Pruebas a través de Revisiones

Pruebas de Sistemas

UNIVERSIDAD SIGLO 21

Objeto de pruebas y sus procesos

Pruebas a través de revisiones

Antes de comenzar con los procesos de prueba de los distintos objetos que se pueden probar en un proceso de desarrollo, daremos una introducción sobre lo que es el proceso de desarrollo y el proceso de pruebas y su evolución.

La industria del *software* ha ido estableciendo, durante el transcurrir de las diferentes décadas, un *proceso básico de desarrollo* de software, que se ha ido consensuando y aceptando. Contar con una serie de pasos o actividades que deben llevarse a cabo con el fin de construir un sistema como producto funcionando es fundamental, como así también esta idea de fases es central para establecer cualquier *metodología*.

Bajo la idea de fases se basa cualquier proceso, y existen así diferentes *modelos de ciclo de vida*, según cómo se lleven adelante esas fases.

La mayoría de las organizaciones basan su proceso de desarrollo de software en alguno de esos modelos de ciclo de vida y también en estándares dedicados al desarrollo de software. Entonces, se tiene un proceso de desarrollo de software basado en alguno de los modelos de ciclo de vida, que, junto con estándares y procedimientos de trabajo, construyen una metodología.

¿Cómo se conforma un proceso de desarrollo básico?

Como ya se mencionó más arriba, toda la actividad de desarrollo se subdivide en distintas fases principales, cada una de ellas con sus productos finales claves que le son propios. La mayoría de las organizaciones, cualquiera sea el modelo de ciclo de vida que se adopte, enmarcan las fases básicas de *análisis, diseño, implementación* y *mantenimiento* como fases de desarrollo, mientras que las fases particulares pueden variar de organización en organización.

Esto implica que se parte de una idea o solicitud que está en la cabeza de alguien y se termina en un sistema operacional y funcionando.

A continuación, se brinda una reseña de cada una de las fases principales antes mencionadas.

Tabla 1: Fases de desarrollo

Fase	Descripción
Análisis	Determina la viabilidad y especificación de los requerimientos
Diseño	Especifica el diseño general y detallado
Implementación	Implica: codificación, pruebas, depuración e instalación
Mantenimiento	Mejora y modificación

Fuente: elaboración propia.

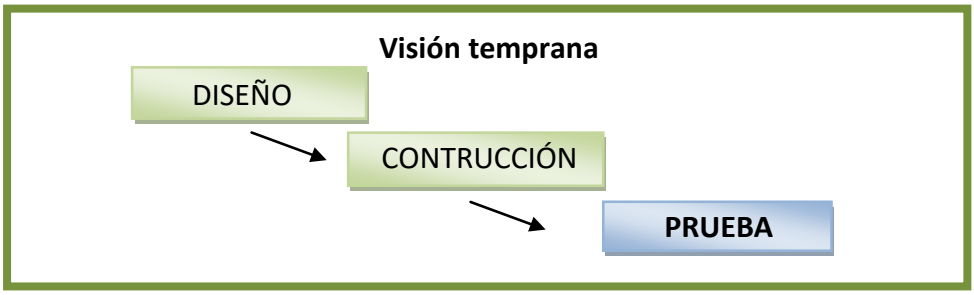
Como se puede observar, la fase de implementación incluye tanto la programación como la *prueba* y también la puesta en producción.

Es frecuente encontrar en muchas organizaciones que cada una de estas actividades (de codificación, pruebas, depuración e instalación), dentro de la fase *implementación*, son tenidas como fases separadas, esto es, que esta fase se encontraría distinguiblemente explosionada. No obstante, para los efectos de lo que vamos a desarrollar, es correcto verlo de cualquiera de las dos maneras, y de otras también. Lo importante es no dejar de ver las fases básicas, independientemente de qué actividades contengan dentro.

Tener una metodología de desarrollo de software efectiva significa que, cada uno de los pasos son definidos y detallados para cada fase de desarrollo. Los pasos para realizar las pruebas deberían ser un componente de la metodología general del software. Pero como siempre, en la práctica, las actividades de pruebas son definidas a la ligera o de una manera muy liviana, sin muchas especificaciones. Aun cuando están los procedimientos de pruebas descriptos, en la mayoría de las organizaciones están desactualizados y así ya no son efectivos.

