

RESUMEN SINTETICO DE INGENIERIA

Software:

Puede verse como un producto que transforma información o sino como un vehículo de entrega de producto, un software de comunicación, un sistema operativo, entre otros.

Según la IEEE un software son instrucciones, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Es un elemento lógico, que se desarrolla y se desgasta en base a los cambios que se le hacen a lo largo del tiempo (a diferencia del hardware).

El software puede ser **genérico**, que son sistemas aislados vendidos al mercado abierto, o **personalizados** que son hechos para un cliente en particular. Hoy en día se desarrolla software cada vez más genérico para después adaptarlo a lo que quieren los clientes.

Ingeniería de software:

Es una disciplina que comprende todos los aspectos de producir un software, desde juntar los requerimientos incluyendo evolución y desarrollo, hasta su constante mantenimiento, actualización y conservación hasta que se decide retirarlo de servicio.

Participan los **clientes ejecutivos**, es decir los que patrocinan y financian el proyecto, **los desarrolladores** que construyen el sistema y **el usuario** que utiliza el sistema

Ingeniería por épocas:

- Años 60: se estudia como explotar la tecnología para satisfacer las necesidades de las organizaciones existentes.
- Años 70: surge el software realizable en tiempo planeado y controlado.
- Años 80: aumento de importancia de la productividad de desarrollo.
- Años 90 – actualidad: creciente necesidad de satisfacer necesidades explícitas e implícitas del usuario.

Un **ingeniero de software** debe conocer:

- Tecnologías.
- Productos.
- Técnicas de administración de proyectos.
- Sus responsabilidades incluyen:
 - Mantener confidencialidad entre empleados y clientes.
 - No aceptar tareas que no puede realizar.
 - Conocer sobre patentes y leyes de copyright.
 - Usar sus habilidades de forma apropiada y sin malas intenciones.

Para esto hay organizaciones con códigos de ética a respetar.

RESUMEN SINTETICO DE INGENIERIA

Técnicas de comunicación

La primer tarea es saber lo que el usuario **quiere, como , cuando y porqué**. Comunicarse es la base para saber las necesidades del cliente , a su vez, es la principal fuente de error. Al hablar de necesidades, se habla de **requerimientos**.

Los **requerimientos** son descripciones de algo que el sistema puede hacer para lograr el propósito del mismo. Según la IEEE es una condición que necesita el usuario para resolver el problema o un objetivo.

Los requerimientos se obtienen de la documentación , stakeholders, o especificaciones de sistemas similares.

Los **stakeholder** son personas que serán afectadas por el sistema directa o indirectamente como usuarios finales, ingenieros, gerentes y expertos del dominio.

Hay tres puntos de vista:

1. **Interactuadores:** Personas o sistemas que interactúan directamente con el sistema.
2. **Indirecto:** No utilizan el sistema ellos mismos pero influyen en los requerimientos de algún modo.
3. **Dominio:** Características y restricciones del dominio que influyen en los requerimientos.

Elicitación de requerimientos

La **elicitación de requerimientos** es el proceso para adquirir todo lo necesario para producir un modelo de las necesidades del dominio de problema.

Los objetivos de la elicitación son:

- ☐ Conocer el dominio del problema y el sistema actual del mismo.
- ☐ Comunicarse con clientes para saber sus necesidades y expectativas sobre el sistema a crear.

Existen **problemas de comunicación** que dificultan la obtención de requerimientos:

- ☐ El cliente no puede expresarse claramente.
- ☐ No sabe que quiere, o como se implementa la tecnología.
- ☐ Tiene miedo de ser incompetente por no saber sobre tecnología.
- ☐ Etc.

Las técnicas de elicitación disponen de dos métodos:

Discretos: Cosas como mostrar la documentación y datos existentes, investigar y visitar el lugar u observar al ambiente de trabajo, esta ultima es barata y útil porque se sabe que hacen, pero las personas pueden no hacer su trabajo como acostumbran ya que se pueden sentir incomodos por ser observados.

Interactivos: Pueden ser cuestionarios para recabar información y opiniones de encuestados, son económicos, de respuesta rápida y fácil análisis, pero pueden tener pocas respuestas, estar incompletos y no permiten un análisis corporal.

RESUMEN SINTETICO DE INGENIERIA

También se pueden usar entrevistas que recolectan información interactuando cara a cara, conociendo la opinión y sentimientos del entrevistado, aunque son costosas, consumen tiempo y recursos, no pueden realizarse a distancia y dependen del entrevistador.

