



Argentina
programa
4.0

Testing Manual, Ágil y con herramientas

Clase 1

Testing

Que vamos a hacer hoy



AGENDA 23/02/2023

- 1- Diferencia entre validación y verificación.
- 2- Introducción al testing y la calidad de software.
- 3- Diferencias entre QA, QC
- 4- Diferencias entre Testing y Debugging.
- 5- Independencia de las pruebas.
- 6- Principio de las pruebas.
- 7- Cuestionario de autoevaluación

Validación
≠
Verificación

Validación y Verificación

Son dos actividades o procesos de **control** encaminadas a determinar si **se está construyendo el producto correcto de la manera correcta**

Validación ¿Estamos construyendo el producto correcto?, o en su defecto,
¿Construimos el producto correcto?

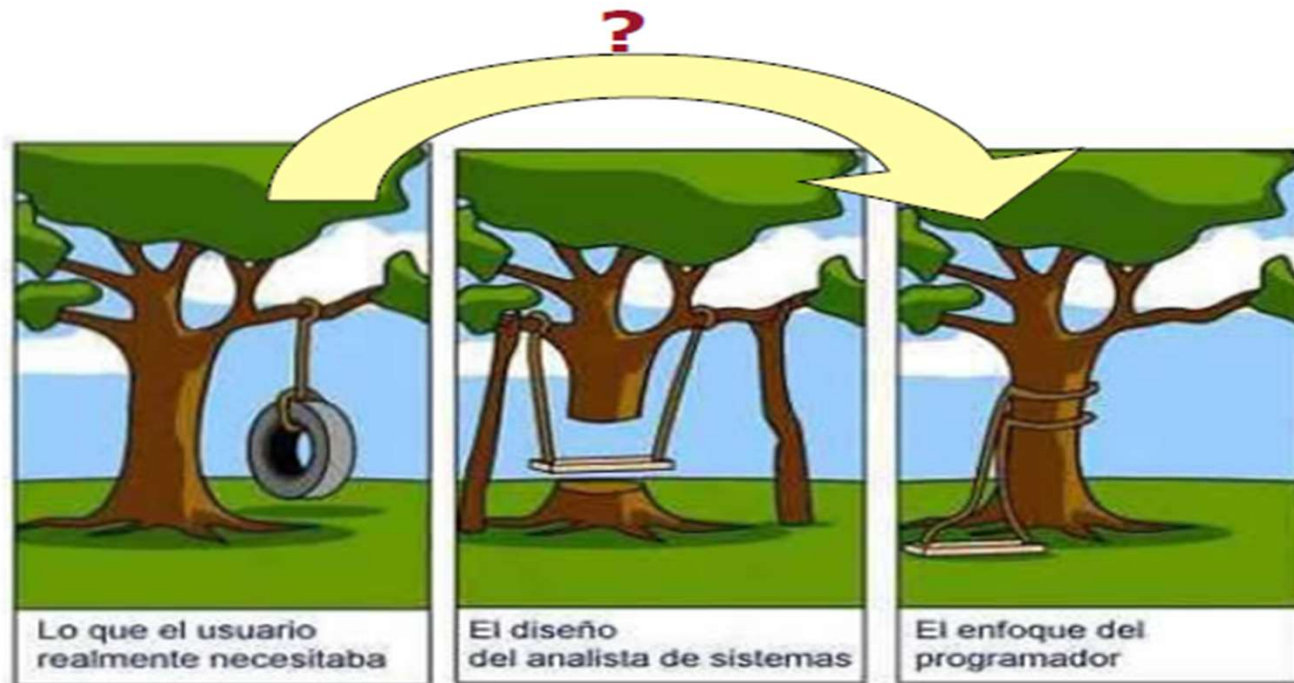
Las actividades de validación se dan después de que el software ha sido desarrollado para determinar si se han implementado correctamente los requerimientos.

Verificación ¿Estamos construyendo el producto de la manera correcta?

Las actividades de verificación son desarrolladas de manera concurrente con las actividades del desarrollo de software.

