

# **INGENIERIA DE SOFTWARE I**

## **¿Qué es el software?**

El software no son solo los programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.

El software se desarrolla, no se fabrica como otros productos, por consiguiente trae un mayor costo en la ingeniería que en la producción. El software no se desgasta, no sigue una curva clásica de envejecimiento. Es inmune a los males que desgastan al hardware, debido a que el problema no está en el tiempo de operación, sino en los cambios.

## **¿Qué es la ingeniería de software?**

La ingeniería de software es una disciplina de ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. La misma se encarga de asegurar la producción de productos correctos, utilizables, de alta calidad y costo-efectivos.

La noción de ingeniería de software fue propuesta inicialmente en 1968 en una conferencia para discutir lo que en ese entonces se llamó la “crisis del software”. Esta crisis del software fue el resultado de la introducción de las nuevas computadoras hardware basadas en circuitos integrados. Su poder hizo que las aplicaciones hasta ese entonces irrealizables fueran una propuesta factible. El software resultante fue de ordenes de magnitud mas grande y mas complejo que los sistemas de software previos.

### **Evolucion:**

En las décadas del 50-60 el hardware era de propósito general, con continuos cambios. El software se construía “a medida”, por lo cual el desarrollo era sin planificación.

*DOCUMENTACION INEXISTENTE*. En los años 60, con la sofisticación creciente de los sistemas de software aparecen los conceptos de Multiprogramación Multiusuario Minicomputadoras, los sistemas en tiempo real y las bases de datos. Surge una diversidad de lenguajes de programación. Nace el producto software.

*"MANTENIMIENTO DE SOFT"*. Crecieron también las dificultades para desarrollarlos o adaptarlos. Las dificultades desbordaban los recursos técnicos de una heterogénea clase profesional. “CRISIS DEL SOFTWARE”. Aquí es cuando la expresión “Ingeniería de software” se utilizó por primera vez, en la conferencia de la OTAN. (1970-90) Complejidad. Microprocesadores. Redes. Procesos distribuidos. PC. Software de aplicación en todas las áreas. Baja espectacular del costo del hardware.

### ***PRODUCTIVIDAD***

(1990-2000) Tecnologías O.O. Procesamiento paralelo. Recurso de procesamiento ilimitado. Capacidad de aprendizaje del usuario es un límite. *"INGENIERÍA"*.

(2000-act) Aplicaciones WEB. Sistemas Expertos. Reconocimiento de Patrones. Redes Neuronales. Computación Ubicua. Código abierto. *COMUNICACION Y DISTRIBUCION*

La ingeniería de software surgió como reacción a las dificultades de desarrollar software sobre la base de habilidad, experiencia o intuición individuales. La aplicación de “*métodos sistemáticos*” y repetibles permite la producción y evolución de software por organizaciones distribuidas geográficamente y en el tiempo. La aplicación de estándares facilita la integración, reusabilidad y mantenimiento de los productos.

## Proceso de software

Es el conjunto de metodos, tecnicas y practicas que guian a los ingenieros de software en el desarrollo y evolucion del software.

El proceso de software es tambien llamado ciclo de vida del software, porque describe la vida de un producto de software desde su concepción hasta su implementacion, entrega, utilización y mantenimiento.

Un proceso de desarrollo de software debe describirse de manera flexible que permita a los que diseñan y construyen el software utilizar las herramientas y tecnicas preferidas; el proceso en si ayuda a mantener la consistencia y calidad en los productos que son producidos por muchas personas diferentes. El conjunto de procedimientos debe estar organizado de tal modo que los productos se construyan para satisfacer un conjunto de metas o estandares.

Consta de cuatro pasos genericos para llevarse a cabo:

- 1) Especificación del software: donde los ingenieros de sistema hablan con los usuarios finales, clientes (stakeholders) y toda persona relacionada con el sistema para ver cual es el proposito del mismo, las metas que se proponen y las restricciones.
- 2) Desarrollo del software: en esta etapa los ingenieros diseñan y programan el sistema.
- 3) Validación del software: aquí el sistema se valida para asegurar que se hizo lo que el cliente requeria.
- 4) Evolucion del software: donde el software se va modificando para adaptarse a los cambios requeridos por el cliente y los cambios del mercado.

