

# VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

---

# Bibliografía

---

Rakitin, Steven R. Software Verification and Validation for Practitioners and Managers. Artech House. 2001. Second Edition.

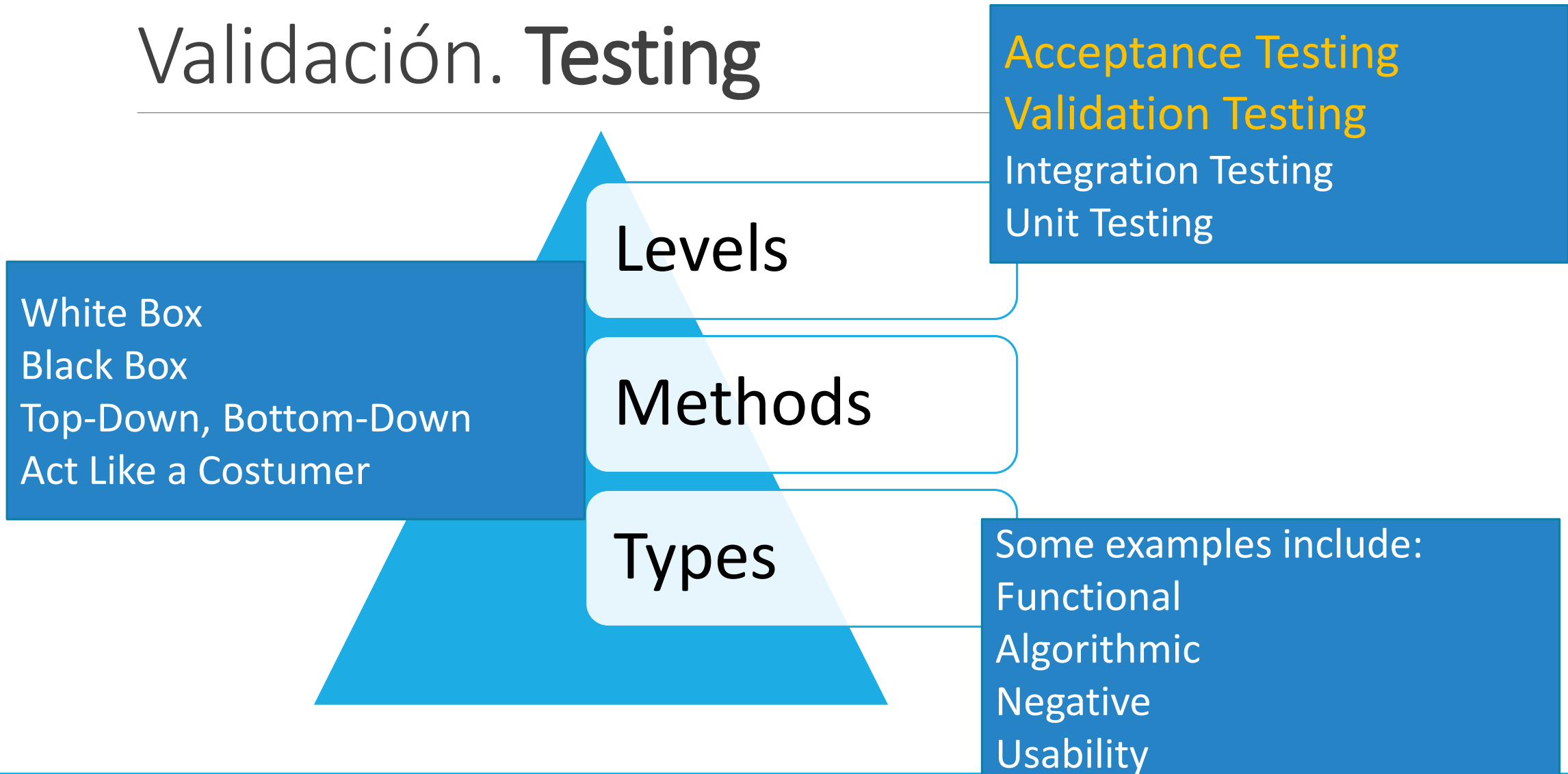
# Validación

# Validación

---

- Clase anterior
  - Conceptos Verificación Validación
  - Proceso de desarrollo y proceso de validación de forma concurrente
  - Validación informal
  - Validación formal