También está el Brainstorming donde se generan todas las ideas posibles en un corto tiempo y se planea en conjunto llevando a cabo reuniones en grupos para analizar problemas, dichas reuniones ahorran tiempo, involucran usuarios y obtienen desarrollo creativo, pero es difícil conseguir tiempo, gente que se integre y organice al respecto.

Los métodos discretos son menos perturbadores que los métodos interactivos. Como un método por si sólo no da mucho información se combinan varios de estos para una mayor efectividad.

Tipos de requerimientos

Funcionales: un requisito funcional describe como interactuará el sistema con el ambiente, esto incluye, que debe hacer el sistema, incluso como NO debe comportarse y la funcionalidad del mismo, los requisitos funcionales son independientes de la solución.

No funcionales: un requisito no funcional implica restricciones del sistema que limitan las elecciones para resolver el problema presentado, requerimientos de producto, organizacionales y externos.

Otros tipos...

Del dominio: Características y restricciones del dominio al sistema, difíciles de interpretar.

Por prioridad: estos requisitos se dividen en tres

1. Deben ser satisfechos.
2. Deseables pero no indispensables.
3. Se pueden eliminar si se da el caso.

Del usuario: Declaraciones y diagramas que se espera que el sistema provea, pueden ser poco claros.

Del sistema: Establecen servicios y restricciones del sistema.

La **ingeniería de requerimientos** es el proceso donde las necesidades del cliente son transformadas en especificaciones precisas, claras, consistentes y completas del comportamiento del sistema.

Permite gestionar necesidades, ayudando a predecir cronogramas de proyectos, esto disminuye costos y retrasos, provocando una mejora de la calidad, la comunicación del equipo y evitando rechazos de usuarios finales.

Estudio de viabilidad

Se elabora un informe que recomienda la conveniencia o no de realizar el proceso, responde a las siguientes preguntas:

- ☐ ¿El sistema contribuye a los objetivos de la organización?
- ☐ ¿Es implementable con la tecnología actual?

RESUMEN SINTETICO DE INGENIERIA

- ¿Es implementable con las restricciones de costo y tiempo?
- ¿Puede integrarse a otros sistemas que existen en la organización?

Al contestar esas preguntas se redacta un informe para saber si se continua o no con el proyecto.

Mas conceptos de requerimientos

La **especificación de requerimientos** tiene como objetivo permitir que los desarrolladores expliquen como han entendido lo que el cliente quiere para el sistema, indicar las características del sistema resultante y mostrar las demostraciones para convencer al cliente que el sistema es lo que había pedido.

Hay un estándar de la IEEE que brinda una colección de buenas prácticas, posee un listado de cosas que espera que haga el sistema y definiciones de términos técnicos

La **validación de requerimientos** es el proceso de certificar la corrección del modelo de requerimientos contra lo que quiere el usuario. Trata de demostrar que los requerimientos definidos son los que estipula el sistema, se describe el ambiente sobre el que opera.

No alcanza validar post desarrollo porque cuanto más tarde se detecta un error más cuesta corregirlo, y esto provoca que el error se vuelva una bola de nieve de defectos, por eso conviene validar al especificar los requerimientos.

Las **técnicas de especificación de requerimientos** pueden ser **estáticas** donde se describe el sistema con entidades u objetos, atributos y relaciones, pero no describe como las relaciones cambian en el tiempo, éstas técnicas son útiles y adecuadas si el tiempo no es un factor mayor, sino pueden ser **dinámicas** donde se considera el sistema en función de los cambios que ocurren a lo largo del tiempo, se considera que el sistema está en un estado particular hasta que un estímulo la obliga a cambiar de estado.

Historias de usuario

Se utilizan en metodologías de desarrollo ágiles, son acompañadas de discusiones con usuarios y pruebas de validación. Debe ser limitada ya que es una forma rápida de administrar requisitos sin elaborar documentos formales y sin tener mucho tiempo para administrar los mismos.

Las HU pueden también responder a requisitos cambiantes, con posibilidad de discutirse con clientes

Suelen responder a las preguntas:

- ¿Quién se beneficia?
- ¿Que se quiere?
- ¿Cuál es el beneficio?
- Con el esquema **como rol quiero algo para poder tener un beneficio**.

Las HU son independientes unas de otras, negociables, estimables en el tiempo, pequeñas, se pueden verificar y son valoradas por clientes o usuarios.

