa es la prototipación, generar un modelo que puede ser modificado fácilmente hasta obtener el modelo final (Ejemplo, GeneXus)
- Lenguajes a utilizar, dependen de la plataforma, el sistema operativo, la arquitectura sobre la cual correrá el sistema.

PRUEBA DEL SISTEMA

- Permite asegurar que el sistema funciona de acuerdo con las especificaciones
- Dos tipos de pruebas:
    - De escritorio: la realizan los programadores con datos de prueba
    - Real: se realiza conjuntamente con los usuarios. Luego la verdadera prueba de fuego se realiza en la práctica, con el uso real del sistema por parte de los usuarios.

IMPLANTACION Y PUESTA EN MARCHA

IMPLICA:

- Instalar equipamiento (hardware, cableado, comunicaciones) si corresponde
- Instalar Software (Sistema y eventualmente Software de Base de Datos)
- Generar los archivos de datos necesarios
- En esta etapa se realiza el período de carga de datos básicos: crear Clientes, Vendedores, Artículos de Stock, etcétera, es decir los Mantenimientos (Altas, Bajas y Modificaciones de Archivos o Tablas), Mantenimientos de datos Básicos para comenzar con el sistema.
- Capacitación del personal

Existen diferentes estrategias de puesta en marcha. En todos los casos, es sumamente importante la conversión de los archivos, de los datos, de un sistema a otro.

PARALELO

Consiste en que se instala el nuevo sistema y continúa funcionando el sistema anterior, por un período que se determina y varía según la empresa (ejemplo 90 días). Es el método más seguro, ya que se abandona el sistema anterior cuando el sistema nuevo está probado totalmente.

VENTAJAS: Si el sistema nuevo no cumple con los requerimientos, al seguir trabajando con el anterior, hay un tiempo prudencial para efectuar las correcciones necesarias. Se continúa trabajando normalmente y no hay riesgo de pérdida de información. La prueba del nuevo sistema se hace sin la exigencia de un plazo a término inamovible.

DESVENTAJAS: Se duplica el trabajo para los usuarios ya que trabajan en dos sistemas a la vez. Se corre el riesgo de que los usuarios, acostumbrados al sistema anterior, demoren excesivamente la prueba del nuevo, ya que no cuentan con la exigencia de un plazo a

término inamovible. El sistema actual lo conocen y dominan bien, el nuevo es un cambio y un desafío.

DIRECTO

Consiste en sustituir de un día para el otro el sistema anterior por el nuevo. Requiere que el nuevo sistema haya sido probado en forma muy exigente. No es un método aconsejable y se utiliza en situaciones límite, por ejemplo, el sistema anterior está dando serios problemas en su funcionamiento, hay una ruptura entre la empresa y el proveedor del software y se debe contratar una nueva, etcétera.

VENTAJAS: Obliga a que se utilice y pruebe en forma inmediata, se busquen los errores y éstos se corrijan en tiempo récord porque no hay un sistema respaldo. No hay resistencia de los usuarios porque no tienen otro remedio que utilizar y adaptarse al sistema nuevo, el cambio es impuesto.

DESVENTAJAS: Se corre el riesgo de que si el sistema nuevo no funciona adecuadamente, no se puede volver atrás. Otro riesgo importante es que si el nuevo sistema funciona mal, se pierde la integridad de los datos convertidos del sistema anterior al nuevo, ya que se estuvo trabajando varios días con el nuevo y los datos han sido modificados, se han actualizado y se perdió la situación de los datos al momento de la conversión.

PILOTO, PARCIAL O POR ETAPAS

Consiste en tomar un departamento o sección de la empresa, como modelo, y se prueba allí el sistema nuevo, mientras en el resto de la empresa se sigue utilizando el sistema anterior. Una vez aprobado, se generaliza a los demás sectores de la empresa.

VENTAJAS: Es un método seguro muy similar al paralelo. Da tiempo a las pruebas y corrección de errores, se pulen los defectos y no se pone en riesgo la integridad de los datos (ejemplo, una empresa con cinco locales en Shoppings, se implanta en uno, mientras los otros siguen con el sistema anterior)

DESVENTAJAS: ídem paralelo.

MANTENIMIENTO Y EVOLUCION

Los sistemas no permanecen fijos e inamovibles por mucho tiempo, quizás algunos paquetes con marco legal como Contabilidad o Sueldos, escapen a esta regla, pero los sistemas a medida o parcialmente a medida, que resuelven la gestión de una actividad empresarial en particular, necesitan modificaciones, adaptaciones, en el marco de la creciente dinámica y evolución de los negocios y las empresas.
Este tema en particular va a ser desarrollado en profundidad en la segunda parte del año, pero en esta etapa, como parte del ciclo de vida de un sistema de información, vamos a ver algunos ejemplos de la necesidad de mantener un sistema.