# Validación. Testing



# Validación

---

- Las actividades de validación están definidas alrededor de tres procesos básicos:

- ✓ Testing

- ✓ Medición

- ✓ Confiabilidad del software

# Validación

---

- Los objetivos del nivel denominado “validation or system testing” son determinar si el software cumple los requerimientos definidos.
- Algunas organizaciones involucran activamente a los clientes en el proceso de testeo proporcionando software de pre lanzamiento para la evaluación del cliente, llamado comúnmente **Testeo Alpha y Beta**.

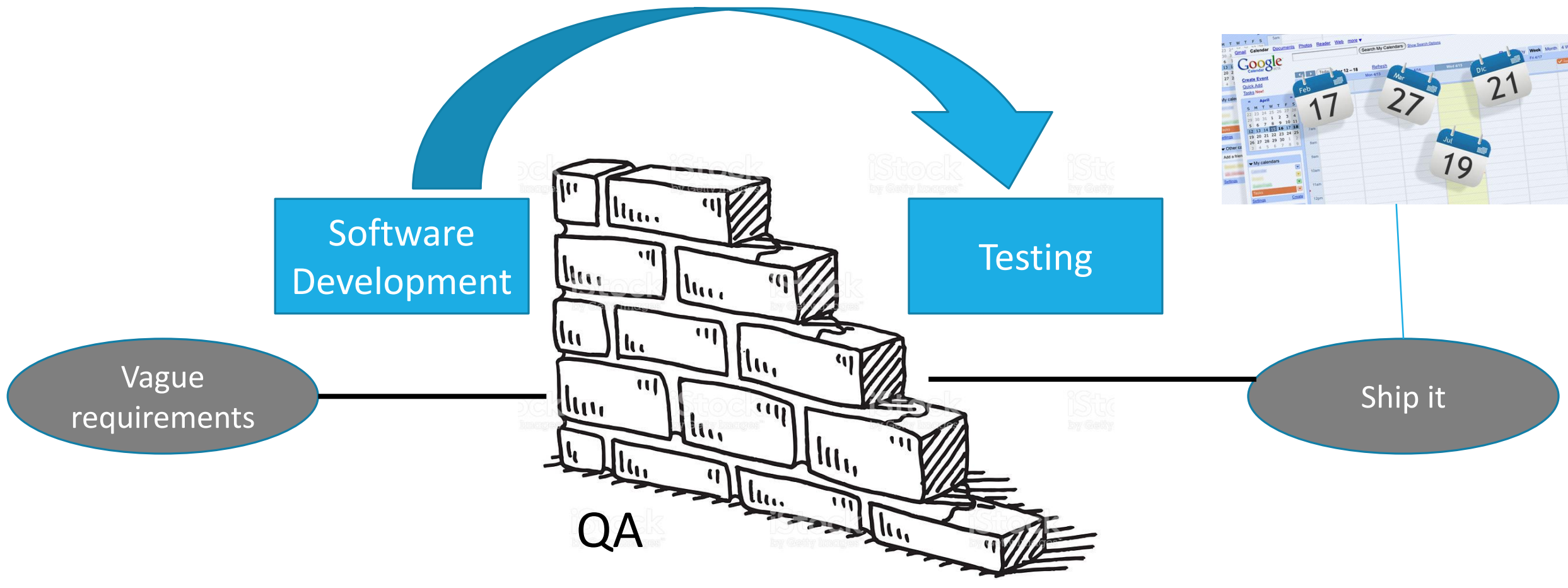
# Validación

---

- El siguiente nivel llamado “Acceptance Testing” es similar al anterior, pero el cliente se encuentra presente o es involucrado activamente.

