

# ¿Cómo detectar errores y evitar su amplificación?

Linda María Fernández Olvera  
UPIICSA-IPN  
Ingeniería de Diseño  
lfernandezol1500@alumno.ipn.mx

1

**Abstract – Un error no identificado fácilmente podría complicar su corrección a largo plazo, porque generaría amplificación y con ello costos elevados para su solución. A través de procesos de revisión es posible identificarlos y tratarlos.**

**Índice de Términos – errores, métricas, revisión, defectos, modelo de referencia, costos.**

## I. INTRODUCCIÓN

Las revisiones del software son un “filtro” para el proceso del software. Se aplican en varios puntos durante la ingeniería de software y sirven para descubrir errores y defectos a fin de poder eliminarlos.

Si bien las personas son buenas para detectar algunos de sus propios errores, muchas clases de ellos pasan desapercibidos con más facilidad para quien los comete que para otras personas.

Detectar errores a través de una revisión técnica al principio del proceso, es menos caro corregirlo, que al final darse cuenta y que este se amplificó y se volvió un conjunto de errores a gran escala, teniendo que repetir el proceso de desarrollo.

## II. DESARROLLO

Los términos defecto y falla implican un problema de calidad descubierto después de haberse liberado el software a los usuarios finales. Los errores se detectan antes de que el software se entregue a los usuarios finales.

El objetivo principal de las revisiones técnicas es encontrar errores durante el proceso a fin de que no se conviertan en defectos después de liberar el software. El beneficio obvio de las revisiones técnicas es el descubrimiento temprano de los errores, de modo que no se propaguen a la siguiente etapa del proceso del software.

### A. Amplificación y eliminación del defecto

Para la generación y detección de errores durante las acciones de diseño y generación de código de un proceso de

software, puede usarse un modelo de amplificación del defecto, como el que se muestra en la Figura 1. Durante la acción, los errores se generan de manera inadvertida. La revisión puede fracasar en descubrir los errores nuevos que se generan y los cometidos en etapas anteriores, lo que da como resultado cierto número de errores pasados por alto.

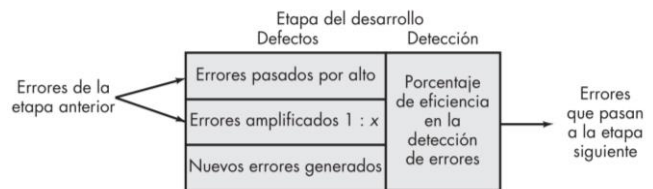


Figura 1. Modelo de amplificación del defecto

Debe dedicarse tiempo y esfuerzo a la realización de revisiones y su organización de desarrollo debe destinar el dinero para ello. Esto deja pocas dudas acerca de lo que puede pagar ahora o de que después deberá pagar mucho más.

### B. Métricas de Revisión y su empleo

Las revisiones técnicas son una de las muchas acciones que se requieren como parte de las buenas prácticas de la ingeniería de software. Cada acción requiere un esfuerzo humano dirigido. Como el esfuerzo disponible para el proyecto es finito, es importante que una organización de software comprenda la eficacia de cada acción, definiendo un conjunto de métricas que puedan utilizarse para evaluar esa eficacia.

Las siguientes métricas para la revisión pueden obtenerse conforme se efectúe ésta:

- Esfuerzo de preparación,  $E_p$ : esfuerzo (en horas-hombre) requerido para revisar un producto del trabajo antes de la reunión de revisión real.
- Esfuerzo de evaluación,  $E_a$ : esfuerzo requerido (en horas-hombre) que se dedica a la revisión real.
- Esfuerzo de la repetición,  $E_r$ : esfuerzo (en horas-hombre) que se dedica a la corrección de los errores descubiertos durante la revisión.
- Tamaño del producto del trabajo, TPT: medición del tamaño del producto del trabajo que se ha revisado (por ejemplo, número de modelos UML o número de

páginas de documento o de líneas de código).