Por ejemplo:

- Correcciones por fallas o pulgas del sistema
- Correcciones de errores de relevamiento y análisis, no se cumple con determinados requerimientos
- Por modificaciones que surgen con el uso del sistema, conceptos de requerimientos no previstos en el relevamiento por el usuario
- Modificaciones por requerimientos derivados del crecimiento dinámico de la empresa (evolución)
- Modificaciones por requerimientos del avance tecnológico.
- Modificaciones por cambio en las disposiciones legislativas o normativas, ejemplo, cambios de aranceles o acuerdos en Importaciones
- Necesidad de anexar nuevos módulos al sistema. A veces, determinados módulos del proyecto se dejan para más adelante en el tiempo

GARANTIA: período en el cual se prueban y revisan los sistemas y se corrigen fallas derivadas del uso o se modifica o adapta la utilidad del sistema a la realidad de la empresa. Se acompaña a los usuarios en la prueba a fondo del sistema a efectos de eliminar rápidamente las dificultades y que el sistema responda realmente a las necesidades de la empresa, a entera satisfacción, dentro del período de garantía, por ejemplo fijado en un plazo de 90 días.

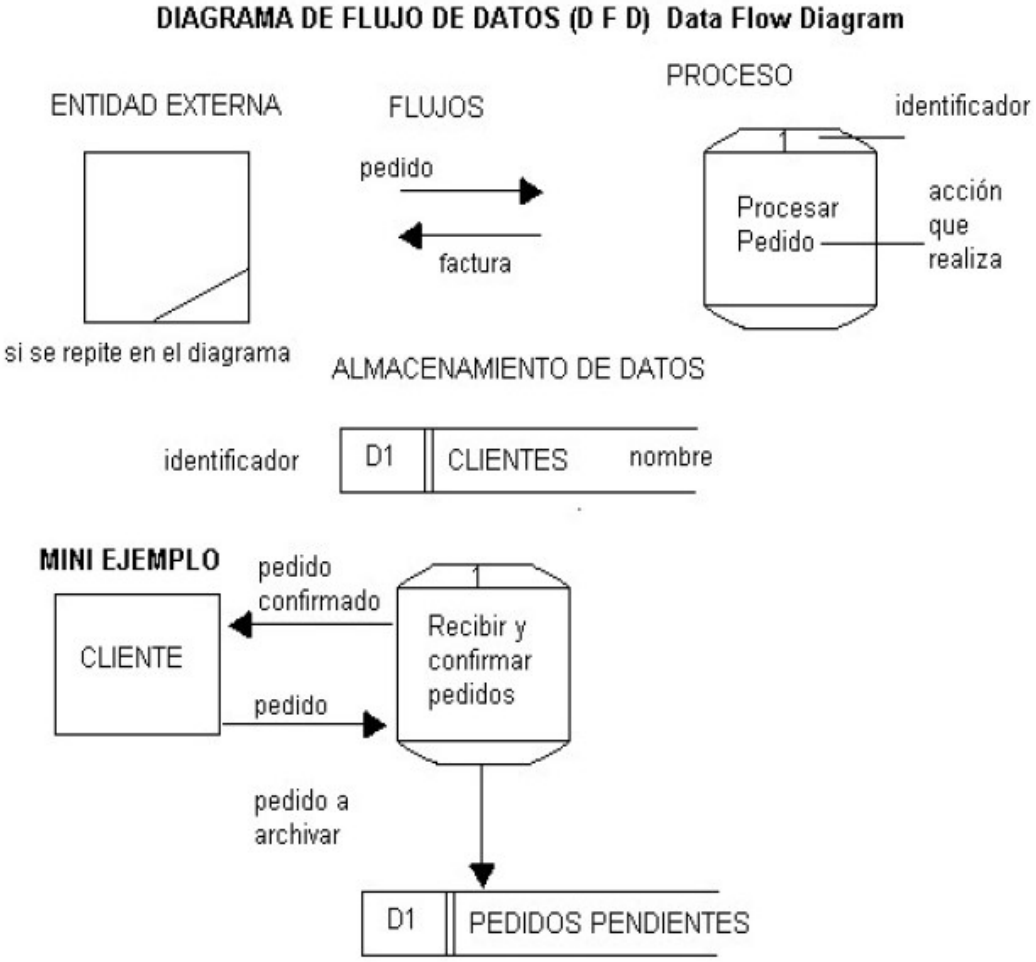
MANTENIMIENTO: tenemos dos tipos, Interno (propio de la empresa con centro de cómputos que ha desarrollado el sistema) y Externo, brindado por la empresa desarrollista contratada para desarrollarlo. En ambos casos, el mantenimiento tiene un costo, sea por concepto de sueldos del personal del centro de cómputos o por honorarios mensuales de la empresa desarrollista externa.

