

Unidad 5

Herramientas de QA

- Cobertura de Testing
- Navegabilidad
- Factores y Dilemas de la Calidad de Software
- Niveles de Staging

Cobertura (coverage)

La concepción que utilizaremos de la Cobertura es:

- aquel porcentaje (o medida porcentual) en la que nuestras pruebas (manuales o automatizadas) recorren la totalidad de caminos posibles dentro de la logica de un programa.

Ejemplo - Cobertura (coverage)

Suponga que se crean 10 requisitos y se crean 100 pruebas -
Luego se ejecutan 90 pruebas.

La cobertura de prueba es del 90 %.

Ahora, en función de esta métrica, los evaluadores pueden
crear casos de prueba adicionales para las pruebas restantes.

Valor - Cobertura (coverage)

Eliminar defectos en etapas tempranas

- Identificar “gaps” en los requerimientos y defectos en etapas tempranas del development cycle
- Ahorra dolores de cabezas a futuro

Cobertura para mayor cobertura

- Con la cobertura se crean mas test cases para asegurar una mayor cobertura lo que lleva a menor cdad de defectos en etapas posteriores
- Aumenta satisfacción del cliente

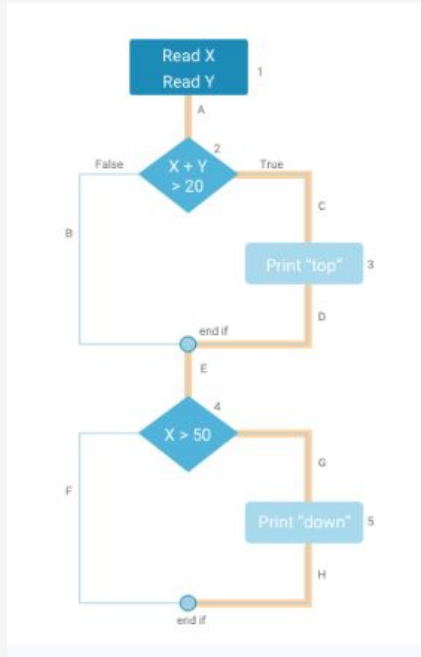
Remueve casos redundantes

- La cobertura ayuda a eliminar tests que no tienen sentido o que se han tornado inútiles
- Esto beneficia a tener test suites mas livianas y rápidas.

Aumento en ROI

- Con menos defectos en ambientes productivos, podemos tener un impacto significativo en el ROI - por que?

Tecnicas- Cobertura (coverage)



Path that covers all the statements in the flowchart:

1A - 2C - 3D - E - 4G - 5H

Esto es solo UN tipo de cobertura - **Statement**

Ejercicio

Tecnicas- Cobertura (coverage)

- Navegar [aquí](https://www.simform.com/blog/test-coverage/) para ver los tipos de cobertura (<https://www.simform.com/blog/test-coverage/>)
- Seleccionar un tipo de Cobertura
 - Decision/Branch coverage
 - Path Coverage
 - Condition Coverage
 - Boundary Value Coverage
- Exponer la cobertura seleccionada al resto de la clase
 - Generar un ejemplo y dibujar el diagrama en caso de ser posible
 - Explicar en que casos utilizaria este estilo y por que
- Sacar conclusiones de forma grupal

Navegabilidad

- Se entiende por navegabilidad a **la facilidad con que pueden recorrerse las distintas secciones de un sitio web o aplicación sin perderse en él**
- **¿Cuál es la diferencia entre navegabilidad, usabilidad y accesibilidad?**
 - No hay que confundir ambos términos
 - **navegabilidad** se refiere a la facilidad para desplazarse de manera voluntaria entre los contenidos
 - **la usabilidad** se relaciona más directamente con la claridad, simpleza y estilo de los componentes.

Ejemplos preceptos básicos de navegabilidad

- La facilidad de navegación dentro de un sitio es una de las razones por las cuales:
 - Los usuario recomiendan un Sitio/App
 - Los usuarios utilizan el Sitio/App
 - Los usuarios “confían” en el Sitio/App
 - Los usuarios deciden volver o no.



Preceptos basicos de navegabilidad

Que componentes/Caracteristicas ayudan a la navegabilidad de in sitio/app?

- **Menú de navegación**
 - permite al usuario desplazarse al menos por las principales secciones del mismo.
- **Ausencia botones “volver”**
 - En algunos sitios se puede ver que cuando se llega a ciertas secciones o páginas, la única alternativa que se presenta al usuario que se encuentra en ella es la de hacer click en un botón “**volver**” o “**atrás**”.
 - Este tipo de navegación termina siendo desagradable para la mayoría de los usuarios, ya que para leer otro artículo debe seguir al menos dos pasos.
- **Boton al Home**
 - Incluir algún enlace a la página de inicio desde todas las demás páginas del sitio da seguridad al usuario y facilita el reinicio de la navegación por el sitio en caso de haberse extraviado.

