

# Entrenamiento: Pruebas de Aceptación de Usuario



# TEMARIO

Reglamento

Presentación de Participantes

Objetivo del curso

Conceptos Básicos de Pruebas

Objetivo de las Pruebas UAT

Metodología Genérica de Pruebas – Metodología de desarrollo Cascada

Fase de Pruebas de Aceptación de Usuario – Metodología de desarrollo Cascada

Metodología Genérica de Pruebas – Marco de trabajo Ágil

Fase de demostración (Demo) – Marco de Trabajo Ágil

Gestión de Pruebas UAT: Estimar, Planear, Iniciar, Realizar, Controlar y Monitorear, Reportar,

Medir y Cerrar pruebas UAT

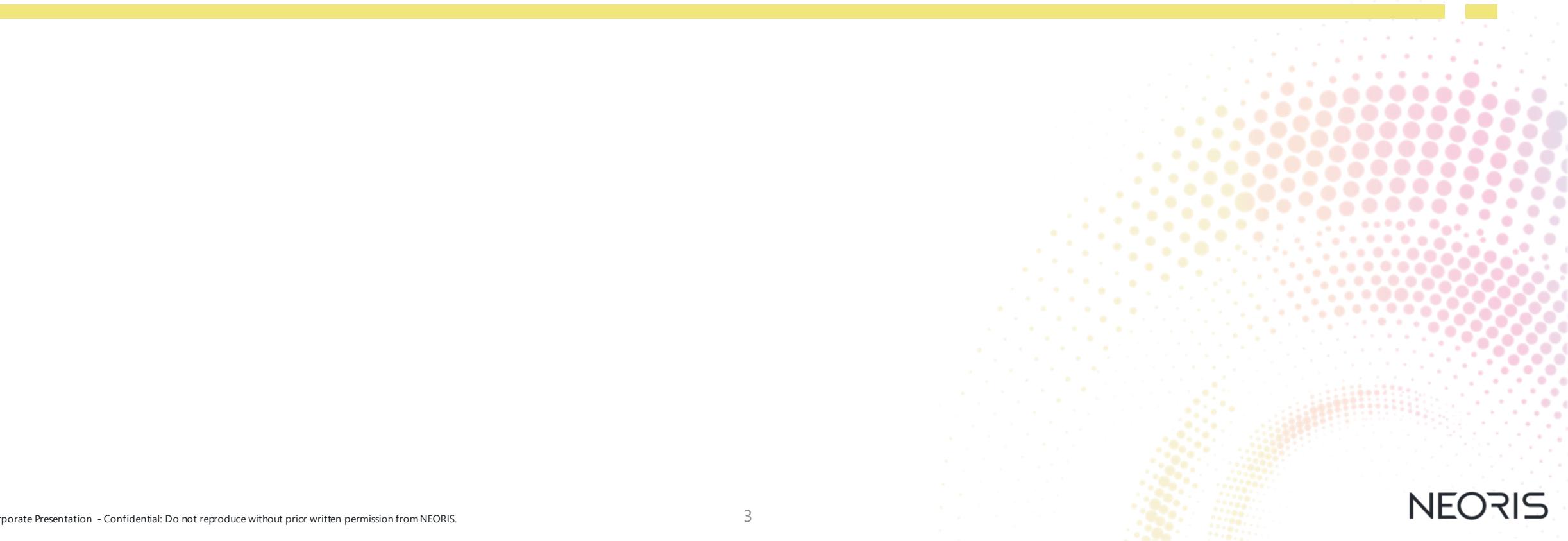
Evaluación





# REGLAMENTO

---



NEORIS



## REGLAMENTO

- Iniciar y terminar a tiempo.
- Alta participación de todos.
- Si se requiere tomar una llamada o contestar un mensaje por celular se puede hacer fuera del salón.
- Escuchar atenta y respetuosamente.
- Levantar la mano para preguntar en forma ordenada y poder resolver todas las dudas.
- En caso de no tener una respuesta a la mano, se documentará como pendiente y la respuesta se revisa la sesión de dudas.



# PRESENTACIÓN

---



NEORIS



## PRESENTACIÓN

- Nombre
- Puesto
- Departamento
- ¿Que esperas del curso?
- ¿Qué tanto consideras que conoces sobre la disciplina de Testing?
- ¿Qué tanto consideras que conoces sobre el proceso de Pruebas de Certificación de Usuario?



# OBJETIVO



## OBJETIVO

El Objetivo de este curso es que el asistente:

Comprenda y aplique la metodología de pruebas de Aceptación de Usuario en un modelo de desarrollo de Cascada.

Comprenda y aplique la metodología de pruebas de Aceptación de Usuario en un marco de trabajo Ágil.

Estandarice su modelo de trabajo.

Sea más eficientes en su operación, reduciendo tiempos en la fase de UAT y minimizando errores en productivo.



# CONCEPTOS GENERALES

---





# Que es QA (Quality Assurance) y que es Testing

### Quality Assurance (QA)

- Orientada a prevenir errores.
- Revisa y asegura apego al modelo metodológico de cualquier disciplina.
- Proporciona estatus de la calidad del proyecto en cualquier fase
- Es apoyada por los expertos de cada disciplina para la generación de estándares, políticas y lineamientos.

#### Ejemplo:

- Revisar la generación de la documentación apropiada
- Implementar y monitorear políticas de estándares de diseño y codificación
- Reportar la calidad integral del proyecto

### Quality Control (Testing)

- Orientada a detectar los errores en los requerimientos y en el software
- Asegura que se construya el software correctamente
- Asegura que es el software que el usuario final esta esperando.

#### Ejemplo:

- Detectar huecos funcionales en requerimientos
  - Ejecutar pruebas funcionales
  - Ejecutar pruebas de rendimiento del software
  - Ejecutar pruebas automatizadas.
- 
- Nota: La disciplina de pruebas es una disciplina de Quality Control.



# CONCEPTOS GENERALES DE TESTING

## Actividad 1

En una dinámica grupal, identificar que actividades mostradas en este slide, que es QA y que es QC (Testing)

Se hacen pruebas al celular construido, se mete al agua, se carga por 48 horas, se golpea contra la pared  
QA? QC?

Actividades que tienen como objetivo encontrar errores  
QA? QC?

Actividades que tienen como objetivo prevenir errores  
QA? QC?

Discutir entre los asistentes la diferencia entre cada una de estas actividades. Es muy común que a Testing se le llame "QA", esto es un termino equivocado.

Se asegura que en todo el proceso de construcción de una casa, se sigan los procedimientos, se generen las evidencias, se sigan los lineamientos  
QA? QC?

Se hacen pruebas al sistema construido, se revisa que funcione bien y se buscan errores.  
QA? QC?



## CONCEPTOS GENERALES DE TESTING

¿ Que son las pruebas de software?

Revisa la diferencia entre el comportamiento actual de un sistema versus el comportamiento esperado bajo condiciones controladas.

La revisión debe incluir comportamientos normales y anormales en el sistema (eventos no deseados). Ejemplo

Normal: Revisar que si se haga la factura

Anormal: Revisar que no se deban imprimir 200,000,000,000 de registros

Otros aspectos a considerar acerca de las pruebas:

Asegura que el sistema haga lo que supuestamente tiene que hacer, y no haga lo que se supone no debe de hacer.

Asegurar que el producto cumple con los requerimientos especificados por el cliente.

Analizar la aplicación con la intención de encontrar problemas y defectos.



## CONCEPTOS GENERALES DE TESTING

### ¿Por que probar?

Somos humanos, todos cometemos errores, aun y sean gerentes, analistas, desarrolladores, testers, usuarios finales.

Los proyectos de desarrollo de software no pueden ser desplegados idénticamente(variabilidad), cada uno tiene sus características y es único (en tiempo, esfuerzo, gente, etc), por lo tanto, siempre existe la posibilidad de inyectar errores.

### ¿Cuando probar?

Tan pronto que los requerimientos hayan sido definidos. Podemos comenzar a realizar inspecciones formales, a los requerimientos, casos de uso, historias de usuarios (cualquier documentación que describa el comportamiento del sistema que será construido)

Cada vez que haya un cambio en la codificación, configuración, infraestructura, arquitectura o algún componente, realizando un análisis para determinar que tipo de prueba deberá ser aplicado y ejecutado. "Cada vez que se toca la aplicación, debemos de analizar que pruebas se deben de realizar"



## CONCEPTOS GENERALES DE TESTING

### Actividad 2

En una dinámica grupal, contestar las siguientes preguntas.

¿Por que es necesario probar un producto construido?

Si un sistema que previamente fue probado, tuvo un cambio  
¿Es necesario probar nuevamente? ¿Por qué?

¿Desde que momento podemos empezar a probar?

Después de haber contestado las preguntas, solicitar a los asistentes responderlas con sus propias palabras.



