



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Strategy versioning.

PRESENTADO POR:

Medina Palacios Fernando

GRUPO:

10° B

MATERIA:

Desarrollo móvil integral

CARRERA:

TI. Desarrollo de software multiplataforma

Docente:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 10 de enero del 2025

El versionado de software es una práctica esencial en el desarrollo que permite identificar y gestionar las distintas etapas y actualizaciones de un producto. Una estrategia de versionado bien definida facilita la comunicación entre desarrolladores y usuarios, asegurando claridad sobre las mejoras, correcciones y cambios implementados en cada versión.

Principales estrategias de versionado:

Versionado semántico:

Utiliza un formato de tres números: MAYOR.MENOR. PARCHE.

MAYOR: Se incrementa cuando se realizan cambios incompatibles en la API.

MENOR: Se incrementa al añadir funcionalidades de manera retrocompatible.

PARCHE: Se incrementa al realizar correcciones de errores retrocompatibles.

Esta estrategia proporciona una comprensión clara del estado del software y la naturaleza de los cambios realizados.

Versionado basado en fechas:

Las versiones se identifican utilizando la fecha de lanzamiento, por ejemplo, 2023.05.01.

Es útil para software con actualizaciones frecuentes, permitiendo a los usuarios identificar rápidamente la versión más reciente.

Versionado por número de compilación:

Asigna un número único a cada compilación del software.

Es especialmente útil en entornos de integración y despliegue continuo, facilitando el seguimiento del progreso del desarrollo.

Versionado por ramas:

Utiliza el nombre de la rama de desarrollo como identificador de la versión, por ejemplo, feature-xyz.

Es útil en proyectos con múltiples ramas activas, ayudando a identificar a qué línea de desarrollo pertenece una versión específica.

Versionado híbrido:

Combina dos o más estrategias de versionado, como el versionado semántico junto con fechas.

Es adecuado para proyectos con actualizaciones frecuentes y cambios significativos, proporcionando una visión más completa del estado del software.

Importancia de una estrategia de versionado adecuada:

Es importante tener la claridad en la comunicación ya que esta facilita la comprensión de los cambios y actualizaciones tanto para desarrolladores como para usuarios finales haciendo de esta una gestión eficiente de dependencias Ayudando a otros proyectos o módulos a determinar compatibilidades y requisitos específicos.

A demás el mantenimiento y soporte Permite identificar rápidamente qué versiones incluyen ciertas funcionalidades o correcciones, optimizando el proceso de resolución de problemas.

Seleccionar la estrategia de versionado más adecuada depende de las características del proyecto, la frecuencia de actualizaciones y las necesidades de los usuarios. Una implementación coherente y consistente de la estrategia elegida contribuye significativamente al éxito y mantenibilidad del software a largo plazo.

Bibliografías

Preston-Werner, T. (n.d.). Semantic Versioning 2.0.0. Semantic Versioning. Retrieved January 10, 2025, from <https://semver.org/>

(N.d.-a). Fastercapital.com. Retrieved January 10, 2025, from <https://fastercapital.com/es/tema/diferentes-tipos-de-estrategias-de-control-de-versiones.html>

(N.d.-b). Gitlab.com. Retrieved January 10, 2025, from <https://about.gitlab.com/handbook/engineering/release-management/versioning-strategies/>