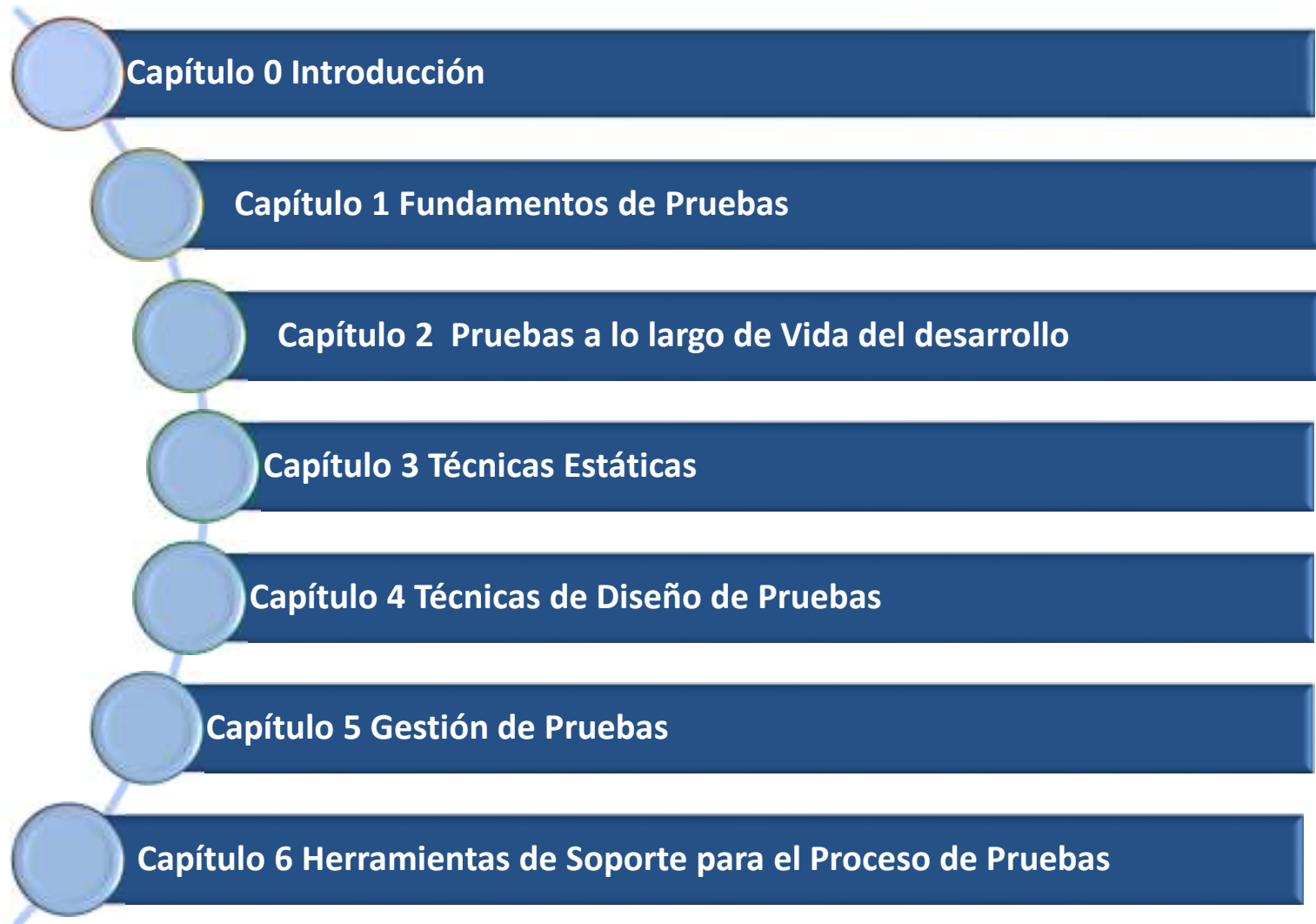
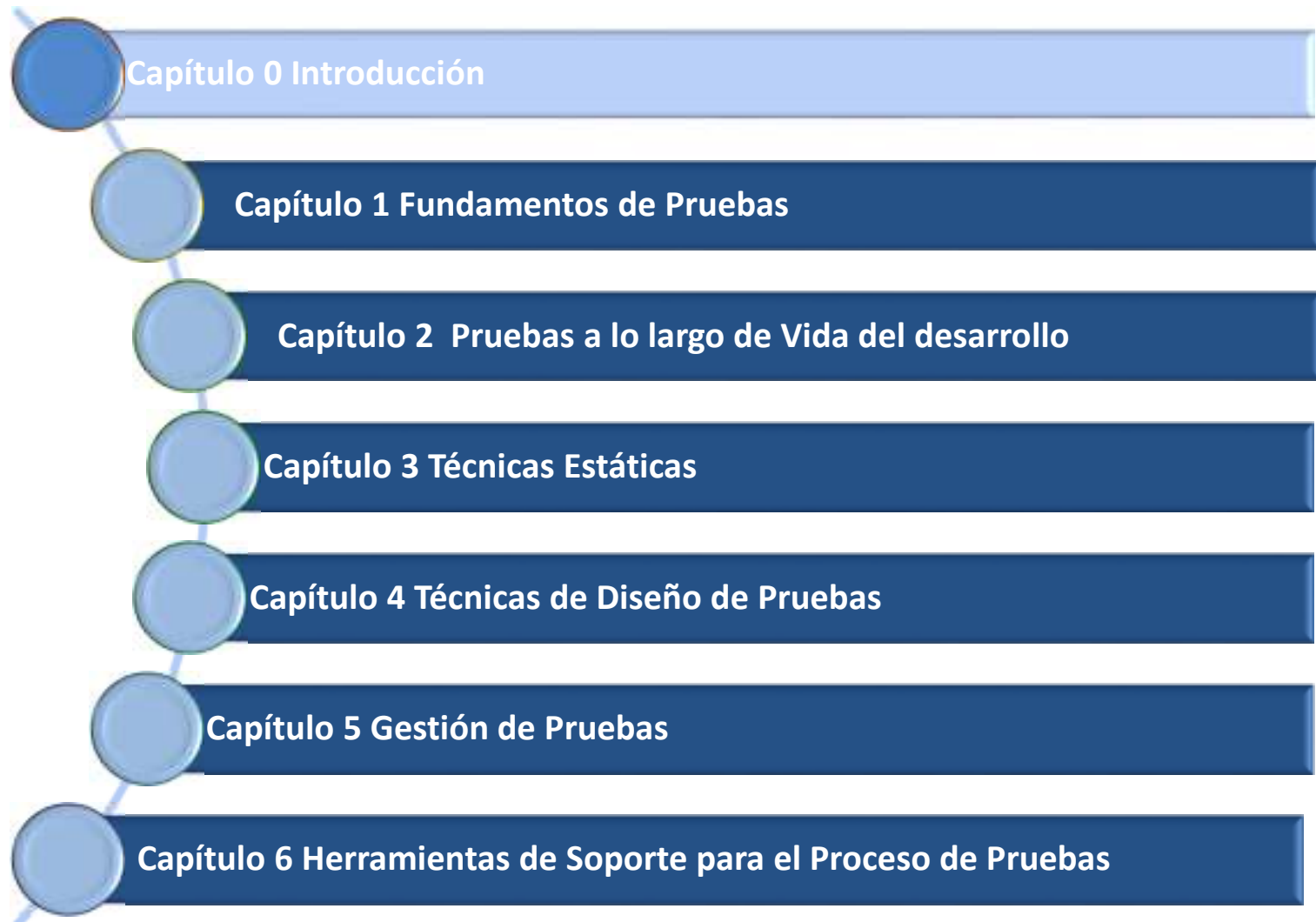


**Curso de preparación
Tester Certificado
Nivel Básico**

Agenda



Agenda



Capítulo 0 Introducción

- 0.1 Presentación
- 0.2 Antecedentes
- 0.3 Metodología
- 0.4 Evaluación
- 0.5 Objetivos del certificado
- 0.6 Audiencia



0.1: Presentación

Capítulo 0 Introducción

1- Presentación

Capítulo 0 Introducción

1- Presentación

Luis Alberto Castillo González

TESTER CONSAGRADO

▶ **Estudios**

- Ingeniería de sistemas
- Especialización en Gerencia de Proyectos de Ingeniería

▶ **Experiencia**

12 años de experiencia en proyectos de software en empresas como:

- Colsubsidio
- SQA S.A. Proyectos: Monitor, Cliente: Skandia, Flujo de Caja, Cliente: Skandia, Proveedores de Tecnología, Cliente: GS1 - SINOI, Cliente: Proexport Colombia, PAC, Plan de Ahorro Corporativo, Cliente: Skandia Facshet, Cliente: Skandia, Elempleo.com, Cliente: Leadersearch S.A. CEET
- Q-Vision S.A. Proyectos: Elempleo.com, Cliente: Leadersearch S.A. CEET, Futura, Cliente: COLFONDOS.
- Coordinador de Proyectos y Gestión del Conocimiento
- TATA Consultancy Services. Proyecto Telefónica - PL



0.2: Antecedentes

Capítulo 0 Introducción

2- Antecedentes

- ▶ Este curso esta basado en el programa de estudios de Nivel Básico Internacional de 2010 del Internacional Software Testing Qualifications Board (**ISTQB**) www.istqb.org
- ▶ Comenzó en el Reino Unido con el British Computer Society's Information System Examination (**ISEB**), 1998.
- ▶ Apoyo de **ASQF**, de Alemania en 2002 comenzó a apoyar el esquema de calificación de tester Alemán.
- ▶ Este programa esta basado en **ISEB** y en **ASQF**
- ▶ RICK CRAIG – Libro en internet – www.sqe.com
- ▶ www.testingexperience.com Jorge Díaz