2. ANALISIS ESTRUCTURADO

Como dijimos en la creación de un sistema, de un software, tenemos las etapas de RELEVAMIENTO, ANALISIS, ESPECIFICACION, DESARROLLO E IMPLANTACION.
Los analistas en las etapas de **Relevamiento, Análisis y Especificación**, utilizan una técnica denominada **ANALISIS ESTRUCTURADO DE SISTEMAS**, que comprende un concepto también llamado TOP DOWN como sinónimo de Estructurado ***y que implica partir un problema grande en varios pequeños y resolverlos***. Esto se ve reflejado luego en la programación, es decir un sistema está compuesto de cientos de programas que resuelven determinados puntos y a su vez, se apoyan en otros programas o funciones para lograrlo. Los programas se "llaman" unos a otros en tiempo de ejecución, reciben un valor o varios y continúan ejecutándose.

EL ANALISIS ESTRUCTURADO

- Intenta superar las dificultades de comunicación entre usuarios y analistas
- Parte de la premisa de que si no se interpretan adecuadamente las necesidades de los usuarios seguramente el sistema fracase.
- Se centra en especificar qué es necesario que haga el sistema y no cómo se cumplirán los requerimientos.
- Utiliza un modelo gráfico que permite ver lo que el sistema debe hacer independientemente de las restricciones físicas, este modelo sirve para cualquier plataforma o sea es independiente del lenguaje, de la plataforma, de la estructura de red, etcétera.



3. PROTOTIPACION

Es otro enfoque para desarrollar sistemas. Consiste en crear un prototipo y en sucesivas aproximaciones llegar al sistema deseado. No se puede dividir en etapas perfectamente diferenciadas.
Un prototipo es un programa que contiene algunos o varios elementos del sistema deseado y que fundamentalmente consiste en la interfase con el usuario. Dicha interfase muestra cómo serán las pantallas de diálogo y el formato de las salidas por pantalla o impresas. En sucesivas aproximaciones y mediante la incorporación de funciones se llega al sistema final.

VENTAJAS: El usuario ve un programa funcionando mucho antes. Participa más activamente en el desarrollo. Se obtienen resultados en mucho menos tiempo.

DESVENTAJAS: No responde a un enfoque organizado. Es posible que el sistema resultante no tenga la flexibilidad necesaria y en poco tiempo puede ser desechado. No toma en cuenta todas las alternativas posibles al no haberse analizado, en profundidad, con detenimiento.

Si el lenguaje utilizado permite la prototipación (en el caso de GeneXus la prototipación es un paso fundamental e ineludible en el proceso de programación, el prototipo creado termina siendo el producto final cuando se pasa de la fase de Diseño a la de Producción), se obtienen todas las ventajas de la prototipación y ninguna de sus desventajas.
Si el lenguaje utilizado es un lenguaje tradicional, Cobol, Pascal, FoxPro, Visual Basic, etcétera, generar un prototipo no es tan fácil y obviamente se trata de armar una “fachada” del sistema final a efectos de mostrar al usuario.
Haciendo esta salvedad, podemos obtener llegar a las siguientes conclusiones:

> El uso de prototipos es recomendable en aquellos casos en que los desarrolladores tengan amplio conocimiento de la temática y experiencia en problemas similares. También es más eficiente en aquellos problemas muy puntuales y muy rígidos donde se prevé que no va a haber cambios y el sistema está claramente definido.
> También se usa para evaluar situaciones extraordinarias donde el encargado del diseño o implantación no tiene información o experiencia.
> En situaciones de riesgo y altos costos, en los que el diseño es novedoso o no se ha probado aún.
> Es una estrategia de desarrollo apropiada cuando no es fácil determinar los requerimientos del sistema de información.