## CONCEPTOS GENERALES DE TESTING

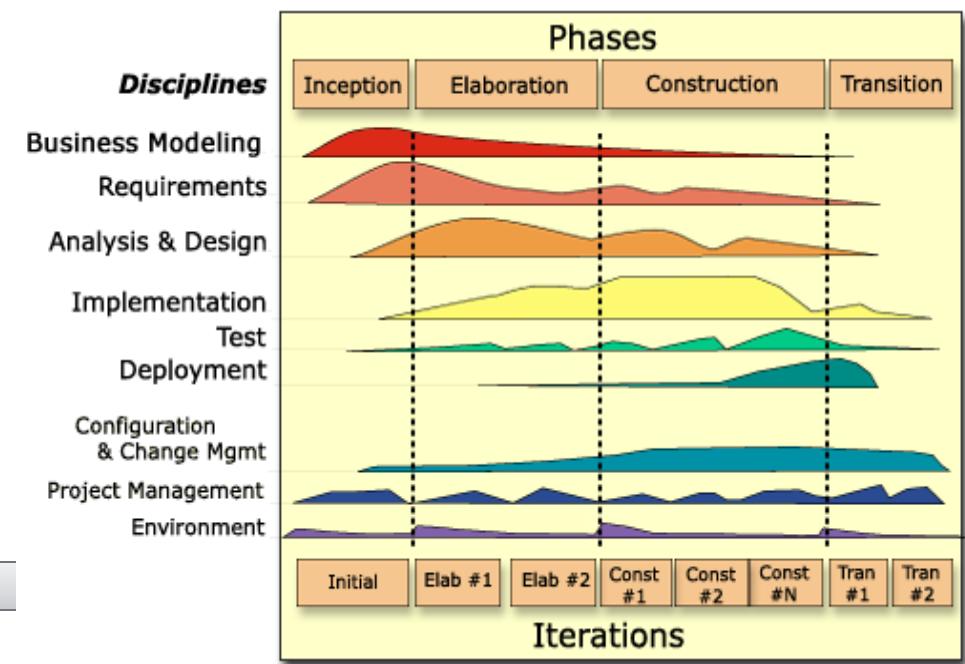
Las pruebas en una aplicación es un esfuerzo continuo

Las actividades de pruebas en un proyecto, deben de empezar desde el inicio del proyecto, y terminan hasta que el usuario final "firma" la aceptación del sistema.

El esfuerzo de pruebas inicia desde el inicio del proyecto porque:

Se necesita planear el esfuerzo

Se necesita generar la estrategia de pruebas desde el inicio,  
¿Que probar? Como probar? ¿Cuando probar? ¿Quienes  
van a probar? Etc.



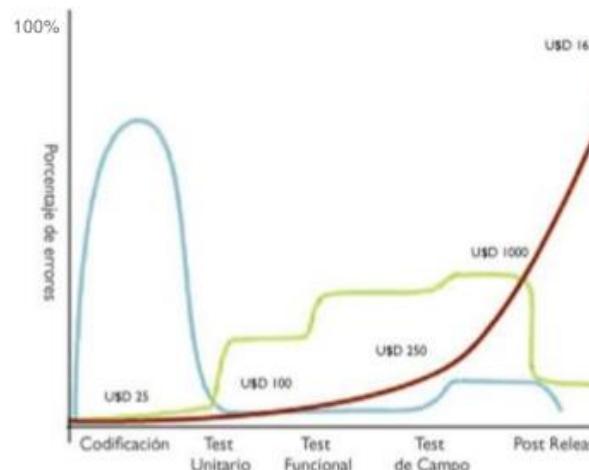


## CONCEPTOS GENERALES DE TESTING

### Relación Costo corrección defectos -vs-Fase de desarrollo

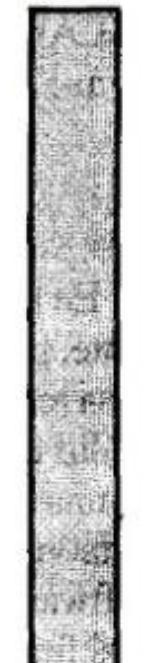
1000 r

#### Costo de corregir defectos



40-1000 veces

veces



En fase de explotación

Costo de reparar un defecto en relación a las diferentes etapas del ciclo de desarrollo.

Hay diferentes fuentes en la industria que mencionan esta relación.

Las fuentes tienen algunas diferencias en el costo de corregir defectos al final de la fase de desarrollo en comparación con fases tempranas

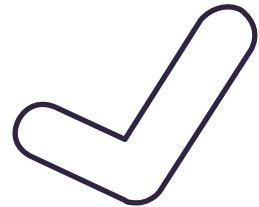
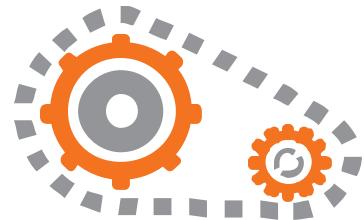
TODAS las fuentes coinciden que corregir en las fases finales, siempre será mucho mas caro que corregir en fases tempranas.



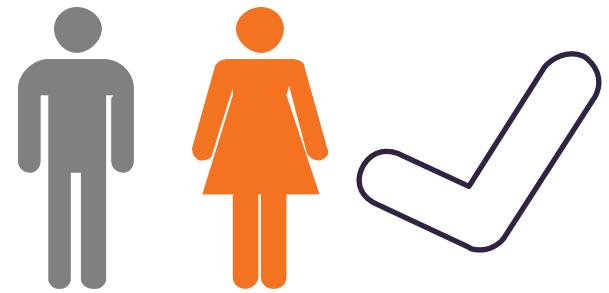


## CONCEPTOS GENERALES DE TESTING

¿Cuál es el objetivo de la disciplinas de pruebas?



Asegurar que el sistema  
funciona correctamente.



Asegurar que es el sistema que  
el usuario final está esperando.



## CONCEPTOS GENERALES DE TESTING

### Actividad 3

En una dinámica grupal, contestar las siguientes preguntas.

Preguntarle a los asistentes, ¿cuales son los 2 objetivos principales de realizar testing?

Pensar en la construcción de una casa de 3 pisos, el cliente comenta que el la pidió de 2 pisos! Que fue un error!

Si el proyecto aun esta en planos: Que hay que hacer y cuanto esfuerzo y costo se necesita para corregir el problema

Si el proyecto esta en obra negra: Que hay que hacer y cuanto esfuerzo y costo se necesita para corregir el problema

Si el proyecto ya se terminó, la casa esta amueblada: Que hay que hacer y cuanto esfuerzo y costo se necesita para corregir el problema.

¿Con el Software pasa lo mismo? ¡¡¡¡SI!!!!



## CONCEPTOS GENERALES DE TESTING

¿Como me aseguro que el sistema funciona correctamente?

- Detectando huecos funcionales
- Realizando pruebas negativas
- Realizando pruebas de varios niveles
- Realizando pruebas de Interfaz Gráfica
- Realizando Pruebas de límites de datos
- Realizando pruebas de concurrencia funcional
- Realizando pruebas no funcionales

**¡Tratar de romper el sistema!**

¿Como me aseguro que es el sistema que el usuario final esta esperando?

- Detectando diferencias entre requerimientos y alcance de pruebas
- Validando matrices de pruebas con usuario final
- Definiendo alcance de pruebas de Aceptación en etapas tempranas
- Coordinando pruebas de aceptación basado en alcance aprobado por el cliente final.