RESUMEN SINTETICO DE INGENIERIA

Los **criterios de aceptación** es el criterio por el cual se define si una HU puede ser dada como hecha, se deben definir en la etapa inicial antes de codear porque complementan las HU y dan a entender como se espera que se comporte el sistema. Si hay mas de 4 criterios para una historia se debe considerar dividirla.

Beneficios:

- ❑ Rápida implementación de cada historia.
- ❑ Se requiere poco mantenimiento.
- ❑ El proyecto se puede dividir en pequeñas entregas.
- ❑ Se puede estimar el esfuerzo de desarrollo.
- ❑ Ideal para proyectos con requisitos cambiantes y poco claros.

Desventajas

- ❑ Una HU sin criterios de aceptación puede quedar ambigua, lo que dificulta su uso.
- ❑ Requieren un contacto permanente con el cliente.
- ❑ Difíciles de usar en proyectos grandes.
- ❑ Se necesitan desarrolladores adecuados.

Las **épicas** son conjuntos de HU que se agrupan por algún denominador común.

Casos de Uso

Proceso de modelado de las funcionalidades del sistema en termino de los eventos que interactúan entre los usuarios y el sistema. Surge en el modelado orientado a objetos para luego volverse independiente y aplicable a cualquier metodología de desarrollo.

Beneficios:

- ❑ Captura requerimientos funcionales.
- ❑ El sistema se descompone en piezas manejables.
- ❑ Los CU tienen lenguaje común y entendible.
- ❑ Permite estimaciones de alcance y esfuerzo a realizar.
- ❑ Proporcionan una línea base para definir planes de prueba y documentación.
- ❑ Permite el seguimiento de requisitos.

Esta compuesto por un **diagrama** que ilustra interacciones entre sistema y usuario, los **escenarios** que describen la interacción entre el actor y sistema para realizar la funcionalidad.

RESUMEN SINTETICO DE INGENIERIA

El CU representa un objetivo individual del sistema y describe la secuencia de actividades e interacciones para alcanzar dicho objetivo.

Los actores inician una actividad, representan un papel desempeñado por un usuario que interactúa, este actor puede ser una persona, sistema externo o dispositivo externo que dispare un **evento**.

Las **relaciones** pueden ser:

- Asociaciones (relación entre actor-CU).
- Extensiones (un CU extiende la funcionalidad de otro CU).
- Uso o inclusión (combina los pasos comunes de 2 o mas CU que tienen partes iguales).
- Dependencia (indica que un CU no puede realizarse hasta que pase otro CU).
- Herencia (relación entre actores donde uno hereda las funcionalidades de uno o varios).

Los escenarios describen la interacción con el escenario y eventos alternativos (surgen de imprevistos).

Los **pasos** para crear un CU son:

1. Identificar los actores.
2. Identificar los CU para requerimientos.
3. Construir el diagrama.
4. Realizar los escenarios.

CU – Características importantes

- Un CU debe representar una funcionalidad concreta, la descripción de los pasos del CU debe contener más de un paso para representar la interacción usuario-sistema.
- El uso de condicionales se limita a la invocación de excepciones.
- Las precondiciones no deben representarse en cursos alternativos, ya que no van a ocurrir.
- Los usos deben ser accedidos por lo menos desde dos CU

Diagramas de transición y estado

Los DTE tienen su origen en las máquinas de estado finito, es decir, describen el sistema como un conjunto de estados donde el sistema reacciona a eventos posibles, DTE es una técnica de elicitación dinámica.

Al estar en un estado S_i , al ocurrir la condición C_j hace que el sistema cambie al estado S_k , representado como una función $F(S_i, C_j) = S_k$

Formalmente, una máquina de estado finito se puede definir como una 5-tupla donde están:

1. Los eventos posibles.
2. El conjunto de estados.
3. Las funciones de transición.
4. El estado inicial.

RESUMEN SINTETICO DE INGENIERIA

5. Un conjunto de estados de aceptación o finales.

Los DTE solo tienen un estado inicial sin condiciones, y deben tener si o si un estado final.

Para **construir un DTE**:

1. Identificar estados.
2. Si hay estados complejos estos se pueden dividir.
3. Se identifican los cambios de estado con flechas.
4. Se analizan las condiciones y acciones para pasar de un estado a otro.
5. Se verifica la consistencia al final.