CARACTERISTICAS DE LA PROTOTIPACION:

- Es una aplicación que funciona
- Se desarrolla con rapidez (mayor o menor de acuerdo al lenguaje)
- Tiene bajo costo, el análisis (o parte de él) se va realizando junto con el prototipo, no hay tiempo invertido en especificación, en diseño. Incluso GeneXus genera el modelo de datos.
- Evoluciona en un proceso iterativo: Se construye, se prueba, se modifica, surge un nuevo modelo y así sucesivamente

- Su finalidad es probar las suposiciones de los usuarios y analistas en cuanto a las características requeridas del sistema

ESTE PROTOTIPO PUEDE:

- Convertirse en el sistema deseado
- Servir de base para el desarrollo del sistema final (hacer las veces de esqueleto)
- Hacer que se abandone la idea de desarrollar el sistema

La prototipación puede coexistir perfectamente con el desarrollo tradicional de sistemas (Relevamiento, Análisis, Especificación, Desarrollo e Implantación). En una primera etapa se realiza el prototipo porque es difícil determinar los requerimientos o porque el usuario quiere ver cómo quedaría la aplicación o quiere ver si es factible su realización como sistema. Luego, utilizando el prototipo como base, se continúa con el desarrollo tradicional.

APLICACIONES COMERCIALES

¿Qué es hacer Software?

Dado un problema cuando se busca la solución, hay que hacer un análisis Costos – Beneficios.
¿Porqué hacemos Software? La solución computarizada es "mejor" (menos costos y más beneficios) que otras formas de resolver el problema.

¿Cómo se divide? En software "casero" y en software "producto"

Requisitos del software producto

- Interfase hombre – máquina
- Eficientes y precisos
- Ser más baratos que otras soluciones
- Tener buena documentación
- Portabilidad
- Robustez
- Legilibidad
- Mantenibilidad (estructurado)

El Software se compone de programas, un sistema es un conjunto de programas que interactúan entre sí con un objetivo común.

PRODUCCION DE SOFT

- Porque hay hardware cada vez más accesible y más usuarios interesados
- Encarada a solucionar problemas no resolubles de otra manera o que utilizando otras técnicas significarían costos mayores o beneficios menores.

¿QUE ES?

Productos que cumplen determinados requisitos.

¿CÓMO ENCARARLA?

Mediante un conjunto de metodologías de producción y control que garanticen que los productos elaborados cumplan determinados requisitos de calidad. A este conjunto de metodologías de producción y control le llamamos INGENIERIA DE SOFTWARE.
Se utilizan no sólo en la etapa de programación sino también en el análisis del problema, interacción con el usuario, documentación.

La llamada CRISIS DEL SOFT (principios y mediados de los 70's) ocurrió cuando la demanda de usuarios no especializados en el tema fue más alta, más gente tenía acceso al hardware pero no había software adecuado. Se intentó satisfacer la demanda con software empírico, no pudiendo además garantizar la correctitud, robustez, diseño, etcétera. No había normas ni metodología, por lo tanto no fue posible satisfacer la demanda con software de calidad. Esto llevó a la necesidad de encarar la producción de software como una actividad industrial auxiliada o guiada por ciertas técnicas de desarrollo.

En ese entorno nacen las ideas de la Programación Estructurada (Dijtra, Whirt), Técnicas de Diseño de Programas (Michael Jakcson, Jean Dominique Warnier), Diseño en Módulos (Lenguaje Modula, Tads). Todo este esfuerzo fue encarado hacia la programación. La parte de Análisis, demoró un poco más, aparecieron teorías para el Análisis Estructurado (Jourdon, De Marco, Constantine). Aún quedaba pendiente y por cierto tiempo se dejó de lado, la parte de interacción con el usuario, qué es lo que dice, lo que quiere, cómo entender cabalmente sus necesidades y requerimientos.
Recapitulando, podemos decir que frente a la Crisis del Soft, el problema fue la gran demanda de soft que no era posible cubrir con técnicas tradicionales ("artesanales") y la respuesta fue el mejoramiento de las técnicas de producción, apuntando a aumentar LA PRODUCTIVIDAD Y LA CALIDAD DE LOS PRODUCTOS, en áreas como Eficiencia, Mantenibilidad, Portabilidad, Robustez, Confiabilidad, etcétera.

AREAS CLAVES EN LA PRODUCCIÓN DE SOFTWARE

En las empresas de desarrollo de Software o Centro de Cómputos de grandes empresas (E.E.U.U.) se realizaron una serie de estudios sobre los costos de esfuerzos a aplicarse sobre los productos que desarrollaban. Los resultados fueron:

- Alrededor de un 50% del esfuerzo se destinaba al mantenimiento de productos ya elaborados.
- Un 30% de la producción a corregir pequeños errores que no habían sido detectados en la etapa de relevamiento o prueba.
- Un 20% se dedicaba a producir nuevos productos.