VOLVER



Preceptos basicos de navegabilidad

Regla de los tres clicks

It suggests that a user of a website should be able to find any information with no more than three mouse clicks.

Esta regla no siempre es posible pero cuando lo sea, hay que intentar cumplirla.

- Si se superan los **tres clicks**, es porque la navegabilidad necesita mejoras
- El paradigma de la navegabilidad es poder acceder desde cualquier parte del sitio a otro **con solo hacer un click**, aunque no siempre es posible.



Qué Factores de calidad que se persiguen con la Navegabilidad?

Robustez

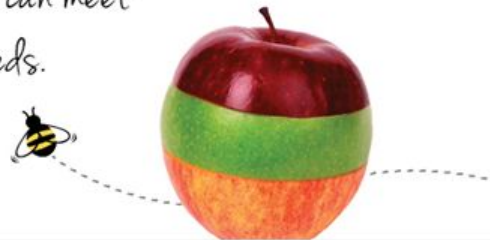
- Grado en el que el software maneja **entradas erróneas** de datos o en el que se presenta interacción inapropiada por parte del usuario.
- ¿El software reconocerá el **error** si entran datos en el límite de lo permitido o más allá y, lo que es más importante, continuará operando sin fallar ni degradarse?
- ¿La interfaz reconocerá los errores cognitivos o de manipulación y guiará en forma explícita al usuario de vuelta al camino correcto?
- ¿La interfaz da un **diagnóstico y guía útiles** cuando se descubre una condición de error (asociada con la funcionalidad del software)?

Qué Factores de calidad que se persiguen con la Navegabilidad? -(cont)

Riqueza :

- Grado en el que la interfaz provee un conjunto abundante de características.
- Puede **personalizarse** la interfaz según las necesidades específicas del usuario?

*Our customizable system can meet
your specific business needs.*



EL DILEMA DE LA CALIDAD DEL SOFTWARE

Bertrand Meyer : “**punto medio mágico**” – Investigador, desarrollador

*Si produce un sistema de software de **mala calidad**, usted pierde porque nadie lo querrá comprar.*

*Por otro lado, si dedica un tiempo infinito, demasiado **esfuerzo y enormes** sumas de dinero para obtener un elemento perfecto de software, entonces tomará tanto tiempo terminarlo y será tan **caro** de producir ,que de todos modos quedará fuera del negocio.*

En cualquier caso, habrá perdido la ventana de mercado, o simplemente habrá agotado sus recursos.

*De modo que las personas de la industria tratan de situarse en ese **punto medio mágico** donde el producto es suficientemente bueno para no ser rechazado de inmediato, no en la evaluación, pero tampoco es un objeto perfeccionista ni con demasiado trabajo que lo convierta en algo que requiera demasiado tiempo o dinero para ser terminado.*

EL DILEMA DE LA CALIDAD DEL SOFTWARE (cont)

Software “suficientemente bueno”

Contiene las funciones y características de alta calidad que desean los usuarios, pero al mismo tiempo tiene otras más oscuras y especializadas que contienen errores conocidos.

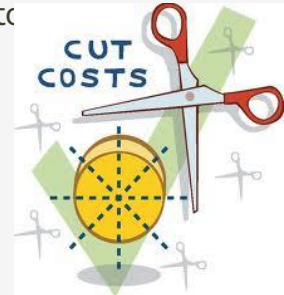
El vendedor de software espera que la gran mayoría de usuarios finales **perdone los errores** gracias a que estén muy contentos con la funcionalidad de la aplicación.



EL DILEMA DE LA CALIDAD DEL SOFTWARE (cont)

El costo de la calidad

- *La calidad es importante, pero cuesta tiempo y dinero —demasiado tiempo y dinero!!!*
- La calidad tiene un costo, pero la mala calidad también lo tiene —no sólo para los usuarios finales que deban vivir con el software defectuoso, sino también para la organización del software que lo elaboró y que debe darle mantenimiento



Niveles de Staging

Cuando nos referimos a niveles de Staging , desde el punto de vista del QA nos referimos a un entorno de prueba (stage).

Característica principal:

- Réplica casi exacta de un entorno de producción para pruebas de software.
- **Objetivo:**
 - Probar código, versiones y actualizaciones y poder garantizar la calidad en un entorno similar al de producción antes de la implementación (release) de la aplicación.