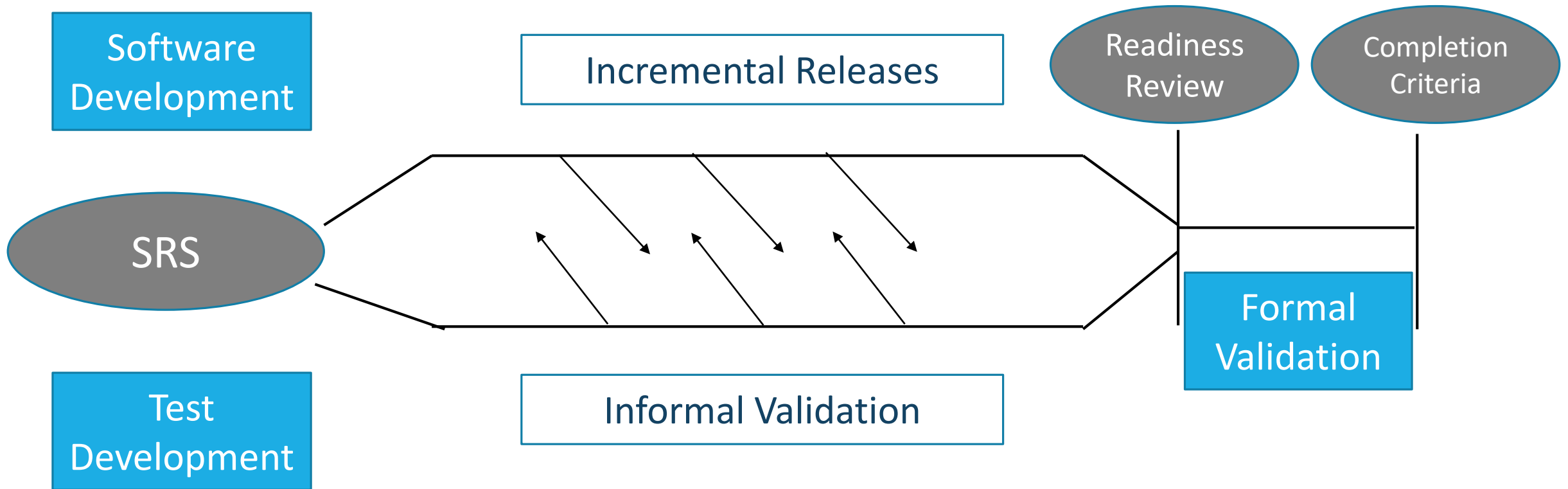


# Validación



Typical validation testing process

# Validación



**Concurrent development/validation testing model**

# Validación

---

- Agenda

- Planeación del proceso de Testing

- Plan del Test

- Estimación del Test
    - Desarrollo del Test y Validación informal
    - Revisión para la preparación de la validación y Validación Formal
    - Criterio para finalizar el proceso

- Procedimiento del Test

- Reporte del Test

# Validación

---

- La planeación es el proceso a través del cual se define la forma en la que los recursos del testing serán empleados, con el fin de emplearlos de manera efectiva.
- Documentos para la planeación del testeo. Rakitin (2001):
  - ✓ Plan del Test.
  - ✓ Procedimiento del Test.
  - ✓ Reporte del Test.

# Validación

---

- ✓ Plan del Test.
- ✓ Procedimiento del Test.
- ✓ Reporte del Test.

# Validación

---

- Plan del Test.
  - Define el alcance del trabajo que se realizará.
  - Qué debe hacerse y quién debe hacerlo.
  - Contiene información que permite a los administradores del proyecto calendarizar y ejecutar la actividad del testing.

# Validación. Plan del test

---

- Escrito normalmente por el equipo de Aseguramiento de la Calidad/Quality assurance (QA).
- Incluye todos los test a ejecutar por los ingenieros de software (desde los test de unidad, integración, etc.).

