



Capítulo 24 - Gestión de la calidad

Conferencia 1

Temas cubiertos

- ✧ La calidad del software
- ✧ Estándares de software
- ✧ Revisiones e inspecciones
- ✧ Software de medición y métrica

Gestión de la calidad del software

- ✧ Preocupados por asegurar que se alcance el nivel de calidad requerido en un producto de software.
- ✧ Tres preocupaciones principales:
 - A nivel organizativo, la gestión de la calidad se ocupa de establecer un marco de procesos y normas organizativas que permitan obtener programas informáticos de alta calidad.
 - A nivel de los proyectos, la gestión de la calidad supone la aplicación de procesos de calidad específicos y la comprobación de que se han seguido esos procesos planificados.
 - A nivel de proyecto, la gestión de calidad también se ocupa de establecer un plan de calidad para un proyecto. El plan de calidad debe establecer los objetivos de calidad del proyecto y definir los procesos y normas que se utilizarán.

Actividades de gestión de la calidad

- ✧ La gestión de la calidad proporciona una comprobación independiente del proceso de desarrollo de los programas informáticos.
- ✧ El proceso de gestión de la calidad comprueba los resultados del proyecto para garantizar que sean coherentes con las normas y objetivos de la organización
- ✧ El equipo de calidad debe ser independiente del equipo de desarrollo para que puedan tener una visión objetiva del software. Esto les permite informar sobre la calidad del software sin verse influenciados por cuestiones de desarrollo de software.

Gestión de la calidad y desarrollo de programas informáticos



Planificación de la calidad



- ✧ Un plan de calidad establece las cualidades deseadas del producto y cómo se evalúan y define los atributos de calidad más significativos.
- ✧ El plan de calidad debe definir el proceso de evaluación de la calidad.
- ✧ Debe establecer qué normas organizativas deben aplicarse y, en caso necesario, definir nuevas normas a utilizar.

Planes de calidad



- ✧ Estructura del plan de calidad
 - Introducción del producto;
 - Planes de productos;
 - Descripciones del proceso;
 - Objetivos de calidad;
 - Riesgos y gestión de riesgos.
- ✧ Los planes de calidad deben ser documentos cortos y concisos
 - Si son demasiado largos, nadie los leerá.

Alcance de la gestión de calidad



- ✧ La gestión de la calidad es particularmente importante para los sistemas grandes y complejos. La documentación de la calidad es un registro de los progresos y apoya la continuidad del desarrollo a medida que cambia el equipo de desarrollo.
- ✧ En el caso de los sistemas más pequeños, la gestión de la calidad necesita menos documentación y debería centrarse en el establecimiento de una cultura de la calidad.

La calidad del software



- ✧ La calidad, simplificando, significa que un producto debe cumplir con su especificación.
- ✧ Esto es problemático para los sistemas de software
 - Existe una tensión entre los requisitos de calidad del cliente (eficiencia, fiabilidad, etc.) y los requisitos de calidad del desarrollador (mantenimiento, reutilización, etc.);
 - Algunos requisitos de calidad son difíciles de especificar de manera inequívoca;
 - Las especificaciones de los programas informáticos suelen ser incompletas y a menudo inconsistentes.
- ✧ El enfoque puede ser la "adecuación al propósito" más que la conformidad de las especificaciones.

El software de adecuación para el propósito



- ✧ ¿Se han seguido las normas de programación y documentación en el proceso de desarrollo?
- ✧ ¿Se ha probado correctamente el software?
- ✧ ¿Es el software lo suficientemente fiable para ser puesto en uso?
- ✧ ¿El rendimiento del software es aceptable para un uso normal?
- ✧ ¿El software es utilizable?
- ✧ ¿El software está bien estructurado y es comprensible?

Atributos de calidad del software



Seguridad	Comprensión	Portabilidad
Seguridad	Testabilidad	Usabilidad
Fiabilidad	Adaptabilidad	Reutilización
Resistencia	Modularidad	Eficiencia
Robustez	Complejidad	Aprendizaje

Conflictos de calidad



- ✧ No es posible que ningún sistema sea optimizado para todos estos atributos - por ejemplo, mejorar la robustez puede llevar a una pérdida de rendimiento.
- ✧ Por consiguiente, el plan de calidad debe definir los atributos de calidad más importantes para el software que se está desarrollando.
- ✧ El plan debe incluir también una definición del proceso de evaluación de la calidad, una forma convenida de evaluar si alguna calidad, como la mantenibilidad o la robustez, está presente en el producto.

