

# INGENIERÍA DE SOFTWARE AVANZADA MIS

## (Sesión 9)

### 4. CALIDAD DEL SOFTWARE

#### 4.1 Definición de calidad

#### 4.2 Características del software

**Objetivo: Comprender el concepto de calidad aplicado al diseño de software**

#### **La calidad y la Ingeniería de software**

El concepto de calidad en los productos de software debe formularse de forma particular. Primero es conveniente indicar sus características diferenciadoras frente a otros productos: el software se desarrolla, no se fabrica en el sentido clásico; es inmaterial y no se deteriora con el uso o el tiempo (aunque tiene un ciclo de vida); su fiabilidad es difícil de comprobar; la mayoría del software se construye a medida y necesita de actualización permanente; es dependiente del entorno donde se ejecuta.

El concepto de calidad en los productos de software debe formularse de forma particular. Primero es conveniente indicar sus características diferenciadoras frente a otros productos: el software se desarrolla, no se fabrica en el sentido clásico; es inmaterial y no se deteriora con el uso o el tiempo (aunque tiene un ciclo de vida); su fiabilidad es difícil de comprobar; la mayoría del software se construye a medida y necesita de actualización permanente; es dependiente del entorno donde se ejecuta.

La ingeniería de software es una disciplina cuyo horizonte de madurez está aun lejos, y que se caracteriza por la proliferación de normas, métodos y herramientas incompatibles entre si.

La calidad en la ingeniería del software, que depende en gran medida de la pericia del equipo que lo desarrolla, puede definirse como un conjunto de características o cualidades, tales como: eficiencia, fiabilidad, usabilidad, funcionalidad, "mantenibilidad", portabilidad, etc., variando la importancia de cada una de ellas de un producto a otro.

Dicho de otra manera, es el cumplimiento de los requerimientos contractuales por parte

del producto software desarrollado, así como durante el proceso de desarrollo.

En las empresas de Ingeniería de software, la calidad se obtiene mejorando día a día el proceso de producción, mantenimiento y gestión del software. Para optimizar la calidad de los productos y/o servicios es preciso conocer al cliente y sus necesidades, conocer a la competencia y poseer un modelo de calidad.

Esto último permitirá incrementar la fiabilidad, reducir el mantenimiento, aumentar la satisfacción del cliente, mejorar la dirección del proyecto, detectar errores pronto / rápido e incrementar el beneficio. Pero también la madurez de los procesos de desarrollo y las técnicas adecuadas para mejorarlos y tener una gestión adecuada de las mejoras de dichos procesos.

Hay que saber en qué nivel de madurez, en cuanto a la calidad, se encuentra la organización, para poder determinar qué tipo de acciones son las más adecuadas en cada momento. Una organización inmadura se caracteriza por:

- Realizar procesos improvisados; incluso procesos especificados no son seguidos ni se exige su cumplimiento.
- Son organizaciones que "reaccionan"; los directivos "apagafuegos" se centran en resolver las crisis momentáneas.
- Sobrepasan presupuestos y calendarios, pues no se basan en estimaciones reales.
- Suelen disminuir la calidad y la funcionalidad para cumplir fechas.
- Carecen de bases objetivas para juzgar la calidad o para resolver problemas.
- Reducen o eliminan actividades de mejora de la calidad (revisiones, pruebas, etc.) cuando los proyectos se retrasan.

Crosby propone un modelo de 5 etapas de madurez de la gestión de calidad en la empresa:

Fuente: <http://www.computerworld.es/archive/la-calidad-y-la-ingenieria-de-software>