Capítulo 0 Introducción

2- Antecedentes

- ▶ El certificado básico de ISTQB no expira, ni debe ser renovado
- ▶ Los Consejos/Comités de Pruebas ("Testing Boards") (locales) de cada país conforman la estructura de la organización del 'International Software Testing Qualification Board (www.istqb.org)
- ▶ Por ejemplo en España: Spanish Software Testing Board (SSTQB). en Alemania: The German Testing Board (GTB), en Hispanoamérica: Hispanic America Software Testing Qualification Board (HASTQB) www.hastqb.org conformado por Argentina, Bolivia, Colombia, Chile, Ecuador, México, Uruguay y Venezuela

Capítulo 0 Introducción

2- Antecedentes

► **Certificaciones Existentes**

- Probador Certificado Nivel Básico
- Probador Certificado Nivel Avanzado para Analista de Pruebas
- Probador Certificado Nivel Avanzado para Jefe de Pruebas
- Probador Certificado Nivel Avanzado para Analista Técnico de Pruebas
- Probador Certificado Nivel Avanzado Completo
- Probador Certificado Nivel Experto

Es pre-requisito el probador certificado nivel básico

0.3: Metodología del Curso

Capítulo 0 Introducción

3- Metodología del curso

- ▶ Teoría
- ▶ Ejercicios
- ▶ Resumen
- ▶ Simulacros de examen por capítulo
- ▶ Simulacro de examen completo

Capítulo 0 Introducción

3- Evaluación de la certificación

- ▶ Después de tomar el curso de formación tendrá las bases para presentar el examen y obtener el certificado de tester Certificado Nivel Básico ("Certified Tester, Foundation Level I").
- ▶ La evaluación es realizada por un examinador perteneciente a una organización independiente (por ejemplo iSQI* o gasq**), ya sea presencial o en línea.

<http://www6.pearsonvue.com/Dispatcher?application=VTCLocator&action=actStartApp&v=W2L&cid=517>

- ▶ El examen es de selección múltiple, con una duración de 60 minutos. Cada pregunta tiene 4 opciones de las cuales debe seleccionar solo 1.
- ▶ Para ser aprobado el examen se debe responder correctamente 26 de las 40 preguntas que tiene el examen. Equivale a un 65%.

0.4: Objetivos del Certificado

Capítulo 0 Introducción

4- Objetivos del certificado

- ▶ Obtener el reconocimiento de pruebas como una especialización esencial y profesional
- ▶ Marco de trabajo estándar – Lenguaje comun
- ▶ Subir el perfil de tester
- ▶ Aprender y promover buenas practicas de pruebas

0.5: Audiencia

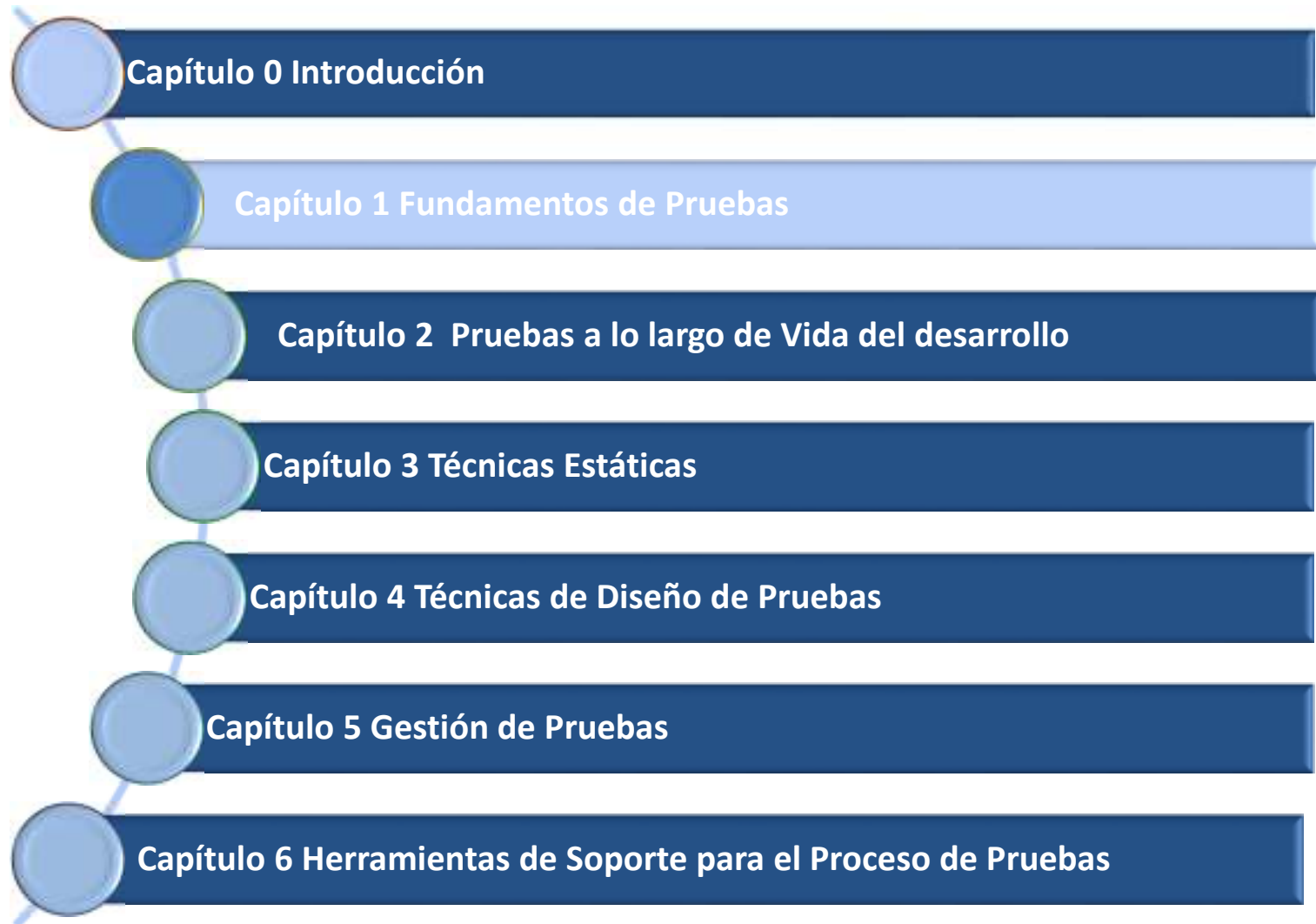
Capítulo 0 Introducción

5- Audiencia

► Esta calificación es de nivel básico para tester, pero también es apropiada para todos aquellos que deseen un entendimiento como:

- ♦ Gerente de Proyectos
- ♦ Gerentes de Calidad
- ♦ Gerentes de Desarrollo
- ♦ Analistas de Negocio
- ♦ Directores de Tecnología
- ♦ Consultores de Gerencia

Agenda



Capítulo 1 Fundamentos de Pruebas

- 1.1 ¿Por qué es necesario probar?
- 1.2 ¿Qué es probar?
- 1.3 Principios generales de pruebas
- 1.4 Proceso básico de pruebas
- 1.5 Psicología de las pruebas
- 1.6 Código Deontológico

1.1: ¿Por qué es necesario probar?

Capítulo I - Fundamentos de pruebas (K2)

1- ¿Porque es Necesario Probar?

- ▶ La Guerra del Golfo en 1991, un misil Patriot falló en la intercepción de un misil SCUD.



- ▶ 28 soldados murieron
- ▶ Esto debido a un fallo de software, que tenia un calculo errado acumulado en el tiempo.

Capítulo 1 Fundamentos de pruebas (K2)

1- ¿Porque es Necesario Probar?

- ▶ Que consecuencias tuvo este fallo?
 - Perdidas humanas
 - Perdidas materiales
 - Perdida monetarias
 - Perdida de Credibilidad en las fuerzas militares

Capítulo 1 Fundamentos de pruebas (K2)

1- ¿Porque es Necesario Probar?

- ▶ En 1990 El servicio de Larga Distancia de AT&T estuvo fuera por 9 horas debido a una actualización del software.



Capítulo 1 Fundamentos de pruebas (K2)

1- ¿Porque es Necesario Probar?

- ▶ Que consecuencias tuvo este fallo?
 - Indisponibilidad del servicio.
 - Perdidas monetarias por el no consumo de minutos
 - Retiro de clientes
 - Posible perdida de vidas, por no poder llamar en una urgencia
 - Perdida de credibilidad en la empresa

Capítulo 1 Fundamentos de pruebas (K2)

1- ¿Porque es Necesario Probar?

- ▶ Hoy en día todas **las compañías funcionan con software**, ya sea su Core del negocio o su maquinaria.
- ▶ Los **grandes proyectos** de las compañías, en su mayoría son de implementación **de software**.
- ▶ Uno de los **procesos de apoyo** mas importantes de las compañías tienen que ver con el área de tecnología.

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Términos del Glossary

Error (“Error”) Definición de la IEEE 610:

Acción humana que produce un resultado incorrecto. Ej.: Error en una sentencia de código , un dato incorrecto.

Defecto (“Defect”):

Desperfecto de un componente o sistema que puede provocar que el componente deje de ejecutar sus funciones Ej.: Una declaración incorrecta o de definición de datos es un defecto, si durante la ejecución, puede provocar un fallo en el componente o sistema.

Fallo (“Failure”):

Desviación de un componente o sistema respecto de la prestación, el servicio o resultado esperados. Ej.: Fallos ambientales como radiación, magnetismo, campos eléctricos y polución, que pueden causar fallas o influir en la ejecución del software al cambiar las condiciones del hardware.

