



# Aseguramiento de la Calidad

Universidad de Mendoza

Julieta Barrionuevo  
Juan Manuel Ojeda

---

# Clase N°1

## **Introduccion a la Materia**

- **Presentacion de Profesores**
- **Metodología de clase**
- **Regimen de promocion**
- **Generalidades de la Materia**

# Contenidos

- ❑ Introducción a la calidad del software.
- ❑ SDLC ( Software development life cycle )
- ❑ Qué es Aseguramiento de la Calidad de software
- ❑ Tipos de Testing
  - ❑ Pruebas de Caja Negra y Caja Blanca
  - ❑ Pruebas de unidad
  - ❑ Integracion, Performance, Carga
- ❑ Procesos y Pruebas de Calidad de software
- ❑ Rol y responsabilidades del tester
- ❑ Ambientes de Prueba ( Staging )
- ❑ Herramientas y tipos de Prueba
- ❑ Deteccion y reporte de Defectos-Bugs

# Objetivos

- ? Incorporar **Conceptos generales** sobre calidad de sw. y su significado
- ? Conocer las **competencias** del profesional de la calidad de Software
- ? Comprender la **Importancia** de trabajar para y con Calidad (Mitos y Realidades )
- ? Comprender de qué manera se llevan a cabo **revisiones de software** eficaces.
- ? Comprender el valor del area de QA en los distintos tipos de metodologías ágiles
- ? Adquirir destrezas para la creación de casos de **prueba eficaces**.
- ? Adquirir destrezas para **testeo de software** utilizando distintas metodologías.
- ? Integrar conocimientos de **análisis de sistemas y metodologías Ágiles**
- ? Obtener conocimiento para proponer/comprender estrategias de QA
- ? Conocer **herramientas de Prueba** y el uso efectivo de las mismas

## Inserción en el Mercado Laboral

- Conocer conceptos sobre **Mejora de procesos**
- Prepararnos para la Demanda del Mercado.

## Requisitos para obtener la Promoción/Regularidad:

**Polinomio** = Promedio (Trabajos Prácticos) \* **0.2** +  
Promedio(Exámenes Parciales) \* **0.3** + Asistencia \* **0.1** +  
Proyecto Final \* **0.4**

- ? **Promoción** = Polinomio  $\geq 6$
- ? **Regularidad** =  $4 \leq$  Polinomio  $< 6$
- ? **Recursado** = Polinomio  $< 4$

# Importa la Calidad de software?

Un ejemplo extremo:

*En el mes de noviembre de 2000, en un hospital de Panamá, 28 pacientes recibieron dosis masivas de rayos gama durante su tratamiento contra diversos tipos de cáncer. En los meses que siguieron, 5 de estos pacientes murieron por envenenamiento radiactivo y 15 más sufrieron complicaciones serias. ¿Qué fue lo que ocasionó esta tragedia? Un paquete de software, desarrollado por una compañía estadounidense, que fue modificado por técnicos del hospital para calcular las dosis de radiación para cada paciente.*

*Los tres médicos panameños que “customizaron” el software para que diera capacidad adicional fueron acusados de asesinato en segundo grado. La empresa de Estados Unidos enfrentó litigios serios en los dos países. Gage y McCormick comentan lo siguiente:*

*Éste no es un relato para prevenir a los médicos, aun cuando luchen por estar fuera de la cárcel si no entienden o hacen mal uso de la tecnología. Tampoco es la narración de cómo pueden salir heridos, o algo peor, los seres humanos a causa del software mal diseñado o poco explicado, aunque hay muchos ejemplos al respecto.*

# CAPITULO 1



**Introducción a la calidad del software.**

# Clase N°2

## Introducción a la calidad del software.

- Definiciones de Calidad
- Software de calidad
- Historia
- Tipos de Calidad “high level “
- Ejercicios practicos
- “Que es Calidad” desde diferentes puntos