Validación y Verificación

Son actividades encaminadas a determinar si **se está construyendo el producto correcto de la manera correcta**



Validación y Verificación



FACTURACIÓN

AFIP lanza desde el 1º de junio un nuevo servicio para monotributistas A, B y C y sociales que permite emitir tickets a consumidores finales en 3 pasos.

legalecancador.com.ar

Definiciones de Testing

Es la verificación dinámica de la
adecuación del sistema a los
requerimientos.

“Testing es el infinito proceso de
comparar lo invisible con lo ambiguo”

Por James Bach

“El testing es una forma efectiva de mostrar la presencia de errores pero es una forma inadecuada de mostrar la ausencia de los mismos.”

Por Dijkstra

Definiciones de calidad

CALIDAD

“Propiedad o conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie”. (RAE)

CALIDAD

“Grado en el que un conjunto de características inherentes a un objeto (producto, servicio, proceso, persona, organización, sistema o recurso) cumple con los requisitos” (ISO 9000)

CALIDAD del SOFTWARE



Existen diversas
definiciones

El cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados, y de las características implícitas que se espera de todo software desarrollado profesionalmente.

R. Pressman

El grado en que el producto software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas.

ISO/IEC 25000

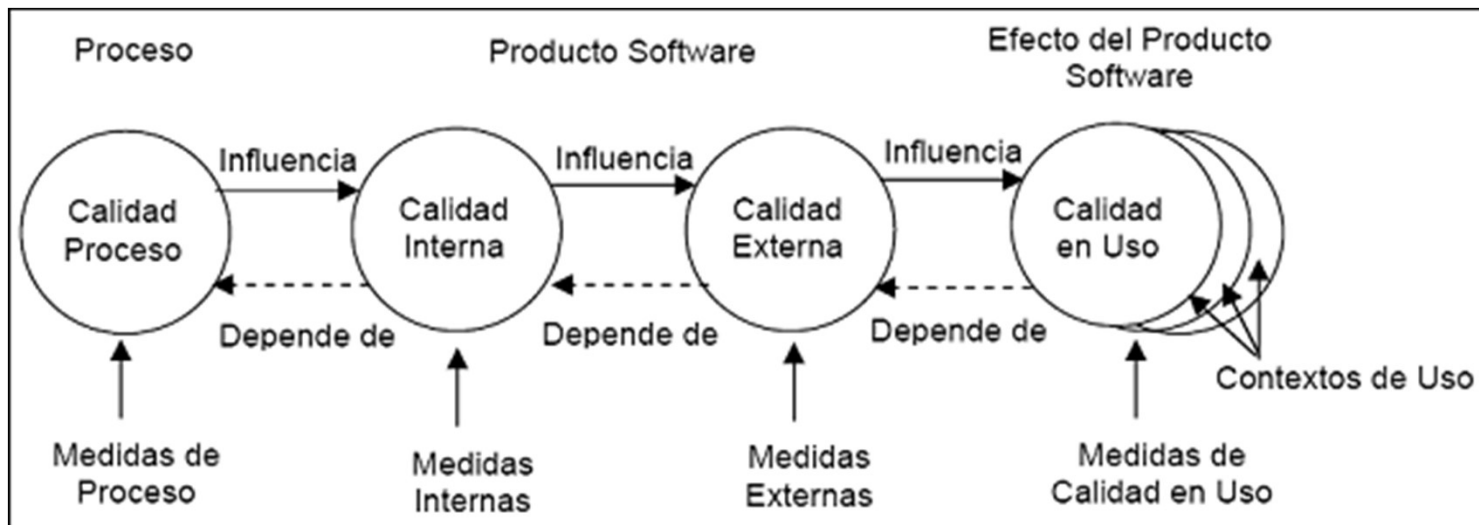
CALIDAD del SOFTWARE

La calidad del software puede ser abordada desde distintos enfoques:

