

VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

Bibliografía

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

Validación

Validación

- Recapitulemos definiciones:
- ✓ **Verificación.** Proceso de evaluar un sistema o componente para determinar si los productos de una fase determinada satisfacen las condiciones impuestas al inicio de la fase.

¿ESTAMOS CONSTRUYENDO EL PRODUCTO DE FORMA CORRECTA?

Validación

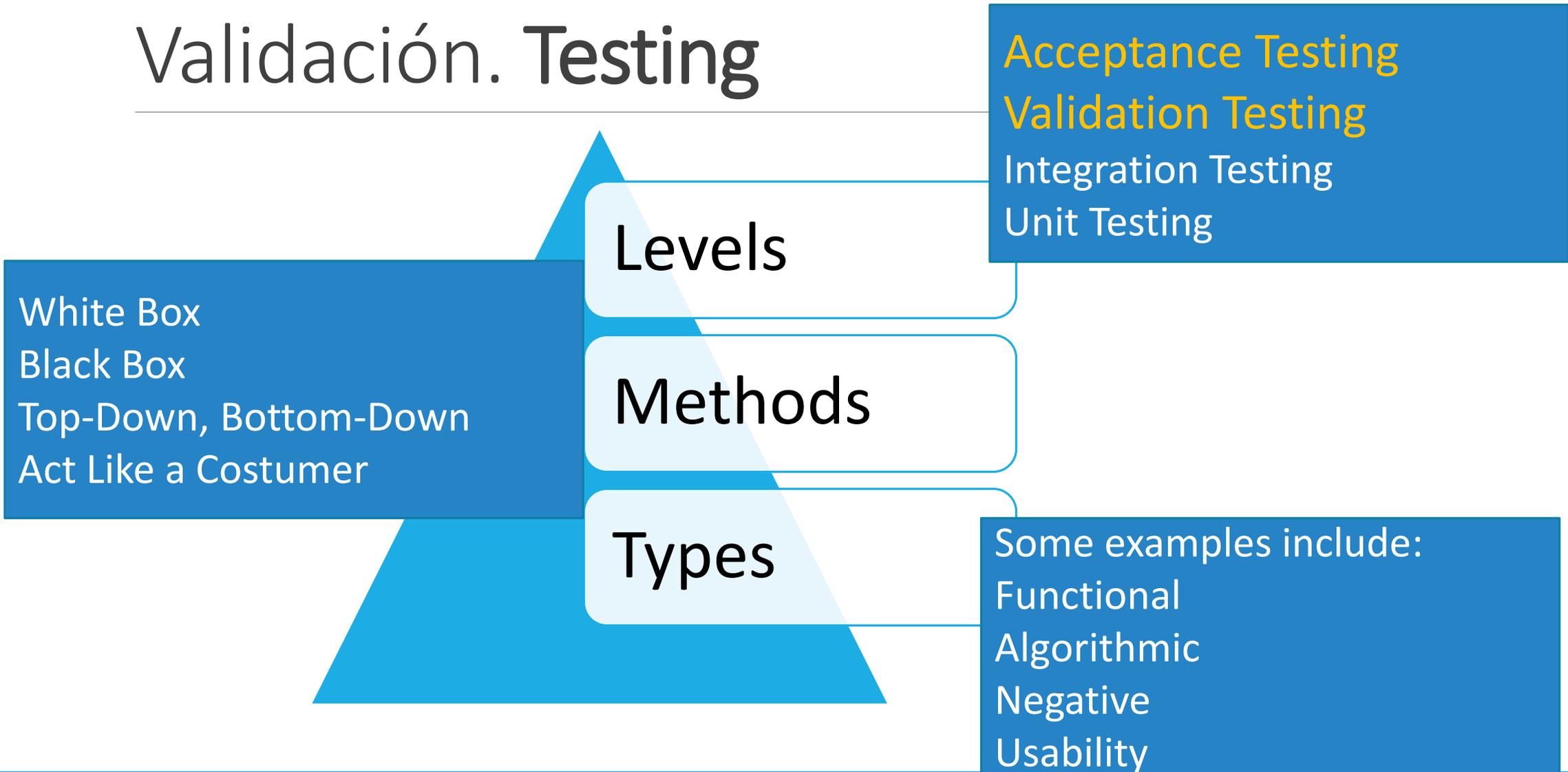
- ✓ **Validación.** El proceso de evaluar un sistema o componente durante o al final de un proceso de desarrollo para determinar si satisface los requerimientos especificados.

¿CONTRUIMOS EL PRODUCTO CORRECTO?

Validación

- Las actividades de validación están definidas alrededor de tres procesos básicos:
 - ✓ Testing
 - ✓ Medición
 - ✓ Confiabilidad del software

Validación. Testing



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

- Los clientes frecuentemente se rehúsan a participar en esta actividad por dos razones:
 - a) No existe garantía de que los problemas encontrados por los clientes sean resueltos.
 - b) Se requiere invertir tiempo y recurso que el cliente generalmente no tiene.