En una visión temprana, la prueba de sistemas era vista como una fase de desarrollo, después de programar. “Los sistemas eran diseñados, construidos, y después probados y depurados” (Uninotas, 2017, https://bit.ly/2KGaC4d).

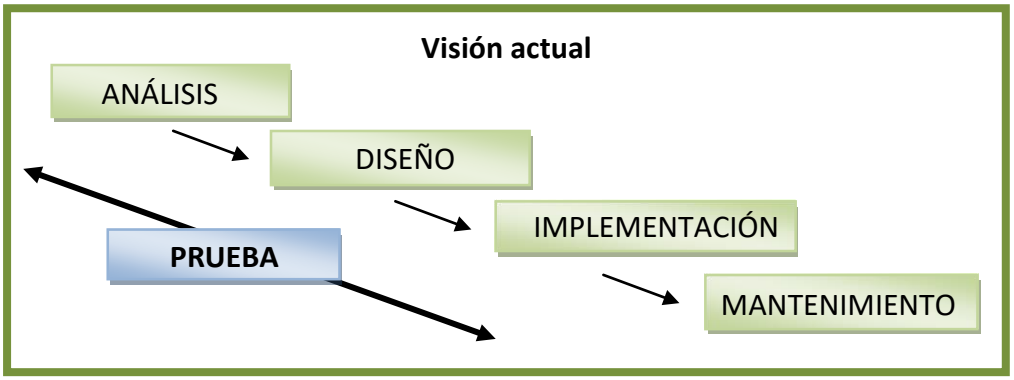
Figura 1: Visión temprana



Fuente: elaboración propia.

A medida que las visiones de las pruebas fueron madurando, tal como se puede apreciar en la perspectiva histórica de las pruebas, se reconoce que la visión correcta de éstas respecto del proceso de desarrollo, es aquella que tiene al ciclo de vida de las pruebas embebido en el ciclo de vida de desarrollo. Es decir, las pruebas acompañando el proceso de desarrollo y no como una fase más, llevada adelante una vez que la codificación se realizó.

Figura 2: Visión actual



Fuente: elaboración propia.

La prueba debe ser desarrollada de manera paralela al proceso de desarrollo y como una actividad que tiene sus propias fases de *análisis, diseño, implementación, ejecución* y *mantenimiento*.

Tabla 2: Prueba

Fase	Descripción
Análisis	Planifica y determina los objetivos y requerimientos de pruebas
Diseño	Especifica los casos de prueba para ser ejecutados
Implementación	Construye los procedimientos de ejecución
Ejecución	Ejecuta y re-ejecuta los casos de prueba
Mantenimiento	Graba y actualiza los casos de prueba en función de los cambios de software

Fuente: elaboración propia.

Las revisiones como una técnica de pruebas

Es justificadamente necesario tener una metodología de pruebas bien integrada con el proceso de desarrollo. En un proyecto de software, su avance no puede ser medido únicamente contando las tareas terminadas. Un gerente de proyecto de software no puede esperar a que el sistema esté construido para evaluar su calidad, pues puede llegarse a las etapas finales del proyecto y concluirse que tiene que ser rehecho porque no está en acuerdo con lo que fue solicitado.

El papel del proceso de pruebas, en el proceso de desarrollo, es fundamental para mejorar la calidad y reducir el riesgo de detectar errores durante la fase de operaciones del sistema.

Haciendo un análisis y mediciones en cualquier proyecto de desarrollo, se concluiría tal como las estadísticas que da el International Software Testing Qualifications Board (ISTQB) (2011), según Boehm (1981), que responden a la siguiente premisa:

El costo de eliminar desperfectos aumenta con el tiempo de persistencia del defecto en el sistema. Por ello es importante remarcar que el descubrimiento de errores en algún periodo temprano, posibilita la corrección de los mismos a bajos costos. Dicho en otras palabras, cuanto antes se detecta un defecto en el producto, más económico es corregirlo. Así emerge la visión de que los defectos deben ser detectados tan pronto

como sea posible respecto del proceso y que no es necesario para ello la evaluación del producto recién cuando este pueda ser ejecutado.