ETAPA	GESTION DE LA CALIDAD	COSTES TOTALES DE LA CALIDAD (% VENTAS)
Incertidumbre.	La calidad, lejos de ser considerada por la dirección como una de las más importantes herramientas de gestión con que puede contar toda organización, es postergada a un rincón de algún departamento operativo (producción, ingeniería, etc.), donde exclusivamente se realizan ciertas labores de evaluación y selección.	Si bien no se realiza ninguna estimación del coste, este puede llegar a sobrepasar el 20%.
Despertar.	La empresa comienza a asumir la importancia de la calidad pero todavía le otorga un papel secundario, por lo que no invierte lo que debería en ello.	Se estima un coste del 3% cuando realmente asciende al 18%.
Ilustración.	La calidad comienza a ser entendida y analizada desde una óptica de trabajo, que posibilita importantes avances en el tratamiento de los problemas. Dichos problemas son resueltos de manera ordenada, eficaz y definitiva, sin buscar culpables. El departamento de calidad, por fin asume definitivamente un papel relevante en la organización.	Se estima un coste del 8% cuando realmente asciende al 12%.
Sabiduría.	Se da un especial énfasis a la prevención. Los trabajadores y directivos reconocen la influencia decisiva de la calidad en los éxitos conseguidos.	Se estima un coste del 6,5% cuando realmente asciende al 8%.
Certeza.	Situación casi idílica, no obstante alcanzable en algunos casos, donde la gestión de la calidad, centrada en la prevención, es tan eficaz que los problemas casi no existen.	Los costes estimados en el 2,5% coinciden con los reales.

Fuente: Crosby, 1.991: pp. 38-39.

1 Incertidumbre: realidad confusa y sin compromiso.

2 Despertar: reconocimiento de la importancia de la gestión de la calidad pero sin un compromiso de invertir en ella.

3 Ilustración: compromiso de mejora, enfrentando los problemas sin desviar la acción a otros.

4 Sabiduría: participación, medición precisa e intentos de hacer permanentes las mejoras logradas.

5 Certeza: gestión de la calidad como parte esencial de los sistemas de la empresa.

El "Software Engineering Institute" de la Universidad de Carnegie - Mellon, se basó en el modelo de "madurez" de Phillip Crosby para crear un método para valorar el proceso de desarrollo de software: Modelo de Madurez de la Capacidad para Crear Software, CMM (Capability Maturity Model)

El CMM: Es un método. Por una parte se centra en el estudio de los procesos de software, en contraposición al interés por los productos software. Ejemplo: definición y desarrollo de los requisitos del software; generación de datos de prueba; planificación de la instalación del software. Por otra, se fundamenta en que la "madurez" del proceso es un indicador para construir software de calidad.

El método tiene 5 niveles de "madurez" y 18 áreas clave, cada una de éstas incluyen prácticas clave, que a su vez encierran preguntas concretas.

El modelo, basado en sondeos, permite que las empresas puedan conocer en qué nivel

se encuentra actualmente su proceso de producción de software, qué tipo de mejoras prioritarias se deben realizar y en qué aspectos o actividades críticas del proceso de desarrollo pueden intervenir con cierta garantía de éxito (estrategia de mejora del proceso)

Finalmente es recomendable tener en cuenta las siguientes "medidas" a aplicar para asegurar la calidad de los productos software:

- En el contrato de software: formular claramente y de forma precisa las responsabilidades, productos a entregar, etc. del proveedor; usar cláusulas estándar y listas de comprobación; revisar los borradores del contrato (comprador económico, auditor, abogado, etc.); negociar el borrador del contrato con el proveedor; obtener la orden del contrato firmada; modificar el contrato en caso de "problemas" serios.
- En el producto software: definir los requerimientos de calidad del producto; utilizar herramientas apropiadas para describir los requerimientos; evaluar la viabilidad; guías de programación; guías de documentación; documentación técnica consistente; gestión de la configuración; revisión de la documentación técnica; revisión de código; pasos de test; predicción del rendimiento; formación técnica de los miembros del equipo
- En el proceso de software: proveer con antelación suficiente los recursos necesarios: personas, hardware, software, herramientas, etc.; estructurar el proceso de software por fases; descripción clara del trabajo de cada miembro del equipo; reuniones del proyecto planificadas; reuniones de decisión de fase planificadas; control periódico de las tendencias y costes de los hitos; control de resultados e información de los hitos; monitorización periódica del riesgo y su prevención; información y motivación de los miembros del equipo; formación de los miembros del equipo.
- En la documentación de software: estándares de contenido: estructura, sumario, glosario, índice, etc.; estándar de presentación: formato de la hoja, identificación, clasificación, paginación, estado, etc.; instrucciones al equipo: guías de estilo de formulación, visualización, ejemplos; suministro de herramientas; administración de los documentos de forma profesional; revisiones / versiones.

La calidad es hoy una de las mayores ventajas competitivas para las empresas desarrolladoras de software

La calidad es un concepto vacío si no se tienen en cuenta las necesidades reales de los clientes. Así pues, algunas definiciones de calidad serían: calidad es el cumplimiento de los requisitos del cliente; calidad es satisfacer al cliente; o calidad es lo que el cliente dice

o piensa que debe ser.