***** Un error Introduce un defecto, un defecto causa un fallo.**

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Error
Equivocación
del programador



Defecto



FALLA (Accidente)

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

El papel de las pruebas en el desarrollo de software, mantenimiento y en producción

- ▶ Las pruebas ayudan a reducir el riesgo de:
 - ▶ Errores en operación
 - ▶ Incurrir en costos por fallos en producción
 - ▶ Perder credibilidad y cuota en el mercado
 - ▶ Exceso de pruebas ineficaces

Si son arreglados antes de pasar a producción

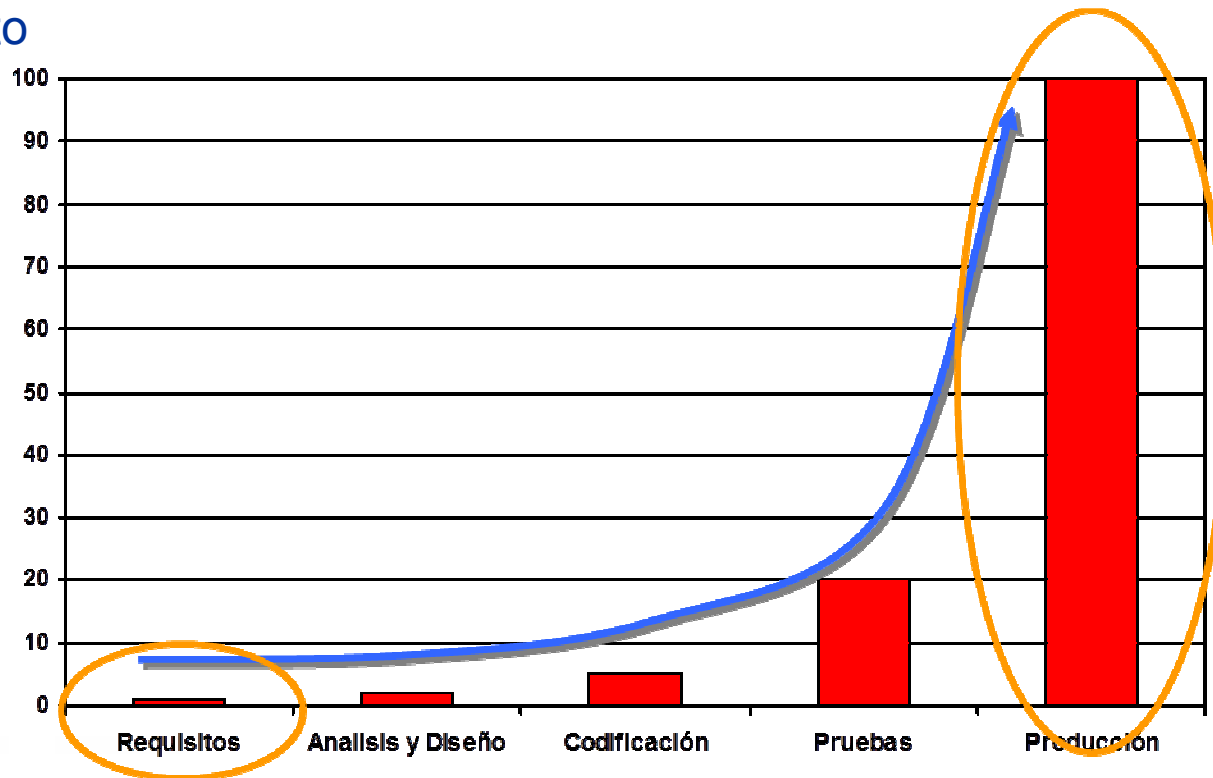
- ▶ Las pruebas verifican el cumplimiento de los requisitos contractuales, legales o estándares

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Costos en la corrección de problemas, incluye la perdida de oportunidades y clientes

- Los defectos deben ser corregidos en la fase donde fueron inyectados
- Entre mas tarde se detecte el defecto, mayor costo tendrá para el proyecto



Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Ejemplo

- ▶ Los costos de encontrar un error en la etapa de requerimientos es de 1,5 dólares.
- ▶ En la etapa de Análisis y diseño cuesta el doble que en etapa de requerimientos
- ▶ En Codificación, vale dos coma cinco veces que la etapa de análisis y diseño
- ▶ En Pruebas cuesta tres veces mas que la etapa de codificación
- ▶ En producción cuesta 5 veces mas que en codificación
- ▶ $(1,50) * (2) * (2,5) * (3) * (5)$

Etapas	Costo	Peso	Valor total
Requerimientos	\$ 1,50		
Análisis		2	\$ 3,00
Codificación		2,5	\$ 7,50
Pruebas		3	\$ 22,50
Producción		5	\$ 112,50

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Que es Calidad de Software?

Es cumplir con la totalidad de la funcionalidad y características de un producto de software, satisfaciendo las necesidades especificadas e implícitas. (ISO/IEC 9126)

Grado en el cual un componente, sistema o proceso satisface requisitos especificados y/o necesidades y expectativas del usuario/cliente. **[Según IEEE 610]**

Aseguramiento de Calidad de Software

Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad. (pruebas)

Control de calidad de Software

Técnicas y actividades de carácter operativo, utilizadas para **verificar los requisitos relativos a la calidad**, centradas en mantener bajo control **el proceso** de desarrollo y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.(Auditorias, ppqa)

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Atributos de Calidad ISO IEC 9126

Interna

Medible a partir de las características intrínsecas, como el código fuente

Externa

Medible en el comportamiento del producto, como en una prueba

En Uso

Durante la utilización efectiva por parte del usuario

La calidad de software esta constituida por atributos funcionales y no funcionales de calidad

NOTA: Para poder medir cada atributo hay que hacer diferentes tipos de pruebas

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Atributos funcionales de calidad

FUNCIONALIDAD

Correctitud: Satisface los atributos/capacidades requeridos

Compleitud: satisface todos los requisitos

Incluye según iso 9126

Adecuación /
Idoneidad

Precisión

Conformidad

Interoperabilidad

Seguridad

Atributos NO funcionales de calidad

F
I
A
B
I
L
I
D
A
D

U
S
A
B
I
L
I
D
A
D

E
F
I
C
I
E
N
C
I
A

M
A
N
T
E
N
I
B
I
L
I
D
A
D

P
O
R
T
A
B
I
L
I
D
A
D

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

► Atributos No Funcionales

- ◆ **Fiabilidad** : Madurez, Tolerancia a Fallos, Capacidad de recuperación
- ◆ **Usabilidad** : Aprendizaje, Comprensión, Operatividad, Atractividad
- ◆ **Eficiencia** : Comportamiento en el tiempo y en los recursos
- ◆ **Mantenibilidad** : Estabilidad, Facilidad de Análisis, Facilidad de pruebas, Facilidad de Cambio
- ◆ **Portabilidad** : Adaptabilidad, Co-Existencia, Capacidad de Instalación , Capacidad de remplazar

Capítulo 1 Fundamentos de pruebas (K2)

1- Porque es Necesario Probar?

Cuantas Pruebas son suficientes?

Criterios de salida

- **Pruebas basadas en riesgos:** Responsabilidad en casos de fallo, probabilidad de la ocurrencia de fallos, aspectos relativos a factores económicos y propios del proyecto.
- Tener en cuenta el **proyecto y sus restricciones** como tiempo y presupuesto
- Las pruebas deben proveer **información suficiente** para toma de decisiones
- Se debe **aprender de proyectos anteriores** (lecciones aprendidas)
- Ya se cumplieron los criterios de salida?

1.2: ¿Qué es probar?

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Objetivos de las pruebas

- **Encontrar defectos**

Comprobar la funcionalidad

- **Mejorar la confianza del nivel de calidad**

Cuando se ha pasado por proceso de pruebas, se reduce la probabilidad que el usuario encuentre defectos.

- **Suministrar información**

Adquirir conocimiento sobre los defectos en un objeto de prueba. Es importante la generación de métricas del estado de las pruebas y riesgos relativos al sistema de software

- **Evitar fallos**

**El problema no es que el aplicativo tenga defectos, el problema es no saberlo

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

¿Cuáles son los tipos de métricas claves?

- **Métricas de Progreso**
- **Métricas de Defectos**
- **Métricas de Calidad**
- **Métricas de Cobertura**

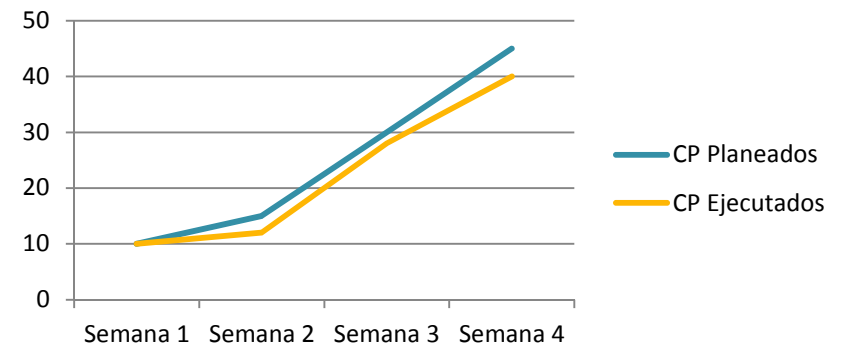


Capítulo 1 Fundamentos de pruebas (K2)

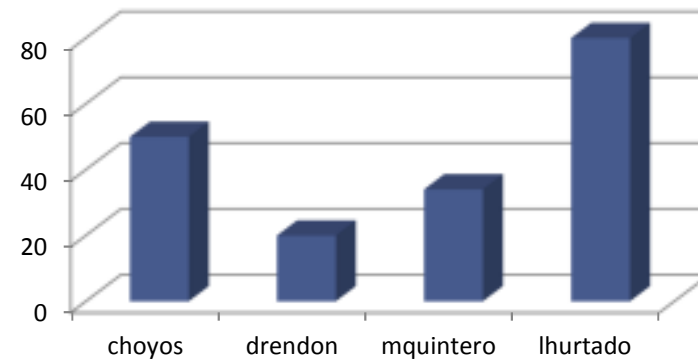
2- ¿Qué es probar?