¿Cuántos test se necesitan?

¿Cuánto tiempo tomará desarrollar los test?

¿Cuánto tiempo tomará ejecutar estos test?

# Validación

---

- En la planeación del esfuerzo para realizar testing de validación, el plan debe incluir lo siguiente:
  - ✓ Estimación del Test.
  - ✓ Desarrollo del Test y validación informal.
  - ✓ Revisión para la preparación de la validación formal.
  - ✓ Criterio para finalizar el Test.



# Validación

---

- En la planeación del esfuerzo para realizar testing de validación, el plan debe incluir lo siguiente:
  - ✓ Estimación del Test.
  - ✓ Desarrollo del Test y validación informal.
  - ✓ Revisión para la preparación de la validación formal.
  - ✓ Criterio para finalizar el Test.

# Validación

---

## ■ Estimación del Test.

Consiste en los tres componentes siguientes:

- ❖ Estimación de la cantidad de test necesarios.
- ❖ Estimación del tiempo de desarrollo del test.
- ❖ Estimación del tiempo de ejecución del test.

# Validación

---

- Estimación de la cantidad de test necesarios.

La estimación de la cantidad de test necesarios depende de los requerimientos encontrados en la Especificación de Requerimientos del Software (ERS).

Según Humphrey, no existe una forma mágica de seleccionar una cantidad suficiente de conjunto de pruebas a practicas.

# Validación

---

- El objetivo es testear de forma razonablemente completa todas las clases válidas para la operación normal. Y testear de forma exhaustiva comportamientos inusuales y condiciones ilegales.
- En base a ello, se determinan los recursos, gente y equipo, y se elabora un calendario realista y exacto.

# Validación

---

- La estimación del test refleja el número de pruebas requeridas basada en factores como:
  - ❖ Testear todas las características y funciones definidas en el documento ERS y documentos relacionados.
  - ❖ Incluir un número apropiado de pruebas ALAC (Act Like a Customer).
  - ❖ Alcanzar una meta de cobertura de Test.
  - ❖ Alcanzar una meta de confiabilidad de software.

# Validación

---

- Para realizar la estimación:

Revisar cada requerimiento del ERS, basado en experiencias anteriores, estimar el número de tests requeridos para determinar si el software ha cumplido con los requerimientos.

**Elaborar una tabla como la siguiente:**

# Validación

---

Referencia al ERS	No. Estimado de tests requeridos	Notas
1.1.1	3	2 test positivos, 1 test negativo
1.1.2	2	2 test automatizados
1.1.3	4	4 test manuales
1.1.4	2	2 tests de usabilidad
...		
Total	165	

Estimación de Tests requeridos

# Validación

---

- Además de los test que se encuentran relacionados directamente al ERS, deben desarrollarse un número razonable de test ALAC, representativos del comportamiento del cliente ante el producto.
- Esto significa desarrollar pruebas:
  - ✓ Hacer lo incorrecto. *“Do it wrong”*.
  - ✓ Usar combinaciones de entradas erróneas o ilegales.
  - ✓ No hacer lo suficiente. *“Don’t do enough”*
  - ✓ Hacer nada. *“Do nothing”*.
  - ✓ Hacer demasiado. *“Do too much”*.



# Validación

---

- Cuando se lleve a cabo la estimación del número de tests requeridos para cada función o característica, tomar en cuenta los siguientes aspectos:
  - ✓ **Complejidad del Test:** Es mejor desarrollar varias pruebas pequeñas que pocas de gran tamaño.
  - ✓ **Plataformas diferentes:** Las pruebas requieren modificarse para diferentes S.O?
  - ✓ **Test automáticos o manuales:** Toma más tiempo crear tests automatizados, pero requieren muy poca o ninguna intervención humana en su ejecución.

# Validación

---

- La primera vez que se lleva a cabo una estimación encontrarán que el estimado y el número actual de pruebas desarrolladas es muy diferente.

## Aprender de la experiencia pasada

Se recomienda elaborar un análisis Post-mortem, para así determinar las discrepancias.