# Definicion de calidad

## *Definicion de Calidad según Norma ISO 8204:*

**“Totality of characteristics of an entity that bears on its ability to satisfy stated and implied needs.”**

Algo así como: “Totalidad de las características de una entidad que satisfacen las necesidades explicitas e implícitas “

# Software con calidad?

- ? ¿Hace lo que se le pide?
- ? ¿Lo hace de forma **confiable** todo el tiempo?
- ? ¿Utiliza los recursos adecuados para la ejecución de sus funciones?
- ? ¿Pueden prevenirse accesos no autorizados a la información que maneja?
- ? ¿Es **fácil** y **cómodo** de manejar?



# Ejercicio 1 - Generalidades

Este primer ejercicio servirá únicamente para verificar el conocimiento, de lo que entendemos, son nociones básicas de cualquier profesional de Calidad. No hay que preocuparse si nos equivocamos al responderlas porque de hecho, estamos aquí para aprender.

Completar el siguiente ejercicio formando grupos de no más de 4 personas para después realizar la puesta en común con el resto de la clase.

- a. Todos sabemos que la traducción de la palabra “Test” es prueba, por lo tanto la traducción de “Tester” sería algo así como “Probador”. En este caso nos estamos refiriendo al “Probador de Software”, ahora bien, en jerga informática se utiliza el término “QA”.

Responder lo siguiente:

- a.1 - Cual es el significado en español de “QA”?
- a.2 - Existe diferencia entre los roles de un TESTER y QA? si es así indique cuales
- a.3 - Cuales son los objetivos principales del Área o profesionales de Calidad de software?

# Un poco de Historia...

? 1990:

? Las principales corporaciones desperdiciaban miles de millones de dólares en software que no tenía las características ni la funcionalidad que se habían prometido.

? **CIO Magazine [Lev01]**

Dió la alerta: “**Dejemos de desperdiciar \$78 mil millones de dólares al año**”, y lamentaba el hecho de que “**las empresas estadounidenses gastan miles de millones de dólares en software que no hace lo que se supone que debe hacer**”.

## Terminología Relacionada

- ✓ Software Engineering
- ✓ Software Quality
- ✓ Quality Assurance & Control
- ✓ Software Process
- ✓ Reviews & Inspections
- ✓ Test Strategy
- ✓ Test Planning
- ✓ Test Estimation
- ✓ Test Requirements
- ✓ Test Design
- ✓ Test Techniques
- ✓ Static Test Techniques
- ✓ Test Metrics
- ✓ Test Execution
- ✓ Test Environment
- ✓ Test Reporting
- ✓ Test Process
- ✓ Testing Standards
- ✓ Test Tools
- ✓ Configuration Management
- ✓ Test Team Management
- ✓ Defect Prevention

# Ejercicio 1 - Repaso general

Completar el siguiente ejercicio formando grupos de no más de 4 personas para después realizar la puesta en común con el resto de la clase.

- a. Todos sabemos que la traducción de la palabra “Test” es prueba, por lo tanto la traducción de “Tester” sería algo así como “Probador”. En este caso nos estamos refiriendo al “Probador de Software”, ahora bien, en jerga informática se utiliza el término “QA”.

Responder lo siguiente:

- a.1 - Cual es el significado en español de “QA”?
- a.2 - Existe diferencia entre los roles de un TESTER y QA? si es así indique cuales
- a.3 - Cuales son los objetivos principales del Área o profesionales de Calidad de software?

## Ejercicio 2- Terminología del Area de QA

En caso de conocer el significado de los siguientes términos. Indique en forma breve, que es y para qué sirven :

b.1 - Test Case

b.2 - Issue Tracker

b.3 - Use Case

b.4 - Calidad estructural

b.5 – Calidad Funciona

b.6 - SQM

b.7 - Acceptance Criteria

# Tipos de Calidad - “High Level”

## **Calidad Funcional ( Functional Quality ):**

Responde al “QUE HACE”

Refleja que TAN BIEN el producto cumple o complementa un diseño que ha sido provisto por el cliente relacionado con los requerimientos funcionales o especificaciones.