## Modelo de proceso de software

Es una representación abstracta de un proceso de software. Existen varios modelos de proceso de software y cada uno representa una descripción, desde una perspectiva particular, de la manera en que el desarrollo del software se hace en la realidad. Los modelos de proceso tambien nos prescriben la manera en que se debe avanzar el desarrollo del software. Existen 3 tipos genéricos:

- 1) Modelo en cascada: aquí se consideran las actividades fundamentales del proceso, tales como la especificación, el desarrollo, la validación y la evolucion y se las representa como fases separadas. No se puede avanzar de etapa hasta no haber terminado la anterior, se documenta cada fase.
- 2) Desarrollo evolutivo: se basa en la idea de desarrollar una implementacion inicial, exponiendola a los comentarios del usuario y refinandola a traves de las diferentes versiones hasta que se desarrolla un sistema adecuado. Las actividades de especificación, desarrollo y validación se entrelazan, en vez de separarse, con una retroalimentación entre ellas.

Existen dos tipos de desarrollo evolutivo:

- *Desarrollo exploratorio*: el objetivo es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final; se empieza con las partes del sistema que se comprenden mejor,
  - *Prototipos desechables*: el objetivo es comprender los requerimientos del cliente y desarrollar una definición mejorada de los requerimientos para el sistema; se centra en experimentar con los requerimientos del cliente que no se comprenden del todo.
- 3) Ingenieria de software basada en componentes: este enfoque se basa en la existencia de un numero significativo de componentes reutilizables. El

proceso de desarrollo del sistema se basa en integrar estos componentes en el sistema más que en desarrollarlos desde cero.

Para sistemas grandes, se recomienda un proceso mixto que incorpore las mejores características del modelo en cascada y del desarrollo evolutivo. Esto puede implicar desarrollar un prototipo desechable utilizando un enfoque evolutivo para resolver incertidumbres en la especificación del sistema. Puede entonces re-implementarse utilizando un enfoque más estructurado. Las partes del sistema bien comprendidas se pueden especificar y desarrollar utilizando un proceso basado en el modelo en cascada. Las otras partes del sistema, como la interfaz del usuario, que son difíciles de especificar por adelantado, se deben desarrollar siempre utilizando un enfoque de programación exploratoria.

## **Requerimientos**

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina INGENIERIA DE REQUERIMIENTOS.

En algunos casos, un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo, es una definición detallada y formal de una función del sistema  
Requerimientos del usuario: son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

Deben describir los requerimientos funcionales y no funcionales de tal forma que sean comprensibles por los usuarios del sistema sin conocimiento técnico detallado.

Requerimientos del sistema: establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema debe ser preciso. Debe definir exactamente que es lo que se va a implementar.

Son versiones extendidas de los requerimientos del usuario, agregan detalle y explican como el sistema debe proporcionar los requerimientos del usuario.

Diferentes niveles de especificación del sistema son de utilidad debido a que comunican la información del sistema a diferentes tipos de lectores, ya que estos utilizan los requerimientos de distinta manera.

Requerimientos funcionales: son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

Pueden ser requerimientos funcionales de usuario o requerimientos funcionales de sistema (cuando se expresan como requerimientos del usuario, habitualmente se describen de una forma bastante abstracta. Sin embargo, los requerimientos funcionales del sistema describen con detalle la función de este, sus entradas y salidas, excepciones, etc.)

Requerimientos no funcionales: son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema

Se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Requerimientos del dominio: son requerimientos que provienen del dominio de aplicación del sistema y que reflejan las características y restricciones de ese dominio. Pueden ser funcionales y no funcionales.