Métricas Progreso

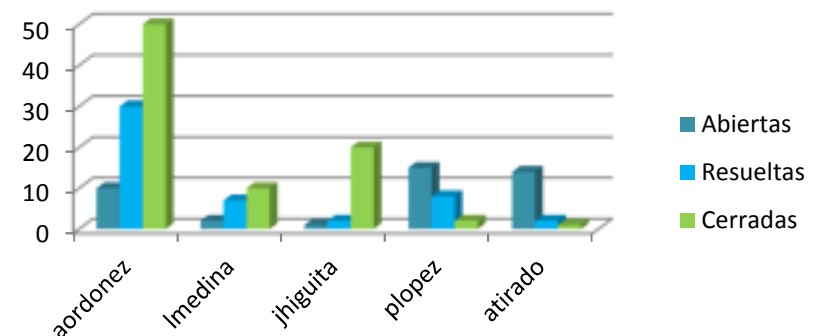
¿Cómo están avanzando desarrollo y las pruebas?



¿Cuál es la eficiencia de los testers?



¿Cuál es la eficiencia de los desarrolladores?

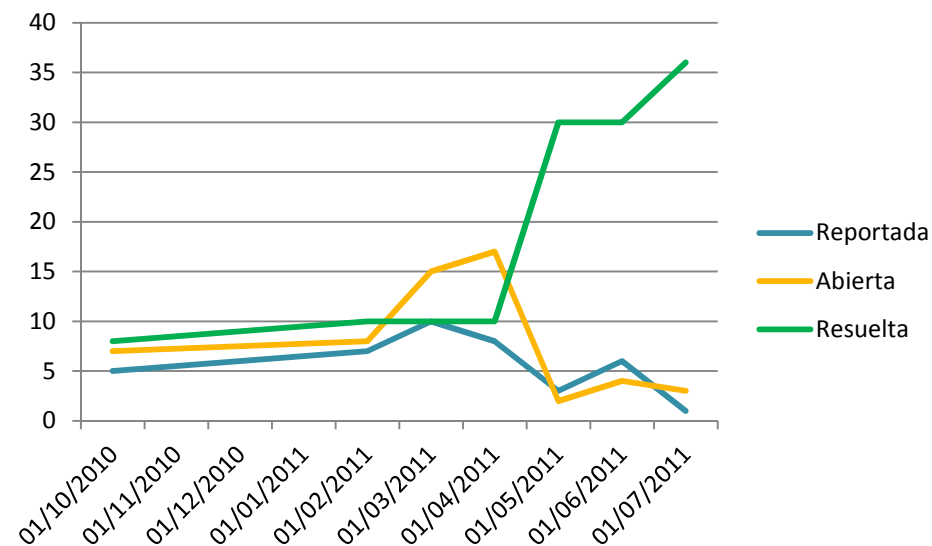


Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Métricas Defectos

¿Se solucionan lo suficientemente rápido los defectos?

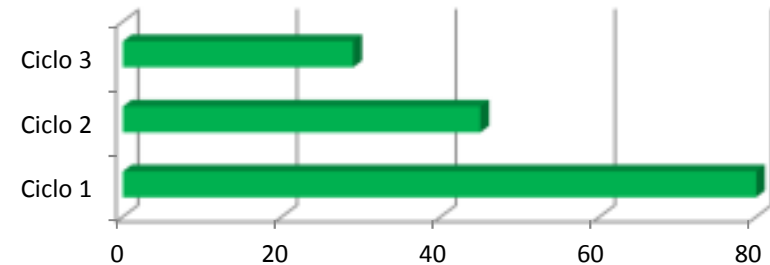


Capítulo 1 Fundamentos de pruebas (K2)

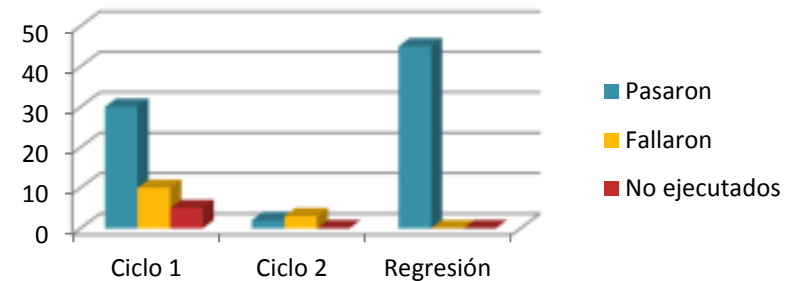
2- ¿Qué es probar?

Métricas Calidad

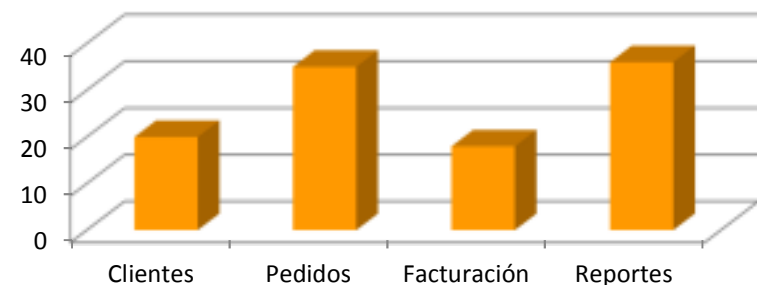
¿Cuántos defectos se reportaron por ciclo de pruebas?



¿Cuál es el estado de ejecución de los casos de prueba por cada ciclo?



¿Cuántos casos de prueba fueron diseñados para cada módulo?



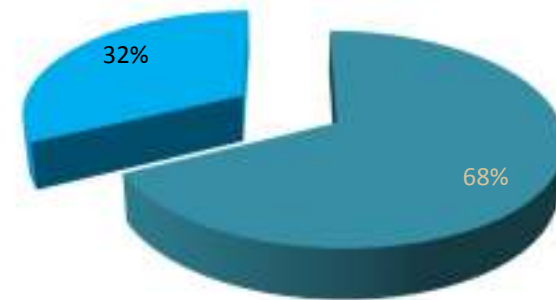
Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

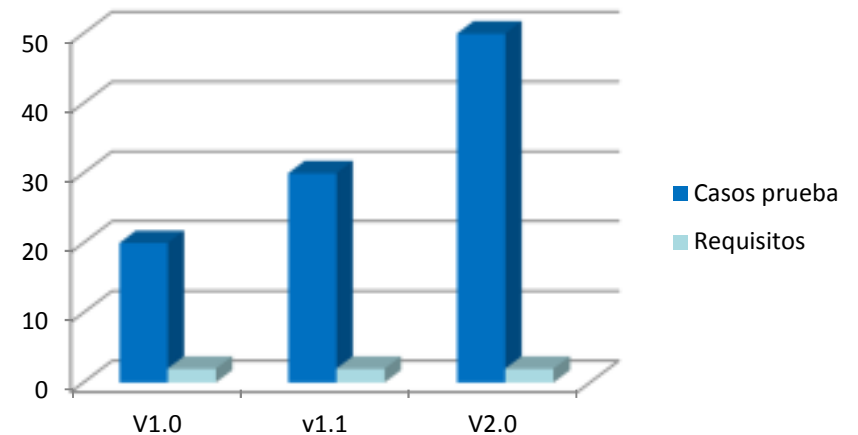
Métricas Cobertura

¿Cuántos requisitos se están cubriendo con casos de prueba?

■ Requisitos cubiertos ■ Requisitos no cubiertos



¿Cuántos casos de prueba hay por reléase?

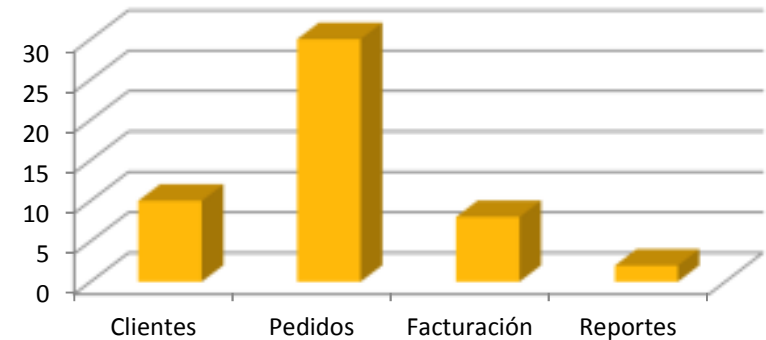


Capítulo 1 Fundamentos de pruebas (K2)

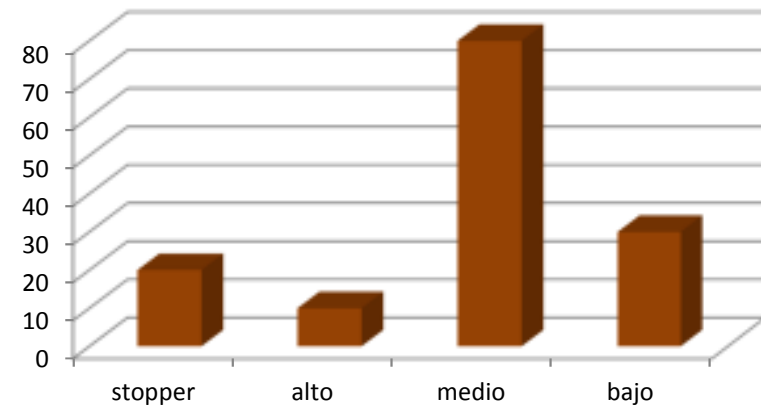
2- ¿Qué es probar?

Métricas Defectos

¿Dónde están los defectos?
¿Qué severidad tienen?
¿De que tipo son?



¿De que severidad son los incidentes encontrados?

