CC51A – Ingeniería de Software

SQA (Software Quality Assurance)

Sergio Ochoa D.

Parte de este material ha sido cedido por el Prof. Jaime Navón, PhD.
(www.ing.puc.cl/~jnavon)

Estructura de la Presentación

- Introducción a SQA
- El SQA Group
- Revisiones Técnicas Formales
- Algunas Directrices ...
- Ventajas y Desventajas
- Datos de la Realidad ...

El Grupo SQA

- Sirve como el representante del cliente (mira el producto desde el punto de vista del cliente).
- Asiste al grupo de desarrollo para lograr un producto final de “buena calidad” (que cumple con una lista de requisitos más exigente).
- Trata de responder preguntas del tipo:
  - Se satisfacen los criterios de calidad especificados?
  - Se ha desarrollado en base a los estándares establecidos?
- Coordina el manejo de los cambios (control de configuración).
- Identifica riesgos en forma temprana.
- Se preocupa de la estandarización de procesos y productos.

SQA (Aseguramiento de la Calidad)

Calidad de Software: Cumplimiento de los requisitos explícitos: funcionales, de calidad y de restricción.

Observaciones:

- Los requisitos son la base sobre la cual se juzga la calidad de un producto.
- Si no se siguen los criterios de desarrollo definidos por los estándares usados, muy probablemente no se conseguirán buenos productos.
- Hay además requisitos implícitos que a menudo no se mencionan (fácil de mantener por ejemplo) que también es deseable respetar.

Actividades del SQA Group

Preparar el plan de SQA (evaluaciones, auditorías, estándares, procedimientos de reporte de errores, templates de documentos, casos de prueba, estandarización de procedimientos, etc).

- El plan se prepara durante la etapa de planeación del proyecto.
- El plan es revisado por todas las partes interesadas.
- El plan es la base de todas las actividades de SQA.

Participar en el diseño del desarrollo del proceso específico a ser usado en el proyecto (puede chequear cumplimiento de estándares externos como ISO 9001, etc)

- Revisar que las actividades se lleven a cabo de acuerdo al proceso descrito (identifica, documenta y monitorea las desviaciones).

Actividades del SQA Group

Participar en el diseño del desarrollo del proceso específico a ser usado en el proyecto (puede chequear cumplimiento de estándares externos como ISO 9001, etc)

- Auditar ciertos productos del proceso.
- Asegurar que las desviaciones en los productos sean manejadas y documentadas de acuerdo a un procedimiento.
- Registrar cualquier ítem insatisfactorio y reportar a la administración superior.

Actividades del SQA Group

Definición de los casos de prueba.

- En base a los requisitos de usuario, el SQA group debe generar los casos de prueba, para las pruebas a nivel de usuario (pruebas de usuario).
- En base a los requisitos de software, el SQA group debe generar el resto de los casos de prueba: prueba de módulos, de integración y de sistema (al menos).

Ejecución de los casos de prueba.

- Durante el período de pruebas, el SQA Group debe coordinar las pruebas y revisar el cumplimiento de los requisitos definidos.
- Reportar anomalías detectadas durante las pruebas.

Revisiones Técnicas Formales

Las revisiones constituyen una de las actividades mas importantes de SQA.

Propósito:

- Descubrir errores (función, lógica o implementación).
- Verificar que el software satisface los requisitos.
- Asegurar que se cumplen los estándares predefinidos.
- Lograr software desarrollado en forma uniforme.
- Hacer más manejable los proyectos.

Se eliminan errores en forma relativamente temprana (barato y fácil de corregir).

Revisiones Técnicas Formales

Puede hacerse con:

- Documentos de diseño.
- Código fuente (listo para ser compilado).
- Planificaciones.

Cada revisión se conduce en forma de una reunión cuidadosamente planeada y controlada.

Hay muchos tipos de revisiones... Algunos de ellos los vimos en la clase de “Software Technical Reviews”.

Revisiones Técnicas Formales

- Participan entre 3 y 5 personas, y requiere reparación previa.
- Foco en un segmento específico pequeño (1 módulo o documento).
- Participan los revisores (uno actúa de jefe) y el autor, y el moderador.