- Errores menores detectados,  $Err_{menores}$ : número de errores detectados que pueden clasificarse como menores (requieren menos de algún esfuerzo especificado para corregirse).
- Errores mayores detectados,  $Err_{mayores}$ : número de errores encontrados que pueden clasificarse como mayores (requieren más que algún esfuerzo especificado para corregirse).

Estas métricas pueden mejorarse, asociando el tipo de producto del trabajo que se revisó con las métricas obtenidas.

### Análisis de las métricas

Antes de comenzar el análisis deben hacerse algunos cálculos sencillos. El esfuerzo total de revisión y el número total de errores descubiertos se definen como sigue:

$$E_{revisión} = E_p + E_d + E_r$$

$$Err_{tot} = Err_{menores} + Err_{mayores}$$

La densidad del error representa los errores encontrados por unidad de producto del trabajo revisada.

$$\text{Densidad del error} = \frac{Err_{tot}}{TPT}$$

Si las revisiones se llevan a cabo para varios tipos distintos de productos del porcentaje de errores no descubiertos por cada revisión se confronta con el número total de errores detectados en todas las revisiones. Además, puede calcularse la densidad del error para cada producto del trabajo.

Una vez recabados los datos para muchas revisiones efectuadas en muchos proyectos, los valores promedio de la densidad del error permiten estimar el número de errores por hallar en un nuevo documento.

### Eficacia del costo de las revisiones

Es difícil medir en tiempo real la eficacia del costo de cualquier revisión técnica. Una organización de ingeniería de software puede evaluar la eficacia de las revisiones y su relación costo beneficio sólo después de que éstas han terminado, de que las unidades de medida de la revisión se han recabado, de que los datos promedio han sido calculados y de que la calidad posterior del software ha sido medida.

Las revisiones no quitan tiempo, lo ahorran.

### C. Revisiones: Espectro de formalidad

Las revisiones técnicas deben aplicarse con un nivel de formalidad apropiado para el producto que se va a elaborar, para el plazo que tiene el proyecto y para el personal que realice el trabajo.

La Figura 2, ilustra un modelo de referencia para las revisiones técnicas que identifica cuatro características que contribuyen a la formalidad con la que se efectúa una revisión.

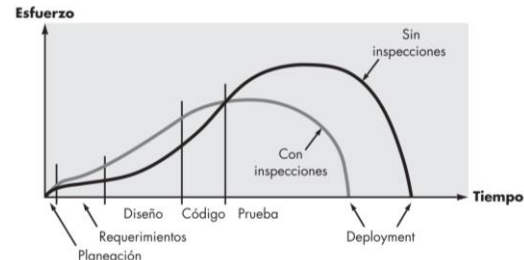


Figura 2. Esfuerzo realizado, con y sin revisiones

Cada una de las características del modelo de referencia ayuda a definir el nivel de formalidad de la revisión. La formalidad de una revisión se incrementa cuando:

1. Se definen explícitamente roles distintos para los revisores.
2. Hay suficiente cantidad de planeación y preparación para la revisión.
3. Se define una estructura distinta para la revisión (incluso tareas y productos internos del trabajo).
4. El seguimiento por parte de los revisores tiene lugar para cualesquiera correcciones que se efectúen.

Los resultados de ésta serán registrados de manera formal y el equipo decidirá sobre el estado del producto del trabajo con base en el resultado de la revisión. Los miembros del equipo también verificarán que las correcciones se hagan de manera adecuada.

### D. Revisiones informales

Las revisiones informales incluyen una simple verificación de escritorio de un trabajo de ingeniería de software, hecha con algún colega, o una reunión casual (con más de dos personas) con objeto de revisar un producto o aspectos orientados a la revisión de programación por pares.

Una verificación de escritorio simple o una reunión casual realizada con un colega constituye una revisión. Sin embargo, como no hay una planeación o preparación por adelantado, ni agenda o estructura de la reunión, y no se da seguimiento a los errores descubiertos, la eficacia de tales revisiones es mucho menor que la de los enfoques más formales. Pero una verificación de escritorio sencilla descubre errores que de otro modo se pagarían en el proceso del software.