## **Calidad Estructural**

Responde al “COMO LO HACE”

Se refiere al “COMO” el producto cumple con los requisitos no funcionales que dan soporte a los requisitos funcionales ( robusto, mantenibilidad, portabilidad )

## **SQM**

El objetivo del Software Quality management es el de administrar la calidad de software, definir los procesos y los requisitos para esos procesos. Debe a su vez implementar métricas para cada proceso, sus salidas y un canal de feedback.



# Herramientas del Area de QA

Dentro de las herramientas podemos mencionar las siguientes

- **Casos de Uso**
- **Diagramas O.O ( Clase , ER , Secuencia)**
- **Wireframes**
- **SQL – Bases de Datos**
- **Documentación del sistema bajo prueba**

Además de las mencionadas existe una herramienta que hoy por hoy es la herramienta por excelencia utilizada por persona que se dedica a la calidad de software. Esta herramienta se llama Caso de prueba (Test Case )



Qué entendemos por  
calidad de Software?

# ¿QUÉ ES CALIDAD?

David Garvin , de Harvard Business School,



## ? El **punto de vista trascendental** :

- ? El *punto de vista del usuario* que concibe la calidad en términos de las metas específicas del usuario final. Si un producto las **satisface**, tiene calidad.
- ? Cada usuario tiene diferentes estándares de calidad

## ? **Ejemplo:**

- ? **McDonald's** valora que la aplicación pueda realizar un pedido en tiempo y forma, **Burger** que puedan descargar descuentos, otros sólo con visualizar sucursales y otros con consultar precios están satisfechos.. etc

# ¿QUÉ ES CALIDAD?

? El **punto de vista del fabricante** :

? La define en términos de las **especificaciones originales** del producto. Si éste las cumple, tiene calidad.

? **Ejemplo:**

? Yo hago una aplicación, defino la calidad de la aplicación en base a si cumple con las especificaciones que YO necesito



# ¿QUÉ ES CALIDAD?



? El *punto de vista del producto* :

- ? Tiene que ver con las características inherentes (funciones y características) de un producto
- ? Cuanto de lo que se desea está presente en el producto?
- ? El consumidor confunde "**calidad** con **precio**", entiende que mientras más caro es el producto, más cantidad de atributo está presente en el mismo.

# ¿QUÉ ES CALIDAD?

- ? El **punto de vista basado en el valor** :
  - ? De acuerdo con lo que un cliente está dispuesto a **pagar** por un producto.
- ? Ejemplo:
  - ? **McDonald's** pagará más dinero por una aplicación con mayor seguridad
  - ? **Burguer** pagará más dinero por un mejor diseño, usabilidad.



# ¿QUÉ ES CALIDAD?

- ? La **calidad del diseño** se refiere a las características que los diseñadores especifican para un producto (tipo de materiales, tolerancias y especificaciones del desempeño)

En el desarrollo del software, la *calidad del diseño* incluye el **grado en el que el diseño cumple** las funciones y características especificadas en el modelo de requerimientos.

## ? Ejemplo:

- ? Macdonals requiere que se puedan visualizar las dif sucursales, los diseñadores implentan un módulo de visualización en google maps..

# ¿QUÉ ES CALIDAD?

La calidad es importante, pero que si el usuario no está satisfecho, nada de lo demás importa.



# ¿QUÉ ES CALIDAD?

? Robert Glass [Ing de soft-escritor, especializado en calidad] afirma que es mejor plantear una relación más intuitiva:

**satisfacción del usuario** = producto que funciona + **buena calidad** + **entrega dentro del presupuesto y plazo**

# CALIDAD DEL SOFTWARE

## ? Cómo la definimos?

*Proceso eficaz de software* que se aplica de manera que crea un **producto útil** que proporciona valor medible a quienes lo **producen** y a quienes lo **utilizan**.