Los requerimientos del dominio derivan del dominio de aplicación del sistema mas que de las necesidades específicas de los usuarios. Normalmente incluyen terminología especializada del dominio o referencias a conceptos del dominio, por lo que a menudo es difícil de comprender por los ingenieros de software. Los expertos del dominio pueden dejar afuera de un requerimiento información sencillamente porque para ellos es obvia, pero puede no serlo para los desarrolladores del sistema y es posible que implementen el requerimiento de forma equivocada.

La evolución de los requerimientos durante el proceso de ingeniería de requerimientos y después de que un sistema este en uso es inevitable. Desde una perspectiva evolutiva los requerimientos se dividen en dos clases:

Requerimientos duraderos: son requerimientos estables que se derivan de la actividad principal de la organización y que están relacionados directamente con el dominio del sistema.(ej en un hospital siempre habrán requerimientos que se refieren a pacientes, médicos, enfermeras y tratamientos)

Requerimientos volátiles: son requerimientos que probablemente cambian durante el proceso de desarrollo del sistema o después de que este se haya puesto en funcionamiento(ej serían requerimientos resultantes de las políticas gubernamentales sobre sanidad(hospital)).

Características o propiedades de los requerimientos:

- Necesario: Su omisión provoca una deficiencia.
- Conciso: Fácil de leer y entender
- Completo: No necesita ampliarse
- Consistente: No contradictorio con otro
- No ambiguo: Tiene una sola implementación
- Verificable: Puede testearse a través de inspecciones, pruebas, etc.

### **Ingeniería de requerimientos**

Ingeniería de requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones.

Ingeniería de requerimientos es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos.

La meta del proceso de ingeniería de requerimientos es crear y mantener un documento de requerimientos del sistema. El proceso general corresponde a cuatro subprocesos de alto nivel de la ingeniería de requerimientos, estos son:

- 1) Estudio de viabilidad
- 2) Obtención y análisis de requerimientos

- 3) Especificación de requerimientos
- 4) Validación de requerimientos
- 5) Gestion de requerimientos

Estudio de viabilidad: para todos los sistemas nuevos, el proceso de ingeniería de requerimientos debería empezar con un estudio de viabilidad. La entrada de este es un conjunto de requerimientos de negocio preliminares, una descripción resumida del sistema y de cómo este pretende contribuir a los procesos de negocio. Los resultados del estudio de viabilidad deberían ser un informe que recomiende si merece o no la pena seguir con la ingeniería de requerimientos y el proceso de desarrollo del sistema. Un estudio de viabilidad es un estudio corto y debería resolver las siguientes preguntas:

- El sistema contribuye a los objetivos generales de la organización?
- El sistema se puede implementar con la tecnología actual ?
- El sistema se puede implementar con las restricciones de costo y tiempo?
- El sistema puede integrarse a otros que existen en la organización?

Obtención y análisis de requerimientos: La extracción de los requerimientos es una parte especialmente crítica del proceso, debemos trabajar con los usuarios y los clientes (stakeholders), para comprender un problema cuando todavía no se ha encontrado una solución. Una de las maneras de realizar un análisis de problema consiste en identificar las personas, los procesos y los recursos involucrados, y después documentar las relaciones que existen entre ellos. Se averigua que elementos de datos pasan de un rol a otro, y cuales procesos transforman los datos de una forma o estado a otra. Durante la extracción de los requerimientos se interroga a usuarios y clientes sobre los mismos aspectos. Por lo general resulta útil separar los requerimientos en tres categorías:

- 1) Requerimientos que deben ser absolutamente satisfechos.
- 2) Requerimientos que son muy deseables pero no indispensables.
- 3) Requerimientos que son posibles, pero que podrían eliminarse.