Usar estimaciones de proyectos pasados para apoyar el desarrollo de nuevas estimaciones.

# Validación

---

- Una vez que se ha estimado el número de test requeridos, el siguiente paso es determinar cuánto esfuerzo se requiere para desarrollar estos tests.

**Nuevamente, acudimos a la experiencia pasada para desarrollar un estimado del tiempo de desarrollo de tests.**

# Validación

---

- Se emplea un promedio [**horas-persona/test**].
- El promedio de tiempo debe incluir tiempo requerido para:
  - a) Escribir el primer borrador del script de la prueba.
  - b) Ejecutar el test una vez.
  - c) Encontrar errores y revisar el script del test.

# Validación

---

## ■ Los primeros pasos...

- 1) Escribe unas cuantas pruebas.
- 2) Mide el tiempo requerido para escribirlas.
- 3) Ejecuta las pruebas una vez.
- 4) Mide el tiempo requerido para la ejecución.
- 5) Corrige los errores.
- 6) Mide el tiempo requerido para la corrección.
- 7) Realiza la sumatoria de los tiempos registrados en los pasos 2) 4) y 6) para cada prueba.
- 8) Promedia el tiempo requerido en el proceso para cada prueba.
- 9) Ajusta en el camino.

# Validación

---

- Una vez que se tiene el promedio de tiempo estimado para el desarrollo de las pruebas, elaborar la siguiente tabla:

<b>Número de tests estimados</b>	<b>Promedio de Tiempo de desarrollo de test (persona-horas/test)</b>	<b>Tiempo de desarrollo de pruebas estimado</b>
165	3.5	577.5

Tiempo de desarrollo de tests estimado

# Validación

---

- De forma similar se requiere la estimación del tiempo de ejecución del test. El cual involucra:
  1. El tiempo requerido para configurar los sistemas para las pruebas.
  2. Ejecutar las pruebas.
  3. Reportar los problemas.

# Validación

---

- Experiencia pasada para lograr una estimación del promedio.
- Agregar de un 25% a un 50% al tiempo de ejecución, para realizar testing de regresión. Este porcentaje variará dependiendo de:
  - a) Inspecciones requeridas.
  - b) Integridad del proceso de corrección de defectos.
  - c) Cantidad de código nuevo vs código reusado/modificado.
  - d) Cantidad de testing de unidad y de integración realizado.
  - e) Experiencia de los desarrolladores.



# Validación

---

Número estimado de tests	Promedio de tiempo de ejecución de test (horas-persona/test)	Tiempo estimado de ejecución de test (horas-persona)	Testing de regresión estimado (50%) (horas-persona)	Estimación total del tiempo de ejecución (horas-persona)
165	1.5	247.5	123.75	<b>371.25</b>

Tiempo de ejecución de test estimado

# Validación

---

- Con base en estas tres medidas:
  - ❖ Número de tests requeridos.
  - ❖ Tiempo de desarrollo de los tests.
  - ❖ Tiempo de ejecución del test.

Y los recursos disponibles, es posible desarrollar un calendario de testing de validación realista.

# Validación

---

- En la planeación del esfuerzo para realizar testing de validación, el plan debe incluir lo siguiente:
  - ✓ Estimación del Test.
  - ✓ Desarrollo del Test y validación informal.
  - ✓ Revisión para la preparación de la validación formal.
  - ✓ Criterio para finalizar el Test.