### ? Un proceso eficaz de software

- ? Infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. (administración de proyectos, prácticas de Ing de soft, administración de los cambios, revisiones técnicas)

# CALIDAD DEL SOFTWARE

## ? *Un producto útil*

- ? Entrega contenido, funciones y características que el **usuario final desea**;
- ? Satisface los requerimientos establecidos en forma explícita por los participantes.

## ? *Al agregar valor para el productor y para el usuario de un producto*

- ? Proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales.
- ? Software de alta calidad requiere un menor **esfuerzo de mantenimiento, menos errores** que corregir y **poca asistencia** al cliente.
- ? Usuarios obtienen valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios.

# Clase N°3

## Teoría de la Calidad de Software

- Repaso de Metodologías Agiles (Scrum )
- Dimensiones de Calidad - Segun Garvin
- Factores de Calidad - Segun Garvin
- Factores de Calidad - Segun Mcall
- Repaso para TP1

# David A. Garvin

Garvin se incorporó como profesor de la escuela de negocios de Harvard en 1979. Publicó un libro titulado Operations Strategy: Text and cases . Garvin se ha distinguido no solamente por ser profesor de casos sino que ha realizado numerosos aportes relacionadas con el uso de casos para la enseñanza.

**12 de mayo de 1952**, Nueva York, Nueva York, Estados Unidos

**30 de abril de 2017**, Lexington, Massachusetts, Estados Unidos

# Dimensiones de la calidad de Garvin

Sugiere que la calidad debe tomarse en cuenta, adoptando un punto de vista multidimensional que comience con la evaluación de la **conformidad** y termine con una visión trascendental (**estética**).

## ? Calidad del desempeño

- ? ¿El software entrega todo el contenido, las funciones y las características especificadas como parte del modelo de requerimientos, de manera que da valor al usuario final?

## ? Calidad de las características.

- ? ¿El software tiene características que sorprenden y agradan la primera vez que lo emplean los usuarios finales?

## ? Confiabilidad

- ? ¿El software proporciona todas las características y capacidades sin fallar?
- ? ¿Está disponible cuando se necesita? ¿Entrega funcionalidad libre de errores?

# Dimensiones de la calidad de Garvin

## ? Conformidad.

- ? ¿El software concuerda con los estándares locales y externos que son relevantes para la aplicación?
- ? ¿Concuerda con el diseño *de facto* y las convenciones de código?
- ? Por ejemplo, ¿la interfaz de usuario está de acuerdo con las reglas aceptadas del diseño para la selección de menú o para la entrada de datos?

## ? Durabilidad.

- ? ¿El software puede recibir mantenimiento (cambiar) o corregirse (depurarse) sin la generación inadvertida de eventos colaterales?
- ? ¿Los cambios ocasionarán que la tasa de errores o la confiabilidad disminuyan con el tiempo?

# Dimensiones de la calidad de Garvin

## ? Servicio

- ? ¿Existe la posibilidad de que el software reciba **mantenimiento** (cambios) o correcciones (depuración) a corto plazo?
- ? ¿El equipo de apoyo puede adquirir toda la información necesaria para hacer cambios o corregir defectos?

## ? Estética

- ? Una entidad estética posee cierta **elegancia**, un flujo único y una “**presencia**” obvia que es difícil de cuantificar y que, no obstante, resulta evidente.

## ? Percepción.

- ? En ciertas situaciones, existen **prejuicios** que influirán en la **percepción** de la calidad por parte del usuario.
- ? **Por ejemplo**, si se introduce un producto de software elaborado por un proveedor que en el pasado ha demostrado mala calidad, la percepción de la calidad del producto tendrá influencia negativa. De manera similar, si un vendedor tiene una **reputación** excelente se percibirá buena calidad, aun si ésta en realidad no existe.