Surgen las *pruebas estáticas*, que son la evaluación sin la ejecución del programa.

Dentro de las llamadas pruebas estáticas, están contempladas las *revisiones*.

Si bien muchos consideran las revisiones y las pruebas como dos cosas completamente diferentes, es decir, dentro de las pruebas no incluyen a las revisiones, esto responde a una vieja visión de las pruebas, donde probar significaba ejecutar casos de pruebas. Ya con el pensamiento de que las pruebas miden la calidad del software, está claro que las revisiones son una clase general de métodos de pruebas.

Es así como Hetzel (1988) afirma que las revisiones son la única técnica de pruebas de software disponible en las etapas tempranas de desarrollo, y que esencialmente son vistas como pruebas sobre las que se tiene la seguridad de que son efectivas.

¿Qué puede revisarse?

Existen diferentes tipos de revisiones dadas por el objeto revisado. Pueden ser revisiones:

- **Procedimentales**: tienen que ver con revisar cómo se está llevando a cabo la tarea respecto de lo que está establecido en el proceso, procedimiento o instructivo. (Uninotas, 2017, https://bit.ly/2KGaC4d)
- **Documentales**: revisa la documentación y registros que se generan a través del proceso, revisando que documentalmente esté de acuerdo a lo establecido, que se haya empleado el documento patrón correcto, que sea legible, que se entienda, entre otros. Ejemplos de ello pueden ser: revisión de plantillas de casos de pruebas generados, plan de pruebas, plan de proyecto. (Uninotas, 2017, https://bit.ly/2KGaC4d)
- **De producto**: son las revisiones de los productos generados a través del proceso de desarrollo. Ejemplo: revisiones de requerimientos, especificaciones, diseño, código, casos de pruebas, planes de proyecto, entre otros. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Por lo descripto anteriormente, se puede decir que “todo puede ser revisado” en el proceso de desarrollo, y que algunas revisiones apuntarán al proceso de cómo se está llevando a cabo y otras, al producto.

Cada organización establecerá los tipos más convenientes a su necesidad, pudiendo variar incluso de proyecto en proyecto, como así también los

momentos de llevarlas a cabo. Si bien todo puede ser revisado, descartado está, que tiene un costo, y entonces es necesario satisfacer la relación costo-beneficio.

Cualquiera sea el tipo de revisión, el objetivo es obtener información de confianza, en general, del estado o calidad del trabajo realizado.

Plan de revisiones

Definidas las revisiones como técnica de pruebas, sean formales o informales, las mismas requieren que se planifique cuidadosamente: qué se revisará, cuáles son los resultados esperados, y quiénes las llevarán a cabo. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Para poder lograr un proceso de pruebas efectivo, es necesario seleccionar una serie de puntos de control formal del proceso de desarrollo. ¿Y cómo se hace eso? Pues bien, si tenemos un conocimiento claro de las debilidades de nuestro proceso de desarrollo y de nuestro equipo, podemos establecer controles en los puntos donde se asume que se cometerán más errores. También es de suponer, de acuerdo con las estadísticas, que muchos de los defectos detectados en el producto final tienen su origen en los requerimientos, por lo que incorporar revisiones en este punto podría ser fundamental.

Todo esto variará de organización en organización, de proyecto en proyecto, según los criterios que se empleen para determinar lo puntos que requieren control.

Por regla general de técnicas estáticas, las revisiones pueden estar en cualquier punto del proceso de desarrollo.

Cualquier plan que se establezca sirve para estructurar la revisión y responder a las preguntas: ¿qué debe ser revisado?, ¿cuándo se tiene que parar? y ¿quién hace o quiénes participan de la revisión?

Mínimamente, un *plan de revisiones* tiene que especificar:

- Quién / quiénes asisten.
- Información que es necesaria antes de la revisión.
- Precondiciones que se deben alcanzar antes de llevar a cabo la revisión.
- Listas de chequeos (también llamadas listas de verificación o checklists[1]) u otras herramientas de indicaciones de qué será cubierto.