Calidad del proceso y del producto



- ✧ La calidad de un producto desarrollado se ve influida por la calidad del proceso de producción.
- ✧ Esto es importante en el desarrollo de software, ya que algunos atributos de calidad del producto son difíciles de evaluar.
- ✧ Sin embargo, existe una relación muy compleja y poco comprendida entre los procesos de software y la calidad del producto.
 - La aplicación de las aptitudes y la experiencia individuales es particularmente importante en el desarrollo de programas informáticos;
 - Factores externos como la novedad de una aplicación o la necesidad de un calendario de desarrollo acelerado pueden perjudicar la calidad del producto.

Calidad basada en el proceso



Estándares de software



- ✧ Las normas definen los atributos necesarios de un producto o proceso. Desempeñan un papel importante en la gestión de la calidad.
- ✧ Las normas pueden ser internacionales, nacionales, de organización o de proyecto.
- ✧ Las normas de producto definen las características que todos los componentes de software deben exhibir, por ejemplo, un estilo de programación común.
- ✧ Las normas de proceso definen cómo debe promulgarse el proceso de software.

Importancia de las normas



- ✧ Encapsulación de las mejores prácticas... evita la repetición de errores pasados.
- ✧ Son un marco para definir lo que significa la calidad en un entorno determinado, es decir, la visión que tiene esa organización de la calidad.
- ✧ Proporcionan continuidad - el nuevo personal puede entender la organización entendiendo las normas que se utilizan.

Normas de productos y procesos



Normas de productos	Normas de proceso
Formulario de revisión del diseño	Conducta de revisión del diseño
Documento de requisitos	Presentación de un nuevo código para construcción de sistemas
Formato de la cabecera del método	Proceso de lanzamiento de la versión
Estilo de programación Java	Proceso de aprobación del plan de proyecto
Formato del plan del proyecto	Proceso de control de cambios
Formulario de solicitud de cambio	Proceso de grabación de prueba

Problemas con las normas



- ✧ Es posible que los ingenieros de software no los consideren pertinentes y actualizados.
- ✧ A menudo implican demasiados formularios burocráticos.
- ✧ Si no cuentan con el apoyo de herramientas de software, a menudo se requiere un tedioso trabajo de llenado de formularios para mantener la documentación asociada a las normas.

Elaboración de normas



- ✧ Involucrar a los profesionales en el desarrollo. Los ingenieros deben entender el fundamento de una norma.
- ✧ Revise las normas y su uso regularmente. Los estándares pueden quedar obsoletos rápidamente y esto reduce su credibilidad entre los profesionales.
- ✧ Las normas detalladas deben tener un instrumento especializado apoyo. El exceso de trabajo de oficina es lo más importante queja contra las normas.
 - Los formularios basados en la web no son lo suficientemente buenos.

Marco normativo ISO 9001



- ✧ Un conjunto de normas internacionales que pueden utilizarse como base para desarrollar sistemas de gestión de la calidad.
- ✧ La ISO 9001, la más general de estas normas, se aplica a las organizaciones que diseñan, desarrollan y mantienen productos, incluidos los programas informáticos.
- ✧ La norma ISO 9001 es un marco para el desarrollo de normas de software.
 - En él se establecen los principios generales de calidad, se describen los procesos de calidad en general y se exponen las normas y procedimientos de organización que deben definirse. Éstos deben documentarse en un manual de calidad de la organización.

Procesos básicos de la ISO 9001



ISO 9001 y gestión de la calidad



Certificación ISO 9001



- ✧ Las normas y procedimientos de calidad deben documentarse en un manual de calidad de la organización.
- ✧ Un organismo externo puede certificar que el manual de calidad de una organización se ajusta a las normas ISO 9000.
- ✧ Algunos clientes exigen que los proveedores tengan la certificación ISO 9000, aunque cada vez se reconoce más la necesidad de flexibilidad.

Puntos clave



- ✧ La gestión de la calidad de los programas informáticos se ocupa de garantizar que los programas informáticos tengan un número reducido de defectos y que alcancen los niveles requeridos de mantenimiento, fiabilidad, portabilidad, etc.
- ✧ La SQM incluye la definición de normas para procesos y productos y el establecimiento de procesos para comprobar que se han seguido estas normas.
- ✧ Las normas de software son importantes para la garantía de calidad, ya que representan una identificación de las "mejores prácticas".
- ✧ Los procedimientos de gestión de la calidad pueden documentarse en un manual de calidad de la organización, basado en el modelo genérico de manual de calidad sugerido en la norma ISO 9001.