**¡Involucrar al cliente en validaciones desde etapas tempranas!**



## CONCEPTOS GENERALES DE TESTING

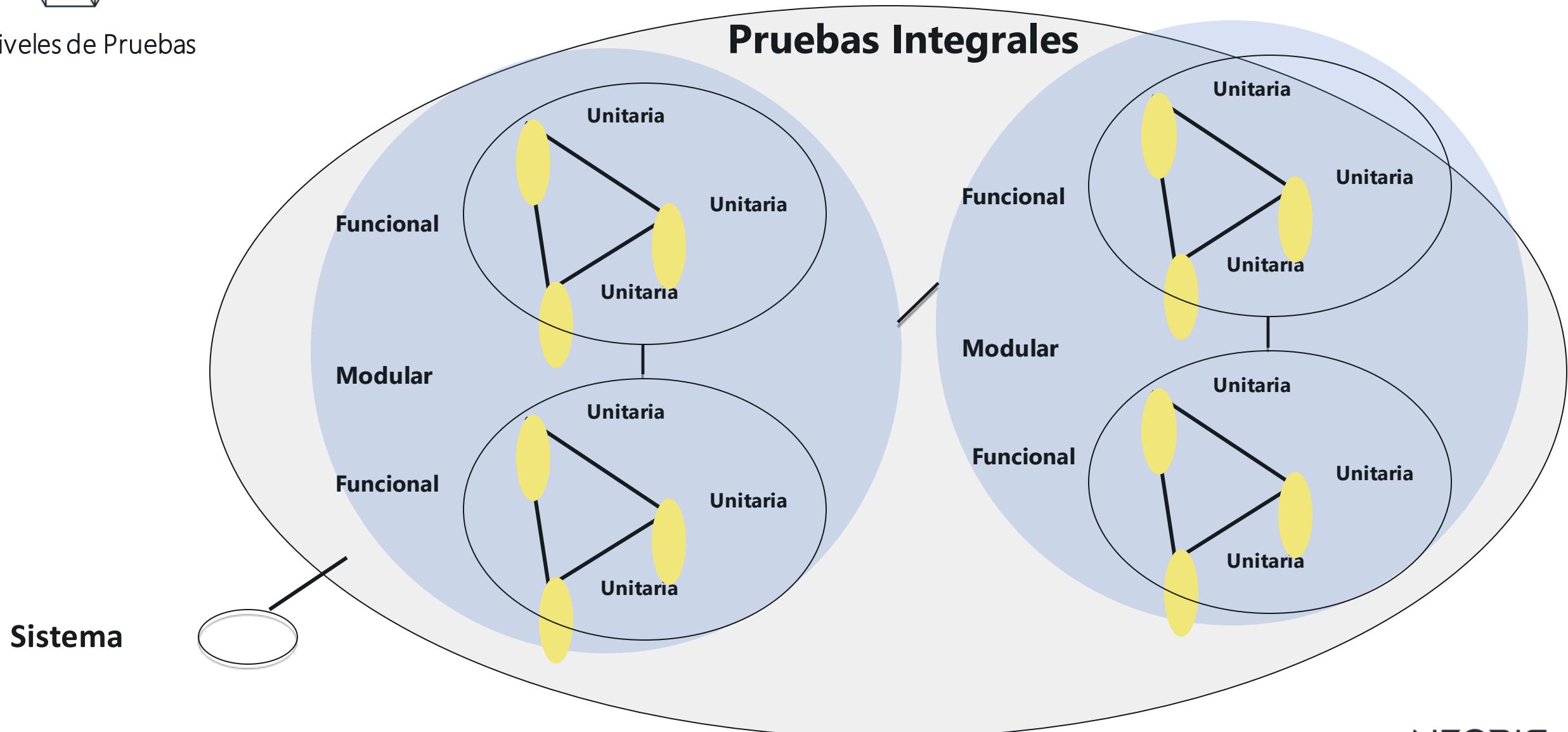
Niveles de Pruebas





## CONCEPTOS GENERALES DE TESTING

Niveles de Pruebas





# CONCEPTOS GENERALES DE TESTING

## Niveles de Pruebas

### Unitarias

Funcionalidades probadas de manera aislada sin interactuar con otras. Los tipos de prueba incluidas son límites, error handling, GUI.

Ejemplo:

#### Casos de Prueba:

Revisar máximos +1 de todos los campos de texto

Revisar la usabilidad de la pantalla

Revisar ortografía de la pantalla

Escribir el nombre del cliente  
".ñ{j?=)(/&%\$#! Insert into .[]<>"

Probar el "loop" del método  
"GetCliente" con variables aleatorias (Prueba unitaria caja blanca).

### Funcionales

En este nivel se validan las funcionalidades básicas agrupando funcionalidades directamente relacionadas verificando las reglas de negocio.

Ejemplo:

#### Casos de Prueba :

Crear factura

Probar reporte de factura

Realizar Devolución de factura

### Modulares

En este nivel de pruebas se busca la validación de las reglas de negocio que aplican a cierto módulo, esto es, si el módulo contiene ciertas funcionalidades, el objetivo consiste en validar las funcionalidades relacionadas.

Ejemplo:

#### Caso de Prueba 1:

Crear factura

Consultar Factura Creada

Devolver factura creada

Revisar Saldo de Facturas del cliente



## CONCEPTOS GENERALES DE TESTING

### Niveles de Pruebas

#### Integrales

Validación de las reglas de negocio que aplican a funcionalidades de diferentes módulos, esto es, si el módulo A contiene ciertas funcionalidades relacionadas con funcionalidades del módulo B, se validan las funcionalidades relacionadas.

#### Ejemplo:

Caso de Prueba 1:

Crear un pedidos (Módulo Abasto)

Pagar factura del pedido (Modulo Facturación)

Revisar póliza creada por la factura (Módulo Finanzas)

#### Sistema

En este nivel de pruebas se busca la validación de las reglas de negocio que aplican a funcionalidades de diferentes plataformas, esto es, si la plataforma A contiene ciertas funcionalidades relacionadas con la plataforma B, el objetivo consiste en validar las funcionalidades relacionadas.

#### Ejemplo:

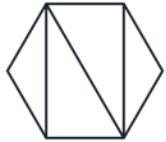
Caso de Prueba 1:

Dar de alta usuario en SAP

Dar de alta tarjeta en IB

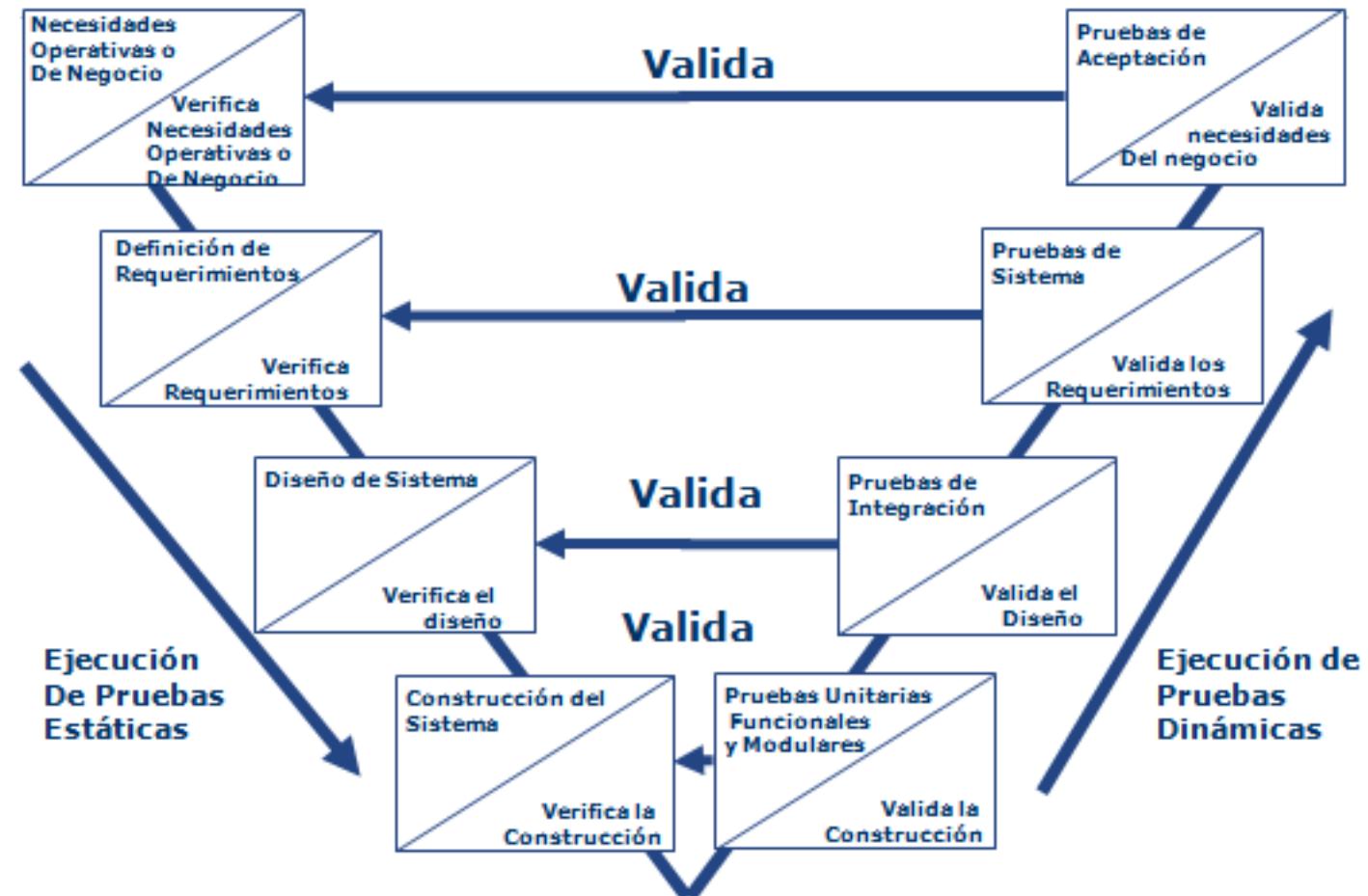
Dar crédito en IB (del usuario recién creado en SAP)

Revisar Pólizas en SAP (del usuario al que recién se le dio crédito).



## CONCEPTOS GENERALES DE TESTING

Concepto "V" de Pruebas ■





## CONCEPTOS GENERALES DE TESTING

### Entradas de Pruebas

Cada nivel de prueba puede tener una entrada diferente, a continuación algunos ejemplos.

| Entrada                  | Límites | Interfaz Gráfica | Funcional | Modular | Integral | Sistema | UAT |
|--------------------------|---------|------------------|-----------|---------|----------|---------|-----|
| Caso de uso              |         |                  | X         | X       | X        |         |     |
| Requerimientos           |         |                  | X         |         |          |         | X   |
| Especificación Funcional |         |                  | X         | X       |          |         |     |
| Especificación de Diseño |         |                  |           | X       | X        |         |     |
| Diseño Técnico           | X       |                  | X         |         |          |         |     |
| Diseño de Pantallas      |         | X                |           |         |          |         |     |
| Procesos de Negocio      |         |                  |           |         | X        | X       |     |
| Business Blue Prints     |         |                  |           |         | X        |         |     |



## CONCEPTOS GENERALES DE TESTING

### Actividad 4

En una dinámica grupal, contestar las siguientes preguntas.

