Personal Project by Abner Soberon Martinez

Strategic Guide: Software Development Methodologies for Building a Social Network

Project Overview

This document presents a personal project analysis written by Abner Soberon Martinez. The purpose of this document is to evaluate the most appropriate software development methodologies for creating a large-scale, persistent online world: a fully-featured MMORPG. The game will include a robust website for user registration and community interaction, along with in-game systems such as character classes, races, vocation upgrades, dungeons, castle sieges, missions, skill leveling, guild wars, and planned annual content expansions.

Software Development Methodologies Analysis

Waterfall Model

Advantages:

- Clearly defined phases, easy to follow.
- Strong documentation ensures traceability.

Disadvantages:

- Not flexible to changes once phases are complete.
- No working product until the final stage.

Recommended Use: Best for requirement analysis and architectural planning, especially in security modules.

Agile Model

Advantages:

- Highly flexible and adaptive to changes.
- Delivers working increments frequently.

Disadvantages:

- Minimal documentation can lead to knowledge gaps.
- Requires constant client collaboration.

Recommended Use: Ideal for modules like the user feed, blog system, and social interactions.

Incremental Model

Advantages:

- Early prototypes available.
- Manages risk by building core features first.

Disadvantages:

- High resource requirement.

- May cause architectural inconsistencies.

Recommended Use: Great for adding features like groups or community pages progressively.

RAD Model

Advantages:

- Very fast development cycles.

- Code reuse reduces effort and increases productivity.

Disadvantages:

- Not suitable for new or unfamiliar technologies.

- Costly due to specialized tools and skilled staff.

Recommended Use: Suitable for prototyping sections like Instagram, TikTok-style feeds, or UX interfaces.

Spiral Model

Advantages:

- Integrated risk management at every phase.

- Allows ongoing feedback and refinement.

Disadvantages:

- Complex and potentially expensive.

- Heavy on documentation.

Recommended Use: Recommended for validating identity, biometric checks, and core system infrastructure.

Iterative Model

Advantages:

- Detects issues early through constant feedback.

- Allows parallel development of subsystems.

Disadvantages:

- Not ideal for small projects.

- Requires good understanding of the whole system beforehand.

Recommended Use: Effective for large modular systems such as this multi-functional social network.

Recommended Development Plan by Project Phase

Personal Project by Abner Soberon Martinez

Strategic Guide: Software Development Methodologies for Building a Social Network

1. Concept and Requirements Gathering - (Waterfall Model)

Used to gather and define all the functional and narrative requirements, such as races, classes, lore, and key systems.

2. Core System Design - (Spiral Model)

Applied to map out the architecture, databases, and networking components, with strong risk management for scalability.

3. Core Game and Engine Development - (Agile Model)

Agile iterations are used to progressively develop the base gameplay loop, combat system, UI, and multiplayer mechanics.

4. World Building and Content Expansion - (Incremental Model)

Each continent, city, dungeon, or story arc is added incrementally, allowing smooth scaling and testing.

5. Community Features & Web Integration - (RAD Model)

Used for fast development of external web modules such as forums, account pages, and the official blog.

6. Testing, Balancing, and Launch - (Iterative + Agile)

Ongoing testing, player feedback, and balancing in closed/open betas before final launch.

7. Long-Term Maintenance and Updates - (Agile + Incremental)

Yearly updates, patches, and expansion packs will follow this combined methodology to ensure continuity and innovation.