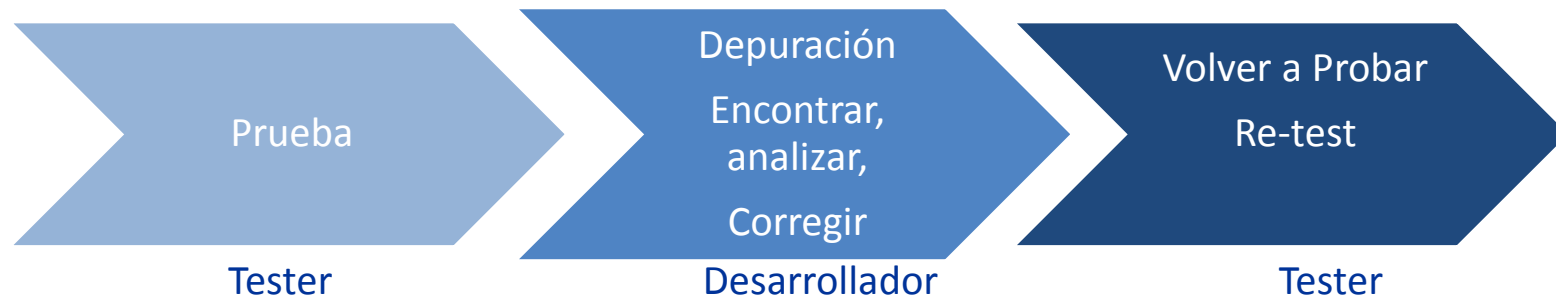


Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

► Depurar(**Debugging**)

El proceso de encontrar, analizar y eliminar las causas de los fallos en el software. (Desarrollador)



Ejemplo: <http://www.youtube.com/watch?v=XT6VFLLWob8>

<http://www.youtube.com/watch?v=XjZG6CTAsu8>

► Re-Prueba (**Re-Test**)

Prueba que ejecuta los casos de prueba que fallaron la última vez, con el fin de verificar el éxito de las acciones correctivas

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Caso de Prueba (**Test Case**) (IEEE 610)

Un conjunto de:

- Valores de entrada
- Pre-condiciones de ejecución
- Resultados esperados en la ejecución
- Pasos describiendo como ejecutar la prueba
- Post-condiciones

Desarrollados para un **objetivo particular** o condición de prueba, para ejercer una **ruta** del programa en particular o para verificar el **cumplimiento de un requisito** específico.

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Los defectos evitados, no requieren ser reparados



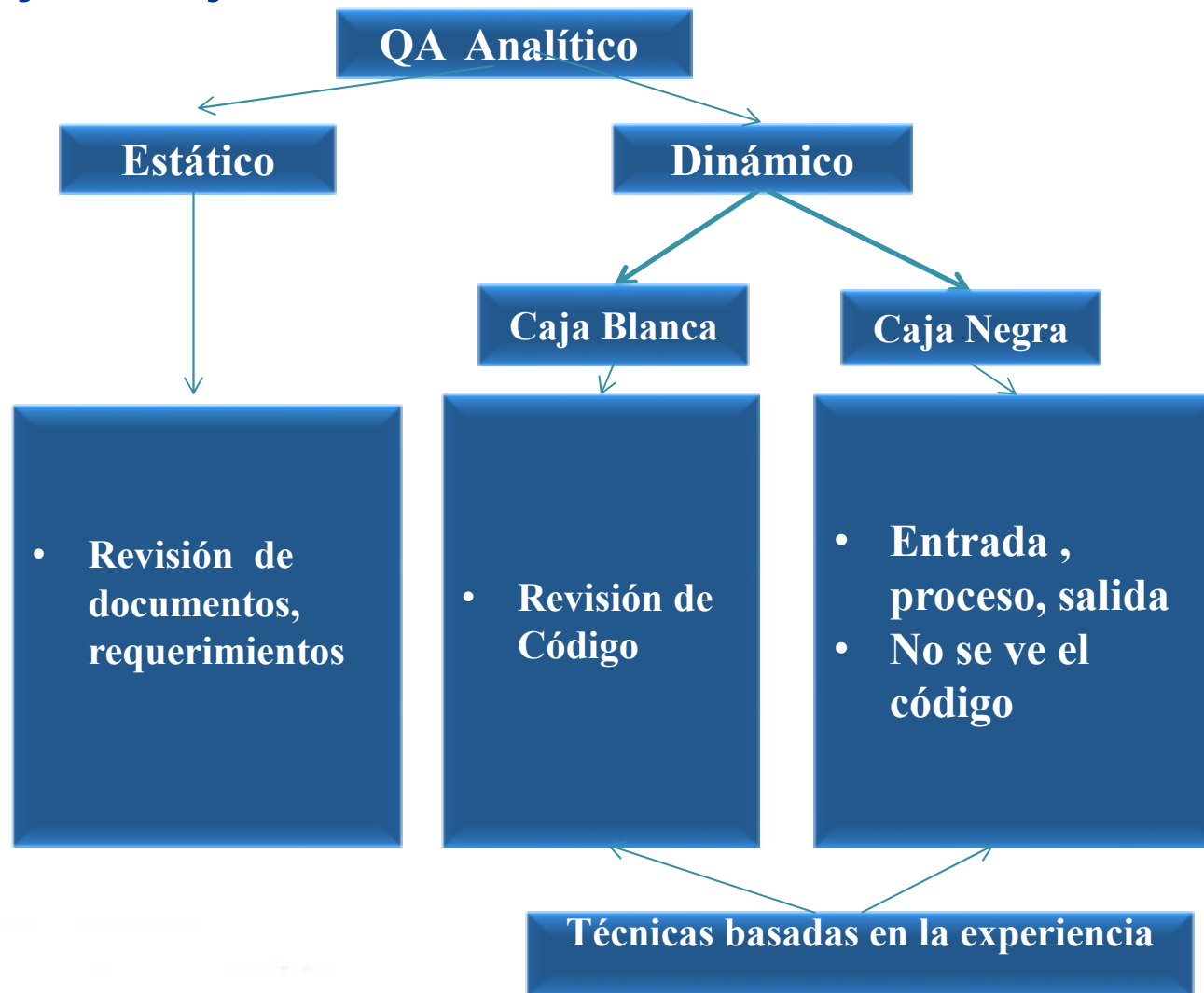
Los defectos introducidos en el pasado no deben ser repetidos

Prevenir defectos

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Las pruebas se deben realizar, tanto en la documentación, como en el código, ya sea ejecutándolo o no.



Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Requisito (**Requirement**) (IEEE 610)

Terminología

Es la descripción de la necesidad de un usuario para resolver:

- Un problema
- Alcanzar un objetivo que se debe cumplir, en un sistema
- Necesidad de satisfacer un contrato, norma, especificación u otro documento formalmente impuesto.

Describen los atributos funcionales y no funcionales

Revisión (**Review**) (IEEE Std 1028)

La evaluación de un producto o estado de un proyecto para detectar discrepancias con respecto a los resultados esperados y para recomendaciones de mejoras. Ej.: revisión técnica, inspección de documentos

Capítulo 1 Fundamentos de pruebas (K2)

2- ¿Qué es probar?

Resumen

- Un error introduce un defecto, el defecto causa un fallo
- Las pruebas tempranas disminuyen los riesgos de encontrar defectos en etapas posteriores, disminuyendo los costos de los proyectos
- El objetivo de las pruebas no solo es encontrar errores, es también mejorar la confianza en el software , dar información y evitar fallos
- El proceso de pruebas tiene muchas mas actividades que solo ejecutar el código.

1.3: Principios del proceso de pruebas de software

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Principio 1 – Las pruebas demuestran la presencia de defectos

- ▶ Las pruebas pueden mostrar que existen defectos, pero **no puede probar que no los hay**.
- ▶ Las pruebas **reducen la probabilidad** de que los defectos sin descubrir **permanezcan en el software**, pero si no se encuentran defectos no es prueba de que esta bien.

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Principio 2- las pruebas exhaustivas son imposibles

Terminología exhaustive testing

(Un enfoque de prueba en la que el conjunto de pruebas incluye todas las combinaciones de valores de entrada y las condiciones previas.)

- ▶ Probar todo no es factible, excepto por los casos triviales.
- ▶ Se debe realizar análisis de riesgo y prioridades para enfocar los esfuerzos de las pruebas

Ej.: (costos, numero de clientes, mas transacciones, transacciones mas costosas)

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Principio 3 – Pruebas tempranas

- ▶ Las pruebas no deben comenzar en la ejecución, deben ir en paralelo a las actividades del ciclo de desarrollo del software o del sistema.
- ▶ Defectos encontrados en etapas tempranas son menos costosos
- ▶ Menor probabilidad de re-trabajos
- ▶ Los requisitos deben ser probados

Principio 4 – Agrupamiento de defectos

- ▶ Un numero pequeño de módulos contiene la mayoría de defectos. Donde hay uno se encuentran varios.
- ▶ Son responsables de la mayoría de los fallos operacionales
- ▶ Se deben crear escenarios nuevos para encontrar mas defectos
- ▶ Las métricas nos ayudan a detectar esto y tomar decisiones

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Principio 5 – Paradoja del pesticida

- ▶ Si las mismas pruebas se repiten varias veces, no encontrara nuevos defectos.
- ▶ Los casos de prueba necesitan ser revisados y corregidos periódicamente
- ▶ Se deben crear nuevas y diferentes pruebas para probar diferentes partes del software o sistemas y así, potencialmente encontrar mas defectos

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Principio 6 – las pruebas dependen del contexto.

- ▶ Las pruebas se desarrollan de forma diferente en diferentes contextos.
- ▶ Por ejemplo un software de **seguridad** critica es probado diferente que un **sitio de comercio electrónico**
- ▶ Definir las pruebas según los riesgos que se detecten en cada contexto
- ▶ Es diferente probar en el ambiente de **desarrollo** que en el de **pruebas y** en el de **producción**

Principio 7 – La falacia de la ausencia de errores

- ▶ Encontrar y arreglar defectos no es de utilidad si el **sistema construido es inservible** y no llena las expectativas y necesidades de los usuarios
- ▶ Si un software que funciona perfecto no es utilizado por el usuario, realmente fue exitoso el proyecto?