La programación por pares se caracteriza por una verificación de escritorio continua. En vez de programar una revisión en algún momento dado, la programación por pares

invita a hacer una revisión continua a medida que se crea el producto (diseño o código). El beneficio es el inmediato descubrimiento de los errores y, en consecuencia, la mejora de la calidad del producto.

### *E. Revisiones técnicas formales*

Una revisión técnica formal (RTF) es una actividad del control de calidad del software realizada por ingenieros de software (y otras personas). Los objetivos de una RTF son:

1. Descubrir los errores en funcionamiento, lógica o implementación de cualquier representación del software.
2. Verificar que el software que se revisa cumple sus requerimientos.
3. Garantizar que el software está representado de acuerdo con estándares predefinidos.
4. Obtener software desarrollado de manera uniforme.
5. Hacer proyectos más manejables.

Sirve como método de capacitación, pues permite que los ingenieros principiantes observen distintos enfoques de análisis, diseño e implementación del software. La RTF también funciona para estimular el respaldo y la continuidad debido a que varias personas se familiarizan con software que de otra manera no hubieran visto.

Cada RTF se realiza como una reunión y tendrá éxito sólo si se planea, controla y ejecuta en forma apropiada. En las secciones que siguen se presentan lineamientos similares a aquellos usados para un walkthrough, como representativos de la revisión técnica formal. Si el lector tiene interés en las inspecciones de software y en obtener más información sobre walkthroughs.

Sin importar cuál formato de RTF se elija, cualquiera de ellos debe cumplir las restricciones siguientes:

- En la revisión deben involucrarse de tres a cinco personas (normalmente).
- Debe haber preparación previa, pero no debe exigir más de dos horas de trabajo de cada persona.
- La duración de la reunión de revisión debe ser de al menos dos horas.

Dadas estas restricciones, debe resultar obvio que una RTF se centra en una parte específica (y pequeña) del software general. Por ejemplo, en vez de tratar de revisar todo el diseño, se hacen walkthroughs para cada componente o grupo pequeño de componentes. Al reducir el alcance, la RTF tiene mayor probabilidad de detectar errores.

A la reunión de revisión acuden el líder de ésta, todos los revisores y el productor. Uno de los revisores adopta el rol de secretario, es decir, quien registra (por escrito) todos los acontecimientos importantes que surjan durante la revisión. La RTF comienza con el análisis de la agenda y una

introducción breve por parte del productor. Después, éste procede a “recorrer” el producto del trabajo, explicando el material, mientras los revisores hacen sus comentarios con base en la preparación que hicieron. Cuando se descubren problemas o errores válidos, el secretario toma nota de ellos.

Al terminar la revisión, todos los asistentes deben decidir si:

- a) Aceptan el producto sin modificaciones.
- b) Lo rechazan debido a errores graves (una vez corregidos, se realiza otra revisión).
- c) Aceptan el producto de manera provisional (se encontraron errores menores que deben corregirse, pero no se necesita otra revisión).

Una vez tomada la decisión, todos los asistentes a la RTF firman el acta que indica su participación y su acuerdo con los descubrimientos del equipo de revisión.

### *Reporte y registro de la revisión*

Durante la RTF, un revisor (el secretario) registra activamente todos los asuntos que se planteen.

Éstos se resumen al final de la reunión y se produce la lista de pendientes de la revisión. Además, se elabora un reporte técnico formal de la revisión. Éste responde tres preguntas:

1. ¿Qué fue lo que se revisó?
2. ¿Quién lo revisó?
3. ¿Cuáles fueron los descubrimientos y las conclusiones?

El resumen del reporte de la revisión es una sola página (quizá con anexos) que se vuelve parte del registro histórico del proyecto y se entrega al líder del proyecto y a otras partes interesadas.

### *Lineamientos para la revisión*

Los lineamientos para efectuar revisiones técnicas formales deben establecerse por adelantado, distribuirse a todos los revisores, llegar al consenso y, después, seguirse. Una revisión sin control con frecuencia es peor que si no se hiciera ninguna. Los siguientes representan un conjunto mínimo de lineamientos para hacer revisiones técnicas formales:

1. Revise el producto, no al productor. Una RTF involucra personas y sus egos. Si se lleva a cabo en forma adecuada, la RTF debe dejar en todos los participantes una sensación de logro. Si se efectúa de modo inapropiado, adopta un aire inquisitorial. Los errores deben señalarse en forma amable.