Validación

- El testeo Beta puede ser una actividad útil para clientes y desarrolladores si es planeada y cuenta con un guion de tal forma que defina la secuencia de actividades; asimismo, los desarrolladores se comprometan a arreglar los defectos encontrados.

Validación

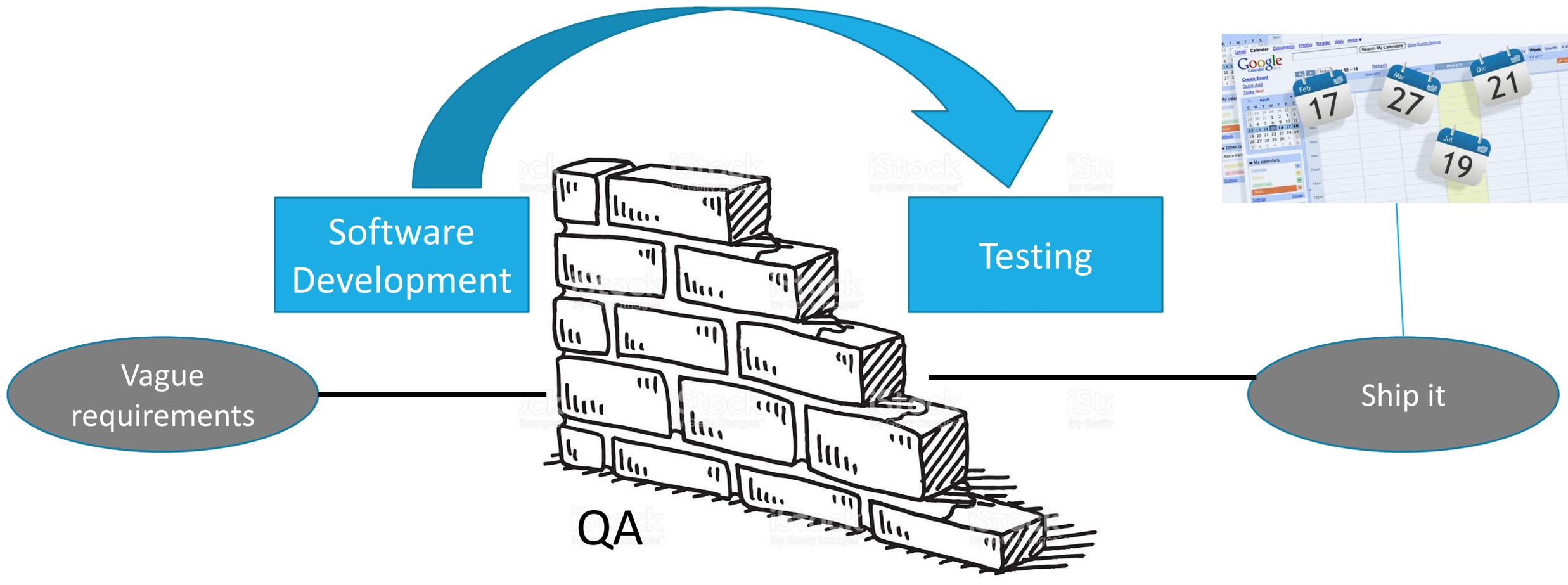
- Por lo tanto, para que el testeo **alpha y beta** sea más efectivo, se debe delimitar al cliente en los aspectos que debe enfocarse y otorgarles **escenarios** específicos de testeo.
- Establecer el **compromiso** ante los clientes de **corregir** los defectos encontrados durante los escenarios analizados; lo que será una motivación para que el cliente invierta los recursos necesarios en el testeo del software.