Redes de Petri

Las redes de Petri son utilizadas para especificar sistemas de tiempo real en los cuales es necesario representar ejecución concurrente. Dichos sistemas son diseñados para permitir ejecución simultanea de componentes de programación en varios o un solo procesador.

Las tareas concurrentes deben estar sincronizadas para lograr una correcta comunicación, es necesario asegurar la sincronía ya que las tareas en paralelo se ejecutan en un orden impredecible, las Redes de Petri NO son secuenciales.

Tienen eventos (TRANSICIONES) y estados (LUGARES o SITIOS).

Los **arcos indican con una flecha** la relación entre sitios y transiciones, se asignan tokens que se representan con números o puntos los cuales sirven para manejar la coordinación de transiciones y lugares para simular la ejecución de la red mediante transiciones habilitadas moviendo tokens de un estado a otro a través de las mismas transiciones.

El accionar de las transiciones está sujeta a condiciones y al ocurrir la transición se hacen verdaderas las post condiciones.

Las RP son **asincrónas** y el orden en que ocurren las cosas es uno de los permitidos.

También requieren una **sincronización** para que varios procesos puedan compartir información y recursos para resolver el problema, aunque esto debe ser controlado para que el sistema se mantenga integro y en correcto funcionamiento.

El sistema NO debe bloquearse en una RP.

Tablas de decisión

Es una herramienta que permite presentar en forma concisa las reglas lógicas que hay que utilizar para decidir las acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico.

Describe el sistema como un conjunto de posibles CONDICIONES satisfechas por el sistema en un momento dado, REGLAS para reaccionar ante los estímulos que ocurren al reunir un conjunto determinado de condiciones y ACCIONES a ser tomadas como resultado.

Se construyen con:

- Condiciones simples(V o F), y acciones simples,
- Hay 2^n reglas donde hay N cantidad de condiciones.

RESUMEN SINTETICO DE INGENIERIA

Hay que tener en cuenta:

- **Especificaciones completas** (cada regla de la tabla debe tener POR LO MENOS una acción).
- **Especificaciones redundantes** (reglas que poseen las mismas acciones).
 - Solución: reducir la tabla agrupando reglas que realizan las mismas acciones
- **Especificaciones contradictorias** (reglas que tienen mismas condiciones y tienen diferentes acciones cada una).

Análisis estructurados

El análisis estructurado es una actividad de construcción de modelos. Mediante una notación creamos modelos que representan el contenido y flujo de la información.

Posee tres objetivos primarios:

- Debe lograr describir lo que quiere el cliente.
- Establecer una base para crear un diseño de software.
- Definir un conjunto de requisitos que sean validables.

El **diagrama de flujo de datos** es una herramienta que permite visualizar el sistema como una red de procesos funcionales conectados entre sí. Representa la transformación de entradas a salidas y se usa en sistemas donde las funciones son más importantes y complejas que los datos que este maneja.

Se usa un rectángulo para representar una entidad externa o sistema que produce información, un círculo para un proceso o transformación y una flecha para uno o más elementos de datos.

- **Diccionario de datos:** listado organizado de todos los datos pertenecientes al sistema, se trata de una definición no ambigua de datos y elementos del sistema que nos permite representar información, revisar consistencia, etc.

El **modelo esencial** indica lo que el sistema debe hacer para satisfacer lo que quiere el usuario con una **mínima(o nula) explicación** de como se hace. Asume que hay tecnología perfecta, pero los usuarios están metidos en detalles de implementación y cuesta pensar en la tecnología perfecta.

El **modelo ambiental** define interfaces entre el sistema y el ambiente donde él mismo es ejecutado. Es lo más importante de la construcción de un modelo de requerimientos, pero a mayor proyecto hay mas flujos, terminadores y listas de acontecimientos

El **modelo de comportamiento** es la representación de como debe ser el comportamiento final que el sistema debe tener para manejar con éxito el ambiente dentro de los requisitos que pidió el usuario.

Modelo de procesos

Un **proceso de software** es un conjunto de tareas en secuencia y modelos asociados que producen un producto de software

RESUMEN SINTETICO DE INGENIERIA

El **modelo de proceso** es una representación **simplificada y abstracta** de un proceso de software, dicho modelo presenta una visión posible sobre el proceso.

El modelo incluye las siguientes características:

- Establece todas las actividades del proceso.
- Utilización de recursos, se encuentra sujeto a restricciones, y genera productos intermedios y finales.
- Se puede componer de subprocesos.
- Cada actividad posee entradas y salidas definidas.
- Las actividades se realizan en secuencia.