ESTO DEBERIA SER AL REVES, PRECISAMENTE. Estos valores, increíblemente, se mantienen en nuestros días en muchas empresas de desarrollo o centros de cómputos, obviamente, dependiendo de la herramienta de desarrollo que utilicen.

FACTORES CRITICOS

De estos valores se desprenden como factores críticos para el éxito de una empresa desarrollista, la mantenibilidad de los sistemas y la confiabilidad en el relevamiento, análisis y prueba del sistema. Otra área clave es la productividad, cómo mejorar la producción de nuevos sistemas.
La producción de software no es totalmente una producción EN SERIE (salvo los paquetes que tienen un marco legal), esto hace que el control de calidad no puede ser implementado eliminando los productos defectuosos (línea de producción, botella de Coca Cola).
La solución es utilizar técnicas que aseguren en el mayor grado posible que el producto elaborado es correcto. Tampoco es posible testear totalmente los productos para determinar los errores en la etapa de prueba de laboratorio.

REQUISITOS DE CALIDAD:

CONFIABILIDAD

- El producto debe ser preciso y eficiente
- Auto contención, que el producto tenga todo lo necesario para funcionar, que no dependa de otros productos (ejemplo Punta Carretas Shopping)
- Robustez, que no se caiga
- Consistente, que responda a las especificaciones del problema, o sea que haga lo que debe hacer.

MANTENIBILIDAD

- Entendible, bien documentado
- Modificable fácilmente, las estructuras deben ser tales que sea fácil de modificar ante nuevos requerimientos, que esté bien modularizado es fundamental.

PRODUCTIVIDAD

- Está relacionada con el tiempo de trabajo insumido y el rendimiento
- Las técnicas para aumentar la productividad van a estar relacionadas con técnicas de trabajo, sean éstas en equipo o individualmente.

TRABAJO EN EQUIPO

Elementos que inciden en la productividad de un equipo:

- Planificación del trabajo
- División y asignación de tareas, evaluación de las condiciones de los integrantes
- La supervisión y el control

- La forma de documentación del trabajo que se está desarrollando
- Relación con el usuario

Teniendo en cuenta todos estos elementos surgieron distintas técnicas de trabajo en equipo
Estructuras Jerárquicas:

- Ingenieros
- Analistas
- Programadores
- Testeadores
- Documentadores
- Instaladores
- Capacitadores
- Los que preparan el mate

Estructuras Democráticas:

- Equipo de Especialistas (formado por especialistas en cada área, análisis, programación, documentación, etcétera).
- Equipo Rotativo, el interés es rotar a las personas en las distintas áreas, por lo general grupos en formación, integrados por gente nueva.

TIPOS DE ORGANIZACIÓN PARA TRABAJO EN EQUIPO

TRADICIONAL (JERARQUICA)

Ventajas:

- Poco costo de coordinación
- Refleja la estructura de resolución del problema (Top Down)

Desventajas:

- Tiende a ascender a sus miembros a tareas que no tienen porqué ejecutar eficientemente
- No tiende a especializar o centrar en un equipo de especialistas las tareas de documentación y testeo (no siempre existen estos roles en esta estructura)
- La capacidad de captación del objetivo del proyecto y su trasmisión al resto del equipo depende sólo del cabeza de equipo.

GRUPO DE ESPECIALISTAS

Ventajas:

- Se valora la capacidad de cada uno de los miembros del equipo para desarrollar tareas particulares, no tendiéndose a “ascenderlos” a tareas que no ejecutan eficientemente.
- Especializa las tareas de documentación y testeo

Desventajas:

- Mayor costo de coordinación (salvo que haya un coordinador general)
- Mayor costo del mantenimiento del equipo (U$S)
- Se corre peligro de pérdida de integridad conceptual, para evitarlo se debe poner mayor esfuerzo en las tareas de coordinación.

GRUPO DE ESPECIALISTAS ROTATIVOS

Ventajas:

- Útil en la etapa de formación del equipo

Desventajas:

- No se aplica rigurosamente el concepto de especialización

PROGRAMADOR JEFE o SUPER PROGRAMADOR

Un solo responsable que hace TODO con un grupo de especialistas que lo ayudan a su requerimiento.

Ventajas:

- Gran integridad conceptual
- Seguridad: nadie conoce más de lo que necesita conocer para cumplir con su tarea

Desventajas:

- El éxito o el fracaso dependen en su mayor parte de una sola persona
- No se aplican los conceptos de especialización de tareas
- No es aplicable a proyectos largos