Estas definiciones necesitan desarrollar los siguientes aspectos básicos: las expectativas del cliente deben ser traducidas en requisitos; desarrollar un sistema para cumplir con dichos requisitos de forma previsible, a la primera; establecer una metodología de actuación para llegar a "cero defectos" en el trabajo; disponer de un procedimiento de medida para el control de la satisfacción del cliente.

Previamente a ello, será preciso conocer perfectamente y a fondo las capacidades de la empresa: su organización y procedimientos internos de gestión; sus métodos estandarizados; su sistema de control y seguimiento de los procesos de desarrollo; y sus herramientas de soporte.

Una organización no preparada para dar calidad, en el sentido esperado por sus clientes, caerá muy rápidamente en la situación de expectativas no cumplidas o no alcanzadas.

La calidad tiene muchos enemigos, entre los más significativos están: la inconsistencia, la falta de disciplina, los compromisos insostenibles y la impaciencia. Esta última tiene relación con la frase: "la calidad no se consigue de la noche a la mañana".

Para establecer un "movimiento" para la calidad, en las empresas, debe iniciarse un mecanismo que identifique las necesidades del cliente. No se debe presuponer nada y la mejor manera de conocer qué quieren y cómo lo quieren es preguntándoles directamente a ellos. Escucharles de forma activa, asesorarles y demostrarles que sus necesidades son prioritarias para la empresa.

Hay una ley universal entorno al concepto de entropía que dice: todo sistema en estado natural, sin el aporte de energía exterior, evoluciona hacia un desorden creciente. Esto nos sugiere que todo aquello que aparece en equilibrio estático, aparentemente va hacia el desorden y por ende no puede alcanzar un estado de armonía. Perfectamente aplicable a los modelos estáticos o dinámicos de calidad.

La calidad es un proceso de mejora continua que no tiene fin y que se encuentra cómoda cuando se ha instalado en un entorno cambiante, donde el cambio cultural es su principal melodía y cuyo objetivo es buscar la completa satisfacción del cliente.

Focalizar en la calidad, como mejora, nos lleva al concepto actual de gestión de la calidad, lo que implica tres tipos de actuaciones: la planificación de la calidad en el futuro; la implantación y el desarrollo de actividades y programas para la consecución de la calidad; el control de los resultados obtenidos.

La calidad dirigida a los resultados (incrementar el rendimiento, mejoras en el control de costes, incrementar el beneficio) es una estrategia que contempla: gestionar la calidad con ciclos de mejora cortos; buscar mejoras inmediatas y rápidas, las que tengan elevadas posibilidades de éxito; que estén unidas a los objetivos de la empresa; no invertir demasiado tiempo en instaurar procedimientos, sí en obtener resultados; implantar sólo los procedimientos absolutamente necesarios que aseguren formación, motivación, participación, compromiso, modelos y métricas.

Pero, ¿por qué fracasan las iniciativas de calidad? Según J. M. Juran, en las empresas que fracasaron se dio un conjunto de los siguientes factores: entre los objetivos prioritarios de la empresa no figuran la calidad, la mejora de la satisfacción del cliente, la reducción de los costes que entraña la falta de calidad y la mejora de los procesos fundamentales; no se ha incorporado la gestión de la calidad a la gestión de la empresa; no se utiliza evaluación comparativa para la calidad (métricas o sistemas para medir la calidad), en muchos casos sólo se introducen métricas relacionadas con las finanzas (ventas, costes, beneficio, etc.) y se olvidaron las relacionadas con la calidad (satisfacción del cliente, calidad competitiva, etc.); no introdujeron la utilización del "reconocimiento", público o privado; no revisaron el sistema de retribución, introduciendo una variable relacionada con la calidad; se olvidaron de formar a los directivos en lo relativo al proceso de gestión de la calidad; no fomentaron la autogestión en la mejora de la calidad; no redujeron su lista de proveedores, quedándose sólo con aquellos que se toman en serio la calidad; fueron reacios a asociarse con sus proveedores (partnership); no revisaron exhaustivamente la gestión de la calidad del proceso operativo; faltó liderazgo personal de la dirección; las iniciativas de calidad se redujeron sólo a exhortaciones tales como lemas y carteles.