Capítulo 1 - Fundamentos de Pruebas Software

03- Principios del proceso de pruebas software

Resumen

- ▶ El proceso de pruebas demuestra la presencia de defectos, no la ausencia de ellos
- ▶ No es posible realizar pruebas exhaustivas
- ▶ Pruebas tempranas
- ▶ Agrupación de defectos, donde hay un error hay mas
- ▶ Paradoja del pesticida, repetir las mismas pruebas no es efectivo
- ▶ Las pruebas dependen del contexto, en que estoy probando
- ▶ La falacia de la ausencia de errores. Debe ser utilizable

1.4: Proceso de pruebas básico

Términos

- **Criterios de salida:** Es el conjunto de las condiciones genéricas y específicas, acordadas con las partes interesadas, para permitir que un proceso que se complete formalmente. El propósito de los criterios de salida es evitar que una tarea sea considerada como completa cuando hay partes pendientes que no se han terminado. Los criterios de salida se utilizan respecto a la prueba y a cuando parar la prueba. [Gilb and Graham]
- **Incidente:** Cualquier evento ocurrido que requiere una investigación [IEEE 1008]

Términos

- **Pruebas de regresión:** Pruebas de un sistema ya probado objeto de una modificación, para garantizar que no se han introducido o descubierto defectos en partes que no han sido modificadas.
- **Base de pruebas:** Todos los documentos que interfieren en la creación de los requisitos de un componente o sistema. La documentación en la que los casos de prueba se basan.
- **Cobertura de las pruebas:** El grado, expresado como un porcentaje que especifica la cobertura que se ha ejecutado con una suite de pruebas.

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

Términos

- **Condición de prueba:** Elemento o componente de un sistema **que puede ser verificado** por uno o mas casos de prueba, Por ejemplo: una función, atributo de calidad, o elemento estructural.
- **Prueba:** El proceso de ejecución de una prueba en el componente o sistema sometido a prueba, **produce resultados reales.**

Conjunto de uno o mas casos de prueba (IEEE829)

- **Registro de pruebas:** Un registro cronológico de los datos pertinentes sobre la ejecución de las pruebas. [IEEE 829].

Términos

- **Procedimiento de pruebas:** Un documento que especifica una secuencia de acciones para la ejecución de una prueba. También es conocido como script de prueba o script de prueba manual. [IEEE 829]
- **Política de pruebas:** Un documento de alto nivel que describe los principios, enfoques y objetivos principales de la organización respecto a las pruebas.
- **Estrategia de pruebas:** Una descripción de alto nivel de los niveles de prueba a realizar y la forma de ejecutar dentro de los niveles de una organización o programa (uno o varios proyectos).

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

Términos

- **Juego de pruebas:** Un conjunto de varios casos de prueba de un componente o sistema sometido a prueba, donde el resultado de la ejecución a menudo se utiliza como condición previa para el siguiente.
- **Informe resumen de pruebas:** un documento que resuma las actividades de prueba y los resultados. También contiene una evaluación de los elementos de prueba correspondientes contra los criterios de salida. [IEEE 829].
- **Utensilios de pruebas (testware):** Los artefactos producidos durante el proceso de prueba requeridos para planear, diseñar y ejecutar pruebas, como documentación, guiones, los insumos, los resultados esperados, hasta set-y desaparecen- procedimientos, archivos, bases de datos, medio ambiente, y cualquier otro software o los servicios públicos utilizados en pruebas. [Según Fewster y Graham]

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

- El proceso de pruebas es mas complejo que la sola tarea de ejecutar casos de prueba.
- El proceso básico de pruebas consiste de las siguientes fases:



- Las fases del proceso de pruebas se pueden **sobreponer** y/o ocurrir al mismo tiempo
- Para que las pruebas sean efectivas se deben **seguir todas las fases**.
- Las fases de pruebas soportan “**Backtraking**” que permite Regresar a una fase ya ejecutada

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

► 1.4.0.



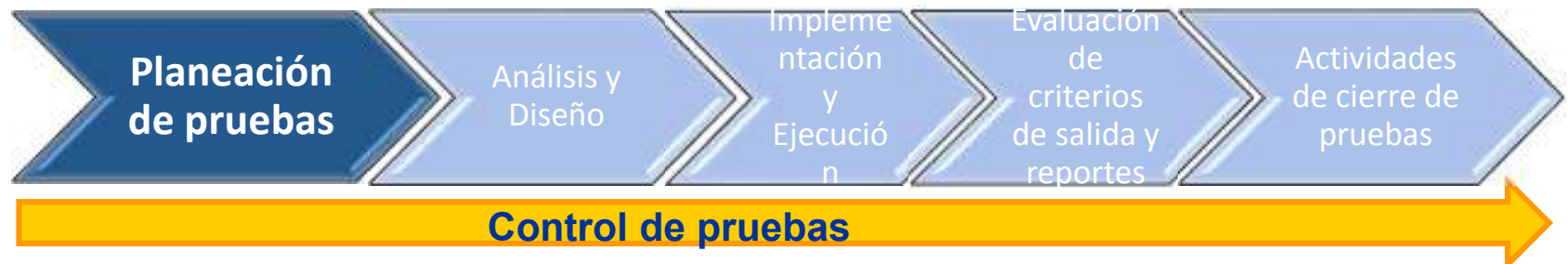
- El control es la actividad continua de comparar el **progreso actual vs el planeado**
- El **control** de la prueba **reporta** el estado incluyendo las **desviaciones** del plan.
- El **control** ayuda para **tomar acciones correctivas y decisiones** en caso que se desvíe la prueba de su objetivo o misión
- Se debe monitorear el proyecto hasta el final

*** Ver mas detalle capítulo 5

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

► 1.4.1.



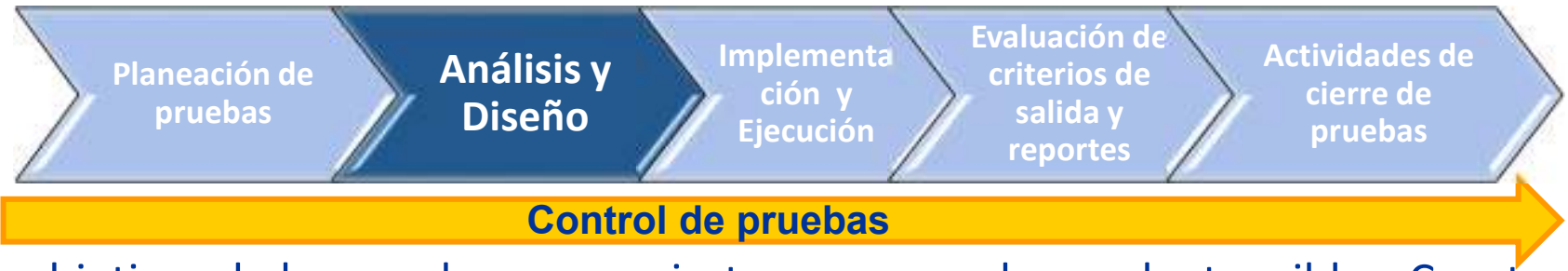
- La planeación es la actividad de verificar la **misión**, definen **los objetivos** y se especifican las **actividades** de las pruebas.
- Se determina el alcance (Estimación tiempos, recursos y presupuesto) de la prueba y los riesgos
- Se establece la estrategia de pruebas y las herramientas a utilizar
- Se determinan los criterios de salida
- La planeación **puede ser actualizada** teniendo en cuenta los resultados de las actividades de control

*** Ver mas detalle capítulo 5

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

► 1.4.2.



- Los objetivos de las pruebas se convierten en casos de prueba tangibles. Consta de las siguientes actividades:
 - **Revisar la base de pruebas**(requerimientos, arquitectura, diseño e interfaces).
 - **Identificar y priorizar las condiciones de prueba** basados en análisis de elementos de prueba, la especificación, comportamiento y estructura. O la viabilidad de crear datos de prueba
 - **Diseñar y priorizar casos de prueba e Identificar los datos de prueba** para cubrir los casos de prueba.
 - Diseñar la configuración del **ambiente** de prueba e identificar la **infraestructura y herramientas** requeridas.
 - Evaluar la **testeabilidad** de prueba de la base de prueba y los objetos de prueba

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

▶ 1.4.3.



- Es la fase en la que se ejecutan los casos de prueba **combinando en un orden particular**, con todos los elementos que se han preparado previamente.
- Se debe contar con el **ambiente configurado**.
- Esta fase cuenta con las siguientes actividades principales:
 - **Ejecutar ya sea manual o automático**, priorizando los casos de pruebas y procedimientos.
 - Preparar y validar ambiente de pruebas, datos de prueba, arnés de prueba, scripts de prueba automatizados, entre otros.

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

▶ 1.4.3.



- Crear los juegos de pruebas (**Test suite** - escenario).
- Registrar los **resultados de la ejecución**, la versión del software
- **Comparar** los resultados obtenidos con los esperados.
- Reportar incidentes y analizarlos (datos, de ambiente, de código)
- Repetir las actividades de pruebas, para **verificar que los defectos se corrigieron y no se inyectaron nuevos**. Ejemplo pruebas de re-test o pruebas de regresión.

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

▶ 1.4.4.



- ▶ Se evalúan las respuestas obtenidas contra los objetivos definidos
- ▶ Se debe realizar la evaluación para cada nivel de prueba.
- ▶ La evaluación de pruebas tiene las siguientes actividades:
 - **Verificar** los registros de las pruebas con **los criterios de salida**, especificados en la planificación de las pruebas.
 - Evaluar si **se requieren mas pruebas** o si los criterios de salida deben ser modificados.
 - Realizar un **informe del resumen** de las pruebas y compartirlo con las personas involucradas.

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

▶ 1.4.5.



- ▶ Se consolida la información de la prueba y se recopilan las experiencias como **lecciones aprendidas**.
- ▶ Se elabora informe **post-mortem** de la prueba
- ▶ Se **generan estadísticas**, conclusiones de los resultados obtenidos.

Capítulo 1 - Fundamentos de Pruebas software

04 - Proceso de pruebas básico

▶ 1.4.5.