¿Cómo se aseguran que el sistema esta construido correctamente?

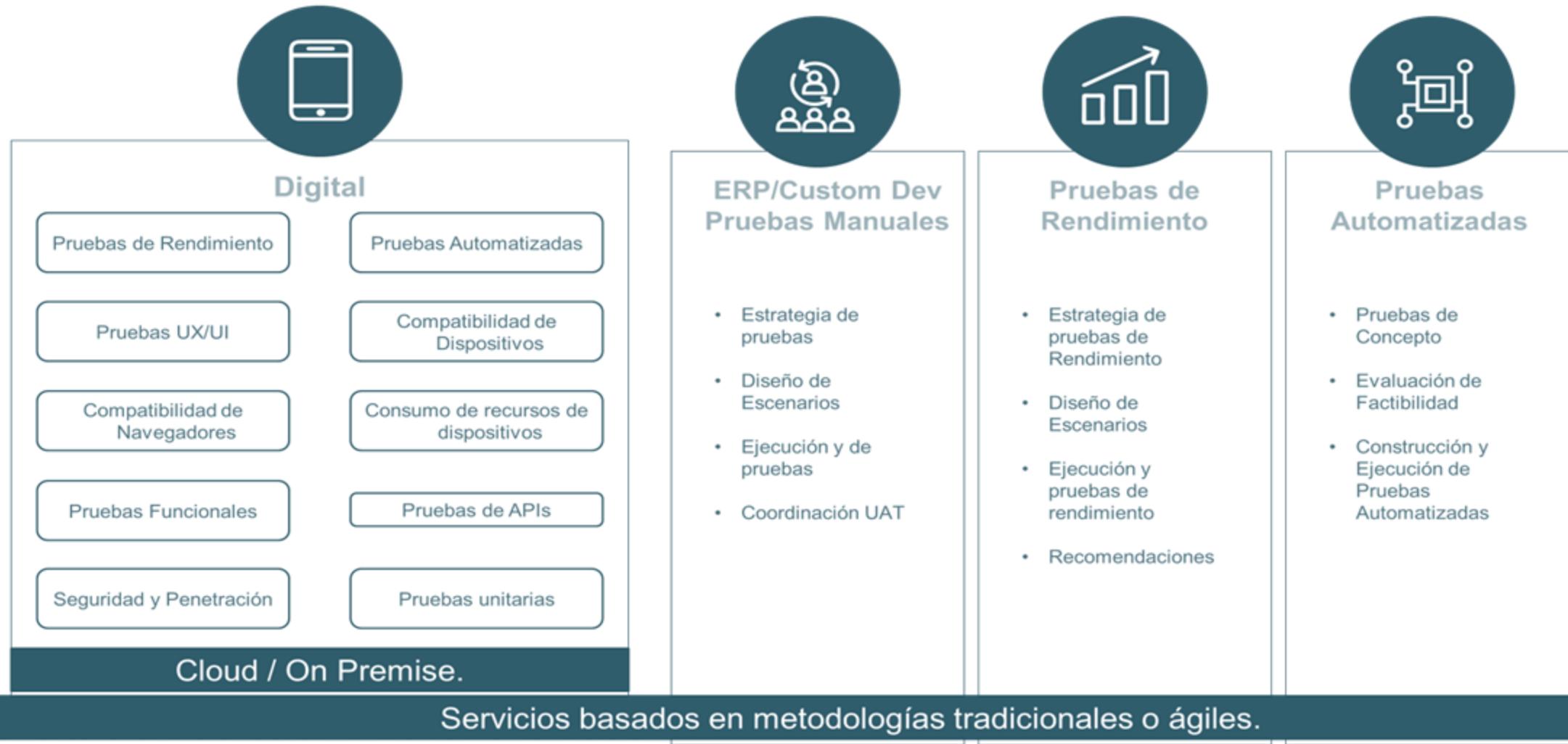
¿Cómo se aseguran que el sistema es el que el usuario final esta esperando?

Mencionar “niveles de pruebas”



# CONCEPTOS GENERALES DE TESTING

## TIPOS DE PRUEBA





# OBJETIVO DE LAS PRUEBAS UAT





## OBJETIVO DE LAS PRUEBAS UAT

Las pruebas de aceptación no son para identificar incidencias

El encontrar defectos (bugs) no es el enfoque principal de las pruebas de aceptación, de hecho, deben evitarse a toda costa, pues además de significar que se ha perdido buena parte del trabajo en la fase de análisis de requerimientos, afectan negativamente la confianza del usuario.

Como buena práctica, el Analista de pruebas (Tester) debe ejecutar pruebas de componente y pruebas de integración sobre áreas que se determine sean de alto riesgo, para favorecer una identificación temprana de los posibles errores. Esto en lugar de esperar que sean las pruebas de aceptación las que identifiquen defectos de alto impacto.



## OBJETIVO DE LAS PRUEBAS UAT

¿Que se valida con las pruebas de aceptación de software?

Los siguientes elementos pueden ser sujeto de pruebas de aceptación de software.

- Los procesos de negocio de sistemas que ya han sido integrados.
- Procesos operacionales y de mantenimiento.
- Procedimientos de usuario.
- Formularios.
- Reportes.
- Datos de configuración.



## OBJETIVO DE LAS PRUEBAS UAT

### ¿Cuando se realizan las pruebas de aceptación?

Las pruebas de aceptación pueden servir para evaluar el grado en que el sistema está listo para ser implementado y usado, aunque no necesariamente es el último nivel del Testing. Por ejemplo, después de las pruebas de aceptación podría estar establecida una prueba de gran escala de integración del sistema.

No necesariamente se limitan a la fase final de integración del sistema, sino que pueden realizarse en distintas etapas del ciclo de vida, por ejemplo:

- Si uno de los componentes del nuevo sistema, es un paquete de software comercial, las pruebas de aceptación de este pueden realizarse tan pronto sea instalado e integrado.
- La prueba de aceptación de usabilidad (un tipo de prueba no funcional de software) de un determinado componente se pueden realizar durante las pruebas de dicho componente.
- La prueba de aceptación de una nueva mejora de funcionalidad se puede realizar antes de la prueba de sistema.



## OBJETIVO DE LAS PRUEBAS UAT

### Casos de prueba que debe incluir la aceptación de usuario

- Con frecuencia, el diseño de casos de pruebas de aceptación de usuario es un subconjunto de los casos de prueba usados en las pruebas funcionales de componente y pruebas de integración.
- El subconjunto se define buscando incluir los principales escenarios de ejecución de la aplicación.
- Dado que el objetivo final de la prueba de aceptación es ganar la confianza del usuario, es recomendable incluir los casos de prueba que él defina dentro de las pruebas de aceptación. Para lo cual se le puede enviar una copia del diseño de casos de prueba y pedir su opinión.
- Si bien puede ocurrir que en la sesión de aceptación de software, el usuario pueda definir casos de prueba adicionales, debe hacerse todo lo posible que estos casos sean identificados de manera temprana, preferiblemente mucho antes de llegar a la sesión de aceptación.



## OBJETIVO DE LAS PRUEBAS UAT

En conclusión, pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema.

Las pruebas de aceptación a menudo también se les denominan pruebas de aceptación de usuario (UAT), pruebas de usuario final, pruebas de aceptación operacional o pruebas de campo.