# Validación

---

- **Desarrollo del Test y validación informal.**
  - ✓ La tarea del desarrollo del test incluye el trabajo requerido para escribir los tests identificados en el plan del test.
  - ✓ Pueden ser empleados diferentes métodos y tipos de tests:

# Validación

Nivel del Test	Objetivos	Realizado por	Ambiente de prueba	Métodos del Test
Unidad	Encontrar errores en lógica, datos y algoritmos en módulos individuales.	Ingenieros de software	Aislado.	Caja blanca
Integración	Encontrar defectos en interfaces entre módulos.	Ingenieros de software	Aislado o simulado.	Caja blanca Top-down Botom-up
Validación	Determinar si el software cumple con el ERS.	QA	Actual.	Funcional y ALAC
Regresión	Determinar si el software aún cumple con el ERS después de cambios.	QA	Actual.	Funcional y ALAC
Aceptación	Determinar si el software cumple con los requerimientos del cliente.	Cliente, QA, equipo del proyecto.	Actual (usualmente en el sitio del cliente)	Funcional y ALAC

# Validación

---

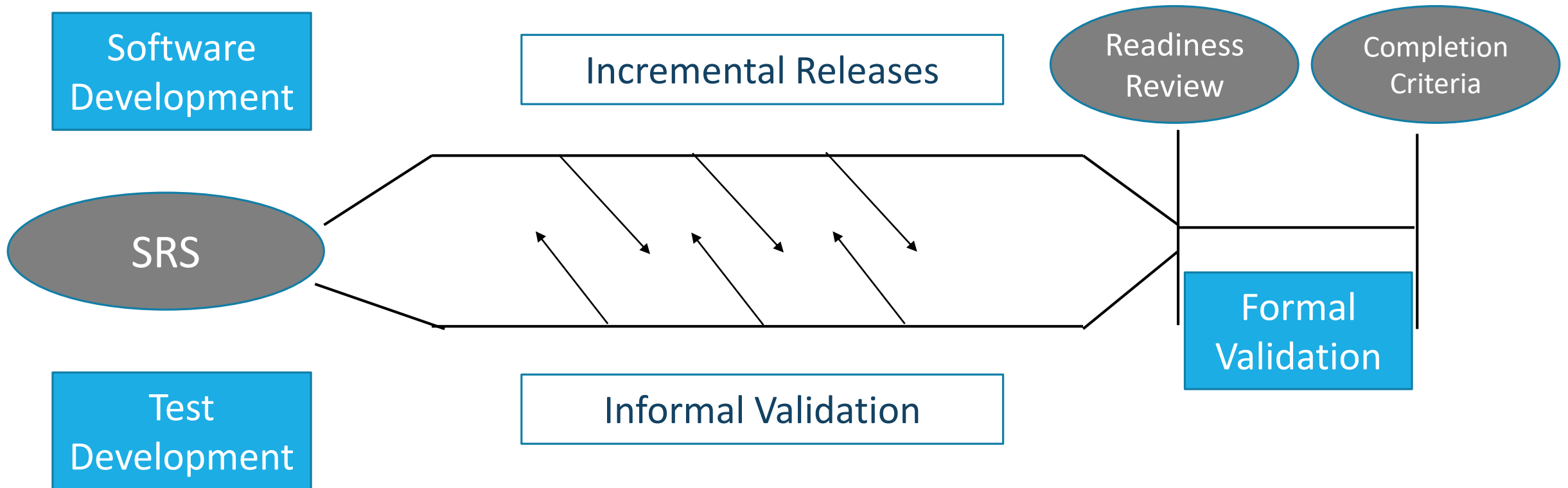
- **White box.** Usar el conocimiento de cómo se encuentra diseñado el software para maximizar la efectividad del test eliminando pruebas repetidas del mismo código.
- **Black box.** Usarlo para testear características sin conocimiento alguno de cómo se encuentran implementadas las características del software.
- **ALAC.** Usar el conocimiento de cómo los clientes emplearán el producto para desarrollar pruebas.

# Validación

---

- El plan de test debe describir cómo el trabajo de desarrollo del test es repartido entre el equipo QA.
- Esta tarea debe sincronizarse con el plan de desarrollo de software; de tal forma que las pruebas sean desarrolladas en el mismo orden que las características son implementadas.

