UTT
UNIVERSIDAD TECNOLÓGICA DE TIJUANA
GOBIERNO DE BAJA CALIFORNIA

Assigment:

Strategy Versioning

BY:

Daniel Chavez Madrigal

GROUP:

9-B

SUBJECT:

Desarrollo Móvil Integral

PROFESSOR:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 06 de enero del 2025

Strategy Versioning

1. Semantic Versioning: Semantic versioning is a widely adopted practice that uses a three-part version number: major, minor, and patch. Each part conveys specific information about the changes:

- **Major versions** indicate significant changes that may not be backward compatible.
- **Minor versions** add new features in a backward-compatible manner.
- **Patch versions** include backward-compatible bug fixes.

2. Feature Flagging: Feature flagging allows developers to enable or disable specific features within an app. This technique helps in controlled rollouts, minimizing risks by releasing new functionalities to a subset of users before a full deployment.

3. Continuous Integration and Continuous Deployment (CI/CD): CI/CD pipelines automate the process of integrating code changes and deploying them to production. This ensures that updates are released frequently and reliably, improving the overall development workflow.

4. A/B Testing: A/B testing involves comparing two versions of an app to determine which one performs better. This data-driven approach helps in optimizing user experience and making informed decisions about feature rollouts.

5. Rollback Plans: Having a rollback plan is crucial for handling unexpected issues during updates. It allows developers to revert to a previous stable version quickly, minimizing downtime and user disruption.

6. User Feedback Loops: Incorporating user feedback into the versioning strategy helps in identifying and addressing issues promptly. This continuous feedback loop ensures that the app evolves according to user needs and preferences.

7. Monitoring and Analytics: Monitoring app performance and analyzing usage data are essential for understanding the impact of updates. This information helps in making data-driven decisions and improving future versions.

8. Backward Compatibility: Ensuring backward compatibility means that new versions of the app work seamlessly with older versions. This is important for maintaining a positive user experience and avoiding disruptions.

9. Security and Data Privacy: Integrating security and data privacy measures into the versioning strategy is vital for protecting user information and maintaining trust.

Advantages	Disadvantages
Clarity: Clearly communicates the nature of changes (major, minor, patch).	**Complexity:** Requires strict adherence to versioning rules, which can be complex.
Predictability: Helps developers and users understand the impact of updates.	**Overhead:** Managing and maintaining version numbers can add overhead to the development process.
Compatibility: Facilitates backward compatibility and dependency management.	**Misuse:** Incorrect versioning can lead to confusion and integration issues.
Automation: Supports automated tools for dependency resolution and updates.	**Initial Learning Curve:** Developers need to learn and consistently apply the versioning scheme.
Standardization: Provides a standardized approach that is widely recognized and adopted.	**Rigidity:** Can be rigid, making it challenging to handle unconventional versioning needs.

References

Pawar, S. (2023, August 8). *Unleashing the power of advanced versioning strategies for mobile apps*. AST Consulting. https://astconsulting.in/mobile-applications/power-of-advanced-versioning-strategies-for-mobile-apps/

Bansal, R. (2023, August 21). *Staying updated: Advanced versioning strategies for mobile apps*. AST Consulting. https://astconsulting.in/mobile-applications/advanced-versioning-strategies/