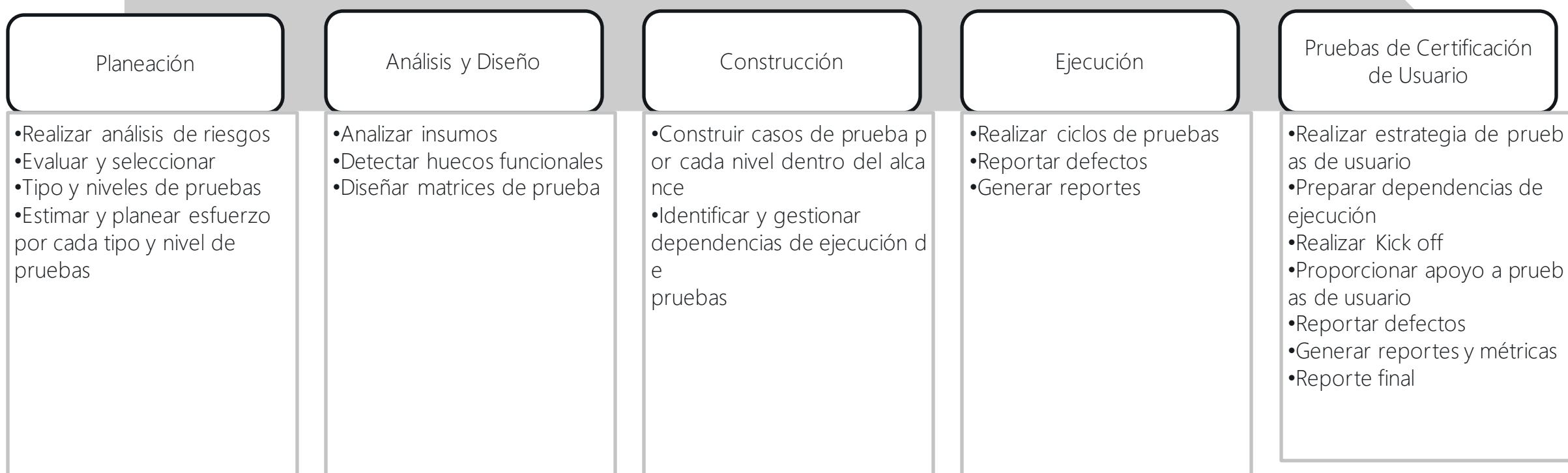


# METODOLOGIA GENERICA DE PRUEBAS– METODOLOGIA DE DESARROLLO CASCADA



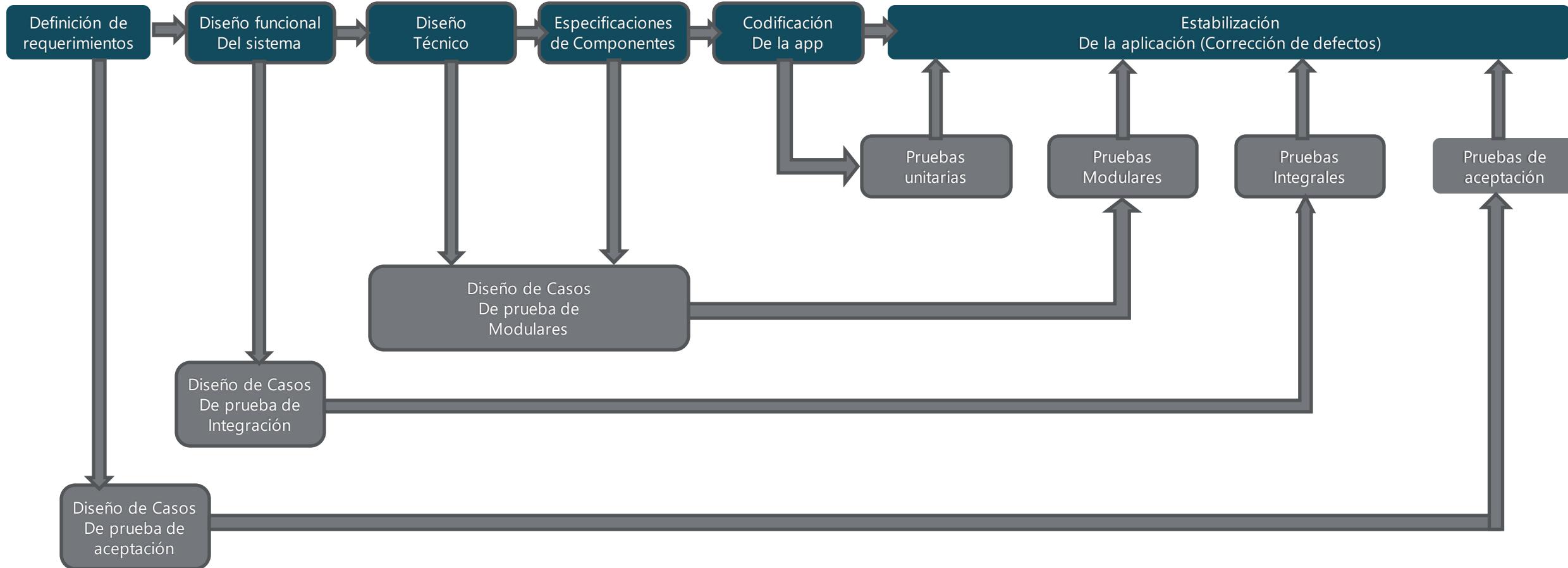


## *METODOLOGÍA GENÉRICA DE PRUEBAS*





## Modelo de Pruebas en Modelo de Cascada





## MODELO LINEAL SECUENCIAL O CASCADA

Característica de las pruebas en modelo de cascada:

- El alcance de las pruebas es típicamente al inverso del nivel de detalle:
  - Pruebas de aceptación (Alcance amplio)– validan requerimientos de negocio (Detalle bajo).
  - Pruebas modulares (Alcance corto) – Validan componentes (Detalle alto).
  - Después de la liberación del código, este entra en un proceso de estabilización que se basa en múltiples ciclos de pruebas típicamente con criterios de salida bien definidos.
  - Puede haber tiempos muertos en los equipos de desarrollo o en los equipos de prueba cuando el otro equipo esta trabajando.
  - Los defectos suelen tener impactos múltiples dependiendo de la fase en la que fueron inyectados.
  - Las pruebas de aceptación las suelen ejecutar los usuarios del negocio, pero pueden ser diseñadas por el equipo de prueba.



# Ejercicios



## EJERCICIOS SOBRE MODELO CASCADA



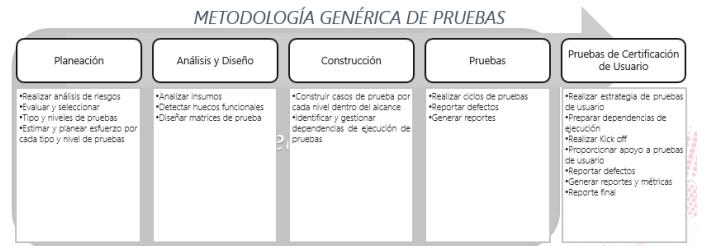
# FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA





# FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA

La etapa de Pruebas de Certificación de Usuario puede realizarse y recibir sus insumos de una metodología de cascada



Metodología de Cascada



## Pruebas de Certificación de Usuario

- Realizar estrategia de pruebas de usuario
- Preparar dependencias de ejecución
- Realizar Kick off
- Proporcionar apoyo a pruebas de usuario
- Reportar defectos
- Generar reportes y métricas

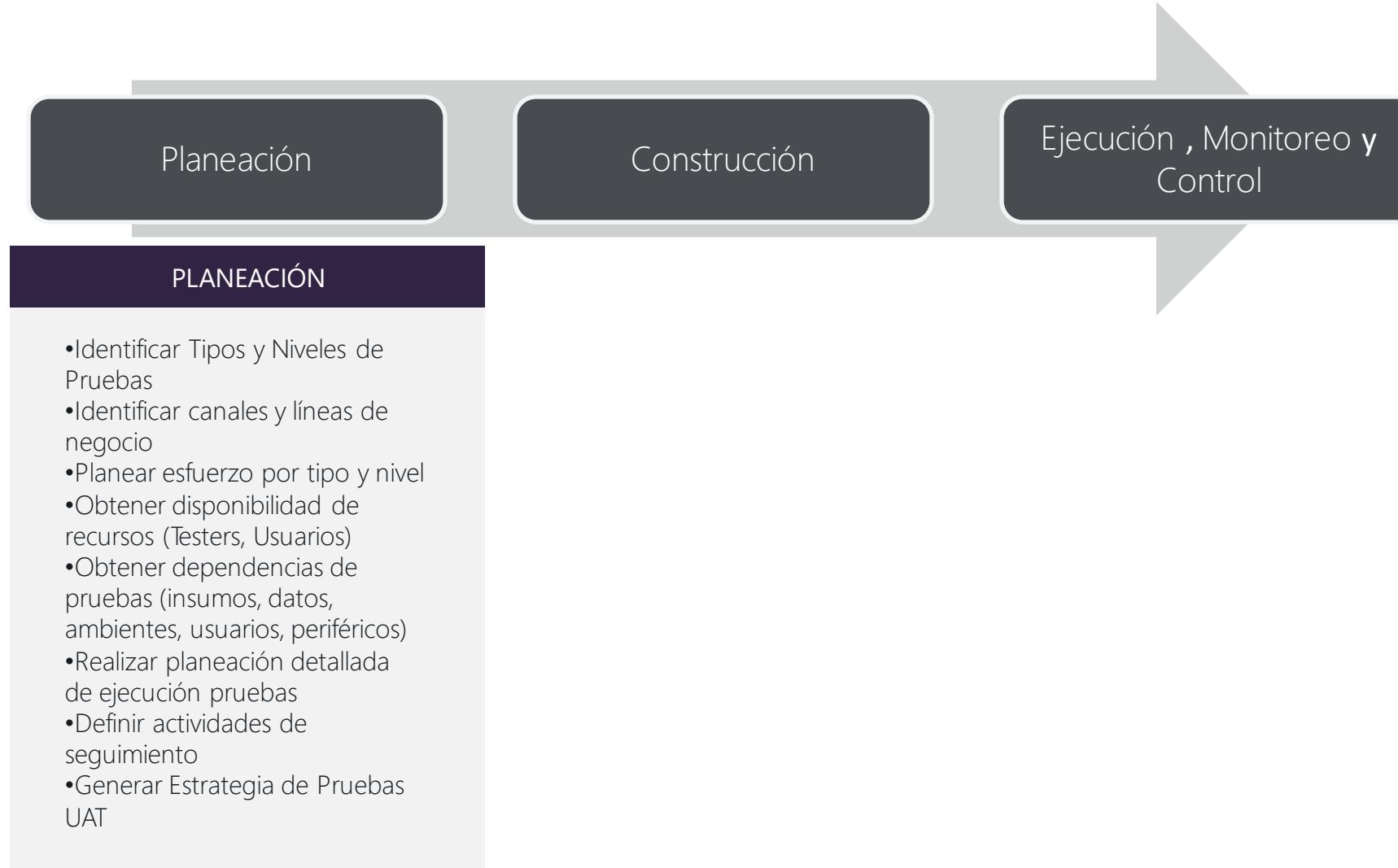


## FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA





# FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA - ACTIVIDADES





# FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA

Planeación

Construcción

Ejecución , Monitoreo y Control

## CONSTRUCCION

- Generar Documento de Kick off
- Configurar herramienta de pruebas (fase, ciclos, flujo de defectos, reportes)
- Gestionar insumos de pruebas
- Casos de prueba seleccionados



# FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA

Planeación

Construcción

Ejecución , Monitoreo y Control

## EJECUCION, MONITOREO Y CONTROL

- Realizar kick off de inicio de pruebas.
- Proporcionar metas diarias de ejecución de pruebas
- Dar soporte en la ejecución de pruebas (Dudas, usuarios, periféricos, datos, etc.) .
- Realizar reuniones de seguimiento (defectos, ejecución diaria, riesgos & problemas, otro).
- Actualización de log de ejecución, incidencias y mejoras
- Reportes de pruebas (métricas y SLAs).