2. Establezca una agenda y sígala. Una RTF debe mantenerse encarrilada y dentro del programa. El líder de la revisión tiene la responsabilidad de que así sea y no debe sentir temor de llamar al orden.

3. Limite el debate y las contestaciones. Cuando el revisor

plantee un asunto, quizá no haya acuerdo universal acerca de su efecto. En vez de perder tiempo en debatir la cuestión, ésta debe registrarse para discutirla después.

4. Enuncie áreas de problemas, pero no intente resolver cada uno. Una revisión no es una sesión para resolver problemas. Es frecuente que la solución de un problema la obtenga el productor, solo o con ayuda de otra persona. La solución de los problemas debe posponerse para después de la reunión de revisión.

5. Tome notas por escrito. A veces es buena idea que el secretario tome notas en un pizarrón a fin de que la redacción y prioridades sean evaluadas por los demás revisores a medida que la información se registra.

6. Limite el número de participantes e insista en la preparación previa. Mantenga limitado el número de personas involucradas. Sin embargo, todos los miembros del equipo de revisión deben prepararse.

7. Desarrolle una lista de verificación para cada producto que sea probable que se revise. Una lista de verificación ayuda al líder del proyecto a estructurar la RTF y a cada revisor a centrarse en los aspectos importantes. Deben desarrollarse listas para los productos del análisis, el diseño, el código e incluso las pruebas.

8. Asigne recursos y programe tiempo para las RTF. Para que las revisiones sean eficaces, deben programarse como tareas del proceso de software.

9. Dé una capacitación significativa a todos los revisores. Para que una revisión sea eficaz, todos los revisores deben recibir cierta capacitación formal.

10. Revise las primeras revisiones. Volver a revisar puede ser benéfico para descubrir problemas con el proceso de revisión en sí mismo. El primer producto por revisar deben ser los lineamientos de la revisión.

#### *Revisiones orientadas al muestreo*

En el mundo real de los proyectos de software, los recursos son limitados y el tiempo, escaso. En consecuencia, es frecuente que las revisiones se omitan, aun cuando se reconozca su valor como un mecanismo de control de calidad.

Para que sea eficaz, el proceso de revisión orientada al muestreo debe tratar de identificar aquellos productos del trabajo que sean objetivos principales para hacer la RTF.

### III. CONCLUSIÓN

El objetivo de toda revisión técnica es detectar errores y descubrir aspectos que tendrían un efecto negativo en el software que se va a desarrollar. Entre más pronto se descubra

y corrija un error, menos probable es que se propague a otros productos del trabajo de la ingeniería de software y que se amplifique, lo que provocaría un mayor esfuerzo para corregirlo.

Un modelo de referencia para la formalidad de la revisión identifica roles de las personas, planeación y preparación, estructura de la reunión, enfoque de corrección y verificación como las características que indican el grado de formalidad con el que se realiza una revisión. Las revisiones informales son de naturaleza casual, pero pueden usarse con eficacia para detectar errores. Las revisiones formales son más estructuradas y tienen una probabilidad mayor de dar como resultado un software de alta calidad.

Las revisiones informales se caracterizan por tener una planeación y preparación mínimas y poco registro de su desarrollo.

Una revisión técnica formal es una reunión estilizada que ha demostrado ser extremadamente eficaz para detectar errores. Los walkthroughs y las inspecciones establecen roles definidos para cada revisor, estimulan la planeación y la preparación previa, requieren la aplicación de lineamientos de revisión definidos y ordenan llevar registros y hacer reportes.

#### REFERENCIAS

- [1] Roger S. Pressman, "Ingeniería del Software, un enfoque práctico" 7ª ed. Connecticut University, New York: McGraw-Hill, 2010, pp. 353 – 365.