Precondiciones:

- Especificación precisa (formal o informal) sobre que es lo que ese código se supone que debe hacer.
- Miembros del grupo deben conocer estándares de codificación.
- No debe usarse para evaluación del personal.
- Dispuesto a aceptar costos extra (ahorro es en largo plazo).

Algunas Directrices…

- Revisar el producto y no al productor
  - Indicar los errores con tino.
  - Hablar en tono constructivo.
  - La idea no es avergonzar al productor.
- Mantenerse estrictamente dentro de la agenda
  - No irse por las ramas.
  - El moderador es el responsable.
  - Limitar el debate. Algunos asuntos pueden dejarse para discusión posterior.
- Enunciar problemas pero no resolverlos
  - El problema debería ser resuelto por el autor.
  - No es una sesión de resolución de problemas.

Algunas Directrices… (cont.)

- Tomar notas y/o grabar la revisión.
- Limitar número de participantes e insistir en la preparación previa.
  - Dos cabezas piensan más que una, pero 14 no necesariamente más que 4.
  - El moderador puede solicitar comentarios por escrito antes de la reunión para asegurar que la preparación de los miembros es la adecuada.
- Desarrollar previamente “checklists” por cada producto a revisar (para mantener el foco de la revisión).
- Asignar recursos y tiempos para estas reuniones
  - No olvidar que es parte del proceso!
  - Proporcionar entrenamiento para los revisores.
  - Es posible incluso revisar (auditar) las revisiones.

Ventajas...

- Se considera más efectivo que las pruebas para encontrar bugs (encuentra la causa del error en lugar del síntoma).
- Los programadores escriben sus programas sabiendo que otros los revisarán => fácil de entender, leer, y mantener.
- Los programadores aprenden de las revisiones, cómo hacer los programas más fáciles de entender.
- Se anula el efecto de “puntos ciegos” (programador es incapaz de ver ciertos errores).
- Es posible imponer estándares de codificación con facilidad.
- Explicar el código hace que el programador lo entienda cada vez mejor (los profesores saben esto).
- Reduce dramáticamente tiempo de pruebas.

Desventajas…

- Problemas de personalidad.
  - Puede haber personas con buenas ideas no se expresan.
  - Puede haber personas con malas ideas se expresan mucho.
  - Algunas personas odian discutir o estar en desacuerdo.
  - Es fácil perderse en cosas triviales (punto, coma, etc.).
- Es agotador (pierde efectividad después de un par de horas o menos).

Funciona en la Práctica

- Bell Northern Research (subsidiaria de Northern Telecom, Compañía de Telecomunicaciones de Canadá)
  - 10 MLOC switching software
  - Se inspeccionaron 2.5 MLOC a lo largo de 2 años.
  - Se encontraron alrededor de 0.8 a 1.0 errores por hora-hombre.
  - Se encontraron alrededor de 37 errores por cada 1000 LOC (estudios dan 50 a 75).
  - 2.5 MLOC => 2500 x 37 = 92500 errores.
  - Se considera que un error detectado en etapa de testing puede tomar hasta 33 horas-hombre para diagnosticar y reparar.
  - Ahorro de tiempo es enorme !!

Funciona en la Práctica

Bell Northern Research (subsidiaria de Northern Telecom, Compañía de Telecomunicaciones de Canadá)

- En la actualidad se está intentado incorporar la Web para hacer las revisiones:
  - Se publica el código.
  - Los revisores pueden estar distribuidos geográficamente.
  - Se pueden realizar revisiones asíncronas o síncronas dependiendo de las herramientas disponibles.

Control Estadístico de la Calidad

- Idea básica es gastar el tiempo en las cosas que son realmente importantes. Debemos asegurarnos de determinar, qué es lo realmente importante?.
  - El número de causas de la gran mayoría de los errores es relativamente limitado.
  - La información sobre defectos es recogida y categorizada.
  - Se intenta asociar cada defecto a la causa de origen (especificación, diseño, violación de estándar, etc.).
- Basándose en el principio de Pareto (20% de las causas producen el 80% de los errores) se aísla el 20% vital.
- Una vez que las principales causas se identifican, se corrigen los problemas que causaron los defectos.