Validación

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

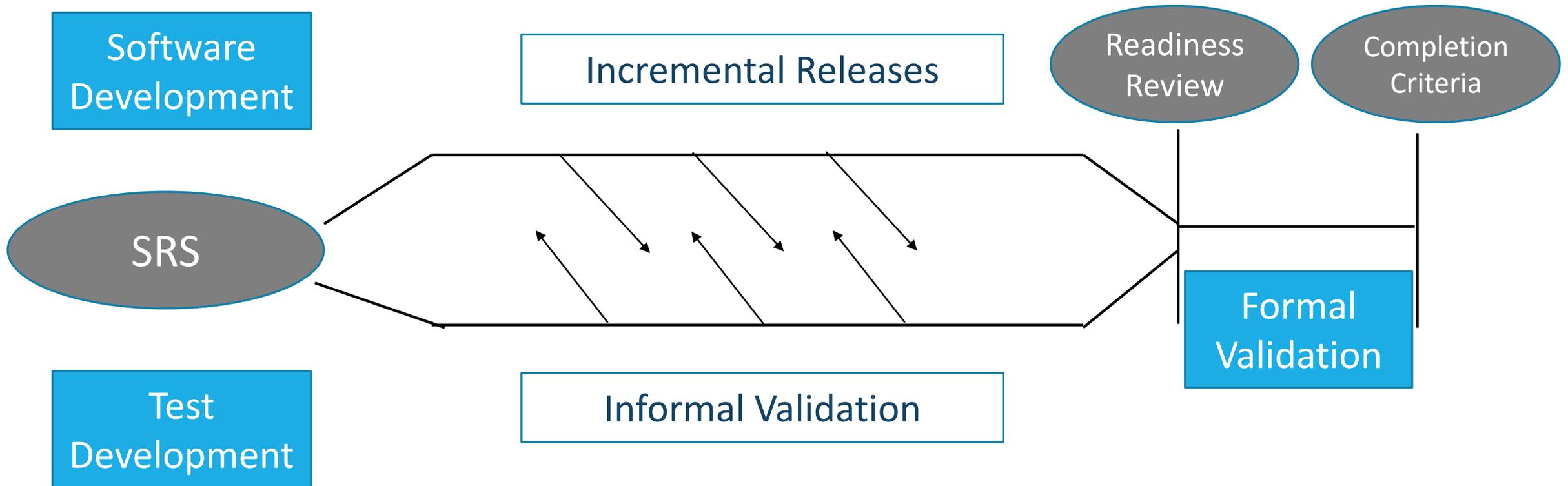
Modelo de testeo Desarrollo/Validación concurrente

Validación



Typical validation testing process

Validación



Concurrent development/validation testing model

Validación

1. En este modelo las actividades de desarrollo de software y testeo de validación, se basan en un SRS (software requirements specification) completo y actualizado.
2. Una vez que el SRS se encuentra disponible, el diseño y codificación inicia.
3. Y de forma similar las actividades de validación, como la planeación del testeo y las actividades de desarrollo.
4. La clave de este modelo concurrente es que ambos se basan en el SRS.
5. Una vez que se termina con la codificación, el testeo unitario y de integración, el producto se pone disponible al equipo QA.
6. Mientras se desarrolla y prueba el producto, QA planea el desarrollo de pruebas basado en el SRS.
7. QA inicia el testeo de las piezas de forma incremental.

Validación

- Validación informal.
- ✓ Conforme se desarrolla el producto, se proporcionan entregas incrementales al equipo QA, que desarrolla sus test en el mismo orden que las características desarrolladas en el incremento.

Validación

- Una vez que la codificación se ha completado, todas las pruebas de validación deben estar escritas y ejecutarse al menos una vez.
- Este proceso es llamado Validación informal, pues las pruebas son ejecutadas informalmente; algunas características no se testearán y algunas pruebas fallarán.

Validación

- La validación informal:

- ✓ Provee una oportunidad para que las pruebas de validación y depurado se desarrollen de manera temprana en el proceso de desarrollo.
- ✓ Provee retroalimentación temprana a los ingenieros de software.
- ✓ Da como resultado que la validación formal sea menos agitada, pues la mayoría de los problemas han sido encontrados y corregidos.

Validación

- Revisión de preparación de validación (Validation Readiness Review).
 - ✓ El propósito de esta revisión es asegurar que todo se encuentra en orden antes de iniciar con la validación formal.
 - ✓ Las principales diferencias entre validación informal y validación formal son:

Validación

- Durante la validación informal, los desarrolladores pueden hacer los cambios necesarios para cumplir con el SRS.
- Durante la validación informal el equipo QA, ejecuta pruebas y hace cambios en las mismas para cumplir con el SRS.

Validación

- Durante la validación formal los únicos cambios que pueden realizarse al código son correcciones de defectos, en respuesta al reporte efectuado durante la validación formal. No pueden agregarse nuevas características.
- Durante la validación formal se ejecuta el mismo conjunto de pruebas ejecutado en la validación informal. No se agregan nuevos tests.

Validación

- El propósito de esta revisión es asegurar que la validación formal inicie solo cuando el proyecto se encuentre listo, pues iniciar la validación formal de manera prematura puede resultar en un desperdicio de esfuerzo, incrementar la frustración y presionar la liberación de productos con demasiados defectos.

Validación

- El control de versiones y la administración de la configuración, son criterios esenciales para iniciar el test de validación.
- El plan de testeo debería definir el criterio a cumplir antes de iniciar la validación formal. Algunos ejemplos son:

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

- **Validación Formal.**
- Después de la Revisión de preparación de validación (Validation Readiness Review) y los criterios se han cumplido, inicia la **validación formal**.
- En este punto los cambios al software están restringidos a la corrección de defectos, **no** se agregan nuevas funcionalidades.

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

- Un elemento clave para un testeo efectivo es saber cuando detener el proceso de testeo.
- Es importante tener un objetivo, definir criterios de completitud medibles, revisados y aprobados en el proceso temprano de desarrollo.
- Algunos ejemplos de criterios de completitud 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.

GRACIAS POR SU ATENCIÓN