El proceso generico de obtención y análisis de requerimientos consta de 4 pasos:

- *Descubrimiento de requerimientos:* aquí se utilizan técnicas como entrevistas, cuestionarios, JADs y brainstorm.
- *Clasificación y organización de requerimientos:* toma la recopilación no estructurada de requerimientos, y los organiza en grupos relacionados y coherentes.
- *Ordenación por prioridades y negociación de requerimientos:* inevitablemente, cuando existen muchos stakeholders, los requerimientos entraran en conflicto. Esta actividad se refiere a ordenar según las prioridades los requerimientos, y a encontrar y resolver los requerimientos en conflicto a través de la organización.
- *Documentación de requerimientos:* se documentan los requerimientos, se pueden producir documentos de requerimientos formales o informales.

Especificación de requerimientos: hay dos clases de documentos de requerimientos que se realizan en la extracción y el análisis de requerimientos. Por una parte, la extracción nos permite escribir un documento de **definición de requerimientos**, escrita en terminos que el cliente puede entender, la definición de requerimientos es un listado completo de todas las cosas que el cliente espera que haga el sistema propuesto.

Representa una comprensión entre el cliente y el desarrollador de lo que el cliente necesita o desea, y por lo general es escrito en forma conjunta por el cliente y el desarrollador. Por la otra parte, la **especificación de requerimientos** reitera la definición en los terminos tecnicos apropiados para el desarrollo del diseño de un sistema; es la contrapartida tecnica al documento de definición de requerimientos y es escrito por analistas de requerimientos.

Objetivos de la especificación de requerimientos:

- Permiten que los desarrolladores expliquen como han entendido lo que el cliente pretende del sistema
- Indican a los diseñadores que funcionalidad y características va a tener el sistema resultante
- Indican al equipo de pruebas que demostraciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es lo que había ordenado.

*Descripción estática de requerimientos:* Se describe el sistema a través de las entidades u objetos, sus atributos y sus relaciones con otros. No describe como las relaciones cambian con el tiempo y permiten razonar sobre comportamiento de sistemas que cambian poco (o nada) en el tiempo

*Descripciones dinámicas:* se considera un sistema en función de los cambios que ocurren a lo largo del tiempo. El sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado.

Las tecnicas para describir un sistema en terminos de estados son:

- 1) Tablas de decisión
- 2) Redes de Petri
- 3) Diagrama de transición de estados
- 4) Tablas de transición de estados
- 5) ...

Validación de requerimientos: trata de mostrar que los requerimientos realmente definen el sistema que el cliente desea. La validación de requerimientos es importante debido a que los errores en los requerimientos pueden conducir a importantes costos si se descubren mas tarde.

Durante el proceso de validación de requerimientos, se deben llevar a cabo verificaciones sobre requerimientos en el documento de requerimientos:

- 1) *Verificaciones de validez*
- 2) *Verificaciones de consistencia*
- 3) *Verificaciones de completitud*
- 4) *Verificaciones de realismo*
- 5) *Verificabilidad*

Se pueden utilizar, en conjunto o de forma individual, varias tecnicas de validación de requerimientos:

- *Revisiones de requerimientos*
- *Construcción de prototipos*

- *Generacion de casos de prueba*

Gestion de requerimientos: los requerimientos para sistemas software grande son siempre cambiantes. Pero porque cambian los requerimientos? :

- Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas (En sistemas grandes hay una comunidad diversa de usuarios)
- Porque los clientes y los usuarios son distintos
- Porque cambió el problema que se estaba resolviendo
- Porque los usuarios cambiaron su forma de pensar o sus percepciones
- Porque cambió el ambiente de negocios (mercado, etc.)

Existe entonces un proceso de gestion de requerimientos.

1) Identificación de requerimientos

2) Gestión del cambio

- Análisis del problema y especificación del cambio
- Análisis del cambio y cálculo de costos
- Implementación del cambio

3) Políticas de rastreo

- Fuente
- Requerimientos
- Diseño

4) Ayuda de herramientas CASE

- Almacenar requerimientos
- Gestionar el cambio
- Gestionar el rastreo

Análisis estructurado: DFD

Análisis estructurado tiempo real: DFC

(final marzo 09)

Describe las ampliaciones de análisis estructurado para modelar sistemas de tiempo real.