Capítulo 24 Gestión de la calidad

25

Revisiones e inspecciones



- ✧ Un grupo examina una parte o la totalidad de un proceso o sistema y su documentación para encontrar posibles problemas.
- ✧ El software o los documentos pueden ser "firmados" en un que significa que el progreso hacia el próximo La fase de desarrollo ha sido aprobada por de la gestión.
- ✧ Hay diferentes tipos de examen con diferentes objetivos
 - Inspecciones para la eliminación de defectos (producto);
 - Exámenes para la evaluación de los progresos (producto y proceso);
 - Revisiones de calidad (producto y normas).

Capítulo 24 Gestión de la calidad

27

El proceso de revisión del software



Capítulo 24 Gestión de la calidad

29

Capítulo 24 - Gestión de la calidad

Conferencia 2

Capítulo 24 Gestión de la calidad

26

Revisiones de calidad



- ✧ Un grupo de personas examina cuidadosamente una parte o la totalidad de un sistema de software y su asociado documentación.
- ✧ Código, diseños, especificaciones, planes de prueba, las normas, etc. pueden ser revisadas.
- ✧ El software o los documentos pueden ser "firmados" en un que significa que el progreso hacia el próximo La fase de desarrollo ha sido aprobada por de la gestión.

Capítulo 24 Gestión de la calidad

28

Reseñas y métodos ágiles



- ✧ El proceso de examen en el desarrollo de software ágil suele ser informal.
 - En Scrum, por ejemplo, hay una reunión de revisión después de cada iteración del software (una revisión de sprint), en la que se pueden discutir cuestiones y problemas de calidad.
- ✧ En la programación extrema, la programación en parejas asegura que el código sea constantemente examinado y revisado por otro miembro del equipo.
- ✧ El XP depende de que los individuos tomen la iniciativa para mejorar y refactorizar el código. Los enfoques ágiles no suelen estar basados en normas, por lo que no suelen considerarse cuestiones de cumplimiento de normas.

Capítulo 24 Gestión de la calidad

30

Inspecciones del programa



- ✧ Se trata de revisiones por pares en las que los ingenieros examinan la fuente de un sistema con el objetivo de descubrir anomalías y defectos.
- ✧ Las inspecciones no requieren la ejecución de un sistema, por lo que pueden utilizarse antes de su aplicación.
- ✧ Pueden aplicarse a cualquier representación del sistema (requisitos, diseño, datos de configuración, datos de prueba, etc.).
- ✧ Se ha demostrado que son una técnica efectiva para descubrir errores de programa.

Listas de control de inspección



- ✧ La lista de errores comunes debe utilizarse para conducir la inspección.
- ✧ Las listas de control de errores son un lenguaje de programación dependientes y reflejan los errores característicos que probablemente se produzcan en el idioma.
- ✧ En general, cuanto más "débil" sea la comprobación del tipo, más grande será la lista de comprobación.
- ✧ Ejemplos: Inicialización, Nombramiento constante, bucle terminación, límites de la matriz, etc.

Una lista de verificación de la inspección (a)



Clase de falla	Comprobación de inspección
Fallas de datos	<ul style="list-style-type: none">• ¿Todas las variables del programa se inicializan antes de que se utilicen sus valores?• ¿Se han nombrado todas las constantes?• ¿Debería el límite superior de los conjuntos ser igual al tamaño del conjunto o al tamaño -1?• Si se utilizan cadenas de caracteres, ¿se asigna explícitamente un delimitador?• ¿Hay alguna posibilidad de desbordamiento del buffer?
Fallas de control	<ul style="list-style-type: none">• Para cada declaración condicional, ¿es la condición correcta?• ¿Es seguro que cada bucle termine?• ¿Las declaraciones compuestas están correctamente entre corchetes?• En las declaraciones de casos, ¿se contabilizan todos los casos posibles?• Si se requiere una pausa después de cada caso en las declaraciones del caso, ¿se ha incluido?
Fallas de	<ul style="list-style-type: none">• ¿Se utilizan todas las variables de entrada?