## FASE DE PRUEBAS DE ACEPTACION DE USUARIO – METODOLOGÍA DE DESARROLLO CASCADA

Application Lifecycle Management

QAComplete Visual Studio

MICRO FOCUS | silk™ Software Test Management, Automation and Performance Testing

Rational. Functional Tester

Vendor

squash™

mantis BUG TRACKER

Test Link

Bugzilla

Open Source

Herramienta de Gestión de pruebas



UAT Cascada: ESTIMAR, PLANEAR, INICIAR, REALIZAR,  
CONTROLAR, MONITOREAR, REPORTAR, MEDIR Y CERRAR.





## ESTIMACIÓN DE PRUEBAS

Hay 2 métodos populares para estimar

### Detallado

- Consiste en analizar cada requerimiento para entender:  
Complejidad  
Longitud  
Riesgos  
Definir supuestos
- Ventajas  
Brinda certeza de la estimación, se reducen riesgos, se encuentran huecos tempranamente
- Desventajas  
Se requiere experiencia y entrenamiento, toma mucho tiempo

### Aproximado

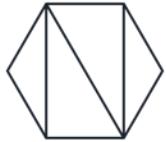
- Consiste en usar un modelo de distribución normal para hacer un estimado
- Ventajas  
Es rápido para tener una estimación aproximada, muy útil para estimar grandes volúmenes de requerimientos
- Desventajas  
Se pueden omitir casos excepcionales que pueden impactar la estimación, no es efectivo para un número reducido de requerimientos



## ESTIMACIÓN DE PRUEBAS

### Método detallado

- Entrada
  - Requerimientos
- Salidas
  - Estimación
- Procedimiento
  - Estudiar requerimiento por requerimiento
  - Determinar: Complejidad (Alto, Medio, Bajo) ,estimación en horas.



## ESTIMACIÓN DE PRUEBAS

### Método Aproximado

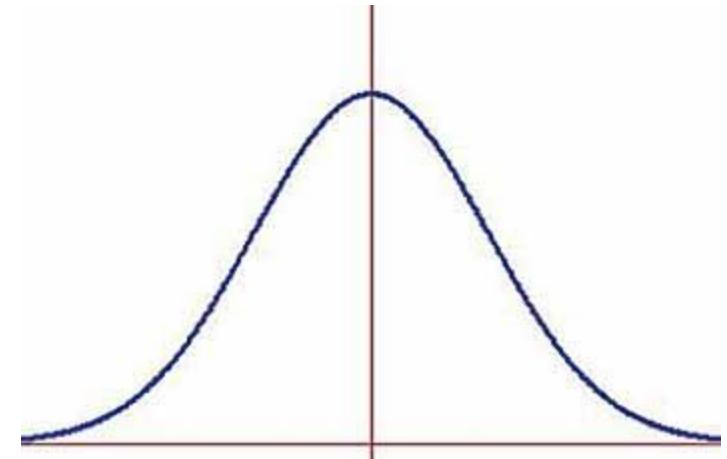
- Entrada
  - Requerimientos
- Salidas
  - Estimación
- Procedimiento
  - Determinar el % de Alto, Medio, Bajo
  - Determinar las horas por complejidad

Ejemplo: Existen 300 requerimientos, y sabemos que:

Un requerimiento de complejidad alta toma 12 hrs de diseño

Un requerimiento de complejidad Media toma 8 hrs de diseño

Un requerimiento de complejidad Baja toma 4 hrs de diseño



| Complejidad | Distribución | Requerimientos | Esfuerzo | Totales   |
|-------------|--------------|----------------|----------|-----------|
| Alto        | 15%          | 45             | 12 hrs   | 540 hrs   |
| Medio       | 70%          | 210            | 8 hrs    | 1,680 hrs |
| Bajo        | 15%          | 45             | 4 hrs    | 540 hrs   |



## PLANEACIÓN DE PRUEBAS UAT

- 1.- Analizar los requerimientos de desarrollo de software.
- 2.- Identificar las funcionalidades nuevas a probar.
- 3.- Identificar las funcionalidades de sistemas existentes que deben probarse.
- 4.- Definir la estrategia de pruebas.
- 5.- Definir los criterios de inicio, aceptación y suspensión de pruebas.
- 6.- Identificar los entornos (ambientes) requeridos.
- 7.- Determinar necesidades de personal y entrenamiento.
- 8.- Establecer la metodología y procedimientos de prueba.
- 9.- Elaborar la planificación de las pruebas.
- 10.- Identificar los riesgos y definir planes de respuesta.



## INICIAR PRUEBAS DE ACEPTACIÓN DE USUARIO

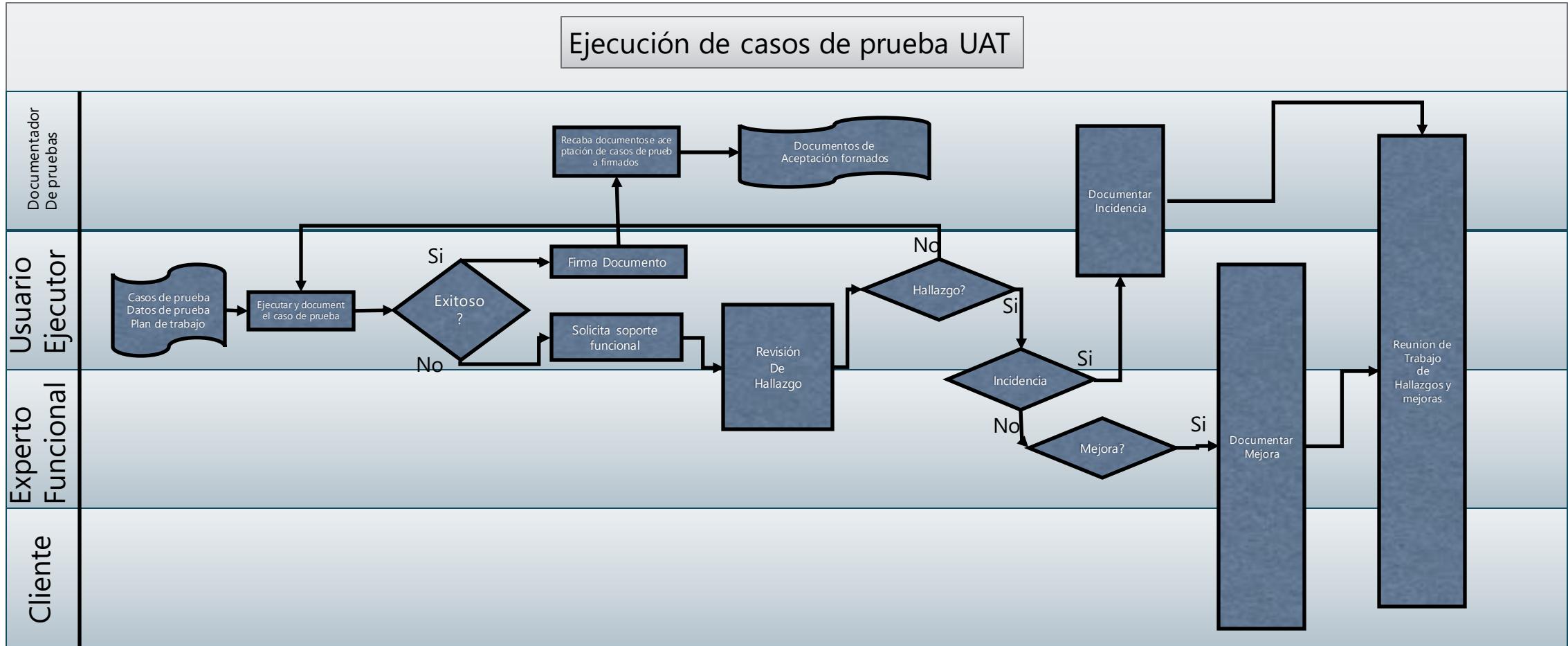
Se generará un documento para la sesión de Kick Off , con el objetivo de explicarle a los usuarios finales la manera en la que se estarán ejecutando y controlando la fase de pruebas de aceptación, este documento contara con los siguientes puntos de manera detallada. Se tomara como base el documento de Estrategia de Pruebas para la definición de éste documento.