# Validación



**Concurrent development/validation testing model**



# Validación

---

- Como ya se analizó anteriormente, **la validación informal**, involucra desarrollo y ejecución de pruebas de forma concurrente con el desarrollo de la funcionalidad.
- En esta sección del plan debe especificarse lo siguiente:
  - i. La manera en la que se desarrollarán y ejecutarán las pruebas para las funcionalidades implementadas.
  - ii. Cómo se reportarán los problemas encontrados durante la validación informal a los desarrolladores.
  - iii. Cómo el equipo de QA rastreará los problemas y verificará la solución.

# Validación

---

- En la planeación del esfuerzo para realizar testing de validación, el plan debe incluir lo siguiente:
  - ✓ Estimación del Test.
  - ✓ Desarrollo del Test y validación informal.
  - ✓ Revisión para la preparación y la validación formal.
  - ✓ Criterio para finalizar el Test.

# Validación

---

- **Revisión para la preparación y la validación formal.**

En esta sección debe establecerse el criterio para la Revisión de la preparación de la validación.

Ya se analizaron algunos ejemplos:

# Validación

---

- El desarrollo de software se encuentra completo.  
(Definiendo previamente lo que significa “completo”).
- El plan de pruebas ha sido revisado, aprobado y se encuentra bajo un control de documentos.
- Se han ejecutado las inspecciones de diseño.
- Se han realizado las inspecciones de código a los módulos críticos.

# Validación

---

- Todos los guiones de las pruebas se encuentran completos y el procedimiento para el test de validación está bajo control de documentos.
- Los guiones de prueba seleccionados han sido revisados.
- Todos los guiones de prueba han sido ejecutados al menos una vez.
- Las herramientas de administración de configuración (control de cambios) se encuentran listas.

# Validación

---

- Los procedimientos para el reporte de problemas se encuentran listos.
- El criterio para completar el testing de validación ha sido desarrollado, revisado y aprobado.

# Validación

---

- Asimismo, en el plan de test debe describirse el proceso de validación formal, incluyendo las siguientes actividades:

# Validación

---

- En la validación formal, se llevan a cabo las siguientes actividades:
  1. Se ejecutan las mismas pruebas que fueron realizadas en la validación informal, registrando los resultados.
  2. Los Reportes de Problemas de Software (RPS), se presentan para cada prueba que falla. (P.e. el módulo x no cubre el requerimiento y).



# Validación

---

3. Se realiza un seguimiento al **Reporte de Problemas de Software (RPS)** actualizando el estatus (abierto, corregido, verificado, no es un defecto).
4. Para cada defecto corregido, el RPS identifica los módulos modificados para arreglar dicho defecto.
5. Se realiza un seguimiento al control de cambios, asegurando que solo son modificados los módulos indicados, sin agregar nuevas funcionalidades o características.

# Validación

---

6. Se realizan revisiones de código informales para asegurar que no se introducen nuevos defectos.
7. Se registra el tiempo requerido para encontrar y corregir el defecto.
8. Se realiza testeo de Regresión (ejecutar testeo de validación nuevamente para asegurar que la corrección de errores no introdujo nuevos errores).

# Validación

---

9. Dar seguimiento al estatus de los test (éxito, fallido, no ejecutado)
10. Registrar el tiempo de testeo acumulado.

# Validación

---

- En la planeación del esfuerzo para realizar testing de validación, el plan debe incluir lo siguiente:
  - ✓ Estimación del Test.
  - ✓ Desarrollo del Test y validación informal.
  - ✓ Revisión para la preparación y la validación formal.
  - ✓ Criterio para finalizar el Test.

# Validación

---

- **Criterio para finalizar el Test.**

El plan de test debe incluir el criterio a cumplir para finalizar con la actividad de testeo.

El criterio debe ser objetivo, fácilmente medible y convenido previamente con el administrador del proyecto.

Algunos ejemplos mencionados anteriormente son:

# Validación

---

- i. Todos los guiones de prueba han sido ejecutados.
- ii. Reporte de Problemas de Software (RPS) han sido resueltos de forma satisfactoria (corregidos, para un lanzamiento posterior, no son defectos, etc).