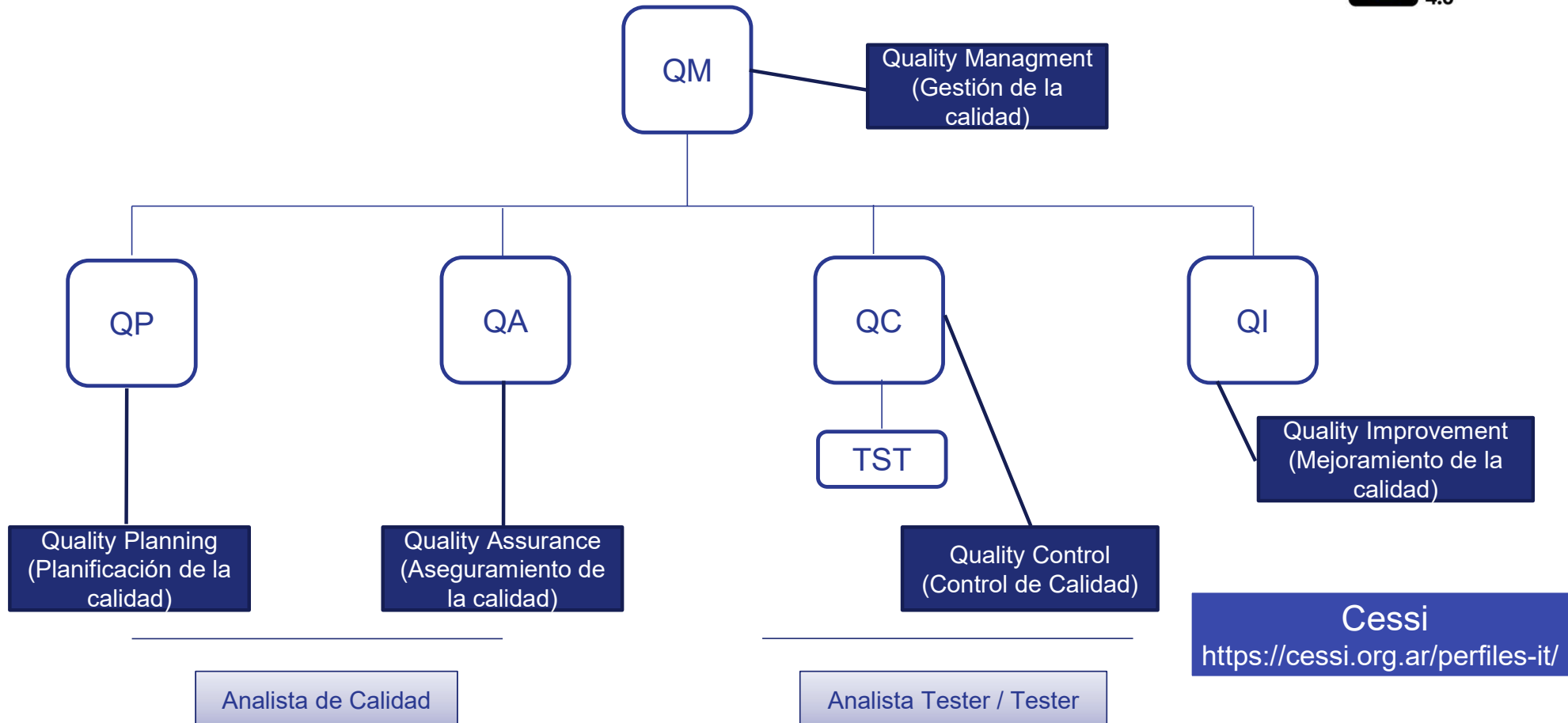
Calidad a nivel
proceso

Calidad a nivel de
producto

Calidad en uso



Diferencia entre QA, QC y Testing



Diferencia entre QA, QC y Testing

QUALITY ASSURANCE

Enfoque proactivo para **prevenir defectos** haciendo énfasis en el proceso usado en el desarrollo del producto, garantizando que lo propuesto en el plan de calidad se lleve a cabo.

Es una **actividad transversal** en el ciclo de vida del desarrollo del producto y se lleva a cabo mediante **auditoras de calidad regulares**, con el objetivo de identificar oportunamente las desviaciones entre lo planeado y lo realizado.

¿Quién es responsable del QA?