| Tópico   | Descripción   |
|--|---|
| Objetivo   | Describir el objetivo de las pruebas de UAT al usuario  |
| Plan de Trabajo Detallado                        | Mostrar el plan de trabajo detallado, incluyendo el horario y meta de ejecución de casos de prueba por día  |
| Requerimientos Dentro de Alcance                 | Mostrar los escenarios dentro del alcance de pruebas, adjuntar el Excel con las matrices  |
| Requerimientos Fueras de Alcance                 | Mostrar los escenarios fuera del alcance de pruebas, adjuntar el Excel con las matrices   |
| Metodología                                      | Definir en este punto lo siguiente; Dinámica de las sesiones de prueba, como se registrarán las ejecuciones y los hallazgos, como se realizará el manejo de los hallazgos, describir los estatus y tipos de los hallazgos , describir los criterios de severidad, definir como se estará dando el soporte a las ejecuciones, etc. |
| Equipo de Pruebas (Cliente y NEORIS)             | Mostrar el nombre de los participantes y a que equipo y rol pertenecen  |
| Roles y Responsabilidades (Cliente y NEORIS)     | Detallar los roles y responsabilidades del equipo de pruebas (Cliente y Neoris)   |
| Equipo de Trabajo (Pares y/o puntos de contacto) | Mostrar los grupos de trabajo que se estarán organizando por área de negocio.   |
| Herramientas a Utilizar                          | Se definirán las herramientas a ser utilizadas  |
| Glosario   | Se explicarán algunos conceptos básicos relacionados con el esfuerzo de pruebas   |



# REALIZAR PRUEBAS DE ACEPTACIÓN DE USUARIO

Proceso de ejecución de casos de prueba, firmas de liberación y administración de hallazgos. En los siguientes slides se explica textualmente este proceso y la forma de llenar los documentos.





## REALIZAR PRUEBAS ACEPTACIÓN DE USUARIO

### Metodología de ejecución de casos de prueba - Detalle

- Se les entregara a los usuarios clave los casos de prueba impresos a ejecutar con los datos de prueba relacionados a los casos de prueba.
- Por cada caso de prueba exitoso el usuario entregará el documento firmado en papel al documentador de pruebas, éstas entregas se realizarán diariamente al finalizar la sesión de trabajo al documentador de pruebas.
- En caso de haber algún hallazgo (incidencia, duda, o funcionalidad no reconocida) , se deberá en este mismo instante levantar la mano y se solicitará la revisión del experto funcional para decidir si es hallazgo o una duda funcional.
- En caso de ser duda funcional se prosigue con la ejecución.
- En caso de ser hallazgo se revisa si es Incidencia, Mejora o cae en un "Área gris" ("área gris" se refiere cuando no se llega a un acuerdo inmediato del tipo de hallazgo).
- En caso de ser Incidencia, se pedirá apoyo al documentador de pruebas para que éste la documente en base a la información que se solicite y se actualiza el registro de incidencias.
- En caso de ser una Mejora, se pedirá poyo al experto funcional y al project manager para que se documente y entre por el proceso de control de cambios.
- En caso de caer en "Área gris" se documenta el hallazgo y se solicita una reunión formal en un horario distinto al planeado para las pruebas para revisar el detalle del mismo, dando como resultado cualquiera de las siguientes opciones:
  - Una re ejecución del caso de prueba debido a una falta de conocimiento funcional.
  - La actualización del log de Incidencias.
  - La actualización del log de Mejoras.



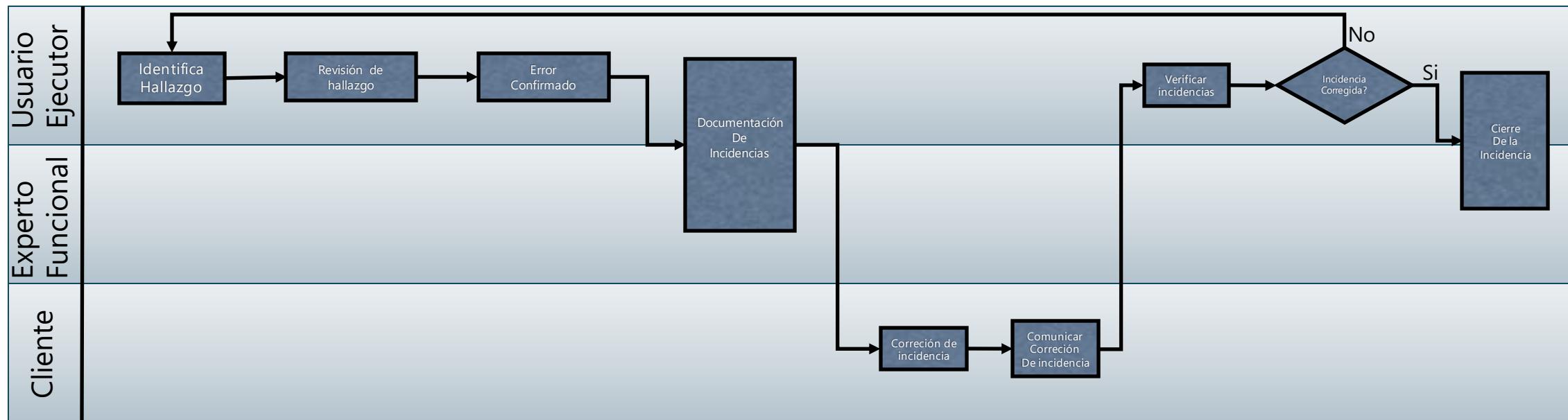
## MONITOREAR Y CONTROLAR PRUEBAS UAT

### Proceso de manejo de incidencias.

El usuario posterior a tener una confirmación con el experto funcional de que se encontró una incidencia, se procede a que el documentador de pruebas la registre con la información solicitada.

Posterior a la corrección de la incidencia, se notificara para que el usuario verifique la corrección de la misma y se pueda cerrar y firmar el caso de prueba

#### Alta y Revisión de incidencias UAT





## MONITOREAR Y CONTROLAR PRUEBAS UAT

### Criterio de Salida

El Criterio de Salida se define como el criterio para marcar como "pasada" la fase de ejecución de pruebas de Aceptación del usuario.

Se estará entregando el sistema sin errores de severidad "Critica" y severidad "Alta".



## MONITOREAR Y CONTROLAR PRUEBAS UAT

Reportar de manera diaria el estatus de las pruebas:

- Casos de prueba pasados.
- Casos de prueba fallados.
- Avance de pruebas.
- Desviación vs la planeación.
- Estatus de defectos.
- Riesgos y problemas.



## CONTROL DE PRUEBAS UAT

### Administración de Riesgos y Problemas

- En cuanto a los Riesgos y Problemas se seguirá el siguiente proceso:
- Cualquier recurso del equipo de pruebas de aceptación podrá identificar un Riesgo o Problemas.
- Al momento de Identificarlo este deberá notificarse al Coordinador de pruebas y al PM para este sea documentado y analizado
- El PM le dará seguimiento al riesgo o problema junto con el cliente
- El PM será el responsable de darle seguimiento hasta asegurarse que haya sido cerrado.
- El documento donde se llevará el registro de riesgos y problemas será mostrado en los reportes diarios de avance para que todo el equipo de trabajo este enterado de los mismos
- Al levantar un Problema o Riesgo este deberá mencionar un plan de mitigación o contingencia

| Rol  | Responsabilidad |
|--|-----------------|
| Dirección  | D               |
| Gerencia , PM, Coordinador   | I, R, D         |
| Usuario Clave, Experto Funcional, Documentador de Pruebas, Líder de Pruebas, Equipo de soporte | I, R            |