▶ Las actividades de cierre de pruebas incluyen las siguientes actividades:

- Verificar entregables planeados vs entregados, cierre de reportes de incidentes o registro de cambios abiertos y la documentación de aceptación.
- Dar por terminado y archivar soporte de pruebas, el ambiente de pruebas y la infraestructura de pruebas para uso posterior.
- Entrega de utensilios de pruebas para la manutención de la organización.
- Analizar las enseñanzas aprendidas para futuras versiones y proyectos y mejorar la madurez de las pruebas.

Resumen

El proceso de pruebas se divide en las siguientes Fases:

- ▶ **Planeación de las pruebas:** Alcance de la prueba (recursos(personas, herramientas, tipos de prueba), tiempo)
- ▶ **Análisis y Diseño de casos de prueba:** Análisis de lo que se necesita y diseño de los casos de prueba y escenarios.
- ▶ **Ejecución:** Consecución de los datos y ejecución de pruebas comparando resultados.
- ▶ **Evaluación de criterios de salida:** Se revisa si se cumplieron los criterios de salida y se realiza un informe del estado de los defectos.
- ▶ **Actividades de cierre:** Lecciones aprendidas, verificación de entregables, cierre de incidencias.
- ▶ **Control de pruebas:** Es transversal a todo el proceso de pruebas. Comparo lo planeado vs lo ejecutado y sus desviaciones

1.5: Psicología de las pruebas

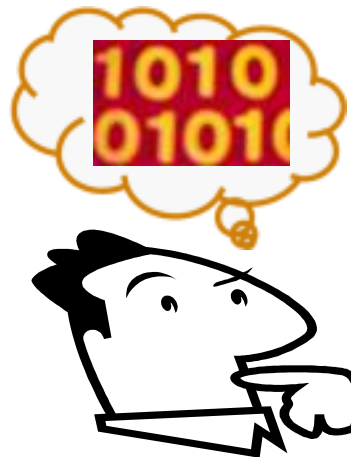
Términos

- ▶ **Predicción de error(error guessing):** Una técnica de diseño de pruebas donde se utiliza la experiencia del probador para anticipar cuales defectos pueden estar presentes en el componente o sistema bajo prueba como un resultado de los errores cometidos, y para diseñar las pruebas específicamente para exponerlos.
- ▶ **Independencia:** La independencia de la prueba: La separación de responsabilidades, que fomenta la realización de pruebas objetivas. [DO-178B]

Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

- ▶ El modo de **pensar cambia de acuerdo al rol** asumido dentro del proyecto de desarrollo de software.
- ▶ La psicología de pruebas esta relacionada al **modo de pensar** y a la **relación interpersonal** que se manejan durante el proyecto, y así lograr una cordial relación entre los Tester y los desarrolladores.



Desarrollador



Tester

Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

Tester Vs Desarrollador

- ▶ Se ha mantenido la creencia que **el tester es el enemigo** natural del **desarrollador**, puesto que a nadie le gusta que pongan en evidencia los defectos.
- ▶ **Se debe pensar en equipo** puesto que la finalidad independiente de la óptica siempre será obtener un **producto de software de calidad**.
- ▶ Para trabajar en equipo se pueden tener las siguientes consideraciones:
 - Colaborar en vez de crear disputas
 - Evidenciar que el objetivo es común
 - Ser neutral a la hora de informar cualquier cambio o error
 - Ponerse en el lugar del otro



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología en el proceso de pruebas

- ▶ **Actitudes que debemos tener en cuenta**
- ▶ El encuentro de errores se puede tomar como una **crítica contra el autor** por esto es muy importante ver el encuentro de errores como **una actividad constructiva y de beneficio** para las dos personas

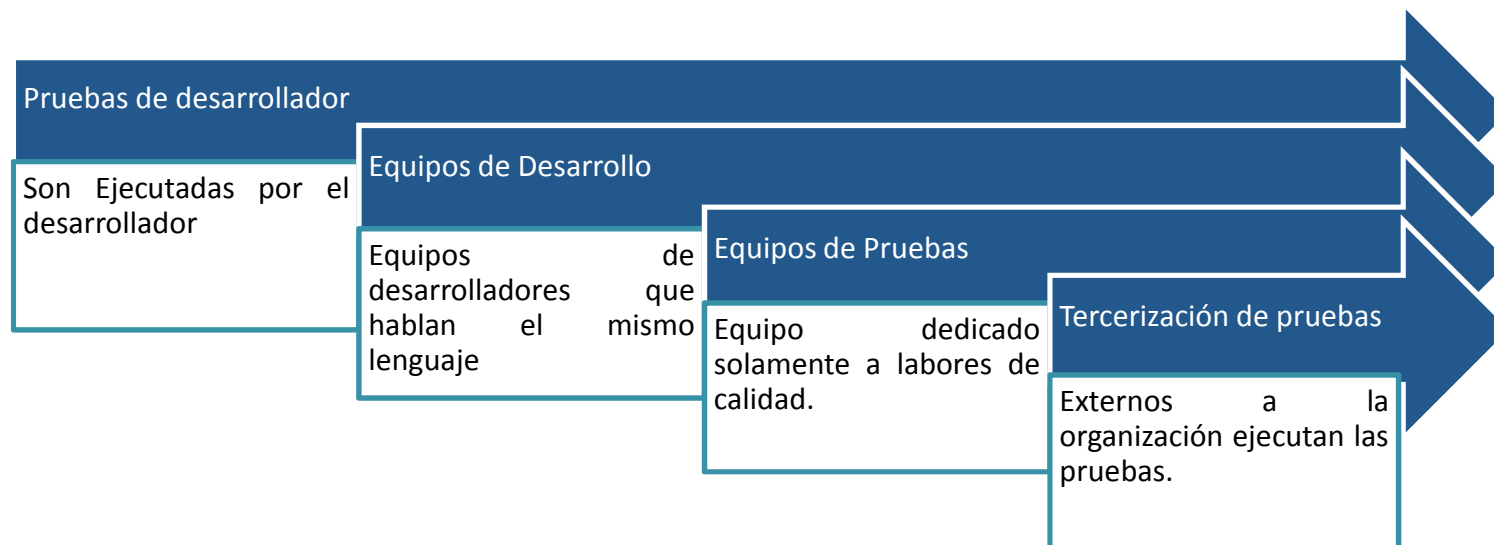


Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

Pruebas independientes

- ▶ La separación de las responsabilidades en el proceso de pruebas **apoya/promueve la evaluación independiente** de los resultados de las pruebas.
- ▶ La independencia de las pruebas pueden ser analizados de forma independiente desde el frente en el que son tratadas.



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

Pruebas de desarrollador

- ▶ Conoce el funcionamiento de la aplicación, ejecuta la prueba en la mayoría de los casos de forma como debería funcionar en los casos correctos, **omiten los datos inválidos**
- ▶ Desarrolla un aprecio a su creación, la principal falencia es que no es fácil que se vean los defectos. **(Tiende a pasar por alto sus propios defectos)**
- ▶ **No se requiere transferir el conocimiento** para el desarrollo de la prueba.
- ▶ Se convierte en **“arte y parte”**, y los defectos propios son mas difíciles de encontrar.

Pruebas de desarrollador

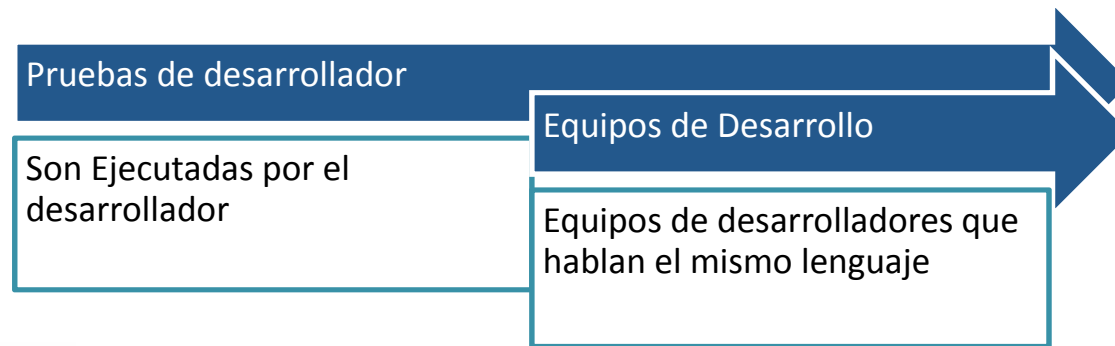
Son Ejecutadas por el desarrollador

Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

Pruebas con equipos de desarrollo

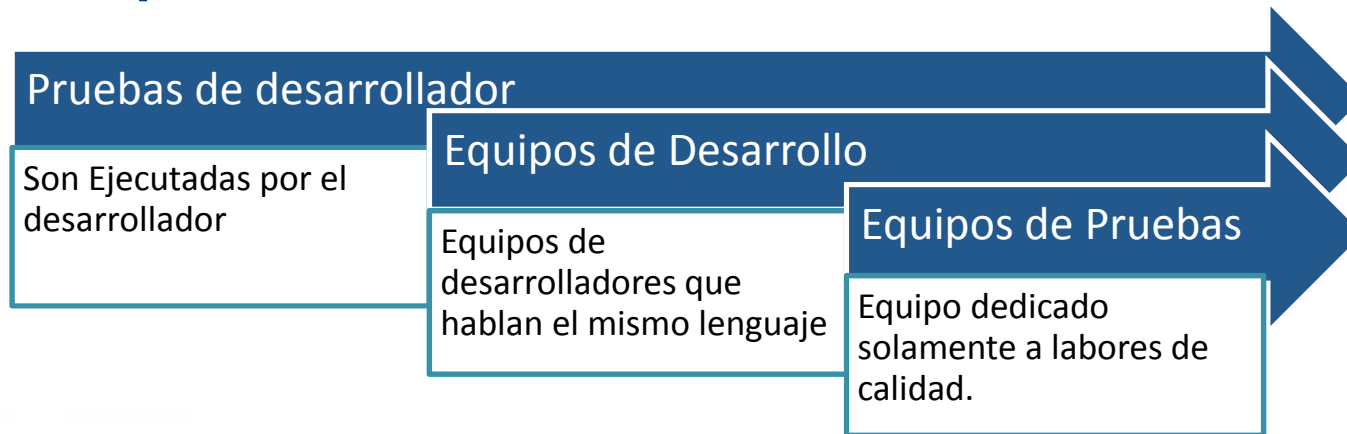
- ▶ Son ejecutadas por equipos de desarrollo, los cuales **hablan el mismo idioma**.
- ▶ Se mantienen **bajos los costos** de transferencia de conocimiento.
- ▶ Los equipos de desarrollo en la mayoría de los casos **recurren a las mismas practicas de desarrollo lo cual hace difícil detectar defectos** recurrentes de programación.
- ▶ Los defectos no siempre serán efectivos por analizar la aplicación a nivel de código y no a nivel de resultados. Ej.: Hace lo mismo de dos formas diferentes, pero ¿cual de ellas es la correcta?