Son modelos simples que representan una visión de un proceso, incluye actividades que son parte de los procesos y productos de software y el papel de las personas involucradas.

Como **características** establece todas las actividades, utiliza recursos estando sujeto a restricciones y genera productos intermedios y finales, puede estar hecho de subprocesos, con cada actividad con salidas y entradas definidas. Sus actividades se organizan en secuencia y las restricciones del mismo pueden ser a una actividad, recurso o producto

Tipos de modelos de procesos de software

Los **prescriptivos** indican un conjunto de elementos del proceso y un flujo de trabajo, ósea la relación de los elementos.

Los **descriptivos** muestran como se realizan en la realidad. **Ambos deberían ser iguales**

Modelo en cascada: Las etapas se representa cayendo en cascada, completándose una antes que comience la siguiente, son útiles para diagramar que hacer y fácil de explicar a clientes, además de ser noob-friendly. Sus problemas es que no hay resultados exactos hasta que esta terminado, los errores mas graves aparecen al final, haciendo difícil su eliminación y demanda muchas pruebas además que congelar una fase es poco realista.

Modelo en cascada con prototipo: Se analizan las necesidades, se arma un prototipado y si hay algún problema se vuelve a esa etapa, hay menos errores que no usar un prototipo.

Modelo en V: Muestra la relación entre actividades de prueba y las de análisis y diseño. La vinculación entre ambos implica que si se encuentra problemas verificando entonces el lado izquierdo de la V puede ejecutarse de nuevo para solucionarlo.

Modelo de prototipos: Es una alternativa de especificación a la ambigüedad de proyectos reales, pudiendo ser **evolutivo** donde el objetivo es obtener el sistema a entregar , permitiendo que el sistema o sus partes se construyan rápido para aclarar y comprender aspectos y así el cliente , desarrollador y usuario quieran lo mismo. O **descartable** donde no hay funcionalidad y se usan herramientas de modelado. Se usan cuando los usuarios no ven modelos abstractos y no determinan lo que quieren inicialmente. Deben ser baratos, rápidos de hacer, fácil de experimentar con énfasis en la UI y contar con las cosas adecuadas.

Modelo desarrollado por fases: Se desarrolla de forma que puede ser entregado en piezas, existen dos sistemas funcionando, el operacional y el que esta en desarrollo. Puede ser **incremental** donde se parte de subsistemas y cada entrega añade un nuevo subsistema o

RESUMEN SINTETICO DE INGENIERIA

iterativo donde se entrega el sistema entero y cada entrega aumenta la funcionalidad de cada subsistema.

Modelo en espiral: Combina actividades de desarrollo con gestión de riesgo, tratando de mejorar los ciclos de vida clásicos y prototipos incorporando objetivos de calidad, además de eliminar errores y alternativas no bonitas al comienzo. Permite iteraciones, vuelta atrás y finalizaciones rápidas.

Metodologías ágiles

Es un enfoque incremental e iterativo de desarrollo de software, teniendo como objetivo producir software de alta calidad con un costo efectivo y en tiempo apropiado.

Tiene que esbozar valores y principios para desarrollar software rápidamente, de forma que se ofrezca una alternativa a los procesos tradicionales.

Se priorizan las tareas que dan resultados directos y reduce la burocracia tanto como sea posible.

Valores del manifiesto de las metodologías ágiles:

- **Individuos e interacciones** más que procesos y herramientas
- **Software operante** más que documentación completa
- **Colaboración con el cliente** más que negociaciones contractuales
- **Respuesta al cambio** más que seguir una planificación.

Como **principios** tiene:

- Satisfacer al cliente con entregas continuas.
- Estar dispuesto a cambios de requerimientos.
- Trabajar en conjunto de usuarios y desarrolladores.
- Hacer proyectos alrededor de motivaciones individuales.
- Dar ambiente y soporte a quienes lo necesiten.
- Crear software funcional.
- Atención a la excelencia técnica y buen diseño.
- La simplicidad es esencial.
- Cada tanto reflexionar sobre como ser más efectivo.

Los **problemas** de los principios son:

- Involucrar al cliente es clave y esta sujeto a otras responsabilidades que dificultan su participación.
- Priorizar cambios es difícil, más cuando las fechas de entrega se acercan.
- La simplicidad requiere trabajo extra.
- Hay organizaciones con procesos definidos muy arraigados, por lo que se les puede dificultar el cambio a las metodologías ágiles.

