

Actividad | #3 | Plan de pruebas y

Reglas para el Versionado

Aseguramiento de la Calidad

Ingeniería en Desarrollo de Software



TUTOR: Jessica Hernández Romero

ALUMNO: Pilar Barajas Cervantes

FECHA: 07/01/2024

Índice

Introducción.....	3
Descripción	4
Justificación	5
Plan de pruebas integral.....	6
Reglas para el versionado e incidencias	7
Conclusión	8
Referencias	9

Introducción

Para desarrollar software de la calidad y libres de errores, el plan de pruebas y los casos de pruebas son muy importantes. El software test plan (STP) se diseña para determinar el ambiente de ejecución de los recursos, así como el calendario de las actividades de las pruebas. Se debe de identificar el dominio y sus características a probar, lo mismo que el tipo de pruebas a realizar. El estándar 829 del IEEE es el indicado a seguir a diseñar un plan de pruebas. Un caso de pruebas es un conjunto de acciones con resultados y salidas previstas, se basa en los requisitos de especificaciones del sistema.

Una incidencia se define como un acceso, intento de acceso, uso divulgación, modificación o destrucción. No autorizada de información, así como un impedimento en la operación normal de las redes, sistemas o recursos informáticos; o una violación de la política de la seguridad entre otras. Es decir, una incidencia es cualquier ocurrencia de un suceso que requiere investigación (según IEEE 1008). El ciclo de vida de una incidencia se divide en diferentes fases.

- fase inicial.**
- Recuperación del incidente.**
- Actividad después del incidente.**

Descripción

El versionado del software es el proceso de asignación de un nombre, código o número único a un software para identificar su nivel de desarrollo. El proceso de versionado del software se refiere a una opción que permita identificar de forma unívoca las nuevas funciones, actualizaciones o los fixes corregidos. De modo que los cambios incluidos en el versionado de un software deban contar con la característica de aumentar las funcionalidades del sistema y solucionar sus errores sin modificar la aplicación o producto a nivel esencial. Cabe destacar que estas actualizaciones del sistema forman parte de la vida útil de un software, por lo que se requiere mantener un control de estas nuevas versiones. Dentro de las principales características para el versionado de software se encuentra que permite la identificación de bugs o problemas de funcionamiento a partir de una versión determinada. El control de versionado de código contribuye también a proporcionar la hoja de ruta de la aplicación o programa que permite observar que este avance o se actualice de acuerdo a lo estipulado. Además de esto se debe tener en cuenta, que para el proceso de versionado de software se requiere fijar siempre las librerías, así como los componentes que se usan. Esto ayudara a evitar los errores de utilización frente a la actualización de estos por parte del sistema o aplicación. También es importante destacar que el versionado de software puede incluir cambios en el sistema, en su estructura de DATABASE, en los protocolos implementados y demás.

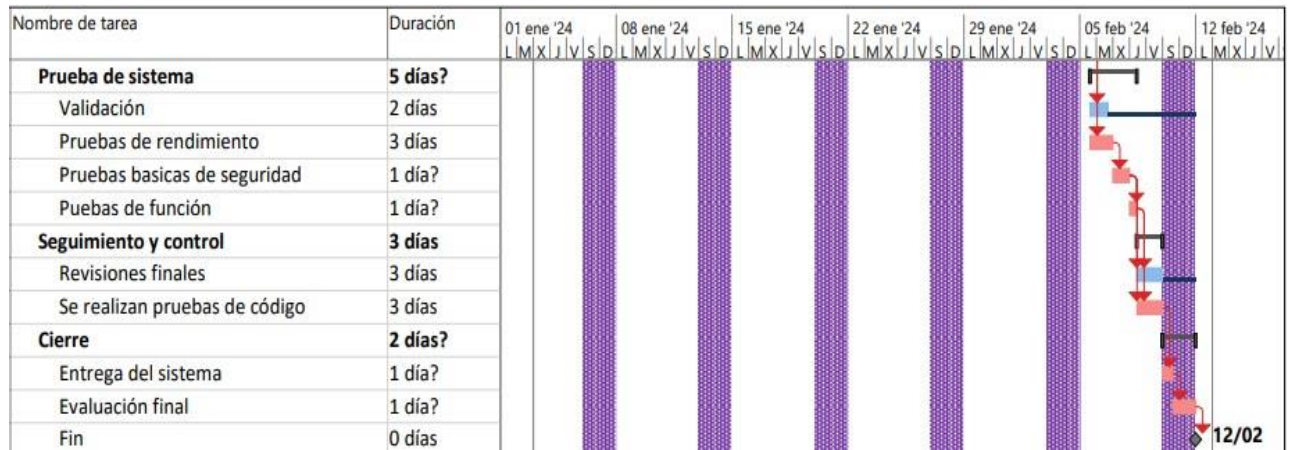
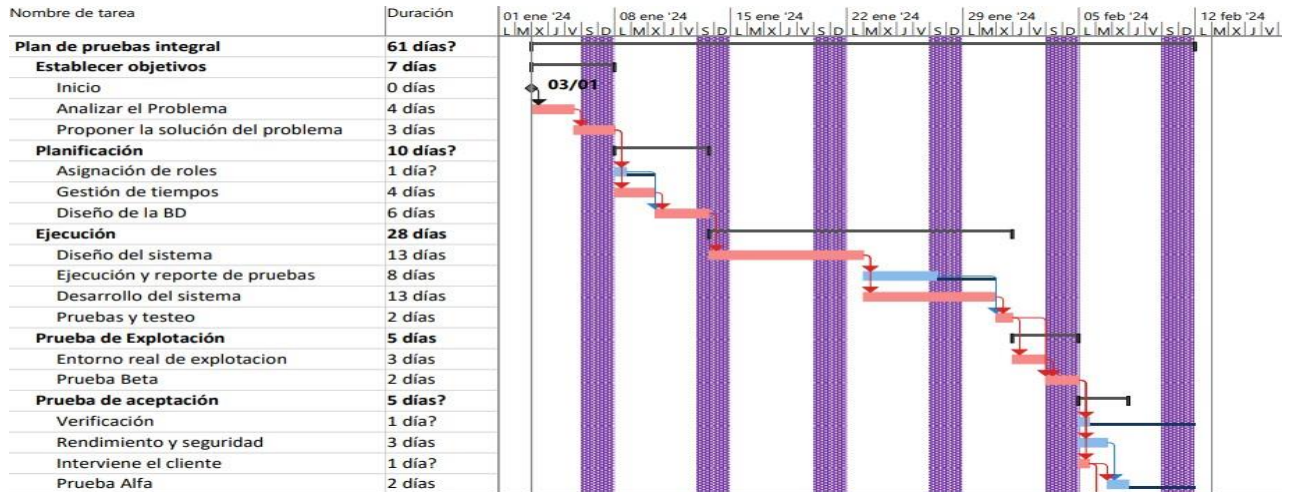
Justificación