**I = Informa, D = Decide, R = Recomienda**



## UAT - Actividad

Formar 2 equipos A y B

Cada equipo define los UAT escenarios

Equipo A: Windows 10

Equipo B: Android

Cada equipo debe exponer sus matrices de escenarios

Nota: No detallar escenarios, solo identificarlos

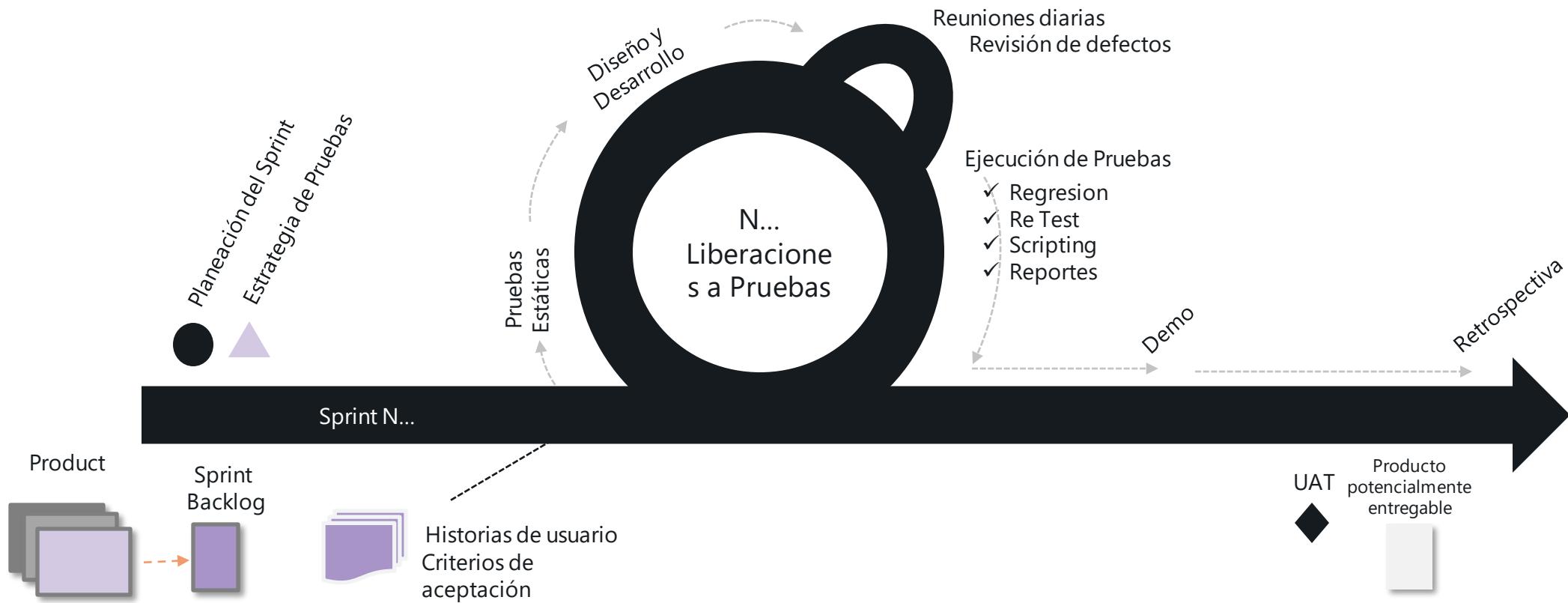
Hacer una estimación de ejecución tomando en cuenta el tamaño de cada equipo



# METODOLOGIA GENERICA DE PRUEBAS – MARCO DE TRABAJO AGIL



# METODOLOGIA GENERICA DE PRUEBAS – MARCO DE TRABAJO AGIL





# METODOLOGIA GENERICA DE PRUEBAS – MARCO DE TRABAJO AGIL

La etapa de Pruebas de Certificación de Usuario puede realizarse y recibir sus insumos de una metodología de cascada o ágil.



Framework Ágil

## Pruebas de Certificación de Usuario

- Realizar estrategia de pruebas de usuario
- Preparar dependencias de ejecución
- Realizar Kick off
- Proporcionar apoyo a pruebas de usuario
- Reportar defectos
- Generar reportes y métricas



## FASE DE DEMOSTRACIÓN (DEMO) – MARCO DE TRABAJO AGIL





## FASE DE DEMOSTRACIÓN (DEMO) – MARCO DE TRABAJO AGIL

Tiene lugar al finalizar el sprint, ya que al alcanzar el tiempo previsto para el mismo, es necesario revisar todo lo construido, ver si se ajusta a lo que necesitan los stakeholders y poder recibir feedback de los mismos.

A esta reunión acuden el Dueño de Producto como representante de los Interesados, además del Equipo de desarrollo y el Scrum Master.



## FASE DE DEMOSTRACIÓN (DEMO) – MARCO DE TRABAJO AGIL

Puntos que se deben considerar para una DEMO:

- Listar todas las historias completadas en su agenda
- Elimine cualquier historia que no deba ser demostrada
- Organice las historias restantes de manera aproximada en escenarios o temas
- Decida si desea que los desarrolladores ayuden a dar partes de la demostración.
- Siempre establezca expectativas y de contexto a lo largo de la demo



## DEMO – ACTIVIDAD 1

- Crear uno o más equipos de trabajo (dependiendo de la asistencia)
- Utilizando aplicaciones comerciales (ej. Tripadvisor, Booking, Despegar, etc), pedirles que准备 una DEMO de un escenario básico de búsqueda y selección
- Requisitos:
  - Laptop 1 por equipo
  - Acceso a internet
  - Proyector



# DEMO: ESTIMAR, PLANEAR, INICIAR, REALIZAR, CONTROLAR, MONITOREAR, REPORTAR, MEDIR Y CERRAR DEMO





## DEMO - ESTIMAR

La DEMO debe ser ágil pero también se tiene que proporcionar el tiempo suficiente para cubrir las actividades.

Para poder ESTIMAR la duración de la DEMO, se deben considerar los siguientes puntos:

- Cantidad de historias a revisar
- Selección y número de asistentes
- Crear un plan de actividades para la reunión



## DEMO - PLANEAR

Para PLANEAR la DEMO, se deben considerar los siguientes puntos:

- Listar las actividades y tiempo que se dedicará a cada actividad
- Adaptar la demostración a las características de la audiencia
- Resolver las dudas que se plantean de forma clara, inmediata y sencilla
- Demostrar las funcionalidades del producto de una en una y forma secuencial
- Las preguntas de sondeo o feedback deben realizarse al final de la demostración para evitar interrupciones



## DEMO - INICIAR

Al INICIAR una DEMO, todos los asistentes deben estar presentes, de lo contrario, se podrán permitir un máximo de 5 minutos, pero siempre a consideración del moderador. Al comenzar con la reunión, tomar en cuenta los siguientes puntos:

- Comenzar a tiempo
- Presentar la agenda a la audiencia
- Explicar el propósito de la DEMO
- Dedicar los primeros minutos a aclarar los puntos fundamentales



## DEMO - REALIZAR

- Después de iniciar la DEMO, permitir que el moderador conduzca la reunión y proceda a realizar las actividades del plan.
- Es importante designar a un moderador o administrador de la DEMO, para controlar la ejecución de las actividades, los tiempos y participación de los asistentes.
- Repasar las historias una a una, mostrando la funcionalidad relacionada en los flujos básicos, también se pueden ejecutar flujos alternos si aplica.
- Los comentarios de retroalimentación serán al final de la demostración.



## DEMO - CONTROLAR

El control incluye la determinación de acciones preventivas o correctivas, durante la presentación, afín de seguir el plan establecido para la DEMO.

En todo momento debemos CONTROLAR el desarrollo de la presentación, el Usuario puede proponer un escenario o combinación de datos en específico, además de aclarar dudas puntuales, para poder hacerlo de una manera correcta.

- Controlar la presentación de los flujos funcionales (debe ser de acuerdo a la lista de historias)
- Controlar el tiempo de las actividades, de tal manera que se pueda cubrir el alcance establecido
- Apegarse al plan de actividades



## DEMO - MONITOREAR

El monitoreo es un aspecto de la administración de la DEMO, que se realiza a lo largo de toda la demostración.

Consiste en recopilar, medir y distribuir la información relativa al desempeño, y en evaluar las mediciones y las tendencias que van a permitir efectuar mejoras al proceso.

- Debe monitorear la diferencia entre lo planificado y lo real
- Identificar cuándo y dónde existieron desviaciones al plan
- Comprobar si la gestión del alcance y la gestión del tiempo para las actividades, fue el correcto



## DEMO - REPORTAR

Como resultado del monitoreo y control de las actividades de la DEMO, se realizará un REPORTE, donde quedará plasmada la situación de la demostración, con el objetivo de que los responsables puedan evaluar su desarrollo.

El reporte debe incluir al menos los siguientes puntos:

- Retroalimentación del cliente: Ya sea porque hay errores o mejoras, que el cliente considera importantes para el negocio.
- Observaciones realizadas como parte del control y monitoreo de la demo.
- Reportar en la herramienta de gestión los defectos identificados, estos defectos deben de leerse de manera textual a la audiencia, de manera de verificar la correcta redacción y se entienda el defecto/mejora por todos los asistentes, en especial el PO y equipo de desarrollo.



## DEMO - MEDIR

El objetivo de la medición es poder cuantificar el estado de cada DEMO, por lo que debemos definir el criterio o las unidades que se usará para medir este estado.

Se deben usar unidades relacionadas con el resultado final de la DEMO, ejemplo: para medir la operación de pintar un edificio, es más útil medir los metros cuadrados pintados que la cantidad de litros de pintura consumida.

- **Deben medirse resultados, no costes o consumos**
- **El plan realizado a partir del alcance y las tareas definidas, así como el esfuerzo estimado, será una línea base para la medición**



## DEMO - CERRAR

El CIERRE es el último rubro de una DEMO, es en esta fase en donde el cliente confirma si el producto cumple o no con sus expectativas, también da el visto bueno (Vo.Bo.) o "Sign off" para implementar el producto a producción.

- Se designan planes de acción para los defectos, mejoras y observaciones reportadas por el usuario.
- Todos los defectos y mejoras deben documentarse con su propio folio para su posterior seguimiento.
- Además se agregan las lecciones aprendidas de la demostración, para mejorar el proceso DEMO en el siguiente sprint.



## DEMO – ACTIVIDAD 2

Utilizar los mismos equipos de la ACTIVIDAD 1 y proporcionar a cada equipo documentación de una aplicación comercial

La documentación debe contener la información necesaria, para preparar y presentar una DEMO con todos sus rubros:

- ESTIMAR
- PLANEAR
- INICIAR
- REALIZAR
- CONTROLAR
- MONITOREAR
- REPORTAR
- MEDIR
- CERRAR



# EVALUACIÓN