Cada miembro del **equipo involucrado en el desarrollo del producto** es responsable. Un ejemplo de actividad de QA en el desarrollo de software es la **verificación de especificaciones técnicas**.



"QA asegura la calidad de los procesos utilizados para crear productos de calidad"

VS

QUALITY CONTROL

Proceso reactivo puntual, enfocado en **identificar y corregir** los defectos encontrados en el producto ya terminado antes de ser liberado o entregado al cliente final.

¿Quién es responsable del QC?

Usualmente es realizado por un **equipo de trabajo independiente**, que se encuentra exclusivamente dedicado a probar el producto para encontrar errores. Un ejemplo de actividad de QC en el desarrollo de software es la **validación funcional**.

"QC es el proceso de probar la calidad de un producto"





Debugging

La depuración de programas es el proceso de **identificar, analizar y corregir errores** de programación. Es conocido también por el término inglés debugging, cuyo significado es **eliminación de bugs**.

Debugging

≠

Testing

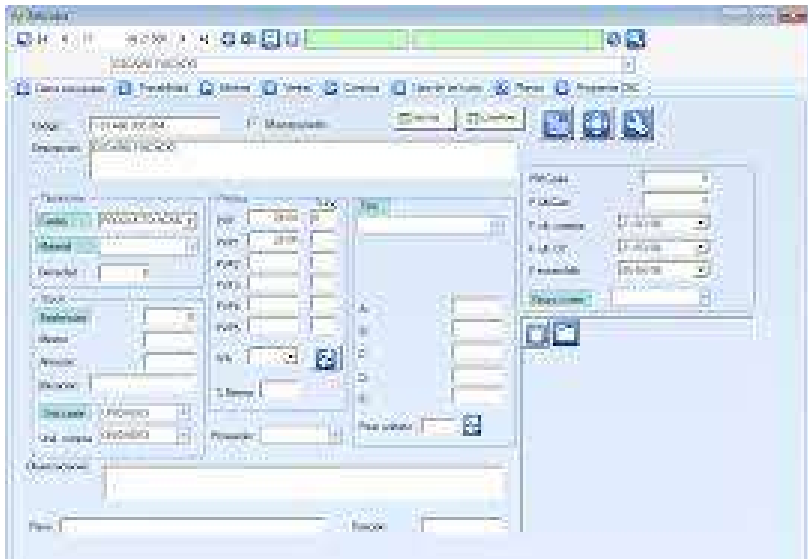
Identificar, analizar y Corregir
los defectos para eliminar los
fallos