Una lista de verificación de la inspección b)



Clase de falla	Comprobación de inspección
Fallas de la interfaz	<ul style="list-style-type: none">• ¿Todas las llamadas de funciones y métodos tienen el número correcto de parámetros?• ¿Coinciden los tipos de parámetros formales y reales?• ¿Están los parámetros en el orden correcto?• Si los componentes acceden a la memoria compartida, ¿tienen el mismo modelo de la estructura de la memoria compartida?
Fallas en la gestión del almacenamiento	<ul style="list-style-type: none">• Si se modifica una estructura de enlace, ¿se han reasignado correctamente todos los enlaces?• Si se utiliza el almacenamiento dinámico, ¿se ha asignado correctamente el espacio?• ¿El espacio es explícitamente repartido después de que ya no es necesario?
Fallas en la gestión de excepciones	<ul style="list-style-type: none">• ¿Se han tenido en cuenta todas las posibles condiciones de error?

Métodos ágiles e inspecciones



- ✧ Los procesos ágiles rara vez utilizan procesos formales de inspección o de examen por homólogos.
- ✧ Más bien, confían en que los miembros del equipo cooperen para comprobar el código de cada uno, y en directrices informales, como "comprobar antes de comprobar", que sugieren que los programadores deben comprobar su propio código.
- ✧ Los profesionales de la programación extrema sostienen que la programación por pares es un sustituto eficaz de la inspección, ya que se trata, en efecto, de un proceso de inspección continua.
- ✧ Dos personas miran cada línea de código y la comprueban antes de que sea aceptada.

Software de medición y métrica



- ✧ La medición de software se ocupa de derivar un valor numérico para un atributo de un producto o proceso de software.
- ✧ Esto permite hacer comparaciones objetivas entre técnicas y procesos.
- ✧ Aunque algunas empresas han introducido programas de medición, la mayoría de las organizaciones todavía no hacen un uso sistemático de la medición por software.
- ✧ Hay pocas normas establecidas en esta área.

La métrica del software

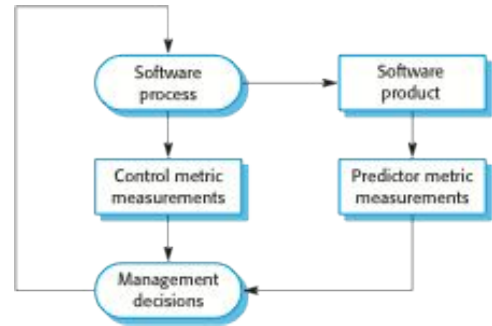


- ❖ Cualquier tipo de medición que se relacione con un sistema de software, proceso o documentación relacionada
 - Líneas de código en un programa, el índice de niebla, número de días-persona necesarios para desarrollar un componente.
- ❖ Permitir que el software y el proceso de software ser cuantificado.
- ❖ Puede utilizarse para predecir los atributos del producto o para controlar el proceso del software.
- ❖ Las métricas del producto pueden utilizarse para las predicciones generales o para identificar componentes anómalos.

Capítulo 24 Gestión de la calidad

37

Medidas de predicción y control



Capítulo 24 Gestión de la calidad

38

Uso de las mediciones



- ❖ Para asignar un valor a los atributos de calidad del sistema
 - Midiendo las características de los componentes del sistema, como su complejidad ciclomática, y luego agregando estas mediciones, se pueden evaluar los atributos de calidad del sistema, como la mantenibilidad.
- ❖ Identificar los componentes del sistema cuya calidad es inferior a la norma
 - Las mediciones pueden identificar componentes individuales con características que se desvían de la norma. Por ejemplo, se pueden medir los componentes para descubrir aquellos de mayor complejidad. Es más probable que estos contengan errores porque la complejidad los hace más difíciles de entender.

Capítulo 24 Gestión de la calidad

39

Supuestos de la métrica

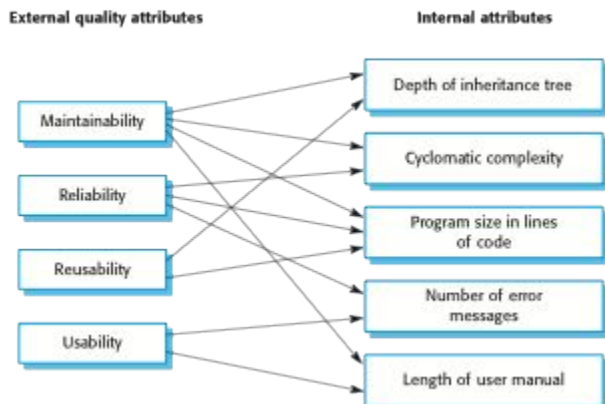


- ❖ Una propiedad del software puede ser medida.
- ❖ La relación existe entre lo que podemos medir y lo que queremos saber. Sólo podemos medir los atributos internos, pero a menudo estamos más interesados en los atributos externos del software.
- ❖ Esta relación se ha formalizado y validado.
- ❖ Puede ser difícil relacionar lo que puede medirse con los atributos de calidad externos deseables.