## 2.2 Factores de la calidad de McCall

Factores que afectan a la calidad del software

Facilidad de mantenimiento  
Flexibilidad  
Facilidad de prueba

Portabilidad  
Reusabilidad  
Interoperatividad

**Revisión del producto**

**Transición del producto**

**Operación del producto**

Corrección    Fiabilidad    Usabilidad    Integridad    Eficiencia

Factores de calidad de McCall (Fuente: Roger Pressman, 2005)

# Factores de la calidad de McCall

OPERACIÓN DEL PRODUCTO: *Características operativas*

? **Corrección – Hace lo que quiero?**

- ? Grado en el que un programa satisface sus especificaciones y en el que cumple con los objetivos de la misión del cliente.

? **Confiabilidad – Lo hace de forma confiable todo el tiempo?**

- ? Grado en el que se espera que un programa cumpla con su función y con la precisión requerida

? **Eficiencia – Se ejecuta de la mejor manera?**

- ? Cantidad de recursos y de código requeridos por un programa para llevar a cabo su función.

## Factores de la calidad de McCall

### ? **Integridad. – Es seguro?**

- ? Grado en el que es posible controlar el acceso de personas no autorizadas al software o a los datos.

### ? **Usabilidad – Esta diseñado para ser usado?**

- ? Esfuerzo que se requiere para aprender, operar, preparar las entradas e interpretar las salidas de un programa.

# Factores de la calidad de McCall

## REVISIÓN DEL PRODUCTO: *Capacidad de soportar cambios*

### ? **Facilidad de recibir mantenimiento.**

- ? Esfuerzo requerido para detectar y corregir un error en un programa

### ? **Flexibilidad.**

- ? Esfuerzo necesario para modificar un programa que ya opera.

### ? **Facilidad de prueba.**

- ? Esfuerzo que se requiere para probar un programa a fin de garantizar que realiza la función que se pretende.

# Factores de la calidad de McCall

## TRANSICIÓN DEL PRODUCTO: *Adaptabilidad a nuevos entornos*

### ? **Portabilidad.**

- ? Esfuerzo que se necesita para transferir el programa de un ambiente de sistema de hardware o software a otro.

### ? **Reusabilidad.**

- ? Grado en el que un programa (o partes de uno) pueden volverse a utilizar en otras aplicaciones (se relaciona con el empaque y el alcance de las funciones que lleva a cabo el programa).

### ? **Interoperabilidad.**

- ? Esfuerzo requerido para acoplar un sistema con otro.



Linux



# Qué Factores de calidad que se persiguen?

Para hacer la evaluación, se necesita determinar atributos específicos y medibles

## ? Intuitiva

? Grado en el que la **interfaz** sigue **patrones** esperados de uso, de modo que hasta un novato la pueda utilizar sin mucha capacitación.

- ? ¿La **interfaz** lleva hacia una comprensión fácil?
- ? ¿Todas las **operaciones** son fáciles de localizar e iniciar?
- ? ¿La interfaz usa una metáfora reconocible?
- ? ¿La **entrada** está especificada de modo que economiza el uso del teclado o del mouse?
- ? ¿La **estética** ayuda a la comprensión y uso?

? Ejemplos de mala usabilidad:

- ? <http://www.lingscars.com/>
- ? <http://arngren.net/>
- ? <http://dokimos.org/index.aspx>
- ? [http://www.web\\_4\\_all.republika.pl/](http://www.web_4_all.republika.pl/)



# Qué Factores de calidad que se persiguen?

## ? Eficiencia :

- ? Grado en el que es posible **localizar** o **iniciar** las **operaciones** y la **información**
  - ? ¿La **distribución** y **estilo** de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?
  - ? ¿Una secuencia de operaciones (o entrada de datos) puede realizarse con economía de movimientos?
  - ? ¿Los **datos de salida** o el contenido están presentados de modo que se entienden de inmediato?
  - ? ¿Las operaciones jerárquicas están organizadas de manera que minimizan la profundidad con la que debe navegar el usuario para hacer que alguna se ejecute?