Todos los defectos de alta prioridad deben ser corregidos.

- iii. Todos los cambios realizados como resultado del RPS han sido testeados.

# Validación

---

- iv. Toda la documentación asociada con el software han sido actualizados para reflejar los cambios realizados durante las pruebas de validación.
- v. El plan de pruebas ha sido revisado y aprobado.

# Validación

---

- ✓ Plan del Test.
- ✓ Procedimiento del Test.
- ✓ Reporte del Test.



# Validación

---

- Procedimiento del Test.
  - Es un documento contenedor de todos los test individuales (scripts de test) que se ejecutarán.
  - Debe contener una copia limpia de todos los scripts de los test sin ser ejecutados.

# Validación

---

- Los scripts del Test definen los pasos detallados que determinan si el software cumple un requerimiento específico.
- Una parte integral de cada script de test es el resultado esperado.

# Validación

---

- ✓ Plan del Test.
- ✓ Procedimiento del Test.
- ✓ Reporte del Test.

# Validación

---

- Reporte del Test.
- Documento que contiene los scripts de test (del procedimiento del Test) una vez ejecutados.

# Validación

---

- Reporte del Test.
  - El reporte del Test documenta los resultados del proceso de validación formal del software.
  - La información encontrada en un reporte de Test incluye:
    - Copia completa de cada script del test con la evidencia de que ha sido ejecutado (p.e. con fecha y firma de la persona que ejecutó el test).

# Validación

---

- Copia del Reporte de problemas del software-Software Problem Report (SPR).
- Lista de los SPR abiertos o sin resolver.
- Identificación de los SPRs encontrados en cada “línea base”.
- Tests de regresión ejecutados para cada software de línea base.

El resto del reporte debe contener información valiosa acerca de los hallazgos localizados durante la actividad de validación formal. Misma que será útil en el análisis postmortem y en el análisis de causa.

# Validación

---

- Anexos.
- ✓ Plantillas para:
  - Software Validation Test Plan.
  - Software Validation Test Procedure.
  - Software Validation Test Report.
  - Software Validation Test Script.
  - Software Release Procedure

# Control de cambios

---

- **Líneas de referencia o líneas base.**

Una especificación o producto que se revisó formalmente y con el que se estuvo de acuerdo, que a partir de entonces sirve como base para un mayor desarrollo y que puede cambiar sólo a través de procedimientos de control de cambio formal.



# Control de cambios

---

- Antes de que un ítem de configuración del software (información creada como parte del proceso de ingeniería del software), se convierta en línea de referencia, los cambios pueden realizarse rápida e informalmente.
- Una vez establecida la línea de referencia, pueden realizarse cambios, pero debe aplicarse un procedimiento formal específico para evaluar y verificar cada uno de ellos

# Control de cambios

---

- Una línea de referencia es un hito en el desarrollo del software.
- Una línea de referencia se marca al entregar uno o más ítems de configuración del software que se aprobaron como consecuencia de una revisión técnica.

# Control de cambios

---

■ Por ejemplo:

1. Los elementos de un modelo de diseño se documentaron y revisaron.
2. Se encontraron y corrigieron errores.
3. Una vez que todas las partes del modelo se revisaron, corrigieron y luego aprobaron, el modelo de diseño se convierte en línea de referencia.
4. Los cambios adicionales a la arquitectura del programa (documentada en el modelo de diseño) pueden realizarse sólo después de que cada uno se evalúa y aprueba.

# Control de cambios

---

- Después de revisar y aprobar los ICS, se colocan en una base de datos del proyecto (también llamada librería de proyecto o repositorio de software).
- Cuando un miembro de un equipo de ingeniería de software quiere hacer una modificación a un ICS que se ha convertido en línea de referencia, se copia de la base de datos del proyecto en el espacio de trabajo privado del ingeniero.
- Sin embargo, este ICS extraído puede modificarse solamente si se siguen controles ACS

# GRACIAS POR SU ATENCIÓN

---