Niveles de Staging (cont)

- **Cuantos Niveles de Staging puedo tener?**

El numero de Niveles de staging varia sgun la embergadura de la infraestructura (y dinero) de la empresa que posee la aplicacion.

- **Es necesario un nivel de Staging solo para QA?**

Si - mientras mas autonomia tenga el equipo de QA de manejar un nivel de stage - mejor sera el resultado del trabajo de dicho equipo

- **Cuantos Niveles se deberian tener como MINIMO?**

Stage1 : Local Dev

Stage 2: Common Dev

Stage 3: QA / UAT

Unidad 6

Reporte de Errores - Metricas y Objetivos

- Reporte de Bugs
- Pruebas Dinamicas / Estaticas
- ISO 9001-2000
- OKRs
- OKRs para equipos de Qa

Errores - Bugs

- ¿Qué es un Bug?
 - Fallo o error en un ordenador o sistema de sw. que provoca un resultado no deseado.
 - Problema presente en un sw. que desencadena un resultado indeseado.

Errores - Bugs


Consideraciones

- No todos los defectos del sistema son causados por errores de código.
- Una fuente común de defectos es causada por la **falta de requerimientos** o requerimientos pobres.
 - Esto sucede por ejemplo con los problemas de
 - requerimientos en aspectos no funcionales, escalabilidad, mantenibilidad, usabilidad, etc.
- Las **fallas** ocurren en el siguiente proceso: un programador comete un error, que resulta en un defecto en el código. Fuente. Si este defecto es ejecutado, en algunas ocasiones el sistema produce resultados incorrectos causando la falla.
- Un simple defecto puede resultar en un amplio rango de síntomas.

Como reportar un bug

- Existen programas llamados **bug trackers** que sirven para reportar bugs o defectos en un sistema.
- Alguno de estos bug-trackers son **Jira**, **redmine**, **mantis**, que son gestores de proyectos y poseen un sistema de tickets para reportar fallas y de esta forma los desarrolladores puedan verlo y corregirlo.
- El reporte debe cumplir con una serie de pautas que son las siguientes:
 - Asociación al proyecto
 - ID único
 - Titulo
 - Descripción amable explicando el bug
 - Pasos para reproducir el bug
 - Resultado actual
 - resultado esperado
 - Entorno de prueba (SO, browsers)
 - Screenshot / Video

Añadido por Dinno Admin hace menos de 1 minuto.

Fecha de inicio: 2013-01-07
 Fecha fin:
 % Realizado:  0%
 Tiempo dedicado: -

 Citar

The search results have overflow when a user insert a long text string

Steps:

- 1.- Go to: <http://qatraining.comli.com>
- 2.- In the search box insert:

"shfrsjkdhfksjdhfksjd hfksjd hfskjdhfkjsdfjhkhkjhkhkjhkfhsdjdhfksjd hfkjsd fhksdjfhsk"

- 3.- Click on "Buscar" button.

Actual Result: The system displays the text string overflow
Expected Result: The system should be display the text string correctly

Please, see the attached screenshot

 [bug.png](#) (64,411 KB)  Dinno Admin, 2013-01-07 12:07

Ejemplo de Bugs en un sitio Web

- **HTML**
 - 1.- Etiquetas mal cerradas
 - 2.- Compatibilidad de HTML 5 entre browsers
- **Javascript**
 - 1.- Variables no definidas
 - 2.- Problemas de Sintaxis
 - 3.- problemas en sitios lentos
 - 4.- problemas cuando no está habilitado
- **Seguridad**
 - 1.- XSS
 - 2.- SQLi - Blind SQLi
 - 3.- Problemas de validación

Ejemplo de Bugs en un sitio Web

? **Flash**

- 1.- Problemas cuando no está activado
- 2.- Problemas de carga
- 3.- Problemas de llamadas recurrentes
- 4.- Problema en navegabilidad en móvil

? **Errores HTTP**

- 1.- 404: links rotos
- 2.- 301/302: Redirecciones incorrectas
- 3.- 403: Servicios que no funcionan por faltas de permisos
- 4.- 500: Servicios caídos

? **Cookies**

- 1.- Cookies sin vencimientos
- 2.- Cookies que no están encriptados
- 3.- problemas de cookies de diferentes dominios (?)