Capítulo 24 Gestión de la calidad

40

Las relaciones entre el software interno y el externo



Capítulo 24 Gestión de la calidad

41

Problemas con la medición en la industria



- ❖ Es imposible cuantificar el retorno de la inversión de la introducción de un programa de métrica organizativa.
- ❖ No existen normas para la métrica del software o procesos estandarizados para la medición y el análisis.
- ❖ En muchas empresas, los procesos de software no están estandarizados y están mal definidos y controlados.
- ❖ La mayoría de los trabajos sobre medición de software se han centrado en métricas basadas en códigos y en procesos de desarrollo basados en planes. Sin embargo, cada vez más software se desarrolla ahora mediante la configuración de sistemas ERP o COTS.
- ❖ La introducción de la medición añade una sobrecarga adicional a los procesos.

Capítulo 24 Gestión de la calidad

42

Métricas del producto



- Una métrica de calidad debería ser un predictor de la calidad del producto.
- Clases de productos métricos
 - Métricas dinámicas que son recogidas por medidas hechas de un programa en ejecución;
 - Métricas estáticas que son recogidas por mediciones hechas de las representaciones del sistema;
 - La métrica dinámica ayuda a evaluar la eficiencia y la fiabilidad
 - La métrica estática ayuda a evaluar la complejidad, la comprensibilidad y la capacidad de mantenimiento.

Métricas dinámicas y estáticas



- La métrica dinámica está estrechamente relacionada con los atributos de calidad del software
 - Es relativamente fácil medir el tiempo de respuesta de un sistema (atributo de rendimiento) o el número de fallos (atributo de fiabilidad).
- La métrica estática tiene una relación indirecta con los atributos de calidad
 - Es necesario tratar de derivar una relación entre estas métricas y propiedades tales como la complejidad, la comprensibilidad y la mantenibilidad.

Métricas del producto de software estático



La métrica del software	Descripción
Fan-in/Fan-out	Fan-in es una medida del número de funciones o métodos que llaman a otra función o método (digamos X). Fan-out es el número de funciones que son llamadas por la función X. Un valor alto de fan-in significa que X está estrechamente acoplado al resto del diseño y los cambios en X tendrán amplios efectos de repercusión. Un valor alto de fan-out sugiere que la complejidad general de X puede ser alta debido a la complejidad de la lógica de control necesaria para coordinar los componentes llamados.
Longitud del código	Esta es una medida del tamaño de un programa. Generalmente, cuanto mayor sea el tamaño del código de un componente, más complejo y propenso a errores es ese componente. Se ha demostrado que la longitud del código es una de las métricas más fiables para predecir la propensión al error en los componentes.

Métricas del producto de software estático



La métrica del software	Descripción
Complejidad ciclomática	Esta es una medida de la complejidad del control de un programa. Esta complejidad de control puede estar relacionada con la comprensibilidad del programa. Discuto la complejidad ciclomática en el capítulo 8.
Longitud de los identificadores	Es una medida de la longitud media de los identificadores (nombres de las variables, clases, métodos, etc.) en un programa. Cuanto más largos sean los identificadores, más probable es que sean significativos y, por lo tanto, más comprensibles para el programa.
Profundidad del anidamiento condicional	Esta es una medida de la profundidad de anidación de los si-declaraciones en un programa. Los enunciados "si" anidados profundamente son difíciles de entender y potencialmente propensos a errores.
Índice de niebla	Es una medida de la longitud media de las palabras y frases de los documentos. Cuanto más alto es el valor del índice de niebla de un documento, más difícil es de