Encontrar fallos producidos por
defectos en el software

Debugging

Testing

Entorno de Pruebas



Encontrar fallos producidos por defectos en el software

Debugging

Entorno de Desarrollo



Identificar, analizar y Corregir los defectos para eliminar los fallos

Independencia de las pruebas

La independencia de las pruebas también dependerá de

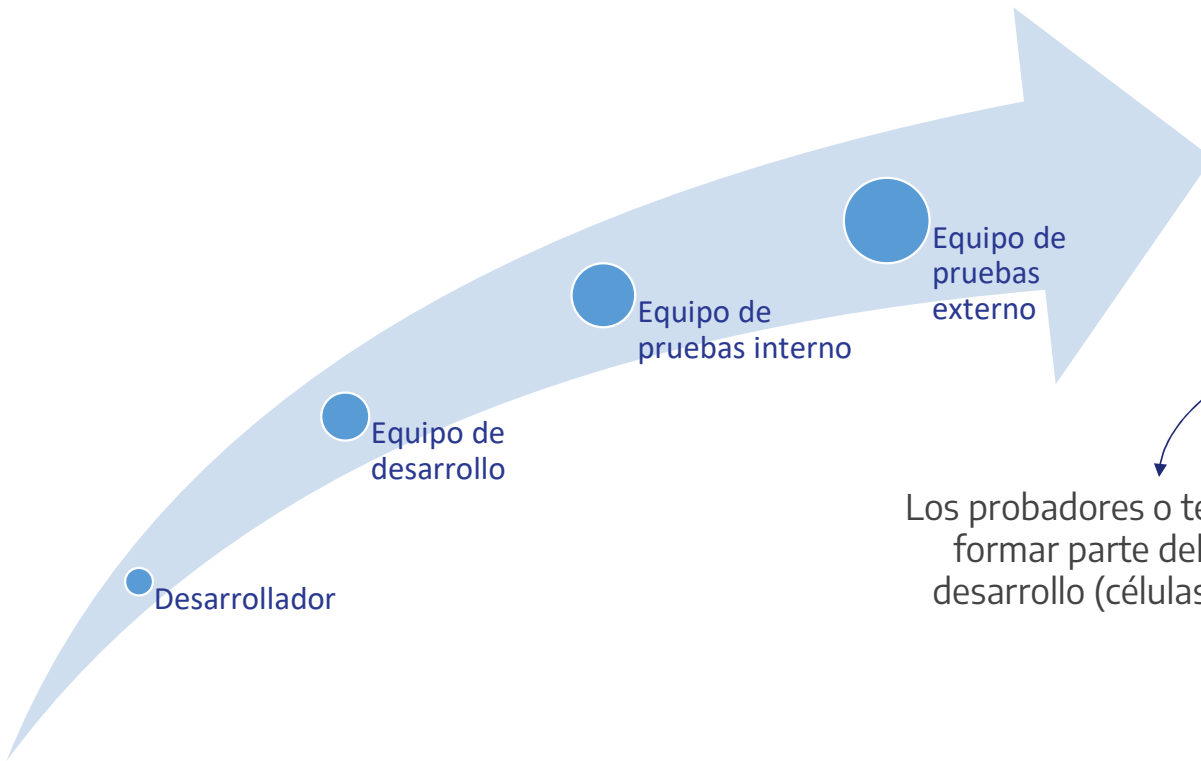
Modelo del ciclo de vida de desarrollo de software

Desarrollo Ágil

Los probadores o testers pueden formar parte del equipo de desarrollo (células de trabajo)

Los probadores o testers pueden formar parte de un equipo de testing o pruebas que es independiente al equipo de desarrolladores

El dueño del producto (product owner) puede realizar actividades de validación de las HU en cada iteración



SIETE Principios de las pruebas

1. Las pruebas ponen en evidencia defectos, no su ausencia
2. Las pruebas exhaustivas no son posibles
3. Probar en fases tempranas
4. Agrupamiento de defectos
5. Paradoja del pesticida
6. Las pruebas son dependientes del contexto
7. Falacia de la ausencia de errores

SIETE Principios de las pruebas

1. Las pruebas ponen en evidencia defectos, no su ausencia

La prueba puede demostrar la presencia de defectos, pero no puede probar que no hay defectos.

Las pruebas reducen la probabilidad de que queden defectos no descubiertos, sino se encontraran defectos, esto no significa una prueba de correctitud

2. Las pruebas exhaustivas no son posibles

No es posible probar todo. Para determinar que pruebas realizar se puede utilizar el análisis de riesgo, las prioridades y las diferentes técnicas de pruebas para centrar los esfuerzos.

3. Probar en fases tempranas

Ayuda a reducir cambios costosos (Ahorra Tiempo y Dinero). Probar en las etapas temprana del ciclo de vida de desarrollo del software, ya sean pruebas estáticas como dinámicas

4. Agrupamiento de defectos

Un pequeño número de módulos contiene la mayor cantidad de defectos

SIETE Principios de las pruebas

5. Paradoja del pesticida

Si las mismas pruebas se repiten una y otra vez, eventualmente estas pruebas ya no encontraran ningún nuevo defecto. Cambiar las pruebas y los datos de pruebas existentes.

6. Las pruebas son dependientes del contexto

La prueba se realiza de manera diferente dependiendo de los contextos de USO y de la metodología de DESARROLLO

7. Falacia de la ausencia de errores

Las organizaciones esperan que se pueda probar todo (Pero los principios 1 y 2 dicen lo contrario)
Por mas que se encuentren un gran número de defectos y se corrijan no se puede asegurar el éxito del sistema



Argentina
programa
4.0

Gracias!

Testing