Ejemplo de Bugs en un sitio Web

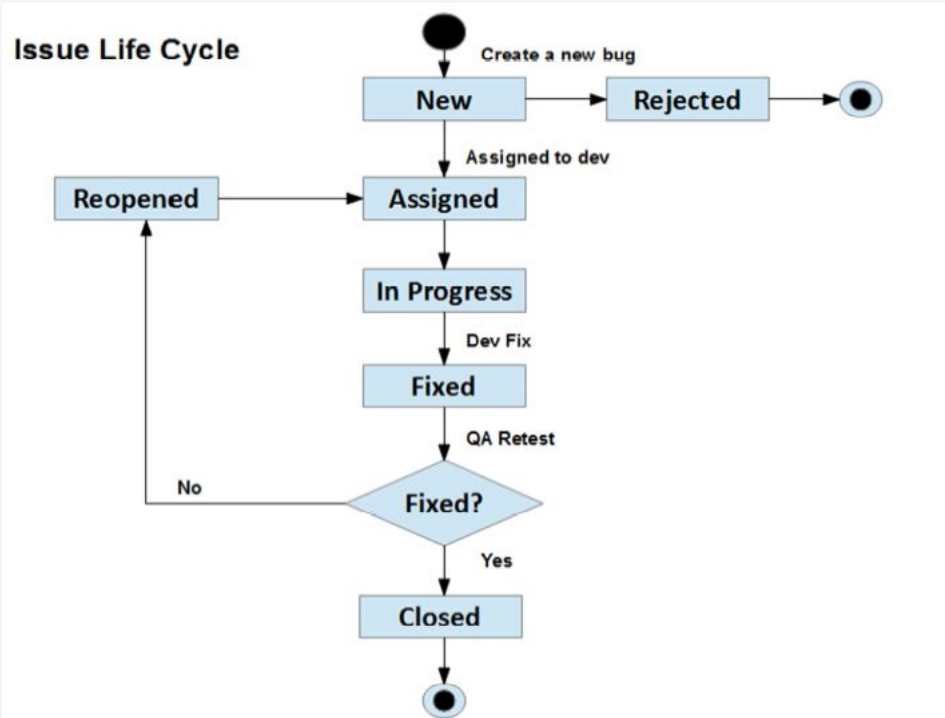
? Rendimiento del sitio

- ? 1.- Archivos pesados como imágenes >> Usar sprites
- ? 2.- muchos archivos js y css >> js/css minifying
- ? 3.- Peticiones constantes al servidor

? Estilos

- ? 1.- Colores y fuentes incorrectas
- ? 2.- problemas de layout
- ? 3.- problema de tamaño de los objetos

Ciclo de vida de un Bug



MÉTODOS DE PRUEBA

? Pruebas estáticas y dinámicas

· Pruebas estáticas

- ? Las pruebas estáticas se basan en el examen **manual**, también llamado revisiones, y en el análisis automatizado, o análisis estático del código o de cualquier documentación del proyecto sin ejecutar el código.
- ? Las **revisiones** constituyen una forma de probar los productos de trabajo del software, incluyendo código, y pueden realizarse antes de ejecutar las pruebas dinámicas.
- ? Los **defectos** detectados durante las revisiones al principio del ciclo de vida (ej.: en requerimientos) a menudo son más baratos de eliminar (ej.: omisiones en los requerimientos).
- ? Este tipo de revisiones pueden realizarse tanto manualmente como mediante herramientas de soporte.
- ? La principal actividad manual consiste en **examinar** un producto de trabajo y hacer comentarios al respecto.
- ? Cualquier **producto** puede ser objeto de una revisión, como por ejemplo, requerimientos, código, test cases, scripts.

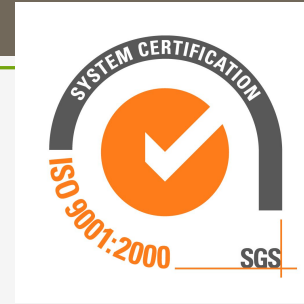
· Pruebas estáticas

- ? **Ejemplo** de este tipo de pruebas estáticas son las **revisiones** por pares o peer reviews . Se pueden realizar revisiones por pares a bugs reportados, a test cases, etc.
- ? Los **beneficios** de realizar estas revisiones incluyen como ya mencionamos,
 - ? *“la detección y corrección temprana”* de defectos,
 - ? *el desarrollo de mejoras en el proceso y en la productividad,*
 - ? *el ahorro de tiempo y dinero, etc.*
- ? Entre los **defectos** típicos que resultan más fáciles localizar en las pruebas estáticas se incluyen:
 - ? desviaciones de estándares.
 - ? Defectos de **requerimientos**.
 - ? Defectos de **diseño, mantenibilidad** insuficiente, y especificaciones de **interfaz** incorrectas