**Ampliaciones al análisis estructurado**

**SISTEMAS DE TIEMPO REAL:**

Características:

- Responden al mundo real
- En un tiempo prefijado

- Deben ser fiables, reinicializables y recuperables a fallas.

Ejemplos: Control de procesos, investigación médica, comunicaciones, etc.

==> AMPLIAR EL ANALISIS ESTRUCTURADO

- Se utilizan para modelizar aplicaciones conducidas por sucesos que producen información de control y/o que procesan información con limitaciones de tiempo.
- Existen variantes: Ward y Mello, Hatley y Pirbhai....

DFC: Muchas aplicaciones de software son dependientes del tiempo y procesan más información orientada al control que a los datos, por ejemplo el control de naves, procesos de fabricación, etc.

Las primeras ampliaciones que se hacen a este método están efectuadas por Ward y Mellor, y posteriormente lo hacen Hatley y Pirbhai y dicen que un sistema de tiempo real debe interactuar con el mundo real en marcos temporales que vienen dados por este mundo real.

Estas ampliaciones permiten reflejar el flujo de control y el procesamiento de control, así como el procesamiento y el flujo de datos.

NOTACION: (Ward y Mellor)

- ✓ Los PROCESOS DE CONTROL representan con burbujas punteadas las funciones que coordinan o sincronizan.
- ✓ Los FLUJOS DE CONTROL representan con líneas punteadas señales o interrupciones
- ✓ Los FLUJOS DE DATOS CONTINUOS representan los datos que ingresan de manera “continua”.
- ✓ Los ALMACENAMIENTOS DE CONTROL representan mediante líneas dobles punteadas los datos de control permanentes del sistema

### **Descripción de 3 o mas problemas q surgen cuando los requisitos deben obtenerse de 3 o mas clientes diferentes.**

Obtener y comprender los requerimientos de estos stakeholders, trae problemas como:

1. Debido a que los diferentes stakeholders tienen requerimientos distintos, que pueden expresar de varias

formas. Los ingenieros de requerimientos tienen que considerar todas las fuentes potenciales

de requerimientos y descubrir las concordancias y los conflictos que pueden surgir.

2. Inevitablemente. los stakeholders tienen opiniones diferentes sobre la importancia y prioridad

de los requerimientos, y algunas veces estas opiniones están reñidas. Durante el proceso.

se deberían organizar frecuentes negociaciones con los stakeholders para que se pueda llegar

a acuerdos.

3. Es imposible satisfacer completamente a todos los stakeholders. pero si algún



stakeholder piensa que sus opiniones no se han considerado adecuadamente.  
deliberadamente  
puede intentar socavar el proceso de ingeniería de requerimientos.

### **DESCRIPCIONES DINAMICAS**

**Tablas de decisión** es una herramienta que permite presentar el sistema en función de posibles condiciones en un tiempo dado, reglas para reaccionar ante los estímulos que ocurren cuando se reúnen determinados conjuntos de condiciones y las acciones a ser tomadas como un resultado.

**Diagrama de transición de estados** un estado identifica un periodo de tiempo de un objeto/entidad en el cual está esperando alguna operación. Una transición es el paso de un estado a otro. Se describe como el sistema responde a eventos internos o externos, muestra los estados del mismo y los eventos que provocan las transiciones de un estado a otro. No muestra el flujo de datos dentro de sistema. Un DTE supone que, en cualquier momento, el sistema está en uno de varios estados posibles y cuando recibe un estímulo dispara una transición a un estado diferente.

**Casos de uso** describe un sistema en términos de sus distintas formas de utilización, cada una de las cuales se conoce como un Caso de Uso. Los Diagramas de CU describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario. Los Diagramas de Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación

**Redes de Petri** esta técnica es utilizada para sistemas de tiempo real donde existen tareas concurrentes, es decir, cuando pueden ocurrir varios eventos en paralelo pero son ejecutados en un orden impredecible. Existen sitios y transiciones.