[1] Listas de chequeos, listas de comprobación, listas de verificación, *checklist* se usarán como sinónimos.

- Criterios o condiciones finales que deben alcanzarse al concluir con la revisión.
- Registros y documentación que debe ser guardada.
- Responsabilidades asignadas para cada instancia.

También deben asignarse las responsabilidades de preparar y entrenar a los participantes de la revisión. (Uninotas, 2017, https://bit.ly/2KGaC4d).

Se debe contar con personal preparado para todas las instancias de la revisión, caso contrario, puede llegar a ser un fracaso o no encontrarse ningún beneficio a partir de ella.

En las revisiones formales son cada vez más utilizadas las *listas de verificación* por los beneficios que traen consigo, a saber:

- Estructura la revisión.
- Medio para la registración de los resultados.
- Una guía para la actividad de revisión.
- Medio para aprender de instancias pasadas.
- Asegura cobertura sistemática y completa.
- Herramienta para cuantificar y medir resultados. (Uninotas, 2017, https://bit.ly/2KGaC4d).

Si bien pueden adquirirse listas de verificación ya establecidas, lo mejor es desarrollarlas a medida, en función de la recolección de una lista de problemas que han ocurrido en varios proyectos y además incorporar todas aquellas nuevas preguntas a las nuevas preocupaciones de los problemas surgidos. “Utilizar listas de verificación que son renovadas permanente y periódicamente genera mejoras, y asegura que las organizaciones aprenden de sus errores pasados”. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Tipos de revisiones

> Ya se mencionó que según el objeto revisado existen tres tipos de revisiones: documentales, procedimentales y de producto.
>
> Y según el modo en que se lleven adelante, las mismas podrán ser formales o informales.
>
> Se llaman revisiones **formales** cuando los profesionales sienten la necesidad de llevar las evaluaciones de manera precisa y generan reportes escritos de sus hallazgos a la gerencia o a la dirección. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Para los problemas u objetos técnicos se eligen en general las revisiones formales porque proveen información más confiable. A éstas se las suele llamar directamente *revisiones técnicas*.

"Las revisiones **informales** son muy diferentes ya que implica que los profesionales que participan compartan sus opiniones, en general no hay registros de hallazgos" (Uninotas, 2017, https://bit.ly/2KGaC4d). Tampoco se emplean listas de verificación como es el caso de las revisiones formales.

Cualquiera sea la definición que se dé, es bueno aclarar que toman forma según se las planifique y diseñe en cada organización o proyecto y eso debe estar dirigido por la necesidad.

La Junta Internacional de calificaciones para pruebas de software (*International Software Testing Qualification Board*), cuya sigla es *ISTQB* "presenta otros tipos de revisiones además de las revisiones formales o revisión técnica y de las revisiones informales ya presentadas"(Uninotas, 2017, https://bit.ly/2KGaC4d). Ellas son las inspecciones y los ensayos o revisiones guiadas.

No es que "Hetzel no las mencione, sino que las relaciona directamente con el objeto revisado. A las inspecciones las encuadra dentro del nombre revisiones lógicas de código y a los ensayos dentro de revisiones de diseño". (Uninotas, 2017, https://bit.ly/2KGaC4d)

Las **inspecciones** tienen las siguientes características:

- Usan listas de verificación (checklists) y métricas como, por ejemplo, problemas por página.
- Debe contar con un rol de moderador capacitado e independiente que dirija la revisión. Y cuando se dice capacitado implica con formación específica.
- Previamente a la revisión, se valora la viabilidad del objeto a someter a la revisión.
- Criterios de entrada y salida especificados (previamente) para la aceptación del producto software.
- Se puede decir que aplica un proceso formal incluyendo las actividades de preparación, ejecución, documentación y seguimiento. Ha preparación previa a la reunión.
- Muchas veces se la desarrolla como una evaluación entre pares.
- Se construye un informe de inspección incluyendo la lista de hallazgos. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Las otras son los ensayos o las **revisiones guiadas** como las denomina el *ISTQB*, y tienen las características siguientes:

- Opcionalmente puede haber una preparación de los revisores previa a la reunión.
- Son sesiones abiertas y la reunión está dirigida por el autor y éste oficia de moderador.
- Mientras se va realizando la presentación por el autor, los revisores tratan de detectar desviaciones y/o áreas que representen un problema. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Efectividad de las revisiones

¿Cuán efectiva es la revisión? La efectividad de cualquier técnica de pruebas depende de la ecuación entre la confianza obtenida o los defectos descubiertos o prevenidos versus el costo invertido en la prueba.