· Pruebas Dinámicas

- ? La ejecución de pruebas dinámicas o análisis dinámico se refiere al conjunto de pruebas que requieren la **ejecución del código** para determinar cómo es su funcionamiento a lo largo del tiempo.
- ? En la prueba dinámica el software en realidad debe ser **compilado** y **ejecutado**; consiste en trabajar con el software, dando valores de entrada y comprobar si el resultado es el esperado. Estas son las actividades de validación.
- ? Pruebas **unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación** son algunas de las metodologías de pruebas dinámicas.
- ? Se parte de test cases específicos por la ejecución del objeto de prueba o programas en ejecución.
- ? Existen variedad de estrategias de testing que se describen más adelante y que son parte de este conjunto de pruebas dinámicas

Elementos básicos ISO 9001:2000



- ? Establecer los elementos de un sistema de administración de la calidad.
 - ? Desarrollar, implementar y mejorar el sistema.
 - ? Definir una política que ponga el énfasis en la importancia del sistema.
- ? Documentar el sistema de calidad.
 - ? Describir el proceso.
 - ? Producir un manual de operación.
 - ? Desarrollar métodos para controlar (actualizar) documentos.
 - ? Establecer métodos de registro.
- ? Apoyar el control y aseguramiento de la calidad.
 - ? Promover la importancia de la calidad entre todos los participantes.
 - ? Centrarse en la satisfacción del cliente.
 - ? Definir un plan de calidad que se aboque a los objetivos, responsabilidades y autoridad.
 - ? Definir mecanismos de comunicación entre los participantes.

Elementos básicos ISO 9001:2000

- ? Establecer **mecanismos de revisión** para el sistema de administración de la calidad.
 - ? Identificar métodos de revisión y mecanismos de retroalimentación.
 - ? Definir procedimientos para dar seguimiento.

- ? **Identificar recursos para la calidad**, incluidos personal, capacitación y elementos de la infraestructura.
 - ? Establecer mecanismos de control.
 - ? Para la planeación
 - ? Para los requerimientos del cliente
 - ? Para las actividades técnicas (tales como análisis, diseño y pruebas)
 - ? Para la vigilancia y administración del proyecto

- ? **Definir métodos de corrección.**
 - ? Evaluar datos y métricas de la calidad.
 - ? Definir el enfoque para la mejora continua del proceso y la calidad.

OKRs

- Que quiere decir la sigla?
 - Objectives and Key Results, que en español traduce como objetivos y resultados clave,
 -
- Que son los OKRs
 - Son objetivos que se ponen las personas/equipos para lograr crecimiento y mejora en los diferentes ámbitos (personal, laboral, económico, profesional, etc.)
- Característica:
 - Los OKR se diferencian de herramientas de productividad como los TODO's (lista de cosas por hacer) porque los segundos son acciones, mientras que los primeros son los resultados obtenidos por estas acciones.

?

Beneficios de los OKRs



Ejemplos de OKRs

Objetivos

- Metas aspiracionales que indican el problema a resolver sin un grado de especificidad exacto

KRs:

- Objetivos MEDIBLES no solo en valor sino en tiempo - estos KRs son los que se monitorean para ver la completitud del Objetivo que los contiene

Objetivo: Tener clientes más felices

KR 1 – Obtener un NPS (Net Promoter Score) de 9

KR 2 – Disminuir el CCR (Customer Churn Rate) a 0

KR 3 – Aumentar la Tasa de Retención a 95%

Diferencia entre Okrs y KPis

Que es un KPi?

- Un KPI (Key performance indicator) es siempre un valor medible, una métrica, **UN NÚMERO**.
 - Un Kpi es una meta numerica que nos ayuda a saber que tan bien/mal esta el valor de un “parametro” en comparacion con una meta preestablecida

Diferencia con los OKRs:

- Mientras que un OKR será un objetivo o un resultado clave. Por ejemplo, “Tasa de abandono” puede ser un KPI de ventas y “aumentar la retención de clientes” (objetivo) “en un 15%” (resultado clave) puede ser el OKR

OKRs para equipos de Calidad

Objective – Improve Our Testing Procedure

Key Result 1: Implement test driven development in 3 new development teams

Key Result 2: Increase unit test coverage to 75% of code

Key Result 3: Make sure satisfaction score of product management to testing team is at least 7.5

Key Result 4: Increase time spent on reviewing codes by 20 minutes each day

Objective – Drive Quality For Features In Our New Release

Key Result 1: Identify 15 bugs by the end of Q2

Key Result 2: Ensure there no more than 1 critical bug reported by the end of Q3

Key Result 3: Have 0 regressions in Q3