Los directivos que triunfan, ¿qué han hecho?: establecieron la visión y la política de calidad; se ocuparon de formar al personal; establecieron los objetivos a alcanzar; planificaron la manera de alcanzar los objetivos, estrategias y tácticas; escucharon la voz del cliente; potenciaron la figura del cliente interno; definieron métricas de calidad; realizaron una reingeniería de sus procesos de negocio; gestionaron y midieron la calidad con el mismo rigor que se hace con la rentabilidad; difundieron y premiaron los resultados.

Como resumen diremos que el esquema de trabajo de un equipo directivo comprometido con la calidad es:

Establece métodos y mecanismos que permitan a la empresa conocer mejor las expectativas de los clientes.

- Evita que se produzca la fragmentación de la empresa.
- Se ocupa de establecer objetivos, normas y estándares de calidad.
- Gestiona al personal, convirtiéndolo en un eficaz y decidido apoyo a la calidad.
- Implanta eficaces sistemas de supervisión y control, orientados a la calidad.
- Incorpora la visión del consumidor para evitar deficiencias en el diseño. - Estimula y desarrolla el trabajo en equipo.
- Establece eficaces mecanismos de "feedback".
- Establece eficaces mecanismos de comunicación.

El modelo de Calidad Total postula que: la calidad es responsabilidad de todos, en especial de los niveles de dirección; la dirección debe convertir toda la empresa en un "sistema de calidad"; la calidad debe ser construida en todas las fases y procesos, empezando por el diseño; la empresa debe garantizar la calidad en las fases de uso, consumo o posesión del producto / servicio; la calidad se genera en todas las áreas de la organización; la calidad no la determina la empresa, la definen y califican los clientes (visión "desde fuera").

Todo lo anterior podemos resumirlo en dos conceptos fundamentales: orientarse a los clientes y organizarse por procesos. La visión de una empresa organizada por procesos es tal que: la actividad empresarial está constituida por "n" procesos, cada uno de ellos genera valor añadido; cada proceso contiene un conjunto de tareas; cada tarea coincide en general con el departamento o unidad de negocio que la lleva a cabo; cada tarea genera un "output" que es el "input" de la tarea que le sigue; se trabaja por la mejora

continua de cada "output" (El KAIZEN); aparece el concepto de "cliente interno".

#### **4.1 Definición de calidad CALIDAD:**

**CONJUNTO DE PROPIEDADES Y DE CARACTERÍSTICAS DE UN PRODUCTO O SERVICIO, QUE LE CONFIEREN APTITUD PARA SATISFACER UNAS NECESIDADES EXPLÍCITAS O IMPLÍCITAS (ISO 8402)**

Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario.

•□Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

#### **Calidad de software**

es el desarrollo de software basado en estándares con la funcionalidad y rendimiento total que satisfacen los requerimientos del cliente.

La **calidad del software** es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

Adicionalmente se puede seguir los siguientes aspectos para evaluar la calidad del software:

Funcionalidad Confiabilidad Usabilidad Eficiencia "Mantenibilidad" Portabilidad

En la Calidad de software se tienen las características propias del software y que son aquellas que tu quieres controlar y asegurar, el software es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla; El software puede tener errores, incidencias pero no son similares a lo que cualquier equipo de carácter físico.



La calidad del software se encuentra a la par con la calidad tradicional, pero un paso atrás, debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad de software tiene 50 a 60 años.

### **Medición del software**

En el software lo que se mide son atributos propios del mismo, se descompone un atributo general en otros más simples de medir, a veces se mide bien o mal ya que la descomposición del atributo genérico de calidad en otros sub-atributos se torna irreal, se mide con datos estadísticos no avalados, es imposible decir que la medición se hace en forma correcta.

El concepto de medida va de más a menos, va de lo general a lo concreto y lo concreto es asociado a la métrica, cuya combinación te daría el nivel de calidad o seguridad de tu producto. Las ciencias bien estructuradas se basan en medidas bien hechas, se basan en la matemática.