El control de versiones sincroniza las versiones y garantiza que los cambios no entren en conflictos con los cambios de otros usuarios. El equipo se basa en control de versiones para ayudar y resolver conflictos incluso cuando los usuarios realizan cambios al mismo tiempo. El control de versiones mantiene un historial de los cambios a medida que el equipo guarda nuevas versiones del código. Los miembros del equipo pueden revisar el historial para averiguar la persona que realizó los cambios, por qué los hizo y en qué momento. El historial ofrece a los equipos la confianza de experimentar ya que fácil revertir a una versión anterior correcta en cualquier momento. El historial permite que cualquier persona base el trabajo en cualquier versión del código, como corregir un error en una versión anterior.

Las características de automatización de control de versiones ahorran tiempo y generan resultados coherentes. La automatización de prueba, el análisis de código y la implementación cuando se guardan nuevas versiones en el control de versiones son solo tres ejemplos. El control de versiones beneficia muchos aspectos de la producción. Los flujos de trabajo de control de versiones evitan el caos a todos los usuarios que usan su propio proceso de desarrollo con herramientas diferentes e incompatibles. Los sistemas de control de versiones proporcionan permisos y cumplimiento de procesos, por lo que todos permanecen en la misma sintonía.

Plan de pruebas integral

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito de desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupos.



Reglas para el versionado e incidencias

El control de versiones es un sistema que registra todos los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo de modo que puedan recuperar versiones específicas más adelante, se suele utiliza software de control de versiones para todo el código normalmente Git.

Reglas para el versionado e incidencias

Versión	Autor	Descripción	Fecha
Revisión	Juan corona Pilar Barajas Vero Barajas	solucionando un error en la validación del formulario web Arreglada la consulta a la BD que muestra los productos del inventario mejorada la interfaz	17/02/2024
Liberar	Diana Andrade	Revisión del código, comparación del código	22/02/2024
Desarrollar	Eduardo Téllez	incluida la validación del formulario web	27/02/2024
Característica	José Padilla	Revisión seguimiento del historial de productos, administración de inventario.	02/03/2024

Conclusión

Algo que sucede normalmente en el mundo del desarrollo es que buscamos maneras para mejorar y optimizar el código, los proyectos y aplicaciones. El detalle es que cosas buenas que ya existen al ser simples suelen pasar desapercibidas. Una de ellas es que el versionado del software o versionado del software simplemente, asignar versión del software depende del país o región. Existen muchas formas de asignar versión al software, cada empresa puede asignar sus reglas y ésta bien, pero como todo en este mundo se busca generar unas reglas y pautas que sirvan como guía. Es importante tener en cuenta que existen diferentes propuestas para asignar versiones, Algo común es realizar el manejo de versiones mediante 3 números: X, Y, Z y cada uno indica cosa diferente.

- El primero(X) se le conoce como versión mayor y nos indica la versión del software.**
- El segundo(Y) se le conoce como versión menor y nos indica nuevas funcionalidades.**
- El tercero(Z) se le conoce como revisión y nos indica que se hizo una revisión del código por algún fallo.**

Ahora que conocemos el significado de cada número viene la pregunta importante ¿Cómo sabemos cuándo cambiarlos y cual cambiar?

- Versión mayor X cuando agregamos nuevas funcionalidades importantes.**
- Versión menor Y. cuando hacemos correcciones menores o cuando arreglamos un error.**
- Revisión Z. cada vez que entregamos el proyecto.**

Referencias

KeepCoding | Los Mejores y Más Completos Bootcamps. (2023b, November 14). KeepCoding Bootcamps. <https://keepcoding.io/>

Wibjorn. (n.d.-c). Microsoft Learn: Build skills that open doors in your career.

<https://learn.microsoft.com/>

EDteam - En español nadie te explica mejor. (n.d.). EDteam - En Español Nadie Te Explica Mejor.

<https://ed.team/>