UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Topic:

Strategy versioning.

By:

Arguelles Galvez Antonio

Group:

10B

Matter:

Integral Mobile Development

Teacher:

Ray Brunett Parra Galaviz

Date:

01/09/2025

Strategy Versioning is the process of managing different versions of a system, application, or API over time. It ensures that changes, updates, and new features are implemented without disrupting existing functionality for users or other systems. This practice is essential for maintaining compatibility, improving user experience, and supporting long-term scalability.

Key Versioning Strategies:

Semantic Versioning (SemVer)

Uses a Major.Minor.Patch format (e.g., 1.2.3).

- **Major:** Incompatible changes.
- **Minor**: New features without breaking changes.
- **Patch**: Bug fixes or small improvements.
- **Example**: If a breaking change is introduced, version 1.2.0 becomes 2.0.0.

Time-Based Versioning

Versions are based on release dates (e.g., 2025.01 for a January 2025 release).

Ensures regular, predictable updates.

Feature-Based Versioning

Versions are updated when significant features are added or modified.

API Versioning

Common in web services to ensure compatibility across different API clients.

Example strategies:

Header-based: Use a version header in the request.

Backward Compatibility

Ensures new versions do not break existing functionality.

Helps maintain a positive user experience and reduces migration risks.

Versioning in React Native Projects

For React Native applications, versioning is essential to manage dependencies, libraries, and API integrations. Using Semantic Versioning is recommended to maintain compatibility and clearly communicate changes.

Advantages of Strategy Versioning

- **Ensures compatibility**: Updates can be made without breaking previous versions.
- **Improves change management:** Makes it easier to introduce new features and fix bugs.
- **Provides clarity for users:** Users can easily identify significant updates through version numbers.
- **Simplifies maintenance:** Helps manage multiple software versions simultaneously.

Disadvantages of Strategy Versioning

- **Increases complexity:** Requires more effort to maintain different active versions.
- **Risk of fragmentation:** Users may run different versions, causing inconsistencies.
- **Consumes resources:** Maintaining older versions requires additional time and resources.
- **Potential confusion:** Without a clear strategy, users may struggle to know which version to use.

Conclusion

Strategy Versioning is crucial for managing software updates while ensuring backward compatibility. It improves scalability and user experience by preventing disruptions in existing functionality. However, it requires careful planning to avoid complexity, fragmentation, and confusion. Proper versioning builds user trust and ensures long-term system stability.