Ejemplos de Categorías

- IES: especificación errónea o incompleta
- MCC: mal interpretación de comunicación con cliente
- IDS: desviación intencional de la especificación
- VPS: violación de estándar de programación
- EDR: error de representación de datos
- IMI: interfaz de módulo inconsistente
- EDL: error en diseño lógico
- IET: testeo erróneo o incompleto
- IID: documentación errónea o incompleta
- PLT: error en lenguaje
- HCI: interfaz ambigua o incompleta
- MIS: miscelaneos

Datos Reales ....

	TOTAL		SERIO		MODERADO		MENOR	
Error	#	%	#	%	#	%	#	%
IES	205	22%	34	27%	68	18%	103	24%
MCC	156	17%	12	9%	68	18%	76	17%
IDS	48	5%	1	1%	24	6%	23	5%
VPS	25	3%	0	0%	15	4%	10	2%
EDR	130	14%	26	20%	68	18%	36	8%
IMI	58	6%	9	7%	18	5%	31	7%
EDL	45	5%	14	11%	12	3%	19	4%
IET	95	10%	12	9%	35	9%	48	11%
IID	36	4%	2	2%	20	5%	14	3%
PLT	60	6%	15	12%	19	5%	26	6%
HCI	28	3%	3	2%	17	4%	8	2%
MIS	56	6%	0	0%	15	4%	41	9%

Datos Reales…

- En este ejemplo, IES + MCC + EDR producen más de la mitad de los errores (53%)
- Si consideramos sólo los errores serios, debería mirarse con atención IES, EDR, EDL, PLT [70%]
- Las acciones a tomar dependen del tipo de error
  - Comprar herramientas case orientadas al análisis
  - Mejorar la calidad de las revisiones
  - Contratar expertos en modelación de datos
  - etc.

Identificación de Riesgos

Riesgo es una situación que en caso de darse, se transforma en un problema para el proyecto…
Por lo tanto, es necesario:

- Identificar posibles riesgos.
- Causas y nivel de manejo de los riesgos.
- Probabilidad de ocurrencia.
- Costo del riesgo (impacto), si es que se convierte en realidad (por lo tanto pasa a ser un problema).
- Medidas de precaución.

Identificación de Riesgos

- Cada medida de precaución debe incluir:
  - Costo.
  - Impacto sobre el proyecto.
- Tomar decisión respecto de cada riesgo desde el punto de vista del negocio.
- No todos los riesgos necesitan ser administrados.

Ejemplos de Riesgos

- Requisitos cambiantes
- Problema pobremente definido
- Rotación de personal
- Cambios externos
- Fallas en la plataforma de desarrollo
- Desconocimiento de la tecnología a utilizar
- Desconocimiento del proceso de desarrollo a utilizar
- Pérdidas de información
- Atraso en la ejecución del proyecto
- Alta dependencia de algún personaje
- ...

Cómo documentar los riesgos?

		Nº Reporte: 15						
		Fecha Reporte: 05/05/2005						
							Impacto de un Riesgo	
Nº	Nombre del riesgo	Breve descripción	Prob.	Impact.	Evaluación		5	Crítico
1	Retiro del Adm. del Proyecto		0.7	5	3.5		4	Alto
2					0		3	Medio Alto
3					0		2	Medio
4					0		1	Medio Bajo
5					0			
6					0			
				Total de riesgos	0.5833333		Suma de Eval. de Riesgo	
Nº	Breve Descripción de Medidas de Contingencia		Riesgos atacados	Responsable				
1	Mantener una organización del equipo de trabajo, tipo "supervisada", que involucre al menos los dos nive		1	GG				
2								
3								
4								
5								
				NOTA:				
				GG: Gerente General				

Conclusiones…

- Las actividades de SQA son necesarias.
- La mayoría de los equipos de trabajo no las incorporan debido al aumento de los costos y tiempos de desarrollo.
- …. al final terminan gastando más y tardando más por no haber incorporado actividades de SQA.
- En el futuro inmediato, debido a la globalización, las empresas que no consideren al SQA como “necesario”, perecerán en manos de aquellos que sí lo hagan.
- El costo del software ha bajado tanto que ahora los clientes buscan calidad.