### **Tipos de medidas**

- Número de errores durante un periodo determinado.
- Fallo en la codificación o diseño de un sistema que causa que el programa no funcione correctamente o falle.
- Tamaño de un producto informático (líneas de código)
- Métrica de punto función (IBM): relaciona funcionalidades que ofrecía.
- Estimación de costes y esfuerzos.
- COCOMO

### **Utilidad de la medida del software**

Consecuencia de su proceso interno de asegurar la calidad, cuantificar los atributos que constituyen la calidad para el usuario final, ahí tenemos los resultados cuantitativos. Saber que aquello que al usuario final le interesa lo tenga o no un producto y permita cuantificar almacenar otros productos.

- Normativa ISO 9126, medida de la calidad de software descomponiendo atributos, para no tener márgenes de error e interpretación.
- Atributo de funcionalidad.
- Atributo de capacidad de respuesta frente a errores externos.
- Atributo de nivel de seguridad. La calidad no puede existir sin

seguridad, un producto sin seguridad sería un producto sin calidad. El observador o usuario final indica que atributos más o menos importantes de seguridad.

No se puede medir la calidad del software de forma correcta debido a su naturaleza, la certificación se da a los procesos, la correcta consecución de los mismos garantizaría un buen software. No se puede medir al software como tal, sino los atributos que la conforman, tales métodos de medida deben ser exactos.

El usuario final mide la calidad del software según lo que tenga o no, es en ese sentido de que la calidad del software depende de quien la juzgue. El hecho de que una empresa tenga certificación en calidad de software no garantiza que su software sea de calidad.

#### **4.2 Características del software**

Los procesos de desarrollo, artefactos, gestión de proyectos, análisis y diseño, especificación de requerimientos, arquitectura, son solo algunos de los componentes que se aglomeran para conformar la ingeniería de software (IS) como disciplina para la creación y mantenimiento de software.

Dentro de ésta, existe un subconjunto de teorías, herramientas y métodos orientados a lo que se denomina la calidad del software. Para resumir de alguna manera la amplitud de este concepto, se puede decir que la calidad de software ha sido usada desde un simple argumento de venta, hasta verdaderos estudios formales y usos de métricas para el desarrollo de software.

Extrañamente dentro de la IS, la calidad del software es muy complicada de definir y de enmarcar en un simple concepto teórico, por lo que en esta nota, me concentraré solo en las diversas características que permiten describirla y en los elementos que importan específicamente al diseñador de software.

Una idea general sobre un software de calidad es aquel que debiera cumplir con los requerimientos funcionales y de performance además de ser mantenible, confiable y aceptable. Veamos cada uno de las principales características que hacen a un software de calidad.

**Mantenibilidad:** el software debe ser diseñado de tal manera, que permita ajustarlo a los cambios en los requerimientos del cliente. Esta característica es crucial, debido al inevitable cambio del contexto en el que se desempeña un software.

**Confiabilidad:** incluye varias características además de la confiabilidad, como la seguridad, control de fallos, etc.

**Eficiencia:** tiene que ver con el uso eficiente de los recursos que necesita un sistema para su funcionamiento.

**Usabilidad:** el software debiera ser utilizado sin un gran esfuerzo por los usuarios para los que fue diseñado, documentado, etc.

Como puede observarse, las diversas características con las que se desea que cumpla un software de calidad varían ampliamente. Algunas tienen que ver con el usuario que interactúa con el sistema, otras con el líder de proyecto y diseñadores, otras características parecen muy abstractas y hasta indefinidas, etc.

Para ordenar este aparente caos de indefiniciones y características abstractas, con el fin de poder medirlas, estimarlas e implementarlas, la IS ha desarrollado desde los primeros días de su existencia, diferentes procesos de desarrollo.

Esta búsqueda para poder controlar y medir la calidad del software, es tal vez una de las principales causas que han inspirado el estudio y definición de un sinnúmero de metodologías, técnicas y herramientas de la IS. Como resultado de mi experiencia personal y lo extraído de otras lecturas, me animo a decir que no es necesario el uso de un gran esfuerzo ni dedicación de gran cantidad de recursos para lograr software de calidad.

Las empresas y equipos de desarrollo deben saber que con la adopción de solo algunas prácticas de la IS, ya es suficiente para estar en el buen camino. Bien, ¿qué se necesita entonces?

- Como ya lo expresara en un post previo, toda empresa o equipo de desarrollo de software debe adoptar un proceso de desarrollo. ¿Cuál?, como mínimo EL QUE LE CONVenga.

Hay una gran variedad de procesos de donde tomar los elementos más convenientes para alinear los desarrollos con algunas características de la calidad del software vistas previamente.