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología en el proceso de pruebas

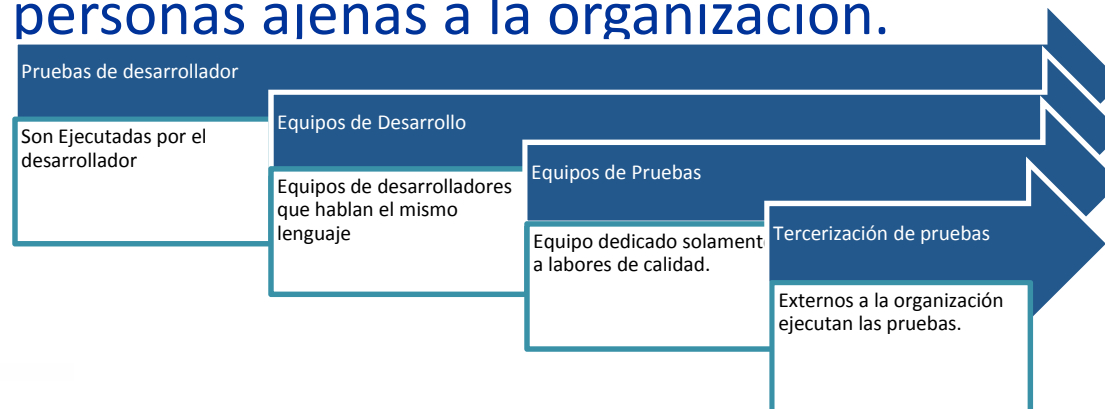
- ▶ **Pruebas con equipos de Pruebas**
- ▶ Son equipos de pruebas, **de la misma organización**.
- ▶ Se dedican únicamente a **validar** la calidad de los productos de software **de forma independiente** de las áreas del proyecto.
- ▶ Los **costos de transferencia de conocimiento son bajos** por tratarse de miembros de la organización.
- ▶ El equipo de pruebas puede intervenir en **cualquiera de los niveles de prueba**.



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

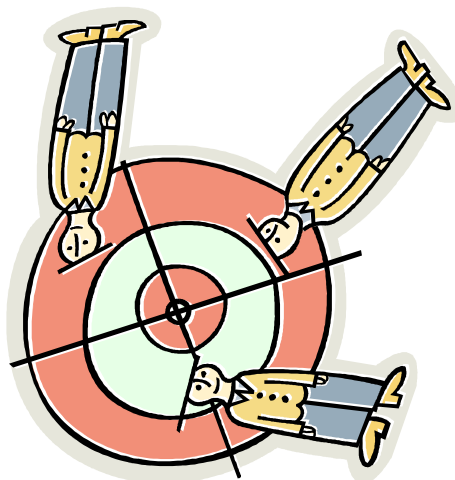
- ▶ **Pruebas con equipos externos de Pruebas**
- ▶ Son equipos de pruebas **de un tercero** o una compañía diferente, también conocido como tercerización - outsourcing
- ▶ Se alcanza el nivel de independencia deseado para tener una **visión objetiva** de la aplicación.
- ▶ **Son expertos** que se dedican e invierten en investigación y metodología
- ▶ **Los costos** de transferencia de conocimiento **son mayores** por tratarse de personas ajenas a la organización.



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

- ▶ **Los objetivos del proyecto**
- ▶ Cada uno de los equipos deberá **alinear los objetivos** propios para que concuerden con los fijados por el líder para el proyecto.
- ▶ Es importante definir al inicio del proyecto los objetivos de las pruebas “ **Es para encontrar defectos**”
- ▶ Mantener la cordialidad entre los equipos de desarrollo y de pruebas.



Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

- ▶ **Características de un buen probador**
- ▶ Debe ser **curioso**.
- ▶ Debe manejar un **pesimismo orientado** a cubrir ocurrencias poco comunes.
- ▶ **Aptitudes para la comunicación constructiva.**
- ▶ Experiencia para poder realizar la **predicción del error**.
- ▶ Atención al detalle , ojo critico
- ▶ La comunicación debe ser constructiva.

Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

- ▶ **Comunicación asertiva**
- ▶ **Colaboración en vez de disputa** – recordar que todos tienen los mismos objetivos
- ▶ **Comunicar los defectos de forma neutral.** Enfocada a los hechos sin criticar la persona. Basarse en los hechos.
- ▶ Entender la frustración de la contraparte (desarrollador) al comunicarle sus errores. **Poder ponerse en los zapatos del otro**, tanto del usuario como del desarrollador
- ▶ **Confirmar que la persona entendió** lo que usted dijo y viceversa

Capítulo 1 - Fundamentos de Pruebas Software

05 - Psicología de las pruebas

- ▶ **Resumen**
- ▶ **EL objetivo deberá ser el mismo** para cada uno de los equipos involucrados en el proyecto.
- ▶ La labor del equipo de pruebas **no es destructiva, es constructivo** el detectar errores y comunicarlos mejorando la calidad del producto de software.
- ▶ La independencia de las pruebas aumenta la calidad del producto de software.
- ▶ El tester debe tener habilidades de comunicación, atención al detalle, curioso.

1.6: Código Deontológico

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

El código de ética identifica las diferentes relaciones entre los involucrados en el proceso de pruebas de software y los individuos, grupos y organizaciones que participan en el proyecto de pruebas.

El código deontológico es necesario para asegurar que la información es utilizada de forma apropiada.

PÚBLICO: Los probadores de software actuarán en coherencia con el interés público; aceptar la completa responsabilidad de su trabajo mitigando sus propios intereses, los del empleador, los del cliente y los de los usuarios con el bienestar público.

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

CLIENTE Y EMPLEADOR: Los probadores de software certificados actuarán en el mejor de los intereses de su cliente y empleador. Por ejemplo utilizar la propiedad de un cliente o empleador solo de manera adecuadamente autorizadas y con el consentimiento y conocimiento de éste.

PRODUCTO: Los probadores de software certificados garantizarán que los entregables que suministran se ajustan a los mas altos estándares de calidad. Por ejemplo garantizando objetivos adecuados y alcanzables para el proyecto de pruebas que trabajen.

CRITERIO: Los probadores de software certificados mantendrán la integridad y la independencia en su criterio profesional.

- ▶ Moderar todos los juicios técnicos por la necesidad de amparar y mantener valores humanos.
- ▶ Firmar sólo los documentos preparados bajo su supervisión o dentro de sus áreas de competencia, y con los que están de acuerdo.

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

GESTIÓN: Los jefes y líderes de pruebas de software certificados suscribirán y fomentarán un enfoque ético en la gestión de las pruebas de software.

- ▶ Garantizar una buena gestión en cualquier proyecto en el que trabajen, incluyendo procedimientos efectivos para promover calidad y reducción del riesgo.
- ▶ Garantizar que se informa a los empleados de los estándares antes de adherirse a ellos.
- ▶ Garantizar que los empleados conocen las políticas y los procedimientos del empleador para la protección de las claves de acceso, ficheros y otra información que sea confidencial para el empleado o para otros.
- ▶ Asignar trabajo sólo después de tener en cuenta la educación y la experiencia, teniendo en cuenta el deseo de mejorar tal educación y experiencia.

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

PROFESIÓN: Los probadores de software certificados aumentarán la integridad y la reputación de la profesión en coherencia con el interés público.

- ▶ Ayudar a desarrollar un ambiente organizativo favorecedor de un comportamiento ético.
- ▶ Promover el conocimiento general de la metodología de pruebas.
- ▶ Diseminar el conocimiento de pruebas de software mediante la participación en organizaciones profesionales, reuniones y publicaciones.

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

COMPAÑEROS: Los probadores de software certificados serán equitativos y apoyaran a sus compañeros fomentando la cooperación con los desarrolladores de software.

- Animar a los compañeros a adherirse a este Código.
- Ayudar a los compañeros en el desarrollo profesional.
- Reconocer completamente el trabajo de otros y abstenerse de atribuirse méritos que no son propios.
- Revisar el trabajo de los demás de forma objetiva, sincera y convenientemente documentada.
- Tratar justamente las opiniones, preocupaciones o quejas de un compañero.
- Ayudar a los compañeros en el conocimiento completo de los estándares de trabajo, incluyendo políticas y procedimientos para proteger claves de acceso, ficheros y otra información confidencial, y medidas de seguridad en general.

Capítulo 1 – Fundamentos de pruebas

Tema 1.6 – Código Deontológico

NIVEL INDIVIDUAL: Los probadores de software certificados participaran en un aprendizaje a lo largo de toda la vida por que lo que respecta a la práctica de su profesión y fomentarán la adopción de un enfoque ético en la práctica de su profesión.

- ▶ Mejorar su conocimiento de los avances en el análisis, la planeación, el diseño, ejecución en pruebas del software y documentos relacionados, junto con la gestión del proceso
- ▶ Mejorar su capacitación para crear software de calidad, seguro, fiable y útil, con un coste y en un plazo razonables.
- ▶ Mejorar su capacidad para producir documentación y casos de prueba precisos informativos y correctamente escritos.



Simulación Capítulo