El conjunto de métricas orientadas a objetos de CK



La métrica orientada al objeto	Descripción
Métodos ponderados por clase (WMC)	Es el número de métodos de cada clase, ponderado por la complejidad de cada uno de ellos. Por lo tanto, un método simple puede tener una complejidad de 1, y un método grande y complejo un valor mucho más alto. Cuanto mayor sea el valor de esta métrica, más compleja será la clase de objeto. Es más probable que los objetos complejos sean más difíciles de entender. Pueden no estar lógicamente cohesivos, por lo que no pueden ser reutilizados eficazmente como superclases en un árbol de herencia.
Profundidad del árbol genealógico (DIT)	Esto representa el número de niveles discretos en el árbol de la herencia donde las subclases heredan atributos y operaciones (métodos) de las superclases. Cuanto más profundo es el árbol de la herencia, más complejo es el diseño. Es posible que haya que comprender muchas clases de objetos para entender las clases de objetos en las hojas del árbol.
Número de niños (NOC)	Es una medida del número de subclases inmediatas de una clase. Mide la amplitud de la jerarquía de una clase, mientras que el DIT mide su profundidad. Un valor alto para NOC puede indicar una mayor reutilización. Puede significar que debe hacerse un mayor esfuerzo en la validación de las clases base debido al número de subclases que dependen de ellas.

El conjunto de métricas orientadas a objetos de CK



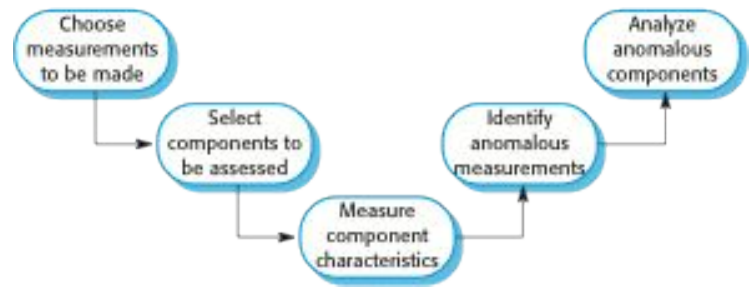
La métrica orientada al objeto	Descripción
Acoplamiento entre clases de objetos (CBO)	Las clases se acoplan cuando los métodos de una clase utilizan métodos o variables de instancia definidos en una clase diferente. El CBO es una medida de cuánto acoplamiento existe. Un valor alto de CBO significa que las clases son altamente dependientes, y por lo tanto es más probable que el cambio de una clase afecte a otras clases del programa.
Respuesta para una clase (RFC)	RFC es una medida del número de métodos que potencialmente podrían ejecutarse en respuesta a un mensaje recibido por un objeto de esa clase. De nuevo, RFC está relacionado con la complejidad. Cuanto más alto es el valor de RFC, más compleja es una clase y por lo tanto es más probable que incluya errores.
Falta de cohesión en los métodos (LCOM)	El LCOM se calcula considerando los pares de métodos de una clase. LCOM es la diferencia entre el número de pares de métodos sin atributos compartidos y el número de pares de métodos con atributos compartidos. El valor de esta métrica ha sido ampliamente debatido y existe en varias variaciones. No está claro si realmente añade alguna información adicional y útil además de la proporcionada por otras métricas.

Análisis de componentes de software



- ✧ El componente del sistema puede ser analizado por separado utilizando una serie de métricas.
- ✧ Los valores de estas métricas pueden entonces compararse para diferentes componentes y, tal vez, con los datos de medición histórica recogidos en proyectos anteriores.
- ✧ Las mediciones anómalas, que se desvían significativamente de la norma, pueden implicar que hay problemas con la calidad de estos componentes.

El proceso de medición del producto



Sorpresas de medición



- ✧ Reducir el número de fallas en un programa lleva a un mayor número de llamadas a la mesa de ayuda
 - El programa se considera ahora más fiable y por lo tanto tiene un mercado más amplio y diverso. El porcentaje de usuarios que llaman al servicio de asistencia puede haber disminuido, pero el total puede aumentar;
 - Un sistema más fiable se utiliza de manera diferente a un sistema en el que los usuarios trabajan en torno a las fallas. Esto lleva a más llamadas a la mesa de ayuda.

Puntos clave



- ✧ En las revisiones de los entregables del proceso de software participa un equipo de personas que comprueban que se siguen los estándares de calidad.
- ✧ En una inspección del programa o en una revisión por pares, un pequeño equipo comprueba sistemáticamente el código. Leen el código en detalle y buscan posibles errores y omisiones
- ✧ La medición de software puede utilizarse para reunir datos sobre el software y los procesos de software.
- ✧ Las métricas de calidad de los productos son particularmente útiles para destacar los componentes anómalos que pueden tener problemas de calidad.