- También se necesita coherencia desde el principio de cada proyecto. En ese momento deben definirse, cuantificarse y/o especificarse las características de calidad a cumplirse en ese producto.
- También se requieren las herramientas necesarias que ayuden al equipo para llevar adelante todas las tareas necesarias en relación a alcanzar los objetivos de calidad planteados.
- Es muy importante también, disponer de personas preparadas técnicamente y liderados por al menos un profesional con experiencia, que formen un equipo con la capacidad de adaptarse y mejorar continuamente.

Como puede observarse, tomar por el camino del desarrollo de software de calidad no significa disponer de grandes inversiones, sino de alinear los recursos disponibles, prepararlos y coordinarlos adecuadamente. Llegado el momento de escalar, o desear el logro de alguna certificación para ampliar mercados, o sencillamente buscar ser una empresa que logre desarrollar productos de calidad, será mucho mejor y más simple, si las empresas siguieran estos lineamientos mínimos para cuando llegue ese momento.

## **CARACTERÍSTICAS DEL SOFTWARE**

- El software se desarrolla o construye; no se manufactura en el sentido clásico. A pesar de que existen similitudes entre el desarrollo del software y la manufactura del hardware, las dos actividades serían diferentes en lo fundamental. En ambas la alta calidad se alcanza por medio del buen diseño, la fase de manufactura del hardware puede incluir problemas de calidad existentes en el software.
- El software no se desgasta. El software es inmune a los males ambientales que desgasten el hardware. Por lo tanto la curva de tasas de fallas para el software debería tener la forma de la “curva idealizada”. Los defectos sin descubrir causan tasas de fallas altas en las primeras etapas de vida de un programa. Sin embargo, los errores se corrigen y la curva se aplanan: el software no se desgasta, pero sí se deteriora.
- A pesar de que la industria tiene una tendencia hacia la construcción por componentes, la mayoría del software aun se construye a la medida. Un componente de software se debe diseñar e implementar de forma que puede utilizarse en muchos programas diferentes.

- Los componentes reutilizables modernos encapsulan tanto los datos como el proceso se aplican a estos, lo que permite al ingeniero de software crear nuevas aplicaciones nuevas a partir de partes reutilizables
- Sistema mínimo. En ocasiones, un programa debe ser asociado a un procesador antes de ser usado, lo cual forma un sistema mínimo. Tal es el caso, aunque no muy común, de los programas que son parte integral de un procesador (chip en una computadora) y que están “microprogramados”.
- Sistema típico. Con más frecuencia, la computadora y los productos de software son administrados por un sistema operativo, el cual interactúa a través de equipo periférico (hardware) con un operador y otros equipos como los dispositivos de almacenamiento, monitores, impresoras, etc.
- Sistema complejo. Los sistemas más difíciles de asegurar la calidad son aquellos donde se involucra el control -a través de la computadora- de sistemas electrónicos, dispositivos mecánicos o hidráulicos, procesos o plantas.

### **Características del entorno de los productos de software**

Para facilitar su conceptualización, se proponen definiciones sencillas y prácticas a continuación:

- **Aplicación:** Representación del giro sobre el cual el software va dirigido (negocios, ingeniería, medicina, etc.).
- **Ambiente de Uso:** Espacio físico y condiciones en que se utiliza el software.
- **Riesgos y consecuencias de fallas:** Lo que puede implicar que el software falle.
- **Computadora anfitriona:** La computadora donde correrá el software.
- **Madurez del desarrollador:** Experiencia en el desarrollo de software similar.
- **Experiencia del usuario :** La familiaridad de los usuarios con computadoras o software similares
- **Apoyo de los desarrolladores:** Asesoría o ayuda por parte de los desarrolladores.
- **Experiencia de los desarrolladores:** Tiempo trabajado en desarrollos de software similares.
- **Interacción con el usuario final:** Comunicación ente usuarios y desarrolladores.
- **Restricciones comerciales:** Falta de presupuesto o de tiempo.

- **Metodología de desarrollo:** El uso de algún método establecido para la creación del software.
- **Lenguajes de programación :** El lenguaje o paquete computacional seleccionado para desarrollar el software
- **Complejidad del software:** Grado en que se tenga que involucrar a muchos elementos físicos (periféricos), que de alguna manera ayudan a la ejecución del software.