Los beneficios de realizar revisiones están centrados en la posibilidad de detectar tempranamente defectos y, además, en detectar defectos que otras técnicas tradicionales de pruebas no lo pueden hacer, o si pueden, requieren una gran inversión de tiempo; ni hablar del costo de reparar los defectos detectados en fases últimas de pruebas.

“Las revisiones detectan defectos en:

- Las especificaciones.
- El diseño y la arquitectura del software.
- En las especificaciones de interfaces" (Uninotas, 2017, https://bit.ly/2KGaC4d).

Además, detectan, por ejemplo, mantenibilidad [2] insuficiente y desviaciones con respecto a estándares acordados (por ejemplo, guías de programación).

Entre los beneficios de las revisiones efectivas están:

- Alto potencial de ahorro a costes más bajos.
- Los defectos en la documentación son detectados y corregidos de forma temprana, y esto lleva a que documentos de alta calidad mejoren el proceso de desarrollo.
- Mejora el índice de comunicación e intercambio de conocimiento. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Las revisiones son la única manera efectiva de realizar pruebas en fases tempranas del desarrollo de software. Hacer una revisión no es difícil, lo que puede costar es convertirla en resultados efectivos, así como sostenerlas sin que el equipo se sienta "en el ojo de la tormenta".

Contrariamente a los beneficios generados, se encuentran las desventajas que pueden presentarse:

- se podrían presentar situaciones de tensión en el caso de enfrentamientos directos con el autor;
- los involucrados en las revisiones deben adquirir conocimientos específicos del producto, lo que implica que es necesaria una buena preparación;
- los participantes (cualquiera sea el rol que se cumpla) influyen directamente en la calidad de la revisión.

Como todas las pruebas de sistemas (las revisiones no escapan), implican un costo o inversión (según como se quiera ver). Algunas estadísticas dicen que entre el 4 % y 8 % del costo total de un proyecto lo consumen

[2] Mantenibilidad es un término creado para la traducción de *mantenibility*, que significa la capacidad de un producto de ser mantenido en el tiempo.

los trabajos de revisión. Y en cuanto al tiempo, la inversión en él es considerable, del 10 % al 15 % del presupuesto total.

Frecuentemente se observan malas experiencias de revisiones, lo que trae aparejado que muchos profesionales tengan una vista distorsionada del valor que tienen.

Hetzel (1988) da una lista de factores críticos de éxito de las revisiones:

- Resultados esperados: es importante conocer el propósito de la revisión. ¿Qué es lo que se va a probar o medir?
- Responsabilidades: clara asignación de responsabilidades de todos los participantes.
- Derechos individuales: dar seguridad a los participantes que sus opiniones y sugerencias individuales estarán protegidas, ya que no es un comité.
- Asistentes: se debe involucrar a la gente correcta.
- Proceso estructurado: establecer procedimientos.
- Moderador: debe ser experto y entrenado en esa función.
- Registros: Reportes escritos y evaluación.
- Los primeros tres factores son generalmente los primeros en no cumplirse, o violarse. (Uninotas, 2017, https://bit.ly/2KGaC4d)

Por lo que se recomienda tenerlos bien en cuenta para asegurar un resultado positivo de la revisión.

Referencias

Boehm, B. (1981). *Software engineering economics.* New Jersey, USA **:** Prentice-Hall.

Hetzel, B. (1988). *An introduction. The complete guide to software testing* (2.da ed.). Massachusetts, USA: QED Information Science.

Uninotas. (2017). *Clases de equivalencia PRUEBAS.* Recuperado de: https://www.uninotas.net/clases-de-equivalencia-pruebas