RESUMEN SINTETICO DE INGENIERIA

Generalmente, el documento es parte del contrato y al no haber mucha documentación es difícil reglamentarlo, además los libros sobre esto son para desarrollar sistemas nuevos, mientras que la ing de software se centra en actualizar y mantener sistemas existentes, que sin documentación se complejiza.

Ejemplos de metodologías

Xtreme programing: Es una disciplina basada en los valores de la sencillez, comunicación , retroalimentación, valentía y respeto.

Consiste en llevar a todo el equipo en la presencia de practicas simples con suficiente información para ver dónde está cada integrante y ajustar las practicas particularmente para cada uno.

Su proceso es la exploración(plantear HU), planificación (estimación de esfuerzo, acuerdos sobre cronogramas, el cliente establece prioridad de cada HU), la iteración (el cliente decide que HU se implementarán), la producción (realizar pruebas adicionales), mantenimiento (mantener funcionamiento y mejorarlo) y muerte del proyecto (no hay más HU para realizar y se documenta o no hay más fondos para el proyecto).

Scrum: Se aplican de manera regular mejores practicas y se consigue el mejor resultado, se origina de un estudio sobre como trabajan equipos altamente productivos.

Tiene un proceso iterativo e incremental buscando poder atacar todos los problemas que surgen durante el desarrollo del proyecto, el nombre scrum sale de que los Sprints se van solapando haciendo que no sea un modelo en cascado tradicional ,parecido al scrum de rugby.

Se aconseja aplicarlo en proyectos difíciles donde el caos es constante.

Calidad

Es una propiedad o conjunto de propiedades inherentes a algo que permiten juzgar su valor. El concepto de calidad resulta ser muy subjetivo para cada ser humano, y , debido a su ambigüedad es importante unificar su definición.

Criterios erróneos sobre la calidad suele ser pensar que la calidad es un producto de lujo o que es intangible o no medible, como pensar que se origina en el depto de calidad.

El concepto de calidad suele variar con el tiempo ya que las percepciones varían por las condiciones de cada persona y el producto que se juzga.

Según Crosbi la calidad es la conformidad con los requisitos, según Shewart la calidad tiene un lado subjetivo, según Feigenbaum se basa en cumplir las expectativas del cliente, Denning habla sobre la satisfacción del usuario.

Vemos que estos gurús de la filosofía de calidad proponen diferentes formas de definir la calidad que tienen su misma base en básicamente decir "Conformidad del cliente y los requisitos del producto".

Según las normas internacionales la calidad es el grado en el que un conjunto de características inherentes cumple con los requisitos. Es un conjunto de propiedades de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas.

RESUMEN SINTETICO DE INGENIERIA

En sistemas de información la calidad es importante por la relevancia que tienen en la actualidad. Se debe analizar la calidad desde un todo donde cada parte que los componen debe tener su análisis de calidad correspondiente.

Estos componentes de calidad son:

- Infraestructura (redes).
- Gestión (presupuesto, planificación, programación).
- Servicio (atención al cliente).
- Datos (los que ingresan al sistema)
- Información (de los datos).
- Software (aplicaciones).

Que una empresa sea de calidad, implica que cumpla correctamente todos los estándares de calidad de cada componente.

La calidad de Software esta dividida entre el proceso en desarrollo y el producto obtenido, ambos están estandarizados por un conjunto de normas para saber que cosas deben satisfacer para ser considerado de buena calidad . Mejoro mucho en los últimos años mas que nada por la mayor conciencia de la importancia de gestionar la calidad y las técnicas de gestión de calidad.

Normas de un producto de software: Esta la ISO 25000 que tiene divisiones para establecer un modelo de calidad detallado que incluye características para calidad interna, externa y calidad de datos. Establece una medición de calidad con guías de uso y medidas de calidad, a su vez tiene recomendaciones, requisitos y guías para la evaluación de un producto.

Normas de calidad de software: Dividida en modelos de procesos y de evaluación. El CMM es un modelo de evaluación, es un marco de referencia para desarrollar procesos efectivos y para evaluar procesos de la organización y ver prioridades de mejora. El CMMI posee dos vistas, una centrada en la madurez de la organización y otra centrada en actividades de mejora y evaluación de la capacidad de los diferentes procesos.

La familia de ISO 9000 es un conjunto de normas de gestión de calidad aplicables a cualquier tipo de organización para obtener mejoras y arribar a una certificación importante para competir en mercados globales.