**DFD Analisis estructurado** modelización del flujo y contenido de la información mediante esta técnica. Se refleja el flujo de datos dentro del sistema. Existen entidades, almacenes. El modelo esencial está compuesto por el modelo ambiental (declaración de propósito, diagrama de contexto, lista de acontecimientos) y el modelo de comportamiento (DFD, diccionario de datos).

**DFC Ampliación del análisis estructurado** muchas aplicaciones de software son dependientes del tiempo y procesan más información orientada al control que a los datos, por ejemplo el control de naves, procesos de fabricación, etc. Las primeras ampliaciones que se hacen a este método dicen que un sistema de tiempo real debe interactuar con el mundo real en marcos temporales que vienen dados por este mundo real. Estas ampliaciones permiten reflejar el flujo de control y el procesamiento de control, así como el procesamiento y el flujo de datos.

## TECNICAS DE ELICITACION DE REQUERIMIENTOS

**Entrevistas** es una conversación con un propósito específico, que se basa en un formato de preguntas y respuestas en general, predefinidas. Su objetivo es conocer tanto las opiniones como los sentimientos del entrevistado acerca del sistema.

Existen las preguntas abiertas y las preguntas cerradas:

*abiertas*: (no estructuradas) se sugieren comenzar la entrevista con preguntas que no dependan del contexto, para conocer en forma básica el problema, la gente que requiere la solución, el tipo de solución e incluso para evaluar la efectividad de la entrevista misma

Ventajas

- Revela nuevas líneas de preguntas
- Hacen más interesante la entrevista
- Facilitan la expresión del entrevistado
- Permiten espontaneidad

Desventajas

- Pueden dar muchos detalles irrelevantes
- Se puede perder el control de la entrevista
- Parece que el entrevistador no tiene un objetivo claro

*cerradas*: (estructuradas) se dirige al usuario sobre los requerimientos del software. Nos permiten adquirir conocimiento mas profundo del dominio

Ventajas

- Ahorran tiempo
- Se mantiene el control de la entrevista
- Se consiguen datos relevantes

Desventajas

- Pueden aburrir al interesado
- No se obtienen detalles

**Cuestionarios:** nos ayudan a detectar un sentimiento generalizado, a detectar problemas entre usuarios y a cuantificar respuestas. El tipo de información obtenida es la actitud, las creencias y el comportamiento de las personas. Esta tecnica es utilizada cuando las personas estan dispersas geográficamente o cuando son un numero significativo.

**JAD:** se ayuda a los usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse participantes.

Se basa en cuatro principios:

- Dinámica de grupos.
- Uso de ayudas visuales para mejorar la comunicación.
- Mantener un proceso organizado y racional.
- Durante la entrevista se trabajará sobre lo que se generará. (Interfaces, Reportes, etc.)

Esta tecnica es utilizada cuando los grupos de usuarios están intranquilos y quieren algo nuevo o cuando se prevé que la cantidad de ideas sera mayor si se realiza un ejercicio de grupo.

- Ventajas

- Ahorro de tiempo
- Usuarios involucrados
- Desarrollos creativos

- Desventajas

- Es difícil organizar los horarios de los involucrados
- Es complejo encontrar un grupo de participantes integrados y organizados.

El JAD cuenta de 4 fases: la adaptación, la celebración de las reuniones y las conclusiones.

**Lluvia de ideas (BrainStorm):** aquí se promueve el desarrollo de ideas creativas para obtener soluciones. Se realizan reuniones del equipo involucrado en la resolución del problema, conducidas por un director. Es clave para resolver la falta de consenso entre usuarios, útil combinarlo con la toma de decisiones, ayuda a entender el dominio del problema, ayuda a entender: al usuario y al analista.

**JAD vs BrainStorm:** El brainstorming tiene la ventaja de que es muy fácil de aprender y requiere poca organización, pero puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del sistema, por lo que suele aplicarse en los primeros encuentros.